Mattermost Deployment & Operations Guide

Generated from Mattermost Documentation

Table of Contents

ADMINISTRATION GUIDE

Administration Guide 3

Cloud workspace management 8

Compliance with Mattermost 13

- Compliance export 18
- Compliance monitoring 27
- Custom terms of service 34
- Data retention policy 39
- Electronic discovery 47
- Audit Log JSON Schema 53
- Export channel data 65
- Legal Hold 70
- Mattermost Agents Admin Guide 78
- Authentication configuration settings 86
- Bleve search (experimental) 114
- Calls self-hosted deployment 119
- - Mattermost documentation 134
- Cloud workspace subscription, billing, and account settings 138
- Compliance configuration settings 143

Configuring Apache2 as a proxy for Mattermost Server (Unofficial) 151
Store configuration in your database 156
Configuration settings 163
Custom branding tools 168
Customizing Mattermost 173
Deprecated configuration settings 178
Email templates 193
Chinese, Japanese and Korean search 200
Environment configuration settings 206
Environment variables 257
Experimental configuration settings 263
Install Mattermost Boards 285
Integrations configuration settings 290
Manage user surveys 299
Optimize your Mattermost workspace 305
Plugins configuration settings 310
Enable push notifications 333
Enable rate limiting 340
Reporting configuration settings 346
Self-hosted workspace edition and license settings 351
Site configuration settings 356
SMTP email setup 382
System Attributes 389

User management configuration settings 394
Attribute-Based Access Control 406
Customize branding 413
Mattermost error codes 418
Generate a Support Packet 423
Install a license key 431
Migration 436
Monitoring and performance 441
Self-hosted billing 446
Server configuration 451
Server maintenance 456
User attributes 461
User management 467
User provisioning 472
Bulk export tool 477
Cloud Dedicated Bring Your Own Key 486
Mattermost workspace migration 492
Mattermost Cloud data residency 502
Cloud IP Filtering 507
Code signing custom builds 513
Command line tools 518
Configure server health check probes 527
Mattermost feature labels 532

In-product notices 537

Mattermost logging 542

mmctl command line tool 554

Product limits 632

Request a server Health Check 637

Statistics 642

System-wide notifications 649

Manage team and channel members 654

Telemetry 661

User satisfaction surveys 674

AD/LDAP groups 680

AD/LDAP setup 693

Advanced permissions: backend infrastructure 705

Advanced permissions 716

Bulk loading data 724

Certificate-based authentication (Experimental) 740

- Mattermost documentation 745

Connected workspaces 750

Converting OAuth 2.0 Service Providers to OpenID Connect 762

Delegated granular administration 767

Guest accounts 777

Using AD/LDAP synchronized groups to manage team or private channel membership 785

Migrate from Slack 793

Migrate from HipChat to Mattermost 801
Migration guide 807
Migration announcement email 814
Multi-factor authentication 819
SSL client certificate setup 824
Entra ID Single Sign-On 830
GitLab Single Sign-On 836
Google Single Sign-On 842
OpenID Connect Single Sign-On 848
Configure SAML with Microsoft ADFS using Microsoft Windows Server 2016 853
Configure SAML with Microsoft ADFS for Windows Server 2012 864
Before you begin 874
Frequently Asked Questions 879
Configure SAML with Keycloak 885
Configure SAML synchronization with AD/LDAP 894
Configure SAML with Okta 900
Configure SAML with OneLogin 909
SAML Single Sign-On: technical documentation 918
SAML Single Sign-On 926
Provisioning workflows 933
- Mattermost documentation 938
Backing storage benchmarks 943
Collect performance metrics 949

Set server connection details 955
Deploy Prometheus and Grafana for performance monitoring 960
Elasticsearch server setup 967
Ensuring Releases Perform at Scale 973
Enterprise search 979
- Mattermost documentation 987
High availability cluster-based deployment 992
- Mattermost documentation 1009
AWS OpenSearch server setup 1013
Mattermost performance alerting guide 1021
Performance monitoring metrics 1028
Push notification health targets 1037
Redis 1043
Scale Mattermost up to 100000 users 1049
Scale Mattermost up to 15000 users 1055
Scale Mattermost up to 200 users 1061
Scale Mattermost up to 2000 users 1066
Scale Mattermost up to 200000 users 1072
Scale Mattermost up to 30000 users 1078
Scale Mattermost up to 50000 users 1084
Scale Mattermost up to 80000 users 1090
Scale Mattermost up to 90000 users 1096
Scaling for Enterprise 1102

Administrator onboarding tasks 1109
Communicate scheduled maintenance best practices 1117
Downgrade Mattermost Server 1123
Enterprise install and upgrade 1129
Enterprise roll out checklist 1136
Important Upgrade Notes 1146
Notify admin 1180
Open Source Components 1185
Prepare to upgrade Mattermost 1195
Upgrade Mattermost in Kubernetes and High Availability environments 1203
Upgrade Mattermost Server 1212
Welcome email to end users 1221
Upgrade Mattermost 1226

DEPLOYMENT GUIDE

Application architecture 1231
Backup and disaster recovery 1240
Deployment Guide 1251
Deployment troubleshooting 1256
Desktop App Deployment 1261
Desktop managed resources 1266
Desktop App custom dictionaries 1271
Desktop MSI installer and group policy guide 1276

Desktop app troubleshooting 1283
Distribute a custom desktop app 1290
Install desktop app on Linux 1295
Silent Windows desktop distribution 1301
Encryption options 1306
Manually migrate from MySQL to PostgreSQL 1312
Mobile VPN options 1328
Deploy using an EMM provider 1333
Distribute a custom mobile app 1340
Host your own push proxy service 1346
Mobile App Deployment 1352
Mobile apps FAQ 1357
Mobile security features 1370
Mobile deployment troubleshooting 1377
Secure file storage 1384
Automated PostgreSQL migration 1393
Migration guidelines from MySQL to PostgreSQL 1400
Quick Start Evaluation 1408
Deploy Mattermost in Air-Gapped Environments 1414
Enable Email (Recommended) 1423
Step 1: Install Docker 1428
Deploy Mattermost using Containers 1435
Deploy Mattermost on Kubernetes 1445

Deploy Mattermost on Linux 1458
Deploy Mattermost as an Out-of-Band (OOB) Communication System 1471
Docker deployment troubleshooting 1477
(Optional) Use an Image proxy 1482
Before you begin 1487
Before you begin 1493
Prerequisites 1499
Remove Mattermost 1507
Remove Mattermost 1515
Remove Mattermost 1521
Deployment Solution Programs 1527
Prepare your Mattermost Server environment 1533
Prepare your Mattermost MySQL database 1540
Reference Architecture for Mattermost Server 1549
Server deployment planning 1555
(Recommended) Set up an NGINX proxy 1561
(Optional) Set up TLS 1575
PostgreSQL installation troubleshooting 1581
MySQL installation troubleshooting 1586
General deployment troubleshooting 1594
Software and hardware requirements 1605
Configuring transport encryption 1613

SECURITY GUIDE

Secure Mattermost 1623
Dependency Vulnerability Analysis 1628
Mobile Security 1633
Zero Trust with Mattermost 1640
Security Guide 1649

Administration Guide

Welcome to the Mattermost Administration Guide. This guide is organized into sections based on administrative tasks and scenarios to help you effectively manage and optimize your Mattermost workspace.

Whether you’re configuring server settings, managing users, monitoring performance, or ensuring compliance, this guide provides all the information you need. Use the navigation below to access detailed instructions and best practices for each topic.

- Self-hosted billing - Billing and payment options for Mattermost self-hosted deployments.
- Cloud workspace management - Learn how to manage cloud workspaces in Mattermost.
- Server maintenance - Learn about Mattermost server maintenance and best practices.
- Server configuration - Learn about server configuration and settings.
- User provisioning - Learn about user provisioning and management.
- User management - Learn about user management and best practices.
- Monitoring and performance - Learn about monitoring and performance optimization.

- Compliance - Learn about compliance and security best practices.
- Migration - Learn about migrating to Mattermost.

Cloud workspace management

This section of the guide is for system admins of Mattermost Cloud deployments.

> Tip
>
> If you’re the system admin for a Mattermost self-hosted workspace, see the Self-hosted administration documentation.

- Workspace migration - Migrate your workspace using the mmctl tool.
- Cloud data residency - Find information about your data in the Cloud.
- Cloud IP Filtering - Restrict access to your Mattermost Cloud workspace to a specific IP address range.
- Cloud Bring Your Own Key (BYOK) - Learn how to manage data encryption processes within a Mattermost Cloud Enterprise Dedicated deployment.

Compliance with Mattermost

Mattermost is purpose-built to help enterprises keep sensitive data safe and compliant in the strictest, most complex environments. Mattermost Enterprise Edition includes features designed to make compliance with all relevant regulations and internal policies easy to achieve and maintain.

- Compliance exports - Export compliance reports to third-party systems to archive history.
- Compliance monitoring - Enable oversight and prevent unauthorized queries with compliance exports.
- Electronic discovery - Extract data from Mattermost for eDiscovery.
- Data retention - Control how long data is stored in Mattermost with global and custom retention policies to meet data retention compliance requirements.
- Export channel data - Migrate data between systems and back data up for operational continuity.
- Legal Hold - Preserve relevant Mattermost information when litigation is anticipated.
- JSON audit log schema - Learn how to configure Mattermost audit logging using a JSON object.

Compliance export

Mattermost Enterprise customers can archive history or transfer message data to third-party systems for auditing and compliance purposes with compliance exports. Supported integrations include Actiance Vantage, Global Relay, and Proofpoint.

From Mattermost v10.5, compliance exports include performance improvements for large daily data sets with changes affecting output formats, system performance, and logic. Compliance exports provide compliance teams complete information to reconstruct the state of a channel, and to determine who had visibility on an initial message, or when the message was edited or deleted. Compliance teams can track a message by its MessageId as it is edited or deleted, and across batches and exports periods.

Overview

Compliance exports are produced from the System Console, and contain all messages including:

- Messages sent in direct message channels
- File uploads
- Posts from plugins
- Posts from bots/webhooks

Exports include information on channel member history at the time the message was posted.

Set up guide

Use the following guides to configure exports for CSV, Actiance XML, Global Relay EML, or Proofpoint.

> Note
>
> - For self-hosted deployments, compliance exports are written to the `exports` subdirectory of the configured filestore in the chosen format. This will either be in the Local Storage directory or the Mattermost S3 bucket if S3 storage is configured.
> - Alternatively, you can specify an alternate filestore target and generate an S3 presigned URL for compliance exports. See the dedicated export filestore target configuration settings documentation for details.
> - Compliance exports don't contain posts sent before the feature was enabled. For self-hosted deployments, you can export past history via the `export` command line tool.

CSV

1. Go to **System Console > Compliance > Compliance Export**.
2. Set **Enable Compliance Exports** to **true**.
3. Set the **Compliance Export time**. This is the start time of the daily scheduled compliance export job and must be a 24-hour time stamp in the form `HH:MM`. Choose a time when fewer people are using your system.
4. Set the export file format to **CSV**.
5. Select **Save**.

From Mattermost v10.5

You can review export job status in the System Console.

When the daily compliance export job is finished, a parent directory is created named based on when the export was started and the `startTimestamp` and `endTimestamp` of the export, e.g, `administration-guide/comply/compliance-export-2024-08-13-05h08m-1723105062492-1723109100075`. That parent directory contains 1 zip file for each batch, named based on the batch number and the start and end timestamps of the messages in that batch, e.g,

`batch001-1723105062492-1723106622163.zip`. Each zip file contains the same information available in previous Mattermost server releases.

Working from the same example above, the directory would look like this:

```
administration-guide/comply/compliance-
export-2024-08-13-05h08m-1723105062492-1723109100075 ├──
batch001-1723105062492-1723106622163.zip ├──
batch002-1723106622163-1723108196005.zip └──
batch003-1723108196005-1723109100075.zip
```

And each batch would look like this:

```
batch001-1723105062492-1723106622163.zip ├── files ├── metadata.json └──
actiance_export.xml
```

Updated CSV export fields

Post Creation Time is always the `CreateAt` for messages and attachments, or `JoinTime` and `LeaveTime` for participant join and leave events, respectively.

- **Update Time** indicates that the message has been updated, and this is the `updateAt` time.
- **Updated Type** helps differentiate what kind of update it was as one of the following:
  - **EditedNewMsg** indicates that the message has been edited, and this is the new message (post-edit) content.
  - **EditedOriginalMsg** indicates that the message has been edited, and this the original message (pre-edit) content. This message will have another field `EditedNewMsgId`, which is the Id of the message which holds the post-edited message contents.
  - **UpdatedNoMsgChange** indicates that message's content hasn't changed, but the post was updated for some reason, such as reaction, replied-to, a reply was edited, or a reply was deleted.
  - **Deleted** indicates that this message was deleted.

- **FileDeleted** indicates that this message is recording that a file was deleted.

Prior to Mattermost v10.5

The daily compliance export job creates a `.zip` file with a unique job identifier of all messages posted in the last 24 hours. You can unzip the file to easily transform the default `.csv` format into a desired format for your third-party archive system.

For a sample CSV output, download a CSV export file here.

Actiance XML

1. Go to **System Console > Compliance > Compliance Export**.
2. Set **Enable Compliance Exports** to **true**.
3. Set the **Compliance Export time**. This is the start time of the daily scheduled compliance export job and must be a 24-hour time stamp in the form HH:MM. Choose a time when fewer people are using your system.
4. Set the export file format to **Actiance XML**.
5. Select **Save**.

> Note
>
> In Actiance XML exports, channel type is prepended to the channel names.

From Mattermost v10.5

You can review export job status in the System Console. Once you've selected Actiance XML as your file format, you can set up an integration with Actiance Vantage archive system.

When the daily compliance export job is finished, a parent directory is created named based on when the export was started and the `startTimestamp` and `endTimestamp` of the export, e.g, `administration-guide/comply/compliance-export-2024-08-13-05h08m-1723105062492-1723109100075`. That parent directory

contains 1 zip file for each batch, named based on the batch number and the start and end timestamps of the messages in that batch, e.g, `batch001-1723105062492-1723106622163.zip`. Each zip file contains the same information available in previous Mattermost server releases.

Working from the same example above, the directory would look like this:

```
administration-guide/comply/compliance-
export-2024-08-13-05h08m-1723105062492-1723109100075 ├──
batch001-1723105062492-1723106622163.zip ├──
batch002-1723106622163-1723108196005.zip └──
batch003-1723108196005-1723109100075.zip
```

And each batch would look like this:

```
batch001-1723105062492-1723106622163.zip ├── 20240808 └──
actiance_export.xml
```

Updated Actiance XML export fields

If an XML field is empty, it won't be exported. This is a change from previous Mattermost releases, where empty XML nodes were exported.

- `MessageId` is the unique `messageId`.
- `DateTimeUTC` is always the post's `CreateAt` time.
- `UpdatedDateTimeUTC` indicates that the message has been updated, and this is the `updateAt` time.
- `UpdatedType` helps differentiate what kind of update it was, including:
  - `EditedNewMsg` indicates that this message has been edited, and this is the new message (post-edit) content.
  - `EditedOriginalMsg` indicates that this message has been edited, and this the original message (pre-edit) content. This message will have another field `EditedNewMsgId`, which is the Id of the message which holds the post-edited message contents.

- `UpdatedNoMsgChange` indicates that this message's content hasn't changed, but the post was updated for some reason, such as a reaction, replied-to, a reply was edited, or a reply was deleted.
- `Deleted` indicates that the message was deleted.
- `FileDeleted` indicates that the message is recording that a file was deleted.

Prior to Mattermost v10.5

The daily compliance export job creates a `.zip` file with a unique job identifier of all messages posted in the last 24 hours. Once you've selected Actiance XML as your file format, you can set up an integration with Actiance Vantage archive system. For a sample Actiance output, download an Actiance XML export file here.

Global Relay EML

For more information on Global Relay archive system, visit their website.

1. Go to **System Console > Compliance > Compliance Export**.
2. Set **Enable Compliance Export** to **true**.
3. Set the **Compliance Export time**. This is the start time of the daily scheduled compliance export job and must be a 24-hour time stamp in the form HH:MM. Choose a time when fewer people are using your system.
4. Set the **Export Format** to **GlobalRelay EML**.
5. Select **A9/Type 9**, **A10/Type 10**, or **Custom** for the **Global Relay Customer Account**. This is the type of Global Relay customer account your organization has.
   - For **A9/Type 9** and **A10/Type 10** types, set the **Global Relay SMTP username**, **Global Relay SMTP password**, and **Global Relay SMTP email address**, provided by Global Relay.
   - For a **Custom** type, set the **Global Relay SMTP username**, **Global Relay SMTP password**, **Global Relay SMTP email address**, **SMTP Server Name**, and the **SMTP Server Port**, provided by Global Relay. **Custom** type can be used to integrate with Proofpoint.

6. Select **Save**.

> Note
>
> Messages larger than 250 MB will have their attachments removed because they are too large to send to Global Relay. An error is added to the server logs with id `global_relay_attachments_removed`. It includes the post ID the attachments were removed from, as well as the attachment IDs. A ticket is queued to better handle large messages (https://mattermost.atlassian.net/browse/MM-10038).

Once you've selected Global Relay EML as your file format, you can set up an integration with Global Relay archive system. For more information, see Global Relay Archive (https://www.globalrelay.com/products/archive-data-compliance/).

Proofpoint

1. Go to **System Console > Compliance > Compliance Export**.
2. Set **Enable Compliance Export** to **true**.
3. Set the **Compliance Export time**. This is the start time of the daily scheduled compliance export job and must be a 24-hour time stamp in the form HH:MM. Choose a time when fewer people are using your system.
4. Set the **Export Format** to **GlobalRelay EML**.
5. Select **Custom** for the **Global Relay Customer Account** to integrate with Proofpoint.
6. Set the **SMTP username**, **SMTP password**, **SMTP email address**, **SMTP Server Name**, and the **SMTP Server Port**, provided by Proofpoint.
7. Select **Save**.

See the Global Relay section for details on updated Global Relay export fields. Now you can set up an integration with the Proofpoint archive system. For more information, visit the Proofpoint Archive webiste (https://www.proofpoint.com/us/products/archiving-and-compliance/archive).

Frequently Asked Questions (FAQ)

Are Playbooks and Boards data included in compliance exports?

No. Compliance exports only include channel messages, direct messages, file uploads, and posts from plugins/bots/webhooks. Data from Mattermost Playbooks (including run activities, status updates, and retrospectives) and Mattermost Boards (including cards, comments, and board activities) are not included in the compliance export functionality. Organizations requiring compliance archiving of Playbooks and Boards data should consider separate data retention strategies for these features.

How do I export past history?

Run the `export` command line tool. You can specify an `exportFrom` option to export data from a specified timestamp. All posts that were made after this timestamp will be exported.

What happens if I export data manually?

If the compliance export job is run automatically, manually via the System Console, or manually via the CLI (without the `--exportFrom` option), it exports all posts that were made since the last post that the previous execution of the job exported. If this is the first time that the job has ever run, all posts that were made since the feature was enabled will be exported.

If the `--exportFrom` option is specified with the CLI command, all posts that have been made since the supplied timestamp will be exported.

When run manually via the System Console, `.csv` and Actiance XML files are written to the `exports` subdirectory of the configured Local Storage Directory. Files will be written to a folder with names based on an epoch time range. Global Relay EML export format files will be mailed to the configured email address when run manually.

Is there a maximum row limit for CSV files?

No. There's no limit to the number of rows within Compliance Monitoring CSV files.

How do I know if a compliance export job fails?

Mattermost provides the status of each compliance export job in **System Console > Compliance > Compliance Export**. Here, you can see if the job succeeded or failed, including the number of messages and files exported.

In addition, any failures are returned in the server logs for self-hosted deployments. The error log begins with the string `Failed job` and includes a `job_id key/value` pair. Compliance export job failures are identified with worker name `MessageExportWorker`. You can optionally create a script that programmatically queries for such failures and notifies the appropriate system.

> Note
>
> This compliance export feature replaces legacy Compliance Reporting Oversight functionality. We recommend Enterprise customers migrate to the new system. For a sample CSV output of the new compliance export system, download a CSV export file here.

On this page

- Compliance export
- Compliance export
  - Overview
  - Set up guide
    - CSV
      - From Mattermost v10.5
        - Updated CSV export fields
      - Prior to Mattermost v10.5
    - Actiance XML
      - From Mattermost v10.5
        - Updated Actiance XML export fields

        - Prior to Mattermost v10.5
    - Global Relay EML
    - Proofpoint
- Frequently Asked Questions (FAQ)
    - Are Playbooks and Boards data included in compliance exports?
    - How do I export past history?
    - What happens if I export data manually?
    - Is there a maximum row limit for CSV files?
    - How do I know if a compliance export job fails?
- Overview
- Set up guide
    - CSV
    - Actiance XML
    - Global Relay EML
    - Proofpoint
- Frequently Asked Questions (FAQ)
    - Are Playbooks and Boards data included in compliance exports?
    - How do I export past history?
    - What happens if I export data manually?
    - Is there a maximum row limit for CSV files?
    - How do I know if a compliance export job fails?

Compliance monitoring

This feature enables compliance exports to be produced from the System Console, with all query and download actions logged in an audit history to enable oversight and prevent unauthorized queries.

Compliance exports can be filtered to date range, user account, and keyword list. Requests from queries can be downloaded from the user interface in `.csv` format, with a `.json` metafile documenting the query, as well as placed in a directory set by the system admin.

Daily compliance reports may also be generated, supporting integration with compliance solutions like Global Relay.

By default, all Mattermost deployments retain all messages, including edits and deletes, along with all files uploaded.

Enable compliance reporting

To enable the option to generate daily compliance reports:

1. Go to **System Console > Compliance > Compliance Monitoring** and set the **Enable Compliance Reporting** value to **true**.
2. (Optional) In **Compliance Report Directory** specify the directory in which to place completed compliance reports. Defaults to `./data/` if left blank.
3. Select **Save**.

Turn on daily compliance reports

After enabling compliance reporting:

1. Go to **System Console > Compliance > Compliance Monitoring** and set the **Enable Daily Report** value to **true**.
2. Select **Save**.

Your system will now export all new messages posted within a 24-hour period as a `.csv` file to the location specified in **Compliance Report Directory**. This feature can be used in conjunction with centralized compliance reporting systems that move.

Run compliance reports

Compliance reports are exports of all messages in Mattermost that match the report criteria. To run a report:

1. Go to **System Console > Compliance > Compliance Monitoring**.
2. Fill in the following:

> - ***Job Name:*** *Name the compliance report you are about to run (e.g. "HR Audit 455").*
> - ***From:*** *Start date for search in YYYY-MM-DD format (e.g. "2016-03-11").*
> - ***To:*** *End date of search in YYYY-MM-DD format (e.g. "2016-05-11").*
> - ***Emails:*** *Comma-separated list of email addresses of users whose posted messages you want to search (e.g.* `bill@example.com, bob@example.com`*).*
> - ***Keywords:*** *Indicate the words that would be contained in a message for it to be included in the compliance report results.*

3. Select **Run Compliance Report**.

The report will be queued in the display below the fields described above. The properties of each compliance report run is explained as follows:

- **Timestamp:** Time at which the report was requested.
- **Status:** `running` indicates the report is being run; `finished` indicates the report is complete and ready for download.
- **Records:** Shows the number of search results.
- **Type:** `adhoc` indicates the report was requested by completing query fields; `daily` indicates the report is a daily export.

- **Description:** Job Name indicated in request.
- **Requested by:** Email of person requesting the report.
- **Params:** Parameters of the compliance report request.

Each compliance report includes a **Download** link which downloads a compressed file named `adhoc-[UNIQUE_ID].zip`. Inside the file is `meta.json`, which includes the parameters of the search executed and `posts.csv` which includes the contents of messages found by the request.

Compliance query definition stored in `meta.json`

The `meta.json` file contains the following information about the compliance query:

Field	Description	Example
id	Unique identifier for compliance query	ja8z8egap7nq9kqetz3rt98khe
create_at	Timestamp at which compliance query was executed	1463637842478
user_id	Mattermost User ID for person creating query	3bq1shta93yztg3i6aiu1tzi5h
status	Status of query: *finished* or *failed*	"finished"
count	Count of messages found matching keyword	36
desc	User entered description of compliance query	Example Compliance Report
type	Type of compliance query: *adhoc* or *daily*	"adhoc"
start_at	Timestamp at which query began to run	1451606400000
end_at	Timestamp at which query ended	1463529600000
keywords	Comma-separated, case insensitive keywords to match in query	"drinking"

Field	Description	Example
emails	Comma-separated emails of users to search. Blank returns all	`frank.yu@ha.ca, mary.li@hi.co`

Compliance query results stored in `posts.csv` file

`posts.csv` contains the following information about the compliance query results, one search result per row:

Field	Description	Example
TeamName	URL name of team	contosi
TeamDisplayName	Display name of team	Contosi Corporation
ChannelDisplayName	Display name of channel where keyword was found	Community Heartbeat
ChannelName	URL name of channel	community-heartbeat
UserUsername	Username of user posting the message containing keyword	frank.yu
UserEmail	Email of user posting the message containing keyword	“frank.yu@contosi.com”
UserNickname	Nickname of user posting the message containing keyword	fan du
UserType	Type of user posting the message (“user” or “bot”)	user
PostId	Unique ID of message post containing keyword	xt9anyx6x3fx9y84aehgakdpze
PostCreateAt	Timestamp at which post was created	2016-03-02T16:01:59Z

Field	Description	Example
PostDeletedAt	Timestamp at which post was deleted (if applicable)	2016-03-02T16:01:59Z
PostUpdatedAt	Timestamp at which post was last edited (if applicable)	2016-03-02T16:01:59Z
PostParentId	Unique ID of parent post if post is a comment	xt9anyx6x3fx9y84aehgakdpze
PostOriginalId	Unique ID of post if deleted or edited	xt9anyx6x3fx9y84aehgakdpze
PostMessage	Message containing keyword	Drinking from the fire hose
PostFilenames	Comma separated list of filesnames attached to post	["/f../ho.png","/f../hi.png"]

Global Relay support

Mattermost daily compliance reports are compatible with Global Relay compliance solutions through the conversion of Mattermost `.CSV` exports into Global Relay `EML` files.

- This conversion can be done by in-house developers who have previously written scripts to convert other communication systems into Global Relay format based on your organization's specific needs.
- You can also contact your Global Relay account manager about a services project to establish this conversion.

We recommend using the new Compliance Export feature for Global Relay exports.

On this page

- Compliance monitoring
  - Enable compliance reporting
  - Turn on daily compliance reports
  - Run compliance reports
    - Compliance query definition stored in meta.json
    - Compliance query results stored in posts.csv file
  - Global Relay support

Custom terms of service

Increase clarity on legal Mattermost expectations for internal employees and guests with the ability to set custom Terms of Service ("ToS") agreements and re-acceptable periods.

Configuring terms of service

To enable custom terms of service:

1. Go to **System Console > Compliance > Custom Terms of Service**.
2. Set **Enable Custom Terms of Service** to **true**.
3. Set **Custom Terms of Service Text** which contains your terms. Markdown formatting is supported, including lists, headings, and bolding.
4. Set **Re-Acceptance Period** to configure the number of days before Terms of Service acceptance expires, and when the terms must be re-accepted. Set to 0 if you don't want your terms to expire.
5. Select **Save**.

Once saved, all users must accept the terms of service by clicking **I Agree** next time they log in, or on the next page refresh. If they do not accept, they will be logged out.

> Note
>
> If you make an update to your Terms of Service, make sure to update your terms of service link at **System Console > Site Configuration > Customization > Terms of Service link**. This link is presented to all users when they log in, and it's easily accessible to end users after accepting the terms.

Frequently asked questions

What happens if I update my terms of service text?

There will be no impact to your end users. Users will simply be required to accept the new terms on next login or page refresh.

What happens if a user doesn't accept the terms of service?

Users won't be able to log in if they don't accept the Terms of Service.

How do I know if my users accepted the terms of service?

Each time a user agrees to the terms of service, the agreement is recorded in the database:

- A new field `AcceptedServiceTermsId` in the **Users** table indicates the most recent terms of service the user has accepted.
- Each time a user accepts the terms, the action is recorded in the **Audits** table, containing the username, timestamp of acceptance, and other relevant information.

Why isn't this feature in Team Edition for GDPR compliance?

Terms of service is presented to users on login and account creation, and available to users at all times in the link available by going to **System Console > Site Configuration > Customization**.

This feature is intended to meet compliance requirements for large Enterprise companies.

On this page

- Custom terms of service
  - Configuring terms of service

- Frequently asked questions
  - What happens if I update my terms of service text?
  - What happens if a user doesn’t accept the terms of service?
  - How do I know if my users accepted the terms of service?
  - Why isn’t this feature in Team Edition for GDPR compliance?

Data retention policy

Mattermost stores all message history providing an unlimited search history to system admins and end users.

Mattermost Enterprise customers can set a global retention policy as well as custom retention policies to manage how long messages and file uploads are kept in Mattermost channels and direct messages in order to meet data retention compliance requirements.

> Warning
>
> - By default, Mattermost uses a "soft delete" system where messages and files deleted based on user actions are removed from the user interface, but persist in the Mattermost database.
> - Once a message or a file is permanently deleted, the action is irreversible. We recommend using caution when setting up global or custom data retention policies to avoid data loss.

Configure a global data retention policy

To set a global data retention policy:

1. Go to **System Console > Compliance > Data Retention Policies**.
2. Select **Edit** from the menu located to the right of the **Global retention policy** table.
3. Specify a global retention policy for channel messages and direct messages by selecting a **Channel & direct message retention** option from the dropdown, then set how long to keep those messages in hours, days, or years. When a time is set, messages and file attachments older than the duration you set will be deleted. The minimum retention period is one hour.
4. Select a **File retention** option from the dropdown. Set the number of hours, days, or years to keep files. When a time is set, uploaded files which

are older than the duration you set will be deleted from your file storage system (either from your local disk or your Amazon S3 service as specified in **System Console > Environment > File Storage**). The minimum retention period is one hour. The global file policy deletes all files regardless of whether they're in a direct message, private, or public channel.

5. From Mattermost v10.10, you can optionally preserve pinned posts when data retention policies delete messages by enabling Preserve pinned posts. Once enabled, pinned posts won't be deleted even if they exceed the configured retention period.

6. Under the **Policy log** section, select **Edit** to specify the start time of the daily scheduled data retention job. Choose a time when fewer people are using your system.

Select **Save**. Messages and files older than the duration you set will be deleted at the specified server time, as applicable.

Configure a custom data retention policy

To set a custom data retention policy:

1. Go to **System Console > Compliance > Data Retention Policies**.

2. Select **Add policy** to the right of the **Custom retention policies** table.

3. Specify a name for your policy.

4. Specify a custom retention policy for channel and direct messages by selecting a **Channel & direct message retention** option from the dropdown, then set how long to keep uploaded files in days or years. When a time is set, messages and file attachments older than the duration you set will be deleted. The minimum retention period is one day.

5. Assign teams and channels to this policy by selecting **Add teams** and searching for a specific team, or by selecting **Add channels** and searching for a specific channel. If only teams are specified, all channels for selected teams will be included in the a policy.

6. Under the **Policy log** section, select **Edit** to specify the start time of the daily scheduled data retention job. Choose a time when fewer people are

using your system. If a time is already set for a global retention policy, then the same time applies to custom data retention policies.

Save the settings. Messages and files older than the duration you set will be deleted at the specified server time, as applicable.

Run a deletion job manually

You can also run the deletion job manually at any time by selecting **Run Deletion Job Now** in **System Console > Compliance > Data Retention Policy**.

> Note
>
> If using data retention with ElasticSearch, ensure the ElasticSearch aggregate search indexes setting is set to a value that is greater than your data retention policy in days.

Frequently Asked Questions (FAQs)

What happens when a message is deleted?

The message is removed from the Mattermost user interface and deleted from the `Posts` table. The message is no longer searchable and cannot be retrieved in saved posts lists.

Replies that did not exceed the message duration are still displayed in the user interface. However, further replies are no longer possible.

If there was a file attached to the message, it will be removed from the user interface only.

Why can I still see messages that were supposedly deleted?

The web and desktop app cache posts. Posts deleted by a data retention job will be visible to end users until they clear their cache and refresh.

What happens when a file is deleted by the file retention policy?

The file attachment is removed from the Mattermost user interface, deleted from the `FileInfo` table, and from your local disk or Amazon S3 service as specified in **System Console > Environment > File Storage**.

Why do I see `Pending` in the deletion job table with no details?

This usually means another data retention job is in progress. You can verify this in the deletion job table in **System Console > Compliance > Data Retention Policy**.

How is data retention handled in the mobile apps?

When messages or files are deleted, they are no longer searchable in the Mattermost mobile apps.

In v1.5 and later of the iOS and Android apps, messages and files are deleted from local storage in the following cases, if they exceed the retention policy duration:

1. When the user opens the app.
2. When the user puts the app into the background.

In v1.4 and earlier of the mobile apps, messages and files are not cleared from local storage when the data retention policy is enabled.

How do I know if a data retention job fails?

Mattermost provides the status of each data retention job in **System Console** > **Compliance** > **Data Retention Policy**. Here, you can see if the job succeeded or failed, including the details of the error.

Additionally, any failures are returned in the server logs. The error log begins with the string `Failed job` and includes a job_id key/value pair. Data retention job failures are identified with worker name `EnterpriseDataRetention`. You can optionally create a script that programmatically queries for such failures and notifies the appropriate system.

What happens when the data retention period is changed?

Data retention runs once a day at the time specified in the `config.json` file. Changing the retention period does not automatically schedule any additional run of the data retention job - it only updates how long data is kept in Mattermost.

Does the system administrator get any notification when the data retention period is changed?

No, the new config is updated, but the system admin does not receive any feedback on what the effects will be (e.g. reporting of how many messages are to be deleted).

Does the data retention job affect the audits table?

Prior to v5.20, data retention would delete all user activity corresponding to the data retention time configuration. From v5.20, the audit table will retain the user activity corresponding to the data retention time configuration.

Does the data retention job include archived channels?

Posts and attachments in archived channels are affected by the data retention job. If a post exceeds the age configured for the data retention job it will be deleted from the database.

How long does it take to run a deletion query and does it affect server performance?

Data retention runs the actual deletion query in batches, deleting data in blocks of 1000 records per query. This is so the database won't be locked up for extended periods of time with long-running queries. Keeping to this limit keeps the query down to a few milliseconds' execution time on the database itself.

Each batch of data is deleted based on indexes - making the queries quick to execute on small batches. This helps the server remain fully responsive while the process is running.

How do I know whether the data retention job is running/scheduled?

The job scheduler runs the data retention job based on the time specified in the configuration settings. At this time a `DEBUG`-level log line is printed: `Scheduling data retention job`.

When a job server picks up that scheduled job for execution, a `DEBUG`-level log line is generated: `Worker EnterpriseDataRetention: Received a new candidate job`.

When the job is complete, an `INFO`-level log line is generated: `Worker EnterpriseDataRetention: Job is complete`.

On this page

- Data retention policy
  - Configure a global data retention policy
  - Configure a custom data retention policy
  - Run a deletion job manually
  - Frequently Asked Questions (FAQs)
    - What happens when a message is deleted?
    - Why can I still see messages that were supposedly deleted?
    - What happens when a file is deleted by the file retention policy?
    - Why do I see Pending in the deletion job table with no details?
    - How is data retention handled in the mobile apps?
    - How do I know if a data retention job fails?
    - What happens when the data retention period is changed?
    - Does the system administrator get any notification when the data retention period is changed?
    - Does the data retention job affect the audits table?
    - Does the data retention job include archived channels?
    - How long does it take to run a deletion query and does it affect server performance?
    - How do I know whether the data retention job is running/scheduled?

Electronic discovery

Electronic discovery (also known as eDiscovery) refers to a process where electronic data is searched with the intent to use it as evidence in a legal case.

This page describes how to extract data from Mattermost for eDiscovery. There are three primary methods that can be used to accomplish the goal of extracting user post data from Mattermost:

- Mattermost Compliance Exports
- Mattermost RESTful API
- Mattermost database using standard SQL queries

Each of the options is discussed in detail below.

> Note
>
> Litigation hold (also known as legal hold) is an extension of eDiscovery where in addition to searching for records, no electronically stored information nor paper documents may be discarded if they may be deemed relevant for a present or future legal case.

Mattermost compliance exports

Mattermost Enterprise has compliance report export capabilities.

Mattermost can export compliance related data, including the content of messages and who might have seen those messages, in three formats: Actiance XML, Global Relay EML, and generic CSV. Reports can be configured to run on a delay basis and stored in a shared location.

For more information about the exports feature and how to set up reporting, see our documentation.

Mattermost RESTful API

The Mattermost API can be used to export a user's posts in CSV compliance format that is part of Mattermost Enterprise. The following section outlines how to use the API to create and retrieve a report for a specific user via the API. Please note that full documentation for the Mattermost API can be found at https://api.mattermost.com.

To use the API, you must first authenticate [as described here (https://api.mattermost.com/#tag/authentication)](https://api.mattermost.com/#tag/authentication). The account you are authenticating with must have `manage_system` permissions. If you are using curl, you can authenticate using the following command:

```
curl -i -d '{"login_id": "username", "password": "password"}' https://
yourmattermosturl/api/v4/users/login
```

Mattermost will return a response that looks like this:

```
HTTP/2 200 server: nginx/1.10.3 (Ubuntu) date: Thu, 12 Jul 2018 14:43:45 GMT
content-type: application/json content-length: 679 set-cookie:
MMAUTHTOKEN=yi94pwci6ibjfc9phbikhqutbe; Path=/; Expires=Sat, 11 Aug 2018
14:43:45 GMT; Max-Age=2592000; HttpOnly; Secure set-cookie:
MMUSERID=qfjzamfg47bu9gsyyfbqjk4s6a; Path=/; Expires=Sat, 11 Aug 2018
14:43:45 GMT; Max-Age=2592000; Secure token: yi94pwci6ibjfc9phbikhqutbe x-
request-id: 5y45pxyfxpb8updtge1zts39he x-version-id:
5.0.0.5.0.1.18c0fbd759664a85fe95132bb7fc04cf.true
```

Include the `token` value sent in the response as part of the Authorization header on your future API requests with the Bearer method, e.g.:

```
curl -i -H 'Authorization: Bearer yi94pwci6ibjfc9phbikhqutbe http://
yourmattermosturl/api/v4/users/me'
```

Once you're authenticated into Mattermost, you can use the [Compliance API to create a new compliance report (https://api.mattermost.com/#tag/compliance%2Fpaths%2F~1compliance~1reports%2Fpost)](https://api.mattermost.com/#tag/compliance%2Fpaths%2F~1compliance~1reports%2Fpost). The curl based example below demonstrates how to send a request that bases the authentication token and asks Mattermost to create a report that spans posts from Dec 31, 2017 - 8:15 PM to Dec 31, 2018 - 8:15 PM for a user with the email address craig@mattermost.com:

> Note
>
> The data in the JSON payload must be formatted as Unix Epoch. A tool like https://www.epochconverter.com/ can be useful when converting to and from the required format.

```
curl --header "Content-Type: application/json" \ --request POST \ -H
'Authorization: Bearer yi94pwci6ibjfc9phbikhqutbe' \ --data
'{"id":"","create_at":0,"user_id":"craig","status":"","count":0,"desc":"
","type":"","start_at":1514769359000,"end_at":
1546305359000,"keywords":"","emails":"craig@mattermost.com"}' \ https://
yourmattermosturl/api/v4/compliance/reports
```

If the post is successful, Mattermost will return a message that looks like the following, indicating that the server is running the compliance export process:

```
{"id":"du6kektczifqxexeroywpz3nbc"," create_at":1531444617901,
"user_id":"qfjzamfg47bu9gsyyfbqjk4s6a", "status":"running", "count":0,
"desc":" ", "type":"adhoc", "start_at":1514769359000,
"end_at":1546305359000, "keywords":"", "emails":"craig@mattermost.com"}
```

When the export process is complete (the execution time is based on the number of records to return and the current server load), you will need to send another HTTP Post request to Mattermost to retrieve and download a zip file containing the report that looks like the following curl request:

```
curl --request GET \ -H 'Authorization: Bearer p9o1qx457fbc9gdrn39z9ah59o'
\ --data
'{"status_code":0,"id":"du6kektczifqxexeroywpz3nbc","message":"","requestion
_id":""}' \ --output report-zip.zip \ https://yourmattermosturl/api/v4/
compliance/reports/du6kektczifqxexeroywpz3nbc/download
```

When sending the request, you need to get the report ID from the response returned by Mattermost when the report was created. You also need to supply a name to save that file as. In the example above, the file will be saved as `report-zip.zip`.

Mattermost database

Selecting messages from the Mattermost database using standard SQL is quite easy. If you know the username, the following query can be used to select all messages for the specified user:

```
SELECT * FROM mattermost.Posts WHERE UserId = (SELECT Id FROM
mattermost.Users WHERE Username = 'username');
```

If you want to limit the results of the query based on the date and time that the messages were posted, you can modify the above query to:

```
SELECT * FROM mattermost.Posts WHERE UserId = (SELECT Id FROM
mattermost.Users WHERE Username = 'username' AND CreateAt > 1530405832000
AND CreateAt < 1532997832000);
```

> Note
>
> The Mattermost database stores date and time stamps in the Unix Epoch (https://en.wikipedia.org/wiki/Unix_time) format. A tool like the Epoch Converter (https://www.epochconverter.com/) can be useful in converting to and from the required format.

On this page

- Electronic discovery
  - Mattermost compliance exports
  - Mattermost RESTful API
  - Mattermost database

Audit Log JSON Schema

The audit log JSON schema functions as a standardized blueprint or schematic that consistently defines how a single event should appear when being written to the audit log, including: field names, data types, objects, and structure.

An outline of the JSON audit logging schema is provided below. See the JSON data model for additional details.

```
{  "timestamp": "", // Event time  "status": "", // Success or failure of
the audited event or activity  "event_name": "", // Logged event name
"error": { // Error if status = fail  "status_code": 0,  "description":
""  },  "actor": { // The user performing the action  "user_id": "" //
Unique identifier of the event user  "session_id": "" // Unique session
identifier of the event user  "client": "" // User agent of the client/
platform in use by the event user  "ip_address": "" // IPv4/IPv6 IP address
of the event user  },  "event": { // Event-specific data  "parameters":
{} // Map containing parameters of the audited event or activity
"prior_state": {} // Pre-event state of the object  "resulting_state":
{} // Post-event state of the object  "object_type": "" // Object targeted
by the event or activity  },  "meta": {  "api_path": "", // API endpoint
interacted with for event or activity  "cluster_id":
"" // Unique identifier of the cluster in use by the event user  } }
```

Audit log record examples

Update user preferences

```
{  "timestamp": "2022-08-17 20:37:52.846 +01:00",  "event_name":
"updatePreferences",  "status": "success",  "actor": {  "user_id":
"aw8ehkwaziytzry1qqxi9tsqwh",  "session_id": "kth3jyadc3b1p84kbz6y3o75na",
"client": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/
605.1.15 (KHTML, like Gecko) Version/15.6 Safari/605.1.15",  "ip_address":
"192.168.0.169"  },  "event": {  "parameters": {},  "prior_state": {},
"resulting_state": {},  "object_type": ""  },  "meta": {  "api_path": "/api/
v4/users/aw8ehkwaziytzry1qqxi9tsqwh/preferences",  "cluster_id":
"8dxdbfx6fpdwtki1z6n8whtkho"  },  "error": {} }
```

Create post


```
{ "timestamp": "2025-04-30 16:17:44.207 Z", "event_name": "createPost",
"status": "success", "actor": { "user_id": "i764hi6h5bbz8p1955ed4ahj6y",
"session_id": "t7894ft76igtpb788nkkej1yoy", "client": "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/134.0.0.0 Safari/537.36", "ip_address": "172.19.0.8" }, "event":
{ "parameters": { "post": { "channel_id": "pfis7ycuy78o7m3zebajmxqeuo",
"user_id": "i764hi6h5bbz8p1955ed4ahj6y", "message": "Sample post
content" } }, "prior_state": {}, "resulting_state": { "channel_id":
"pfis7ycuy78o7m3zebajmxqeuo", "create_at": 1746029864145, "id":
"xpw97hf6kfncirzhqisb5sym7e", "user_id": "i764hi6h5bbz8p1955ed4ahj6y" },
"object_type": "post" }, "meta": { "api_path": "/api/v4/posts",
"cluster_id": "i5twhjm3ainatcifiy3oksshae" }, "error": {} }
```


System configuration change


```
{ "timestamp": "2025-04-30 16:18:30.803 Z", "event_name": "patchConfig",
"status": "success", "actor": { "user_id": "i764hi6h5bbz8p1955ed4ahj6y",
"session_id": "t7894ft76igtpb788nkkej1yoy", "client": "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/134.0.0.0 Safari/537.36", "ip_address": "172.19.0.8" }, "event":
{ "parameters": {}, "prior_state": { "config_diffs": [ { "actual_val":
false, "base_val": true, "path":
"MetricsSettings.EnableClientMetrics" } ] }, "resulting_state": {},
"object_type": "config" }, "meta": { "api_path": "/api/v4/config/
patch", "cluster_id": "i5twhjm3ainatcifiy3oksshae" }, "error": {} }
```


Audit event types

The following tables list the comprehensive audit event types (`event_name` values) that are captured in Mattermost audit logs:

User Management Events

Event Name	Description
`attachDeviceId`	Attaching device IDs to user sessions
`createUser`	Creating new user accounts
`createUserAccessToken`	Creating user access tokens
`deleteUser`	Deleting user accounts
`demoteUserToGuest`	Demoting users to guest status
`disableUserAccessToken`	Disabling user access tokens
`enableUserAccessToken`	Enabling user access tokens
`followThreadByUser`	Following message threads by user
`getUserAudits`	Retrieving user audit logs
`login`	User login events
`logout`	User logout events
`migrateAuthToLdap`	Migrating user authentication to LDAP
`migrateAuthToSaml`	Migrating user authentication to SAML

Event Name	Description
`patchUser`	Updating user properties
`promoteGuestToUser`	Promoting guest users to regular users
`resetPassword`	Resetting user passwords
`resetPasswordFailedAttempts`	Resetting password failed attempt counters
`revokeUserAccessToken`	Revoking user access tokens
`sendPasswordReset`	Sending password reset emails
`sendVerificationEmail`	Sending email verification messages
`setDefaultProfileImage`	Setting default profile images
`setProfileImage`	Setting custom profile images
`setUnreadThreadByPostId`	Setting unread thread status by post ID
`switchAccountType`	Switching account types
`unfollowThreadByUser`	Unfollowing message threads by user
`updatePassword`	Updating user passwords

Event Name	Description
`updateReadStateAllThreadsByUser`	Updating read state for all threads by user
`updateReadStateThreadByUser`	Updating read state for specific threads by user
`updateUser`	Updating user account information
`updateUserActive`	Updating user active/inactive status
`updateUserAuth`	Updating user authentication settings
`updateUserMfa`	Updating user multi-factor authentication
`updateUserRoles`	Updating user roles and permissions
`verifyUserEmail`	Verifying user email addresses
`verifyUserEmailWithoutToken`	Verifying user email without token

Channel Management Events

Event Name	Description
`addChannelMember`	Adding members to channels
`convertGroupMessageToChannel`	Converting group messages to channels
`createChannel`	Creating new channels
`createChannelBookmark`	Creating channel bookmarks
`createDirectChannel`	Creating direct message channels
`createGroupChannel`	Creating group message channels
`deleteChannel`	Deleting channels
`deleteChannelBookmark`	Deleting channel bookmarks
`moveChannel`	Moving channels between teams
`patchChannel`	Updating channel properties
`patchChannelModerations`	Updating channel moderation settings
`removeChannelMember`	Removing members from channels
`restoreChannel`	Restoring deleted channels

Event Name	Description
`updateChannel`	Updating channel information
`updateChannelBookmark`	Updating channel bookmarks
`updateChannelBookmarkSortOrder`	Updating channel bookmark sort order
`updateChannelMemberNotifyProps`	Updating channel member notification properties
`updateChannelMemberRoles`	Updating channel member roles
`updateChannelMemberSchemeRoles`	Updating channel member scheme roles
`updateChannelPrivacy`	Updating channel privacy settings
`updateChannelScheme`	Updating channel permission schemes

Team Management Events

Event Name	Description
`addTeamMember`	Adding members to teams
`addTeamMembers`	Adding multiple members to teams
`addUserToTeamFromInvite`	Adding users to teams from invitations
`createTeam`	Creating new teams
`deleteTeam`	Deleting teams
`importTeam`	Importing team data
`invalidateAllEmailInvites`	Invalidating all email invitations
`inviteGuestsToChannels`	Inviting guests to channels
`inviteUsersToTeam`	Inviting users to teams
`patchTeam`	Updating team properties
`regenerateTeamInviteId`	Regenerating team invitation IDs
`removeTeamIcon`	Removing team icons
`removeTeamMember`	Removing members from teams

Event Name	Description
`restoreTeam`	Restoring deleted teams
`setTeamIcon`	Setting team icons
`updateTeam`	Updating team information
`updateTeamMemberRoles`	Updating team member roles
`updateTeamMemberSchemeRoles`	Updating team member scheme roles
`updateTeamPrivacy`	Updating team privacy settings
`updateTeamScheme`	Updating team permission schemes

Posts & Content Events

Event Name	Description
`createPost`	Creating new posts
`createSchedulePost`	Creating scheduled posts
`deletePost`	Deleting posts
`deleteScheduledPost`	Deleting scheduled posts
`moveThread`	Moving message threads
`patchPost`	Updating post properties
`restorePostVersion`	Restoring previous post versions
`saveIsPinnedPost`	Saving pinned post status
`searchPosts`	Searching through posts
`updatePost`	Updating post content
`updateScheduledPost`	Updating scheduled posts

Authentication and Security Events

Event Name	Description
`addLdapPrivateCertificate`	Adding LDAP private certificates
`addLdapPublicCertificate`	Adding LDAP public certificates
`addSamlIdpCertificate`	Adding SAML IDP certificates
`addSamlPrivateCertificate`	Adding SAML private certificates
`addSamlPublicCertificate`	Adding SAML public certificates
`completeSaml`	Completing SAML authentication
`extendSessionExpiry`	Extending session expiry times
`idMigrateLdap`	Migrating IDs to LDAP
`linkLdapGroup`	Linking LDAP groups
`removeLdapPrivateCertificate`	Removing LDAP private certificates
`removeLdapPublicCertificate`	Removing LDAP public certificates
`removeSamlIdpCertificate`	Removing SAML IDP certificates
`removeSamlPrivateCertificate`	Removing SAML private certificates

Event Name	Description
`removeSamlPublicCertificate`	Removing SAML public certificates
`revokeAllSessionsAllUsers`	Revoking all sessions for all users
`revokeAllSessionsForUser`	Revoking all sessions for specific user
`revokeSession`	Revoking individual sessions
`syncLdap`	Synchronizing LDAP data
`unlinkLdapGroup`	Unlinking LDAP groups

System Administration Events

Event Name	Description
`clearServerBusy`	Clearing server busy status
`completeOnboarding`	Completing system onboarding
`configReload`	Reloading system configuration
`databaseRecycle`	Recycling database connections
`downloadLogs`	Downloading system logs
`getAppliedSchemaMigrations`	Getting applied schema migrations
`getAudits`	Retrieving audit logs
`getConfig`	Getting system configuration
`getLogs`	Getting system logs
`getOnboarding`	Getting onboarding status
`invalidateCaches`	Invalidating system caches
`migrateConfig`	Migrating configuration
`patchConfig`	Updating configuration properties

Event Name	Description
`queryLogs`	Querying system logs
`restartServer`	Restarting server
`setServerBusy`	Setting server busy status
`updateConfig`	Updating system configuration
`updateViewedProductNotices`	Updating viewed product notices
`upgradeToEnterprise`	Upgrading to Enterprise edition

File Management Events

Event Name	Description
`createUpload`	Creating file uploads
`getFile`	Retrieving files
`getFileLink`	Getting file links
`uploadData`	Uploading data
`uploadFileMultipart`	Uploading multipart files
`uploadFileMultipartLegacy`	Uploading legacy multipart files
`uploadFileSimple`	Uploading simple files

OAuth Applications Events

Event Name	Description
`authorizeOAuthApp`	Authorizing OAuth applications
`authorizeOAuthPage`	OAuth authorization page access
`completeOAuth`	Completing OAuth flow
`createOAuthApp`	Creating OAuth applications
`deauthorizeOAuthApp`	Deauthorizing OAuth applications
`deleteOAuthApp`	Deleting OAuth applications
`getAccessToken`	Getting OAuth access tokens
`loginWithOAuth`	Login with OAuth
`mobileLoginWithOAuth`	Mobile login with OAuth
`regenerateOAuthAppSecret`	Regenerating OAuth app secrets
`signupWithOAuth`	Signup with OAuth
`updateOAuthApp`	Updating OAuth applications

Webhooks Events

Event Name	Description
`createIncomingHook`	Creating incoming webhooks
`createOutgoingHook`	Creating outgoing webhooks
`deleteIncomingHook`	Deleting incoming webhooks
`deleteOutgoingHook`	Deleting outgoing webhooks
`getIncomingHook`	Getting incoming webhooks
`getOutgoingHook`	Getting outgoing webhooks
`regenOutgoingHookToken`	Regenerating outgoing webhook tokens
`updateIncomingHook`	Updating incoming webhooks
`updateOutgoingHook`	Updating outgoing webhooks

Slash Commands Events

Event Name	Description
`createCommand`	Creating slash commands
`deleteCommand`	Deleting slash commands
`executeCommand`	Executing slash commands
`moveCommand`	Moving slash commands
`regenCommandToken`	Regenerating command tokens
`updateCommand`	Updating slash commands

Plugins Events

Event Name	Description
`disablePlugin`	Disabling plugins
`enablePlugin`	Enabling plugins
`getFirstAdminVisitMarketplaceStatus`	Getting first admin visit marketplace status
`installMarketplacePlugin`	Installing marketplace plugins
`installPluginFromURL`	Installing plugins from URL
`removePlugin`	Removing plugins
`setFirstAdminVisitMarketplaceStatus`	Setting first admin visit marketplace status
`uploadPlugin`	Uploading plugins

Groups & LDAP Events

Event Name	Description
`addGroupMembers`	Adding members to groups
`addUserToGroupSyncables`	Adding users to group syncables
`createGroup`	Creating new groups
`deleteGroup`	Deleting groups
`deleteGroupMembers`	Removing members from groups
`linkGroupSyncable`	Linking group syncables to teams/channels
`patchGroup`	Updating group properties
`patchGroupSyncable`	Updating group syncable properties
`restoreGroup`	Restoring deleted groups
`unlinkGroupSyncable`	Unlinking group syncables from teams/channels

Remote Clusters Events

Event Name	Description
`createRemoteCluster`	Creating remote cluster connections
`deleteRemoteCluster`	Deleting remote cluster connections
`generateRemoteClusterInvite`	Generating invites for remote clusters
`inviteRemoteClusterToChannel`	Inviting remote clusters to channels
`patchRemoteCluster`	Updating remote cluster properties
`remoteClusterAcceptInvite`	Accepting remote cluster invites
`remoteClusterAcceptMessage`	Accepting messages from remote clusters
`remoteUploadProfileImage`	Uploading profile images from remote clusters
`uninviteRemoteClusterToChannel`	Removing remote cluster invites from channels
`uploadRemoteData`	Uploading data from remote clusters

Data Retention Events

Event Name	Description
`addChannelsToPolicy`	Adding channels to data retention policies
`addTeamsToPolicy`	Adding teams to data retention policies
`createPolicy`	Creating data retention policies
`deletePolicy`	Deleting data retention policies
`patchPolicy`	Updating data retention policies
`removeChannelsFromPolicy`	Removing channels from data retention policies
`removeTeamsFromPolicy`	Removing teams from data retention policies

Jobs Events

Event Name	Description
`cancelJob`	Canceling background jobs
`createJob`	Creating new background jobs
`jobServer`	Job server operations
`updateJobStatus`	Updating job status/progress

Licensing Events

Event Name	Description
`addLicense`	Adding enterprise licenses
`localAddLicense`	Local license addition (cluster mode)
`localRemoveLicense`	Local license removal (cluster mode)
`removeLicense`	Removing enterprise licenses
`requestTrialLicense`	Requesting trial licenses

Bot Management Events

Event Name	Description
`assignBot`	Assigning bots to users
`convertBotToUser`	Converting bot accounts to user accounts
`convertUserToBot`	Converting user accounts to bot accounts
`createBot`	Creating new bot accounts
`patchBot`	Updating bot account properties
`updateBotActive`	Updating bot account active/inactive status

Custom Emojis Events

Event Name	Description
`createEmoji`	Creating custom emojis
`deleteEmoji`	Deleting custom emojis

Branding Events

Event Name	Description
`deleteBrandImage`	Deleting brand images
`uploadBrandImage`	Uploading brand images

Search Events

Event Name	Description
`purgeBleveIndexes`	Purging Bleve search indexes
`purgeElasticsearchIndexes`	Purging Elasticsearch search indexes

Roles and Schemes Events

Event Name	Description
`createScheme`	Creating permission schemes
`deleteScheme`	Deleting permission schemes
`patchRole`	Updating role permissions
`patchScheme`	Updating permission schemes

Preferences Events

Event Name	Description
`deletePreferences`	Deleting user preferences
`updatePreferences`	Updating user preferences

Channel Categories Events

Event Name | Description	
`createCategoryForTeamForUser` | Creating channel categories for users	
`deleteCategoryForTeamForUser`	Deleting channel categories for users
`updateCategoriesForTeamForUser`	Updating multiple channel categories for users
`updateCategoryForTeamForUser`	Updating single channel category for users
`updateCategoryOrderForTeamForUser`	Updating channel category order for users

Export and Import Events

Event Name	Description
`bulkExport`	Bulk data export operations
`bulkImport`	Bulk data import operations
`deleteExport`	Deleting export files
`deleteImport`	Deleting import files
`generatePresignURLExport`	Generating presigned URLs for exports
`scheduleExport`	Scheduling export operations
`slackImport`	Slack data import operations

Access Control Events

Event Name	Description
`applyIPFilters`	Applying IP filtering rules
`assignAccessPolicy`	Assigning access policies to users/teams
`createAccessControlPolicy`	Creating new access control policies
`deleteAccessControlPolicy`	Deleting access control policies
`unassignAccessPolicy`	Unassigning access policies from users/teams
`updateActiveStatus`	Updating active status of access control policies

Custom Profile Attributes Events

Event Name	Description
`createCPAField`	Creating custom profile attribute fields
`deleteCPAField`	Deleting custom profile attribute fields
`patchCPAField`	Updating custom profile attribute fields
`patchCPAValues`	Updating custom profile attribute values

Outgoing OAuth Connections Events

Event Name	Description
`createOutgoingOauthConnection`	Creating outgoing OAuth connections
`deleteOutgoingOAuthConnection`	Deleting outgoing OAuth connections
`updateOutgoingOAuthConnection`	Updating outgoing OAuth connections
`validateOutgoingOAuthConnectionCredentials`	Validating outgoing OAuth connection credentials

Terms of Service Events

Event Name	Description
`createTermsOfService`	Creating terms of service
`saveUserTermsOfService`	Saving user acceptance of terms of service

Compliance and Audit Events

Event Name	Description
`addAuditLogCertificate`	Adding audit log certificates
`createComplianceReport`	Creating compliance reports
`downloadComplianceReport`	Downloading compliance reports
`getComplianceReport`	Getting compliance reports
`getComplianceReports`	Getting multiple compliance reports
`removeAuditLogCertificate`	Removing audit log certificates

Local Operations Events

Event Name	Description
`localCheckIntegrity`	Local integrity checks
`localCreateChannel`	Local channel creation
`localCreateCommand`	Local command creation
`localCreateIncomingHook`	Local incoming webhook creation
`localCreateTeam`	Local team creation
`localDeleteChannel`	Local channel deletion
`localDeletePost`	Local post deletion
`localDeleteTeam`	Local team deletion
`localDeleteUser`	Local user deletion
`localGetClientConfig`	Local client configuration retrieval
`localGetConfig`	Local configuration retrieval
`localInviteUsersToTeam`	Local user invitation to teams
`localMoveChannel`	Local channel moving

Event Name	Description
`localPatchChannel`	Local channel patching
`localPatchConfig`	Local configuration patching
`localPermanentDeleteAllUsers`	Local permanent deletion of all users
`localRemoveChannelMember`	Local channel member removal
`localRestoreChannel`	Local channel restoration
`localUpdateChannelPrivacy`	Local channel privacy update
`localUpdateConfig`	Local configuration update

> Note
>
> This comprehensive list includes all audit events captured by Mattermost across all major system operations. Additional events may be logged depending on your Mattermost version, enabled features, and configuration settings. Enterprise and Enterprise Advanced features may generate additional audit events.

JSON data model

Name	Type	Description
timestamp	int64	Date/time when event or activity has taken place. Mattermost currently supports three log formats: plain, JSON, and GELF (https://go2docs.graylog.org/current/getting_in_log_data/gelf.html). • Plain log format uses RFC3339 (https://www.rfc-editor.org/rfc/rfc3339) by default. See the plain log format configuration documentation for supported options. • JSON log format uses RFC3339 (https://www.rfc-editor.org/rfc/rfc3339) by default. See the JSON log format configuration documentation for supported options. • GELF log format uses unixtime (https://www.unixtimestamp.com/). See the GELF log format configuration documentation for supported options.
event_name	string	Unique name and identifier of the event type taking place. See the audit event types section for a comprehensive list of all supported event names.
status	string	Success or failure of the audited event.
event	EventData	Event parameters and object states.
actor	EventActor	User involved with the event.
meta	EventMeta	Related event metadata.
error	EventError	The resulting error if the status is in a failed state.

EventData

Field name	Data type	Description
parameters	map	Payload and parameters being processed as part of the request.
prior_state	map	Prior state of the entity being modified. `null` if there was no prior state.
resulting_state	map	Resulting entity after creating or modifying it.
object_type	string	String representation of the entity type (e.g. post)

EventActor

Field name	Data type	Description
user_id	string	Unique identifier of the event actor.
session_id	string	Unique session identifier of the event actor.
client	string	User agent of the client/platform in use by the event actor.
ip_address	string	IPv4/IPv6 IP address of the event actor.

EventMeta

Field name	Data type	Description
api_path	string	The REST endpoint which caused the event.
cluster_id	integer	Cluster identifier.

EventError

Field name	Data type	Description
description	string	(Optional) Error description.
status_code	integer	(Optional) Error status code.

On this page

- Audit Log JSON Schema
  - Audit log record examples
    - Update user preferences
    - Create post
    - System configuration change
  - Audit event types
    - User Management Events
    - Channel Management Events
    - Team Management Events
    - Posts & Content Events
    - Authentication and Security Events
    - System Administration Events
    - File Management Events
    - OAuth Applications Events

    - Webhooks Events
    - Slash Commands Events
    - Plugins Events
    - Groups & LDAP Events
    - Remote Clusters Events
    - Data Retention Events
    - Jobs Events
    - Licensing Events
    - Bot Management Events
    - Custom Emojis Events
    - Branding Events
    - Search Events
    - Roles and Schemes Events
    - Preferences Events
    - Channel Categories Events
    - Export and Import Events
    - Access Control Events
    - Custom Profile Attributes Events
    - Outgoing OAuth Connections Events
    - Terms of Service Events
    - Compliance and Audit Events
    - Local Operations Events
  - JSON data model
    - EventData
    - EventActor
    - EventMeta
    - EventError

Export channel data

Ensure important data isn't trapped in silos by migrating data between systems or backing data up for operational continuity. Once exported, channel data can be analyzed using various tools for insights into team dynamics, productivity, and communication patterns, and to fulfill reporting and auditability requirements.

Enable

> Note
>
> For Mattermost Cloud deployments, no setup is required. See the usage section below for details on exporting channel data.

For self-hosted deployments, a Mattermost system admin must install the Channel Export integration from the in-product App Marketplace:

1. In Mattermost, from the Product menu

   , select **App Marketplace**.

2. Search for or scroll to **Channel Export**, and select **Install**.

3. Once installed, select **Configure**. You're taken to the System Console.

4. On the **Channel Export** configuration page, enable and configure the plugin, and then select **Save**. You can restrict the ability to export channels to system and channel admins only, and you can configure a maximum file size for channel export files.

Upgrade

We recommend upgrading this feature as new versions are released. Generally, updates are seamless and don't interrupt the user experience in Mattermost. Visit the Releases page (https://github.com/mattermost/mattermost-plugin-channel-export/releases) for the latest release, available releases, and compatibiilty considerations.

Usage

You need to be a Mattermost system admin to export channel data.

Use the `/export` slash command in a channel to export the current channel's message data into a CSV-formatted file.

Get help

Mattermost customers can open a Mattermost support case (https://support.mattermost.com/hc/en-us/requests/new). To report a bug, please open a GitHub issue against the Mattermost Channel Export plugin GitHub repository (https://github.com/mattermost/mattermost-plugin-channel-export).

For questions, feedback, and assistance, join our pubic Integrations and Apps channel (https://community.mattermost.com/core/channels/integrations) on the Mattermost Community Server (https://community.mattermost.com/) for assistance.

Customize this integration

This plugin contains both a server and web app portion. See the Channel Export plugin README documentation on GitHub for details. Visit the Mattermost Developer Workflow (https://developers.mattermost.com/extend/plugins/developer-workflow/) and Mattermost Developer environment setup (https://developers.mattermost.com/extend/plugins/developer-setup/) for information about developing, customizing, and extending Mattermost functionality.

On this page

- Export channel data
  - Enable
  - Upgrade
  - Usage
  - Get help
  - Customize this integration

Legal Hold

A Legal Hold, also known as a litigation hold, is a process that an organization uses to preserve all forms of relevant information when litigation is reasonably anticipated. It's a requirement established by the Federal Rules of Civil Procedure (FRCP) in the United States and similar laws in other jurisdictions.

Primary use cases include:

1. **Litigation**: In anticipation or in the event of a lawsuit, organizations need to preserve all relevant documents and electronic data to ensure they can adequately defend their position. A failure to do so could result in court penalties.

2. **Regulatory investigation**: If an organization is being investigated by a regulatory body, it may be required to preserve and produce certain documents or data.

3. **Audits**: During an audit, whether internal or external, an organization might need to put a hold on certain data that is relevant to the audit.

4. **Records management**: In some cases, organizations might use a Legal Hold to temporarily suspend the deletion of data that would otherwise be purged as part of its records management policy.

Mattermost is used as a secure collaboration hub by technical and operational teams, with critical documents and data shared on a daily basis. Thus, Legal Hold is a key requirement for Enterprises and public sector organizations who have deployed Mattermost for their teams, to meet compliance & auditory requirements while minimizing risk.

Mattermost Legal Hold can be combined with eDiscovery integration and data retention policies to customize the data retained and deleted to comply with compliance requirements.

Legal Hold demo (Sneak Peek)

Check out this YouTube sneak peek demo (https://www.youtube.com/watch?v=86c8NoOxlQw&feature=youtu.be) to learn about Mattermost's Legal Hold workflow.

Below are step-by-step instructions on how to carry out a Legal Hold for Mattermost using the Mattermost Legal Hold plugin.

How to carry out a Legal Hold

Step 1: Upgrade to Mattermost Enterprise

Legal Hold is available in Mattermost Enterprise. Learn more about the Enterprise plan & request a quote online at https://mattermost.com/pricing/

Step 2: Establish a Legal Hold policy

Establish a policy for when to implement a Legal Hold. This should be developed in consultation with your legal counsel and should include procedures for identifying relevant users (those who have potentially relevant information).

Establishing a Legal Hold policy first enables you to configure the Mattermost system correctly to meet your compliance & auditory requirements, minimizing associated risk.

Step 3: Set up the Mattermost Legal Hold plugin

Install the plugin

1. Log in to your Mattermost workspace as a system administrator.
2. Download the latest version of the plugin binary release (https://github.com/mattermost/mattermost-plugin-legal-hold/releases/), compatible with Mattermost v8.0.1 and later. If you are using an earlier version of Mattermost, follow our documentation to upgrade to Mattermost v8.0.1 or later.

3. Go to **System Console > Plugins > Plugin Management > Upload Plugin**, and upload the plugin binary you downloaded in the previous step.

4. In the **Installed Plugins** section, scroll to **Legal Hold Plugin**, and select **Enable**.

Configure the plugin

When the Legal Hold integration is enabled, you can configure when it runs using the format `HH:MM ±HHMM` and `+0000` for UTC.

You can configure a custom Amazon S3 bucket for Legal Holds by specifying Amazon S3 configuration settings. If no S3 configuration is specified, the Mattermost server file store used. Learn more about file storage configuration options in our product documentation.

(Optional) Configure a data retention policy

You can optionally configure a data retention policy to control how long data and file attachments are retained in the Mattermost database.

Step 4: Create a Legal Hold

In Mattermost, create a Legal Hold by completing the following steps:

1. Go to **System Console > Plugins > Legal Hold Plugin**, and select **Create new**.

2. Enter a name for the Legal Hold.

3. Specify the user names or user groups of users you want to place on Legal Hold.

4. (Optional) Public channels are excluded by default. You can choose to include public channels that the specified users or user groups are members of, if preferred; however, doing so will significantly increase the amount of data held based on the number public channels available.

5. Specify the number of days that users are placed in Legal Hold with a start date. An end date is optional.

6. Select **Create Legal Hold**. Downloadable data won't be available until the next scheduled job runs.

Manage Legal Holds

While a Legal Hold is in place, you can edit details of the Legal Hold, access the Legal Hold Secret, as well as download a copy of the preserved data to your local machine.

Edit a Legal Hold

Select the **Edit**

icon to change the name of the Legal Hold, add or remove users, change the end date, as well as include or exclude public channels.

Access a Legal Hold secret

A Legal Hold secret enables you to verify the authenticity of the data for a Legal Hold in Mattermost.

Select the **Show**

icon to display the Legal Hold secret key. Keep a copy of this key in a secure location.

To verfiy the contents of the files in this Legal Hold, you must append the processor command with the following flag: `--legal-hold-secret <KEY>`. The output verifies the file and returns the authenticity state of files along with the rest of the output for the processor, as follows:

Success:

```
Secret key was provided, verifying legal holds... - Verifying Legal Hold
*processor9*: Verified
```

Error:

```
... Secret key was provided, verifying legal holds... - Verifying Legal Hold
*processor9*: [Error] hash mismatch for file: legal_hold/
processor9_i7k1dbkipiyojeess6ozi4agyr/index.json ...
```

Download Legal Hold data

Select the **Download**

icon to download a copy of the preserved data to a location on your local machine. Note, no data will be available to download until at least one scheduled job is completed. This may take up to 24 hours.

Step 5: Release a Legal Hold

Once the Legal Hold has completed, release it to take users off of the Legal Hold by selecting the **Release** option to the right of the Legal Hold task.

Important

Once a Legal Hold is released, all data is irretrievably deleted from Mattermost and can't be recovered.

Frequently asked questions

Who can implement Legal Hold?

Only Mattermost system admins can implement a Legal Hold.

Does a user know if they're placed under a Legal Hold in Mattermost?

No, users won't be notified if they're placed under a Legal Hold, and no reference to Legal Holds will be visible in their Mattermost client or accessible via the Mattermost API. This allows for investigations to be conducted without influencing user behavior and without conflicts of interest.

What types of content does Legal Hold cover?

The Legal Hold covers all messages and file uploads shared in conversations where the Legal Hold is active, including messages posted by plugins, bots or webhooks. This includes messages or files shared in public channels, private channels, direct messages and group messages.

However, Legal Hold does not apply to reactions, collaborative playbooks, or audio calls.

Can users delete their messages while on a Legal Hold?

Yes, users can delete messages, but they are retained for the purposes of Legal Hold when implemented with the aforementioned steps.

Can a Legal Hold be applied retroactively to collect past data?

Yes, but this is only guaranteed for existing and future messages/files once Legal Hold is activated. It won't recover messages or files that were deleted before the Legal Hold was activated.

Is Legal Hold the same as e-discovery?

No. While they serve a related use case, they are not the same.

Legal Hold is an initial step to ensure relevant electronically stored information (ESI) is preserved. On the other hand, e-discovery is a multi-step process that uses this preserved data to identify, collect, preserve, process, review, and produce ESI in the context of a legal or investigative process.

How do I enable e-discovery for Mattermost?

Learn more about extracting data for e-discovery in our product documentation.

How do I manage storage costs and version retention in S3?

If you plan to use an existing S3 bucket for Legal Hold data storage, and your existing S3 bucket has versioning enabled, we strongly recommend using a dedicated S3 bucket with versioning disabled.

The Legal Hold plugin frequently modifies files in the `legalhold` directory, and when S3 bucket versioning is enabled, each modification creates a new version. This can result in a rapid accumulation of object versions, increased storage costs, potential performance impact, higher S3 API usage, and complicating version management over time. See the [S3 Lifecycle Rules (https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html)](https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html) documentation for additional details.

On this page

- Legal Hold
  - Legal Hold demo (Sneak Peek)
  - How to carry out a Legal Hold
    - Step 1: Upgrade to Mattermost Enterprise
    - Step 2: Establish a Legal Hold policy
    - Step 3: Set up the Mattermost Legal Hold plugin
      - Install the plugin
      - Configure the plugin
      - (Optional) Configure a data retention policy
    - Step 4: Create a Legal Hold
      - Manage Legal Holds
        - Edit a Legal Hold
        - Access a Legal Hold secret
        - Download Legal Hold data
    - Step 5: Release a Legal Hold
  - Frequently asked questions
    - Who can implement Legal Hold?
    - Does a user know if they're placed under a Legal Hold in Mattermost?
    - What types of content does Legal Hold cover?
    - Can users delete their messages while on a Legal Hold?
    - Can a Legal Hold be applied retroactively to collect past data?
    - Is Legal Hold the same as e-discovery?
    - How do I enable e-discovery for Mattermost?
    - How do I manage storage costs and version retention in S3?

Mattermost Agents Admin Guide

> Note
>
> Mattermost Agents is formally known as Mattermost Copilot.

This guide covers installing, configuring, and managing the Mattermost Agents plugin. You'll learn how to set up AI capabilities for your Mattermost instance and configure them for your organization's needs.

Installation

Prerequisites

Before installing the Agents plugin, ensure your environment meets these requirements:

- Mattermost Server v10.0+
- PostgreSQL database
- For semantic search: PostgreSQL with pgvector extension
- Network access to your chosen LLM provider
- API keys if using a cloud LLM service

Installation Steps

Use pre-installed plugin

From Mattermost v10.3, Agents comes installed automatically and ready for you to configure a large language model (LLM). When no LLMs are configured, the Agents panel prompts users to configure one.

Install latest version

For the most recent features and improvements, you can download and install the latest plugin version from the [GitHub releases page (https://github.com/mattermost/mattermost-plugin-ai/releases)](https://github.com/mattermost/mattermost-plugin-ai/releases).

Install the plugin through the System Console by navigating to **System Console > Plugin Management**, clicking **Upload Plugin**, selecting the downloaded plugin file (.tar.gz), and clicking **Upload**. Enable the plugin after upload completes, then configure plugin settings as detailed in the Configuration section below.

Configuration

Access plugin settings

Navigate to **System Console > Plugins > Agents** to access the configuration interface.

Enable the plugin

Agents is enabled automatically when using the pre-installed version. If you've manually installed a newer version, you may need to enable it by going to **System Console > Plugins > Agents** and setting **Enable Plugin** to **True**, then complete configuration in the System Console.

Basic configuration

If you have an Enterprise license, upload it to unlock additional features.

For general settings, you can toggle to enable or disable the plugin system-wide, enable debug logging for troubleshooting (use only when needed), and configure the hostname allowlist for API calls.

Agent configuration

Configure an LLM for your Agents integration by going to **System Console > Plugins > Agents** and selecting **Add an Agent**. The plugin supports creating multiple Agents with different configurations. The ability to define multiple LLMs for your Agents integration requires a Mattermost Enterprise license.

Select **Add an Agent** to create a new Agent, then configure the agent settings:

Setting	Description
Display Name	User-facing name shown in Mattermost
Agent Username	The mattermost username for the agent. @ mentions to the agent will use this name
Agent Avatar	Custom image for the agent
Service	LLM provider for this agent (OpenAI, Anthropic, Azure OpenAI, OpenAI-compatible)
Send User ID	Whether to send Mattermost user IDs to the LLM provider
Default Model	Specific model to use from your chosen provider
Input Token Limit	Maximum tokens allowed in input (model-dependent)
Output Token Limit	Maximum tokens allowed in output (model-dependent)
Streaming Timeout Seconds	Timeout in seconds for streaming responses
Custom Instructions	Custom instructions that define the agent's personality and capabilities

Setting	Description
Enable Vision	Enable Vision to allow the agent to process images. Requires a compatible model.
Enable Tools	By default some tool use is enabled to allow for features such as integrations with JIRA. Disabling this allows use of models that do not support or are not very good at tool use. Some features will not work without tools.
Access Control	Set which teams, channels, and users can access this agent

Select **Save** to create the agent.

Provider configuration

For each LLM provider you want to use, you'll need to configure authentication. The basic requirements are:

Provider	Required	Optional
OpenAI	API Key	Organization ID
Anthropic	API Key	
Azure OpenAI	API Key, Resource Name, Deployment ID	

See the [Provider Guide (https://docs.mattermost.com/agents/docs/providers.html)](https://docs.mattermost.com/agents/docs/providers.html) for detailed provider-specific configuration.

Custom instructions

Text input in the custom instructions field is included in the prompt for every request. Use this to give your agents extra context or instructions.

For example, you could list your organization's specific acronyms so the agent knows your vernacular and users can ask for definitions. Or you could give it specialized instructions like adopting a specific personality or following a certain workflow. By customizing the instructions for each individual agent, you can create a more tailored AI experience for your specific needs.

Embed search configuration

To enable semantic search capabilities, you'll need to enable the `pgvector` extension in your PostgreSQL database, then configure embeddings provider settings including the provider (OpenAI, etc.), model for embeddings, and dimensions that match your chosen embedding model. Embedding search requires an Enterprise license and is available as an [experimental (https://docs.mattermost.com/manage/feature-labels.html#experimental)](https://docs.mattermost.com/manage/feature-labels.html#experimental) feature. Performance may vary with large datasets.

Configure chunking options based on your needs:

Setting	Recommended Value	Description
Chunking Strategy	Sentences, Paragraphs, or Fixed Size	Choose based on your content type
Chunk Size	512-1024 tokens	Varies by strategy
Chunk Overlap	20-50 tokens	For better context continuity
Minimum Size Ratio	Default	Minimum ratio for chunk size validation

Run the initial indexing process after configuration.

Permission configuration

Configure who can access AI features by setting team-level, channel-level, and user-level permissions for each agent.

Management tasks

Plugin metrics

Metrics for Agents are exposed through the `/plugins/mattermost-ai/metrics` subpath under the existing Mattermost server metrics endpoint. This is controlled by the Listen address for performance configuration setting. It defaults to port `8067`, and the following metrics are available:

- `agents_system_plugin_start_timestamp_seconds`: The time the plugin started.
- `agents_system_plugin_info`: The plugin version and installation ID.
- `agents_api_time_seconds`: How long to execute API.
- `agents_http_requests_total`: The total number of API requests.
- `agents_http_errors_total`: The total number of http API errors.
- `agents_llm_requests_total`: The total number of requests to upstream LLMs.

Post indexing

Post indexing occurs automatically during initial setup and when changing embedding providers:

1. Navigate to **System Console > Plugins > Agents > Embedding Search**
2. Use the reindex controls to:
    - Monitor indexing progress during initial setup.
    - Trigger reindexing when changing embedding providers.
    - Check indexing status.

Backup and restore

The plugin configuration is stored in the Mattermost database. To backup:

1. Ensure your regular Mattermost backup includes plugin configurations
2. For larger deployments, consider backing up indexed vector data separately

Troubleshooting

Logging

Enhanced logging can help diagnose issues:

1. Check server logs for entries with the structured logging field `plugin_ai` set to `mattermost-ai`.
2. Enable **LLM Trace** in the plugin configuration to see detailed request/response information for all LLM interactions.
3. Enable debug logging in the plugin configuration for additional diagnostic information.
4. For production environments, disable debug logging and LLM Trace after troubleshooting to reduce log volume.

Integrations

Currently integrations are limited to direct messages between users and the agents. The integrations won't operate from within public, private, or group message channels.

Built-in tool integrations

Server Search

- **Function**: Semantic search across Mattermost content.
- **Requirements**: Embedding search must be configured and enabled.

- **Security**: Respects user permissions - users only see content they have access to.

User Lookup

- **Function**: Look up Mattermost user information by username
- **Data Available**: Username, full name, email, nickname, position, locale, timezone, last activity, status
- **Permissions**: Requires `VIEW_MEMBERS` permission

Jira Integration

- **Function**: Fetch issues from public Jira instances
- **Requirements**: No additional configuration needed
- **Usage**: Provide Jira instance URL and issue keys
- **Data Retrieved**: Issue summary, description, status, assignee, comments, metadata

GitHub Integration

- **Function**: Fetch GitHub issues and pull requests
- **Requirements**: Mattermost GitHub plugin must be installed and running
- **Authentication**: Users must be logged into GitHub through the Mattermost GitHub plugin
- **Access**: Works with both public and private repositories (based on user permissions)
- **Data Retrieved**: Issue/PR title, number, state, submitter, body content

Security Note: All tool integrations are restricted to direct messages to maintain security boundaries and require explicit user approval before execution.

Model Context Protocol (MCP) Integration

The Model Context Protocol (MCP) integration allows Agents to connect to external tools and services through standardized MCP servers. This experimental (https://docs.mattermost.com/manage/feature-labels.html#experimental) feature enables expanding AI capabilities with custom integrations.

Configuration

1. Navigate to **System Console > Plugins > Agents > MCP Servers**.
2. Enable MCP integration by setting **Enable MCP** to **True**.
3. Configure connection settings:
    - **Idle Timeout**: Set timeout in minutes for inactive client connections (default: 30 minutes)

Add MCP servers

1. Select **Add MCP Server** to configure a new server.
2. Configure server settings:
    - **Server URL**: The endpoint URL for your MCP server.
    - **Custom Headers**: Additional headers required by your MCP server (optional).
    - **Server Name**: Descriptive name for the server (auto-generated if not provided).
3. Select **Save** to add the server.

Management

- **Connection Management**: The system automatically manages user connections to MCP servers
- **Idle Cleanup**: Inactive client connections are automatically closed after the configured timeout

- **Per-User Connections**: Each user gets their own connection to MCP servers for security and isolation

Enterprise features

The following features require an Enterprise license:

- Multiple agent configurations
- Fine-grained access controls
- Embedding search (Experimental)
- MCP Support (Experimental)
- Usage analytics

> Tip
>
> - Looking for demo? [Try it yourself (https://mattermost.com/demo/copilot/)](https://mattermost.com/demo/copilot/)!
> - Download the [Mattermost datasheet (https://mattermost.com/mattermost-copilot-datasheet/)](https://mattermost.com/mattermost-copilot-datasheet/) to learn more about integrating with industry-leading large language models (LLMs).

On this page

- Mattermost Agents Admin Guide
  - Installation
    - Prerequisites
    - Installation Steps
      - Use pre-installed plugin
      - Install latest version
  - Configuration
    - Access plugin settings
    - Enable the plugin
    - Basic configuration

        - Agent configuration
        - Provider configuration
        - Custom instructions
        - Embed search configuration
        - Permission configuration
    - Management tasks
        - Plugin metrics
        - Post indexing
        - Backup and restore
    - Troubleshooting
        - Logging
    - Integrations
        - Built-in tool integrations
            - Server Search
            - User Lookup
            - Jira Integration
            - GitHub Integration
    - Model Context Protocol (MCP) Integration
        - Configuration
        - Add MCP servers
        - Management
    - Enterprise features

Authentication configuration settings

Mattermost supports up to 4 distinct, concurrent methods of user authentication:

- An OpenID provider
- A SAML provider
- An LDAP instance (e.g., Active Directory, OpenLDAP)
- Email and Password

Review and manage the following authentication configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Authentication**:

- Signup
- Email
- Password
- MFA
- AD/LDAP
- SAML 2.0
- OAuth 2.0
- OpenID Connect
- Guest Access

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `EnableUserCreation` value is under `TeamSettings`.
>
> - If using a tool such as [jq (https://stedolan.github.io/jq/)](https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.TeamSettings.EnableUserCreation'`
> - When working with the `config.json` file manually, look for an object such as `TeamSettings`, then within that object, find the key `EnableUserCreation`.

Signup

Access the following configuration settings in the System Console by going to **Authentication > Signup**.

Enable account creation

• **true**: **(Default)** Anyone can sign up for a user account on this server without needing to be invited. Applies to email-based signups only. • **false**: The ability to create accounts is disabled. Selecting **Create Account** displays an error. Applies to email, OpenID Connect, and OAuth 2.0 user account authentication.	• System Config path: **Authentication > Signup** • `config.json` setting: `TeamSettings` > `EnableUserCreation` > `true` • Environment variable: `MM_TEAMSETTINGS_ENABLEUSERCREATION`

Note

- LDAP and SAML users can always create a Mattermost account by logging in using LDAP or SAML user credentials, regardless of whether this configuration setting is enabled.
- From Mattermost v10.9, email addresses enclosed in angle brackets (e.g., `<billy@example.com>`) will be rejected. To avoid issues, ensure all user emails comply with the plain address format (e.g., `billy@example.com`). In addition, we strongly recommend taking proactive steps to audit and update Mattermost user data to align with this product change, as impacted users may face issues accessing Mattermost or managing their user profile. You can update these user emails manually using mmctl user email.
- See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Restrict account creation to specified email domains

This setting limits the email address domains that can be used to create a new account or team. | - System Config path: **Authentication > Signup** You **must** set Require Email Verification | - `config.json` setting: `TeamSettings` > `RestrictCreationToDomains` to `true` for the restriction to function. This setting only affects email login. | - Environment variable: `MM_TEAMSETTINGS_RESTRICTCREATIONTODOMAINS`

Enable open server

• **true**: Users can create accounts on the server without an invitation. • **false**: **(Default)** Users **must** have an invitation to create an account on the server.	• System Config path: **Authentication > Signup** • `config.json` setting: `TeamSettings` > `EnableOpenServer` • Environment variable: `MM_TEAMSETTINGS_ENABLEOPENSERVER`

Enable email invitations

• **true**: **(Default for Cloud deployments)** Allows users to send email invitations. • **false**: **(Default for self-hosted deployments)** Disables email invitations.	• System Config path: **Authentication > Signup** • `config.json` setting: `ServiceSettings` > `EnableEmailInvitations` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLEEMAILINVITATIONS`

> Note
>
> Cloud admins can't modify this configuration setting.

Invalidate pending email invites

This button invalidates email invitations that have not been accepted (by default, invitations expire after 48 hours). This option has no `config.json` setting or environment variable.	• System Config path: **Authentication > Signup** • `config.json` setting: N/A • Environment variable: N/A

Email

Access the following configuration settings in the System Console by going to **Authentication > Email**.

Enable account creation with email

• **true**: **(Default)** Allows creation of team and user accounts with email and password. • **false**: Disables creation of team and user accounts with email and password. Requires a single sign-on (SSO) service to create accounts.	• System Config path: **Authentication > Email** • `config.json` setting: `EmailSettings` > `EnableSignUpWithEmail` • Environment variable: `MM_EMAILSETTINGS_ENABLESIGNUPWITHEMAIL`

> Note
>
> Cloud admins can’t modify this configuration setting.

Require email verification

• **true**: **(Default for Cloud deployments)** Requires email verification for new accounts before allowing the user to sign-in. • **false**: **(Default for self-hosted deployments)** Disables email verification. can be used to speed development by skipping the verification process.	• System Config path: **Authentication > Email** • `config.json` setting: `EmailSettings` > `RequireEmailVerification` > `false` • Environment variable: `MM_EMAILSETTINGS_REQUIREEMAILVERIFICATION`

Enable sign-in with email

• **true**: **(Default)** Allows users to sign-in with email and password. • **false**: Disables authentication with email and password, and removes the option from the login screen. Use this option to limit authentication to single sign-on services.	• System Config path: **Authentication > Email** • `config.json` setting: `EmailSettings` > `EnableSignInWithEmail` • Environment variable: `MM_EMAILSETTINGS_ENABLESIGNINWITHEMAIL`

> Note
>
> - To provide users with only a single email sign in option on the login page, ensure that the enable sign-in with username configuration setting is set to **false**.
> - From Mattermost v10.9, email addresses enclosed in angle brackets (e.g., `<billy@example.com>`) will be rejected. To avoid issues, ensure all user emails comply with the plain address format (e.g., `billy@example.com`). In addition, we strongly recommend taking proactive steps to audit and update Mattermost user data to align with this product change, as impacted users may face issues accessing Mattermost or managing their user profile. You can update these user emails manually using mmctl user email.

Enable sign-in with username

• **true**: **(Default)** Allows authentication with a username and password for accounts created with an email address. This setting does not affect AD/LDAP sign-in. • **false**: Disables authenticaton with a username and removes the sign in option from. from the login screen.	• System Config path: **Authentication > Email** • `config.json` setting: `EmailSettings` > `EnableSignInWithUsername` • Environment variable: `MM_EMAILSETTINGS_ENABLESIGNINWITHUSERNAME`

> Note
>
> We highly recommended that email-based authentication is only used in small teams on private networks.

Password

Access the following configuration settings in the System Console by going to **Authentication > Password**.

Minimum password length

This setting determines the minimum number of characters in passwords. It must be a whole number greater than or equal to 5 and less than or equal to 72. Numerical input. Default is **5**.	• System Config path: **Authentication > Password** • `config.json` setting: `PasswordSettings` > `MinimumLength` • Environment variable: `MM_PASSWORDSETTINGS_MINIMUMLENGTH`

Password requirements

This setting controls password character requirements. By checking the corresponding box, passwords must contain: • **At least one lowercase letter** • **At least one uppercase letter** • **At least one number** • **At least one symbol** out of these: `!"#$%&'()*+,-./:;<=>?@[]^_`|~`.  The error message previewed in the System Console will appear if the user attempts to set an invalid password.  The default for all boxes is unchecked. The default for all settings in `config.json` is `false`.	• System Config path: **Authentication > Password** • `config.json` settings: `PasswordSettings` > `Lowercase` > `false`, `PasswordSettings` > `Uppercase` > `false`, `PasswordSettings` > `Number` > `false`, `PasswordSettings` > `Symbol` > `false` • Environment variables: `MM_PASSWORDSETTINGS_LOWERCASE`, `MM_PASSWORDSETTINGS_UPPERCASE`, `MM_PASSWORDSETTINGS_NUMBER`, `MM_PASSWORDSETTINGS_SYMBOL`

Maximum login attempts

This setting determines the number of failed sign-in attempts a user can make before being locked out and required to go through a password reset by email. Numerical input. Default is **10**.	• System Config path: **Authentication > Password** • `config.json` setting: `ServiceSettings` > `MaximumLoginAttempts` > `10` • Environment variable: `MM_SERVICESETTINGS_MAXIMUMLOGINATTEMPTS`

Enable forgot password link

• **true**: **(Default)** Displays the **Forget Password** link on the Mattermost login page. • **false**: Hides the **Forgot Password** link from the Mattermost login page.	• System Config path: **Authentication > Enable forgot password link** • `config.json` setting: `LdapSettings` > `ForgotPasswordLink` > `true` • Environment variable: `MM_LDAPSETTINGS_FORGOTPASSWORDLINK`

> Note
>
> You can customize the **Forgot Password** link URL by going to **Site Configuration > Customization > Forgot Password Custom Link**. See the configuration documentation for details.

MFA

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > MFA**.

We recommend deploying Mattermost within your own private network, and using VPN clients for mobile access, so that Mattermost is secured with your existing protocols. If you choose to run Mattermost outside your private network, bypassing your existing security protocols, we recommend adding a multi-factor authentication service specifically for accessing Mattermost.

Enable multi-factor authentication

• **true**: Users who sign-in with AD/LDAP or an email address have the option to add multi-factor authentication to their accounts. • **false**: **(Default)** Disables multi-factor authentication.	• System Config path: **Authentication > MFA** • `config.json` setting: `ServiceSettings` > `EnableMultifactorAuthentication` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLEMULTIFACTORAUTHENTICATION`

Enforce multi-factor authentication

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: Requires [multi-factor authentication (MFA) (https://docs.mattermost.com/administration-guide/onboard/multi-factor-authentication.html)](https://docs.mattermost.com/administration-guide/onboard/multi-factor-authentication.html) for users who sign-in with AD/LDAP or an email address. New users must set up MFA. Logged in users are redirected to the MFA setup page until configuration is complete. • **false**: **(Default)** MFA is optional.	• System Config path: **Authentication > MFA** • `config.json` setting: `ServiceSettings` > `EnforceMultifactorAuthentication` > `false` • Environment variable: `MM_SERVICESETTINGS_ENFORCEMULTIFACTORAUTHENTICATION`

> Note
>
> If your system has users who authenticate with methods other than AD/LDAP and email, MFA must be enforced with the authentication provider outside of Mattermost.

AD/LDAP

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > AD/LDAP**. This opens the AD/LDAP setup wizard with step-by-step sections and testing to help configure each setting.

The wizard is organized into the following sections:

- Connection settings: Configure server connection details

- User filters: Set up user identification and filtering
- Account synchronization: Map AD/LDAP attributes to Mattermost user fields
- Group synchronization: Configure group settings and group attributes (if using LDAP groups)
- Synchronization performance: Adjust synchronization timing and performance settings
- Synchronization history: View synchronization status and manually trigger syncs

> Note
>
> Each section includes a **Test** option you can use to verify your configuration incrementally, helping identify and resolve issues early in the setup process.

Connection settings

Configure your AD/LDAP server connection and basic authentication settings. Use the **Test Connection** button in this section to verify your server connection before proceeding to other configuration steps.

Enable sign-in with AD/LDAP

• **true**: Allows sign-in with AD/LDAP. • **false**: **(Default)** Disables sign-in with AD/LDAP.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `Enable` > `false` • Environment variable: `MM_LDAPSETTINGS_ENABLE`

Enable synchronization with AD/LDAP

• **true**: Mattermost periodically syncs users from AD/LDAP. • **false**: **(Default)** Disables AD/LDAP synchronization.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `EnableSync` > `false` • Environment variable: `MM_LDAPSETTINGS_ENABLESYNC`

> Note
>
> Synchronization with AD/LDAP settings in the System Console can be used to determine the connectivity and availability of arbitrary hosts. System admins concerned about this can use custom admin roles to limit access to modifying these settings. See the delegated granular administration documentation for details.

Login field name

This setting will display placeholder text in the login field of the sign-in page. This text can remind users to sign-in with their AD/LDAP credentials. String input. Default is `AD/LDAP Username`.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `LoginFieldName` • Environment variable: `MM_LDAPSETTINGS_LOGINFIELDNAME`

AD/LDAP server

This is the domain name or IP address of the AD/LDAP server. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `LdapServer` • Environment variable: `MM_LDAPSETTINGS_LDAPSERVER`

> Note
>
> Synchronization with AD/LDAP settings in the System Console can be used to determine the connectivity and availability of arbitrary hosts. System admins concerned about this can use custom admin roles to limit access to modifying these settings. See the delegated granular administration documentation for details.

AD/LDAP port

This is the port Mattermost uses to connect to the AD/LDAP server. Numerical input. Default is **389**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `LdapPort` > `389` • Environment variable: `MM_LDAPSETTINGS_LDAPPORT`

Bind username

This is the username for the account Mattermost utilizes to perform an AD/LDAP search. This should be an account specific to Mattermost. Limit the permissions of the account to read-only access to the portion of the AD/LDAP tree specified in the **Base DN** setting. When using Active Directory, **Bind Username** should specify domain in `"DOMAIN/username"` format. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `BindUsername` • Environment variable: `MM_LDAPSETTINGS_BINDUSERNAME`

> Note
>
> This field is required. Anonymous bind is not currently supported.

Bind password

This is the password for the username given in the **Bind Username** setting. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `BindPassword` • Environment variable: `MM_LDAPSETTINGS_BINDPASSWORD`

Connection security

This setting controls the type of security Mattermost uses to connect to the AD/LDAP server, with these options: • **None**: **(Default for self-hosted deployments)** No encryption. With this option, it is **highly recommended** that the connection be secured outside of Mattermost, such as by a stunnel proxy. `config.json` option: `""` • **TLS**: **(Default for Cloud deployments)** Encrypts communication with TLS. `config.json` option: `"TLS"` • **STARTTLS**: Attempts to upgrade an existing insecure connection to a secure connection with TLS. `config.json` option: `"STARTTLS"`	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `ConnectionSecurity` > `""` • Environment variable: `MM_LDAPSETTINGS_CONNECTIONSECURITY`

Skip certificate verification

• **true**: Disables the certificate verification step for TLS and STARTTLS connections. Use this option for testing. **Do not use** this option when TLS is required in production. • **false**: **(Default)** Enables certification verification.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `SkipCertificateVerification` > `false` • Environment variable: `MM_LDAPSETTINGS_SKIPCERTIFICATEVERIFICATION`

Private key

Use this setting to upload the private key file from your LDAP authentication provider, if TLS client certificates are the primary authentication mechanism. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `PrivateKeyFile` • Environment variable: `MM_LDAPSETTINGS_PRIVATEKEYFILE`

Public certificate

Use this setting to upload the public TLS certificate from your LDAP authentication provider, if TLS client certificates are the primary authentication mechanism. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `PublicCertificateFile` • Environment variable: `MM_LDAPSETTINGS_PUBLICCERTIFICATEFILE`

Maximum login attempts

This setting determines the number of failed sign-in attempts a user can make before being locked out and required to go through a password reset by email. You can unlock the account in System Console on the users page. Setting this value lower than your LDAP maximum login attempts ensures that the users won't be locked out of your LDAP server because of failed login attempts in Mattermost. Numerical input. Default is **10**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `MaximumLoginAttempts` > `10` • Environment variable: `MM_LDAPSETTINGS_MAXIMUMLOGINATTEMPTS`

> Note
>
> - Adjust this value to align with your organization's authentication policies.
> - If a user's account is locked, you can unlock it manually by going to **System console > User Management > Users**.

User filters

Define how Mattermost identifies and filters users and groups from your AD/LDAP directory. Use the **Test Filters** button in this section to verify your filters work correctly before proceeding to other configuration steps.

Base DN

This is the **Base Distinguished Name** of the location in the AD/LDAP tree where Mattermost will start searching for users. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `BaseDN` • Environment variable: `MM_LDAPSETTINGS_BASEDN`

User filter

This setting accepts a [general syntax (https://www.ldapexplorer.com/en/manual/109010000-ldap-filter-syntax.htm)](https://www.ldapexplorer.com/en/manual/109010000-ldap-filter-syntax.htm) AD/LDAP filter that is applied when searching for user objects. Only the users selected by the query can access Mattermost. For example, to filter out disabled users, the filter is: `(&(objectCategory=Person)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))`. To filter by group membership, determine the `distinguishedName` of the group, then use group membership general syntax to format the filter. For example, if the security group `distinguishedName` is `CN=group1,OU=groups,DC=example,DC=com`, then the filter is: `(memberOf=CN=group1,OU=groups,DC=example,DC=com)`. The user must explicitly belong to this group for the filter to apply. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `UserFilter` • Environment variable: `MM_LDAPSETTINGS_USERFILTER`

> Note
>
> This filter uses the permissions of the **Bind Username** account to execute the search. This account should be specific to Mattermost and have read-only access to the portion of the AD/LDAP tree specified in the **Base DN** field.

Group filter

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting accepts a [general syntax (https://www.ldapexplorer.com/en/manual/109010000-ldap-filter-syntax.htm)](https://www.ldapexplorer.com/en/manual/109010000-ldap-filter-syntax.htm) AD/LDAP filter that is applied when searching for group objects. Only the groups selected by the query can access Mattermost. String input. Default is `(|(objectClass=group)(objectClass=groupOfNames)(objectClass=groupOfUniqueNames))`.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `GroupFilter` • Environment variable: `MM_LDAPSETTINGS_GROUPFILTER`

> Note
>
> This filter is only used when AD/LDAP Group Sync is enabled. See AD/LDAP Group Sync for more information.

Enable admin filter

• **true**: Enables the **Admin Filter** setting that designates system admins using an AD/LDAP filter. • **false**: **(Default)** Disables the **Admin Filter** setting.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `EnableAdminFilter` > `false` • Environment variable: `MM_LDAPSETTINGS_ENABLEADMINFILTER`

> Note
>
> If this setting is `false`, no additional users are designated as system admins by the filter. Users that were previously designated as system admins retain this role unless the filter is changed or removed.

Admin filter

This setting accepts an AD/LDAP filter that designates the selected users as system admins. Users are promoted to this role on their next sign-in or on the next scheduled AD/LDAP sync. If the Admin Filter is removed, users who are currently logged in retain their Admin role until their next sign-in. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `AdminFilter` • Environment variable: `MM_LDAPSETTINGS_ADMINFILTER`

Guest filter

This setting accepts an AD/LDAP filter to apply when searching for external users with Guest Access to Mattermost. Only users selected by the query can access Mattermost as Guests. See Guest Accounts for more information. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `GuestFilter` • Environment variable: `MM_LDAPSETTINGS_GUESTFILTER`

Account synchronization

Map AD/LDAP user attributes to Mattermost user profile fields. Use the **Test Attributes** button in this section to verify correct attribute mapping and data synchronization before proceeding to other configuration steps.

ID attribute

This is the attribute in the AD/LDAP server that is serves as a unique user identifier in Mattermost. The attribute should have a unique value that does not change, such as `objectGUID` or `entryUUID`. Confirm that these attributes are available in your environment before making any changes. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `IdAttribute` • Environment variable: `MM_LDAPSETTINGS_IDATTRIBUTE`

> Note
>
> If a user's ID Attribute changes, a new Mattermost account is created that is not associated with the previous account. If you need to change this field after users have signed-in, use the mmctl ldap idmigrate command.

Login ID attribute

This is the attribute in the AD/LDAP server that is used for signing-in to Mattermost. This is normally the same as the **Username Attribute**. If your team uses `domain\username` to sign-in to other services with AD/LDAP, you may enter `domain\username` in this field to maintain consistency between sites. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `LoginIdAttribute` • Environment variable: `MM_LDAPSETTINGS_LOGINIDATTRIBUTE`

Username attribute

This is the attribute in the AD/LDAP server that populates the username field in Mattermost. This attribute identifies users in the UI. For example, if a Username Attribute is set to `john.smith`, typing `@john` will show `@john.smith` as an auto-complete option, and posting a message with `@john.smith` will send a notification to that user. This is normally the same as the **Login ID Attribute**, but it can be mapped to a different attribute. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `UsernameAttribute` • Environment variable: `MM_LDAPSETTINGS_USERNAMEATTRIBUTE`

Email attribute

This is the attribute in AD/LDAP server that populates the email address field in Mattermost. Email notifications are sent to this address. The address may be seen by other Mattermost users depending on privacy settings. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting `LdapSettings` > `EmailAttribute` • Environment variable: `MM_LDAPSETTINGS_EMAILATTRIBUTE`

First name attribute

This is the attribute in the AD/LDAP server that populates the first name field in Mattermost. When set, users cannot edit their first name. When not set, users can edit their first name in their profile settings. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `FirstNameAttribute` • Environment variable: `MM_LDAPSETTINGS_FIRSTNAMEATTRIBUTE`

Last name attribute

This is the attribute in the AD/LDAP server that populates the last name field in Mattermost. When set, users cannot edit their last name. When not set, users can edit their last name as part of their profile settings. | String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `LastNameAttribute` • Environment variable: `MM_LDAPSETTINGS_LASTNAMEATTRIBUTE`

Nickname attribute

This is the attribute in the AD/LDAP server that populates the nickname field in Mattermost. When set, users cannot edit their nickname. When not set, users can edit their nickname as part of their profile settings. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `NicknameAttribute` • Environment variable: `MM_LDAPSETTINGS_NICKNAMEATTRIBUTE`

Position attribute

This is the attribute in the AD/LDAP server that populates the position field in Mattermost. When set, users cannot edit their position. When not set, users can edit their position as part of their profile settings. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `PositionAttribute` • Environment variable: `MM_LDAPSETTINGS_POSITIONATTRIBUTE`

Profile picture attribute

This is the attribute in the AD/LDAP server that syncs and locks the profile picture in Mattermost. The image is updated when users sign-in, not when Mattermost syncs with the AD/LDAP server. The image is not updated if the Mattermost image already matches the AD/LDAP image. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `PictureAttribute` • Environment variable: `MM_LDAPSETTINGS_PICTUREATTRIBUTE`

Group synchronization

Configure group mapping for AD/LDAP group synchronization. Use the **Test Group Attributes** button in this section to verify proper group attribute mapping before proceeding to other configuration steps.

Group display name attribute

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This is the AD/LDAP Group Display name attribute that populates the Mattermost group name field. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `GroupDisplayNameAttribute` • Environment variable: `MM_LDAPSETTINGS_GROUPDISPLAYNAMEATTRIBUTE`

> Note
>
> This attribute is only used when AD/LDAP Group Sync is enabled and it is **required**. See the AD/LDAP Group Sync documentation for more information.

Group ID attribute

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This is an AD/LDAP Group ID attribute that sets a unique identifier for groups. This should be a value that does not change, such as `entryUUID` or `objectGUID`. String input.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `GroupIdAttribute` • Environment variable: `MM_LDAPSETTINGS_GROUPIDATTRIBUTE`

> Note
>
> This attribute is only used when AD/LDAP Group Sync is enabled and it is **required**. See the AD/LDAP Group Sync documentation for more information.

Synchronization performance

Configure timing and performance settings for AD/LDAP synchronization. These settings control how often Mattermost syncs with your AD/LDAP server.

Synchronization interval (minutes)

This value determines how often Mattermost syncs with the AD/LDAP server by setting the number of minutes between each sync. Syncing with the AD/LDAP server will update Mattermost accounts to match any changes made to AD/LDAP attributes. Disabled AD/LDAP accounts become deactivated users in Mattermost, and any active sessions are revoked. Use the **AD/LDAP Synchronize Now** button to immediately revoke a session after disabling an AD/LDAP account. Numerical input. Default is **60**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `SyncIntervalMinutes` > `60` • Environment variable: `MM_LDAPSETTINGS_SYNCINTERVALMINUTES`

> Note
>
> LDAP syncs require a large number of database read queries. Monitor database load and adjust the sync interval to minimize performance degradation.

Maximum page size

This setting paginates the results of AD/LDAP server queries. Use this setting if your AD/LDAP server has a page size limit. The recommended setting is **1500**. This is the default AD/LDAP `MaxPageSize`. A page size of **0** disables pagination of results. Numerical input. Default is **0**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `MaxPageSize` > `0` • Environment variable: `MM_LDAPSETTINGS_MAXPAGESIZE`

Query timeout (seconds)

This setting determines the timeout period, in seconds, for AD/LDAP queries. Increase this value to avoid timeout errors when querying a slow server. Numerical input. Default is **60**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `QueryTimeout` > `60` • Environment variable: `MM_LDAPSETTINGS_QUERYTIMEOUT`

Synchronization history

View synchronization status and manually trigger AD/LDAP synchronization. This section includes the **AD/LDAP Synchronize Now** button for immediate synchronization.

AD/LDAP synchronize now

Use this button to immediately sync with the AD/LDAP server. The status of the sync is displayed in the table underneath the button (see the figure below). Following a manual sync, the next sync will occur after the time set in the **Synchronization Interval**.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> If a sync is `Pending` and does not complete, check that **Enable Synchronization with AD/LDAP** is set to `true` .

Config settings not available in the AD/LDAP Wizard

The following AD/LDAP configuration settings are available in the `config.json` file only and aren't available via the AD/LDAP wizard interface in the System Console.

Re-add removed members on sync

Enable this setting to re-add members of the LDAP group that were previously removed from group-synchronized teams or channels during LDAP synchronization. • **true**: Members of the LDAP group who were previously removed are re-added to group-synchronized teams or channels during LDAP synchronization. • **false**: **(Default)** Members of the LDAP group who were previously removed are not re-added to group-synchronized teams or channels during LDAP synchronization.	• System Config path: **Authentication > AD/LDAP** • `config.json` setting: `LdapSettings` > `ReAddRemovedMembers` • Environment variable: `MM_LDAPSETTINGS_READDREMOVEDMEMBERS`

> Note
>
> The mmctl ldap sync command takes precedence over this server configuration setting. If you have this setting disabled, and run the mmctl command with the `--include-removed-members` flag, removed members will be re-added during LDAP synchronization.

SAML 2.0

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > SAML 2.0**.

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Important

In line with Microsoft ADFS guidance, we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).

Enable login with SAML

Note

Available only on Enterprise and Professional (https://mattermost.com/pricing/) plans

• **true**: Enables sign-in with SAML. See SAML Single Sign-On to learn more. • **false**: **(Default)** Disables sign-in with SAML.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `Enable` > `false` • Environment variable: `MM_SAMLSETTINGS_ENABLE`

Enable synchronizing SAML accounts with AD/LDAP

Note

Available only on Enterprise and Professional (https://mattermost.com/pricing/) plans

• **true**: Mattermost updates configured Mattermost user attributes (ex. FirstName, Position, Email) with their values from AD/LDAP. This synchronization may deactivate Mattermost users or remove them from groups, teams, or channels. AD/LDAP synchronization must be enabled and configured through the settings under **Authentication > AD/LDAP**. • **false**: **(Default)** Disables syncing of SAML-authenticated Mattermost users with AD/LDAP.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `EnableSyncWithLdap` > `false` • Environment variable: `MM_SAMLSETTINGS_ENABLESYNCWITHLDAP`

See AD/LDAP Setup to learn more.

Ignore guest users when synchronizing with AD/LDAP

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: When syncing with the AD/LDAP server, Mattermost does not sync any information about SAML-authenticated Guest Users from the AD/LDAP server. Manage guest deactivation manually via **System Console > Users**. • **false**: **(Default)** Syncing Mattermost with the AD/LDAP server updates Guest User attributes and deactivates and removes SAML-authenticated accounts for Guest Users that are no longer active on the AD/LDAP server.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IgnoreGuestsLdapSync` > `false` • Environment variable: `MM_SAMLSETTINGS_IGNOREGUESTSLDAPSYNC`

For more information, see AD/LDAP Setup for details.

Override SAML bind data with AD/LDAP information

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: If the SAML ID attribute is configured, Mattermost overrides the SAML ID attribute with the AD/LDAP ID attribute. If the SAML ID attribute is not present, Mattermost overrides the SAML Email attribute with the AD/LDAP Email attribute. • **false**: **(Default)** Mattermost uses the email attribute to bind users to SAML. This setting is only available when SAML authentication is enabled and AD/LDAP synchronization is enabled.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `EnableSyncWithLdapIncludeAuth` > `false` • Environment variable: `MM_SAMLSETTINGS_ENABLESYNCWITHLDAPINCLUDEAUTH`

Note

- This setting should be **false** unless LDAP sync is enabled. Changing this setting from **true** to **false** will disable the override.
- SAML IDs must match LDAP IDs when the override is enabled.
- For more information, see AD/LDAP Setup for details.

Identity provider metadata URL

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting is the URL from which Mattermost requests setup metadata from the provider. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IdpMetadataURL` • Environment variable: `MM_SAMLSETTINGS_IDPMETADATAURL`

SAML SSO URL

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting is the URL where Mattermost sends a SAML request to start the login sequence. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IdpURL` • Environment variable: `MM_SAMLSETTINGS_IDPURL`

Identity provider issuer URL

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting is the issuer URL for the Identity Provider for SAML requests. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IdpDescriptorURL` • Environment variable: `MM_SAMLSETTINGS_IDPDESCRIPTORURL`

Identity provider public certificate

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

The public authentication certificate issued by your Identity Provider. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IdpCertificateFile` • Environment variable: `MM_SAMLSETTINGS_IDPCERTIFICATEFILE`

Verify signature

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: **(Default)** Mattermost checks that the SAML Response signature matches the Service Provider Login URL. • **false**: The signature is not verified. This is **not recommended** for production. Use this option for testing only.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `Verify` > `true` • Environment variable: `MM_SAMLSETTINGS_VERIFY`

Service provider login URL

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Enter the URL of your Mattermost server, followed by `/login/sso/saml`, i.e. `https://example.com/login/sso/saml`. Use HTTP or HTTPS depending on the configuration of the server. This setting is also known as the Assertion Consumer Service URL.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `AssertionConsumerServiceURL` • Environment variable: `MM_SAMLSETTINGS_ASSERTIONCONSUMERSERVICEURL`

Service provider identifier

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting is the unique identifier for the Service Provider, which in most cases is the same as the Service Provider Login URL. In ADFS, this must match the Relying Party Identifier. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `ServiceProviderIdentifier` • Environment variable: `MM_SAMLSETTINGS_SERVICEPROVIDERIDENTIFIER`

Enable encryption

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: **(Default)** Mattermost will decrypt SAML Assertions that are encrypted with your Service Provider Public Certificate. • **false**: Mattermost does not decrypt SAML Assertions. Use this option for testing only. It is **not recommended** for production.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `Encrypt` > `true` • Environment variable: `MM_SAMLSETTINGS_ENCRYPT`

Service provider private key

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting stores the private key used to decrypt SAML Assertions from the Identity Provider. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `PrivateKeyFile` • Environment variable: `MM_SAMLSETTINGS_PRIVATEKEYFILE`

Service provider public certificate

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting stores the certificate file used to sign a SAML request to the Identity Provider for a SAML login when Mattermost is initiating the login as the Service Provider. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `PublicCertificateFile` • Environment variable: `MM_SAMLSETTINGS_PUBLICCERTIFICATEFILE`

Sign request

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: Mattermost signs the SAML request with the Service Provider Private Key. • **false**: Mattermost does not sign the SAML request.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `SignRequest` • Environment variable: `MM_SAMLSETTINGS_SIGNREQUEST`

Signature algorithm

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting determines the signature algorithm used to sign the SAML request. Options are: `RSAwithSHA1`, `RSAwithSHA256`, `RSAwithSHA512`. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `SignatureAlgorithm` • Environment variable: `MM_SAMLSETTINGS_SIGNATUREALGORITHM`

Canonical algorithm

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting determines the canonicalization algorithm. With these options: • **Canonical1.0**: **(Default)** Exclusive XML Canonicalization 1.0 (omit comments) (https://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/) (`http://www.w3.org/2001/10/xml-exc-c14n#`). `config.json` setting: `Canonical1.0`. • **Canonical1.1**: Canonical XML 1.1 (omit comments) (https://www.w3.org/TR/2008/REC-xml-c14n11-20080502/) (`http://www.w3.org/2006/12/xml-c14n11`). `config.json` setting: `Canonical1.1`. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `CanonicalAlgorithm` • Environment variable: `MM_SAMLSETTINGS_CANONICALALGORITHM`

Email attribute

Note

Available only on Enterprise and Professional (https://mattermost.com/pricing/) plans

This setting determines the attribute from the SAML Assertion that populates the user email address field in Mattermost. Notifications are sent to this email address. This email address may be visible to other users, depending on how the system admin has set-up user privacy. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `EmailAttribute` • Environment variable: `MM_SAMLSETTINGS_EMAILATTRIBUTE`

Username attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting determines the SAML Assertion attribute that populates the username field in the Mattermost UI. This attribute identifies users in the UI. For example, if a username is set to `john.smith`, typing `@john` will show `@john.smith` as an auto-complete option, and posting a message with `@john.smith` will send a notification to that user. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `UsernameAttribute` • Environment variable: `MM_SAMLSETTINGS_USERNAMEATTRIBUTE`

Id attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute used to bind users from SAML to users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `IdAttribute` • Environment variable: `MM_SAMLSETTINGS_IDATTRIBUTE`

Guest attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute used to apply a Guest role to users in Mattermost. See the Guest Accounts documentation for more information. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `GuestAttribute` • Environment variable: `MM_SAMLSETTINGS_GUESTATTRIBUTE`

Enable admin attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: System admin status is determined by the SAML Assertion attribute set in **Admin attribute**. • **false**: **(Default)** System admin status is **not** determined by the SAML Assertion attribute.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `EnableAdminAttribute` > `false` • Environment variable: `MM_SAMLSETTINGS_ENABLEADMINATTRIBUTE`

Admin attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the attribute in the SAML Assertion for designating system admins. Users are automatically promoted to this role when logging in to Mattermost. If the Admin attribute is removed, users that are logged in retain Admin status. The role is revoked only when users log out. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `AdminAttribute` • Environment variable: `MM_SAMLSETTINGS_ADMINATTRIBUTE`

First name attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute that populates the first name of users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `FirstNameAttribute` • Environment variable: `MM_SAMLSETTINGS_FIRSTNAMEATTRIBUTE`

Last name attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute that populates the last name of users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `LastNameAttribute` • Environment variable: `MM_SAMLSETTINGS_LASTNAMEATTRIBUTE`

Nickname attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute that populates the nickname of users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `NicknameAttribute` • Environment variable: `MM_SAMLSETTINGS_NICKNAMEATTRIBUTE`

Position attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute that populates the position (job title or role at company) of users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `PositionAttribute` • Environment variable: `MM_SAMLSETTINGS_POSITIONATTRIBUTE`

Preferred language attribute

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) This setting determines the SAML Assertion attribute that populates the language preference of users in Mattermost. String input.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `LocaleAttribute` • Environment variable: `MM_SAMLSETTINGS_LOCALEATTRIBUTE`

Login button text

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

(Optional) The text that appears in the login button on the sign-in page. String input. Default is **SAML**.	• System Config path: **Authentication > SAML 2.0** • `config.json` setting: `SamlSettings` > `LoginButtonText` • Environment variable: `MM_SAMLSETTINGS_LOGINBUTTONTEXT`

OAuth 2.0

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > OAuth 2.0**. Settings for GitLab OAuth authentication can also be accessed under **Authentication > GitLab** in self-hosted deployments.

Use these settings to configure OAuth 2.0 for account creation and login.

Select OAuth 2.0 service provider

Use this setting to enable OAuth and specify the service provider, with these options: • **Do not allow login via an OAuth 2.0 provider** • **GitLab** (Available in all plans; see GitLab 2.0 OAuth settings) • **Google Apps** (Available in Mattermost Enterprise and Professional; see Google OAuth 2.0 settings) • **Entra ID** (Available in Mattermost Enterprise and Professional; see Entra ID OAuth 2.0 settings)	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: N/A • Environment variable: N/A

GitLab OAuth 2.0 settings

Available onall plansCloudandself-hosteddeployments

> Note
>
> For Enterprise subscriptions, GitLab settings can be found under **OAuth 2.0**

Enable OAuth 2.0 authentication with GitLab

• **true**: Allows team and account creation using GitLab OAuth authentication. Input the **Secret** and **ID** credentials to configure. • **false**: **(Default)** Disables GitLab OAuth authentication.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `Enable` > `false` • Environment variable: `MM_GITLABSETTINGS_ENABLE`

GitLab OAuth 2.0 Application ID

This setting holds the OAuth Application ID from GitLab. Generate the ID by these steps: 1. Login to your GitLab account. 2. Go to **Profile Settings > Applications > New Application** and enter a name. 3. Enter the Redirect URLs: `https://<your-mattermost-url>/login/gitlab/complete` and `https://<your-mattermost-url>/signup/gitlab/complete`. 4. Take the Application ID provided by GitLab and enter it in the Mattermost System Console field, `config.json` setting, or Environment variable. String input.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `Id` • Environment variable: `MM_GITLABSETTINGS_ID`

> Note
>
> GitLab provides the Application Secret Key along with the the ID.

GitLab OAuth 2.0 Application secret key

This setting holds the OAuth Application Secret Key from GitLab. The key is generated at the same time as the **Application ID** (see GitLab OAuth 2.0 Application ID). Enter the key provided by GitLab in the Mattermost System Console field, `config.json` setting, or Environment variable. String input.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `Secret` • Environment variable: `MM_GITLABSETTINGS_SECRET`

GitLab OAuth 2.0 site URL

This setting holds the URL of your GitLab instance, e.g. `https://example.com:3000`. Use `http://` if SSL is not enabled on your GitLab instance.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: N/A • Environment variable: N/A

GitLab OAuth 2.0 User API endpoint

This setting holds the URL of your GitLab User API endpoint, e.g. `https://<your-gitlab-url>/api/v3/user`. Use `http://` if SSL is not enabled on your GitLab instance. Enter the URL in the Mattermost System Console field, `config.json` setting, or Environment variable. String input.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `UserAPIEndpoint` • Environment variable: `MM_GITLABSETTINGS_USERAPIENDPOINT`

GitLab OAuth 2.0 Auth endpoint

This setting holds the URL of your GitLab Auth endpoint, e.g. `https://<your-gitlab-url>/oauth/authorize`. Use `http://` if SSL is not enabled on your GitLab instance. Enter the URL in the Mattermost System Console field, `config.json` setting, or Environment variable. String input.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `AuthEndpoint` • Environment variable: `MM_GITLABSETTINGS_AUTHENDPOINT`

GitLab OAuth 2.0 Token endpoint

This setting holds the URL of your GitLab OAuth Token endpoint, e.g. `https://<your-gitlab-url>/oauth/token`. Use `http://` if SSL is not enabled on your GitLab instance. Enter the URL in the Mattermost System Console field, `config.json` setting, or Environment variable. String input.	• System Config path: **Authentication > OAuth 2.0 (or GitLab)** • `config.json` setting: `GitLabSettings` > `TokenEndpoint` • Environment variable: `MM_GITLABSETTINGS_TOKENENDPOINT`

Google OAuth 2.0 settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Enable OAuth 2.0 authentication with Google

• **true**: Allows team and account creation using Google OAuth authentication. Input the **Client ID** and **Client Secret** credentials to configure. • **false**: **(Default)** Disables Google OAuth authentication. See Google Single Sign-On implementation instructions.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `Enable` > `false` • Environment variable: `MM_GOOGLESETTINGS_ENABLE`

Google OAuth 2.0 Client ID

This setting stores the OAuth Client ID from Google. Generate the ID by going to the **Credentials** section of the Google Cloud Platform APIs & Services menu and selecting **Create Credentials > OAuth client ID**. See Google Single Sign-On for instructions that can be used to implement Google OAuth or OpenID authentication. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `Id` • Environment variable: `MM_GOOGLESETTINGS_ID`

Google OAuth 2.0 Client secret

This setting stores the OAuth Client Secret from Google. The Secret is generated at the same time as the Client ID. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `Secret` • Environment variable: `MM_GOOGLESETTINGS_SECRET`

Google OAuth 2.0 User API endpoint

We recommend `https://people.googleapis.com/v1/people/me?personFields=names,emailAddresses,nicknames,metadata` as the User API Endpoint. Otherwise, enter a custom endpoint in `config.json` with HTTP, or HTTPS, if available on the API server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `UserAPIEndpoint` • Environment variable: `MM_GOOGLESETTINGS_USERAPIENDPOINT`

Google OAuth 2.0 Auth endpoint

We recommend `https://accounts.google.com/o/oauth2/v2/auth` as the Auth Endpoint. Otherwise, enter a custom endpoint in `config.json` with HTTP, or HTTPS, if available on the server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `AuthEndpoint` • Environment variable: `MM_GOOGLESETTINGS_AUTHENDPOINT`

Google OAuth 2.0 Token endpoint

We recommend `https://www.googleapis.com/oauth2/v4/token` as the Token Endpoint. Otherwise, enter a custom endpoint in `config.json` with HTTP, or HTTPS, if available on the server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `GoogleSettings` > `TokenEndpoint` • Environment variable: `MM_GOOGLESETTINGS_TOKENENDPOINT`

Entra ID OAuth 2.0 settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

> Note
>
> In line with Microsoft ADFS guidance we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).

Enable OAuth 2.0 Authentication with Entra ID

• **true**: Allows team and account creation using Entra ID OAuth authentication. • **false**: **(Default)** Disables Entra ID OAuth authentication.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `Enable` > `false` • Environment variable: `MM_OFFICE365SETTINGS_ENABLE`

> Note
>
> See the Entra ID Single Sign-On documentation for details.

Entra ID OAuth 2.0 Application ID

This setting holds the **Application ID** generated when configuring Entra ID as a Single Sign-On service through the Microsoft Azure Portal. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `Id` • Environment variable: `MM_OFFICE365SETTINGS_ID`

> Note
>
> See the Entra ID Single Sign-On documentation for details.

Entra ID OAuth 2.0 Application secret password

This setting holds the **Application Secret Password** generated when configuring Entra ID as a Single Sign-On service through the Microsoft Azure Portal. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `Secret` • Environment variable: `MM_OFFICE365SETTINGS_SECRET`

> Note
>
> See the Entra ID Single Sign-On documentation for details.

Entra ID OAuth 2.0 Directory (tenant) ID

This setting holds the **Directory (tenant) ID** set for Mattermost through the Azure Portal. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `DirectoryId` • Environment variable: `MM_OFFICE365SETTINGS_DIRECTORYID`

> Note
>
> See the Entra ID Single Sign-On documentation for details.

Entra ID OAuth 2.0 User API endpoint

We recommend `https://graph.microsoft.com/v1.0/me` as the User API Endpoint. Otherwise, enter a custom endpoint in `config.json` with `http`, or `https`, if available on the server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `UserAPIEndpoint` • Environment variable: `MM_OFFICE365SETTINGS_USERAPIENDPOINT`

Entra ID OAuth 2.0 Auth endpoint

We recommend `https://login.microsoftonline.com/common/oauth2/v2.0/authorize` as the Auth Endpoint. Otherwise, enter a custom endpoint in `config.json` with `http`, or `https`, if available on the server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `AuthEndpoint` • Environment variable: `MM_OFFICE365SETTINGS_AUTHENDPOINT`

Entra ID OAuth 2.0 Token endpoint

We recommend `https://login.microsoftonline.com/common/oauth2/v2.0/token` as the Token Endpoint. Otherwise, enter a custom endpoint in `config.json` with `http`, or `https`, if available on the server. String input.	• System Config path: **Authentication > OAuth 2.0** • `config.json` setting: `Office365Settings` > `TokenEndpoint` • Environment variable: `MM_OFFICE365SETTINGS_TOKENENDPOINT`

OpenID Connect

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > OpenID Connect**.

Select OpenID Connect service provider

Use this setting to enable OpenID Connect, with these options: • **Do not allow login via an OpenID provider** • **GitLab** (see settings) • **Google Apps** (see settings) • **Entra ID** (see settings) • **OpenID Connect (Other)** (see settings)	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> **GitLab** OpenID is available in all plans. All other providers require Mattermost Enterprise or Professional.

GitLab OpenID settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Enable OpenID Connect authentication with GitLab

• **true**: Allows team and account creation using GitLab OpenID Connect authentication. • **false**: **(Default)** Disables GitLab OpenID Connect authentication.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GitLabSettings` > `Enable` > `false` • Environment variable: `MM_GITLABSETTINGS_ENABLE`

> Note
>
> See the GitLab Single Sign-On documentation for details.

GitLab OpenID site URL

This setting stores the URL of your GitLab instance, e.g. **https://example.com:3000**. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> See **Step 2** of the GitLab Single Sign-On documentation for details.

GitLab OpenID Discovery endpoint

This setting is prepopulated with the Discovery Endpoint for GitLab OpenID Connect. String input. Default is `https://gitlab.com/.well-known/openid-configuration`	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GitLabSettings` > `DiscoveryEndpoint` • Environment variable: `MM_GITLABSETTINGS_DISCOVERYENDPOINT`

> Note
>
> See **Step 2** of the GitLab Single Sign-On documentation for details.

GitLab OpenID Client ID

This setting stores the **Application ID** generated by GitLab. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GitLabSettings` > `Id` • Environment variable: `MM_GITLABSETTINGS_ID`

> Note
>
> See **Step 2** of the GitLab Single Sign-On documentation for details.

GitLab OpenID Client secret

This setting stores the **Application Secret Key** generated by GitLab. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GitLabSettings` > `Secret` • Environment variable: `MM_GITLABSETTINGS_SECRET`

> Note
>
> See **Step 2** of the GitLab Single Sign-On documentation for details.

Google OpenID settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Enable OpenID Connect authentication with Google

> - ***true***: *Allow team creation and account signup using Google OpenID Connect.*
> - ***false***: ***(Default)*** *Google OpenID Connect cannot be used for team creation or account signup.*

• **true**: Allows team and account creation using Google OpenID authentication. • **false**: **(Default)** Disables Google OpenID authentication. See Google Single Sign-On implementation instructions.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GoogleSettings` > `Enable` > `false` • Environment variable: `MM_GOOGLESETTINGS_ENABLE`

Google OpenID Discovery endpoint

This setting is prepopulated with the Discovery Endpoint for Google OpenID Connect. See Configure Mattermost for Google Apps SSO. String input. Default is `https://accounts.google.com/.well-known/openid-configuration`	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GoogleSettings` > `DiscoveryEndpoint` • Environment variable: `MM_GOOGLESETTINGS_DISCOVERYENDPOINT`

Google OpenID Client ID

This setting stores the Client ID generated by Google. See Google Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GoogleSettings` > `Id` • Environment variable: `MM_GOOGLESETTINGS_ID`

Google OpenID Client secret

This setting stores the Client Secret generated by Google. See Google Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `GoogleSettings` > `Secret` • Environment variable: `MM_GOOGLESETTINGS_SECRET`

Entra ID OpenID settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

> Note
>
> In line with Microsoft ADFS guidance, we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).

Enable OpenID Connect authentication with Entra ID

• **true**: Allows team and account creation using Entra ID OpenID Connect authentication. • **false**: **(Default)** Disables Entra ID OpenID Connect authentication. See Entra ID Single Sign-On implementation instructions.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `Office365Settings` > `Enable` > `false` • Environment variable: `MM_OFFICE365SETTINGS_ENABLE`

Entra ID OpenID Directory (tenant) ID

This setting holds the Directory (tenant) ID set for Mattermost through the Microsoft Azure Portal. See Entra ID Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `Office365Settings` > `DirectoryId` • Environment variable: `MM_OFFICE365SETTINGS_DIRECTORYID`

Entra ID OpenID Discovery endpoint

This setting is prepopulated with the Discovery Endpoint for Entra ID OpenID Connect. See Entra ID Single Sign-On implementation instructions. String input. Default is `https://login.microsoftonline.com/common/v2.0/.well-known/openid-configuration`	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `Office365Settings` > `DiscoveryEndpoint` • Environment variable: `MM_OFFICE365SETTINGS_DISCOVERYENDPOINT`

Entra ID Client ID

This setting stores the **Application (client) ID** generated through the Microsoft Azure Portal. See Entra ID Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `Office365Settings` > `Id` • Environment variable: `MM_OFFICE365SETTINGS_ID`

Entra ID Client secret

This setting stores the **Client Secret** generated through the Microsoft Azure Portal. See Entra ID Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `Office365Settings` > `Secret` • Environment variable: `MM_OFFICE365SETTINGS_SECRET`

OpenID Connect (other) settings

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Enable OpenID Connect authentication with other service providers

Available onEnterprise and Professional plansCloudandself-hosteddeployments

• **true**: Allows team and account creation using other OpenID Connect service providers. • **false**: **(Default)** Disables OpenID Connect authentication with other service providers. See OpenID Connect Single Sign-On implementation instructions.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `Enable` > `false` • Environment variable: `MM_OPENIDSETTINGS_ENABLE`

OpenID Connect (other) Button name

This setting is the text for the OpenID login button. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `ButtonText` • Environment variable: `MM_OPENIDSETTINGS_BUTTONTEXT`

OpenID Connect (other) Button color

This setting is the color of the OpenID login button. Use a hex code with a #-sign before the code, for example `#145DBF`. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `ButtonColor` • Environment variable: `MM_OPENIDSETTINGS_BUTTONCOLOR`

OpenID Connect (other) Discovery endpoint

This setting stores the Discovery Endpoint URL from the OpenID provider. The URL should be in the format of `https://myopenid.provider.com/{my_organization}/.well-known/openid-configuration`. See OpenID Connect Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `DiscoveryEndpoint` • Environment variable: `MM_OPENIDSETTINGS_DISCOVERYENDPOINT`

> Note
>
> The **Discovery Endpoint** setting can be used to determine the connectivity and availability of arbitrary hosts. System admins concerned about this can use custom admin roles to limit access to modifying these settings. See the delegated granular administration documentation for details.

OpenID Connect (other) Client ID

This setting stores the Client ID from the OpenID provider. See OpenID Connect Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `Id` • Environment variable: `MM_OPENIDSETTINGS_ID`

OpenID Connect (other) Client secret

This setting stores the Client Secret from the OpenID provider. See OpenID Connect Single Sign-On implementation instructions. String input.	• System Config path: **Authentication > OpenID Connect** • `config.json` setting: `OpenIdSettings` > `Secret` • Environment variable: `MM_OPENIDSETTINGS_SECRET`

Guest access

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Authentication > Guest Access**.

Enable guest access

• **true**: Enables the guest account feature. • **false**: **(Default)** Disables the guest account feature.	• System Config path: **Authentication > Guest Access** • `config.json` setting: `GuestAccountsSettings` > `Enable` > `false` • Environment variable: `MM_GUESTACCOUNTSSETTINGS_ENABLE`

> Note
>
> For billing purposes, activated guest accounts do consume a licensed seat, which is returned when the guest account is deactivated.This means that guest accounts count as a paid user in your Mattermost workspace.

Whitelisted guest domains

Use this setting to restrict the creation of guest accounts. When set, guest accounts require a verified email address from one of the listed domains. String input of one or more domains, separated by commas.	• System Config path: **Authentication > Guest Access** • `config.json` setting: `GuestAccountsSettings` > `RestrictCreationToDomains` • Environment variable: `MM_GUESTACCOUNTSSETTINGS_RESTRICTCREATIONTODOMAINS`

Enforce multi-factor authentication

• **true**: Guests must use MFA. Guests without MFA are redirected to the setup page until MFA is configured. • **false**: **(Default)** MFA is not required for guests.	• System Config path: **Authentication > Guest Access** • `config.json` setting: `GuestAccountsSettings` > `EnforceMultifactorAuthentication` > `false` • Environment variable: `MM_GUESTACCOUNTSSETTINGS_ENFORCEMULTIFACTORAUTHENTICATION`

> Note
>
> This setting defaults to false and cannot be changed if MFA isn't enforced for non-guest users.

Show guest tag

• **true**: **(Default)** Guest tags are visible in Mattermost. • **false**: Guest tags aren't visible in Mattermost.	• System Config path: **Authentication > Guest Access** • `config.json` setting: `GuestAccountsSettings` > `HideTags` > `true` • Environment variable: `MM_GUESTACCOUNTSSETTINGS_HIDETAGS`

> Note
>
> This configuration setting applies to all Mattermost clients, including web, desktop app, and mobile app. See the guest accounts documentation for details.

On this page

- Authentication configuration settings
  - Signup
    - Enable account creation
    - Restrict account creation to specified email domains
    - Enable open server
    - Enable email invitations
    - Invalidate pending email invites
  - Email
    - Enable account creation with email
    - Require email verification
    - Enable sign-in with email
    - Enable sign-in with username
  - Password
    - Minimum password length
    - Password requirements
    - Maximum login attempts
    - Enable forgot password link
  - MFA
    - Enable multi-factor authentication
    - Enforce multi-factor authentication
  - AD/LDAP
    - Connection settings
      - Enable sign-in with AD/LDAP
      - Enable synchronization with AD/LDAP
      - Login field name

    - AD/LDAP server
    - AD/LDAP port
    - Bind username
    - Bind password
    - Connection security
    - Skip certificate verification
    - Private key
    - Public certificate
    - Maximum login attempts
- User filters
    - Base DN
    - User filter
    - Group filter
    - Enable admin filter
    - Admin filter
    - Guest filter
- Account synchronization
    - ID attribute
    - Login ID attribute
    - Username attribute
    - Email attribute
    - First name attribute
    - Last name attribute
    - Nickname attribute
    - Position attribute
    - Profile picture attribute
- Group synchronization
    - Group display name attribute
    - Group ID attribute
- Synchronization performance
    - Synchronization interval (minutes)
    - Maximum page size
    - Query timeout (seconds)

    - Synchronization history
        - AD/LDAP synchronize now
    - Config settings not available in the AD/LDAP Wizard
        - Re-add removed members on sync
- SAML 2.0
    - Enable login with SAML
    - Enable synchronizing SAML accounts with AD/LDAP
    - Ignore guest users when synchronizing with AD/LDAP
    - Override SAML bind data with AD/LDAP information
    - Identity provider metadata URL
    - SAML SSO URL
    - Identity provider issuer URL
    - Identity provider public certificate
    - Verify signature
    - Service provider login URL
    - Service provider identifier
    - Enable encryption
    - Service provider private key
    - Service provider public certificate
    - Sign request
    - Signature algorithm
    - Canonical algorithm
    - Email attribute
    - Username attribute
    - Id attribute
    - Guest attribute
    - Enable admin attribute
    - Admin attribute
    - First name attribute
    - Last name attribute
    - Nickname attribute
    - Position attribute
    - Preferred language attribute

    - Login button text
- OAuth 2.0
    - Select OAuth 2.0 service provider
        - GitLab OAuth 2.0 settings
            - Enable OAuth 2.0 authentication with GitLab
            - GitLab OAuth 2.0 Application ID
            - GitLab OAuth 2.0 Application secret key
            - GitLab OAuth 2.0 site URL
            - GitLab OAuth 2.0 User API endpoint
            - GitLab OAuth 2.0 Auth endpoint
            - GitLab OAuth 2.0 Token endpoint
        - Google OAuth 2.0 settings
            - Enable OAuth 2.0 authentication with Google
            - Google OAuth 2.0 Client ID
            - Google OAuth 2.0 Client secret
            - Google OAuth 2.0 User API endpoint
            - Google OAuth 2.0 Auth endpoint
            - Google OAuth 2.0 Token endpoint
        - Entra ID OAuth 2.0 settings
            - Enable OAuth 2.0 Authentication with Entra ID
            - Entra ID OAuth 2.0 Application ID
            - Entra ID OAuth 2.0 Application secret password
            - Entra ID OAuth 2.0 Directory (tenant) ID
            - Entra ID OAuth 2.0 User API endpoint
            - Entra ID OAuth 2.0 Auth endpoint
            - Entra ID OAuth 2.0 Token endpoint
- OpenID Connect
    - Select OpenID Connect service provider
        - GitLab OpenID settings
            - Enable OpenID Connect authentication with GitLab
            - GitLab OpenID site URL
            - GitLab OpenID Discovery endpoint
            - GitLab OpenID Client ID
            - GitLab OpenID Client secret

        - Google OpenID settings
            - Enable OpenID Connect authentication with Google
            - Google OpenID Discovery endpoint
            - Google OpenID Client ID
            - Google OpenID Client secret
        - Entra ID OpenID settings
            - Enable OpenID Connect authentication with Entra ID
            - Entra ID OpenID Directory (tenant) ID
            - Entra ID OpenID Discovery endpoint
            - Entra ID Client ID
            - Entra ID Client secret
        - OpenID Connect (other) settings
            - Enable OpenID Connect authentication with other service providers
            - OpenID Connect (other) Button name
            - OpenID Connect (other) Button color
            - OpenID Connect (other) Discovery endpoint
            - OpenID Connect (other) Client ID
            - OpenID Connect (other) Client secret
    - Guest access
        - Enable guest access
        - Whitelisted guest domains
        - Enforce multi-factor authentication
        - Show guest tag

Bleve search (experimental)

Bleve is a search engine that uses Lucene-style full-text search and indexing. This style of search and indexing helps overcome limitations of the default database search such as challenges with characters and advanced search capabilities.

The Bleve search engine works as a library integrated into the Mattermost codebase. As it generates indexes in the filesystem of the server that it is running on, it doesn't require an external server to function. Because of this, Bleve should not be enabled in High Availability deployments.

> Note
>
> Bleve search uses the scorch index type on newly-created indexes. This new index type features efficiency improvements and indexes that use significantly less disk space. Go to **System Console > Experimental > Bleve** and select **Purge Index** to run a purge operation. When that's complete, select **Index Now** to reindex. Bleve remains compatible with existing indexes, so currently indexed data will continue to work if a purge and reindex isn't run.

Configuring Bleve in Mattermost

Follow these steps to configure the Mattermost server to use Bleve and generate required indexes. Once the configuration is saved, new posts made to the database will be automatically indexed with Bleve.

Note: During indexing, search results may be incomplete until the indexing job is complete.

1. Open **System Console > Experimental > Bleve**.
2. Set **Enable Bleve Indexing** to **true** to enable the other settings on the page.

3. Set the directory path to use for storing Bleve indexes (e.g.: `/var/opt/mattermost/bleveindexes`). The user running Mattermost should have permissions to access the directory. See our configuration settings documentation for details.
4. Save the configuration.
5. Select **Index Now**. All users, channels, and posts in the database will be indexed oldest to newest.
6. Set **Enable Bleve for search queries** to **true**.
7. Set **Enable Bleve for autocomplete queries** to **true**.

> Note
>
> Search results for files shared before upgrading to Mattermost Server v5.35 may be incomplete until an extraction command is run using the mmctl. After running this command, the search index must be rebuilt. Go to **System Console > Experimental > Bleve > Bulk Indexing**, then select **Index Now** to rebuild the search index to include older file contents.

Using Bleve search

The following conditions are applied when using Bleve search:

- **Unquoted terms:** Search terms that contain non-alphanumeric characters/special characters outside of quotation marks are removed. For example, using `abcd "**" && abc` as a search term will return results for a search for `abcd "**" abc` as the `&&` characters weren't within the quotation marks.
- **Wildcard search:** Wildcard search (e.g., `abc*`) is supported.

How does search work with Bleve disabled?

Mattermost performs full text searches against the database unless you have an Enterprise license and enterprise search configured.

On this page

- Bleve search (experimental)
  - Configuring Bleve in Mattermost
  - Using Bleve search
  - How does search work with Bleve disabled?

Calls self-hosted deployment

Available onall plansself-hosteddeployments

Mattermost Calls is an excellent option for organizations demanding enhanced security and control over their communication infrastructure. Calls is designed to operate securely in self-hosted deployments, including air-gapped environments (https://docs.mattermost.com/configure/calls-deployment.html#air-gapped-deployments), ensuring private communication without reliance on public internet connectivity with flexible configuration options for complex network requirements.

This document provides information on how to successfully make the Calls plugin work on self-hosted deployments. It also outlines some of the most common deployment strategies with example diagrams, and provides the deployment guidelines for the recording, transcription, and live captions service.

Terminology

- WebRTC (https://bloggeek.me/webrtcglossary/webrtc-2/): The set of underlying protocols/specifications on top of which calls are implemented.
- **RTC (Real Time Connection)**: The real-time connection. This is the channel used to send media tracks (audio/video/screen).
- **WS (WebSocket)**: The WebSocket connection. This is the channel used to set up a connection (signaling process).
- NAT (Network Address Translation) (https://bloggeek.me/webrtcglossary/nat/): A networking technique to map IP addresses.
- STUN (Session Traversal Utilities for NAT) (https://bloggeek.me/webrtcglossary/stun/): A protocol/service used by WebRTC clients to help traversing NATs. On the server side it's mainly used to figure out the public IP of the instance.

- TURN (Traversal Using Relays around NAT) (https://bloggeek.me/webrtcglossary/turn/): A protocol/service used to help WebRTC clients behind strict firewalls connect to a call through media relay.

Plugin components

- **Calls plugin**: This is the main entry point and a requirement to enable channel calls.
- **rtcd**: This is an optional service that can be deployed to offload all the functionality and data processing involved with the WebRTC connections. Read more about when and why to use rtcd below.

Requirements

Server

- Run Mattermost server on a secure (HTTPs) connection. This is a necessary requirement on the client to allow capturing devices (e.g., microphone, screen). See the config TLS (https://docs.mattermost.com/deploy/server/setup-tls.html) section for more info.
- See network requirements below.

Client

- Clients need to be able to connect (send and receive data) to the instance hosting the calls through the UDP port configured as `RTC Server Port`. If this is not possible a TURN server should be used to achieve connectivity.
- Depending on the platform or operating system, clients may need to grant additional permissions to the application (e.g., browser, desktop app) to allow them to capture audio inputs or share the screen.

Network

Service	Ports	Protocols	Source	Target	Purpose
API (Calls plugin)	80,443	TCP (incoming)	Mattermost clients (web/desktop/mobile)	Mattermost instance (Calls plugin)	To allow for HTTP and WebSocket connectivity from clients to Calls plugin. This API is exposed on the same connection as Mattermost, so there's likely no need to change anything.
RTC (Calls plugin or `rtcd`)	8443	UDP (incoming)	Mattermost clients (Web/Desktop/Mobile)	Mattermost instance or `rtcd` service	To allow clients to establish connections that transport calls related media (e.g. audio, video). This should be open on any network component (e.g. NAT, firewalls) in between the instance running the plugin (or `rtcd`) and the clients joining calls so that UDP traffic is correctly routed both ways (from/to clients).
RTC (Calls plugin or `rtcd`)	8443	TCP (incoming)	Mattermost clients (Web/Desktop/Mobile)	Mattermost instance or `rtcd` service	To allow clients to establish connections that transport calls related media (e.g. audio, video). This should be open on any network component (e.g. NAT, firewalls) in between the instance running the plugin (or `rtcd`) and the clients joining calls so that TCP traffic is correctly routed both ways (from/to clients). This

Service	Ports	Protocols	Source	Target	Purpose
					can be used as a backup channel in case clients are unable to connect using UDP. It requires `rtcd` version >= v0.11 and Calls version >= v0.17.
API (`rtcd`)	8045	TCP (incoming)	Mattermost instance(s) (Calls plugin)	`rtcd` service	To allow for HTTP/WebSocket connectivity from Calls plugin to `rtcd` service. Can be expose internally as the service only needs to be reachable by the instance(s) running the Mattermost server.
STUN (Calls plugin or `rtcd`)	3478	UDP (outgoing)	Mattermost Instance(s) (Calls plugin) or `rtcd` service	Configured STUN servers	(Optional) To allow for either Calls plugin or `rtcd` service to discover their instance public IP. Only needed if configuring STUN/TURN servers. This requirement does not apply when manually setting an IP or hostname through the ICE Host Override (https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override) config option.

Air-gapped deployments

Mattermost Calls can function in air-gapped environments. Exposing Calls to the public internet is only necessary when users need to connect from outside the local network, and no existing method supports that connection. In such setups:

- Users should connect from within the private/local network. This can be done on-premises, through a VPN, or via virtual machines.
- Configuring a STUN server is unnecessary, as all connections occur within the local network.
- The ICE Host Override (https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override) configuration setting can be optionally set with a local IP address (e.g., 192.168.1.45), depending on the specific network configuration and topology.

Limitations

- All Mattermost customers can start, join, and participate in 1:1 audio calls with optional screen sharing.
- For group calls up to 50 concurrent users, Mattermost Enterprise, Professional, or Mattermost Cloud is required.
- Enterprise customers can also record calls (https://docs.mattermost.com/collaborate/make-calls.html#record-a-call), enable live text captions (https://docs.mattermost.com/collaborate/make-calls.html#live-captions-during-calls) during calls, and transcribe recorded calls (https://docs.mattermost.com/collaborate/make-calls.html#transcribe-recorded-calls). We recommend that Enterprise self-hosted customers looking for group calls beyond 50 concurrent users consider using the dedicated rtcd service.
- For Mattermost self-hosted deployments, System admins need to enable and configure the plugin using the System Console (https://docs.mattermost.com/configure/plugins-configuration-settings.html#calls). The default maximum number of participants is unlimited; however, we recommend a maximum of 50 participants per call. Maximum call participants is configurable by going to **System Console > Plugin**

Management > Calls > Max call participants. Call participant limits greatly depends on instance resources. For more details, refer to the performance section below.

Configuration

For Mattermost self-hosted customers, the calls plugin is pre-packaged, installed, and enabled. Configuration to allow end-users to use it can be found in the [System Console (https://docs.mattermost.com/configure/plugins-configuration-settings.html#calls)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#calls).

Modes of operation

Depending on how the Mattermost server is running, there are several modes under which the Calls plugin can operate. Please refer to the section below on the rtcd service to learn about the `rtcd` and the Selective Forwarding Unit (SFU).

Mattermost deployment	SFU	SFU deployment
Single instance	integrated	
Single instance	rtcd	
High availability cluster-based	integrated	clustered
High availability cluster-based	rtcd	

Single instance

Integrated

This is the default mode when first installing the plugin on a single Mattermost instance setup. The WebRTC service is integrated in the plugin itself and runs alongside the Mattermost server.

rtcd

An external, dedicated and scalable WebRTC service (`rtcd`) is used to handle all calls media routing.

High availability cluster-based

Clustered

This is the default mode when running the plugin in a high availability cluster-based deployment. Every Mattermost node will run an instance of the plugin that includes a WebRTC service. Calls are distributed across all available nodes through the existing load-balancer: a call is hosted on the instance where the initiating websocket connection (first client to join) is made. A single call will be hosted on a single cluster node.

rtcd (High Availability)

Performance

Calls performance primarily depends on two resources: CPU and bandwidth (both network latency and overall throughput). The final consumption exhibits quadratic growth with the number of clients transmitting and receiving media.

As an example, a single call with 10 participants of which two are unmuted (transmitting voice data) will generally consume double the resources than the same call with a single participant unmuted. What ultimately counts towards performance is the overall number of concurrent media flows (in/out) across the server.

Benchmarks

Here are the results from internally conducted performance and ceiling tests on a dedicated `rtcd` instance:

Deployment specifications

- 1x r6i.large nginx proxy
- 3x c5.large MM app nodes (HA)
- 2x db.x2g.xlarge RDS Aurora MySQL v8 (one writer, one reader)
- 1x (c7i.xlarge, c7i.2xlarge, c7i.4xlarge) RTCD
- 2x c7i.2xlarge load-test agents

App specifications

- Mattermost v9.6
- Mattermost Calls v0.28.0
- RTCD v0.16.0
- load-test agent v0.28.0

Media specifications

- Speech sample bitrate: 80Kbps
- Screen sharing sample bitrate: 1.6Mbps

Results

Calls	Participants/ call	Unmuted/ call	Screen sharing	CPU (avg)	Memory (avg)	Bandwidth (in/out)	Instance type (RTCD)
1	1000	2	no	47%	1.46GB	1Mbps / 194Mbps	c7i.xlarge
1	800	1	yes	64%	1.43GB	2.7Mbps / 1.36Gbps	c7i.xlarge
1	1000	1	yes	79%	1.54GB	2.9Mbps / 1.68Gbps	c7i.xlarge
10	100	1	yes	74%	1.56GB	18.2Mbps / 1.68Gbps	c7i.xlarge
100	10	2	no	49%	1.46GB	18.7Mbps / 175Mbps	c7i.xlarge
100	10	1	yes	84%	1.73GB	171Mbps / 1.53Gbps	c7i.xlarge
1	1000	2	no	20%	1.44GB	1.4Mbps / 194Mbps	c7i.2xlarge
1	1000	2	yes	49%	1.53GB	3.6Mbps / 1.79Gbps	c7i.2xlarge
2	1000	1	yes	73%	2.38GB	5.7Mbps / 3.06Gbps	c7i.2xlarge
100	10	2	yes	60%	1.74GB	181Mbps / 1.62Gbps	c7i.2xlarge
150	10	1	yes	72%	2.26GB	257Mbps / 2.30Gbps	c7i.2xlarge
150	10	2	yes	79%	2.34GB	271Mbps / 2.41Gbps	c7i.2xlarge
250	10	2	no	58%	2.66GB	47Mbps / 439Mbps	c7i.2xlarge
1000	2	2	no	78%	2.31GB	178Mbps / 195Mbps	c7i.2xlarge

Calls	Participants/ call	Unmuted/ call	Screen sharing	CPU (avg)	Memory (avg)	Bandwidth (in/out)	Instance type (RTCD)
2	1000	2	yes	41%	2.6GB	7.23Mbps / 3.60Gbps	c7i.4xlarge
3	1000	2	yes	63%	3.53GB	10.9Mbps / 5.38Gbps	c7i.4xlarge
4	1000	2	yes	83%	4.40GB	14.5Mbps / 7.17Gbps	c7i.4xlarge
250	10	2	yes	79%	3.49GB	431Mbps / 3.73Gbps	c7i.4xlarge
500	2	2	yes	71%	2.54GB	896Mbps / 919Mbps	c7i.4xlarge

Note

- The tests focused on a single, vertically scaled RTCD instance to understand the processing limits within a single node. Scaling the RTCD service horizontally should be sufficient to support a higher number of calls.
- RTCD processes were executed with all performance profiling enabled (including block and mutex). This resulted in some computational overhead.
- Both speech and screen samples have slightly higher bitrates than the average produced by a real client (e.g., a browser). This gives us some safety margin over real-world deployments.

Dedicated service

For Enterprise customers we offer a way to offload performance costs through a [dedicated service (https://github.com/mattermost/rtcd)](https://github.com/mattermost/rtcd) that can be used to further scale up calls.

Load testing

We provide a load-test tool (https://github.com/mattermost/mattermost-plugin-calls/tree/main/lt) that can be used to simulate and measure the performance impact of calls.

Monitoring

Both the plugin and the external `rtcd` service expose some Prometheus metrics to monitor performance. We provide an official dashboard (https://grafana.com/grafana/dashboards/23225-mattermost-calls-performance-monitoring/) that can be imported in Grafana. You can refer to Performance monitoring (https://docs.mattermost.com/scale/deploy-prometheus-grafana-for-performance-monitoring.html) for more information on how to set up Prometheus and visualize metrics through Grafana.

Calls plugin metrics

Metrics for the calls plugin are exposed through the `/plugins/com.mattermost.calls/metrics` subpath under the existing Mattermost server metrics endpoint. This is controlled by the [Listen address for performance (https://docs.mattermost.com/configure/environment-configuration-settings.html#listen-address-for-performance)](https://docs.mattermost.com/configure/environment-configuration-settings.html#listen-address-for-performance) configuration setting. It defaults to port `8067`.

> Note
>
> - The [Metrics plugin (https://docs.mattermost.com/scale/collect-performance-metrics.html)](https://docs.mattermost.com/scale/collect-performance-metrics.html) collects application-level metrics only and does not make system or OS-level calls. As a result, data typically derived from system-level metrics may be missing in the Grafana panel.
> - On Mattermost versions prior to v9.5, plugin metrics were exposed through the public `/plugins/com.mattermost.calls/metrics` API endpoint controlled by the [Web server listen address (https://docs.mattermost.com/configure/environment-configuration-settings.html#web-server-listen-address)](https://docs.mattermost.com/configure/environment-configuration-settings.html#web-server-listen-address) configuration setting. This defaults to port `8065`.

Process

- `mattermost_plugin_calls_process_cpu_seconds_total`: Total user and system CPU time spent in seconds.
- `mattermost_plugin_calls_process_max_fds`: Maximum number of open file descriptors.
- `mattermost_plugin_calls_process_open_fds`: Number of open file descriptors.
- `mattermost_plugin_calls_process_resident_memory_bytes`: Resident memory size in bytes.
- `mattermost_plugin_calls_process_virtual_memory_bytes`: Virtual memory size in bytes.

WebRTC connection

- `mattermost_plugin_calls_rtc_conn_states_total`: Total number of RTC connection state changes.
- `mattermost_plugin_calls_rtc_errors_total`: Total number of RTC errors.
- `mattermost_plugin_calls_rtc_rtp_bytes_total`: Total number of sent/received RTP packets in bytes.
  - Note: removed as of v0.16.0
- `mattermost_plugin_calls_rtc_rtp_packets_total`: Total number of sent/received RTP packets.
  - Note: removed as of v0.16.0
- `mattermost_plugin_calls_rtc_rtp_tracks_total`: Total number of incoming/outgoing RTP tracks.
  - Note: added as of v0.16.0
- `mattermost_plugin_calls_rtc_sessions_total`: Total number of active RTC sessions.

Application

- `mattermost_plugin_calls_app_handlers_time_bucket`: Time taken to execute app handlers.
  - `mattermost_plugin_calls_app_handlers_time_sum`
  - `mattermost_plugin_calls_app_handlers_time_count`

Database

- `mattermost_plugin_calls_store_ops_total`: Total number of db store operations.
- `mattermost_plugin_calls_store_methods_time_bucket`: Time taken to execute store methods.
  - `mattermost_plugin_calls_store_methods_time_sum`
  - `mattermost_plugin_calls_store_methods_time_count`

- `mattermost_plugin_calls_cluster_mutex_grab_time_bucket`: Time taken to grab global mutexes.
  - `mattermost_plugin_calls_cluster_mutex_grab_time_sum`
  - `mattermost_plugin_calls_cluster_mutex_grab_time_count`
- `mattermost_plugin_calls_cluster_mutex_locked_time_bucket`: Time spent locked in global mutexes.
  - `mattermost_plugin_calls_cluster_mutex_locked_time_sum`
  - `mattermost_plugin_calls_cluster_mutex_locked_time_count`

WebSocket

- `mattermost_plugin_calls_websocket_connections_total`: Total number of active WebSocket connections.
- `mattermost_plugin_calls_websocket_events_total`: Total number of WebSocket events.

Jobs

- `mattermost_plugin_calls_jobs_live_captions_new_audio_len_ms_bucket`: Duration (in ms) of new audio transcribed for live captions.
  - `mattermost_plugin_calls_jobs_live_captions_new_audio_len_ms_sum`
  - `mattermost_plugin_calls_jobs_live_captions_new_audio_len_ms_count`
- `mattermost_plugin_calls_jobs_live_captions_pktPayloadCh_buf_full`: Total packets of audio data dropped due to full channel.
- `mattermost_plugin_calls_jobs_live_captions_window_dropped`: Total windows of audio data dropped due to pressure on the transcriber.

WebRTC service metrics

Metrics for the `rtcd` service are exposed through the `/metrics` API endpoint under the `rtcd` API listener controlled by the `api.http.listen_address` configuration setting. It defaults to port `8045`.

Process

- `rtcd_process_cpu_seconds_total`: Total user and system CPU time spent in seconds.
- `rtcd_plugin_calls_process_max_fds`: Maximum number of open file descriptors.
- `rtcd_plugin_calls_process_open_fds`: Number of open file descriptors.
- `rtcd_plugin_calls_process_resident_memory_bytes`: Resident memory size in bytes.
- `rtcd_plugin_calls_process_virtual_memory_bytes`: Virtual memory size in bytes.

WebRTC Connection

- `rtcd_rtc_conn_states_total`: Total number of RTC connection state changes.
- `rtcd_rtc_errors_total`: Total number of RTC errors.
- `rtcd_rtc_rtp_bytes_total`: Total number of sent/received RTP packets in bytes.
- `rtcd_rtc_rtp_packets_total`: Total number of sent/received RTP packets.
- `rtcd_rtc_rtp_tracks_total`: Total number of incoming/outgoing RTP tracks.
- `rtcd_rtc_sessions_total`: Total number of active RTC sessions.
- `rtcd_rtc_rtp_tracks_writes_time_bucket`: Time taken to write to outgoing RTP tracks.
  - `rtcd_rtc_rtp_tracks_writes_time_sum`
  - `rtcd_rtc_rtp_tracks_writes_time_count`

WebSocket

- `rtcd_ws_connections_total`: Total number of active WebSocket connections.
- `rtcd_ws_messages_total`: Total number of received/sent WebSocket messages.

Configuration

A sample Prometheus configuration to scrape both plugin and `rtcd` metrics could look like this:

```
scrape_configs: - job_name: node static_configs: - targets:
['rtcd-0:9100','rtcd-1:9100', 'calls-offloader-1:9100', 'calls-
offloader-2:9100'] - job_name: calls metrics_path: /plugins/
com.mattermost.calls/metrics static_configs: - targets:
['app-0:8067','app-1:8067','app-2:8067'] - job_name: rtcd static_configs: -
targets: ['rtcd-0:8045', 'rtcd-1:8045']
```

System tunings

If you want to host many calls or calls with a large number of participants, take a look at the following platform specific (Linux) tunings (this is the only officially supported target for the plugin right now):

```
# Setting the maximum buffer size of the receiving UDP buffer to 16MB
net.core.rmem_max = 16777216 # Setting the maximum buffer size of the
sending UDP buffer to 16MB net.core.wmem_max = 16777216 # Allow to allocate
more memory as needed for more control messages that need to be sent for
each socket connected net.core.optmem_max = 16777216
```

The rtcd service

Note

The rtcd service is available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

The Calls plugin has a built-in [Selective Forwarding Unit (SFU) (https://bloggeek.me/webrtcglossary/sfu/)](https://bloggeek.me/webrtcglossary/sfu/) to route audio and screensharing data. This is the `integrated` option described in the Modes of operation section above. But this SFU functionality can be deployed separately as an external `rtcd` instance.

Reasons to use the `rtcd` service

This section will help you understand when and why your organization would want to use `rtcd`.

> Note
>
> `rtcd` is a standalone service, which adds operational complexity, maintenance costs, and requires an enterprise licence. For those who are evaluating Calls, and for many small instances of Mattermost, the integrated SFU (the one included in the Calls plugin) may be sufficient initially.

The `rtcd` service is the recommended way to host Calls for the following reasons:

- **Performance of the main Mattermost server(s).** When the Calls plugin runs the SFU, calls traffic is added to the processing load of the server running the rest of your Mattermost services. If Calls traffic spikes, it can negatively affect the responsiveness of these services. Using an rtcd service isolates the calls traffic processing to those rtcd instances, and also reduces costs by minimizing CPU usage spikes.
- **Performance, scalability, and stability of the Calls product.** If Calls traffic spikes, or more overall capacity is needed, `rtcd` servers can be added to balance the load. As an added benefit, if the Mattermost traffic spikes, or if a Mattermost instance needs to be restarted, those people in a current call will not be affected - current calls won't be dropped.

Some caveats apply here. Web socket events (for example: emoji reactions, hand raising, muting/unmuting) will not be transmitted while the main Mattermost server is down. But the call itself will continue while the main server restarts.

- **Kubernetes deployments.** In a Kubernetes deployment, `rtcd` is strongly recommended; it is currently the only officially supported way to run Calls.
- **Technical benefits.** The dedicated `rtcd` service has been optimized and tuned at the system/network level for real-time audio/video traffic, where latency is generally more important than throughput.

In general, `rtcd` is the preferred solution for a performant and scalable deployment. With `rtcd`, the Mattermost server will be minimally impacted when hosting a high number of calls.

See the Mattermost rtcd repository documentation on GitHub for details on how to run calls through the service (https://github.com/mattermost/rtcd/blob/master/docs/getting_started.md), as well as:

- Key implementation details (https://github.com/mattermost/rtcd/blob/master/docs/implementation.md)
- Project structure (https://github.com/mattermost/rtcd/blob/master/docs/project_structure.md)
- Configuration overrides (https://github.com/mattermost/rtcd/blob/master/docs/env_config.md)
- Authentication flow (https://github.com/mattermost/rtcd/blob/master/docs/security.md)

Horizontal scalability

The supported way to enable horizontal scalability for Calls is through a form of DNS based load balancing. This can be achieved regardless of how the `rtcd` service is deployed (bare bone instance, Kubernetes, or an alternate way).

In order for this to work, the RTCD Service URL (https://docs.mattermost.com/configure/plugins-configuration-settings.html#rtcd-service-url) should point to a hostname that resolves to multiple IP addresses, each pointing to a running `rtcd` instance. The Mattermost Calls plugin will then automatically distribute calls amongst the available hosts.

The expected requirements are the following:

- When a new `rtcd` instance is deployed, it should be added to the DNS record. The plugin side will then be able to pick it up and start assigning calls to the new host.
- If a `rtcd` instance goes down, it should be removed from the DNS record. The plugin side can then detect the change and stop assigning new calls to that host.

> Note
>
> - Load balancing is done at the call level. This means that a single call will always live on a single `rtcd` instance.
> - There's currently no support for spreading sessions belonging to the same call across a fleet of instances.

Configure recording, transcriptions, and live captions

Before you can start recording, transcribing, and live captioning calls, you need to configure the `calls-offloader` job service. See the calls-offloader (https://github.com/mattermost/calls-offloader/blob/master/docs/getting_started.md) documentation on GitHub for details on deploying and running this service. Performance and scalability recommendations (https://github.com/mattermost/calls-offloader/blob/master/docs/performance.md) related to this service are also available on GitHub.

> Note
>
> If deploying the service in a Kubernetes cluster, refer to the later section on Helm charts.

Once the `calls-offloader` service is running, recordings should be explicitly enabled through the Enable call recordings (https://docs.mattermost.com/configure/plugins-configuration-settings.html#enable-call-recordings) config setting and the service's URL should be configured using Job service URL (https://docs.mattermost.com/configure/plugins-configuration-settings.html#job-service-url).

Call transcriptions can be enabled through the Enable call transcriptions (https://docs.mattermost.com/configure/plugins-configuration-settings.html#enable-call-transcriptions) configuration setting.

Live captions can be enabled through the [Enable live captions (https://docs.mattermost.com/configure/plugins-configuration-settings.html#enable-live-captions)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#enable-live-captions) configuration setting.

> Note
>
> - The call transcriptions functionality is available starting in Calls version v0.22.0.
> - The live captions functionality is available starting in Calls version v0.26.2.

Kubernetes deployments

The Calls plugin has been designed to integrate well with Kubernetes to offer improved scalability and control over the deployment.

This is a sample diagram showing how the `rtcd` standalone service can be deployed in a Kubernetes cluster:

If Mattermost isn't deployed in a Kubernetes cluster, and you want to use this deployment type, see the [Deploy Mattermost on Kubernetes (https://docs.mattermost.com/install/install-kubernetes.html)](https://docs.mattermost.com/install/install-kubernetes.html) documentation.

Helm Charts

The recommended way to deploy Calls related components and services in a Kubernetes deployment is to use the officially provided Helm charts. Related

documentation including detailed information on how to deploy these services can be found in our `mattermost-helm` repository:

- [rtcd Helm chart (https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-rtcd)](https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-rtcd)
- [calls-offloader Helm chart (https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-calls-offloader)](https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-calls-offloader)

Limitations

Due to the inherent complexities of hosting a WebRTC service, some limitations apply when deploying Calls in a Kubernetes environment.

One key requirement is that each `rtcd` process live in a dedicated Kubernetes node. This is necessary to forward the data correctly while allowing for horizontal scaling. Data should generally not go through a standard ingress but directly to the pod running the `rtcd` process.

The general recommendation is to expose one external IP address per `rtcd` instance (Kubernetes node). This makes it simpler to scale as the application is able to detect its own external address (through STUN) and advertise it to clients to achieve connectivity with minimal configuration.

If, for some reason, exposing multiple IP addresses is not possible in your environment, port mapping (NAT) can be used. In this scenario different ports are used to map the respective `rtcd` nodes behind the single external IP. Example:

```
EXT_IP:8443 -> rtcdA:8443 EXT_IP:8444 -> rtcdB:8443 EXT_IP:8445 ->
rtcdC:8443
```

This case requires a couple of extra configurations:

- NAT mappings need to be in place for every `rtcd` node. This is usually done at the ingress point (e.g., ELB, NLB, etc).
- The `RTCD_RTC_ICEHOSTPORTOVERRIDE` config should be used to pass a full mapping of node IPs and their respective port.
  - Example: `RTCD_RTC_ICEHOSTPORTOVERRIDE=rtcdA_IP/8443,rtcdB_IP/8444,rtcdC_IP/8445`

- The `RTCD_RTC_ICEHOSTOVERRIDE` should be used to set the external IP address.

> Note
>
> One option to limit these static mappings is to reduce the size of the local subnet (e.g., to `/29`).

Frequently asked questions

Is there encryption?

Media (audio/video) is encrypted using security standards as part of WebRTC. It's mainly a combination of DTLS and SRTP. It's not e2e encrypted in the sense that in the current design all media needs to go through Mattermost which acts as a media router and has complete access to it. Media is then encrypted back to the clients so it's secured during transit. In short: only the participant clients and the Mattermost server have access to unencrypted call data.

Are there any third-party services involved?

The only external service used is a Mattermost official STUN server (`stun.global.calls.mattermost.com`) which is configured as default. This is primarily used to find the public address of the Mattermost instance if none is provided through the [ICE Host Override (https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override) option. The only information sent to this service is the IP addresses of clients connecting as no other traffic goes through it. It can be removed in cases where the [ICE Host Override (https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override) setting is provided.

> Note
>
> In air-gapped deployments, using STUN servers is not necessary since all connections remain within the local network.

Is using UDP a requirement?

Yes, UDP is the recommended protocol to serve real-time media as it allows for the lowest latency between peers. However, there are a couple of possible solutions to cover clients that due to limitations or strict firewalls are unable to use UDP:

- Since plugin version 0.17 and `rtcd` version 0.11 the RTC service will listen for TCP connections in addition to UDP ones. If configured correctly (e.g. using commonly allowed ports such as 80 or 443) it's possible to have clients connect directly through TCP when unable to do it through the preferred UDP channel.
- Run calls through an external TURN server that listens on TCP and relays all media traffic between peers. However, this is a sub-optimal solution that should be avoided if possible as it will introduce extra latency along with added infrastructural cost.

Do I need a TURN server?

TURN becomes necessary when you expect to have clients that are unable to connect through the configured UDP port. This can happen due to very restrictive firewalls that either block non standard ports even in the outgoing direction or don't allow the use of the UDP protocol altogether (e.g. some corporate firewalls). In such cases TURN is needed to allow connectivity.

We officially support and recommend using [coturn (https://github.com/coturn/coturn)](https://github.com/coturn/coturn) for a stable and performant TURN service implementation.

How will this work with an existing reverse proxy sitting in front of Mattermost?

Generally clients should connect directly to either Mattermost or, if deployed, the dedicated `rtcd` service through the configured UDP port . However, it's also possible to route the traffic through an existing load balancer as long as this has support for routing the UDP protocol (e.g. nginx). Of course this will require additional configuration and potential changes to how the plugin is run as it won't be possible to load balance the UDP flow across multiple instances like it happens for HTTP.

Do calls require a dedicated server to work or can they run alongside Mattermost?

The plugin can function in different modes. By default calls are handled completely by the plugin which runs as part of Mattermost. It's also possible to use a dedicated service to offload the computational and bandwidth costs and scale further (Enterprise only).

Can the traffic between Mattermost and `rtcd` be kept internal or should it be opened to the public?

When possible, it's recommended to keep communication between the Mattermost cluster and the dedicated `rtcd` service under the same private network as this can greatly simplify deployment and security. There's no requirement to expose `rtcd`'s HTTP API to the public internet.

Can Calls be rolled out on a per-channel basis?

Note

Available only for self-hosted deployments

Yes. Mattermost system admins running self-hosted deployments can enable or disable call functionality per channel. Once [test mode (https://docs.mattermost.com/configure/plugins-configuration-settings.html#test-mode)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#test-mode) is enabled for Mattermost Calls:

- Select **Enable calls** for each channel where you want Calls enabled
- Select **Disable calls** for all channels where you want Calls disabled.

Once Calls is enabled for specific channels, users can start making calls in those channels.

> Note
>
> When test mode (https://docs.mattermost.com/configure/plugins-configuration-settings.html#test-mode) is disabled for Mattermost Calls, users in any Mattermost channel can make a call.

Troubleshooting

Connectivity issues

If calls are failing to connect or timing out, it's likely there could be a misconfiguration at either the plugin config or networking level.

For example, the RTC Server Port (UDP) (https://docs.mattermost.com/configure/plugins-configuration-settings.html#rtc-server-port-udp) or the RTC Server Port (TCP) (https://docs.mattermost.com/configure/plugins-configuration-settings.html#rtc-server-port-tcp) may not be open or forwarded correctly.

Connectivity checks

An easy way to check whether data can go through is to perform some tests using the `netcat` command line tool.

On the host running Calls (could be the Mattermost instance itself or the one running `rtcd` depending on the chosen setup), run the following:

```
nc -l -u -p 8443
```

On the client side (i.e., the machine you would normally use to run the Mattermost desktop app or browser), run the following:

```
nc -v -u HOST_IP 8443
```

If connection succeeds, you should be able to send and receive text messages by typing and hitting enter on either side.

> Note
>
> `HOST_IP` should generally be the public (client facing) IP of the Mattermost (or `rtcd`) instance hosting the calls. When set, it should be the value of the [ICE Host Override (https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#ice-host-override) config setting.
>
> `8443` should be changed with the port configured in [RTC Server Port (https://docs.mattermost.com/configure/plugins-configuration-settings.html#rtc-server-port-udp)](https://docs.mattermost.com/configure/plugins-configuration-settings.html#rtc-server-port-udp).
>
> The same checks can be performed to test connectivity through the TCP port using the same commands with `-u` flag removed.

Network packets debugging

A more advanced way to debug networking issues is to use the `tcpdump` command line utility to temporaily monitor network packets flowing in and out of the instance hosting calls.

On the server side, run the following:

```
sudo tcpdump -n port 8443
```

This command will output information (i.e. source and destination addresses) for all the network packets being sent or received through port `8443`. This is a good way to check whether data is getting in and out of the instance and can be used to quickly identify network configuration issues.

On this page

- Calls self-hosted deployment
  - Terminology
  - Plugin components
  - Requirements
    - Server
    - Client

    - Network
        - Air-gapped deployments
- Limitations
- Configuration
- Modes of operation
    - Single instance
        - Integrated
        - rtcd
    - High availability cluster-based
        - Clustered
        - rtcd (High Availability)
- Performance
    - Benchmarks
        - Deployment specifications
        - App specifications
        - Media specifications
        - Results
    - Dedicated service
    - Load testing
    - Monitoring
        - Calls plugin metrics
        - WebRTC service metrics
        - Configuration
    - System tunings
- The rtcd service
    - Reasons to use the rtcd service
    - Horizontal scalability
- Configure recording, transcriptions, and live captions
- Kubernetes deployments
    - Helm Charts
    - Limitations
- Frequently asked questions
    - Is there encryption?

        - Are there any third-party services involved?
        - Is using UDP a requirement?
        - Do I need a TURN server?
        - How will this work with an existing reverse proxy sitting in front of Mattermost?
        - Do calls require a dedicated server to work or can they run alongside Mattermost?
        - Can the traffic between Mattermost and rtcd be kept internal or should it be opened to the public?
        - Can Calls be rolled out on a per-channel basis?
    - Troubleshooting
        - Connectivity issues
            - Connectivity checks
            - Network packets debugging

- Mattermost documentation

Note

The rtcd service is available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Cloud workspace subscription, billing, and account settings

Review and manage the following aspects of your Mattermost cloud-based deployment by selecting the **Product**

menu, selecting **System Console**, and then selecting **Billing and Account**:

- Access billing history
- Manage the product subscription and account details for your Mattermost Cloud deployment.
- Review trial details.
- Talk to a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) for assistance.

Compliance configuration settings

Review and manage the following compliance configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Compliance**:

- Data Retention Policies
- Compliance Export
- Compliance Monitoring
- Custom Terms of Service

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `MessageRetentionHours` value is under `DataRetentionSettings`.
>
> - If using a tool such as jq (https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.DataRetentionSettings.MessageRetentionHours'`
> - When working with the `config.json` file manually, look for an object such as `DataRetentionSettings`, then within that object, find the key `MessageRetentionHours`.

Data retention policies

Changes to properties in this section require a server restart before taking effect.

> Warning
>
> - Once a message or a file is deleted, the action is irreversible. Please be careful when setting up a custom data retention policy.
> - From Mattermost v9.5, data retention removes Elasticsearch indexes based on the day of the retention cut-off time.

Access the following configuration settings in the System Console by going to **Compliance > Data Retention Policies**.

Global retention policy for messages

Set how long Mattermost keeps messages across all teams and channels. Doesn't apply to custom retention policies. Requires the global retention policy for messages configuration setting to be set to `true`.

By default, messages are kept forever. If **Hours**, **Days**, or **Years** is chosen, set how many hours, days, or years messages are kept in Mattermost. Messages older than the duration you set will be deleted nightly. The minimum message retention time is one hour.

This feature's `config.json` setting is `"MessageRetentionHours": 1` with numerical input.

> Note
>
> From Mattermost v9.5, `MessageRetentionDays` has been deprecated in favor of `MessageRetentionHours`. See deprecated configuration settings for details.

Global retention policy for files

Set how long Mattermost keeps files across all teams and channels. Doesn't apply to custom retention policies. Requires the global retention policy for files configuration setting to be set to `true`.

By default, files are kept forever. If **Hours**, **Days**, or **Years** is chosen, set how many hours, days, or years files are kept in Mattermost. Files older than the duration you set will be deleted nightly. The minimum file retention time is one hour.

This feature's `config.json` setting is `"FileRetentionHours": 1` with numerical input.

> Note
>
> From Mattermost v9.5, `FileRetentionDays` has been deprecated in favor of `FileRetentionHours`. See deprecated configuration settings for details.

Preserve pinned posts

From Mattermost v10.10, controls whether pinned posts are preserved when data retention policies delete messages. When enabled, pinned posts won't be deleted by data retention policies, even if they exceed the configured retention period.

True: Pinned posts are preserved and won't be deleted by data retention policies.

False: **(Default)** Pinned posts are deleted according to the configured data retention policy.

This feature's `config.json` setting is `"DataRetentionSettings.PreservePinnedPosts": false` with options `true` and `false`.

> Note
>
> - This global configuration setting must be enabled with mmctl using the mmctl config set command.
> - This configuration setting applies to team and channel policies as well as data retention, and can't be overridden in those more granular team or channel policies.
> - Files attached to the pinned message aren't preserved.
> - Only the pinned post is preserved. If it's attached to a thread or if it's the root post of a thread, the other threaded messages aren't preserved.

Custom retention policy

Set how long Mattermost keeps messages and files across specific teams and channels by specifying a name for the custom retention policy, setting a duration value in days or years, and specifying the teams and channels that will follow this policy.

Data deletion time

Set the start time of the daily scheduled data retention job. Choose a time when fewer people are using your system. Must be a 24-hour time stamp in the form `HH:MM`.

This setting is based on the local time of the server.

> This feature's `config.json` setting is `"DeletionJobStartTime": "02:00"` with 24-hour timestamp input in the form `"HH:MM"`.

Run deletion job now

Start a Data Retention deletion job immediately. You can monitor the status of the job in the data deletion job table within the Policy Log section.

Compliance export

Access the following configuration settings in the System Console by going to **Compliance > Compliance Export**.

Enable compliance export

True: Mattermost will generate a compliance export file that contains all messages that were posted in the last 24 hours. The export task is scheduled to run once per day. See the documentation to learn more.

False: Mattermost doesn't generate a compliance export file.

> This feature's `config.json` setting is `"EnableExport": false` with options `true` and `false`.

Compliance export time

Set the start time of the daily scheduled compliance export job. Choose a time when fewer people are using your system. Must be a 24-hour time stamp in the form `HH:MM`.

This setting is based on the local time of the server.

> This feature's `config.json` setting is `"DailyRunTime": 01:00` with 24-hour timestamp input in the form `"HH:MM"`.

Export file format

File format of the compliance export. Corresponds to the system that you want to import the data into.

Currently supported formats are CSV, Actiance XML, and Global Relay EML.

If Global Relay is chosen, the following options will be presented:

Global Relay customer account

Type of Global Relay customer account your organization has. Can be one of: `A9/Type 9`, `A10/Type 10`, or `Custom`.

> This feature's `config.json` setting is `"CustomerType": "A9"` with options `"A9"`, `"A10"`, and `CUSTOM`.

Global Relay SMTP username

The username for authenticating to the Global Relay SMTP server.

> This feature's `config.json` setting is `"SmtpUsername": ""` with string input.

Global Relay SMTP password

The password associated with the Global Relay SMTP username.

> This feature's `config.json` setting is `"SmtpPassword": ""` with string input.

Global Relay email address

The email address your Global Relay server monitors for incoming compliance exports.

This feature's `config.json` setting is `"EmailAddress": ""` with string input.

SMTP server name

The SMTP server name URL that will receive your Global Relay EML file when a custom customer account type is configured.

This feature's `config.json` setting is `".MessageExportSettings.GlobalRelaySettings.CustomSMTPServerName": ""` with string input.

SMTP server port

The SMTP server port that will receive your Global Relay EML file when a custom customer account type is configured. Default is 25.

This feature's `config.json` setting is `".MessageExportSettings.GlobalRelaySettings.CustomSMTPPort": 25` with string input.

Message export batch size

Note

This setting isn't available in the System Console and can only be set in `config.json`.

Determines how many new posts are batched together to a compliance export file.

This feature's `config.json` setting is `"BatchSize": 10000` with numerical input.

Run compliance export job now

This button initiates a compliance export job immediately. You can monitor the status of the job in the compliance export job table.

Compliance monitoring

Cloudandself-hosteddeployments

Settings used to enable and configure Mattermost compliance reports.

Access the following configuration settings in the System Console by going to **Compliance > Compliance Monitoring**.

Enable compliance reporting

True: Compliance reporting is enabled in Mattermost.

False: Compliance reporting is disabled.

> This feature's `config.json` setting is `"Enable": false` with options `true` and `false`.

Compliance report directory

Sets the directory where compliance reports are written.

> This feature's `config.json` setting is `"Directory": "./data/"` with string input.

Enable daily report

True: Mattermost generates a daily compliance report.

False: Daily reports are not generated.

This feature's `config.json` setting is `"EnableDaily": false` with options `true` and `false`.

Batch size

Set the size of the batches in which posts will be read from the database to generate the compliance report. This setting is currently not available in the System Console and can only be set in `config.json`.

This feature's `config.json` setting is `"BatchSize": 30000` with default value `30000`.

Custom terms of service

Available onEnterprise plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Compliance > Custom Terms of Service**.

Enable custom terms of service

> Note
>
> This configuration setting can only be modified using the System Console user interface.

True: New users must accept the Terms of Service before accessing any Mattermost teams on desktop, web, or mobile. Existing users must accept them after login or a page refresh. To update the Terms of Service link displayed in account creation and login pages, go to **System Console > Legal and Support > Terms of Service Link**.

False: During account creation or login, users can review Terms of Service by accessing the link configured via **System Console > Legal and Support > Terms of Service link**.

Custom terms of service text

Text that will appear in your custom Terms of Service. Supports Markdown-formatted text.

Re-acceptance period

The number of days before Terms of Service acceptance expires, and the terms must be re-accepted.

Defaults to 365 days. 0 indicates the terms do not expire.

On this page

- Compliance configuration settings
  - Data retention policies
    - Global retention policy for messages
    - Global retention policy for files
    - Preserve pinned posts
    - Custom retention policy
    - Data deletion time
    - Run deletion job now
  - Compliance export
    - Enable compliance export
    - Compliance export time
    - Export file format
    - Global Relay customer account
    - Global Relay SMTP username
    - Global Relay SMTP password
    - Global Relay email address
    - SMTP server name
    - SMTP server port
    - Message export batch size

    - Run compliance export job now
- Compliance monitoring
    - Enable compliance reporting
    - Compliance report directory
    - Enable daily report
    - Batch size
- Custom terms of service
    - Enable custom terms of service
    - Custom terms of service text
    - Re-acceptance period

Configuring Apache2 as a proxy for Mattermost Server (Unofficial)

Important

This unofficial guide is maintained by the Mattermost community and this deployment configuration is not yet officially supported by Mattermost, Inc. Community testing, feedback and improvements are welcome and greatly appreciated (https://github.com/mattermost/docs/issues/1295). You can edit this page on GitHub (https://github.com/mattermost/docs/blob/master/source/administration-guide/configure/config-proxy-apache2.rst).

On a Debian-based operating system such as Ubuntu, Apache2 proxy configuration is done in the `/etc/apache2/sites-available` directory. Red Hat-based systems organize Apache configuration files differently (https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-web_servers). If you're setting up Mattermost on a subdomain, you'll want to create a new configuration file along the lines of `mysubdomain.mydomain.com.conf`.

To configure Apache2 as a proxy

1. SSH into your server.
2. Make sure the Apache modules `mod_rewrite` , `mod_proxy`, `mod_proxy_http`, and `mod_proxy_wstunnel` are installed and enabled. If not, follow the instructions from your Linux distribution to do so.
3. Create the above mentioned configuration file. It is often helpful to start with a copy of `000-default.conf` or `default-ssl.conf` (on Ubuntu).
4. Edit your configuration using the guide below:

> 1. *If you're not setting up a subdomain, your* `ServerName` *will simply be set to* `mydomain.com`*.*
> 2. `ServerAlias` *can been added too if you want to capture* `www.mydomain.com`*.*

> 3. *Remember to change the values to match your server's name, etc.*
> 4. *If you have enabled TLS in the Mattermost settings, you must use the protocol* `wss://` *instead of* `ws://` *in the* `RewriteRule`*.*
> 5. *To serve requests on a different port (such as 8443), in addition to setting the port in the VirtualHost element, add* `Listen 8443` *on a separate line before the VirtualHost line.*

```
<VirtualHost *:80>  # If you're not using a subdomain you may need to set a
ServerAlias to:  # ServerAlias www.mydomain.com  ServerName
mysubdomain.mydomain.com  ServerAdmin hostmaster@mydomain.com
ProxyPreserveHost On  # Set web sockets  RewriteEngine On  RewriteCond %
{REQUEST_URI} /api/v[0-9]+/(users/)?websocket [NC]  RewriteCond %
{HTTP:UPGRADE} ^WebSocket$ [NC,OR]  RewriteCond %{HTTP:CONNECTION}
^Upgrade$ [NC]  RewriteRule .* ws://127.0.0.1:8065%{REQUEST_URI} [P,QSA,L]
<Location />  Require all granted  ProxyPass http://127.0.0.1:8065/
ProxyPassReverse http://127.0.0.1:8065/  ProxyPassReverseCookieDomain
127.0.0.1 mysubdomain.mydomain.com  </Location> </VirtualHost>
```

5. (Debian/Ubuntu only) Because you'll likely have not set up the subdomain before now on Apache2, run `a2ensite mysubdomain.mydomain.com` to enable the site (do not run `a2ensite mysubdomain.mydomain.com.conf`).
6. Restart Apache2.

> - *On Ubuntu 14.04 and RHEL 6:* `sudo service apache2 restart`
> - *On Ubuntu 16.04+ and RHEL 7+:* `sudo systemctl restart apache2`

You should be all set! Ensure that your Mattermost config file is pointing to the correct URL (which may include a port), and then ensure that your socket connection is not dropping once deployed. To prevent external access to Mattermost on port 8065, in the config file, set `ListenAddress` to `localhost:8065` instead of `:8065`.

Store configuration in your database

You can use your database as the single source of truth for the active configuration of your Mattermost installation. This changes the Mattermost binary from reading the default `config.json` file to reading the configuration settings stored within a configuration table in the database. Mattermost has been running our community server (https://community.mattermost.com) on this option since the feature was released, and recommends its use for those on High Availability deployments.

Benefits to using this option:

- Conveniently manages configuration changes directly from the System Console, even in High Availability deployments and read-only containerized environments.
- Ensures all servers in a High Availability deployment have the same configuration, even when new servers are added to the cluster.
- Automatically deploys SAML certificates and keys to all servers in the cluster.

> Tip
>
> The Mattermost configuration database and Mattermost application database are 2 different entities. It's possible to store Mattermost configuration in one database and Mattermost data in another database.
>
> To do so, you must update the Datasource configuration setting to a new data source name, which can be done while the application is running. Explicitly setting the `MM_SQLSETTINGS_DATASOURCE` environment variable to override what has been defined in the configuration, whether it's in a database, or in a file, allows the correct data source name to be passed to the Mattermost application.

How to migrate configuration to the database

These instructions cover migrating the Mattermost configuration to your database and updating your `systemd` configuration to load it from the database.

Important

- These instructions assume you have Mattermost server installed at `/opt/mattermost`. If you're running Mattermost in a different directory you'll have to modify the paths to match your environment.
- If you're running Mattermost in a High Availability cluster-based deployment, you must complete all of the steps below on each server in the cluster.

Get your database connection string

The first step is to get your master database connection string. We recommend accessing your `config.json` file to make a copy of the value in `SqlSettings.DataSource`, or your equivalent environment variable, `MM_SQLSETTINGS_DATASOURCE`.

Important

- `SqlSettings.DataSource` must start with `postgres://` or `mysql://`. If it doesn't, add it to the beginning based on the database in use. For example: `postgres://mmuser:really_secure_password@localhost:5432/mattermost?sslmode=disable&connect_timeout=10`
- If you see `\u0026`, replace it with `&`. For example: `mysql://mmuser:really_secure_password@tcp(127.0.0.1:3306)/mattermost?charset=utf8mb4,utf8&writeTimeout=30s`

Create an environment file

Create the file `/opt/mattermost/config/mattermost.environment` to set the `MM_CONFIG` environment variable to the database connection string. For example:

```
MM_CONFIG='postgres://mmuser:mostest@localhost:5432/mattermost_test?
sslmode=disable&connect_timeout=10'
```

Run this command to verify the permissions on your Mattermost directory:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

Enable local mode

Edit the `config.json` to enable local mode by setting `EnableLocalMode` to `true`. See the local mode documentation for details on activating and using local mode.

Restart Mattermost

Run the following command to restart the Mattermost server and apply the configuration change:

```
sudo systemctl restart mattermost
```

Migrate configuration from `config.json`

You can use the mmctl config migrate command to migrate the configuration by running the following command:

```
./bin/mmctl config migrate path/to/config.json "postgres://
mmuser:mostest@localhost:5432/mattermost_test?
sslmode=disable&connect_timeout=10" --local
```

Important

- If you're using a High Availability cluster-based deployment, you only need to run this command once and migrate the configuration from one server in the cluster.
- When migrating configuration, Mattermost incorporates configuration from any existing `MM_*` environment variables set in the current shell. See Environment Variables documentation for details.

- As with the environment file, you'll have to escape any single quotes in the database connection string.
- Any existing SAML certificates will be migrated into the database as well so they are available for all servers in the cluster. When the certificates expire, you can upload new certificates using the System Console or mmctl, which triggers a database update. Replacing the certificate files manually requires a reload of the Mattermost server to re-pull the certificates. Configuration files are stored in the `configurationfiles` table in the database.

When configuration in the database is enabled, any changes to the configuration are recorded to the `Configurations` and `ConfigurationFiles` tables. Furthermore, `ClusterSettings.ReadOnlyConfig` is ignored, enabling full use of the System Console.

If you have configuration settings that must be set on a per-server basis you should add them as environment variables to the `mattermost.environment` file. These must be on their own line, and you must escape them properly.

Modify the Mattermost `systemd` file

Find the `mattermost.service` file using the following command:

```
sudo systemctl status mattermost.service
```

The second line of output will have the location of the running `mattermost.service`.

```
Loaded: loaded (/lib/systemd/system/mattermost.service; enabled; vendor
preset: enabled)
```

Edit this file as *root* to add the below text just above the line that begins with `ExecStart`:

```
EnvironmentFile=/opt/mattermost/config/mattermost.environment
```

Here's a complete `mattermost.service` file with the `EnvironmentFile` line added:

```
[Unit] Description=Mattermost After=network.target After=postgresql.service
Requires=postgresql.service [Service] Type=notify EnvironmentFile=/opt/
mattermost/config/mattermost.environment ExecStart=/opt/mattermost/bin/
mattermost TimeoutStartSec=3600 KillMode=mixed Restart=always RestartSec=10
WorkingDirectory=/opt/mattermost User=mattermost Group=mattermost
LimitNOFILE=49152 [Install] WantedBy=postgresql.service
```

Verify that the configuration was migrated correctly

Configurations are stored in the `Configurations` table in the database. Run the following query to verify that you've migrated the configuration successfully:

```
SELECT * FROM Configurations WHERE Active=true;
```

There should be exactly one line returned, and the `Value` field for that line should match your `config.json` file.

Reload `systemd` files and restart Mattermost

Run these commands to reload the daemon and restart Mattermost using the new `MM_CONFIG` environment variable.

```
sudo systemctl daemon-reload sudo systemctl restart mattermost
```

Important

Once you start using configuration in the database, you shouldn't manually edit the active configuration row. You should edit or update the configuration in one of the following ways:

- Use the System Console to make changes to the configuration.
- Use `mmctl` to make changes to the configuration.

The Mattermost server keeps active configuration in memory and writes new ones to the database only when there is a change. This way we avoid polling the database to process changes to the configuration. Publishing the changes to the cluster are handled by the application itself.

Rolling back

If you run into issues with your configuration in the database you can roll back to the `config.json` file by commenting out the `MM_CONFIG` line in `/opt/mattermost/config/mattermost.environment` and restarting Mattermost with `systemctl restart mattermost`.

Troubleshooting

Server fails to start

Providing the `--disableconfigwatch` flag while not actually pointing at a file will fail to start the server with an appropriate error message.

On this page

- Store configuration in your database
  - How to migrate configuration to the database
    - Get your database connection string
    - Create an environment file
    - Enable local mode
    - Restart Mattermost
    - Migrate configuration from config.json
    - Modify the Mattermost systemd file
      - Verify that the configuration was migrated correctly
    - Reload systemd files and restart Mattermost
  - Rolling back
  - Troubleshooting
    - Server fails to start

Configuration settings

System admins for both self-hosted and Cloud Mattermost deployments can manage Mattermost configuration using the System Console by selecting the **Product**

menu and selecting **System Console**.

> Note
>
> - In self-hosted Mattermost deployments, configuration settings are maintained in the `config.json` configuration file, located in the `mattermost/config` directory, or stored in the database. System admins managing self-hosted deployments can also modify the `config.json` file directly using a text editor.
> -

Mattermost configuration settings are organized into the following categories within the System Console:

- Self-hosted workspace edition and license settings
- Cloud workspace subscription, billing, and account settings
- Reporting configuration settings
- User management configuration settings
- System attributes
- Environment configuration settings
- Site configuration settings
- Authentication configuration settings
- Plugins configuration settings

- Integrations configuration settings
- Compliance configuration settings
- Experimental configuration settings
- Deprecated configuration settings
- Bleve search

Configuration in database

Note

Available only for self-hosted deployments

Self-hosted system configuration can be stored in the database. This changes the Mattermost binary from reading the default `config.json` file to reading the configuration settings stored within a configuration table in the database. See the Mattermost database configuration documentation for migration details.

Environment variables

Note

Available only for self-hosted deployments

You can use environment variables to manage Mattermost configuration. Environment variables override settings in `config.json`. If a change to a setting in `config.json` requires a restart to take effect, then changes to the corresponding environment variable also require a server restart.

Configuration reload

Note

Available only for self-hosted deployments

The "config watcher", the mechanism that automatically reloads the `config.json` file, has been deprecated in favor of the mmctl config reload command that you must run to apply configuration changes you've made. This improves configuration performance and robustness.

Deprecated configuration settings

See the deprecated configuration settings documentation for details on all deprecated Mattermost configuration settings that are no longer supported.

On this page

- Configuration settings
  - Configuration in database
  - Environment variables
  - Configuration reload
  - Deprecated configuration settings

Custom branding tools

Use custom branding tools to present a Mattermost experience that's tailored to the branding of your organization.

Enable custom branding

1. Go to **System Console > Site Configuration > Customization > Enable Custom Branding** and set the value to **true**.
2. Choose a **Site Name**, upload a **Custom Brand Image**, then enter **Custom Brand Text**.
3. Choose **Save**.

Users should see your new custom branding on the login page of your Mattermost server the next time they log in.

You can remove the custom brand image by selecting the **X** in the top right corner of the image. To apply the change, choose **Save**. Note that the value must be set to **true** in order to perform this step.

More about settings available in **Customization**:

Site name

Choose the name of your Mattermost site to be shown in the UI. The site name appears in the header and footer of the site login page, team selection page, team creation page, account creation page, email invitations, and

replaces “Mattermost” on the **About** page. Note that the site name applies to the entire site and not just a specific team on the site. The site name is limited to 30 characters or less.

Custom branding image

Upload a company logo or custom image representative of your site that is displayed on the left side of the site login page. Supported image formats are JPG, PNG, TIFF, and BMP. The recommended image size is 200-500px in width and height, and less than 2 MB since it’s loaded for every user who logs in on desktop.

Custom brand text

Write custom text to display your company tagline or a welcome prompt. Custom text will be shown below the custom brand image on the left side of site login page on desktop. You can format this text to a maximum of 500 characters using the same Markdown formatting syntax as used in Mattermost messages.

Site description

Use this field to describe the purpose of your site. The site description is shown below the site name on the login page. This text defaults to **All team communication in one place, searchable and accessible anywhere**. This text can be a maximum of 1024 characters and is not formatted with Markdown.

On this page

- Custom branding tools
  - Enable custom branding
    - Site name
    - Custom branding image
    - Custom brand text
    - Site description

Customizing Mattermost

There are several ways to customize your Mattermost server.

If customizing Mattermost, please avoid branding that could be confused with the Mattermost brand. For example, it's okay to brand as "Healthcare Central" because it's a completely different brand. "Mattermost Healthcare Central" is not okay, because it can potentially be confused with the Mattermost brand. Please see the Mattermost trademark guidelines (https://mattermost.com/trademark-standards-of-use/) for details.

Mattermost Web App

The Mattermost webapp is licensed under the Apache 2.0 license. To modify and use with the Mattermost server, you can:

1. Install the Mattermost server by following one of our installation guides
2. Fork the mattermost (https://github.com/mattermost/mattermost) repository
3. Go to the web app code located in the `webapp` directory and make your changes
4. Run `make package` to create `mattermost-webapp.tar.gz`
5. Copy `mattermost-webapp-tar.gz` to the location Mattermost was installed in Step 1
6. Rename the existing `client` folder to `client-original`
7. Run `tar -xvf mattermost-webapp.tar.gz` to extract your new customized `client` folder
8. If you're installing Mattermost 7.5 - 7.10, copy the `products` folder from `client-original` into `client`
9. Restart your Mattermost server

It is possible to customize certain parts of the webapp without forking by using our Custom Branding settings.

To replace the logo in email notifications, change the file located in the `/images` directory. To change the app icon, modify the `/app/components/app_icon.js` file.

Mattermost Server

There are a few things you can customize in the Mattermost server without forking:

1. Modify text in the Mattermost interface by modifying the `en.json` file.
2. Customize or hide help and support links by modifying your configuration settings.
3. Customize the email notifications by editing the HTML files in `/templates`.

Mattermost mobile apps

The Mattermost mobile apps can be customized if you choose to build the apps yourself.

To brand the mobile apps:

1. Fork the [mattermost-mobile (https://github.com/mattermost/mattermost-mobile)](https://github.com/mattermost/mattermost-mobile) repository.
2. Replace the name, images, and any key text strings.
3. Compile the custom apps.
4. Deploy the apps to an app store.

While most organizations deploy to internal enterprise app stores, you are welcome to deploy to iTunes and Google Play as long as the branding is not confusable with official Mattermost products.

Mattermost desktop apps

The Mattermost desktop apps can be customized if you choose to build the apps yourself.

To brand the desktop apps:

1. Fork the mattermost/desktop (https://github.com/mattermost/desktop) repository
2. Replace the name, images, and any key text strings
3. Refer to the Mattermost Developer documentation for help with compiling the apps
4. Share the desktop application with your users

On this page

- Customizing Mattermost
  - Mattermost Web App
  - Mattermost Server
  - Mattermost mobile apps
  - Mattermost desktop apps

Deprecated configuration settings

The following Mattermost configuration settings are deprecated and are no longer supported in current Mattermost releases:

- Bleve settings
- Elasticsearch settings
- Service settings
- Database settings
- Data retention settings
- Users and teams settings
- SAML 2.0 settings
- Legacy sidebar settings
- Town Square channel settings
- Custom emoji settings
- Timezone settings
- High availablity settings
- Rest API V3 settings
- Integrations settings
- Permission policy settings
- Image settings
- Experimental display settings
- Experimental API endpoint settings
- Shared channels settings
- Other deprecated settings

Bleve settings

Bulk Indexing Time Window Seconds

Removed in May 16, 2022 release

This setting isn't available in the System Console and can only be set in `config.json`.

Determines the maximum time window for a batch of posts being indexed by the Bulk Indexer. This setting serves as a performance optimization for installs with over ~10 million posts in the database. You can approximate this value based on the average number of seconds for 2,000 posts to be added to the database on a typical day in production. Setting this value too low will cause bulk indexing jobs to run slowly.

> This feature's `config.json` setting is `"BulkIndexingTimeWindowSeconds": 3600` with numerical input.

Elasticsearch settings

Bulk Indexing Time Window

Removed in May 16, 2022 release

This setting isn't available in the System Console and can only be set in `config.json`.

Determines the maximum time window for a batch of posts being indexed by the Bulk Indexer. This setting serves as a performance optimization for installs with over ~10 million posts in the database. You can approximate this value based on the average number of seconds for 2,000 posts to be added to the database on a typical day in production. Setting this value too low will cause bulk indexing jobs to run slowly.

This feature's `config.json` setting is `"BulkIndexingTimeWindowSeconds": 3600` with numerical input.

Service settings

Enable reliable websockets

This configuration setting has been deprecated, and the ability to buffer messages during a connection loss has been promoted to general availability from Mattermost v6.3. This setting is enabled for older clients to maintain backwards compatibility.

This setting isn't available in the System Console and can only be set in `config.json`.

Enable this setting to make websocket messages more reliable by buffering messages during a connection loss and then re-transmitting all unsent messages when the connection is revived.

This feature's `config.json` setting is `"EnableReliableWebsockets": true` with options `true` and `false`.

Data prefetch

Removed in February 16, 2021 release

True: Messages in all unread channels are pre-loaded from the server whenever the client reconnects to the network to eliminate loading time when users switch to unread channels.

False: Messages are fetched on-demand from the server when users switch channels.

This feature's `config.json` setting is `"ExperimentalDataPrefetch": true` with options `true` and `false`.

Segment write key

Removed in March 16, 2017 release

For deployments seeking additional tracking of system behavior using Segment.com, you can enter a Segment `WRITE_KEY` using this field. This value works like a tracking code and is used in client-side JavaScript and will send events to Segment.com attributed to the account you used to generate the `WRITE_KEY`.

> This feature's `config.json` setting is `"SegmentDeveloperKey": ""` with string input.

Limit access to config settings prior to login

Removed in December 16, 2018 release

Enable this setting to limit the number of config settings sent to users prior to login.

Supported for Mattermost server v5.1.0 and later, and Mattermost Mobile apps v1.10.0 and later.

> This feature's `config.json` setting is `"ExperimentalLimitClientConfig": "false"` with options `true` and `false`.

Disable legacy MFA API endpoint

Deprecated. Not used in Mattermost v6.0 and later.

Database settings

At rest encrypt key

Removed in August 23, 2018 release

This setting isn't available in the System Console and can only be set in `config.json`. It's a legacy setting used to encrypt data stored at rest in the database, and no fields are encrypted using `AtRestEncryptKey`.

A 32-character key for encrypting and decrypting sensitive fields in the database. When using high availability, this value must be identical in each instance of Mattermost.

> This feature's `config.json` setting is `"AtRestEncryptKey": ""` with string input.

Amazon S3 bucket endpoint

Removed in November 16th, 2016 release

Set an endpoint URL for Amazon S3 buckets.

> This feature's `config.json` setting is `"AmazonS3BucketEndpoint": ""` with string input.

Amazon S3 Location Constraint

Removed in November 16th, 2016 release

True: S3 region is location constrained.

False: S3 region is not location constrained.

> This feature's `config.json` setting is `"AmazonS3LocationConstraint": false` with options `true` and `false`.

Amazon S3 lowercase bucket

Removed in November 16th, 2016 release

True: S3 bucket names are fully lowercase.

False: S3 bucket names may contain uppercase and lowercase letters.

This feature's `config.json` setting is `"AmazonS3LowercaseBucket": false` with options `true` and `false`.

Data retention settings

MessageRetentionDays

Deprecated in Mattermost v9.5 release in favor of MessageRetentionHours

Set how long Mattermost keeps messages across all teams and channels. This setting doesn't apply to custom retention policies. The minimum time is 1 hour.

From Mattermost v9.5, this setting has been replaced by MessageRetentionHours which provides more granular control over message retention periods.

This feature's `config.json` setting is `"MessageRetentionDays": 365` with numerical input.

FileRetentionDays

Deprecated in Mattermost v9.5 release in favor of FileRetentionHours

Set how long Mattermost keeps files across all teams and channels. This setting doesn't apply to custom retention policies. The minimum time is 1 hour.

From Mattermost v9.5, this setting has been replaced by FileRetentionHours which provides more granular control over file retention periods.

This feature's `config.json` setting is `"FileRetentionDays": 365` with numerical input.

Users and teams settings

Enable team directory

Removed in May 16th, 2016 release

True: Teams that are configured to appear in the team directory will appear on the system main page. Teams can configure this setting from **Team Settings > Include this team in the Team Directory**.

False: Team directory on the system main page is disabled.

> This feature's `config.json` setting is `"EnableTeamListing": false` with options `true` and `false`.

Allow team admins to edit others' posts

Deprecated. Not used in Mattermost v6.0 and later.

Enable team creation

Deprecated. Not used in Mattermost v6.0 and later.

SAML 2.0 settings

Use new SAML library

Removed in December 16, 2020 release

True: Enable an updated SAML Library, which does not require the XML Security Library (xmlsec1) to be installed.

False: Continue using the existing implementation which uses the XML Security Library (xmlsec1).

This feature's `config.json` setting is `"UseNewSAMLLibrary": false` with options `true` and `false`.

Legacy sidebar settings

Enable legacy sidebar

Deprecated. Not used in Mattermost v6.0 and later.

Experimental sidebar features

Deprecated. Not used in Mattermost v5.32 and later

> Note
>
> This experimental configuration setting has been deprecated, and the ability to organize channels in the sidebar has been promoted to general availability from Mattermost v5.32. See the Organizing Your Sidebar documentation for details on customizing the sidebar.

Disabled: Users cannot access the experimental channel sidebar feature set.

Enabled (Default On): Enables the experimental sidebar features for all users on this server. Users can disable the features in **Settings > Sidebar > Experimental Sidebar Features**. Features include custom collapsible channel categories, drag and drop to reorganize channels, and unread filtering.

Enabled (Default Off): Users must enable the experimental sidebar features in **Settings**.

This feature's `config.json` setting is `"ExperimentalChannelSidebarOrganization": off` with options `off`, `default_on` and `default_off`.

Sidebar organization

Deprecated. Not used in Mattermost v6.0 and later.

Enable X to leave channels from left hand sidebar

Deprecated. Not used in Mattermost v6.0 and later.

Autoclose direct messages in sidebar

Deprecated. Not used in Mattermost v6.0 and later.

Town Square channel settings

Town Square is hidden in left hand sidebar

Deprecated. Not used in Mattermost v6.0 and later.

Town Square is read-only

From Mattermost v.6.0, this feature has been deprecated in favor of advanced access controls which allows you to set any channel as read-only, including Town Square.

Custom emoji settings

Restrict custom emoji creation

Deprecated. Not used in Mattermost v6.0 and later.

Timezone settings

Timezone

This configuration setting has been promoted to General Availability and is no longer configurable in Mattermost v6.0 and later.

Select the timezone used for timestamps in the user interface and email notifications.

True: The **Timezone** setting is visible in the Settings and a timezone is automatically assigned in the next active session.

False: The **Timezone** setting is hidden in the Settings.

This feature's `config.json` setting is `"ExperimentalTimezone": true` with options `true` and `false`.

High availability settings

Inter-node listen address

Deprecated. Not used in Mattermost v4.0 and later

The address the Mattermost Server will listen on for inter-node communication. When setting up your network you should secure the listen address so that only machines in the cluster have access to that port. This can be done in different ways, for example, using IPsec, security groups, or routing tables.

This feature's `config.json` setting is `"InterNodeListenAddress": ":8075"` with string input.

Inter-Node URLs

Deprecated. Not used in Mattermost v4.0 and later

A list of all the machines in the cluster, such as `["http://10.10.10.2", "http://10.10.10.4"]`. It is recommended to use the internal IP addresses so all the traffic can be secured.

> This feature's `config.json` setting is `"InterNodeUrls": []` with string array input consisting of the machines in the cluster.

Use gossip

Removed in Mattermost v6.0

True: The server attempts to communicate via the gossip protocol over the gossip port specified.

False: The server attempts to communicate over the streaming port.

> This feature's config.json setting is `"UseExperimentalGossip": true` with options `true` and `false`.

Streaming port

Deprecated. Not used in Mattermost v6.0 and later

The port used for streaming data between servers.

> This feature's `config.json` setting is `"StreamingPort": ":8075"` with string input.

Maximum idle database connections

Deprecated. Not used in Mattermost v7.0 and later

The maximum number of idle connections held open to the database. Numerical input. Default is **20**.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.MaxIdleConns": 20,` • Environment variable: `MM_SQLSETTINGS_MAXIDLECONNS`

Maximum idle connections for high availability

Deprecated. Not used in Mattermost v7.0 and later

The maximum number of idle connections held open from one server to all others in the cluster. Numerical input. Default is **100**.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.MaxIdleConns: 100,` • Environment variable: `MM_CLUSTERSETTINGS_MAXIDLECONNS`

Maximum idle connections per host

Deprecated. Not used in Mattermost v7.0 and later

The maximum number of idle connections held open from one server to another server in the cluster. Numerical input. Default is **128**.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.MaxIdleConnsPerHost: 128",` • Environment variable: `MM_CLUSTERSETTINGS_MAXIDLECONNSPERHOST`

Idle connection timeout

Deprecated. Not used in Mattermost v7.0 and later

The amount of time, in milliseconds, to leave an idle connection open between servers in the cluster. Numerical input. Default is **90000**.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.IdleConnTimeoutMilliseconds: 90000",` • Environment variable: `MM_CLUSTERSETTINGS_IDLECONNTIMEOUTMILLISECONDS`

REST API V3 settings

Allow use of API v3 endpoints

Removed in June 16, 2018 release

Set to `false` to disable all version 3 endpoints of the REST API. Integrations that rely on API v3 will fail and can then be identified for migration to API v4. API v3 is deprecated and will be removed in the near future. See https://api.mattermost.com for details.

This feature's `config.json` setting is `"EnableAPIv3": false` with options `true` and `false`.

Integrations settings

Restrict managing integrations to Admins

Deprecated. Not used in Mattermost v6.0 and later.

Patch React DOM used by plugins

Deprecated. Not used in Mattermost v8.0 and later

This setting enables the patching of the React DOM library when loading web app plugins so that the plugin uses the version matching the web app. This

should only be needed temporarily after upgrading to Mattermost v7.7 for plugins that have not been updated yet. Changes to this setting require a server restart before taking effect.

See the Important Upgrade Notes for more information.

True: Web app plugins that package their own version of React DOM are patched to instead use the version of React DOM provided by the web app.

False: Web app plugins are loaded as normal.

This feature's `config.json` setting is `"PatchPluginsReactDOM": false` with options `true` and `false`.

Permission policy settings

Removed in June 16, 2018 release

> Note
>
> From Mattermost v5.0, these settings are found in the Advanced Permissions page instead of configuration settings.

Enable sending team invites from

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Set policy on who can invite others to a team using the **Send Email Invite**, **Get Team Invite Link**, and **Add Members to Team** options on the product

menu. If **Get Team Invite Link** is used to share a link, you can expire the invite code from **Team Settings > Invite Code** after the desired users have joined the team. Options include:

All team members: Allows any team member to invite others using an email invitation, team invite link, or by adding members to the team directly.

Team and System Admins: Hides the email invitation, team invite link, and the add members to team buttons in the product menu from users who are not team admins or system admins.

System Admins: Hides the email invitation, team invite link, and add members to team buttons in the product menu from users who are not system admins.

This feature's `config.json` setting is `"RestrictTeamInvite": "all"` with options `"all"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable public channel creation for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to create public channels.

All team members: Allow all team members to create public channels.

Team Admins and System Admins: Restrict creating public channels to team admins and system admins.

System Admins: Restrict creating public channels to system admins.

This feature's `config.json` setting is `"RestrictPublicChannelCreation": "all"` with options `"all"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable public channel renaming for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to rename and set the header or purpose for Public channels.

All channel members: Allow all channel members to rename Public channels.

Channel Admins, Team Admins, and System Admins: Restrict renaming public channels to channel admins, team admins, and system admins who are members of the channel.

Team Admins and System Admins: Restrict renaming public channels to Team Admins and system admins who are members of the channel.

System Admins: Restrict renaming public channels to system admins who are members of the channel.

This feature's `config.json` setting is `"RestrictPublicChannelManagement": "all"` with options `"all"`, `"channel_admin"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable public channel deletion for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to delete Public channels. Deleted channels can be recovered from the database using a command line tool.

All channel members: Allow all channel members to delete public channels.

Channel Admins, Team Admins, and System Admins: Restrict deleting public channels to channel admins, team admins, and system admins who are members of the channel.

Team Admins and System Admins: Restrict deleting public channels to team admins and system admins who are members of the channel.

System Admins: Restrict deleting public channels to system admins who are members of the channel.

This feature's `config.json` setting is `"RestrictPublicChannelDeletion": "all"` with options `"all"`, `"channel_admin"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable private channel creation for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to create private channels.

All team members: Allow all team members to create private channels.

Team Admins and System Admins: Restrict creating private channels to team admins and system admins.

System Admins: Restrict creating private channels to system admins.

This feature's `config.json` setting is `"RestrictPrivateChannelCreation": "all"` with options `"all"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable private channel renaming for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to rename and set the header or purpose for Private channels.

All channel members: Allow all channel members to rename private channels.

Channel Admins, Team Admins, and System Admins: Restrict renaming private channels to channel admins, team admins, and system admins who are members of the private channel.

Team Admins and System Admins: Restrict renaming private channels to team admins and system admins who are members of the private channel.

System Admins: Restrict renaming private channels to system admins who are members of the private channel.

This feature's `config.json` setting is `"RestrictPrivateChannelManagement": "all"` with options `"all"`, `"channel_admin"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable managing of private channel members for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Set policy on who can add and remove members from Private channels.

All team members: Allow all team members to add and remove members.

Team Admins, Channel Admins, and System Admins: Allow only team admins, channel admins, and system admins to add and remove members.

Team Admins, and System Admins: Allow only team admins and system admins to add and remove members.

System Admins: Allow only system admins to add and remove members.

This feature's `config.json` setting is `"RestrictPrivateChannelManageMembers": "all"` with options `"all"`, `"channel_admin"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Enable private channel deletion for

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to delete Private channels. Deleted channels can be recovered from the database using a command line tool.

All channel members: Allow all channel members to delete private channels.

Channel Admins, Team Admins, and System Admins: Restrict deleting private channels to channel admins, team admins, and system admins who are members of the Private channel.

Team Admins and System Admins: Restrict deleting private channels to Team Admins and system admins who are members of the Private channel.

System Admins: Restrict deleting private channels to system admins who are members of the private channel.

This feature's `config.json` setting is `"RestrictPrivateChannelDeletion": "all"` with options `"all"`, `"channel_admin"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Allow which users to delete messages

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Restrict the permission level required to delete messages. Team admins, channel admins, and system admins can delete messages only in channels where they are members. Messages can be deleted any time.

Message authors can delete their own messages, and Administrators can delete any message: Allow authors to delete their own messages, and allow team admins, channel admins, and system admins to delete any message.

Team Admins and System Admins: Allow only team admins and system admins to delete messages.

System Admins: Allow only system admins to delete messages.

This feature's `config.json` setting is `"RestrictPostDelete": "all"` with options `"all"`, `"team_admin"`, and `"system_admin"` for the above settings, respectively.

Allow users to edit their messages

Removed in June 16, 2018 release

> Note
>
> From v5.0 this has been replaced by advanced permissions which offers Admins a way to restrict actions in Mattermost to authorized users only. See the Advanced Permissions documentation for more details.

Set the time limit that users have to edit their messages after posting.

Any time: Allow users to edit their messages at any time after posting.

Never: Do not allow users to edit their messages.

{n} seconds after posting: Users can edit their messages within the specified time limit after posting. The time limit is applied using the `config.json` setting `PostEditTimeLimit` described below.

This feature's `config.json` setting is `"AllowEditPost": "always"` with options `"always"`, `"never"`, and `"time_limit"` for the above settings, respectively.

Post edit time limit

When post editing is permitted, setting this to `-1` allows editing any time, and setting this to a positive integer restricts editing time in seconds. If post editing is disabled, this setting does not apply.

Note: This setting does not affect plugins, shared channels, integration actions, or Mattermost products.

This feature's `config.json` setting is `"PostEditTimeLimit": -1` with numerical input.

Image settings

Attachment thumbnail width

Removed in July 16th, 2017 release

Width of thumbnails generated from uploaded images. Updating this value changes how thumbnail images render in future, but does not change images created in the past.

This feature's `config.json` setting is `"ThumbnailWidth": 120` with numerical input.

Attachment thumbnail height

Removed in July 16th, 2017 release

Height of thumbnails generated from uploaded images. Updating this value changes how thumbnail images render in future, but does not change images created in the past.

This feature's `config.json` setting is `"ThumbnailHeight": 100` with numerical input.

Image preview width

Removed in July 16th, 2017 release

Maximum width of preview image. Updating this value changes how preview images render in future, but does not change images created in the past.

This feature's `config.json` setting is `"PreviewWidth": 1024` with numerical input.

Image preview height

Removed in July 16th, 2017 release

Maximum height of preview image. Setting this value to `0` instructs Mattermost to auto-size the preview image height based on the source image aspect ratio and the preview image width. Updating this value changes how preview images render in future, but does not change images created in the past.

This feature's `config.json` setting is `"PreviewHeight": 0` with numerical input.

Profile picture width

Removed in July 16th, 2017 release

The width to which profile pictures are resized after being uploaded via **Account Settings > Profile**.

This feature's `config.json` setting is `"ProfileWidth": 128` with numerical input.

Profile picture height

Removed in July 16th, 2017 release

The height to which profile pictures are resized after being uploaded via **Account Settings > Profile**.

This feature's `config.json` setting is `"ProfileHeight": 128` with numerical input.

Experimental display settings

Supported timezones path

Removed in April 16, 2019 release

Set the path of the JSON file that lists supported timezones when `ExperimentalTimezone` is set to `true`.

The file must be in the same directory as your `config.json` file if you set a relative path. Defaults to `timezones.json`.

This feature's `config.json` setting is `"SupportedTimezonesPath": "timezones.json"` with string input.

Shared channels settings

Enable remote cluster service (Experimental)

Deprecated in November 16th, 2024 release in favor of Connected Workspaces configuration settings

This setting isn't available in the System Console and can only be set in `config.json`.

Enable this setting to add, remove, and view remote clusters for shared channels.

True: System admins can manage remote clusters using the System Console.

False: (**Default**) Remote cluster management is disabled.

> This feature's `config.json` setting is `"ExperimentalSettings.EnableRemoteClusters": false` with options `true` and `false`.

Enable shared channels (Experimental)

Deprecated in November 16th, 2024 release in favor of Connected Workspaces configuration settings

This setting isn't available in the System Console and can only be set in `config.json`.

Enable the ability to establish secure connections between Mattermost instances, and invite secured connections to shared channels where secure connections can participate as they would in any public and private channel.

True: System admins can establish secure connections between Mattermost instances.

False: (**Default**) The ability to establish secure connections is disabled.

> This feature's `config.json` setting is `"ExperimentalSettings.EnableSharedChannels": false` with options `true` and `false`.

Other deprecated settings

Disable Post Metadata

This setting isn't available in the System Console and can only be set in `config.json`.

True: Disabling post metadata is only recommended if you are experiencing a significant decrease in performance around channel and post load times.

False: Load channels with more accurate scroll positioning by loading post metadata.

> This feature's `config.json` setting is `"DisablePostMetadata": false` with options `true` and `false`.

Enable AD/LDAP group sync

True: Enables AD/LDAP Group Sync configurable under **User Management > Groups**.

False: Disables AD/LDAP Group Sync and removes **User Management > Groups** from the System Console.

For more information on AD/LDAP Group Sync, please see the AD/LDAP Group Sync documentation.

> This feature's `config.json` setting is `"ExperimentalLdapGroupSync": false` with options `true` and `false`.

Disable inactive server email notifications

This setting isn't available in the System Console and can only be set in `config.json`.

This configuration setting disables the ability to send inactivity email notifications to Mattermost System Admins.

This feature's `config.json` setting is `"EnableInactivityEmail": true` with options `true` and `false`.

Disable Apps Bar

This setting is enabled for all customers by default from Mattermost v8.0. This setting disables the Apps Bar and moves all Mattermost integration icons from the vertical pane on the far right back to the channel header.

Enable OpenTracing (Experimental)

Removed in March 16, 2025 release

This setting isn't available in the System Console and can only be set in `config.json`.

True: A Jaeger client is instantiated and is used to trace each HTTP request as it goes through App and Store layers. Context is added to App and Store and is passed down the layer chain to create OpenTracing 'spans'.

By default, in order to avoid leaking sensitive information, no method parameters are reported to OpenTracing. Only the name of the method is reported.

False: OpenTracing is not enabled.

This feature's `config.json` setting is `"EnableOpenTracing": false` with options `true` and `false`.

On this page

- Deprecated configuration settings
  - Bleve settings
    - Bulk Indexing Time Window Seconds
  - Elasticsearch settings
    - Bulk Indexing Time Window

- Service settings
  - Enable reliable websockets
  - Data prefetch
  - Segment write key
  - Limit access to config settings prior to login
  - Disable legacy MFA API endpoint
- Database settings
  - At rest encrypt key
  - Amazon S3 bucket endpoint
  - Amazon S3 Location Constraint
  - Amazon S3 lowercase bucket
- Data retention settings
  - MessageRetentionDays
  - FileRetentionDays
- Users and teams settings
  - Enable team directory
  - Allow team admins to edit others’ posts
  - Enable team creation
- SAML 2.0 settings
  - Use new SAML library
- Legacy sidebar settings
  - Enable legacy sidebar
  - Experimental sidebar features
  - Sidebar organization
  - Enable X to leave channels from left hand sidebar
  - Autoclose direct messages in sidebar
- Town Square channel settings
  - Town Square is hidden in left hand sidebar
  - Town Square is read-only
- Custom emoji settings
  - Restrict custom emoji creation
- Timezone settings
  - Timezone

- High availability settings
  - Inter-node listen address
  - Inter-Node URLs
  - Use gossip
  - Streaming port
  - Maximum idle database connections
  - Maximum idle connections for high availability
  - Maximum idle connections per host
  - Idle connection timeout
- REST API V3 settings
  - Allow use of API v3 endpoints
- Integrations settings
  - Restrict managing integrations to Admins
  - Patch React DOM used by plugins
- Permission policy settings
  - Enable sending team invites from
  - Enable public channel creation for
  - Enable public channel renaming for
  - Enable public channel deletion for
  - Enable private channel creation for
  - Enable private channel renaming for
  - Enable managing of private channel members for
  - Enable private channel deletion for
  - Allow which users to delete messages
  - Allow users to edit their messages
  - Post edit time limit
- Image settings
  - Attachment thumbnail width
  - Attachment thumbnail height
  - Image preview width
  - Image preview height
  - Profile picture width
  - Profile picture height

- Experimental display settings
  - Supported timezones path
- Shared channels settings
  - Enable remote cluster service (Experimental)
  - Enable shared channels (Experimental)
- Other deprecated settings
  - Disable Post Metadata
  - Enable AD/LDAP group sync
  - Disable inactive server email notifications
  - Disable Apps Bar
  - Enable OpenTracing (Experimental)

Email templates

Mattermost has a few email templates that are sent out when a specific event occurs. Most of the time these templates do not need to be modified. In case additional modifications are necessary, all available props in each email are listed below. The 'Content'-Field is just to give a quick description of the prop. Please check the i18n strings for the exact wording.

The email templates are located in the Mattermost server directory in the `templates` folder. The corresponding strings for each prop can be found in the `i18n` folder.

> Note
>
> The props between different email templates are not interchangeable without additional server code changes.

> Warning
>
> Changes made inside of the `templates` or `i18n` folder might get overwritten during a server update. Please make sure to backup them accordingly.

Available templates

Email footer

Purpose: This is appended to all outgoing emails sent by Mattermost.

Props:

Prop	Content	i18n String
Footer	Message intro	api.templates.email_footer
Organization	Name of Organization	api.templates.email_organization
EmailInfo1	First line of footer	api.templates.email_info1
EmailInfo2	Second line of footer	api.templates.email_info2
EmailInfo3	Third line of footer	api.templates.email_info3
SupportEmail	Email for Mattermost support	–

SendChangeUsernameEmail

Purpose: Sent to the user when username has been changed.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.username_change_body.title
Info	Message body	api.templates.username_change_body.info
Warning	Warning text	api.templates.email_warning

SendEmailChangeVerifyEmail

Purpose: Sent to the user when an email change has been requested. Contains verification link and button.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.email_change_verify_body.title
Info	Message body	api.templates.email_change_verify_body.info
VerifyUrl	URL for email verification	–
VerifyButton	Button for email verification	api.templates.email_change_verify_body.button

SendEmailChangeEmail

Purpose: Sent to the user when the email has been changed successfully.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.email_change_body.title
Info	Message body	api.templates.email_change_body.info
Warning	Warning text	api.templates.email_warning

SendVerifyEmail

Purpose: Sent to the user upon account creation to verify email address.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.verify_body.title
Info	Message body	api.templates.verify_body.info
VerifyUrl	URL for email verification	–
Button	Button for email verification | api.templates.verify_body.button	

SendSignInChangeEmail

Purpose: Sent to the user when the login method has been changed (i.e. from email to LDAP, etc.)

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.signin_change_email.body.title
Info	Message body	api.templates.signin_change_email.body.info
Warning	Warning text	api.templates.email_warning

SendWelcomeEmail

Purpose: Sent to the user when the account has been created. May also contain download links to Apps as well as email verification links.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.welcome_body.title
Info	Message body	api.templates.welcome_body.info
Button	Button for confirmation	api.templates.welcome_body.button
Info2	Continuation of message body	api.templates.welcome_body.info2
Info3	Continuation of message body	api.templates.welcome_body.info3

Optional Props:

Prop	Content	i18n String
AppDownloadInfo	Info for App Downloads	api.templates.welcome_body.app_download_info
AppDownloadLink	Download link for Apps	–
VerifyUrl	Link for verification	–

SendPasswordChangeEmail

Purpose: Sent to the user when password has been changed.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.password_change_body.title
Info	Message body	api.templates.password_change_body.info
Warning	Warning text	api.templates.email_warning

SendAccessTokenEmail

Purpose: Sent to the user when an access token has been added to the account.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.user_access_token_body.title
Info	Message body	api.templates.user_access_token_body.info
Warning	Warning text	api.templates.email_warning

SendPasswordResetEmail

Purpose: Sent to the user when password request has been initiated.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.reset_body.title
Info1	Message body	api.templates.reset_body.info1
Info2	Continuation of message body	api.templates.reset_body.info2
ResetUrl	URL to reset password	–
Button	Button for confirmation	api.templates.reset_body.button

SendMfaChangeEmail

Purpose: Sent to the user when multi-factor authentication method has been changed.

Body Props when MFA is activated:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.mfa_activated_body.title
Info	Message body	api.templates.mfa_activated_body.info
Warning	Warning text	api.templates.email_warning

Body Props when MFA is deactivated:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.mfa_deactivated_body.title
Info	Message body	api.templates.mfa_deactivated_body.info
Warning	Warning text	api.templates.email_warning

SendDeactivateAccountEmail

Purpose: Sent to the user when account has been deactivated.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.deactivate_body.title
Info	Message body	api.templates.deactivate_body.info
Warning	Warning text	api.templates.deactivate_body.warning

SendInviteEmails

Purpose: Sent to the user when team invite via email has been used.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Title	Message heading	api.templates.invite_body.title
Info1	Message body	api.templates.invite_body.info
Button	Button for confirmation	api.templates.invite_body.button
ExtraInfo	Additional info about Mattermost	api.templates.invite_body.extra_info
TeamURL	URL to the team the user has been invited to	–
Link	URL for team invite confirmation (not to be confused with TeamURL)	–

NotificationEmailBody

Purpose: Sent to the user as a notification for new messages or mentions.

Body Props:

Prop	Content	i18n String
SiteURL	URL of the Mattermost server	–
Button	Button to post	api.templates.post_body.button
TeamLink	URL to Team	–

This email can change depending on the settings and type of channel the notification is sent for.

For group channels:

With full notification contents enabled:

Prop	Content	i18n String
BodyText	Message intro	app.notification.body.intro.group_message.full
Info1	Channel name	app.notification.body.text.group_message.full
Info2	Message contents	app.notification.body.text.group_message.full2
SenderName	Name of sender	–

Without:

Prop	Content	i18n String
BodyText	Message intro	app.notification.body.intro.group_message.generic
Info	Timestamp	app.notification.body.text.group_message.generic

For direct messages:

With full notification contents enabled:

Prop	Content	i18n String
BodyText	Message intro	app.notification.body.intro.direct.full
Info1	Empty for direct messages	–
Info2	Message contents	app.notification.body.text.direct.full
SenderName	Name of sender	–

Without:

Prop	Content	i18n String
BodyText	Message intro	app.notification.body.intro.direct.generic
Info	Timestamp	app.notification.body.text.direct.generic

Notifications:

With full notification contents enabled:

Prop	Content	i18n String
BodyText	Message intro	app.notification.body.intro.notification.full
Info1	Channel name	app.notification.body.text.notification.full
Info2	Message contents	app.notification.body.text.notification.full2
SenderName	Name of sender	–

Without:

Prop	Content	i18n String
BodyText	URL of the Mattermost server	app.notification.body.intro.notification.generic
Info	Message heading	app.notification.body.text.notification.generic

On this page

- Email templates
  - Available templates
    - Email footer
    - SendChangeUsernameEmail
    - SendEmailChangeVerifyEmail
    - SendEmailChangeEmail
    - SendVerifyEmail
    - SendSignInChangeEmail
    - SendWelcomeEmail
    - SendPasswordChangeEmail

- SendAccessTokenEmail
- SendPasswordResetEmail
- SendMfaChangeEmail
- SendDeactivateAccountEmail
- SendInviteEmails
- NotificationEmailBody

Chinese, Japanese and Korean search

Enabling search for Chinese, Japanese and Korean (CJK) requires special configuration, since these languages do not contain spaces.

See database requirements documentation for how to set up search for these languages.

Below is additional information on how to configure the database for different languages.

中文 / Chinese

尽管在 Mattermost 8.0 更新后，官方推荐为了更好的性能请使用 PostgreSQL 作为后端数据库。

但就目前而言，使用 MySQL 能够更容易的实现中文语言的全文搜索功能，在妥善配置 ngram 后，根据官方数据库构造重新生成索引即可达成。 具体的操作方式，可参考： [Cannot search CJK contents (https://github.com/mattermost/mattermost/issues/2033#issuecomment-182336690)](https://github.com/mattermost/mattermost/issues/2033#issuecomment-182336690)。

有关 PostgreSQL 的配置方式，请参考以下流程：

配置 SCWS

```
# 取得 SCWS 代码 wget http://www.xunsearch.com/scws/down/scws-1.2.3.tar.bz2 # 解压缩 tar xvjf scws-1.2.3.tar.bz2 # 进入解压后的目录 cd scws-1.2.3 # 执行配置脚本、编译并安装 ./configure --prefix=/usr/local/scws ; make ; make install # 可选：检查文件是否存在 ls -al /usr/local/scws/lib/libscws.la /usr/local/scws/bin/scws -h # 可选：将词典安装在 /usr/local/scws/etc 中 cd /usr/local/scws/etc wget http://www.xunsearch.com/scws/down/scws-dict-chs-gbk.tar.bz2 wget http://www.xunsearch.com/scws/down/scws-dict-chs-utf8.tar.bz2 tar xvjf scws-dict-chs-gbk.tar.bz2 tar xvjf scws-dict-chs-utf8.tar.bz2
```

配置 Zhparser

```
# 下载 Zhparser 源代码 git clone https://github.com/amutu/zhparser.git # 进入下载后的目录 cd zhparser # 编译并安装 SCWS_HOME=/usr/local/scws make && make install
```

> Note
>
> 自 Mattermost 6.0 起，官方已不再使用 mattermost/mattermost-prod-db 作为数据库镜像，你可以直接使用安装在服务器上的 PostgreSQL 数据库，或者使用 PostgreSQL 官方的 Docker 镜像。
>
> 如果使用 Docker 镜像作为数据库，可以预先执行以下命令，安装依赖（请根据实际的 PostgreSQL 版本选择）。
>
> ```
> # 更新本地缓存 apt update # 配置 SCWS 时需要的依赖 apt install wget make gcc # 配置 Zhparser 时需要的依赖 apt install git postgresql-server-dev-13
> ```

创建 extension 并增加解析配置

```
-- 创建 extension CREATE EXTENSION zhparser -- 创建 text search configuration CREATE TEXT SEARCH CONFIGURATION simple_zh_cfg (PARSER = zhparser); -- 配置 token mapping ALTER TEXT SEARCH CONFIGURATION simple_zh_cfg ADD MAPPING FOR n,v,a,i,e,l WITH simple;
```

更新 PostgreSQL 配置

将 postgresql.conf 中的 default_text_search_config 的值更改为 simple_zh_cfg。

更改后，需要重启数据库方可生效。

> Note
>
> 配置完成后，需根据 Mattermost 官方仓库中的 SQL 建表语句重新创建索引，方可正式启用中文语言的全文搜索功能。
>
> 未尽事宜，可以参考以下链接：
>
> - SCWS 官方文档
> - Zhparser 官方文档 (https://github.com/amutu/zhparser/blob/master/README.md)
> - Mattermost 建表语句 (https://github.com/mattermost/mattermost/tree/master/server/channels/db/migrations/postgres)

日本語 / Japanese

日本語翻訳の改善は大歓迎です。自由に変更していただいて結構です。

検索設定

Mattermost で日本語検索をするためにはデータベースの設定変更が必要です

- MySQL
- Postgres

日本語(CJK)検索設定のドキュメントの改善にご協力ください

ガイド

Qiita上で Mattermost のインストールおよび構成のガイドを提供しています。詳細については、こちら (https://qiita.com/tags/mattermost) をご覧ください。

한국어 / Korean

이 문제에 대한 논의는 이 이슈 (https://github.com/mattermost/mattermost/issues/2033) 에서 시작되었습니다.

한국어 버전 이용 시 문제점을 발견하면 Localization 채널 (https://community.mattermost.com/core/channels/localization) 또는 한국어 채널 (https://community.mattermost.com/core/channels/i18n-korean) 에서 의견을 제시할 수 있습니다.

검색을 위한 데이터베이스 설정

PostgreSQL: PostgreSQL 데이터베이스에서는 따로 설정이 필요하지 않습니다.

MySQL: MySQL에서는 전문 검색(Full-text search) 기능에 제한이 있기 때문에 추가적인 작업이 필요합니다.

MySQL 해결 방법

1. n-gram parser 를 이용하기 위해서는 MySQL의 버전이 5.7.6 이상이어야 합니다.
2. MySQL의 구성 파일에서 n-gram의 최소 토큰 크기를 다음과 같이 설정합니다.

```
[mysqld] ft_min_word_len = 2 innodb_ft_min_token_size = 2
```

3. 데이터베이스를 재시작합니다. (이 과정은 반드시 필요합니다.)
4. 일부 테이블의 전문 검색 색인을 다음과 같이 재구성합니다.

- 게시물 검색을 위한 설정 (참조 (https://github.com/mattermost/mattermost/issues/2033#issuecomment-182336690))

```
DROP INDEX idx_posts_message_txt ON Posts; CREATE FULLTEXT INDEX
idx_posts_message_txt ON Posts (Message) WITH PARSER ngram;
```

- 해시 태그 검색을 위한 설정 (참조 (https://github.com/mattermost/mattermost/pull/4555))

```
DROP INDEX idx_posts_hashtags_txt ON Posts; CREATE FULLTEXT INDEX
idx_posts_hashtags_txt ON Posts (Hashtags) WITH PARSER ngram;
```

- 사용자 검색을 위한 설정

  `Users.idx_users_txt_all` 과 `Users.idx_users_names_all` 을 n-gram 없이 재구성합니다.

On this page

- Chinese, Japanese and Korean search
  - 中文 / Chinese
    - 配置 SCWS
    - 配置 Zhparser
    - 创建 extension 并增加解析配置
    - 更新 PostgreSQL 配置
  - 日本語 / Japanese
    - 検索設定
    - ガイド
  - 한국어 / Korean
    - 검색을 위한 데이터베이스 설정
    - MySQL 해결 방법

Environment configuration settings

Review and manage the following environmental configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Environment**:

- Web server
- Database
- Enterprise search
- File storage
- Image proxy
- SMTP
- Push notification server
- High availability
- Rate limiting
- Logging
- Session lengths
- Performance monitoring
- Developer
- Mobile security
- config.json-only settings

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `SiteURL` value is under `ServiceSettings`.
>
> - If using a tool such as [jq (https://stedolan.github.io/jq/)](https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.ServiceSettings.SiteURL'`
> - When working with the `config.json` file manually, look for an object such as `ServiceSettings`, then within that object, find the key `SiteURL`.

Web server

Configure the network environment in which Mattermost is deployed by going to **System Console > Environment > Web Server**, or by updating the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Site URL

The URL that users use to access Mattermost. The port number is required if it's not a standard port, such as 80 or 443. This field is required. Select the **Test Live URL** button in the System Console to validate the Site URL.	• System Config path: **Environment > Web Server** • `config.json` setting: `.ServiceSettings.SiteURL",` • Environment variable: `MM_SERVICESETTINGS_SITEURL`

> Note
>
> - The URL may contain a subpath, such as `https://example.com/company/mattermost`.
> - If you change the Site URL value, log out of the Desktop App, and sign back in using the new domain.
> - If Site URL is not set:
> - Email notifications will contain broken links, and email batching will not work.
> - Authentication via OAuth 2.0, including GitLab, Google, and Entra ID, will fail.
> - Plugins may not work as expected.

Maximum URL length

The longest URL, in characters, including query parameters, accepted by the Mattermost server. Longer URLs are rejected, and API calls fail with an error. Numeric value. Default is **2048** characters.	• System Config path: N/A • `config.json` setting: `.ServiceSettings.MaximumURLLength: 2048",` • Environment variable: `MM_SERVICESETTINGS_MAXIMUMURLLENGTH`

Web server listen address

The address and port to which to bind and listen. Specifying `:8065` will bind to all network interfaces. Specifying `127.0.0.1:8065` will only bind to the network interface having that IP address. If you choose a port of a lower level (called "system ports" or "well-known ports", in the range of 0-1023), you must have permissions to bind to that port.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.ListenAddress",` • Environment variable: `MM_SERVICESETTINGS_LISTENADDRESS`

Forward port 80 to 443

Forward insecure traffic from port 80 to port 443. • **true**: Forwards all insecure traffic from port 80 to secure port 443. • **false**: **(Default)** When using a proxy such as NGINX in front of Mattermost this setting is unnecessary and should be set to false.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.Forward80To443: false",` • Environment variable: `MM_SERVICESETTINGS_FORWARD80TO443`

Web server connection security

Connection security between Mattermost clients and the server. • **Not specified**: Mattermost will connect over an unsecure connection. • **TLS**: Encrypts the communication between Mattermost clients and your server.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.ConnectionSecurity",` • Environment variable: `MM_SERVICESETTINGS_CONNECTIONSECURITY`

See the setting up TLS for Mattermost for details.

TLS certificate file

The path to the certificate file to use for TLS connection security. String input.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.TLSCertFile",` • Environment variable: `MM_SERVICESETTINGS_TLSCERTFILE`

TLS key file

The path to the TLS key file to use for TLS connection security. String input.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.TLSKeyFile",` • Environment variable: `MM_SERVICESETTINGS_TLSKEYFILE`

Use Let's Encrypt

Enable the automatic retrieval of certificates from Let's Encrypt. • **true**: The certificate will be retrieved when a client attempts to connect from a new domain. This will work with multiple domains. • **false**: **(Default)** Manual certificate specification based on the TLS Certificate File and TLS Key File specified above.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.UseLetsEncrypt: false",` • Environment variable: `MM_SERVICESETTINGS_USELETSENCRYPT`

See the setting up TLS for Mattermost for details on setting up Let's Encrypt.

Let's Encrypt certificate cache file

The path to the file where certificates and other data about the Let's Encrypt service will be stored. File path input.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.LetsEncryptCertificateCacheFile",` • Environment variable: `MM_SERVICESETTINGS_LETSENCRYPTCERTIFICATECACHEFILE`

Read timeout

Maximum time allowed from when the connection is accepted to when the request body is fully read. Numerical input in seconds. Default is **300** seconds.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.ReadTimeout: 300",` • Environment variable: `MM_SERVICESETTINGS_READTIMEOUT`

Write timeout

• If using HTTP (insecure), this is the maximum time allowed from the end of reading the request headers until the response is written. • If using HTTPS, it's the total time from when the connection is accepted until the response is written. Numerical input in seconds. Default is **300** seconds.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.WriteTimeout: 300",` • Environment variable: `MM_SERVICESETTINGS_WRITETIMEOUT`

Idle timeout

Set an explicit idle timeout in the HTTP server. This is the maximum time allowed before an idle connection is disconnected. Numerical input in seconds. Default is **300** seconds.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.IdleTimeout: 300",` • Environment variable: `MM_SERVICESETTINGS_IDLETIMEOUT`

Webserver mode

We recommend gzip to improve performance unless your environment has specific restrictions, such as a web proxy that distributes gzip files poorly. • **gzip**: **(Default)** The Mattermost server will serve static files compressed with gzip to improve performance. gzip compression applies to the HTML, CSS, Javascript, and other static content files that make up the Mattermost web client. • **Uncompressed**: The Mattermost server will serve static files uncompressed. • **Disabled**: The Mattermost server will not serve static files.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.WebserverMode: gzip",` • Environment variable: `MM_SERVICESETTINGS_WEBSERVERMODE`

Enable insecure outgoing connections

Configure Mattermost to allow insecure outgoing connections. • **true**: Outgoing HTTPS requests, including S3 clients, can accept unverified, self-signed certificates. For example, outgoing webhooks to a server with a self-signed TLS certificate, using any domain, will be allowed, and will skip TLS verification. • **false**: **(Default)** Only secure HTTPS requests are allowed.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.EnableInsecureOutgoingConnections: false",` • Environment variable: `MM_SERVICESETTINGS_ENABLEINSECUREOUTGOINGCONNECTIONS`

> Warning
>
> Enabling this feature makes these connections susceptible to man-in-the-middle attacks.

Managed resource paths

A comma-separated list of paths within the Mattermost domain that are managed by a third party service instead of Mattermost itself. Links to these paths will be opened in a new tab/window by Mattermost apps. For example, if Mattermost is running on `https://mymattermost.com`, setting this to conference will cause links such as `https://mymattermost.com/conference` to open in a new window.	• System Config path: **Environment > Web Server** • `config.json` setting: `".ServiceSettings.ManagedResourcePaths",` • Environment variable: `MM_SERVICESETTINGS_MANAGEDRESOURCEPATHS`

> Note
>
> When using the Mattermost Desktop App, additional configuration is required to open the link within the Desktop App instead of in a browser. See the desktop managed resources documentation for details.

Reload configuration from disk

Note

You must change the database line in the `config.json` file, and then reload configuration to fail over without taking the server down. Select the **Reload configuration from disk** button in the System Console after changing your database configuration. Then, go to **Environment > Database** and select **Recycle Database Connections** to complete the reload.	• System Config path: **Environment > Web Server** • `config.json` setting: N/A • Environment variable: N/A

Purge all caches

Purge all in-memory caches for sessions, accounts, and channels. Select the **Purge All Caches** button in the System Console to purge all caches.	• System Config path: **Environment > Web Server** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Purging the caches may adversely impact performance. high availability cluster-based deployments will attempt to purge all the servers in the cluster.

Websocket URL

You can configure the server to instruct clients on where they should try to connect websockets to. String input.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.WebsocketURL: "",` • Environment variable: `MM_SERVICESETTINGS_WEBSOCKETURL`

> Note
>
> We strongly recommend configuring a single websocket URL that matches the Site URL configuration setting.

License file location

Note

The path and filename of the license file on disk. On startup, if Mattermost can't find a valid license in the database from a previous upload, it looks in this path for the license file. String input. Can be an absolute path or a path relative to the `mattermost` directory.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.LicenseFileLocation: "",` • Environment variable: `MM_SERVICESETTINGS_LICENSEFILELOCATION`

TLS minimum version

The minimum TLS version used by the Mattermost server. String input. Default is **1.2**.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.TLSMinVer: 1.2",` • Environment variable: `MM_SERVICESETTINGS_TLSMINVER`

> Note
>
> This setting only takes effect if you are using the built-in server binary directly, and not using a reverse proxy layer, such as NGINX.

Trusted proxy IP header

Specified headers that will be checked, one by one, for IP addresses (order is important). All other headers are ignored. String array input consisting of header names, such as `["X-Forwarded-For", "X-Real-Ip"]`.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.TrustedProxyIPHeader: []",` • Environment variable: `MM_SERVICESETTINGS_TRUSTEDPROXYIPHEADER`

> Note
>
> - The default value of `[]` means that no header will be trusted.
> - We recommend keeping the default setting when Mattermost is running without a proxy to avoid the client sending the headers and bypassing rate limiting and/or the audit log.
> - For environments that use a reverse proxy, this issue does not exist, provided that the headers are set by the reverse proxy. In those environments, only explicitly whitelist the header set by the reverse proxy and no additional values.

Enable Strict Transport Security (HSTS)

• **true**: Adds the Strict Transport Security (HSTS) header to all responses, forcing the browser to request all resources via HTTPS. • **false**: **(Default)** No restrictions on TLS transport. Strict Transport Security (HSTS) header isn't added to responses.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.TLSStrictTransport: false",` • Environment variable: `MM_SERVICESETTINGS_TLSSTRICTTRANSPORT`

See the [Strict-Transport-Security (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security) documentation for details.

Secure TLS transport expiry

The time, in seconds, that the browser remembers a site is only to be accessed using HTTPS. After this period, a site can't be accessed using HTTP unless `TLSStrictTransport` is set to `true`. Numerical input. Default is **63072000** (2 years).	• System Config path: N/A • `config.json` setting: `".ServiceSettings.TLSStrictTransportMaxAge: 63072000",` • Environment variable: `MM_SERVICESETTINGS_TLSSTRICTTRANSPORTMAXAGE`

See the [Strict-Transport-Security (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security) documentation for details.

TLS cipher overwrites

Set TLS ciphers overwrites to meet requirements from legacy clients which don't support modern ciphers, or to limit the types of accepted ciphers. If none specified, the Mattermost server assumes a set of currently considered secure ciphers, and allows overwrites in the edge case. String array input.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.TLSOverwriteCiphers: []",` • Environment variable: `MM_SERVICESETTINGS_TLSOVERWRITECIPHERS`

> Note
>
> - This setting only takes effect if you are using the built-in server binary directly and not using a reverse proxy layer, such as NGINX.
> - See the `ServerTLSSupportedCiphers` variable in [/model/config.go (https://github.com/mattermost/mattermost/blob/master/server/public/model/config.go)](https://github.com/mattermost/mattermost/blob/master/server/public/model/config.go) for a list of ciphers considered secure.

Goroutine health threshold

Set a threshold on the number of goroutines when the Mattermost system is considered to be in a healthy state. When goroutines exceed this limit, a warning is returned in the server logs. Numeric input. Default is **-1** which turns off checking for the threshold.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.GoroutineHealthThreshold: -1",` • Environment variable: `MM_SERVICESETTINGS_GOROUTINEHEALTHTHRESHOLD`

Allow cookies for subdomains

• **true**: **(Default)** Allows cookies for subdomains by setting the domain parameter on Mattermost cookies. • **false**: Cookies not allowed for subdomains.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.AllowCookiesForSubdomains: true",` • Environment variable: `MM_SERVICESETTINGS_ALLOWCOOKIESFORSUBDOMAINS`

Cluster log timeout

Note

Define the frequency, in milliseconds, of cluster request time logging for performance monitoring. Numerical input. Default is **2000** milliseconds (2 seconds).	• System Config path: N/A • `config.json` setting: `".ServiceSettings.ClusterLogTimeoutMilliseconds: 2000",` • Environment variable: `MM_SERVICESETTINGS_CLUSTERLOGTIMEOUTMILLISECONDS`

See the performance monitoring documentation for details.

Maximum payload size

The maximum payload size in bytes for all APIs except APIs that receive a file as an input. For example, the upload attachment API or the API to upload a custom emoji. Numerical value. Default is **300000** (300 kB).	• System Config path: N/A • `config.json` setting: `".ServiceSettings.MaximumPayloadSizeBytes: 300000",` • Environment variable: `MM_SERVICESETTINGS_MAXIMUMPAYLOADSIZEBYTES`

Database

self-hosteddeployments

Configure the database environment in which Mattermost is deployed by going to **System Console > Environment > Database**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

[Mattermost Academy Learn about setting up the Mattermost database (https://mattermost.com/pl/mattermost-academy-database-configuration-training)](https://mattermost.com/pl/mattermost-academy-database-configuration-training)

Driver name

The type of database. Can be either: • **mysql**: **(Default)** Enables driver to MySQL database. • **postgres**: Enables driver to PostgreSQL database.	• System Config path: N/A • `config.json` setting: `".SqlSettings.DriverName",` • Environment variable: `MM_SQLSETTINGS_DRIVERNAME`

Data source

The connection string to the master database. String input.	• System Config path: N/A • `config.json` setting: `".SqlSettings.DataSource",` • Environment variable: `MM_SQLSETTINGS_DATASOURCE`

PostgreSQL databases

When **Driver Name** is set to postgres, use a connection string in the form of: `postgres://mmuser:password@hostname_or_IP:5432/mattermost_test?sslmode=disable&connect_timeout=10`

To use TLS with PostgreSQL databases

The parameter to encrypt connection against a PostgreSQL server is sslmode. The library used to interact with PostgreSQL server is [pq (https://pkg.go.dev/github.com/lib/pq)](https://pkg.go.dev/github.com/lib/pq). Currently, it's not possible to use all the values that you could pass to a standard PostgreSQL Client `psql "sslmode=value"` See the [SSL Mode Descriptions (https://www.postgresql.org/docs/current/libpq-ssl.html)](https://www.postgresql.org/docs/current/libpq-ssl.html) documentation for details.

Your database admin must configure the functionality according to the supported values described below.

Short description of the `sslmode` parameter	Value	Example of a data source name
Don't use TLS / SSL encryption against the PostgreSQL server. Default value in file `config.json`	`disable`	`postgres://mmuser:password@hostname_or_IP:5432/mattermost_test?sslmode=disable&connect_timeout=10`
The data is encrypted and the network is trusted. Default value is `sslmode` when omitted.	`require`	`postgres://mmuser:password@hostname_or_IP:5432/mattermost_test?sslmode=require&connect_timeout=10`
The data is encrypted when connecting to a trusted server.	`verify-ca`	`postgres://mmuser:password@hostname_or_IP:5432/mattermost_test?sslmode=verify-ca&connect_timeout=10`
The data is encrypted when connecting to a trusted server.	`verify-full`	`postgres://mmuser:password@hostname_or_IP:5432/mattermost_test?sslmode=verify-full&connect_timeout=10`

MySQL Databases

When Driver Name is set to mysql, we recommend using collation over using charset.

To specify collation:

```
"SqlSettings": { "DataSource": "<mmuser:password>@tcp(hostname or IP:3306)/
mattermost?charset=utf8mb4,utf8&collation=utf8mb4_general_ci", [...] }
```

If collation is omitted, the default collation, `utf8mb4_general_ci` is used:

```
"SqlSettings": { "DataSource": "<mmuser:password>@tcp(hostname or IP:3306)/
mattermost?charset=utf8mb4,utf8", [...] }
```

> Note
>
> If you're using MySQL 8.0 or later, the default collation has changed to `utf8mb4_0900_ai_ci`. See our Database Software Requirements documentation for details on MySQL 8.0 support.

To use TLS with MySQL Databases

The parameter to encrypt connection against a MySQL server is `tls`.

The library used to interact with MySQL is [Go-MySQL-Driver (https://pkg.go.dev/github.com/go-sql-driver/mysql)](https://pkg.go.dev/github.com/go-sql-driver/mysql).

For the moment, it's not possible to use all the values that you could pass to a standard MySQL Client `mysql --ssl-mode=value`. See [Connection-Encryption Option Summary (https://dev.mysql.com/doc/refman/8.0/en/connection-options.html#option_general_ssl-mode)](https://dev.mysql.com/doc/refman/8.0/en/connection-options.html#option_general_ssl-mode) documentation for a version 8.0 example.

Your database admin must configure the functionality according to supported values described below.

Short description of the `tls` parameter	Value	Example of a data source name
Don't use TLS / SSL encryption against MySQL server.	`false`	`"<mmuser:password>@tcp(hostname or IP:3306)/mattermost_test?charset=utf8mb4,utf8&writeTimeout=30s&tls=false"`
Use TLS / SSL encryption against MySQL server.	`true`	`"<mmuser:password>@tcp(hostname or IP:3306)/mattermost_test?charset=utf8mb4,utf8&writeTimeout=30s&tls=true"`
Use TLS / SSL encryption with a self-signed certificate against MySQL server.	`skip-verify`	`"<mmuser:password>@tcp(hostname or IP:3306)/mattermost_test?charset=utf8mb4,utf8&writeTimeout=30s&tls=skip-verify"`
Use TLS / SSL encryption if server advertises a possible fallback; unencrypted	`preferred`	`"<mmuser:password>@tcp(hostname or IP:3306)/mattermost_test?charset=utf8mb4,utf8&writeTimeout=30s&tls=preferred"`

Short description of the `tls` parameter	Value	Example of a data source name
if it's not advertised.		

AWS High Availablity RDS cluster deployments

For an AWS High Availability RDS cluster deployment, point this configuration setting to the write/read endpoint at the **cluster** level to benefit from the AWS failover handling. AWS takes care of promoting different database nodes to be the writer node. Mattermost doesn't need to manage this. See the high availablility database configuration documentation for details.

Maximum open connections

The maximum number of open connections to the database. Numerical input. Default is **300** for self-hosted deployments, and **100** for Cloud deployments.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.MaxOpenConns": 300,` • Environment variable: `MM_SQLSETTINGS_MAXOPENCONNS`

Query timeout

The amount of time to wait, in seconds, for a response from the database after opening a connection and sending the query. Numerical input in seconds. Default is **30** seconds.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.QueryTimeout: 30",` • Environment variable: `MM_SQLSETTINGS_QUERYTIMEOUT`

Maximum connection lifetime

Maximum lifetime for a connection to the database, in milliseconds. Use this setting to configure the maximum amount of time a connection to the database may be reused Numerical input in milliseconds. Default is **3600000** milliseconds (1 hour).	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.ConnMaxLifetimeMilliseconds: 3600000",` • Environment variable: `MM_SQLSETTINGS_CONNMAXLIFETIMEMILLISECONDS`

Maximum connection idle timeout

Maximum time a database connection can remain idle, in milliseconds. Numerical input in milliseconds. Default is **300000** (5 minutes).	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.ConnMaxIdleTimeMilliseconds: 300000",` • Environment variable: `MM_SQLSETTINGS_CONNMAXIDLETIMEMILLISECONDS`

Minimum hashtag length

Minimum number of characters in a hashtag. This value must be greater than or equal to **2**.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.MinimumHashtagLength: 3",` • Environment variable: `MM_SQLSETTINGS_MINIMUMHASHTAGLENGTH`

> Note
>
> MySQL databases must be configured to support searching strings shorter than three characters. See the MySQL documentation for details.

SQL statement logging

Executed SQL statements can be written to the log for development. • **true**: Executing SQL statements are written to the log. • **false**: **(Default)** SQL statements aren't written to the log.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.Trace: false",` • Environment variable: `MM_SQLSETTINGS_TRACE`

Recycle database connections

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Select the **Recycle Database Connections** button to manually recycle the connection pool by closing the current set of open connections to the database within 20 seconds, and then creating a new set of connections. To fail over without stopping the server, change the database line in the `config.json` file, select **Reload Configuration from Disk** via **Environment > Web Server**, then select **Recycle Database Connections**.	• System Config path: **Environment > Database** • `config.json` setting: N/A • Environment variable: N/A

Disable database search

When *enterprise-scale search </administration-guide/scale/enterprise-search>*, database search can be disabled from performing searches. • **true**: Disables the use of the database to perform searches. If another search engine isn't configured, setting this value to `true` will result in empty search results. • **false**: **(Default)** Database search isn't disabled.	• System Config path: **Environment > Database** • `config.json` setting: `".SqlSettings.DisableDatabaseSearch: false",` • Environment variable: `MM_SQLSETTINGS_DISABLEDATABASESEARCH`

Search behavior in Mattermost depends on which search engines are enabled:

- When Elasticsearch or AWS OpenSearch is enabled, Mattermost will try to use it first.
- If Elasticsearch fails or is disabled, Mattermost will attempt to use Bleve, if enabled. If this occurs, you will see the warning `Encountered error on SearchPostsInTeamForUser`.
- If these fail or are disabled, Mattermost tries to search the database directly, if this is enabled.

- If all of the above methods fail or are disabled, the search results will be empty.

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas:

- **Reduced Database Load**: When database search is enabled, every search query executed by users needs to interact with the database, leading to additional load on the database server. By disabling database search, you can avoid these queries, thereby reducing the database load.
- **Improved Response Time**: Database searches can be time-consuming, especially with large datasets. Disabling database search can result in faster response times because the system no longer spends time fetching and processing search results from the database.
- **Offloading Search to Indexing Services**: Disabling database search often means that searches are offloaded to specialized indexing services like Elasticsearch, which are optimized for search operations. These services can provide faster and more efficient search capabilities compared to traditional database searches.
- **Lower Resource Consumption**: Running search queries directly against the database can be resource-intensive (using CPU and memory). With database search disabled, these resources can be allocated to other critical functions, improving overall system performance.
- **Enhanced Scalability**: As the number of users and data volume grow, database search can become less efficient. Specialized search services are designed to scale more effectively, enhancing overall system scalability and performance.
- However, the ability to perform database searches in Mattermost is a critical feature for many users, particularly when other search engines aren't enabled. Disabling this feature will result in users seeing an error if they attempt to use the Mattermost Search box. It's important to balance performance improvements with the needs of your organization and users.

Applied schema migrations

A list of all migrations that have been applied to the data store based on the version information available in the `db_migrations` table. Select **About Mattermost** from the product menu to review the current database schema version applied to your deployment.

Active Search Backend

Read-only display of the currently active backend used for search. Values can include `none`, `database`, `elasticsearch`, or `bleve`.

Read replicas

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Specifies the connection strings for the read replica databases.	• System Config path: N/A • `config.json` setting: `".SqlSettings.DataSourceReplicas": []` • Environment variable: `MM_SQLSETTINGS_DATASOURCEREPLICAS`

> Note
>
> - Each database connection string in the array must be in the same form used for the Data source setting.
> - Space separate multiple read replicas in the array to allow Mattermost to load balance read queries across multiple database instances. For example, `MM_SQLSETTINGS_DATASOURCEREPLICAS=dc-1 dc-2`

AWS High Availability RDS cluster deployments

For an AWS High Availability RDS cluster deployment, point this configuration setting directly to the underlying read-only node endpoint within the RDS cluster to circumvent the failover/load balancing that AWS/RDS takes care of (except for the write traffic). Mattermost has its own method of balancing the read-only connections and can also balance those queries to the data source/write+read connection should those nodes fail. See the high availablility database configuration documentation for details.

Search replicas

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Specifies the connection strings for the search replica databases. A search replica is similar to a read replica, but is used only for handling search queries.	• System Config path: N/A • `config.json` setting: `".SqlSettings.DataSourceSearchReplicas": []` • Environment variable: `MM_SQLSETTINGS_DATASOURCESEARCHREPLICAS`

> Note
>
> Each database connection string in the array must be in the same form used for the Data source setting.

AWS High Availability RDS cluster deployments

For an AWS High Availability RDS cluster deployment, point this configuration setting directly to the underlying read-only node endpoint within the RDS cluster to circumvent the failover/load balancing that AWS/RDS takes care of (except for the write traffic). Mattermost has its own method of balancing the read-only connections and can also balance those queries to the data source/

write+read connection should those nodes fail. See the high availablility database configuration documentation for details.

Replica lag settings

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

String array input specifies a connection string and user-defined SQL queries on the database to measure replica lag for a single replica instance. These settings monitor absolute lag based on binlog distance/transaction queue length, and the time taken for the replica to catch up. String array input consists of: • `DataSource`: The database credentials to connect to the database instance. • `QueryAbsoluteLag`: A plain SQL query that must return a single row. The first column must be the node value of the Prometheus metric, and the second column must be the value of the lag used to measure absolute lag. • `QueryTimeLag`: A plain SQL query that must return a single row. The first column must be the node value of the Prometheus metric, and the second column must be the value of the lag used to measure the time lag.	• System Config path: N/A • `config.json` setting: `".SqlSettings.ReplicaLagSettings": []` • Environment variable: `MM_SQLSETTINGS_REPLICALAGSETTINGS`

> Note
>
> - The `QueryAbsoluteLag` and `QueryTimeLag` queries must return a single row.
> - To properly monitor this, you must set up performance monitoring for Mattermost.

1. Configure the replica lag metric based on your database type. See the following tabs for details on configuring this for each database type.

> *AWS Aurora*
>
> *Add the configuration highlighted below to your `SqlSettings.ReplicaLagSettings` array. You only need to add this once because replication statistics for AWS Aurora nodes are visible across all server instances that are members of the cluster. Be sure to change the `DataSource` to point to a single node in the group.*
>
> *For more information on Aurora replication stats, see the [AWS Aurora documentaion (https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora_global_db_instance_status.html)](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora_global_db_instance_status.html).*
>
> *Example*
>
> ```
> { "SqlSettings": { "ReplicaLagSettings": [ { "DataSource": "replica-1", "QueryAbsoluteLag": "select server_id, highest_lsn_rcvd-durable_lsn as bindiff from aurora_global_db_instance_status() where server_id=<>", "QueryTimeLag": "select server_id, visibility_lag_in_msec from aurora_global_db_instance_status() where server_id=<>" } ] } }
> ```
>
> *MySQL Group Replication*
>
> *Add the configuration highlighted below to your `SqlSettings.ReplicaLagSettings` array. You only need to add this once because replication statistics for all nodes are shared across all server instances that are members of the MySQL replication group. Be sure to change the `DataSource` to point to a single node in the group.*

> *For more information on group replication stats, see the MySQL documentation.*
>
> *Example*
>
> ```
> { "SqlSettings": { "ReplicaLagSettings": [ { "DataSource": "replica-1", "QueryAbsoluteLag": "select member_id, count_transactions_remote_in_applier_queue FROM performance_schema.replication_group_member_stats where member_id=<>", "QueryTimeLag": "" } ] } }
> ```
>
> *PostgreSQL replication slots*
>
> 1. *Add the configuration highlighted below to your `SqlSettings.ReplicaLagSettings` array. This query should run against the **primary** node in your cluster, to do this change the `DataSource` to match the SqlSettings.DataSource setting you have configured.*
>
> *For more information on pg_stat_replication, see the PostgreSQL documentation.*
>
> > ***Example:***
> >
> > ```
> > { "SqlSettings": { "ReplicaLagSettings": [ { "DataSource": "postgres://mmuser:password@localhost:5432/mattermost_test?sslmode=disable&connect_timeout=10.", "QueryAbsoluteLag": "select username, pg_wal_lsn_diff(pg_current_wal_lsn(),replay_lsn) as metric from pg_stat_replication;", "QueryTimeLag": "" } ] } }
> > ```
>
> 2. *Grant permissions to the database user for `pg_monitor`. This user should be the same user configured above in the `DataSource` string.*
>
> > *For more information on roles, see the PostgreSQL documentation.*
> >
> > ```
> > sudo -u postgres psql postgres=# GRANT pg_monitor TO mmuser;
> > ```

2. Save the config and restart all Mattermost nodes.

3. Navigate to your Grafana instance monitoring Mattermost and open the [Mattermost Performance Monitoring v2 (https://grafana.com/grafana/dashend-user-guide/boards/15582-mattermost-performance-monitoring-v2/)](https://grafana.com/grafana/dashend-user-guide/boards/15582-mattermost-performance-monitoring-v2/) dashboard.

4. The `QueryTimeLag` chart is already setup for you utilizing the existing `Replica Lag` chart. If using `QueryAbsoluteLag` metric clone the `Replica Lag` chart and edit the query to use the below absolute lag metrics and modify the title to be `Replica Lag Absolute`.

```
mattermost_db_replica_lag_abs{instance=~"$server"}
```

Replica monitor interval (seconds)

Available onall plansself-hosteddeployments

Specifies how frequently unhealthy replicas will be monitored for liveness check. Mattermost will dynamically choose a replica if it's alive. Numerical input. Default is 5 seconds.	• System Config path: N/A • `config.json` setting: `".SqlSettings.ReplicaMonitorIntervalSeconds": 5` • Environment variable: `MM_SQLSETTINGS_REPLICAMONITORINTERVALSECONDS`

Enterprise search

Available onEnterprise plansself-hosteddeployments

Core database search happens in a relational database and is intended for deployments under about 2–3 million posts and file entries. Beyond that scale, enabling enterprise search with Elasticsearch or AWS OpenSearch is highly recommended for optimum search performance before reaching 3 million posts.

For deployments with over 3 million posts, Elasticsearch or AWS OpenSearch is required to avoid significant performance issues, such as timeouts, with message searches and @mentions.

You can configure Mattermost enterprise search by going to **System Console > Environment > Elasticsearch**. The following configuration settings apply to both Elasticsearch and AWS OpenSearch. You can also edit the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Enable Elasticsearch indexing

Configure Mattermost to index new posts automatically. • **true**: Indexing of new messages occurs automatically. • **false**: **(Default)** Indexing of new messages is disabled, and new messages aren't indexed.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.EnableIndexing: false",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_ENABLEINDEXING`

> Note
>
> If indexing is disabled and then re-enabled after an index is created, purge and rebuild the index to ensure complete search results.

Backend type

Both Elasticsearch and AWS OpenSearch provide enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Learn more about enterprise search in our product documentation.

The type of search backend. • `elasticsearch` - (**Default**) • `opensearch` - Required for AWS OpenSearch.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.Backend: elasticsearch",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_BACKEND`

Learn more about enterprise search version support.

Server connection address

The address of the Elasticsearch or AWS OpenSearch server.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.ConnectionUrl",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CONNECTIONURL`

CA path

Optional path to the Custom Certificate Authority certificates for the Elasticsearch or AWS OpenSearch server.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.CA",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CA`

Note

- Available from Mattermost v7.8. The certificate path should be `/opt/mattermost/data/elasticsearch/` or `/opt/mattermost/data/opensearch` and configured in the System Console as `./elasticsearch/cert.pem` or `./opensearch/cert.pem`.
- Can be used in conjunction with basic authentication credentials or can replace them. Leave this setting blank to use the default Certificate Authority certificates for the operating system.

Client certificate path

Available from Mattermost v7.8. Can be used in conjunction with basic auth credentials or to replace them.

Optional client certificate for the connection to the Elasticsearch or AWS OpenSearch server in the PEM format.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.ClientCert",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CLIENTCERT`

Client certificate key path

Available from Mattermost v7.8. Can be used in conjunction with basic auth credentials or to replace them.

Optional key for the client certificate in the PEM format.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.ClientKey",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CLIENTKEY`

Skip TLS verification

The certificate step for TLS connections can be skipped. • **true**: Skips the certificate verification step for TLS connections. • **false**: **(Default)** Mattermost requires certificate verification.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.SkipTLSVerification: false",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_SKIPTLSVERIFICATION`

Server username

(Optional) The username to authenticate to the Elasticsearch or AWS OpenSearch server. String input.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.UserName",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_USERNAME`

Server password

(Optional) The password to authenticate to the Elasticsearch or AWS OpenSearch server. String input.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.Password",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_PASSWORD`

Enable cluster sniffing

Configure Mattermost to automatically find and connect to all data nodes in a cluster. • **true**: Sniffing finds and connects to all data nodes in your cluster automatically. • **false**: **(Default)** Cluster sniffing is disabled.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.Sniff: false",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_SNIFF`

Select the **Test Connection** button in the System Console to validate the connection between Mattermost and the Elasticsearch or AWS OpenSearch server.

Bulk indexing

Configure Mattermost to start a bulk index of all existing posts in the database, from oldest to newest.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> - Always purge indexes before bulk indexing.
> - Select the **Index Now** button in the System Console to start a bulk index of all posts, and review all index jobs in progress.
> - Elasticsearch or AWS OpenSearch is available during indexing, but search results may be incomplete until the indexing job is complete.
> - If an in-progress indexing job is canceled, the index and search results will be incomplete.

Rebuild channels index

Purge the channels index adn re-index all channels in the database, from oldest to newest.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: N/A • Environment variable: N/A

Select the **Rebuild Channels Index** button in the System Console to purge the channels index. Ensure no other indexing jobs are in progress via the **Bulk Indexing** table before starting this process. During indexing, channel auto-complete is available, but search results may be incomplete until the indexing job is complete.

Purge indexes

Purge the entire Elasticsearch index.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: N/A • Environment variable: N/A

Select the **Purge Indexes** button in the System Console to purge the index. After purging the index, create a new index by selecting the **Index Now** button.

Indexes to skip while purging

Specify index names to ignore while purging indexes. Separate multiple index names with commas. Use an asterisk (*) to match a sequence of index name characters.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `ElasticsearchSettings.IgnoredPurgeIndexes` • Environment variable: `MM_ELASTICSEARCHSETTINGS_IGNOREDPURGEINDEXES`

Enable Elasticsearch for search queries

Configure Mattermost to use Elasticsearch or AWS OpenSearch for all search queries using the latest index. • **true**: Elasticsearch or AWS OpenSearch is used for all search queries using the latest index. Search results may be incomplete until a bulk index of the existing message database is completed. • **false**: **(Default)** Database search is used for search queries.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.EnableSearching: false",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_ENABLESEARCHING`

If indexing is disabled and then re-enabled after an index is created, purge and rebuild the index to ensure complete search results.

Enable Elasticsearch for autocomplete queries

Configure Mattermost to use Elasticsearch or AWS OpenSearch for all autocompletion queries on users and channels using the latest index. • **true**: Elasticsearch or AWS OpenSearch will be used for all autocompletion queries on users and channels using the latest index. • **false**: **(Default)** Database autocomplete is used.	• System Config path: **Environment > Elasticsearch** • `config.json` setting: `".Elasticsearchsettings.EnableAutocomplete: false",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_ENABLEAUTOCOMPLETE`

Autocompletion results may be incomplete until a bulk index of the existing users and channels database is finished.

Post index replicas

The number of replicas to use for each post index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.PostIndexReplicas: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_POSTINDEXREPLICAS`

> Note
>
> - If this setting is changed, the changed configuration only applies to newly-created indexes. To apply the change to existing indexes, purge and rebuild the index after changing this setting.
> - If there are `n` data nodes, the number of replicas per shard for each index should be `n-1`.
> - If the number of nodes in an Elasticsearch or AWS OpenSearch cluster changes, this configuration setting, as well as Channel Index Replicas and User Index Replicas must also be updated accordingly.

Post index shards

The number of shards to use for each post index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.PostIndexShards: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_POSTINDEXSHARDS`

> Note
>
> If this configuration setting is changed, the changed configuration only applies to newly-created indexes. To apply the change to existing indexes, purge and rebuild the index after changing this setting.

Channel index replicas

The number of replicas to use for each channel index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.ChannelIndexReplicas: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CHANNELINDEXREPLICAS`

> Note
>
> If there are `n` data nodes, the number of replicas per shard for each index should be `n-1`. If the number of nodes in an Elasticsearch or AWS OpenSearch cluster changes, this configuration setting, as well as Post Index Replicas and User Index Replicas must also be updated accordingly.

Channel index shards

The number of shards to use for each channel index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.ChannelIndexShards: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_CHANNELINDEXSHARDS`

User index replicas

The number of replicas to use for each user index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.UserIndexReplicas: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_USERINDEXREPLICAS`

> Note
>
> If there are `n` data nodes, the number of replicas per shard for each index should be `n-1`. If the number of nodes in an Elasticsearch or AWS OpenSearch cluster changes, this configuration setting, as well as Post Index Replicas and User Index Replicas must also be updated accordingly.

User index shards

The number of shards to use for each user index. Numerical input. Default is **1**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.UserIndexShards: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_USERINDEXSHARDS`

Aggregate search indexes

Elasticsearch or AWS OpenSearch indexes older than the age specified by this setting, in days, will be aggregated during the daily scheduled job. Numerical input. Default is **365** days.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.AggregatePostsAfterDays: 365",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_AGGREGATEPOSTSAFTERDAYS`

> Note
>
> If you're using data retention and enterprise search, configure this with a value greater than your data retention policy.

Post aggregator start time

The start time of the daily scheduled aggregator job. Must be a 24-hour time stamp in the form `HH:MM` based on the local time of the server. Default is **03:00** (3 AM)	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.PostsAggregatorJobStartTime: 03:00",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_POSTSAGGREGATORJOBSTARTTIME`

Index prefix

The prefix added to the Elasticsearch or AWS OpenSearch index name.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.IndexPrefix",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_INDEXPREFIX`

> Note
>
> When this setting is used, all Elasticsearch or AWS OpenSearch indexes created by Mattermost are given this prefix. You can set different prefixes so that multiple Mattermost deployments can share an Elasticsearch or AWS OpenSearch cluster without the index names colliding.

Global search prefix

Enable global search across multiple Elasticsearch indices with the same index prefix. This is helpful for setups with multiple data centers where Elasticsearch instances share data using cross-cluster replication. It allows for easier and unified searching across distributed indices. Value must be a prefix of `IndexPrefix`.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.GlobalSearchPrefix",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_GLOBALSEARCHPREFIX`

Live indexing batch size

The number of new posts needed before those posts are added to the Elasticsearch or AWS OpenSearch index. Once added to the index, the post becomes searchable. On servers with more than 1 post per second, we suggest setting this value to the average number of posts over a 20 second period of time. Numerical input. Default is **1**. Every post is indexed synchronously as they are created.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.LiveIndexingBatchSize: 1",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_LIVEINDEXINGBATCHSIZE`

> Note
>
> It may be necessary to increase this value to avoid hitting the rate limit or resource limit of your Elasticsearch or AWS OpenSearch cluster on installs handling more than 1 post per second.

What exactly happens when I increase this value?

The primary impact is that a post will be indexed into Elasticsearch or AWS OpenSearch after the threshold of posts is met, which then makes the posts searchable within Mattermost. So, if you set this based on recommendations for larger servers, and you make a post, you cannot find it via search for ~10–20 seconds, on average. Realistically, no users should see or feel this impact due to the limited number of users who are actively **searching** for a post this

quickly. You can set this value to a lower or higher average depending on your Elasticsearch or AWS OpenSearch server specifications.

During busy periods, this delay will be faster as more traffic is occurring, causing more posts and a quicker time to hit the index number. During slower periods, expect the reverse.

How to find the right number for your server

1. You must understand how many posts your server makes every minute. Run the query below to calculate your server's average posts per minute.

   > *Note that this query can be heavy, so we recommend that you run it during non-peak hours. Additionally, you can adjust the `WHERE` clause to see the posts per minute over a different time period. Right now `31536000000` represents the number of milliseconds in a year.*
   >
   > ```
   > SELECT  AVG(postsPerMinute) as averagePostsPerMinute FROM (  SELECT
   > count(*) as postsPerMinute,  date_trunc('minute',
   > to_timestamp(createat/1000))  FROM posts  WHERE createAt > (
   > (extract(epoch from now()) * 1000 ) - 31536000000)  GROUP BY
   > date_trunc('minute', to_timestamp(createat/1000)) ) as ppm;
   > ```

2. Decide the acceptable index window for your environment, and divide your average posts per minute by that. We suggest 10-20 seconds. Assuming you have `600` posts per minute on average, and you want to index every 20 seconds (`60 seconds / 20 seconds = 3`) you would calculate `600 / 3` to come to the number `200`. After 200 posts, Mattermost will index the posts into Elasticsearch or AWS OpenSearch. So, on average, there would be a 20-second delay in searchability.

3. Edit the `config.json` or run mmctl to modify the `LiveIndexingBatchSize` setting

   > ***In the ``config.json``***
   >
   > ```
   > {  "ElasticsearchSettings": {  "LiveIndexingBatchSize": 200  } }
   > ```
   >
   > ***Via mmctl***

```
mmctl config set ElasticsearchSettings.LiveIndexingBatchSize 200
```

Via an environment variable

```
MM_ELASTICSEARCHSETTINGS_LIVEINDEXINGBATCHSIZE = 200
```

4. Restart the Mattermost server.

Batch size

The number of posts for a single batch during a bulk indexing job. Numerical input. Default is **10000**.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.BatchSize :10000",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_BATCHSIZE`

Request timeout

The timeout, in seconds, for Elasticsearch or AWS OpenSearch calls. Numerical input in seconds. Default is **30** seconds.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.RequestTimeoutSeconds :30",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_REQUESTTIMEOUTSECONDS`

Trace

Options for printing Elasticsearch or AWS OpenSearch trace errors. • **error**: Creates the error trace when initializing the Elasticsearch or AWS OpenSearch client and prints any template creation or search query that returns an error as part of the error message. • **all**: Creates the three traces (error, trace and info) for the driver and doesn't print the queries because they will be part of the trace log level of the driver. • **not specified**: **(Default)** No error trace is created.	• System Config path: N/A • `config.json` setting: `".Elasticsearchsettings.Trace",` • Environment variable: `MM_ELASTICSEARCHSETTINGS_TRACE`

File storage

Available onall plansself-hosteddeployments

Configure file storage settings by going to **System Console > Environment > File Storage**, or by editing the `config.json` file as described in the following tables.

[Mattermost Academy Learn about file storage (https://mattermost.com/pl/mattermost-academy-configure-file-storage-training)](https://mattermost.com/pl/mattermost-academy-configure-file-storage-training)

> Note
>
> Mattermost currently supports storing files on the local filesystem and Amazon S3 or S3-compatible containers. We have tested Mattermost with MinIO (https://min.io/) and Digital Ocean Spaces (https://docs.digitalocean.com/products/spaces/) products, but not all S3-compatible containers on the market. If you are looking to use other S3-compatible containers, we recommend completing your own testing.

File storage system

The type of file storage system used. Can be either Local File System or Amazon S3. • **local**: **(Default)** Files and images are stored in the specified local file directory. • **amazons3**: Files and images are stored on Amazon S3 based on the access key, bucket, and region fields provided. The driver is compatible with MinIO (Beta) and Digital Ocean Spaces.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.DriverName: local",` • Environment variable: `MM_FILESETTINGS_DRIVERNAME`

Local storage directory

The local directory to which files are written when the **File storage system** is set to **local**. Can be any directory writable by the user Mattermost is running as, and is relative to the directory where Mattermost is installed. Defaults to **./data/**.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.Directory",` • Environment variable: `MM_FILESETTINGS_DIRECTORY`

When **File storage system** is set to **amazons3**, this setting has no effect.

Maximum file size

The maximum file size for message attachments and plugin uploads. This value must be specified in mebibytes in the System Console, and in bytes in the `config.json` file. The default is `104857600` bytes (**100** mebibytes).	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.MaxFileSize: 104857600",` • Environment variable: `MM_FILESETTINGS_MAXFILESIZE`

> Note
>
> - Verify server memory can support your setting choice. Large file sizes increase the risk of server crashes and failed uploads due to network disruptions.
> - When uploading plugin files, a `Received invalid response from the server` error typically indicates that `MaxFileSize` isn't large enough to support the plugin file upload, and/or that proxy settings may not be sufficient.
> - If you use a proxy or load balancer in front of Mattermost, the following proxy settings must be adjusted accordingly:
> - For NGINX, use `client_max_body_size`.
> - For Apache, use `LimitRequestBody`.

Enable document search by content

Enable users to search the contents of documents attached to messages. • **true**: **(Default)** Documents are searchable by their content. • **false**: Documents aren't searchable by their content. When document content search is disabled, users can search for files by file name only.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.ExtractContent: true",` • Environment variable: `MM_FILESETTINGS_EXTRACTCONTENT`

Note

Document content search results for files shared before upgrading to Mattermost Server v5.35 may be incomplete until an extraction command is executed using the mmctl. If this command is not run, users can search older files based on file name only.

You can optionally install the following [dependencies (https://github.com/sajari/docconv#dependencies)](https://github.com/sajari/docconv#dependencies) to extend content searching support in Mattermost to include file formats beyond PDF, DOCX, and ODT, such as DOC, RTF, XML, and HTML:

- **tidy**: Used to search the contents of HTML documents.
- **wv**: Used to search the contents of DOC documents.
- **popplerutils**: Used to significantly improve server performance when extracting the contents of PDF documents.
- **unrtf**: Used to search the contents of RTF documents.
- **JusText**: Used to search HTML documents.

If you choose not to install these dependencies, you'll see log entries for documents that couldn't be extracted. Any documents that can't be extracted are skipped and logged so that content extraction can proceed.

Enable searching content of documents within ZIP files

Enables users to search the contents of compressed ZIP files attached to messages. • **true**: Contents of documents within ZIP files are returned in search results. This may have an impact on server performance for large files. the specified local file directory. • **false**: **(Default)** The contents of documents within ZIP files aren't returned in search results.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.ArchiveRecursion: false",` • Environment variable: `MM_FILESETTINGS_ARCHIVERECURSION`

Note

- You can search for document content within ZIP files when using Mattermost in a web browser or the desktop app.
- Searching document contents adds load to your server.
- For large deployments, or teams that share many large, text-heavy documents, we recommend you review our hardware requirements, and test enabling this feature in a staging environment before enabling it in a production environment.

Amazon S3 bucket

The name of the bucket for your S3-compatible object storage instance. A string with the S3-compatible bucket name.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3Bucket",` • Environment variable: `MM_FILESETTINGS_AMAZONS3BUCKET`

Amazon S3 path prefix

The prefix you selected for your **Amazon S3 bucket** in AWS. A string containing the path prefix.	• System Config path: N/A • `config.json` setting: `".FileSettings.AmazonS3PathPrefix",` • Environment variable: `MM_FILESETTINGS_AMAZONS3PATHPREFIX`

Amazon S3 region

The AWS region you selected when creating your **Amazon S3 bucket** in AWS. A string with the AWS region containing the bucket. If no region is set, Mattermost attempts to get the appropriate region from AWS, and sets it to **us-east-1** if none found.	• System Config path: **Environment > File Storage** • `config.json` setting: `` `".FileSettings.AmazonS3Region",` • Environment variable: `MM_FILESETTINGS_AMAZONS3REGION`

For MinIO or Digital Ocean Spaces, leave this setting empty.

Amazon S3 access key ID

A string with the access key for the S3-compatible storage instance. Your EC2 administrator can supply you with the Access Key ID.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3AccessKeyId",` • Environment variable: `MM_FILESETTINGS_AMAZONS3ACCESSKEYID`

> Note
>
> This is required for access unless you are using an [Amazon S3 IAM Role (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html) with Amazon S3.

Amazon S3 endpoint

The hostname of your S3-compatible instance. A string with the hostname of the S3-compatible storage instance. Defaults to **s3.amazonaws.com**.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3Endpoint: s3.amazonaws.com",` • Environment variable: `MM_FILESETTINGS_AMAZONS3ENDPOINT`

> Note
>
> For Digital Ocean Spaces, the hostname should be set to **<region>.digitaloceanspaces.com**, where **<region>** is the abbreviation for the region you selected when setting up the Space. It can be **nyc3**, **ams3**, or **sgp1**.

Amazon S3 secret access key

The secret access key associated with your Amazon S3 Access Key ID. A string with the secret access key for the S3-compatible storage instance.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3SecretAccessKey",` • Environment variable: `MM_FILESETTINGS_AMAZONS3SECRETACCESSKEY`

Enable secure Amazon S3 connections

Enable or disable secure Amazon S3 connections. • **true**: **(Default)** Enables only secure Amazon S3 connections. • **false**: Allows insecure connections to Amazon S3.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3SSL: true",` • Environment variable: `MM_FILESETTINGS_AMAZONS3SSL`

Amazon S3 signature v2

By default, Mattermost uses Signature v4 to sign API calls to AWS, but under some circumstances, v2 is required. • **true**: Use Signature v2 signing process. • **false**: **(Default)** Use Signature v4 signing process.	• System Config path: N/A • `config.json` setting: `".FileSettings.AmazonS3SignV2: false",` • Environment variable: `MM_FILESETTINGS_AMAZONS3SIGNV2`

See the [AWS (https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_sigv.html)](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_sigv.html) documentation for information about when to use the Signature v2 signing process.

Enable server-side encryption for Amazon S3

Enable server-side encryption for Amazon S3. • **true**: Encrypts files in Amazon S3 using server-side encryption with Amazon S3-managed keys. • **false**: **(Default)** Doesn't encrypt files in Amazon S3.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3SSE: false",` • Environment variable: `MM_FILESETTINGS_AMAZONS3SSE`

Enable Amazon S3 debugging

Enable or disable Amazon S3 debugging to capture additional debugging information in system logs. • **true**: Log additional debugging information is logged to the system logs. • **false**: **(Default)** No Amazon S3 debugging information is included in the system logs. Typically set to **false** in production.	• System Config path: **Environment > File Storage** • `config.json` setting: `".FileSettings.AmazonS3Trace: false",` • Environment variable: `MM_FILESETTINGS_AMAZONS3TRACE`

Select the **Test Connection** button in the System Console to validate the settings and ensure the user can access the server.

Amazon S3 storage class

Some Amazon S3-compatible storage solutions require the storage class parameter to be present in upload requests, otherwise they will be rejected. Configure this storage class as the storage class required by your S3-compatible solution.

The storage class to use for uploads to S3-compatible storage solutions. String input. Default is an empty string `""`. Select **Test Connection** to test the configured connection.	• System Config path: **Environment > File Storage** • `config.json` setting: `.FileSettings.AmazonS3StorageClass: ""`, • Environment variable: `MM_FILESETTINGS_AMAZONS3STORAGECLASS`

> Note
>
> Most Amazon S3-compatible storage solutions assign a default storage class of `STANDARD` when no storage class is provided. See the [Amazon S3 storage class (https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html#AmazonS3-PutObject-request-header-StorageClass)](https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html#AmazonS3-PutObject-request-header-StorageClass) documentation for details about supported storage classes.

Export Amazon S3 storage class

The storage class to use for exports to S3-compatible storage solutions. String input. Default is an empty string `""`.	• System Config path: N/A • `config.json` setting: `.FileSettings.ExportAmazonS3StorageClass: "",` • Environment variable: `MM_FILESETTINGS_EXPORTAMAZONS3STORAGECLASS`

> Note
>
> Most Amazon S3-compatible storage solutions assign a default storage class of `STANDARD` when no storage class is provided. See the [Amazon S3 storage class (https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html#AmazonS3-PutObject-request-header-StorageClass)](https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html#AmazonS3-PutObject-request-header-StorageClass) documentation for details about supported storage classes.

Amazon S3 request timeout

The amount of time, in milliseconds, before requests to Amazon S3 storage time out. Default is 30000 (30 seconds).	• System Config path: N/A • `config.json` setting: `".FileSettings.AmazonS3RequestTimeoutMilliseconds: 30000` • Environment variable: `MM_FILESETTINGS_AMAZONS3REQUESTTIMEOUTMILLISECONDS`

Amazon S3 upload part size

The size, in bytes, of each part in a multi-part upload to Amazon S3. Numeric value. Default is 5242880 (5MB).	• System Config path: N/A • `config.json` setting: `".FileSettings.AmazonS3UploadPartSizeBytes: 5242880` • Environment variable: `MM_FILESETTINGS_AMAZONS3UPLOADPARTSIZEBYTES`

> Note
>
> A smaller part size can result in more requests and an increase in latency, while a larger part size can result in more memory being allocated.

Amazon S3 exported upload part size

The size, in bytes, of each part in a multi-part exported to Amazon S3. Numeric value. Default is 104857600 (100MB).	• System Config path: N/A • `config.json` setting: `".FileSettings.ExportAmazonS3UploadPartSizeBytes: 104857600` • Environment variable: `MM_FILESETTINGS_EXPORTAMAZONS3UPLOADPARTSIZEBYTES`

> Note
>
> A smaller part size can result in more requests and an increase in latency, while a larger part size can result in more memory being allocated.

Amazon S3 request timeout

The amount of time, in milliseconds, before requests to Amazon S3 storage time out. Default is 30000 (30 seconds).	• System Config path: N/A • `config.json` setting: `".FileSettings.AmazonS3RequestTimeoutMilliseconds: 30000` • Environment variable: `MM_FILESETTINGS_AMAZONS3REQUESTTIMEOUTMILLISECONDS`

Initial font

The font used in auto-generated profile pictures with colored backgrounds and username initials. A string with the font file name. Default is **nunito-bold.ttf**.	• System Config path: N/A • `config.json` setting: `".FileSettings.InitialFont: nunito-bold.ttf",` • Environment variable: `MM_FILESETTINGS_INITIALFONT`

Image proxy

Available onall plansself-hosteddeployments

An image proxy is used by Mattermost apps to prevent them from connecting directly to remote self-hosted servers. Configure an image proxy by going to **System Console > Environment > Image Proxy**, or by editing the `config.json` file as described in the following tables.

Enable image proxy

An image proxy anonymizes Mattermost app connections and prevents them from accessing insecure content. • **true**: Enables an image proxy for loading external images. • **false**: **(Default)** Disables the image proxy.	• System Config path: **Environment > Image Proxy** • `config.json setting`: `".ImageProxySettings.Enable": true",` • Environment variable: `MM_IMAGEPROXYSETTINGS_ENABLE`

See the image proxy documentation to learn more.

Image proxy type

The type of image proxy used by Mattermost. • **local**: **(Default)** The Mattermost server itself acts as the image proxy. • **atmos/camo**: An external atmos/camo image proxy is used.	• System Config path: **Environment > Image Proxy** • `config.json setting`: `".ImageProxySettings.ImageProxyType": "local",` • Environment variable: `MM_IMAGEPROXYSETTINGS_IMAGEPROXYTYPE`

See the image proxy documentation to learn more.

Remote image proxy URL

The URL of the atmos/camo proxy. This setting isn't needed when using the **local** image proxy.	• System Config path: **Environment > Image Proxy** • `config.json setting`: `".ImageProxySettings.RemoteImageProxyURL",` • Environment variable: `MM_IMAGEPROXYSETTINGS_REMOTEIMAGEPROXYURL`

Remote image proxy options

The URL signing key passed to an atmos/camo image proxy. This setting isn't needed when using the **local** image proxy type.	• System Config path: **Environment > Image Proxy** • `config.json setting`: `".ImageProxySettings.RemoteImageProxyOptions",` • Environment variable: `MM_IMAGEPROXYSETTINGS_REMOTEIMAGEPROXYOPTIONS`

See the image proxy documentation to learn more.

SMTP

Available onall plansself-hosteddeployments

Configure SMTP email server settings by going to **System Console > Environment > SMTP**, or by editing the `config.json` file as described in the following tables.

SMTP server

The location of the SMTP email server used for email notifications.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SMTPServer",` • Environment variable: `MM_EMAILSETTINGS_SMTPSERVER`

SMTP server port

The port of SMTP email server. Numerical input.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SMTPPort",` • Environment variable: `MM_EMAILSETTINGS_SMTPPORT`

Enable SMTP authentication

Enable or disable SMTP authentication. • **true**: SMTP username and password are used for authenticating to the SMTP server. • **false**: **(Default)** Mattermost doesn't attempt to authenticate to the SMTP server.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.EnableSMTPAuth": false",` • Environment variable: `MM_EMAILSETTINGS_ENABLESMTPAUTH`

SMTP server username

The username for authenticating to the SMTP server. String input.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SMTPUsername",` • Environment variable: `MM_EMAILSETTINGS_SMTPUSERNAME`

SMTP server password

The password associated with the SMTP username. String input.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SMTPPassword",` • Environment variable: `MM_EMAILSETTINGS_SMTPPASSWORD`

SMTP connection security

Specify connection security for emails sent using SMTP. • **Not specified**: **(Default)** Send email over an unsecure connection. • **TLS**: Communication between Mattermost and your email server is encrypted. • **STARTTLS**: Attempts to upgrade an existing insecure connection to a secure connection using TLS.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.ConnectionSecurity",` • Environment variable: `MM_EMAILSETTINGS_CONNECTIONSECURITY`

Skip server certificate verification

Configure Mattermost to skip the verification of the email server certificate. • **true**: Mattermost won't verify the email server certificate. • **false**: **(Default)** Mattermost verifies the email server certificate.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SkipServerCertificateVerification": false",` • Environment variable: `MM_EMAILSETTINGS_SKIPSERVERCERTIFICATEVERIFICATION`

Enable security alerts

Enable or disable security alerts. • **true**: **(Default)** System admins are notified by email if a relevant security fix alert is announced. Requires email to be enabled. • **false**: Security alerts are disabled.	• System Config path: **Environment > SMTP** • `config.json setting`: `".ServiceSettings.EnableSecurityFixAlert": true",` • Environment variable: `MM_SERVICESETTINGS_ENABLESECURITYFIXALERT`

See the Telemetry documentation to learn more.

SMTP server timeout

The maximum amount of time, in seconds, allowed for establishing a TCP connection between Mattermost and the SMTP server. Numerical value in seconds.	• System Config path: **Environment > SMTP** • `config.json setting`: `".EmailSettings.SMTPServerTimeout",` • Environment variable: `MM_EMAILSETTINGS_SMTPSERVERTIMEOUT`

Push notification server

Available onall plansself-hosteddeployments

Configure mobile push notifications for Mattermost by going to **System Console > Environment > Push Notification Server**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Enable push notifications

Enable or disable Mattermost push notifications. • **Do not send push notifications**: Mobile push notifications are disabled. • **Use HPNS connection with uptime SLA to send notifications to iOS and Android apps**: **(Default)** Use Mattermost's hosted push notification service. • **Use TPNS connection to send notifications to iOS and Android apps**: Use Mattermost's test push notification service. • **Manually enter Push Notification Service location**: When building your own custom mobile apps, you must host your own mobile push proxy service, and specify that URL in the **Push Notification Server** field.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".EmailSettings.SendPushNotifications": true",` • Environment variable: `MM_EMAILSETTINGS_SENDPUSHNOTIFICATIONS`

Hosted Push Notifications Service (HPNS)

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Mattermost Enterprise, Professional, and Cloud customers can use Mattermost's Hosted Push Notification Service (HPNS). The HPNS offers:

- Access to a publicly-hosted Mattermost Push Notification Service (MPNS) [available on GitHub. (https://github.com/mattermost/mattermost-push-proxy)](https://github.com/mattermost/mattermost-push-proxy)

- An explicit privacy policy (https://mattermost.com/data-processing-addendum/) for the contents of unencrypted messages.
- Encrypted TLS connections:
  - Between HPNS and Apple Push Notification Services
  - Between HPNS and Google's Firebase Cloud Messaging Service
  - HPNS and your Mattermost Server
- Production-level uptime expectations.
- Out-of-box configuration for new servers means nothing is required to enable HPNS for new deployments. HPNS can be enabled for existing deployments.

> Note
>
> - The HPNS only works with pre-built apps Mattermost deploys through the Apple App Store (https://www.apple.com/app-store/) and Google Play Store (https://play.google.com/store/games?hl=en). If you build your own mobile apps, you must also host your own Mattermost push proxy server.
> - You must ensure that the push proxy can be reached on the correct port. For HPNS, it's port 443 from the Mattermost server.
> - Mattermost doesn't store any notification data. Any data being stored is at the server level only, such as the `device_id`, since the HPNS needs to know which device the notification must be sent to.

Test Push Notifications Service (TPNS)

Non-commercial and self-hosted customers can use Mattermost's free, basic Test Push Notifications Service (TPNS).

Note

- The TPNS isn't recommended for use in production environments, and doesn't offer production-level update service level agreements (SLAs).
- The TPNS isn't available for Mattermost Cloud deployments.
- The TPNS only works with the pre-built mobile apps that Mattermost deploys through the Apple App Store (https://www.apple.com/app-store/) and Google Play Store (https://play.google.com/store/games?hl=en). If you have built your own mobile apps, you must also host your own Mattermost push proxy service.
- You must ensure that the push proxy can be reached on the correct port. For TPNS, it's port 80 from the Mattermost server.
- If you don't need or want Mattermost to send mobile push notifications, disabling this configuration setting in larger deployments may improve server performance in the following areas:
  - Reduced Processing Load: Generating and sending push notifications requires processing power and resources. By disabling them, the server can allocate those resources to other tasks.
  - Decreased Network Traffic: Push notifications involve network communication. Disabling them reduces the amount of data being transferred, which can enhance overall network performance.
  - Lower Database Load: Each push notification may involve reading from and writing to the database. Reducing these operations decreases the load on the database, improving response times for other queries.
  - Faster Response Times: With fewer tasks to handle related to notifications, the system can respond faster to other requests from users, leading to a better user experience.
  - Simplified Error Handling: Push notification services can sometimes fail or have latency issues, requiring additional error handling. Disabling these notifications simplifies the system's operations.

- However, disabling push notifications can negatively impact user experience, communication efficiency, and overall productivity. It's important to balance performance improvements with the needs of your organization and users.

ID-only push notifications

Available onEnterprise plansCloudandself-hosteddeployments

Admins can enable mobile notifications to be fully private to protect a Mattermost customer against breaches in iOS and Android notification infrastructure by limiting the data sent to Apple and Google through a Mattermost configuration setting.

The standard way to send notifications to iOS and Android applications requires sending clear text messages to Apple or Google so they can be forwarded to a user's phone and displayed on iOS or Android. While Apple or Google assure the data is not collected or stored, should the organizations be breached or coerced, all standard mobile notifications on the platform could be compromised.

To avoid this risk, Mattermost can be configured to replace mobile notification text with message ID numbers that pass no information to Apple of Google. When received by the Mattermost mobile application on a user's phone, the message IDs are used to privately communicate with their Mattermost server and to retrieve mobile notification messages over an encrypted channel. This means that, at no time, is the message text visible to Apple or Google's message relay system. The contents of the message also won't reach Mattermost.

> Note
>
> Because of the extra steps to retrieve the notifications messages under Mattermost's private mobility capability with ID-only push notifications, end users may experience a slight delay before the mobile notification is fully displayed compared to sending clear text through Apple and Google's platform.

See our configuration settings documentation to learn more about the ID-only push notifications configuration setting. See our Mobile Apps FAQ documentation for details on using ID-only push notifications for data privacy.

Push notification server location

The physical location of the Mattermost Hosted Push Notification Service (HPNS) server. Select from **US (Default)** or **Germany** to automatically populate the **Push Notification Server** field server URL.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".EmailSettings.PushNotificationServer",` • Environment variable: `MM_EMAILSETTINGS_PUSHNOTIFICATIONSERVER`

Maximum notifications per channel

The maximum total number of users in a channel before @all, @here, and @channel no longer send desktop, email, or mobile push notifications to maximize performance. Numerical input. Default is **1000**.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".TeamSettings.MaxNotificationsPerChannel: 1000",` • Environment variable: `MM_EMAILSETTINGS_MAXNOTIFICATIONSPERCHANNEL`

Note

- We recommend increasing this value a little at a time, monitoring system health by tracking performance monitoring metrics, and only increasing this value if large channels have restricted permissions controlling who can post to the channel, such as a read-only channel.
- Reducing this configuration setting value to **10** in larger deployments may improve server performance in the following areas:
  - Reduced Load on Notification System: Each notification generates a certain amount of computational and network load. By limiting the number of notifications per channel, the system processes fewer notifications, thereby reducing the load on servers.
  - Database Efficiency: Notifications are typically stored in a database. Fewer notifications mean less frequent database writes and reads, leading to quicker database operations and reduced latency.
  - Minimized Client Processing: Users' clients (e.g., desktop and mobile apps) have to fetch and process notifications. With fewer notifications, clients can operate more efficiently, reducing memory and CPU usage on users' devices.
  - Improved User Experience: An overload of notifications can lead to performance lags and a cluttered experience for users. Limiting the number ensures that users receive only the most important notifications, which can enhance usability and response times.
  - Network Bandwidth: High numbers of notifications can consume a lot of bandwidth, particularly if they are being sent to many users. Fewer notifications can lead to lower overall network usage and potentially faster delivery of critical messages.
  - Server Load Balancing: By reducing the number of notifications, the workload can be more evenly distributed across the servers, leading to better load balancing and preventing any single server from becoming a bottleneck.

High availability

Available onEnterprise plansself-hosteddeployments

You can configure Mattermost as a high availability cluster-based deployment by going to **System Console > Environment > High Availability**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

In a Mattermost high availability cluster-based deployment, the System Console is set to read-only, and settings can only be changed by editing the `config.json` file directly. However, to test a high availability cluster-based environment, you can disable `ClusterSettings.ReadOnlyConfig` in the `config.json` file by setting it to `false`. This allows changes applied using the System Console to be saved back to the configuration file.

Enable high availability mode

You can enable high availability mode. • **true**: The Mattermost server will attempt inter-node communication with the other servers in the cluster that have the same cluster name. This sets the System Console to read-only mode to keep the servers' `config.json` files in sync. • **false**: **(Default)** Mattermost high availability mode is disabled.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.Enable",` • Environment variable: `MM_CLUSTERSETTINGS_ENABLE`

Cluster name

The cluster to join by name in a high availability cluster-based deployment. Only nodes with the same cluster name will join together. This is to support blue-green deployments or staging pointing to the same database.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.ClusterName",` • Environment variable: `MM_CLUSTERSETTINGS_CLUSTERNAME`

Override hostname

You can override the hostname of this server. • This property can be set to a specific IP address if needed; however, we don't recommend overriding the hostname unless it's necessary. • If left blank, Mattermost attempts to get the hostname from the operating system or uses the IP address.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.OverrideHostname",` • Environment variable: `MM_CLUSTERSETTINGS_OVERRIDEHOSTNAME`

See the high availability cluster-based deployment documentation for details.

Use IP address

You can configure your high availability cluster-based deployment to communicate using the hostname instead of the IP address. • **true**: **(Default)** The cluster attempts to communicate using the IP address specified. • **false**: The cluster attempts to communicate using the hostname.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.UseIPAddress: true",` • Environment variable: `MM_CLUSTERSETTINGS_USEIPADDRESS`

Enable experimental gossip encryption

Gossip encryption uses AES-256 by default, and this value isn't configurable by design. • **true**: **(Default for Cloud deployments)** All communication through the cluster using the gossip protocol will be encrypted. • **false**: **(Default for self-hosted deployments)** All communication using gossip protocol remains unchanged. protocol remains unencrypted.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.EnableExperimentalGossipEncryption: false",` • Environment variable: `MM_CLUSTERSETTINGS_ENABLEEXPERIMENTALGOSSIPENCRYPTION`

Note

- The Gossip protocol is based on principles outlined in the SWIM protocol developed by researchers at Cornell University (https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf). The gossip protocol is a communication mechanism in distributed systems where nodes randomly exchange information to ensure data consistency across the network. It is decentralized, scalable, and fault-tolerant, making it ideal for systems with numerous nodes. Information is spread in a manner similar to social gossip, with nodes periodically "gossiping" updates to random peers until the network converges to a consistent state. Widely used in distributed databases, blockchain networks, and peer-to-peer systems, the protocol is simple to implement and resilient to node failures. However, it can suffer from redundancy and propagation delays in large networks.
- Alternatively, you can manually set the `ClusterEncryptionKey` row value in the **Systems** table. A key is a byte array converted to base64. Set this value to either 16, 24, or 32 bytes to select AES-128, AES-192, or AES-256 respectively.

Enable gossip compression

We recommend that you disable this configuration setting for better performance. • **true**: **(Default for self-hosted deployments)** All communication through the cluster uses gossip compression. This setting is enabled by default to maintain compatibility with older servers. • **false**: **(Default for Cloud deployments)** All communication using the gossip protocol remains uncompressed.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.EnableGossipCompression: true",` • Environment variable: `MM_CLUSTERSETTINGS_ENABLEGOSSIPCOMPRESSION`

Gossip port

The port used for the gossip protocol. Both UDP and TCP should be allowed on this port. Numerical input. Default is **8074**.	• System Config path: **Environment > High Availability** • `config.json` setting: `".ClusterSettings.GossipPort: 8074",` • Environment variable: `MM_CLUSTERSETTINGS_GOSSIPPORT`

Read only config

• **true**: **(Default)** Changes made to settings in the System Console are ignored. • **false**: Changes made to settings in the System Console are written to `config.json`.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.ReadOnlyConfig: true,` • Environment variable: `MM_CLUSTERSETTINGS_READONLYCONFIG`

Network interface

An IP address used to identify the device that does automatic IP detection in high availability cluster-based deployments. String input.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.NetworkInterface: "",` • Environment variable: `MM_CLUSTERSETTINGS_NETWORKINTERFACE`

Bind address

An IP address used to bind cluster traffic to a specific network device. This setting is used primarily for servers with multiple network devices or different Bind Address and Advertise Address like in deployments that involve NAT (Network Address Translation). String input.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.BindAddress: "",` • Environment variable: `MM_CLUSTERSETTINGS_BINDADDRESS`

Advertise address

The IP address used to access the server from other nodes. This settings is used primary when cluster nodes are not in the same network and involve NAT (Network Address Translation). String input.	• System Config path: N/A • `config.json` setting: `".ClusterSettings.AdvertiseAddress: "",` • Environment variable: `MM_CLUSTERSETTINGS_ADVERTISEADDRESS`

Rate limiting

Available onall plansself-hosteddeployments

Rate limiting prevents your Mattermost server from being overloaded with too many requests, and decreases the risk and impact of third-party applications or malicious attacks on your server.

Configure rate limiting settings by going to **System Console > Environment > Rate Limiting**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Important

Mattermost rate limiting configuration settings are intended for small deployments of Mattermost up to a few hundred users, and is not intended for larger, Enterprise-scale deployments.

Enable rate limiting

Enable or disable rate limiting to throttle APIs to a specified number of requests per second. • **true**: APIs are throttled at the rate specified by the Maximum queries per second configuration setting. • **false**: **(Default)** API access isn't throttled.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.Enable: false",` • Environment variable: `MM_RATELIMITSETTINGS_ENABLE`

Maximum queries per second

Throttle the API at this number of requests per second when rate limiting is enabled. Numerical input. Default is **10**. Increase this value to accept more requests each second, and decrease this value to allow fewer requests.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.PerSec: 10,` • Environment variable: `MM_RATELIMITSETTINGS_PERSEC`

Maximum burst size

The maximum number of requests allowed beyond the per second query limit when rate limiting is enabled. Numerical input. Default is **100**. Increase this value to allow for more concurrent requests to be handled, and decrease this value to limit this capacity.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.MaxBurst: 100,` • Environment variable: `MM_RATELIMITSETTINGS_MAXBURST`

Memory store size

The maximum number of user sessions connected to the system as determined by vary rate limit settings when rate limiting is enabled. Numerical input. Default is **10000**. Typically set to the number of users in the system. We recommend setting this value to the expected number of users. A higher value may result in underutilized resources, and a lower value may result in user sessions/tokens expiring too frequently.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.MemoryStoreSize: 10000,` • Environment variable: `MM_RATELIMITSETTINGS_MEMORYSTORESIZE`

Vary rate limit by remote address

Configure Mattermost to rate limit API access by IP address when rate limiting is enabled. • **true**: **(Default)** Rate limit API access by IP address. Recommended when using a proxy. • **false**: Rate limiting does not vary by IP address.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByRemoteAddr: true,` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYREMOTEADDR`

Vary rate limit by user

Configure Mattermost to rate limit API access by authentication token or not when rate limiting is enabled. • **true**: Rate limit API access by user authentication token. Recommended when using a proxy. • **false**: **(Default)** Rate limiting does not vary by user authentication token.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByUser: false,` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYUSER`

Vary rate limit by HTTP header

Configure Mattermost to vary rate limiting API access by the HTTP header field specified. Recommended when you're using a proxy. • When configuring NGINX, set this to **X-Real-IP**. • When configuring AmazonELB, set this to **X-Forwarded-For**.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByHeader: "",` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYHEADER`

Logging

Available onall plansself-hosteddeployments

Mattermost provides 3 independent logging systems that can be configured separately with separate log files and rotation policies to meet different operational and compliance needs:

- Log Settings

- Notification Log Settings
- Audit Log Settings

By default, all Mattermost editions write logs to both the console and to the `mattermost.log` file in a machine-readable JSON format. Mattermost Enterprise and Professional customers can additionally log directly to syslog and TCP socket destination targets.

Log settings

Configure general logging by going to **System Console > Environment > Logging**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Output logs to console

Configure Mattermost to output general logs to the console. • **true**: **(Default)** Output log messages are written to the console based on the console log level configuration. The server writes messages to the standard output stream (stdout). • **false**: Output log messages aren't written to the console.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.EnableConsole": true",` • Environment variable: `MM_LOGSETTINGS_ENABLECONSOLE`

Console log level

The level of detail in general log events written when Mattermost outputs log messages to the console. • **DEBUG**: **(Default)** Outputs verbose detail for developers debugging issues. • **ERROR**: Outputs only error messages. • **INFO**: Outputs error messages and information around startup and initialization.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.ConsoleLevel": DEBUG",` • Environment variable: `MM_LOGSETTINGS_CONSOLELEVEL`

Output console logs as JSON

Configure Mattermost to output general console logs as JSON. • **true**: **(Default)** Logged events are written in a machine-readable JSON format. • **false**: Logged events are written in plain text.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.ConsoleJson": true",` • Environment variable: `MM_LOGSETTINGS_CONSOLEJSON`

Typically set to **true** in a production environment.

Colorize plain text console logs

Enables system admins to display plain text general log level details in color. • **true**: When logged events are output to the console as plain text, colorize log levels details. • **false**: **(Default)** Plain text log details aren't colorized in the console.	• System Config path: N/A • `config.json setting`: `".LogSettings.EnableColor": false",` • Environment variable: `MM_LOGSETTINGS_ENABLECOLOR`

Output logs to file

Configure Mattermost to output general console logs to a file. • **true**: **(Default)** Logged events are written based on the file log level configuration to a `mattermost.log` file located in the directory configured via `file location`. • **false**: Logged events aren't written to a file.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.EnableFile": true",` • Environment variable: `MM_LOGSETTINGS_ENABLEFILE`

> Note
>
> Typically set to **true** in a production environment. When enabled, you can download the `mattermost.log` file locally by going to **System Console > Reporting > Server Logs**, and selecting **Download Logs**.

File log directory

The location of the general log files. String input. If left blank, log files are stored in the `./logs` directory.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.FileLocation": "",` • Environment variable: `MM_LOGSETTINGS_FILELOCATION`

> Note
>
> The path you configure must exist, and Mattermost must have write permissions for this directory.

File log level

The level of detail in general log events when when Mattermost outputs log messages to a file. • **DEBUG**: Outputs verbose detail for developers debugging issues. • **ERROR**: Outputs only error messages. • **INFO**: **(Default)** Outputs error messages and information around startup and initialization.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.FileLevel": INFO",` • Environment variable: `MM_LOGSETTINGS_FILELEVEL`

Output file logs as JSON

Configure Mattermost to output general file logs as JSON. • **true**: **(Default)** Logged events are written in a machine-readable JSON format. • **false**: Logged events are written in plain text.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.FileJson": true",` • Environment variable: `MM_LOGSETTINGS_FILEJSON`

Typically set to **true** in a production environment.

Enable webhook debugging

Configure Mattermost to capture the contents of general incoming webhooks to console and/or file logs for debugging. • **true**: **(Default)** The contents of incoming webhooks are printed to log files for debugging. • **false**: The contents of incoming webhooks aren't printed to log files.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.EnableWebhookDebugging": true",` • Environment variable: `MM_LOGSETTINGS_ENABLEWEBHOOKDEBUGGING`

> Note
>
> Enable debug logs by changing the file log level to `DEBUG` to include the request body of incoming webhooks in logs.

Output logs to multiple targets

Configure Mattermost to allow any combination of console, local file, syslog, and TCP socket targets, and send general log records to multiple targets. String input can contain a filespec to another configuration file, a database DSN, or JSON.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.AdvancedLoggingJSON":: "",` • Environment variable: `MM_LOGSETTINGS_ADVANCEDLOGGINGJSON`

> Note
>
> - See the Mattermost logging documentation for details. These targets have been chosen as they support the vast majority of log aggregators, and other log analysis tools, without needing additional software installed.
> - Logs are recorded asynchronously to reduce latency to the caller.
> - Advanced logging supports hot-reloading of logger configuration.

Maximum field size

Enables system admins to limit the size of general log fields during logging. Numerical value. Default is **2048**.	• System Config path: N/A • `config.json setting`: `".LogSettings.MaxFieldSize": 2048",` • Environment variable: `MM_LOGSETTINGS_MAXFIELDSIZE`

Enable diagnostics and error reporting

Whether or not general diagnostics and error reports are sent to Mattermost, Inc. • **true**: **(Default)** Send diagnostics and error reports. • **false**: Diagnostics and error reports aren't sent.	• System Config path: **Environment > Logging** • `config.json setting`: `".LogSettings.EnableDiagnostics": "",` • Environment variable: `MM_LOGSETTINGS_ENABLEDIAGNOSTICS`

> Note
>
> See the telemetry docummentation for details on the information Mattermost collects.

Enable verbose diagnostics

Whether or not verbose general diagnostics information is sent. • **true**: Send verbose diagnostics information. • **false**: **(Default)** Verbose diagnostics information is not sent.	• System Config path: N/A • `config.json setting`: `".LogSettings.VerboseDiagnostics": false,` • Environment variable: `MM_LOGSETTINGS_VERBOSEDIAGNOSTICS`

Enable Sentry reporting

Whether or not general error reports are sent to Sentry. • **true**: **(Default)** Send error reports to Sentry. Default matches the EnableDiagnostics setting. • **false**: Error reports are not sent to Sentry.	• System Config path: N/A • `config.json setting`: `".LogSettings.EnableSentry": true,` • Environment variable: `MM_LOGSETTINGS_ENABLESENTRY`

Notification logging

Configure logging specifically for Mattermost notifications by editing the `config.json` file as described in the following tables. These settings operate independently from the main `LogSettings` and allow you to customize logging behavior specifically for the notification subsystem. Changes to these configuration settings require a server restart before taking effect.

Output logs to console

Configure Mattermost to output notification logs to the console. • **true**: **(Default)** Output log messages are written to the console based on the console log level configuration. The server writes messages to the standard output stream (stdout). • **false**: Output log messages aren't written to the console.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.EnableConsole": true",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_ENABLECONSOLE`

Console log level

The level of detail in notification log events written when Mattermost outputs log messages to the console. • **DEBUG**: **(Default)** Outputs verbose detail for developers debugging issues. • **ERROR**: Outputs only error messages. • **INFO**: Outputs error messages and information around startup and initialization.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.ConsoleLevel": DEBUG",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_CONSOLELEVEL`

Output console logs as JSON

Configure Mattermost to output notification console logs as JSON. • **true**: **(Default)** Logged events are written in a machine-readable JSON format. • **false**: Logged events are written in plain text.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.ConsoleJson": true",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_CONSOLEJSON`

Typically set to **true** in a production environment.

Colorize plain text console logs

Enables system admins to display plain text notification log level details in color. • **true**: When logged events are output to the console as plain text, colorize log levels details. • **false**: **(Default)** Plain text log details aren't colorized in the console.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.EnableColor": false",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_ENABLECOLOR`

Output logs to file

Configure Mattermost to output notification console logs to a file. • **true**: **(Default)** Logged events are written based on the file log level configuration to a `notifications.log` file located in the directory configured via `file location`. • **false**: Logged events aren't written to a file.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.EnableFile": true",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_ENABLEFILE`

File log directory

The location of the notification log files. String input. If left blank, log files are stored in the `./logs` directory.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.FileLocation": "",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_FILELOCATION`

> Note
>
> The path you configure must exist, and Mattermost must have write permissions for this directory.

File log level

The level of detail in notification log events when Mattermost outputs log messages to a file. • **DEBUG**: Outputs verbose detail for developers debugging issues. • **ERROR**: Outputs only error messages. • **INFO**: **(Default)** Outputs error messages and information around startup and initialization.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.FileLevel": INFO",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_FILELEVEL`

Output file logs as JSON

Configure Mattermost to output notification file logs as JSON. • **true**: **(Default)** Logged events are written in a machine-readable JSON format. • **false**: Logged events are written in plain text.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.FileJson": true",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_FILEJSON`

Output logs to multiple targets

Configure Mattermost to allow any combination of console, local file, syslog, and TCP socket targets, and send notification log records to multiple targets. String input can contain a filespec to another configuration file, a database DSN, or JSON.	• System Config path: N/A • `config.json setting`: `".NotificationLogSettings.AdvancedLoggingJSON"::"",` • Environment variable: `MM_NOTIFICATIONLOGSETTINGS_ADVANCEDLOGGINGJSON`

Audit logging

Configure audit logging by going to **System Console > Compliance > Audit Logging**, or by editing the `config.json` file as described in the following tables. These settings operate independently from the main `LogSettings` and allow you to customize logging behavior specifically for the audit subsystem.

Changes to these configuration settings require a server restart before taking effect.

Output audit logs to file

Whether to write audit log files to disk. • **true**: Logged events are written to the file specified by the audit file name configuration setting. • **false**: **(Default)** Audit log files aren't written.	• System Config path: **Compliance > Audit Logging** • `config.json setting`: `".ExperimentalAuditSettings.FileEnabled": false,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILEENABLED`

> Note
>
> When `FileEnabled` is set to **true**, then the audit file name must be set.

Audit file name

The name of the audit log files. The path that you set to the audit file must exist and Mattermost must have write permissions in it. **Example:** `/var/log/mattermost_audit.log`	• System Config path: **Compliance > Audit Logging** • `config.json setting`: `".ExperimentalAuditSettings.FileName": "",` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILENAME`

> Note
>
> The file name must be set to enable audit logging.

Maximum file size

The maximum size in megabytes for audit log files before they are rotated. Numerical input. Default is **100**.	• System Config path: **Compliance > Audit Logging** • `config.json` setting: `".ExperimentalAuditSettings.FileMaxSizeMB": 100,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILEMAXSIZEMB`

Maximum file age

The maximum age in days for audit log files before they are deleted. Numerical input. Default is **0** (no limit).	• System Config path: **Compliance > Audit Logging** • `config.json` setting: `".ExperimentalAuditSettings.FileMaxAgeDays": 0,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILEMAXAGEDAYS`

Maximum file backups

The maximum number of audit log file backups to retain. Numerical input. Default is **0** (no limit).	• System Config path: **Compliance > Audit Logging** • `config.json` setting: `".ExperimentalAuditSettings.FileMaxBackups": 0,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILEMAXBACKUPS`

Compress audit log files

Whether to compress rotated audit log files. • **true**: Rotated audit log files are compressed. • **false**: **(Default)** Rotated audit log files are not compressed.	• System Config path: **Compliance > Audit Logging** • `config.json setting`: `".ExperimentalAuditSettings.FileCompress": false,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILECOMPRESS`

Audit log queue size

The maximum number of audit log entries that can be queued. Numerical input. Default is **1000**.	• System Config path: **Compliance > Audit Logging** • `config.json setting`: `".ExperimentalAuditSettings.FileMaxQueueSize": 1000,` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_FILEMAXQUEUESIZE`

Audit log certificate

Certificate configuration for audit logging. String input. Default is blank.	• System Config path: N/A • `config.json setting`: `".ExperimentalAuditSettings.Certificate": "",` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_CERTIFICATE`

Output audit logs to multiple targets

Configures Mattermost to output audit log records to multiple targets.	• System Config path: **Compliance > Audit Logging** • `config.json` setting: `".ExperimentalAuditSettings.AdvancedLoggingJSON": {},` • Environment variable: `MM_EXPERIMENTALAUDITSETTINGS_ADVANCEDLOGGINGJSON`

> Note
>
> - See the Mattermost logging documentation for details on advanced logging configuration. These targets have been chosen as they support the vast majority of log aggregators, and other log analysis tools, without needing additional software installed.
> - Audit logs are recorded asynchronously to reduce latency to the caller.
> - Advanced audit logging supports hot-reloading of logger configuration.

Session lengths

Available onall plansself-hosteddeployments

User sessions are cleared when a user tries to log in, and sessions are cleared every 24 hours from the sessions database table. Configure session lengths by going to **System Console > Environment > Session Lengths**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Extend session length with activity

Improves the user experience by extending sessions and keeping users logged in if they are active in their Mattermost apps. • **true**: **(Default)** Sessions are automatically extended when users are active in their Mattermost client. User sessions only expire when users aren't active in their Mattermost client for the entire duration of the session lengths defined. • **false**: Sessions won't extend with activity in Mattermost. User sessions immediately expire at the end of the session length or based on the session idle timeout configured.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.ExtendSessionLengthWithActivity: true,` • Environment variable: `MM_SERVICESETTINGS_EXTENDSESSIONLENGTHWITHACTIVITY`

Terminate sessions on password change

Enable or disable session revocation when a user's password changes. • **true**: **(Default for new deployments)** Session revocation is enabled. All sessions of a user expire if their password is changed (by themselves or by a system admin). If the password change is initiated by the user, their current session isn't terminated. • **false**: **(Default for existing deployments)** Session revocation is disabled. When users change their password, only the user's current session is revoked. When a system admin changes the user's password, none of the user's sessions are revoked.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.TerminateSessionsOnPasswordChange: true,` • Environment variable: `MM_SERVICESETTINGS_TERMINATESESSIONSONPASSWORDCHANGE`

Session length for AD/LDAP and email

Set the number of hours counted from the last time a user entered their credentials into the web app or the desktop app to the expiry of the user's session on email and AD/LDAP authentication. Numerical input in hours. Default is **720** hours.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.SessionLengthWebInHours: 720,` • Environment variable: `MM_SERVICESETTINGS_SESSIONLENGTHWEBINHOURS`

> Note
>
> After changing this setting, the new session length takes effect after the next time the user enters their credentials.

Session length for mobile

Set the number of hours counted from the last time a user entered their credential into the mobile app to the expiry of the user's session. Numerical input in hours. Default is **720** hours.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.SessionLengthMobileInHours: 720,` • Environment variable: `MM_SERVICESETTINGS_SESSIONLENGTHMOBILEINHOURS`

> Note
>
> After changing this setting, the new session length takes effect after the next time the user enters their credentials.

Session length for SSO

Set the number of hours from the last time a user entered their SSO credentials to the expiry of the user's session. This setting defines the session length for SSO authentication, such as SAML, GitLab, and OAuth 2.0. Numerical input in hours. Default is **720** hours. Numbers as decimals are also valid values for this configuration setting.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.SessionLengthSSOInHours: 720,` • Environment variable: `MM_SERVICESETTINGS_SESSIONLENGTHSSOINHOURS`

> Note
>
> - After changing this setting, the new session length takes effect after the next time the user enters their credentials.
> - If the authentication method is SAML, GitLab, or OAuth 2.0, users may automatically be logged back in to Mattermost if they are already logged in to SAML, GitLab, or with OAuth 2.0.

Session cache

Set the number of minutes to cache a session in memory. Numerical input in minutes. Default is **10** minutes.	• System Config path: **Environment > Session Lengths** • `config.json` setting: `".ServiceSettings.SessionCacheInMinutes: 10,` • Environment variable: `MM_SERVICESETTINGS_SESSIONCACHEINMINUTES`

Session idle timeout

The number of minutes from the last time a user was active on the system to the expiry of the user's session. Once expired, the user will need to log in to continue. Numerical input in minutes. Default is **43200** (30 days). Minimum value is **5** minutes, and a value of **0** sets the time as unlimited.	• System Config path: N/A • `config.json` setting: `".ServiceSettings.SessionIdleTimeoutInMinutes: 43200,` • Environment variable: `MM_SERVICESETTINGS_SESSIONIDLETIMEOUTINMINUTES`

> Note
>
> - This setting has no effect when extend session length with activity is set to **true**.
> - This setting applies to the webapp and the desktop app. For mobile apps, use an EMM provider to lock the app when not in use. |
> - In high availability mode, enable IP hash load balancing for reliable timeout measurement.

Performance monitoring

Available onEnterprise plansself-hosteddeployments

Configure performance monitoring by going to **System Console > Environment > Performance Monitoring**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

See the performance monitoring documentation to learn more about setting up performance monitoring.

Enable performance monitoring

Enable or disable performance monitoring. • **true**: Performance monitoring data collection and profiling is enabled. • **false**: **(Default)** Mattermost performance monitoring is disabled.	• System Config path: **Environment > Performance Monitoring** • `config.json setting`: `".MetricsSettings.Enable": false",` • Environment variable: `MM_METRICSSETTINGS_ENABLE`

See the performance monitoring documentation to learn more.

Enable client performance monitoring

Enable or disable client performance monitoring. • **true**: Client performance monitoring data collection and profiling is enabled. • **false**: **(Default)** Mattermost client performance monitoring is disabled.	• System Config path: **Environment > Performance Monitoring** • `config.json setting`: `".MetricsSettings.EnableClientMetrics": false",` • Environment variable: `MM_METRICSSETTINGS_ENABLE`

Client side user ids

A list of comma-separated user IDs you want to track for client-side webapp metrics. Limited to 5 user IDs. Blank by default.	• System Config path: **Environment > Performance Monitoring** • `config.json setting`: `".MetricsSettings.ClientSideUserIds"` • Environment variable: `MM_METRICSSETTINGS_LISTENADDRESS`

> Note
>
> The total number of user IDs is limited to 5 to ensure performance. Adding more IDs can overwhelm Prometheus due to high label cardinality. To avoid performance issues, we recommend minimizing changes to this list.

Listen address

The port the Mattermost server will listen on to expose performance metrics, when enabled. Numerical input. Default is **8067**.	• System Config path: **Environment > Performance Monitoring** • `config.json setting`: `".MetricsSettings.ListenAddress": 8067",` • Environment variable: `MM_METRICSSETTINGS_LISTENADDRESS`

Developer

Available onall plansself-hosteddeployments

Configure developer mode by going to **System Console > Environment > Developer**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Enable testing commands

Enable or disable the `/test` slash command. • **true**: **(Default)** The `/test` slash command is enabled to load test accounts and test data. • **false**: The `/test` slash command is disabled.	• System Config path: **Environment > Developer** • `config.json setting`: `".ServiceSettings.EnableTesting": true",` • Environment variable: `MM_SERVICESETTINGS_ENABLETESTING`

Enable developer mode

Enable or disable developer mode. • **true**: **(Default)** Javascript errors are shown in a banner at the top of Mattermost the user interface. Not recommended for use in production. • **false**: Users are not alerted to Javascript errors.	• System Config path: **Environment > Developer** • `config.json setting`: `".ServiceSettings.EnableDeveloper": true",` • Environment variable: `MM_SERVICESETTINGS_ENABLEDEVELOPER`

Enable client debugging

Enable or disable client-side debugging settings found in **Settings > Advanced > Debugging** for individual users. • **true**: Those settings are visible and can be enabled by users. • **false**: **(Default)** Those settings are hidden and disabled.	• System Config path: **Environment > Developer** • `config.json setting`: `".ServiceSettings.EnableClientPerformanceDebugging": false",` • Environment variable: `MM_SERVICESETTINGS_ENABLECLIENTPERFORMANCEDEBUGGING`

See the client debugging documentation to learn more.

Allow untrusted internal connections

> Warning
>
> This setting is intended to prevent users located outside your local network from using the Mattermost server to request confidential data from inside your network. Care should be used when configuring this setting to prevent unintended access to your local network.

Limit the ability for the Mattermost server to make untrusted requests within its local network. A request is considered "untrusted" when it's made on behalf of a client.	• System Config path: **Environment > Developer** • `config.json setting`: `".ServiceSettings.AllowedUntrustedInternalConnections": "",` • Environment variable: `MM_SERVICESETTINGS_ALLOWEDUNTRUSTEDINTERNALCONNECTIONS`

This setting is a whitelist of local network addresses that can be requested by the Mattermost server. It's configured as a whitespace-separated list of hostnames, IP addresses, and CIDR ranges that can be accessed.

Requests that can only be configured by system admins are considered trusted and won't be affected by this setting. Trusted URLs include ones used for OAuth login or for sending push notifications.

The following features make untrusted requests and are affected by this setting:

- Integrations using webhooks, slash commands, or message actions. This prevents them from requesting endpoints within the local network.
- Link previews. When a link to a local network address is posted in a chat message, this prevents a link preview from being displayed.
- The local image proxy. If the local image proxy is enabled, images located on the local network cannot be used by integrations or posted in chat messages.

Some examples of when you may want to modify this setting include:

- When installing a plugin that includes its own images, such as [Matterpoll (https://github.com/matterpoll/matterpoll)](https://github.com/matterpoll/matterpoll), you'll need to add the Mattermost server's domain name to this list.
- When running a bot or webhook-based integration on your local network, you'll need to add the hostname of the bot/integration to this list.

- If your network is configured in such a way that publicly-accessible web pages or images are accessed by the Mattermost server using their internal IP address, the hostnames for those servers must be added to this list.

> Note
>
> - The public IP of the Mattermost application server itself is also considered a reserved IP.
> - Use whitespaces instead of commas to list the hostnames, IP addresses, or CIDR ranges. For example: `webhooks.internal.example.com`, `127.0.0.1`, or `10.0.16.0/28`.
> - IP address and domain name rules are applied before host resolution.
> - CIDR rules are applied after host resolution, and only CIDR rules require DNS resolution.
> - Mattermost attempts to match IP addresses and hostnames without even resolving. If that fails, Mattermost resolve using the local resolver (by reading the `/etc/hosts` file first), then checking for matching CIDR rules. For example, if the domain "webhooks.internal.example.com" resolves to the IP address `10.0.16.20`, a webhook with the URL `https://webhooks.internal.example.com/webhook` can be whitelisted using `webhooks.internal.example.com`, or `10.0.16.16/28`, but not `10.0.16.20`.

Mobile security

From Mattermost v10.7 and mobile app v2.27, you can configure biometric authentication, prevent Mattermost use on jailbroken or rooted devices, and can block screen captures without relying on an EMM Provider. Configure these options by going to **System Console > Environment > Mobile Security**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Enable biometric authentication

Note

Available only on [Enterprise Advanced (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Enforce biometric authentication, with PIN/passcode fallback, before accessing the app. Users will be prompted based on session activity and server switching rules. • **true**: Biometric authentication is enabled. • **false**: **(Default)** Biometric authentication is disabled.	• System Config path: **Environment > Mobile Security** • `config.json setting`: `".NativeAppSettings.MobileEnableBiometrics": false",` • Environment variable: `MM_NATIVEAPPSETTINGS_MOBILEENABLEBIOMETRICS`

> Note
>
> Users must authenticate in the following situations:
>
> - Adding a new server: When a new server is added to the mobile app and biometric authentication is enabled.
> - Opening the mobile app: At app launch when the active server requires authentication.
> - Returning after background use: After the app has been in the background for 5 minutes or more and the active server requires authentication.
> - Using multiple servers: When accessing a server for the first time, after 5 minutes of inactivity on a server, and when the last authentication attempt fails.

Enable jailbreak/root protection

Note

Available only on [Enterprise Advanced (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Prevent access to the app on devices detected as jailbroken or rooted. If a device fails the security check, users will be denied access or prompted to switch to a compliant server. • **true**: Jailbreak/Root protection is enabled. • **false**: **(Default)** Jailbreak/Root protection is disabled.	• System Config path: **Environment > Mobile Security** • `config.json setting`: `".NativeAppSettings.MobileJailbreakProtection": false",` • Environment variable: `MM_NATIVEAPPSETTINGS_MOBILEJAILBREAKPROTECTION`

> Note
>
> See the Expo SDK documentation to learn more about how checks are performed for this functionality.

Prevent screen capture

Note

Available only on [Enterprise Advanced (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Block screenshots and screen recordings when using the mobile app. Screenshots will appear blank, and screen recordings will blur (iOS) or show a black screen (Android). Also applies when switching apps. • **true**: Screen capture blocking is enabled. • **false**: **(Default)** Screen capture blocking is disabled.	• System Config path: **Environment > Mobile Security** • `config.json setting`: `".NativeAppSettings.MobilePreventScreenCapture": false",` • Environment variable: `MM_NATIVEAPPSETTINGS_MOBILEPREVENTSCREENCAPTURE`

config.json-only settings

Available onall plansself-hosteddeployments

Disable Customer Portal requests

Enable or disable customer portal requests. • **true**: **(Default)** Server-side requests made to the customer portal are disabled. • **false**: Server-side requests made to the Mattermost Customer Portal are enabled, but will always fail in air-gapped and restricted deployment environments.	• System Config path: **N/A** • `config.json setting`: `CloudSettings` > `Disable` > `false,` • Environment variable: `MM_CLOUDSETTINGS_DISABLE`

> Note
>
> Cloud admins can't modify this configuration setting.

Enable API team deletion

This setting isn't available in the System Console and can only be set in `config.json`.

True: The `api/v4/teams/{teamid}?permanent=true` API endpoint can be called by team admins and system admins (or users with appropriate permissions), or by running the mmctl team delete command to permanently delete a team.

False: The API endpoint cannot be called. Note that `api/v4/teams/{teamid}` can still be used to soft delete a team.

This feature's `config.json` setting is `"EnableAPITeamDeletion": false` with options `true` and `false`.

Enable API user deletion

This setting isn't available in the System Console and can only be set in `config.json`.

True: The `api/v4/users/{userid}?permanent=true` API endpoint can be called by system admins (or users with appropriate permissions), or by running the mmctl user delete command, to permanently delete a user.

False: The API endpoint cannot be called. Note that `api/v4/users/{userid}` can still be used to soft delete a user.

This feature's `config.json` setting is `"EnableAPIUserDeletion": false` with options `true` and `false`.

Enable API channel deletion

This setting isn't available in the System Console and can only be set in `config.json`.

True: The `api/v4/channels/{channelid}?permanent=true` API endpoint can be called by system admins (or users with appropriate permissions), or by running the mmctl channel delete command, to permanently delete a channel.

False: The API endpoint cannot be called. Note that `api/v4/channels/{channelid}` can still be used to soft delete a channel.

> This feature's `config.json` setting is `"EnableAPIChannelDeletion": false` with options `true` and `false`.

Enable desktop app developer mode

From Desktop App v5.10, this setting enables developer debugging options available by going to the **View > Developer Tools** menu in the Mattermost desktop app.

This setting isn't available in the System Console and can only be enabled in `config.json` by setting the environment variable `MM_DESKTOP_DEVELOPER_MODE` to `true`. This setting is disabled by default.

True: Unlocks the following options in the Desktop App for the purposes of troubleshooting and debugging. You should only enable this setting if instructed to by a Mattermost developer:

> - ***Browser Mode Only***: *Completely disables the preload script and stops web app components from knowing they're in the desktop app. This option should be the best indicator of whether a web app component is causing performance and/or memory retention issues. This option disables notifications, cross-tab navigation, unread/mentions badges, the calls widget, and breaks resizing on macOS.*
> - ***Disable Notification Storage***: *Turns off maps that hold references to unread notifications until they've been selected & read. This option is good for debugging in cases where Mattermost is holding onto too many references to unused notifications.*

> - ***Disable User Activity Monitor**: Turns off the interval that checks whether the user is away or not. This option is good for debugging whether a user's availability status is causing unexpected desktop app behavior.*
> - ***Disable Context Menu**: Turns off the context menu attached to the BrowserViews. This option is good as a library santity check.*
> - ***Force Legacy Messaging API**: Forces the app to revert back to the old messaging API instead of the newer contextBridge API. This option is a good santity check to confirm whether the new API is responsible for holding onto memory.*
> - ***Force New Messaging API**: Forces the app to use the contextBridge API and completely disables the legacy one. This option forces off listeners for the legacy API.*

Redis cache backend

Available onEnterprise Advanced plansself-hosteddeployments

From Mattermost v10.4, Mattermost Enterprise customers can configure Redis (https://redis.io/) (Remote Dictionary Server) as an alternative cache backend. Redis is an open-source, in-memory data structure store that can be used as a database, cache, and message broker. It supports various data structures and is a top choice for its performance because its able to store data in memory and provide very quick data access.

Using Redis as a caching solution can help ensure that Mattermost for enterprise-level deployments with high concurrency and large user bases remains performant and efficient, even under heavy usage.

Configure a Redis cache by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Cache type

Define the cache type. • **lru**: **(Default)** Mattermost uses the in-memory cache store. • **redis**: Mattermost uses the configured Redis cache store.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `CacheType,` > `lru` • Environment variable: `MM_CACHESETTINGS_CACHETYPE`

Redis address

The hostname of the Redis host. String input.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `RedisAddress,` • Environment variable: `MM_CACHESETTINGS_REDISADDRESS`

Redis password

The password of the Redis host. String input. Leave blank if there is no password.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `RedisPassword,` • Environment variable: `MM_CACHESETTINGS_REDISPASSWORD`

Redis database

The database of the Redis host. Zero-indexed number up to 15. Typically set to `0`. Redis allows a maximum of 16 databases.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `RedisDB,` • Environment variable: `MM_CACHESETTINGS_REDISDB`

Disable client cache

Disables the client-side cache of Redis. • **true**: Client-side cache of Redis is disabled. Typically used as a test option, and not in production environments. • **false**: **(Default)** Client-side cache of Redis is enabled.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `DisableClientCache,` > `false` • Environment variable: `MM_CACHESETTINGS_REDISDB`

Redis cache prefix

Adds a prefix to all Redis cache keys.	• System Config path: **N/A** • `config.json setting`: `CacheSettings` > `RedisCachePrefix` • Environment variable: `MM_CACHESETTINGS_REDISCACHEPREFIX`

> Tip
>
> Adding a prefix to all Redis cache keys reduces key collisions, simplifies debugging, isolates data, and provides a clear structure for managing and scaling Redis-based systems. In environments where multiple systems or tenants use the same Redis instance, prefixes become critical for maintaining data integrity and operational efficiency.

Enable webhub channel iteration

Control the performance of websocket broadcasting in channels. When enabled, improves websocket broadcasting performance; however, performance may decrease when users join or leave a channel. Not recommended unless you have at least 200,000 concurrent users actively using Mattermost. Disabled by default.	• System Config path: **N/A** • `config.json setting`: `ServiceSettings` > `EnableWebHubChannelIteration,` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLEWEBHUBCHANNELITERATION`

Enable dedicated export filestore target

Enables the ability to specify an alternate filestore target for Mattermost bulk exports and compliance exports.

- **True**: A new `ExportFileBackend()` is generated under `FileSettings` using new configuration values for the following configuration settings:

  - `ExportDriverName`
  - `ExportDirectory`
  - `ExportAmazonS3AccessKeyId`
  - `ExportAmazonS3SecretAccessKey`
  - `ExportAmazonS3Bucket`
  - `ExportAmazonS3PathPrefix`
  - `ExportAmazonS3Region`
  - `ExportAmazonS3Endpoint`
  - `ExportAmazonS3SSL`
  - `ExportAmazonS3SignV2`
  - `ExportAmazonS3SSE`
  - `ExportAmazonS3Trace`
  - `ExportAmazonS3RequestTimeoutMilliseconds`
  - `ExportAmazonS3PresignExpiresSeconds`

- **False**: (**Default**) Standard file storage is used. Standard file storage will also be used when the configuration setting or value is omitted.

- System Config path: **N/A**
- `config.json setting`: `FileSettings` > `DedicatedExportStore`
- Environment variable: `MM_FILESETTINGS_DEDICATEDEXPORTSTORE`

> Note
>
> - When an alternate filestore target is configured, Mattermost Cloud admins can generate an S3 presigned URL for exports using the `/exportlink [job-id|zip file|latest]` slash command. See the Mattermost data migration documentation for details. Alternatively, Cloud and self-hosted admins can use the mmctl export generate-presigned-url command to generate a presigned URL directly from mmctl.
> - Generating an S3 presigned URL requires the feature flag `EnableExportDirectDownload` to be set to `true`, the storage must be compatible with generating an S3 link, and this experimental configuration setting must be set to `true`. Presigned URLs for exports aren't supported for systems with shared storage.

On this page

- Environment configuration settings
- Environment configuration settings
  - Web server
    - Site URL
    - Maximum URL length
    - Web server listen address
    - Forward port 80 to 443
    - Web server connection security
    - TLS certificate file
    - TLS key file
    - Use Let's Encrypt
    - Let's Encrypt certificate cache file
    - Read timeout
    - Write timeout
    - Idle timeout
    - Webserver mode
    - Enable insecure outgoing connections
    - Managed resource paths

        - Reload configuration from disk
        - Purge all caches
        - Websocket URL
        - License file location
        - TLS minimum version
        - Trusted proxy IP header
        - Enable Strict Transport Security (HSTS)
        - Secure TLS transport expiry
        - TLS cipher overwrites
        - Goroutine health threshold
        - Allow cookies for subdomains
        - Cluster log timeout
        - Maximum payload size
    - Database
        - Driver name
        - Data source
            - PostgreSQL databases
            - MySQL Databases
            - AWS High Availablity RDS cluster deployments
        - Maximum open connections
        - Query timeout
        - Maximum connection lifetime
        - Maximum connection idle timeout
        - Minimum hashtag length
        - SQL statement logging
        - Recycle database connections
        - Disable database search
        - Applied schema migrations
        - Active Search Backend
        - Read replicas
            - AWS High Availability RDS cluster deployments
        - Search replicas
            - AWS High Availability RDS cluster deployments

    - Replica lag settings
    - Replica monitor interval (seconds)
  - Enterprise search
    - Enable Elasticsearch indexing
    - Backend type
    - Server connection address
    - CA path
    - Client certificate path
    - Client certificate key path
    - Skip TLS verification
    - Server username
    - Server password
    - Enable cluster sniffing
    - Bulk indexing
    - Rebuild channels index
    - Purge indexes
    - Indexes to skip while purging
    - Enable Elasticsearch for search queries
    - Enable Elasticsearch for autocomplete queries
    - Post index replicas
    - Post index shards
    - Channel index replicas
    - Channel index shards
    - User index replicas
    - User index shards
    - Aggregate search indexes
    - Post aggregator start time
    - Index prefix
    - Global search prefix
    - Live indexing batch size
    - Batch size
    - Request timeout
    - Trace

- File storage
  - File storage system
  - Local storage directory
  - Maximum file size
  - Enable document search by content
  - Enable searching content of documents within ZIP files
  - Amazon S3 bucket
  - Amazon S3 path prefix
  - Amazon S3 region
  - Amazon S3 access key ID
  - Amazon S3 endpoint
  - Amazon S3 secret access key
  - Enable secure Amazon S3 connections
  - Amazon S3 signature v2
  - Enable server-side encryption for Amazon S3
  - Enable Amazon S3 debugging
  - Amazon S3 storage class
  - Export Amazon S3 storage class
  - Amazon S3 request timeout
  - Amazon S3 upload part size
  - Amazon S3 exported upload part size
  - Amazon S3 request timeout
  - Initial font
- Image proxy
  - Enable image proxy
  - Image proxy type
  - Remote image proxy URL
  - Remote image proxy options
- SMTP
  - SMTP server
  - SMTP server port
  - Enable SMTP authentication
  - SMTP server username

    - SMTP server password
    - SMTP connection security
    - Skip server certificate verification
    - Enable security alerts
    - SMTP server timeout
- Push notification server
    - Enable push notifications
        - Hosted Push Notifications Service (HPNS)
        - Test Push Notifications Service (TPNS)
        - ID-only push notifications
    - Push notification server location
    - Maximum notifications per channel
- High availability
    - Enable high availability mode
    - Cluster name
    - Override hostname
    - Use IP address
    - Enable experimental gossip encryption
    - Enable gossip compression
    - Gossip port
    - Read only config
    - Network interface
    - Bind address
    - Advertise address
- Rate limiting
    - Enable rate limiting
    - Maximum queries per second
    - Maximum burst size
    - Memory store size
    - Vary rate limit by remote address
    - Vary rate limit by user
    - Vary rate limit by HTTP header

- Logging
  - Log settings
    - Output logs to console
    - Console log level
    - Output console logs as JSON
    - Colorize plain text console logs
    - Output logs to file
    - File log directory
    - File log level
    - Output file logs as JSON
    - Enable webhook debugging
    - Output logs to multiple targets
    - Maximum field size
    - Enable diagnostics and error reporting
    - Enable verbose diagnostics
    - Enable Sentry reporting
  - Notification logging
    - Output logs to console
    - Console log level
    - Output console logs as JSON
    - Colorize plain text console logs
    - Output logs to file
    - File log directory
    - File log level
    - Output file logs as JSON
    - Output logs to multiple targets
  - Audit logging
    - Output audit logs to file
    - Audit file name
    - Maximum file size
    - Maximum file age
    - Maximum file backups
    - Compress audit log files
    - Audit log queue size

        - Audit log certificate
        - Output audit logs to multiple targets
    - Session lengths
        - Extend session length with activity
        - Terminate sessions on password change
        - Session length for AD/LDAP and email
        - Session length for mobile
        - Session length for SSO
        - Session cache
        - Session idle timeout
    - Performance monitoring
        - Enable performance monitoring
        - Enable client performance monitoring
        - Client side user ids
        - Listen address
    - Developer
        - Enable testing commands
        - Enable developer mode
        - Enable client debugging
        - Allow untrusted internal connections
    - Mobile security
        - Enable biometric authentication
        - Enable jailbreak/root protection
        - Prevent screen capture
    - config.json-only settings
        - Disable Customer Portal requests
        - Enable API team deletion
        - Enable API user deletion
        - Enable API channel deletion
        - Enable desktop app developer mode
        - Redis cache backend
            - Cache type
            - Redis address

        - Redis password
        - Redis database
        - Disable client cache
        - Redis cache prefix
    - Enable webhub channel iteration
    - Enable dedicated export filestore target
- Web server
    - Site URL
    - Maximum URL length
    - Web server listen address
    - Forward port 80 to 443
    - Web server connection security
    - TLS certificate file
    - TLS key file
    - Use Let’s Encrypt
    - Let’s Encrypt certificate cache file
    - Read timeout
    - Write timeout
    - Idle timeout
    - Webserver mode
    - Enable insecure outgoing connections
    - Managed resource paths
    - Reload configuration from disk
    - Purge all caches
    - Websocket URL
    - License file location
    - TLS minimum version
    - Trusted proxy IP header
    - Enable Strict Transport Security (HSTS)
    - Secure TLS transport expiry
    - TLS cipher overwrites
    - Goroutine health threshold
    - Allow cookies for subdomains

    - Cluster log timeout
    - Maximum payload size
- Database
    - Driver name
    - Data source
        - PostgreSQL databases
        - MySQL Databases
        - AWS High Availablity RDS cluster deployments
    - Maximum open connections
    - Query timeout
    - Maximum connection lifetime
    - Maximum connection idle timeout
    - Minimum hashtag length
    - SQL statement logging
    - Recycle database connections
    - Disable database search
    - Applied schema migrations
    - Active Search Backend
    - Read replicas
        - AWS High Availability RDS cluster deployments
    - Search replicas
        - AWS High Availability RDS cluster deployments
    - Replica lag settings
    - Replica monitor interval (seconds)
- Enterprise search
    - Enable Elasticsearch indexing
    - Backend type
    - Server connection address
    - CA path
    - Client certificate path
    - Client certificate key path
    - Skip TLS verification
    - Server username

    - Server password
    - Enable cluster sniffing
    - Bulk indexing
    - Rebuild channels index
    - Purge indexes
    - Indexes to skip while purging
    - Enable Elasticsearch for search queries
    - Enable Elasticsearch for autocomplete queries
    - Post index replicas
    - Post index shards
    - Channel index replicas
    - Channel index shards
    - User index replicas
    - User index shards
    - Aggregate search indexes
    - Post aggregator start time
    - Index prefix
    - Global search prefix
    - Live indexing batch size
    - Batch size
    - Request timeout
    - Trace
  - File storage
    - File storage system
    - Local storage directory
    - Maximum file size
    - Enable document search by content
    - Enable searching content of documents within ZIP files
    - Amazon S3 bucket
    - Amazon S3 path prefix
    - Amazon S3 region
    - Amazon S3 access key ID
    - Amazon S3 endpoint

    - Amazon S3 secret access key
    - Enable secure Amazon S3 connections
    - Amazon S3 signature v2
    - Enable server-side encryption for Amazon S3
    - Enable Amazon S3 debugging
    - Amazon S3 storage class
    - Export Amazon S3 storage class
    - Amazon S3 request timeout
    - Amazon S3 upload part size
    - Amazon S3 exported upload part size
    - Amazon S3 request timeout
    - Initial font
  - Image proxy
    - Enable image proxy
    - Image proxy type
    - Remote image proxy URL
    - Remote image proxy options
  - SMTP
    - SMTP server
    - SMTP server port
    - Enable SMTP authentication
    - SMTP server username
    - SMTP server password
    - SMTP connection security
    - Skip server certificate verification
    - Enable security alerts
    - SMTP server timeout
  - Push notification server
    - Enable push notifications
      - Hosted Push Notifications Service (HPNS)
      - Test Push Notifications Service (TPNS)
      - ID-only push notifications
    - Push notification server location

    - Maximum notifications per channel
- High availability
    - Enable high availability mode
    - Cluster name
    - Override hostname
    - Use IP address
    - Enable experimental gossip encryption
    - Enable gossip compression
    - Gossip port
    - Read only config
    - Network interface
    - Bind address
    - Advertise address
- Rate limiting
    - Enable rate limiting
    - Maximum queries per second
    - Maximum burst size
    - Memory store size
    - Vary rate limit by remote address
    - Vary rate limit by user
    - Vary rate limit by HTTP header
- Logging
    - Log settings
        - Output logs to console
        - Console log level
        - Output console logs as JSON
        - Colorize plain text console logs
        - Output logs to file
        - File log directory
        - File log level
        - Output file logs as JSON
        - Enable webhook debugging
        - Output logs to multiple targets

        - Maximum field size
        - Enable diagnostics and error reporting
        - Enable verbose diagnostics
        - Enable Sentry reporting
    - Notification logging
        - Output logs to console
        - Console log level
        - Output console logs as JSON
        - Colorize plain text console logs
        - Output logs to file
        - File log directory
        - File log level
        - Output file logs as JSON
        - Output logs to multiple targets
    - Audit logging
        - Output audit logs to file
        - Audit file name
        - Maximum file size
        - Maximum file age
        - Maximum file backups
        - Compress audit log files
        - Audit log queue size
        - Audit log certificate
        - Output audit logs to multiple targets
- Session lengths
    - Extend session length with activity
    - Terminate sessions on password change
    - Session length for AD/LDAP and email
    - Session length for mobile
    - Session length for SSO
    - Session cache
    - Session idle timeout

- Performance monitoring
  - Enable performance monitoring
  - Enable client performance monitoring
  - Client side user ids
  - Listen address
- Developer
  - Enable testing commands
  - Enable developer mode
  - Enable client debugging
  - Allow untrusted internal connections
- Mobile security
  - Enable biometric authentication
  - Enable jailbreak/root protection
  - Prevent screen capture
- config.json-only settings
  - Disable Customer Portal requests
  - Enable API team deletion
  - Enable API user deletion
  - Enable API channel deletion
  - Enable desktop app developer mode
  - Redis cache backend
    - Cache type
    - Redis address
    - Redis password
    - Redis database
    - Disable client cache
    - Redis cache prefix
  - Enable webhub channel iteration
  - Enable dedicated export filestore target

Environment variables

You can use environment variables to manage the configuration. Environment variables override settings in `config.json`. If a change to a setting in `config.json` requires a restart for it to take effect, then changes to the corresponding environment variable also require a server restart.

The name of the environment variable for any setting can be derived from the name of that setting in `config.json`. For example, to derive the name of the Site URL setting:

1. Find the setting in `config.json`. In this case, *ServiceSettings.SiteURL*.
2. Add `MM_` to the beginning and convert all characters to uppercase and replace the `.` with `_`. For example, *MM_SERVICESETTINGS_SITEURL*.
3. The setting becomes `export MM_SERVICESETTINGS_SITEURL="http://example.com"`.

Note

- If Mattermost is run from an initialization file, environment variables can be set via `Environment=<>`, or `EnvironmentFile=<path/to/file>`. In the second case, the file specified contains the list of environment variables to set.
- From Mattermost v7.5, environment configuration parsing supports JSON for `MM_PLUGINSETTINGS_PLUGINS` and `MM_PLUGINSETTINGS_PLUGINSTATES`. This is especially helpful for Helm configuration files, provided all plugins are configured at the same time. For example, `MM_PLUGINSETTINGS_PLUGINSTATES="{\"com.mattermost.calls\":{\"Enable\":true},\"com.mattermost.nps\":{\"Enable\":true}}"`.
- When settings are configured through an environment variable, system admins can't modify them in the System Console. If a setting is configured through an environment variable, and any other changes are made in the System Console, the value stored of the environment variable will be written back to the `config.json` as that setting's value.
- For any setting that's not set in `config.json` or in environment variables, the Mattermost server uses the setting's default value as documented in the sections below on this page.

Warning

- Environment variables for Mattermost settings that are set within the active shell will take effect when migrating configuration. For more information, see the configuration in a database documentation.
- Database connection strings for the database read and search replicas need to be formatted using URL encoding (https://www.w3schools.com/tags/ref_urlencode.asp). Incorrectly formatted strings may cause some characters to terminate the string early, resulting in issues when the connection string is parsed.

Override Mattermost license file

You can use an environment variable to override any license in the database or file configuration without replacing those licenses. When starting the server, specify the license key as `MM_LICENSE` with the contents of a license file.

> Note
>
> If `MM_LICENSE` is set to a non-empty string, but the license specified is not valid, the Mattermost server will be started without a license.
>
> In a High Availability deployment, using an environment variable to override a server license only affects the individual app server and doesn't propagate to other servers in the cluster.

Load custom configuration defaults

This custom configuration applies only if the values are not already present in the current server configuration.

1. Create a JSON file that contains the custom configuration defaults. For example, `custom.json`.
2. When starting the server, point the custom defaults environment variable to the defaults file: `MM_CUSTOM_DEFAULTS_PATH=custom.json`.

On this page

- Environment variables
  - Override Mattermost license file
  - Load custom configuration defaults

Experimental configuration settings

Review and manage the following experimental configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Experimental > Features**:

- Experimental System Console configuration settings
- Experimental Bleve configuration settings
- Experimental audit logging configuration settings
- Experimental job configuration settings
- Experimental configuration settings for self-hosted deployments only

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, one `LoginButtonColor` value is under `LdapSettings`.
>
> - If using a tool such as [jq (https://stedolan.github.io/jq/)](https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.LdapSettings.LoginButtonColor'`
> - When working with the `config.json` file manually, look for an object such as `LdapSettings`, then within that object, find the key `LoginButtonColor`.

Experimental System Console configuration settings

AD/LDAP login button color

Specify the color of the AD/LDAP login button for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonColor": ""` with string input.

AD/LDAP login button border color

Note

Specify the color of the AD/LDAP login button border for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonBorderColor": ""` with string input.

AD/LDAP login button text color

Note

Specify the color of the AD/LDAP login button text for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonTextColor": ""` with string input.

Change authentication method

Note

True: Users can change their sign-in method to any that is enabled on the server, either via their Profile or the APIs.

False: Users cannot change their sign-in method, regardless of which authentication options are enabled.

This feature's `config.json` setting is `"ExperimentalEnableAuthenticationTransfer": true` with options `true` and `false`.

Link metadata timeout

Adds a configurable timeout for requests made to return link metadata. If the metadata is not returned before this timeout expires, the message will post without requiring metadata. This timeout covers the failure cases of broken URLs and bad content types on slow network connections.

This feature's `config.json` setting is `"LinkMetadataTimeoutMilliseconds": 5000``` with numerical input.

Email batching buffer size

Specify the maximum number of notifications batched into a single email.

Note

- We recommend increasing the buffer size from the default value if you see the following error in the Mattermost logs: `Email batching job's receiving buffer was full. Please increase the EmailBatchingBufferSize. Falling back to sending immediate mail.` Increasing this value will ensure emails are queued up, without impacting server performance.
- Notifications will be sent instantly if the queue of emails exceeds the email batching interval configured.

This feature's `config.json` setting is `EmailBatchingBufferSize`: 256`` with numerical input.

Email batching interval

Specify the maximum frequency, in seconds, which the batching job checks for new notifications.

Note

- We recommend decreasing the email batching interval from the default value if you see the following error in the Mattermost logs: `Email batching job's receiving buffer was full. Please increase the EmailBatchingBufferSize. Falling back to sending immediate mail.` .
- Longer batching intervals may increase performance.
- Notifications will be sent instantly if the queue of emails exceeds the email batching interval configured.

This feature's `config.json` setting is `EmailBatchingInterval": 30` with numerical input.

Email login button color

Specify the color of the email login button for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonColor": ""` with string input.

Email login button border color

Specify the color of the email login button border for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonBorderColor": ""` with string input.

Email login button text color

Specify the color of the email login button text for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonTextColor": ""` with string input.

Enable account deactivation

True: Ability for users to deactivate their own account from **Settings > Advanced > Deactivate Account**. If a user deactivates their own account, they will get an email notification confirming they were deactivated. Available only when authentication is set to use email/password. Not available when authentication uses SAML or AD/LDAP.

False: Ability for users to deactivate their own account is disabled.

This feature's `config.json` setting is `"EnableUserDeactivation": false` with options `true` and `false`.

Enable automatic replies

True: Users can enable Automatic Replies in **Settings > Notifications**. Users set a custom message that will be automatically sent in response to Direct Messages.

False: Disables the Automatic Direct Message Replies feature and hides it from **Settings**.

This feature's `config.json` setting is `"ExperimentalEnableAutomaticReplies": false` with options `true` and `false`.

Enable channel viewed websocket messages

This setting determines whether `channel_viewed WebSocket` events are sent, which synchronize unread notifications across clients and devices.

This feature's `config.json` setting is `"EnableChannelViewedMessages": true` with options `true` and `false`.

> Note
>
> Disabling this experimental configuration setting in larger deployments may improve server performance in the following areas:
>
> - Reduced Database Load: When channel_viewed events are disabled, the server no longer needs to log these events in the database. This reduces the number of write and update operations, which can be substantial in a busy server with many users frequently switching channels.
> - Decreased Network Traffic: Disabling these events means there are fewer messages sent between the server and clients. This reduction in network traffic can lower latency and improve the overall responsiveness of the server, especially for users with slower connections.
> - Lower Server CPU Usage: Processing channel_viewed events requires CPU resources to handle database transactions and network communication. Without these events, the server's CPU can be utilized more efficiently for other tasks, improving the overall performance.
> - Improved User Experience: With reduced server load and network traffic, users may experience faster loading times and a more fluid interaction with the application.
> - However, disabling this configuration setting affects some functionality, such as accurate tracking of read and unread messages in channels. It's important to balance performance improvements with the needs of your organization and users.

Enable default channel leave/join system messages

This setting determines whether team leave/join system messages are posted in the default `town-square` channel.

True: Enables leave/join system messages in the default `town-square` channel.

False: Disables leave/join messages from the default `town-square` channel. These system messages won't be added to the database either.

This feature's `config.json` setting is `"ExperimentalEnableDefaultChannelLeaveJoinMessages": true` with options `true` and `false`.

Enable hardened mode

True: Enables a hardened mode for Mattermost that makes user experience trade-offs in the interest of security.

False: Disables hardened mode.

Changes made when hardened mode is enabled:

- Failed login returns a generic error message instead of a specific message for username and password.
- If multi-factor authentication (MFA) is enabled, the route to check if a user has MFA enabled always returns true. This causes the MFA input screen to appear even if the user does not have MFA enabled. The user may enter any value to pass the screen. Note that hardened mode does not affect user experience when MFA is enforced.
- Password reset does not inform the user that they can not reset their SSO account through Mattermost and instead claims to have sent the password reset email.
- Mattermost sanitizes all 500 errors before returned to the client. Use the supplied `request_id` to match user facing errors with the server logs.
- Standard users authenticated via username and password can't use post props reserved for integrations, such as `override_username` or `override_icon_url`.

This feature's `config.json` setting is `"ExperimentalEnableHardenedMode": false` with options `true` and `false`.

Enable theme selection

Note

True: Enables the **Display > Theme** tab in **Settings** so users can select their theme.

False: Users cannot select a different theme. The **Display > Theme** tab is hidden in **Settings**.

This feature's `config.json` setting is `"EnableThemeSelection": true` with options `true` and `false`.

Allow custom themes

Note

True: Enables the **Display > Theme > Custom Theme** section in **Settings**.

False: Users cannot use a custom theme. The **Display > Theme > Custom Theme** section is hidden in **Settings**.

This feature's `config.json` setting is `"AllowCustomThemes": true` with options `true` and `false`.

Default theme

Note

Set a default theme that applies to all new users on the system.

This feature's `config.json` setting is `"DefaultTheme": "default"` with options `"default"`, `"organization"`, `"mattermostDark"`, and `"windows10"`.

Enable tutorial

True: Users are prompted with a tutorial when they open Mattermost for the first time after account creation.

False: The tutorial is disabled. Users are placed in Town Square when they open Mattermost for the first time after account creation.

This feature's `config.json` setting is `"ServiceSettings.EnableTutorial": true` with options `true` and `false`.

Enable onboarding flow

True: New Mattermost users are shown key tasks to complete as part of initial onboarding.

False: User onboarding tasks are disabled. Users are placed in Town Square when they open Mattermost for the first time after account creation.

This feature's `config.json` setting is `"ServiceSettings.EnableOnboardingFlow": true` with options `true` and `false`.

Enable user typing messages

This setting determines whether "user is typing…" messages are displayed below the message box when using Mattermost in a web browser or the desktop app.

This feature's `config.json` setting is `"EnableUserTypingMessages": true` with options `true` and `false`.

> Note
>
> Disabling this experimental configuration setting in larger deployments may improve server performance in the following areas:
>
> - Reduced Server Load: Typing events generate additional websocket traffic. Disabling them can reduce the amount of data that needs to be handled by the server, improving the overall response time and decreasing server load.
> - Lower Network Traffic: When typing events are enabled, every keystroke generates a network event. This can lead to a significant amount of network traffic, particularly in busy channels. Disabling these events reduces the amount of information transmitted over the network.
> - Client Performance: On the client side, processing typing events requires resources. By not having to handle these events, the client can be more responsive and use less memory and CPU.

User typing timeout

This setting defines how frequently "user is typing…" messages are updated, measured in milliseconds.

This feature's `config.json` setting is `"TimeBetweenUserTypingUpdatesMilliseconds": 5000` with numerical input.

User's status and profile fetching poll interval

This setting configures the number of milliseconds to wait between fetching user statuses and profiles periodically. Set to `0` to disable.

This feature's `config.json` setting is `"ExperimentalSettings.UsersStatusAndProfileFetchingPollIntervalMilliseconds": 3000` with numerical input.

> Note
>
> Decrease this configuration setting value to increase how often Mattermost checks for and retrieves updated user profile datails. Reducing this value can be particularly helpful to reduce the likelyhood of usernames being displayed in channels as **Someone** due to outdated or missing data.

Primary team

The primary team of which users on the server are members. When a primary team is set, the options to join other teams or leave the primary team are disabled.

If the team URL of the primary team is `https://example.mattermost.com/myteam/`, then set the value to `myteam` in `config.json`.

This feature's `config.json` setting is `"ExperimentalPrimaryTeam": ""` with string input.

SAML login button color

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Specify the color of the SAML login button for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonColor": ""` with string input.

SAML login button border color

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Specify the color of the SAML login button border for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonBorderColor": ""` with string input.

SAML login button text color

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Specify the color of the SAML login button text for white labeling purposes. Use a hex code with a #-sign before the code. This setting only applies to the mobile app.

This feature's `config.json` setting is `"LoginButtonTextColor": ""` with string input.

Use channel name in email notifications

True: Channel and team name appears in email notification subject lines. Useful for servers using only one team.

False: Only team name appears in email notification subject line.

This feature's `config.json` setting is `"UseChannelInEmailNotifications": false` with options `true` and `false`.

User status away timeout

This setting defines the number of seconds after which the user's status indicator changes to "Away", when they are away from Mattermost.

This feature's `config.json` setting is `"UserStatusAwayTimeout": 300` with numerical input.

Disable data refetching on browser refocus

This setting disables re-fetching of channel and channel members on browser focus.

True: Mattermost won't refetch channels and channel members when the browser regains focus. This may result in improved performance for users with many channels and channel members.

False: (Default) Mattermost will refetch channels and channel members when the browser regains focus.

This feature's `config.json` setting is `"ExperimentalSettings.DisableRefetchingOnBrowserFocus": false` with options `true` and `false`.

Disable wake up reconnect handler

This setting disables attempts to detect when the computer has woken up and refetch data.

True: Mattermost won't attempt to detect when the computer has woken up and refetch data. This might reduce the amount of regular network traffic the app is sending.

False: (**Default**) Mattermost attempts to detect when the computer has woken up and refreshes data.

This feature's `config.json` setting is `"ExperimentalSettings.DisableWakeUpReconnectHandler": false` with options `true` and `false`.

Delay channel autocomplete

This setting controls whether or not the channel link autocomplete triggers immediately when after a tilde is typed when composing a message. This setting makes the channel autocomplete, such as `~town-square`, less obtrusive for people who use tildes `~` as punctuation.

True: The autocomplete appears after the user types a tilde followed by two or more characters. For example, typing `~to` will show the autocomplete, but typing `~` will not.

False: **(Default)** The autocomplete appears immediately after the user types a tilde. For example, typing `~` will show the autocomplete.

This feature's `config.json` setting is `"ExperimentalSettings.DelayChannelAutocomplete": false` with options `true` and `false`.

YouTube referrer policy

This setting resolves issues where YouTube video previews display as unavailable.

True: The referrer policy for embedded YouTube videos is set to `strict-origin-when-cross-origin`.

False: (Default) The referrer policy is set to `no-referrer` which enhances user privacy by not disclosing the source URL, but limits the ability to track user engagement and traffic sources in analytics tools.

This feature's `config.json` setting is `"ExperimentalSettings.YoutubeReferrerPolicy": false` with options `true` and `false`.

Experimental Bleve configuration settings

Available onall plansself-hosteddeployments

Access the following configuration settings in the System Console by going to **Experimental > Bleve**, or by editing the `config.json` file as described in the following tables:

Enable Bleve indexing

True: The indexing of new posts occurs automatically. Search queries will not use bleve search until Enable Bleve for search queries is enabled.

False: The indexing of new posts does not occur automatically.

This feature's `config.json` setting is `"EnableIndexing": false` with options `true` and `false`.

Index directory

Directory path to use for storing bleve indexes.

> Tip
>
> The bleve index directory path isn't required to exist within the `mattermost` directory. When it exists outside of the `mattermost` directory, no additional steps are needed to preserve or reindex these files as part of a Mattermost upgrade. See our Upgrading Mattermost Server documentation for details.

This feature's `config.json` setting is `"IndexDir": ""` with string input.

Bulk index now

Select **Index Now** to index all users, channels, and posts in the database from oldest to newest. Bleve is available during indexing, but search results may be incomplete until the indexing job is complete.

Purge indexes

Select **Purge Index** to remove the contents of the Bleve index directory. Search results may be incomplete until a bulk index of the existing database is rebuilt.

Enable Bleve for search queries

True: Search queries will use bleve search.

False: Search queries will not use bleve search.

This feature's `config.json` setting is `"EnableSearching": false` with options `true` and `false`.

Enable Bleve for autocomplete queries

True: Autocomplete queries will use bleve search.

False: Autocomplete queries will not use bleve search.

This feature's `config.json` setting is `"EnableAutocomplete": false` with options `true` and `false`.

Experimental audit logging configuration settings

Enable the following settings to output audit events in the System Console by going to **Compliance > Audit Logging**, or in the `config.json` file.

> Note
>
> The ability to enable and configure audit logging is currently in Beta.

Advanced logging

Available only onEnterpriseplansAvailable only forClouddeployments

Output log and audit records to any combination of console, local file, syslog, and TCP socket targets for a Mattermost Cloud deployment. See the advanced logging documentation for details about logging options.

Enable audit logging

Available onEnterprise plansCloudandself-hosteddeployments

When audit logging is enabled in a self-hosted instance, you can specify size, backup interval, compression, maximium age to manage file rotation, and timestamps for audit logging, as defined below. You can specify these settings independently for audit events and AD/LDAP events.

True: Audit logging files are enabled, and audit files are written locally to a file for a self-hosted deployment.

False: Audit logging files aren't enabled, and audit logs aren't written locally to a file for a self-hosted deployment.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileEnabled": false",` with options `true` and `false`.

File name

Available onEnterprise plansself-hosteddeployments

Specify the path to the audit file for a self-hosted deployment.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileName": ""` with string input consisting of a user-defined path (e.g. `/var/log/mattermost_audit.log`).

Max file size

Available onEnterprise plansself-hosteddeployments

This is the maximum size, in megabytes, that the file can grow before triggering rotation for a self-hosted deployment. The default setting is `100`.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileMaxSizeMB": 100` with numerical input.

Max file age

Available onEnterprise plansself-hosteddeployments

This is the maximum age, in days, a file can reach before triggering rotation for a self-hosted deployment. The default value is `0`, indicating no limit on the age.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileMaxAgeDays": 0` with numerical input.

Maximum file backups

Available onEnterprise plansself-hosteddeployments

This is the maximum number of rotated files kept for a self-hosted deployment. The oldest is deleted first. The default value is `0`, indicating no limit on the number of backups.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileMaxBackups": 0` with numerical input.

File compression

Available onEnterprise plansself-hosteddeployments

When `true`, rotated files are compressed using `gzip` in a self-hosted deployment.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileCompress": false` with options `true` and `false`.

Maximum file queue

Available onEnterprise plansself-hosteddeployments

This setting determines how many audit records can be queued/buffered at any point in time when writing to a file for a self-hosted deployment. The default is `1000` records. This setting can be left as default unless you are seeing audit write failures in the server log and need to adjust the number accordingly.

> This feature's `config.json` setting is `".ExperimentalAuditSettings.FileMaxQueueSize": 1000` with numerical input.

Certificate

Cloud Enterprise customers can upload and manage a certificate for audit logging encryption on Syslog or TCP logging targets. The ability to upload a

certificate is only available when the feature flag `ExperimentalAuditSettingsSystemConsoleUI` is enabled.

Upload the certificate PEM file in the System Console by going to **System Console > Audit Log Settings > Certificate** and selecting **File/Remove Certificate**. The certificate file can be stored in the filestore or stored locally on the filesystem.

Advanced logging

Available onEnterprise plansself-hosteddeployments

Output log and audit records to any combination of console, local file, syslog, and TCP socket targets for a Mattermost self-hosted deployment. See the advanced logging documentation for details about logging options.

Experimental configuration settings for self-hosted deployments only

Note

Available only for self-hosted deployments

Access the following self-hosted configuration settings by editing the `config.json` file as described in the following tables. These configuration settings are not accessible through the System Console.

> Tip
>
> Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `SiteURL` value is under `ServiceSettings`.
>
> - If using a tool such as jq (https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.ServiceSettings.SiteURL'`
> - When working with the `config.json` file manually, look for the key `ServiceSettings`, then within that object, find the key `SiteURL`.

Allowed themes

Note

Available only on Enterprise and Professional (https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Select the themes that can be chosen by users when `EnableThemeSelection` is set to `true`.

This feature's `config.json` setting is `"AllowedThemes": []` with string array input consisting of the options `"default"`, `"organization"`, `"mattermostDark"`, and `"windows10"`, such as `["mattermostDark", "windows10"]`.

File Location

Note

Available only on Enterprise (https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Set the file location of the compliance exports. By default, they are written to the `exports` subdirectory of the configured Local Storage directory.

This feature's `config.json` setting is `"FileLocation": "export"` with string input.

Push notification buffer

This setting isn't available in the System Console and can only be set in `config.json`.

Used to control the buffer of outstanding Push Notification messages to be sent. If the number of messages exceeds that number, then the request making the Push Notification will be blocked until there's room.

This feature's `config.json` setting is `"PushNotificationBuffer": 1000"` with numerical input.

Restrict system admin

This setting isn't available in the System Console and can only be set in `config.json`.

True: **(Default for Cloud deployments)** Restricts the system admin from viewing and modifying a subset of server configuration settings from the System Console. Not recommended for use in on-prem installations. This is intended to support Mattermost Private Cloud in giving the system admin role to users but restricting certain actions only for Cloud Admins.

False: **(Default for self-host deployments)** No restrictions are applied to the system admin role.

This feature's `config.json` setting is `"RestrictSystemAdmin": "false"` with options `true` and `false`.

> Note
>
> The ability to restrict the system admin from viewing and modifying a subset of server configuration settings is currently in Beta.

Enable client-side certification

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

True: Enables client-side certification for your Mattermost server. See the documentation to learn more.

False: Client-side certification is disabled.

This feature's `config.json` setting is `"ClientSideCertEnable": false` with options `true` and `false`.

Client-side certification login method

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Used in combination with the `ClientSideCertEnable` configuration setting.

Primary: After the client side certificate is verified, user's email is retrieved from the certificate and is used to log in without a password.

Secondary: After the client side certificate is verified, user's email is retrieved from the certificate and matched against the one supplied by the user. If they match, the user logs in with regular email/password credentials.

This feature's `config.json` setting is `"ClientSideCertCheck": "secondary"` with options `"primary"` and `"secondary"`.

Export output directory

This setting isn't available in the System Console and can only be set in `config.json`.

The directory where the exported files are stored. The path is relative to the `FileSettings` directory. By default, exports are stored under `./data/export`.

> This feature's `config.json` setting under the `ExportSettings` section is `Directory: ./export` with string input.

Export retention days

This setting isn't available in the System Console and can only be set in `config.json`.

The number of days to retain the exported files before deleting them.

> This feature's `config.json` setting under the `ExportSettings` section is `RetentionDays: 30` with numerical input.

Maximum image resolution

This setting isn't available in the System Console and can only be set in `config.json`.

Maximum image resolution size for message attachments in pixels.

> This feature's `config.json` setting is `"MaxImageResolution": 33177600` with numerical input.

Maximum image decoder concurrency

This setting isn't available in the System Console and can only be set in `config.json`.

Indicates how many images can be decoded concurrently at once. The default value of `-1` configures Mattermost to automatically use the number of CPUs present.

> Note
>
> - This configuration setting affects the total memory consumption of the server. The maximum memory of a single image is dictated by `MaxImageResolution * 24 bytes`, where the default maximum image resolution value is 33MB.
> - Therefore, a good rule of thumb to follow is that `33MB * MaxImageDecoderConcurrency * 24` should be less than the total memory for the server.
> - For example, if you have a 4-core server, you should leave aside at least `33 * 4 * 24 = 3168MB` memory for image processing. Otherwise, adjust the MaxImageResolution configuration setting to adjust the amount of memory needed for image processing.

This feature's `config.json` setting is `"MaxImageDecoderConcurrency": "-1"` with numerical input.

Initial font

This setting isn't available in the System Console and can only be set in `config.json`.

Font used in auto-generated profile pics with colored backgrounds.

This feature's `config.json` setting is `"InitialFont": "luximbi.ttf"` with string input.

Amazon S3 signature v2

This setting isn't available in the System Console and can only be set in `config.json`.

By default, Mattermost uses Signature V4 to sign API calls to AWS, but under some circumstances, V2 is required. For more information about when to use V2, see https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_sigv.html.

True: Use Signature Version 2 Signing Process.

False: Use Signature Version 4 Signing Process.

This feature's `config.json` setting is `"AmazonS3SignV2": false` with options `true` and `false`.

Amazon S3 path

This setting isn't available in the System Console and can only be set in `config.json`.

Allows using the same S3 bucket for multiple deployments.

This feature's `config.json` setting is `"AmazonS3PathPrefix: ""` with string input.

GitLab scope

This setting isn't available in the System Console and can only be set in `config.json`.

Standard setting for OAuth to determine the scope of information shared with OAuth client. Not currently supported by GitLab OAuth.

This feature's `config.json` setting is `"Scope": ""` with string input.

Global relay SMTP server timeout

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

The number of seconds that can elapse before the connection attempt to the SMTP server is abandoned. The default value is 1800 seconds. This setting is currently not available in the System Console and can only be set in `config.json`.

This feature's `config.json` setting is `"GlobalRelaySettings.SMTPServerTimeout": "1800"` with numerical input.

Google scope

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Standard setting for OAuth to determine the scope of information shared with OAuth client. Recommended setting is `profile email`.

This feature's `config.json` setting is `"Scope": "profile email"` with string input.

Import input directory

This setting isn't available in the System Console and can only be set in `config.json`.

The directory where the imported files are stored. The path is relative to the `FileSettings` directory. By default, imports are stored under `./data/import`.

> This feature's `config.json` setting under the `ImportSettings` section is `Directory: ./import` with string input.

Import retention days

This setting isn't available in the System Console and can only be set in `config.json`.

The number of days to retain the imported files before deleting them.

> This feature's `config.json` setting under the `ImportSettings` section is `RetentionDays: 30` with numerical input.

Export from timestamp

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Set the Unix timestamp (seconds since epoch, UTC) to export data from.

> This feature's `config.json` setting is `"ExportFromTimestamp": 0` with numerical input.

Block profile rate

This setting isn't available in the System Console and can only be set in `config.json`. Changes to this setting require a server restart before taking effect.

Value that controls the [fraction of goroutine blocking events reported in the blocking profile (https://pkg.go.dev/runtime#SetBlockProfileRate)](https://pkg.go.dev/runtime#SetBlockProfileRate).

The profiler aims to sample an average of one blocking event per rate nanoseconds spent blocked.

To include every blocking event in the profile, set the rate to `1`. To turn off profiling entirely, set the rate to `0`.

> This feature's `config.json` setting is `"BlockProfileRate": 0` with options `0` and `1`.

Entra ID Scope

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Standard setting for OAuth to determine the scope of information shared with OAuth client. Recommended setting is `User.Read`.

> This feature's `config.json` setting is `"Scope": "User.Read"` with string input.

Enable plugin uploads

This setting isn't available in the System Console and can only be set in `config.json`.

True: Enables plugin uploads by system admins at **Plugins > Management**. If you do not plan to upload a plugin, set to `false` to control which plugins are installed on your server. See documentation to learn more.

False: Disables plugin uploads on your Mattermost server.

> This feature's `config.json` setting is `"EnableUploads": false` with options `true` and `false`.

Allow insecure download URL

This setting isn't available in the System Console and can only be set in `config.json`.

True: Enables downloading and installing a plugin from a remote URL.

False: Disables downloading and installing a plugin from a remote URL.

> This feature's `config.json` setting is `"AllowInsecureDownloadUrl": false` with options `true` and `false`.

Enable plugin health check

This setting isn't available in the System Console and can only be set in `config.json`.

True: Enables plugin health check to ensure all plugins are periodically monitored, and restarted or deactivated based on their health status. The health check runs every 30 seconds. If the plugin is detected to fail 3 times within an hour, the Mattermost server attempts to restart it. If the restart fails 3 successive times, it's automatically disabled.

False: Disables plugin health check on your Mattermost server.

> This feature's `config.json` setting is `"EnableHealthCheck": true` with options `true` and `false`.

Plugin directory

This setting isn't available in the System Console and can only be set in `config.json`.

The location of the plugin files. If blank, they are stored in the `./plugins` directory. The path that you set must exist and Mattermost must have write permissions in it.

> This feature's `config.json` setting is `"Directory": "./plugins"` with string input.

Client plugin directory

This setting isn't available in the System Console and can only be set in `config.json`.

The location of client plugin files. If blank, they are stored in the `./client/plugins` directory. The path that you set must exist and Mattermost must have write permissions in it.

> This feature's `config.json` setting is `"ClientDirectory": "./client/plugins"` with string input.

Scoping IDP provider ID

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Allows an authenticated user to skip the initial login page of their federated Azure AD server, and only require a password to log in.

This feature's `config.json` setting is `"ScopingIDPProviderId": ""` with string input.

Scoping IDP provider name

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting isn't available in the System Console and can only be set in `config.json`.

Adds the name associated with a user's Scoping Identity Provider ID.

This feature's `config.json` setting is `"ScopingIDPName": ""` with string input.

Group unread channels

This setting isn't available in the System Console and can only be set in `config.json`.

This setting applies to the new sidebar only. You must disable the Enable Legacy Sidebar configuration setting to see and enable this functionality in the System Console.

Default Off: Disables the unread channels sidebar section for all users by default. Users can enable it in **Settings > Sidebar > Group unread channels separately**.

Default On: Enables the unread channels sidebar section for all users by default. Users can disable it in **Settings > Sidebar > Group unread channels separately**.

This feature's `config.json` setting is `"ExperimentalGroupUnreadChannels": "default_off"` with options `"default_off"` and `"default_on"`.

Enable channel category sorting

Available onall plansClouddeployments

From Mattermost v10.10, when this experimental feature is enabled, users can assign channels to new or existing channel categories when creating or renaming channels. This configuration setting applies only to cloud-based deployments.

True: Users can assign channels to new or existing channel categories when creating or renaming channels.

False: **(Default)** Disables the ability to automatically assign channels to new or existing channel categories.

This feature's `config.json` setting is `"ExperimentalEnableChannelCategorySorting": false` with options `true` and `false`.

Strict CSRF token enforcement

This setting isn't available in the System Console and can only be set in `config.json`.

True: Enables CSRF protection tokens for additional hardening compared to the currently used custom header. When the user logs in, an additional cookie is created with the CSRF token contained.

False: Disables CSRF protection tokens.

This feature's `config.json` setting is `"ExperimentalStrictCSRFEnforcement": false` with options `true` and `false`.

Developer flags

This setting isn't available in the System Console and can only be set in `config.json`.

This configuration setting specifies a list of strings where each string is a flag used to set the content security policy (CSP) for the Mattermost Web App. Each flag must be in the format `flag=true` (e.g. `unsafe-eval=true,unsafe-inline=true`). Not recommended for production environments.

The following values are currently supported:

- `unsafe-eval`: Adds the `unsafe-eval` CSP directive to the root webapp, allowing increased debugging in developer environments.
- `unsafe-inline`: Adds the `unsafe-inline` CSP directive to the root webapp, allowing increased debugging in developer environments.

This configuration setting is disabled by default and requires developer mode to be enabled.

> This feature's `config.json` setting is `"DeveloperFlags": ""` with string input.

Enable post search

This setting isn't available in the System Console and can only be set in `config.json`.

If this setting is enabled, users can search for messages in their Mattermost instance.

> This feature's `config.json` setting is `"EnablePostSearch": true` with options `true` and `false`.

> Note
>
> Disabling this experimental configuration setting in larger deployments may improve server performance in the following areas:
>
> - Reduced Database Load: When post search is enabled, every search query adds additional load to the database. Disabling search reduces these queries, leading to better database performance and lower response times for other operations.
> - Lower Memory Usage: Search functionality often requires indexing of posts, which consumes memory. By disabling search, the memory required for maintaining these indexes is freed up for other uses, improving overall system performance.
> - Faster Write Operations: When post search is enabled, indexing has to be updated with every new post, edit, or deletion. Disabling search avoids this overhead, allowing for faster write operations.
> - Performance Consistency: Without the search feature, the application avoids potential performance bottlenecks and can maintain more consistent performance levels, especially under heavy usage scenarios with a high number of posts.
> - Simplified System Maintenance: Managing search indexes can be complex and resource-intensive. Disabling search simplifies this aspect of system maintenance, potentially reducing the risk of performance issues related to search index corruption or degradation.
> - However, the ability to search messages in Mattermost is a critical feature for many users, and disabling this feature will result in users seeing an error if they attempt to use the Mattermost Search box. It's important to balance performance improvements with the needs of your organization and users.

Enable file search

This setting isn't available in the System Console and can only be set in `config.json`.

Important

This experimental configuration setting enables users to search documents attached to messages by filename. To enable users to search documents by their content, you must also enable the `ExtractContent` configuration setting. See our Enable Document Search by Content documentation for details. Document content search is available in Mattermost Server from v5.35, with mobile support coming soon.

True: Supported document types are searchable by their filename.

False: File-based searches are disabled.

This feature's `config.json` setting is `"EnableFileSearch": true` with options `true` and `false`.

Note

Disabling this experimental configuration setting in larger deployments may improve server performance in the following areas:

- Less Data to Index: Indexing files in addition to messages adds to the amount of data that needs to be processed and stored in the search index.
- Fewer Complex Queries: Searching through files can require more complex queries, which can be more resource-intensive and time-consuming as compared to searching through messages alone.
- Lower IO Operations: Searching files can generate more input/output operations, impacting the overall disk performance, especially if the system handles a large volume of file uploads and searches.
- However, the ability to search for files in Mattermost is a critical feature for many users, and disabling this feature will result in users seeing an error if they attempt to use the Mattermost Search box. It's important to balance performance improvements with the needs of your organization and users.

Enable user status updates

This setting isn't available in the System Console and can only be set in `config.json`.

Turn status updates off to improve performance. When status updates are off, users appear online only for brief periods when posting a message, and only to members of the channel in which the message is posted.

This feature's `config.json` setting is `"EnableUserStatuses": true` with options `true` and `false`.

Websocket secure port

This setting isn't available in the System Console and can only be set in `config.json`. Changes to this setting require a server restart before taking effect.

(Optional) This setting defines the port on which the secured WebSocket is listening using the `wss` protocol. Defaults to `443`. When the client attempts to make a WebSocket connection it first checks to see if the page is loaded with HTTPS. If so, it will use the secure WebSocket connection. If not, it will use the unsecure WebSocket connection. IT IS HIGHLY RECOMMENDED PRODUCTION DEPLOYMENTS ONLY OPERATE UNDER HTTPS AND WSS.

This feature's `config.json` setting is `"WebsocketSecurePort": 443` with numerical input.

> Note
>
> This is a client only override that doesn't affect the listening port of the server process which is controlled by the Web server listen address setting.

Websocket port

This setting isn't available in the System Console and can only be set in `config.json`. Changes to this setting require a server restart before taking effect.

(Optional) This setting defines the port on which the unsecured WebSocket is listening using the `ws` protocol. Defaults to `80`. When the client attempts to make a WebSocket connection it first checks to see if the page is loaded with HTTPS. If so, it will use the secure WebSocket connection. If not, it will use the unsecure WebSocket connection. IT IS HIGHLY RECOMMENDED PRODUCTION DEPLOYMENTS ONLY OPERATE UNDER HTTPS AND WSS.

This feature's `config.json` setting is `WebsocketPort": 80` with numerical input.

> Note
>
> This is a client only override that doesn't affect the listening port of the server process which is controlled by the Web server listen address setting.

Enable local mode for mmctl

This setting isn't available in the System Console and can only be set in `config.json`.

True: Enables local mode for mmctl.

False: Prevents local mode for mmctl.

This feature's `config.json` setting is `"EnableLocalMode": false` with options `true` and `false`.

> Tip
>
> When trying to use local mode with mmctl, ensure you're using the same user when running the server and mmctl, or clean up the socket file before switching to a new user. If you encounter an error like `socket file "/var/tmp/mattermost_local.socket" doesn't exists, please check the server configuration for local mode`, this can be resolved by setting this configuration setting to `true`.

Enable local mode socket location

This setting isn't available in the System Console and can only be set in `config.json`.

The path for the socket that the server will create for mmctl to connect and communicate through local mode. If the default value for this key is changed, you will need to point mmctl to the new socket path when in local mode, using the `--local-socket-path /new/path/to/socket` flag in addition to the `--local` flag.

If nothing is specified, the default path that both the server and mmctl assumes is `/var/tmp/mattermost_local.socket`.

This feature's `config.json` setting is `"LocalModeSocketLocation": "/var/tmp/mattermost_local.socket"` with string input.

Default channels

This setting isn't available in the System Console and can only be set in `config.json`.

Default channels every user is added to automatically after joining a new team. Only applies to Public channels, but affects all teams on the server.

When not set, every user is added to the `town-square` channel by default.

> Note
>
> Even if `town-square` isn't listed, every user is added to that channels automatically when joining a new team.

This feature's `config.json` setting is `"ExperimentalDefaultChannels": []` with string array input consisting of channel names, such as `["announcement", "developers"]`.

Experimental job configuration settings

Available onall plansself-hosteddeployments

Settings to configure how Mattermost schedules and completes periodic tasks such as the deletion of old posts with Data Retention enabled or indexing posts with Elasticsearch. These settings control which Mattermost servers are designated as a Scheduler, a server that queues the tasks at the correct times, and as a Worker, a server that completes the given tasks.

When running Mattermost on a single machine, both `RunJobs` and `RunScheduler` should be enabled. Without both of these enabled, Mattermost will not function properly.

When running Mattermost in High Availability mode, `RunJobs` should be enabled on one or more servers while `RunScheduler` should be enabled on all servers under normal circumstances. A High Availability cluster-based deployment will have one Scheduler and one or more Workers. See the below sections for more information.

Run jobs

This setting isn't available in the System Console and can only be set in `config.json`.

Set whether or not this Mattermost server will handle tasks created by the Scheduler. When running Mattermost on a single machine, this setting should always be enabled.

When running Mattermost in High Availablity mode, one or more servers should have this setting enabled. We recommend that your High Availability cluster-based deployment has one or more dedicated Workers with this setting enabled while the remaining Mattermost app servers have it disabled.

This feature's `config.json` setting is `"RunJobs": true` with options `true` and `false`.

Run scheduler

This setting isn't available in the System Console and can only be set in `config.json`.

Set whether or not this Mattermost server will schedule tasks that will be completed by a Worker. When running Mattermost on a single machine, this setting should always be enabled.

When running Mattermost in High Availablity mode, this setting should always be enabled. In a High Availability cluster-based deployment, exactly one of the servers will be designated as the Scheduler at a time to ensure that duplicate tasks aren't created. See High Availability documentation for more details.

> Warning
>
> We strongly recommend that you not change this setting from the default setting of `true` as this prevents the `ClusterLeader` from being able to run the scheduler. As a result, recurring jobs such as LDAP sync, Compliance Export, and data retention will no longer be scheduled. In previous Mattermost Server versions, and this documentation, the instructions stated to run the Job Server with `RunScheduler: false`. The cluster design has evolved and this is no longer the case.

This feature's `config.json` setting is `"RunScheduler": true` with options `true` and `false`.

Clean up old database jobs

This setting isn't available in the System Console and can only be set in `config.json`.

Defines the threshold in days beyond which older completed database jobs are removed. This setting is disabled by default, and must be set to a value greater than or equal to `0` to be enabled.

This feature's `config.json` setting is `"JobSettings.CleanupJobsThresholdDays": -1` with numerical input.

Clean up outdated database entries

This setting only applies to configuration in the database. It isn't available in the System Console and can be set via mmctl or changed in the database.

Defines the threshold in days beyond which outdated configurations are removed from the database.

This feature's `config.json` setting is `"JobSettings.CleanupConfigThresholdDays": 30` with numerical input.

On this page

- Experimental configuration settings
  - Experimental System Console configuration settings
    - AD/LDAP login button color
    - AD/LDAP login button border color
    - AD/LDAP login button text color
    - Change authentication method

    - Link metadata timeout
    - Email batching buffer size
    - Email batching interval
    - Email login button color
    - Email login button border color
    - Email login button text color
    - Enable account deactivation
    - Enable automatic replies
    - Enable channel viewed websocket messages
    - Enable default channel leave/join system messages
    - Enable hardened mode
    - Enable theme selection
    - Allow custom themes
    - Default theme
    - Enable tutorial
    - Enable onboarding flow
    - Enable user typing messages
    - User typing timeout
    - User's status and profile fetching poll interval
    - Primary team
    - SAML login button color
    - SAML login button border color
    - SAML login button text color
    - Use channel name in email notifications
    - User status away timeout
    - Disable data refetching on browser refocus
    - Disable wake up reconnect handler
    - Delay channel autocomplete
    - YouTube referrer policy
  - Experimental Bleve configuration settings
    - Enable Bleve indexing
    - Index directory
    - Bulk index now

    - Purge indexes
    - Enable Bleve for search queries
    - Enable Bleve for autocomplete queries
  - Experimental audit logging configuration settings
    - Advanced logging
    - Enable audit logging
    - File name
    - Max file size
    - Max file age
    - Maximum file backups
    - File compression
    - Maximum file queue
    - Certificate
    - Advanced logging
  - Experimental configuration settings for self-hosted deployments only
    - Allowed themes
    - File Location
    - Push notification buffer
    - Restrict system admin
    - Enable client-side certification
    - Client-side certification login method
    - Export output directory
    - Export retention days
    - Maximum image resolution
    - Maximum image decoder concurrency
    - Initial font
    - Amazon S3 signature v2
    - Amazon S3 path
    - GitLab scope
    - Global relay SMTP server timeout
    - Google scope
    - Import input directory
    - Import retention days

    - Export from timestamp
    - Block profile rate
    - Entra ID Scope
    - Enable plugin uploads
    - Allow insecure download URL
    - Enable plugin health check
    - Plugin directory
    - Client plugin directory
    - Scoping IDP provider ID
    - Scoping IDP provider name
    - Group unread channels
    - Enable channel category sorting
    - Strict CSRF token enforcement
    - Developer flags
    - Enable post search
    - Enable file search
    - Enable user status updates
    - Websocket secure port
    - Websocket port
    - Enable local mode for mmctl
    - Enable local mode socket location
    - Default channels
  - Experimental job configuration settings
    - Run jobs
    - Run scheduler
    - Clean up old database jobs
    - Clean up outdated database entries

Install Mattermost Boards

Mattermost Boards is not enabled by default for new Mattermost Enterprise instances. If the Mattermost Boards plugin isn't enabled for your Mattermost workspace, self-hosted Enterprise customers can install this plugin by manually uploading the binary release file into the System Console and enabling it.

Mattermost Cloud Enterprise customers can request the Boards plugins be enabled by contacting their Mattermost Account Manager or by creating a support ticket (https://support.mattermost.com/hc/en-us/requests/new?ticket_form_id=11184911962004) request.

> Note
>
> The Mattermost Boards plugin is currently in maintenance mode. While adoption of this plugin is still supported for Enterprise customers, we recommend system admins proceed at their own risk. Mattermost is no longer actively developing new features or future improvements for the Boards plugin, beyond addressing bug fixes reported by Enterprise customers and any necessary security patches.
>
> In the future, Mattermost plans to deliver a core Boards feature as part of the platform. Migration from the Boards plugin to this upcoming feature will be required, which could potentially involve downtime or data migration complexities. Keep this in mind when deploying the Boards plugin in production environments.

Install the plugin

A Mattermost system admin must perform the following steps in Mattermost.

1. Download the latest Mattermost Boards plugin binary file from the Mattermost Boards Releases (https://github.com/mattermost/mattermost-plugin-end-user-guide/project-management/releases) page.

2. Log in to your Mattermost System Console as a system admin.
3. Go to **Plugins > Plugin Management**.
4. Select **Upload Plugin** to upload the Boards plugin binary file you downloaded from the Mattermost Releases page.

Mattermost configuration

1. Go to **Plugins > Mattermost Boards**.
2. Select **Enable Plugin** to enable the plugin.
3. You can optionally enable **Enable Publicly-Shared Boards** so that board editors can share boards with people outside of your Mattermost instance, who can view the board without needing a Mattermost account.
4. Select **Save** to save your changes.

Access

See the navigate boards documentation to learn how to create and access boards.

Work with boards

See the project and task management end user documentation to learn how to align, define, organize, track, and manage work across teams.

On this page

- Install Mattermost Boards
  - Install the plugin
  - Mattermost configuration
  - Access
  - Work with boards

Integrations configuration settings

Review and manage the following integration configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Integrations**:

- Integrations management
- Bot Accounts
- GIF
- CORS

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `EnableIncomingWebhooks` value is under `ServiceSettings`.
>
> - If using a tool such as [jq (https://stedolan.github.io/jq/)](https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.ServiceSettings.EnableIncomingWebhooks'`
> - When working with the `config.json` file manually, look for an object such as `ServiceSettings`, then within that object, find the key `EnableIncomingWebhooks`.

Integrations management

Access the following configuration settings in the System Console by going to **Integrations > Integration Management**.

Enable incoming webhooks

Developers building integrations can create webhook URLs for public channels and private channels. See the incoming webhooks (https://developers.mattermost.com/integrate/webhooks/incoming/) developer documentation to learn about creating webhooks, viewing samples, and letting community know about integrations you've built.

True: Incoming webhooks are allowed. To manage incoming webhooks, select **Integrations** from the Mattermost Product menu. The webhook URLs created can be used by external applications to create posts in any public or private channels that you have access to.

False: The **Integrations > Incoming Webhooks** section of the Mattermost Product menu is hidden and all incoming webhooks are disabled.

Important

Security note: By enabling this feature, users may be able to perform phishing attacks (https://en.wikipedia.org/wiki/Phishing) by attempting to impersonate other users. To combat these attacks, a BOT tag appears next to all posts from a webhook. Enable at your own risk.

This feature's `config.json` setting is `"EnableIncomingWebhooks": true` with options `true` and `false`.

Enable outgoing webhooks

Developers building integrations can create webhook tokens for public channels. Trigger words are used to fire new message events to external integrations. For security reasons, outgoing webhooks are only available in public channels. See the outgoing webhooks (https://developers.mattermost.com/integrate/webhooks/outgoing/) developer documentation to learn about creating webhooks and viewing samples.

True: Outgoing webhooks will be allowed. To manage outgoing webhooks, select **Integrations** from the Mattermost Product menu.

False: The **Integrations > Outgoing Webhooks** of the Mattermost Product menu is hidden and all outgoing webhooks are disabled.

Important

Security note: By enabling this feature, users may be able to perform [phishing attacks (https://en.wikipedia.org/wiki/Phishing)](https://en.wikipedia.org/wiki/Phishing) by attempting to impersonate other users. To combat these attacks, a BOT tag appears next to all posts from a webhook. Enable at your own risk.

This feature's `config.json` setting is `"EnableOutgoingWebhooks": true` with options `true` and `false`.

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas:

- Reduced Network Traffic: Outgoing webhooks generate network requests to external services. Disabling them reduces the amount of traffic and resource usage related to those requests.
- Decreased Load on Server: Handling webhook events and managing connections to external services uses server resources. By disabling outgoing webhooks, the server workload is reduced, allowing it to allocate more resources to other important tasks.
- Improved Response Time: When outgoing webhooks are enabled, the server waits for the external services to return responses, potentially slowing down the performance if the external services are slow or unresponsive. Disabling them removes this dependency, leading to faster response times for user requests.
- Lower Memory Usage: Webhooks need memory to process and store data about the requests and responses. Disabling them can free up memory which can be used to improve overall server performance.
- Simplified Error Handling: Managing errors and retries for outgoing webhook failures can add complexity and overhead. Disabling outgoing webhooks can simplify error handling and reduce the processing overhead associated with it.
- However, outgoing webhooks are often essential for integrating Mattermost with other services and workflows. It’s important to balance performance improvements with the needs of your organization and users.

Enable custom slash commands

Slash commands send events to external integrations that send a response back to Mattermost.

True: Allow users to create custom slash commands from **Main Menu > Integrations > Commands**.

False: Slash commands are hidden in the **Integrations** user interface.

This feature's `config.json` setting is `"EnableCommands": false` with options `true` and `false`.

Enable OAuth 2.0 service provider

True: Mattermost acts as an OAuth 2.0 service provider allowing Mattermost to authorize API requests from external applications.

False: Mattermost does not function as an OAuth 2.0 service provider.

This feature's `config.json` setting is `"EnableOAuthServiceProvider": true` with options `true` and `false`.

> Note
>
> Cloud admins can't modify this configuration setting.

Integration request timeout

The number of seconds to wait for external integration HTTP requests, before timing out, including custom slash commands (https://developers.mattermost.com/integrate/slash-commands/custom/), outgoing webhooks (https://developers.mattermost.com/integrate/webhooks/outgoing/), interactive messages (https://developers.mattermost.com/integrate/plugins/interactive-messages/), and interactive dialogs (https://developers.mattermost.com/integrate/plugins/interactive-dialogs/). Increase this value if you have external integrations that can take some time to generate an HTTP response, or experience delayed responses due to latency.

This feature's `config.json` setting is `"OutgoingIntegrationRequestsDefaultTimeout": 3`.

Enable integrations to override usernames

True: Webhooks, slash commands, OAuth 2.0 apps, and other integrations, will be allowed to change the username they are posting as. If no username is present, the username for the post is the same as it would be for a setting of `False`.

False: **(Default)** Custom slash commands can only post as the username of the user who used the slash command. OAuth 2.0 apps can only post as the username of the user who set up the integration. For incoming webhooks and outgoing webhooks, the username is "webhook". See https://developers.mattermost.com/integrate/other-integrations/ for more details.

This feature's `config.json` setting is `"EnablePostUsernameOverride": false` with options `true` and `false`.

Enable integrations to override profile picture icons

True: Webhooks, slash commands, and other integrations, will be allowed to change the profile picture they post with.

False: **(Default)** Webhooks, slash commands, and OAuth 2.0 apps can only post with the profile picture of the account they were set up with. See https://developers.mattermost.com/integrate/other-integrations/ for more details.

This feature's `config.json` setting is `"EnablePostIconOverride": false` with options `true` and `false`.

Enable personal access tokens

True: Users can create [personal access tokens (https://developers.mattermost.com/integrate/admin-guide/admin-personal-access-token/)](https://developers.mattermost.com/integrate/admin-guide/admin-personal-access-token/) for integrations in **Profile > Security**. They can be used to authenticate against the API and give full access to the account.

To manage who can create personal access tokens or to search users by token ID, go to the **System Console > Users** page.

False: Personal access tokens are disabled on the server.

This feature's `config.json` setting is `"EnableUserAccessTokens": false` with options `true` and `false`.

Bot accounts

Access the following configuration settings in the System Console by going to **Integrations > Bot Accounts**.

Enable bot account creation

True: **(Default for Cloud deployments)** Users can create bot accounts for integrations in **Integrations > Bot Accounts**. Bot accounts are similar to user accounts except they cannot be used to log in. See documentation to learn more.

False: **(Default for self-hosted deployments)** Bot accounts cannot be created through the user interface or the RESTful API. Plugins can still create and manage bot accounts.

This feature's `config.json` setting is `"EnableBotAccountCreation": false` with options `true` and `false`.

Disable bot accounts when owner is deactivated

True: When a user is deactivated, disables all bot accounts managed by the user. To re-enable bot accounts, go to **Integrations > Bot Accounts**.

False: When a user is deactivated, all bot accounts managed by the user remain active.

> This feature's `config.json` setting is `"DisableBotsWhenOwnerIsDeactivated": false` with options `true` and `false`.

GIF

Access the following configuration settings in the System Console by going to **Integrations > GIF**.

Enable GIF picker

True: Allow users to select GIFs from the emoji picker via a GIPHY integration.

False: GIFs cannot be selected in the emoji picker.

> This feature's `config.json` setting is `"EnableGifPicker": true` with options `true` and `false`.

Important

Link previews must be enabled in order to display GIF link previews. Mattermost deployments restricted to access behind a firewall must open port 443 (for all request types) for this feature to work.

CORS

Available onall plansself-hosteddeployments

Access the following configuration settings in the System Console by going to **Integrations > CORS**.

Enable cross-origin requests from

Enable HTTP cross-origin requests from specific domains.

- Type `*` to allow CORS from any domain.
- Enter a specific domain or multiple domains separated by spaces.
- Type `null` to prevent CORS from any domain.
- Leave blank to disable it and use the Mattermost **Site URL** instead.

> Note
>
> Ensure you've entered your Site URL before enabling this setting to prevent losing access to the System Console after saving. If you lose access to the System Console after changing this setting, you can set your Site URL through the `config.json` file.

This feature's `config.json` setting is `"AllowCorsFrom": ""` with string input.

CORS exposed headers

Whitelist of headers that will be accessible to the requester.

This feature's `config.json` setting is `"CorsExposedHeaders": ""` with string input.

CORS allow credentials

True: Requests that pass validation will include the `Access-Control-Allow-Credentials` header.

False: Requests won't include the `Access-Control-Allow-Credentials` header.

This feature's `config.json` setting is `"CorsAllowCredentials": false` with options `true` and `false`.

CORS debug

True: Prints messages to the logs to help when developing an integration that uses CORS. These messages will include the structured key value pair `"source": "cors"`.

False: Debug messages not printed to the logs.

This feature's `config.json` setting is `"CorsDebug": false` with options `true` and `false`.

Embedding

Available onall plansself-hosteddeployments

Access the following configuration settings in the System Console by going to **Integrations > Embedding**.

Frame ancestors

Enter a space-separated list of domains that are allowed to embed the Mattermost web client via an iFrame. Leave blank to disallow embedding. Leave blank to disable embedding. Blank by default.

This feature's `config.json` setting is `"FrameAncestors".`

> Note
>
> Embedding Mattermost via an iFrame can provide seamless integration for collaboration into an organization's existing tools and workflows. However, you must ensure that correct configurations are in place to allow communication between the iframe and the parent domain without violating security.

On this page

- Integrations configuration settings
  - Integrations management
    - Enable incoming webhooks
    - Enable outgoing webhooks
    - Enable custom slash commands
    - Enable OAuth 2.0 service provider
    - Integration request timeout
    - Enable integrations to override usernames
    - Enable integrations to override profile picture icons
    - Enable personal access tokens
  - Bot accounts
    - Enable bot account creation
    - Disable bot accounts when owner is deactivated
  - GIF
    - Enable GIF picker
  - CORS
    - Enable cross-origin requests from
    - CORS exposed headers
    - CORS allow credentials
    - CORS debug
  - Embedding
    - Frame ancestors

Manage user surveys

In your self-hosted Mattermost deployment, use the Mattermost User Survey integration to gather direct feedback from your Mattermost users to identify what's working well and what's not with your Mattermost instance. All user responses are stored in and remain within your self-hosted deployment, and no telemetry data is sent back to Mattermost. You can export a CSV report of NPS scores and user responses for further analysis, or to share your feedback data with Mattermost.

You can schedule when each survey begins, define how long each survey lasts, specify teams excluded from the survey, as well as customize both a welcome message and a user question you want feedback on.

Setup

You must be a Mattermost system admin to upload the plugin to your Mattermost self-hosted deployment enable it, create surveys, and export survey responses using the System Console.

The User Survey integration is compatible with the following Mattermost Server versions:

> - *v9.11.2 (Extended Support Release)*
> - *v9.8 or later*
> - *v9.5.2+ (Extended Support Release)*
> - *v8.1 (Extended Support Release)*

Install

Important

Mattermost offers an additonal User Satisfaction Surveys option with limited customization options within the **Plugins** list where surveys are enabled by

default. We recommend disabling the User Satisfaction Surveys functionality when using this user survey integration.

1. Log in to your Mattermost workspace as a system administrator.
2. In Mattermost, from the Product menu

   , select **App Marketplace**.
3. Search for or scroll to User Survey, and select **Install**.
4. Once installed, select **Configure**. You're taken to the System Console, directly to the **User Survey** integration page, under **Plugins**.

> Note
>
> From Mattermost v9.11.2 (ESR) and Mattermost Cloud v10, this plugin is pre-packaged with the Mattermost Server. If your Mattermost deployment is on a release prior to v9.11.2, download the latest plugin binary release (https://github.com/mattermost/mattermost-plugin-user-survey/releases), and upload it to your server via **System Console > Plugin Management**.

Upgrade

We recommend updating this integration as new versions are released. Generally, updates are seamless and don't interrupt the user experience in Mattermost.

Visit the Releases page (https://github.com/mattermost/mattermost-plugin-user-survey/releases) for the latest release, available releases, and compatibiilty considerations.

Enable

Go to **System Console > Plugins > User Survey** to enable this integration.

Once the integration is installed and enabled, create surveys by completing configuration in the System Console, as described below.

Create surveys

Under **Survey setup**, specify the date, time, and details for a new survey:

1. **Send next survey at**: Specify the date and time when the survey will begin rolling out to users.

> *Note*
>
> *A single survey can be active at a time. If you already have an active survey running, you'll need to reschedule your new survey to start on a date after the current survey expires. Alternatively, you can end the active survey early by selecting **End survey**.*

2. **Survey expiry (days)**: Specify how long the survey will be open to responses in days.
3. **Exclude specific teams**: Specify who will receive the survey. You can send the survey to all users, to all users on specific teams, or omit all users from specific teams.

4. **Survey message text**: Customize the introductory message text users see when prompted to complete the survey.
5. **Linear scale question (1-10)**: This question is required that helps calculate the NPS score.
6. **Textual question**: This question is required that helps gather user feedback.
7. **Textual question (Optional)**: You can optionally specify a text-based question for your survey.
8. Select **Save**. Your new survey is scheduled, and displays under **Active and past surveys** once the survey is shared with users.

Active surveys

When a survey starts, all users recieve a direct message from a feedback bot containing the active survey. The survey includes one mandatory scale-based question, and optional text-based questions.

Users must answer all required questions in order to submit their response. When a user selects **Submit**, their responses are recorded.

Export survey responses

Select **Export responses** to download a CSV file containing NPS scores and user responses gatherered through the survey. You can export data from active surveys which will contain data collected so far.

On this page

- Manage user surveys
  - Setup
  - Install
    - Upgrade
  - Enable
    - Create surveys
  - Active surveys
  - Export survey responses

Optimize your Mattermost workspace

With workspace optimizations, system admins can review their workspace health and growth scores, then take advantage of recommended actions for ensuring their workspace is running smoothly and teams are maximizing productivity.

System admins can access their workspace optimization page in the System Console by selecting the **Product**

menu, selecting **System Console**, and going to **Reporting > Workspace Optimization**.

How is the overall score calculated?

The highest score possible is 100. Your score is calculated based on the type of issue reported and the level of potential security risk introduced to your Mattermost deployment if ignored.

Each item on the dashboard is calculated based on its individual impact score. These differ depending on whether they're problems, warnings, or suggestions. For example, if SSL encryption isn't configured in your workspace, Mattermost reports that as a problem, which reduces your score until it's addressed.

Warnings impact your score less than problems, and suggestions have the least impact on your score.

Want to improve your overall workspace optimization score? Take action towards the problems, warnings, and suggestions reported on this dashboard. We recommend the following workspace optimizations.

Recommendations

The following optimization areas can alert you to workspace suggestions, warnings, or problems that may require your attention:

Optimization category	Suggestions, Warnings, or Problems Detected	Additional Information
Mattermost release	Are you on the latest Mattermost release?	You're notified when updates are available. See the Upgrade Mattermost product documentation for details on upgrading your workspace.
Configuration issues	• **SSL**: Should your workspace be more secure with SSL? • **Session Length**: The default value may not provide an optimal user experience. • **File Storage**: Write access to the configured file storage location is required. [Mattermost Academy Learn about file storage (https://mattermost.com/pl/mattermost-academy-configure-file-storage-training)](https://mattermost.com/pl/mattermost-academy-configure-file-storage-training)	See the product documentation to learn more: • Set up SSL • Configure session length • Configure file storage
Workspace access	Is the Mattermost workspace may not be accessible to users?	If your web server settings don't pass a live URL test, your workspace may not be accessible to others. See the Web server configuration settings product documentation to learn more:

Optimization category	Suggestions, Warnings, or Problems Detected	Additional Information
Search performance	As your user base grows, is search getting slower?	See the Enterprise search product documentation to learn more.
Data privacy	Do you need more control and insights into your data?	See the product documentation to learn more: • Data Retention • Compliance Export
User authentication	• **AD/LDAP**: As your user base grows, would you benefit from easier onboarding, automated deactivations, and role assignments? • **Guest accounts**: Do you want to control user access to channels and teams with guest accounts?	See the product documentation to learn more: • AD/LDAP • Guest accounts

On this page

- Optimize your Mattermost workspace
  - How is the overall score calculated?
  - Recommendations

Plugins configuration settings

Review and manage the following plugin configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Plugins**:

- Plugin management
- Calls
- AI Agents
- GitLab
- GitHub
- Jira
- Legal Hold
- Microsoft Calendar Integration
- MS Teams
- Performance metrics
- Collaborative playbooks
- User satisfaction surveys
- ServiceNow
- Zoom
- config.json-only settings

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `Enable` value is under `PluginSettings`.
>
> - If using a tool such as jq (https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.PluginSettings.Enable'`
> - When working with the `config.json` file manually, look for an object such as `PluginSettings`, then within that object, find the key `Enable`.

Plugin management

Access the following configuration settings in the System Console by going to **Plugins > Plugin Management**.

Enable plugins

• **true**: **(Default)** Enables plugins on your Mattermost server. See the Use plugins with Mattermost (https://developers.mattermost.com/integrate/plugins/using-and-managing-plugins/) documentation for details. • **false**: Disables plugins on your Mattermost server.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `Enable` > `true` • Environment variable: `MM_PLUGINSETTINGS_ENABLE`

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas:

- Resource Consumption: Plugins consume system resources such as CPU, memory, and disk. Disabling them frees up these resources, allowing the core Mattermost application to run more efficiently.
- Reduced Complexity: Each plugin can add additional logic and processing requirements. By reducing the number of active plugins, you lower the complexity and potential points of failure in the system.
- Faster Load Times: Disabling plugins can lead to faster server startup and lower latency during user interactions, as there are fewer components for the system to initialize and manage.
- Stability: Some plugins may have bugs or performance issues that can affect the overall performance and stability of the Mattermost instance. Disabling problematic or under-utilized plugins can enhance the stability of the system.
- Maintenance and Updates: Managing fewer plugins reduces the overhead associated with maintaining and updating them, which can contribute to smoother operation and less downtime
- However, plugins are often essential for integrating Mattermost with other services and workflows. It’s important to balance performance improvements with the needs of your organization and users.

Require plugin signature

Note

Available only for self-hosted deployments

• **true**: **(Default)** Enables plugin signature validation for managed and unmanaged plugins. • **false**: Disables plugin signature validation for managed and unmanaged plugins.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `RequirePluginSignature` > `true` • Environment variable: `MM_PLUGINSETTINGS_REQUIREPLUGINSIGNATURE`

Note

- Pre-packaged plugins are not subject to signature validation. Plugins installed through the Marketplace are always subject to signature validation at the time of download.
- Enabling this configuration will result in plugin file uploads being disabled in the System Console.

Automatic prepackaged plugins

• **true**: **(Default)** Mattermost automatically installs and upgrades any enabled pre-packaged plugins. If a newer version is installed, no changes are made. • **false**: Mattermost does not automatically install or upgrade pre-packaged plugins. Pre-packaged plugins may be installed manually from the Marketplace, even when offline.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `AutomaticPrepackagedPlugins` > `true` • Environment variable: `MM_PLUGINSETTINGS_AUTOMATICPREPACKAGEDPLUGINS`

Upload Plugin

Note

Available only for self-hosted deployments

• **true**: Enables you to upload plugins from the local computer to the Mattermost server. • **false**: **(Default)** Disables uploading of plugins from the local computer to the Mattermost server.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `EnableUploads` > `false` • Environment variable: `MM_PLUGINSETTINGS_ENABLEUPLOADS`

> Note
>
> - When plugin uploads are enabled, the error `Received invlaid response from the server` when uploading a plugin file typically indicates that the MaxFileSize configuration setting isn't large enough to support the plugin file upload. Additional proxy setting updateds may also be required.
> - The ability to upload plugin files is disabled when the Require plugin signature configuration setting is enabled.

Enable Marketplace

• **true**: **(Default)** Enables the plugin Marketplace on your Mattermost server for all system admins. • **false**: Disables the plugin Marketplace on your Mattermost server for all system admins.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `EnableMarketplace` > `true` • Environment variable: `MM_PLUGINSETTINGS_ENABLEMARKETPLACE`

Enable remote Marketplace

• **true**: **(Default)** Mattermost attempts to connect to the endpoint set in **Marketplace URL**. If the connection fails, an error is displayed, and the Marketplace only shows pre-packaged and installed plugins. • **false**: Mattermost does not attempt to connect to a remote Marketplace. The Marketplace shows only pre-packaged and installed plugins. Use this setting if your Mattermost server cannot connect to the Internet.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `EnableRemoteMarketplace` > `true` • Environment variable: `MM_PLUGINSETTINGS_ENABLEREMOTEMARKETPLACE`

Note

- From Mattermost v9.1, set this configuration setting value to `true` to access a configured remote marketplace URL.
- For Mattermost v9.0, the `MM_FEATUREFLAGS_STREAMLINEDMARKETPLACE` feature flag must be set to `false`, and this configuration setting must be set to `true` to access a configured remote marketplace URL.
- Each Mattermost host must have network access to the endpoint set in MarketplaceURL.

Marketplace URL

This setting stores the URL for the remote Markeplace. String input. Default is **https://api.integrations.mattermost.com**	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `MarketplaceURL` • Environment variable: `MM_PLUGINSETTINGS_MARKETPLACEURL`

Installed plugin state

This setting is a list of installed plugins and their status as enabled or disabled. The `config.json` setting is an object. The object keys are plugin IDs, e.g. `com.mattermost.apps`. Each key maps to an object that contains an `Enable` key that can be set as `true` or `false`.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `PluginStates` • Environment variable: `MM_PLUGINSETTINGS_PLUGINSTATES`

Plugin settings

This setting contains plugin-specific data. The `config.json` setting is an object. The object keys are plugin IDs, e.g. `com.mattermost.apps`. Each key maps to an object that contains plugin-specific data.	• System Config path: **Plugins > Plugin Management** • `config.json` setting: `PluginSettings` > `Plugins` • Environment variable: `MM_PLUGINSETTINGS_PLUGINS`

Calls

Access the following configuration settings in the System Console by going to **Plugins > Calls**.

Enable plugin

• **true**: (Default) Enables the Calls plugin on your Mattermost workspace. • **false**: Disables the Calls plugin on your Mattermost workspace.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `PluginStates` > `com.mattermost.calls` > `Enable` • Environment variable: `MM_PLUGINSETTINGS_PLUGINSTATES_COM_MATTERMOST_CALLS`

RTC server address (UDP)

Note

Available only for self-hosted deployments

This setting controls the IP address the RTC server listens for UDP connections. All calls UDP traffic will be served through this IP. Changing this setting requires a plugin restart to take effect. If left unset (default value) the service will listen on all the available interfaces.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` `Plugins` > `com.mattermost.calls` > `udpserveraddress` • Environment variable: `MM_CALLS_UDP_SERVER_ADDRESS`

> Note
>
> This setting is only applicable when not running calls through the standalone `rtcd` service.

RTC server address (TCP)

Note

Available only for self-hosted deployments

This setting controls the IP address the RTC server listens for TCP connections. All calls TCP traffic will be served through this IP. Changing this setting requires a plugin restart to take effect. If left unset (default value) the service will listen on all the available interfaces.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `tcpserveraddress` • Environment variable: `MM_CALLS_TCP_SERVER_ADDRESS`

> Note
>
> This setting is available starting in plugin version 0.17, and is only applicable when not running calls through the standalone `rtcd` service.

RTC server port (UDP)

Note

Available only for self-hosted deployments

This setting controls the UDP port listened on by the RTC server. All calls UDP traffic will be served through this port. Changing this setting requires a plugin restart to take effect. Default is **8443**.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `udpserverport` • Environment variable: `MM_CALLS_UDP_SERVER_PORT`

> Note
>
> This setting is only applicable when not running calls through the standalone `rtcd` service.

RTC server port (TCP)

Note

Available only for self-hosted deployments

This setting controls the TCP port listened on by the RTC server. All calls TCP traffic will be served through this port. Changing this setting requires a plugin restart to take effect. Default is **8443**.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `tcpserverport` • Environment variable: `MM_CALLS_TCP_SERVER_PORT`

> Note
>
> This setting is available starting in plugin version 0.17, and is only applicable when not running calls through the standalone `rtcd` service.

Enable on specific channels

Admins can't configure this setting from Mattermost v7.7; it's hidden and always enabled

Note

Available only for self-hosted deployments

• **true**: Channel admins can enable or disable calls on specific channels. Participants in DMs/GMs can also enable or disable calls. • **false**: Only system admins can enable or disable calls on specific channels.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `allowenablecalls` • Environment variable: `MM_CALLS_ALLOW_ENABLE_CALLS`

Test mode

This setting was called Enable on all channels until Mattermost v7.7. It was renamed to defaultenabled in code and Test Mode in-product.

Note

Available only for self-hosted deployments

• **false**: Test mode is enabled and only system admins can start calls in channels. • **true**: Live mode is enabled and all team members can start calls in channels.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `defaultenabled` • Environment variable: `MM_CALLS_DEFAULT_ENABLED`

> Note
>
> Use this setting as a system admin to confirm calls work as expected. When **false**, users attempting to start calls are prompted to contact a system admin, and system admins are prompted to confirm that calls are working as expected before switching to live mode.

ICE host override

Note

Available only for self-hosted deployments

This setting can be used to override the host addresses that get advertised to clients when connecting to calls. The accepted formats are the following: • A single IP address (e.g. `10.0.0.1`). • A single hostname or FQDN (e.g. `calls.myserver.tld`). • (starting in v0.17.0) A comma separated list of externalAddr/internalAddr mappings (e.g. `10.0.0.1/172.0.0.1,10.0.0.2/172.0.0.2`). This is an optional field. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `icehostoverride` • Environment variable: `MM_CALLS_ICE_HOST_OVERRIDE`

Note

- This setting is only applicable when not running calls through the standalone `rtcd` service.
- Depending on the network infrastructure (e.g. instance behind a NAT device) it may be necessary to set this field to the client facing external IP for clients to connect. When empty or unset, the RTC service will attempt to find the instance's public IP through STUN.
- A hostname (e.g. domain name) can be specified in this setting, but an IP address will be passed to clients. This means that a DNS resolution happens on the Mattermost instance which could result in a different IP address from the one the clients would see, causing connectivity to fail. When in doubt, we recommend using an IP address directly or confirming that the resolution on the host side reflects the one on the client.

ICE host port override

Note

Available only for self-hosted deployments

This setting can be used to override the port used in the ICE host candidates that get advertised to clients when connecting to calls. This can be useful in case there are additional network components (e.g. NLBs) in front of the RTC server that may route the calls traffic through a different port. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `icehostportoverride` • Environment variable: `MM_CALLS_ICE_HOST_PORT_OVERRIDE`

> Note
>
> This value will apply to both UDP and TCP host candidates.

RTCD service URL

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The URL to a running rtcd (https://github.com/mattermost/rtcd) service instance that will host the calls. When set (non empty) all the calls will be handled by this external service. This is an optional field. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `rtcdserviceurl` • Environment variable: `MM_CALLS_RTCD_SERVICE_URL`

Note

- The environment variable `MM_CALLS_RTCD_URL` is deprecated in favor of `MM_CALLS_RTCD_SERVICE_URL`.
- The client will self-register the first time it connects to the service and store the authentication key in the database. If no client ID is explicitly provided, the diagnostic ID of the Mattermost installation will be used.
- The service URL supports credentials in the form `http://clientID:authKey@hostname`. Alternatively these can be passed through environment overrides to the Mattermost server, namely `MM_CALLS_RTCD_CLIENT_ID` and `MM_CALLS_RTCD_AUTH_KEY`

Note

- The client will self-register the first time it connects to the service and store the authentication key in the database. If no client ID is explicitly provided, the diagnostic ID of the Mattermost installation will be used.
- The service URL supports credentials in the form `http://clientID:authKey@hostname`. Alternatively these can be passed through environment overrides to the Mattermost server, namely `MM_CALLS_RTCD_CLIENT_ID` and `MM_CALLS_RTCD_AUTH_KEY`

Max call participants

Note

Available only for self-hosted deployments

This setting limits the number of participants that can join a single call. Default is **0** (no limit).	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `maxcallparticipants` • Environment variable: `MM_CALLS_MAX_CALL_PARTICIPANTS`

> Note
>
> - The environment variable `MM_CALLS_MAX_PARTICIPANTS` is deprecated in favor of `MM_CALLS_MAX_CALL_PARTICIPANTS`.
> - This setting is optional, but the recommended maximum number of participants is **50**. Call participant limits greatly depends on instance resources. See the Calls self-hosted deployment documentation for details.

ICE servers configurations

Note

Available only for self-hosted deployments

This setting stores a list of ICE servers (STUN/TURN) in JSON format to be used by the service. This is an optional field. Changing this setting may require a plugin restart to take effect. Default is `[{"urls": ["stun:stun.global.calls.mattermost.com:3478"]}]`	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `iceserversconfigs` • Environment variable: `MM_CALLS_ICE_SERVERS_CONFIGS`

> Note
>
> - The configurations above, containing STUN and TURN servers, are sent to the clients and used to generate local candidates.
> - If hosting calls through the plugin (i.e. not using the rtcd service) any configured STUN server may also be used to find the instance's public IP when none is provided through the ICE Host Override option.

Example

```
[ { "urls":[ "stun:stun.global.calls.mattermost.com:3478" ] }, { "urls":[ "turn:turn.example.com:3478" ], "username":"webrtc", "credentials":"turnpassword" } ]
```

Example (Using generated TURN credentials)

```
[{ "urls": ["turn:turn.example.com:443"] }]
```

> Note
>
> To get TURN generated credentials to work you must provide a secret through the *TURN static auth secret* setting below.

TURN static auth secret

Note

Available only for self-hosted deployments

A static secret used to generate short-lived credentials for TURN servers. This is an optional field.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `turnstaticauthsecret` • Environment variable: `MM_CALLS_TURN_STATIC_AUTH_SECRET`

TURN credentials expiration

Note

Available only for self-hosted deployments

The expiration, in minutes, of the short-lived credentials generated for TURN servers. Default is **1440** (one day).	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `turncredentialsexpirationminutes` • Environment variable: `MM_CALLS_TURN_CREDENTIALS_EXPIRATION_MINUTES`

Server side TURN

Note

Available only for self-hosted deployments

• **true**: The RTC server will use the configured TURN candidates for server-initiated connections. • **false**: TURN will be used only on the client-side. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `serversideturn` • Environment variable: `MM_CALLS_SERVER_SIDE_TURN`

Allow screen sharing

Note

Available only for self-hosted deployments

• **true**: Call participants will be allowed to share their screen. • **false**: Call participants won't be allowed to share their screen. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `allowscreensharing` • Environment variable: `MM_CALLS_ALLOW_SCREEN_SHARING`

Enable simulcast for screen sharing (Experimental)

Note

Available only for self-hosted deployments

• **true**: Enables simulcast for screen sharing. This can help to improve screen sharing quality. • **false**: Disables simulcast for screen sharing.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enablesimulcast` • Environment variable: `MM_CALLS_ENABLE_SIMULCAST`

> Note
>
> - This functionality requires Calls plugin version >= v0.16.0 and `rtcd` version >= v0.10.0 (when in use).
> - Avoid enabling both this experimental configuration setting and the Enable AV1 experimental configuration setting at the same time.

Enable call recordings

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

• **true**: Allows call hosts to record meeting video and audio. • **false**: **(Default)** Call recording functionality is not available to hosts. Recordings include the entire call window view along with participants' audio track and any shared screen video. Recordings are stored in Mattermost. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enablerecordings` • Environment variable: `MM_CALLS_ENABLE_RECORDINGS`

Job service URL

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The URL to a running job service where all the processing related to recordings happens. The recorded files produced are stored in Mattermost. This is a required field. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `jobserviceurl` • Environment variable: `MM_CALLS_JOB_SERVICE_URL`

> Note
>
> - The client will self-register the first time it connects to the service and store the authentication key in the database. If no client ID is explicitly provided, the diagnostic ID of the Mattermost installation will be used.
> - The service URL supports credentials in the form `http://clientID:authKey@hostname`. Alternatively these can be passed through environment overrides to the Mattermost server, namely `MM_CALLS_JOB_SERVICE_CLIENT_ID` and `MM_CALLS_JOB_SERVICE_AUTH_KEY`.
> - As of Calls v0.25 it's possible to override the site URL used by jobs to connect by setting the `MM_CALLS_RECORDER_SITE_URL` or `MM_CALLS_TRANSCRIBER_SITE_URL` environment variables respectively. This can be helpful to avoid the jobs from connecting through the public Site URL configured in Mattermost and thus potentially bypass the public network.

Maximum call recording duration

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The maximum duration of a call recording in minutes. The default is **60**. The maximum is **180**. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `maxrecordingduration` • Environment variable: `MM_CALLS_MAX_RECORDING_DURATION`

Call recording quality

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The audio and video quality of call recordings. Available options are: *Low*, *Medium* and *High*. The default is **Medium**. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `recordingquality` • Environment variable: `MM_CALLS_RECORDING_QUALITY`

> Note
>
> The quality setting will affect the performance of the job service and the file size of recordings. Refer to the deployment section for more information.

Enable call transcriptions

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

• **true**: Enables automatic transcriptions of calls. • **false**: **(Default)** Call transcriptions functionality is disabled. Transcriptions are generated from the call participants' audio tracks and the resulting files are attached to the call thread when the recording ends. Captions will be optionally rendered on top of the recording file video player.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enabletranscriptions` • Environment variable: `MM_CALLS_ENABLE_TRANSCRIPTIONS`

> Note
>
> - The ability to enable call transcriptions in Mattermost calls is currently in Beta.
> - This server-side configuration setting is available from plugin version 0.22.
> - Call transcriptions require call recordings to be enabled.

Transcriber model size

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The speech-to-text model size to use. Heavier models will produce more accurate results at the expense of processing time and resources usage. Available options are: *Tiny*, *Base* and *Small*. The default is **Base**. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `transcribermodelsize` • Environment variable: `MM_CALLS_TRANSCRIBER_MODEL_SIZE`

> Note
>
> This setting is available starting in plugin version 0.22. The model size setting will affect the performance of the job service. Refer to the configure call recordings, transcriptions, and live captions documentation for more information.

Call transcriber threads

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The number of threads used by the post-call transcriber. This must be in the range [1, numCPUs]. The default is 2. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `transcribernumthread` • Environment variable: `MM_CALLS_TRANSCRIBER_NUM_THREADS`

> Note
>
> The call transcriber threads setting will affect the performance of the job service. Refer to the configure call recordings, transcriptions, and live captions documentation for more information. This setting is available starting in plugin version 0.26.2.

Enable live captions

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

• **true**: Enables live captioning of calls. • **false**: **(Default)** Live captions functionality is disabled. Live captions are generated from the call participants' audio tracks and the resulting captions can be optionally displayed on the call clients by selecting the **[cc]** option.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enablelivecaptions` • Environment variable: `MM_CALLS_ENABLE_LIVE_CAPTIONS`

> Note
>
> - The ability to enable live call captions in Mattermost calls is currently in Beta.
> - This server-side configuration setting is available starting in plugin version 0.26.2.
> - Live captions require call recordings and call transcriptions to be enabled.

Live captions: Model size

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The speech-to-text model size to use for live captions. While heavier models can produce more accurate results, live captioning requires the transcriber to process up to ten seconds of audio within two seconds. Therefore a maximum of size *base* is recommended. Available options are: *Tiny*, *Base* and *Small*. The default is **Tiny**. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `livecaptionsmodelsize` • Environment variable: `MM_CALLS_LIVE_CAPTIONS_MODEL_SIZE`

> Note
>
> This setting is available starting in plugin version 0.26.2. The model size setting will affect the performance of the job service. Refer to the performance and scalability recommendations (https://github.com/mattermost/calls-offloader/blob/master/docs/performance.md) documentation for more information.

Live captions: Number of transcribers used per call

Note

Available only on Enterprise (https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The number of separate live captions transcribers for each call. Each transcribes one audio stream at a time. The product of LiveCaptionsNumTranscribers * LiveCaptionsNumThreadsPerTranscriber must be in the range [1, numCPUs]. The default is 1. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `livecaptionsnumtranscribers` • Environment variable: `MM_CALLS_LIVE_CAPTIONS_NUM_TRANSCRIBERS`

> Note
>
> This setting is available starting in plugin version 0.26.2. The live captions number of transcribers setting will affect the performance of the job service. Refer to the performance and scalability recommendations (https://github.com/mattermost/calls-offloader/blob/master/docs/performance.md) documentation for more information.

Live captions: Number of threads per transcriber

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The number of threads per live-captions transcriber. The product of `LiveCaptionsNumTranscribers` * `LiveCaptionsNumThreadsPerTranscriber` must be in the range [1, numCPUs]. The default is 2. This is a required value.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `livecaptionsnumthreadspertranscriber` • Environment variable: `MM_CALLS_LIVE_CAPTIONS_NUM_THREADS_PER_TRANSCRIBER`

> Note
>
> This setting is available starting in plugin version 0.26.2. The live captions number of threads per transcriber setting will affect the performance of the job service. Refer to the [performance and scalability recommendations (https://github.com/mattermost/calls-offloader/blob/master/docs/performance.md)](https://github.com/mattermost/calls-offloader/blob/master/docs/performance.md) documentation for more information

Live captions language

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Available only for self-hosted deployments

The language passed to the live captions transcriber. Should be a 2-letter ISO 639 Set 1 language code, e.g. 'en'. If blank, the lange will be set to 'en' (English) as default.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `livecaptionslanguage` • Environment variable: `MM_CALLS_LIVE_CAPTIONS_LANGUAGE`

(Experimental) Enable IPv6

Note

Available only for self-hosted deployments

• **true**: The RTC service will work in dual-stack mode, listening for IPv6 connections and generating candidates in addition to IPv4 ones. • **false**: **(Default)** The RTC service will only listen for IPv4 connections. Changing this setting requires a plugin restart to take effect.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enableipv6` • Environment variable: `MM_CALLS_ENABLE_IPV6`

> Note
>
> This setting is available starting in plugin version 0.17, and is only applicable when not running calls through the standalone `rtcd` service.

Enable call ringing

• **true**: Ringing functionality is enabled. Direct and group message participants receive a desktop app alert and a ringing notification when a call starts. • **false**: **(Default)** Ringing functionality is disabled.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enableringing` • Environment variable: `MM_CALLS_ENABLE_RINGING`

> Note
>
> The ability to enable call ringing in Mattermost calls is in Beta.

Enable AV1 (Experimental)

• **true**: Enables the ability to use the AV1 codec to encode screen sharing tracks. Can result in improved screen sharing quality via clients that support AV1 encoding. • **false**: **(Default)** AV1 codec is disabled for screen sharing tracks.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enableAV1` • Environment variable: `MM_CALLS_ENABLE_AV1`

> Note
>
> Avoid enabling both this experimental configuration setting and the Enable simulcast for screen sharing experimental configuration setting at the same time.

Enable DC signaling (Experimental)

• **true**: Clients will use WebRTC data channels for signaling of media tracks (i.e., voice, screen). This can result in a more efficient and less race-prone process, especially in case of poor network connections. • **false**: **(Default)** Clients will use WebSockets for signaling media tracks.	• System Config path: **Plugins > Calls** • `config.json` setting: `PluginSettings` > `Plugins` > `com.mattermost.calls` > `enabledcsignaling` • Environment variable: `MM_CALLS_ENABLE_DC_SIGNALING`

> Note
>
> - Version v0.18.0 or higher of the rtcd service is required for this functionality to work when hosting calls through the dedicated WebRTC service.
> - Use caution when enabling this experimental configuration setting since it determines how the system handles part of the setup for WebRTC-based calls. Enabling this configuration setting may make the call setup a bit faster or more reliable in certain situations.

AI Agents

Available onall plansCloudandself-hosteddeployments

> Note
>
> Mattermost Agents is formally known as Mattermost Copilot.

Access the following Mattermost Agents configuration settings in the System Console by going to **Plugins > Agents**.

Enable plugin

• **true**: Enables the Agents plugin on your Mattermost workspace. • **false**: **(Default)** Disables the Agents plugin.	• System Config path: **Plugins > Agents** • `config.json` setting: N/A • Environment variable: N/A

Display name

The bot's display name in Mattermost used to distinguish the bot from other bots in the system. String input.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Agent username

The bot's username that can be used to @mention the bot in a channel. String input.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Agent avatar

Upload an image to use as the agent's avatar in Mattermost. Image upload interface.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Service

Select the LLM service provider to use for AI assistance. Available options: **OpenAI**, **OpenAI Compatible**, **Azure**, **Anthropic**, and **Ask Sage**.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Username

The username used to authenticate with the **Ask Sage** LLM service. String input required.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Password

The password used to authenticate with the **Ask Sage** LLM service. String input required. This value is encrypted when stored.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

API URL

The endpoint that Mattermost will use to communicate with the LLM's API. Required for **OpenAI Compatible** and **Azure** LLM services. String input (URL format).	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

API key

The key used to authenticate requests to the LLM's API. Required for **OpenAI**, **OpenAI Compatible**, and **Anthropic** LLM services. String input. This value is encrypted when stored.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Organization ID

Ensures that requests are billed and processed under the correct organization, where applicable. Supported for **OpenAI**, **OpenAI Compatible**, and **Azure** LLM services. String input.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Send user ID

• **true**: Includes unique user identifiers in API requests to the LLM for analytics, personalization, and auditing purposes. • **false**: **(Default)** Does not include user identifiers. Review LLM data privacy policies before enabling this setting.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> We recommend reviewing LLM data privacy policies to confirm whether transmitting user information is acceptable and secure within your organization's regulatory framework. Do not enable when you need to conform to strict privacy regulations (e.g., GDPR) that limit sharing user-identifiable data with external services.

Default model

The specific LLM that will be used to process queries if no other model is explicitly selected. Supported for all LLM services. String input (model name).	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Input token limit

The maximum number of tokens (chunks of text, including words, punctuation, or special characters) that the selected LLM can process in a single prompt or request. Numerical value. Directly impacts the size of user queries that can be handled.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Output token limit

The maximum number of tokens (chunks of text, including words, punctuation, or special characters) that the LLM can generate in its response to a query. Numerical value. Must be greater than 0 for **Anthropic** LLM services.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Streaming timeout

Determines how long the system will wait for a response from the LLM when using streaming (real-time) output mode. Supported for **OpenAI**, **OpenAI Compatible**, and **Azure** LLM services. Numerical value (in seconds). If the LLM takes longer than the configured timeout, the connection is terminated.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Streaming allows LLMs to display the response gradually as it's being generated, creating a smoother and more interactive experience for users.

Custom instructions

Preset specific contextual or behavioral guidance for the LLM. Helps tailor the model's responses to align with your organization's needs, tone, or expectations. Supported for all LLM services. Text input. Instructions that the model will implicitly follow for every interaction, providing consistency and adaptability.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Custom instructions can include behavioral guidance, tone preferences, contextual functions, and organizational preferences. This ensures responses adhere to your organization's language and tone guidelines and aligns the model's behavior with specific roles or purposes.

Enable vision

• **true**: Enables the LLM to process and generate responses that incorporate image-related input or output. Supported for **OpenAI**, **OpenAI Compatible**, **Azure**, and **Anthropic** LLM services. • **false**: **(Default)** Disables vision capabilities.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> This feature is in Beta. When enabled, the LLM can interact with prompts that include image-related input, such as image analysis, visual-related assistance, and visual outputs, where supported.

Enable tools

• **true**: Enables the LLM to leverage additional tools or plugins to enhance its capabilities. Supported for **OpenAI**, **OpenAI Compatible**, **Azure**, and **Anthropic** LLM services. • **false**: **(Default)** Disables tool capabilities.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> When enabled, advanced features beyond basic query processing allow the LLM to perform specialized tasks like retrieving data, integrating with external APIs, or performing computations, where supported.

Channel access

Determines whether the bot can consume the contents of a given channel and provide answers only from content available in the channel. Supported for all LLM services. Available options: **Allow for all channels**, **Allow for selected channels**, **Block selected channels**, and **Block all channels**.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

User access

Determines whether users who chat with this bot can get private assistance about content across all channels the user has access to. Supported for all LLM services. Available options: **Allow for all users**, **Allow for selected users**, and **Block selected users**.	• System Config path: **Plugins > Agents > Add an AI Bot** • `config.json` setting: N/A • Environment variable: N/A

Default agent

Select the default bot to use for AI functions when multiple agents are configured. Based on defined agents.	• System Config path: **Plugins > Agents > AI Functions** • `config.json` setting: N/A • Environment variable: N/A

Allowed upstream hostnames

Comma-separated list of hostnames that LLMs are allowed to contact when using tools. Supports wildcards like `*.mydomain.com`. Example: `mattermost.atlassian.net` to allow JIRA tool use.	• System Config path: **Plugins > Agents > AI Functions** • `config.json` setting: N/A • Environment variable: N/A

Enable LLM trace

• **true**: Enables tracing of LLM requests and outputs full conversation data to the logs. Supported for all LLM services. • **false**: **(Default)** Disables LLM request tracing.	• System Config path: **Plugins > Agents > Debug** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Use this setting for debugging purposes only. When enabled, it may log sensitive conversation data.

Enable embedding search

Available onEnterprise plansCloudandself-hosteddeployments

• **Composite**: Enables experimental embedding search capabilities for semantic search across Mattermost content using pgvector and OpenAI-compatible endpoints. • **Disabled**: **(Default)** Disables embedding search capabilities.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Embedding search requires an Enterprise license and is available as an experimental feature. You must also enable the `pgvector` extension in your PostgreSQL database. Performance may vary with large datasets.

Embedding provider type

Select the provider for generating embeddings for semantic search. Available options: **OpenAI** and **OpenAI Compatible**.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

API Key

The API key used to authenticate requests to the embedding provider's API. Required for **OpenAI Compatible** embedding providers. String input. This value is encrypted when stored.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Model

The specific embedding model to use for generating vector representations of content. Must be compatible with the selected embedding provider. String input (model name).	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

API URL

The endpoint URL for the **OpenAI-compatible** embedding API. Required string input (URL format).	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Dimensions

The dimensionality of the embedding vectors, which must match the chosen embedding model. Common values include 1536 for OpenAI text-embedding-ada-002 and 3072 for text-embedding-3-large. Numerical input. Common values are 768, 1024, or 1536, depending on the model.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Chunking strategy

The method used to split content into smaller chunks for embedding generation. Available options: **Sentences**, **Paragraphs**, **Fixed Size**. Choose based on your content type and search requirements.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Chunk size

The maximum size of each content chunk in characters. Recommended range is 512-1024 characters. The optimal value varies by chunking strategy. Numerical input.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Chunk overlap

The number of tokens that consecutive chunks share for better context continuity. Recommended range is 20-50 characters for **Fixed Size** chunking. Numerical input.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Minimum chunk size ratio

The minimum ratio for chunk size validation to ensure sentence and paragraph chunks meet size requirements. Used to filter out chunks that are too small releative to the configured chunk size. Numerical input (decimal ratio).	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

Reindex all posts

Select **Reindex Posts** to trigger a complete reindexing of all posts for embedding search. Use this control to rebuild the search index when changing embeddingproviders, models, or chunking configurations. Monitor indexing progress during the reindexing process.	• System Config path: **Plugins > Agents > Embedding Search** • `config.json` setting: N/A • Environment variable: N/A

GitLab

Available onall plansCloudandself-hosteddeployments

See the Connect GitLab to Mattermost product documentation for available Mattermost configuration options.

GitHub

Available onall plansCloudandself-hosteddeployments

See the Connect GitHub to Mattermost product documentation for available Mattermost configuration options.

Jira

Available onall plansCloudandself-hosteddeployments

See the Connect Jira to Mattermost product documentation for available Mattermost configuration options.

Legal hold

Available onEnterprise plansCloudandself-hosteddeployments

See the Legal holds product documentation for details.

Microsoft Calendar Integration

Available onall plansCloudandself-hosteddeployments

See the Connect Microsoft Calendar Integration to Mattermost product documentation for available Mattermost configuration options.

Microsoft Teams Meetings

Available onall plansCloudandself-hosteddeployments

See the Connect Microsoft Teams Meetings to Mattermost product documentation for available Mattermost configuration options.

MS Teams

Available onall plansCloudandself-hosteddeployments

Mattermost for Microsoft Teams enables you to break through siloes in a mixed Mattermost and Teams environment by forwarding real-time chat notifications from Teams to Mattermost.

> Tip
>
> Download our [Mattermost for Microsoft Teams datasheet (https://mattermost.com/mattermost-for-microsoft-teams-datasheet/)](https://mattermost.com/mattermost-for-microsoft-teams-datasheet/) to learn how Mattermost helps your organization get more from your Microsoft tools.

Access the following configuration settings in the System Console by going to **Plugins > MS Teams**.

[Mattermost Academy Learn about integrating with Microsoft Teams (https://academy.mattermost.com/p/new-mattermost-for-microsoft-teams-integration)](https://academy.mattermost.com/p/new-mattermost-for-microsoft-teams-integration)

Enable plugin

Enable the Mattermost for Microsoft Teams plugin for all Mattermost teams. • **true**: Enables MS Teams plugin on your Mattermost workspace. • **false**: **(Default)** Disables the MS Teams plugin.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Use the Enabled Teams configuration option to specify which Mattermost teams synchronize direct and group messages with Microsoft Teams chats.

Tenant ID

Specify the Microsoft Teams Tenant ID from the Azure portal.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Client ID

Specify the Microsoft Teams Client ID of your registered OAuth app in the Azure portal.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Client secret

Specify the client secret of your registered OAuth app in Azure portal. Alpha-numeric value.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

At rest encryption key

Regenerate a new encryption secret. This encryption secret will be used to encrypt and decrypt the OAuth token. Alpha-numeric value.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Select **Regenerate** to generate a new key.

Webhook secret

Generate the webhook secret that Microsoft Teams will use to send messages to Mattermost.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Select **Regenerate** to generate a new key.

Use the evaluation API pay model

Enable this only for testing purposes. You need the pay model to be able to support enough message notifications to work in a real world scenario. • **true**: Enables the evaluation API pay model. • **false**: **(Default)** Disables the evaluation API pay model.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Sync notifications

Notify connected users in Mattermost on receipt of a chat or group chat from Microsoft Teams. • **true**: **(Default)** Sync notifications of chat messages for any connected user that enables the feature. • **false**: Do not sync notifications.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Maximum size of attachments to support complete one time download

Specify the maximum file size, in mebibytes (MiB), of attachments that can be loaded into memory. Attachment files larger than this value will be streamed from Microsoft Teams to Mattermost. Numerical value. Default is **20** MiB.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Buffer size for streaming files

Specify the buffer size, in mebibytes (MiB), for streaming attachment files from Microsoft Teams to Mattermost. Numerical value. Default is **20** MiB.	• System Config path: **Plugins > MS Teams** • `config.json` setting: N/A • Environment variable: N/A

Performance metrics

Available onEnterprise plansCloudandself-hosteddeployments

See the Monitor performance metrics product documentation for available Mattermost configuration options.

Collaborative playbooks

Available onEnterprise plansCloudandself-hosteddeployments

Use collaborative playbooks in Mattermost to provide structure, monitoring and automation for repeatable, team-based processes integrated with the Mattermost platform.

Access the following configuration settings in the System Console by going to **Plugins > Collaborative playbooks**.

Enable plugin

• **true**: **(Default)** Enables collaborative Playbooks on your Mattermost workspace. • **false**: Disables collaborative Playbooks on your Mattermost workspace.	• System Config path: **Plugins > Collaborative playbooks** • `config.json` setting: • Environment variable:

Enabled teams

Enable collaborative playbooks for all Mattermost teams, or for only selected teams.	• System Config path: **Plugins > Collaborative playbooks** • `config.json` setting: • Environment variable:

Enable experimental features

• **true**: Enables experimental playbooks features on your Mattermost workspace. • **false**: Disables experimental playbooks features on your Mattermost workspace.	• System Config path: **Plugins > Collaborative playbooks** • `config.json` setting: • Environment variable:

User satisfaction surveys

Available onall plansCloudandself-hosteddeployments

This plugin enables Mattermost to send user satisfaction surveys to gather feedback and improve product quality directly from your Mattermost users. Please refer to the [Mattermost Privacy Policy (https://mattermost.com/privacy-policy/)](https://mattermost.com/privacy-policy/) for more information on the collection and use of information received through Mattermost services.

Access the following configuration settings in the System Console by going to **Plugins > User Satisfaction Surveys**.

Enable plugin

• **true**: (Default) Enables the Mattermost User Satisfaction Surveys plugin on your Mattermost workspace. • **false**: Disables the Mattermost User Satisfaction Surveys plugin on your Mattermost workspace.	• System Config path: **Plugins > User Satisfaction Surveys** • `config.json` setting: `PluginSettings.PluginStates.com.mattermost.user-survey.Enable` • Environment variable: N/A

Enable user satisfaction survey

• **true**: A survey is sent to all users every quarter. Results are used by Mattermost, Inc. to improve the product. • **false**: (Default) User satisfaction surveys are disabled.	• System Config path: **Plugins > User Satisfaction Surveys** • `config.json` setting: `PluginSettings.Plugins.com.mattermost.user-survey.systemconsolesetting.EnableSurvey` • Environment variable: N/A
Note: See the [Mattermost Privacy Policy (https://mattermost.com/privacy-policy/)](https://mattermost.com/privacy-policy/) for more information on the collection and use of information by Mattermost.	

ServiceNow

Available onall plansCloudandself-hosteddeployments

See the Connect ServiceNow to Mattermost product documentation for available Mattermost configuration options.

Zoom

Available onall plansCloudandself-hosteddeployments

See the Connect Zoom to Mattermost product documentation for available Mattermost configuration options.

---

config.json-only settings

Available onall plansself-hosteddeployments

Signature public key files

This setting isn't available in the System Console and can only be set in `config.json`.

In addition to the Mattermost plugin signing key built into the server, each public key specified here is trusted to validate plugin signatures.

> This feature's `config.json` setting is `"SignaturePublicKeyFiles": {}` with string array input consisting of contents that are relative or absolute paths to signature files.

Chimera OAuth proxy URL

This setting isn't available in the System Console and can only be set in `config.json`.

Specify the [Chimera (https://github.com/mattermost/chimera)](https://github.com/mattermost/chimera) URL used by Mattermost plugins to connect with pre-created OAuth applications.

> This feature's `config.json` setting is `"ChimeraOAuthProxyUrl": {}` with string input.

On this page

- Plugins configuration settings
  - Plugin management
    - Enable plugins
    - Require plugin signature
    - Automatic prepackaged plugins
    - Upload Plugin
    - Enable Marketplace
    - Enable remote Marketplace
    - Marketplace URL
    - Installed plugin state
    - Plugin settings

- Calls
  - Enable plugin
  - RTC server address (UDP)
  - RTC server address (TCP)
  - RTC server port (UDP)
  - RTC server port (TCP)
  - Enable on specific channels
  - Test mode
  - ICE host override
  - ICE host port override
  - RTCD service URL
  - Max call participants
  - ICE servers configurations
  - TURN static auth secret
  - TURN credentials expiration
  - Server side TURN
  - Allow screen sharing
  - Enable simulcast for screen sharing (Experimental)
  - Enable call recordings
  - Job service URL
  - Maximum call recording duration
  - Call recording quality
  - Enable call transcriptions
  - Transcriber model size
  - Call transcriber threads
  - Enable live captions
  - Live captions: Model size
  - Live captions: Number of transcribers used per call
  - Live captions: Number of threads per transcriber
  - Live captions language
  - (Experimental) Enable IPv6
  - Enable call ringing
  - Enable AV1 (Experimental)
  - Enable DC signaling (Experimental)

- AI Agents
  - Enable plugin
  - Display name
  - Agent username
  - Agent avatar
  - Service
  - Username
  - Password
  - API URL
  - API key
  - Organization ID
  - Send user ID
  - Default model
  - Input token limit
  - Output token limit
  - Streaming timeout
  - Custom instructions
  - Enable vision
  - Enable tools
  - Channel access
  - User access
  - Default agent
  - Allowed upstream hostnames
  - Enable LLM trace
  - Enable embedding search
  - Embedding provider type
  - API Key
  - Model
  - API URL
  - Dimensions
  - Chunking strategy
  - Chunk size
  - Chunk overlap
  - Minimum chunk size ratio

    - Reindex all posts
- GitLab
- GitHub
- Jira
- Legal hold
- Microsoft Calendar Integration
- Microsoft Teams Meetings
- MS Teams
    - Enable plugin
    - Tenant ID
    - Client ID
    - Client secret
    - At rest encryption key
    - Webhook secret
    - Use the evaluation API pay model
    - Sync notifications
    - Maximum size of attachments to support complete one time download
    - Buffer size for streaming files
- Performance metrics
- Collaborative playbooks
    - Enable plugin
    - Enabled teams
    - Enable experimental features
- User satisfaction surveys
    - Enable plugin
    - Enable user satisfaction survey
- ServiceNow
- Zoom
- config.json-only settings
    - Signature public key files
    - Chimera OAuth proxy URL

Enable push notifications

Configure mobile push notifications for Mattermost by going to **System Console > Environment > Push Notification Server**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Enable or disable Mattermost push notifications. • **Do not send push notifications**: Mobile push notifications are disabled. • **Use HPNS connection with uptime SLA to send notifications to iOS and Android apps**: **(Default)** Use Mattermost's hosted push notification service. • **Use TPNS connection to send notifications to iOS and Android apps**: Use Mattermost's test push notification service. • **Manually enter Push Notification Service location**: When building your own custom mobile apps, you must host your own mobile push proxy service, and specify that URL in the **Push Notification Server** field.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".EmailSettings.SendPushNotifications": true",` • Environment variable: `MM_EMAILSETTINGS_SENDPUSHNOTIFICATIONS`

Hosted Push Notifications Service (HPNS)

Mattermost Enterprise, Professional, and Cloud customers can use Mattermost's Hosted Push Notification Service (HPNS). The HPNS offers:

- Access to a publicly-hosted Mattermost Push Notification Service (MPNS) available on GitHub. (https://github.com/mattermost/mattermost-push-proxy)
- An explicit privacy policy (https://mattermost.com/data-processing-addendum/) for the contents of unencrypted messages.
- Encrypted TLS connections:
  - Between HPNS and Apple Push Notification Services
  - Between HPNS and Google's Firebase Cloud Messaging Service
  - HPNS and your Mattermost Server
- Production-level uptime expectations.
- Out-of-box configuration for new servers means nothing is required to enable HPNS for new deployments. HPNS can be enabled for existing deployments.

> Note
>
> - The HPNS only works with pre-built apps Mattermost deploys through the Apple App Store (https://www.apple.com/app-store/) and Google Play Store (https://play.google.com/store/games?hl=en). If you build your own mobile apps, you must also host your own Mattermost push proxy server.
> - You must ensure that the push proxy can be reached on the correct port. For HPNS, it's port 443 from the Mattermost server.
> - Mattermost doesn't store any notification data. Any data being stored is at the server level only, such as the `device_id`, since the HPNS needs to know which device the notification must be sent to.

Test Push Notifications Service (TPNS)

Non-commercial and self-hosted customers can use Mattermost's free, basic Test Push Notifications Service (TPNS).

Note

- The TPNS isn't recommended for use in production environments, and doesn't offer production-level update service level agreements (SLAs).
- The TPNS isn't available for Mattermost Cloud deployments.
- The TPNS only works with the pre-built mobile apps that Mattermost deploys through the Apple App Store (https://www.apple.com/app-store/) and Google Play Store (https://play.google.com/store/games?hl=en). If you have built your own mobile apps, you must also host your own Mattermost push proxy service.
- You must ensure that the push proxy can be reached on the correct port. For TPNS, it's port 80 from the Mattermost server.
- If you don't need or want Mattermost to send mobile push notifications, disabling this configuration setting in larger deployments may improve server performance in the following areas:
  - Reduced Processing Load: Generating and sending push notifications requires processing power and resources. By disabling them, the server can allocate those resources to other tasks.
  - Decreased Network Traffic: Push notifications involve network communication. Disabling them reduces the amount of data being transferred, which can enhance overall network performance.
  - Lower Database Load: Each push notification may involve reading from and writing to the database. Reducing these operations decreases the load on the database, improving response times for other queries.
  - Faster Response Times: With fewer tasks to handle related to notifications, the system can respond faster to other requests from users, leading to a better user experience.
  - Simplified Error Handling: Push notification services can sometimes fail or have latency issues, requiring additional error handling. Disabling these notifications simplifies the system's operations.

- However, disabling push notifications can negatively impact user experience, communication efficiency, and overall productivity. It's important to balance performance improvements with the needs of your organization and users.

ID-only push notifications

Admins can enable mobile notifications to be fully private to protect a Mattermost customer against breaches in iOS and Android notification infrastructure by limiting the data sent to Apple and Google through a Mattermost configuration setting.

The standard way to send notifications to iOS and Android applications requires sending clear text messages to Apple or Google so they can be forwarded to a user's phone and displayed on iOS or Android. While Apple or Google assure the data is not collected or stored, should the organizations be breached or coerced, all standard mobile notifications on the platform could be compromised.

To avoid this risk, Mattermost can be configured to replace mobile notification text with message ID numbers that pass no information to Apple of Google. When received by the Mattermost mobile application on a user's phone, the message IDs are used to privately communicate with their Mattermost server and to retrieve mobile notification messages over an encrypted channel. This means that, at no time, is the message text visible to Apple or Google's message relay system. The contents of the message also won't reach Mattermost.

> Note
>
> Because of the extra steps to retrieve the notifications messages under Mattermost's private mobility capability with ID-only push notifications, end users may experience a slight delay before the mobile notification is fully displayed compared to sending clear text through Apple and Google's platform.

See our configuration settings documentation to learn more about the ID-only push notifications configuration setting. See our Mobile Apps FAQ documentation for details on using ID-only push notifications for data privacy.

Push notification server location

The physical location of the Mattermost Hosted Push Notification Service (HPNS) server. Select from **US (Default)** or **Germany** to automatically populate the **Push Notification Server** field server URL.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".EmailSettings.PushNotificationServer",` • Environment variable: `MM_EMAILSETTINGS_PUSHNOTIFICATIONSERVER`

Maximum notifications per channel

The maximum total number of users in a channel before @all, @here, and @channel no longer send desktop, email, or mobile push notifications to maximize performance. Numerical input. Default is **1000**.	• System Config path: **Environment > Push Notification Server** • `config.json setting`: `".TeamSettings.MaxNotificationsPerChannel: 1000",` • Environment variable: `MM_EMAILSETTINGS_MAXNOTIFICATIONSPERCHANNEL`

Note

- We recommend increasing this value a little at a time, monitoring system health by tracking performance monitoring metrics, and only increasing this value if large channels have restricted permissions controlling who can post to the channel, such as a read-only channel.
- Reducing this configuration setting value to **10** in larger deployments may improve server performance in the following areas:
  - Reduced Load on Notification System: Each notification generates a certain amount of computational and network load. By limiting the number of notifications per channel, the system processes fewer notifications, thereby reducing the load on servers.
  - Database Efficiency: Notifications are typically stored in a database. Fewer notifications mean less frequent database writes and reads, leading to quicker database operations and reduced latency.
  - Minimized Client Processing: Users' clients (e.g., desktop and mobile apps) have to fetch and process notifications. With fewer notifications, clients can operate more efficiently, reducing memory and CPU usage on users' devices.
  - Improved User Experience: An overload of notifications can lead to performance lags and a cluttered experience for users. Limiting the number ensures that users receive only the most important notifications, which can enhance usability and response times.
  - Network Bandwidth: High numbers of notifications can consume a lot of bandwidth, particularly if they are being sent to many users. Fewer notifications can lead to lower overall network usage and potentially faster delivery of critical messages.
  - Server Load Balancing: By reducing the number of notifications, the workload can be more evenly distributed across the servers, leading to better load balancing and preventing any single server from becoming a bottleneck.

On this page

- Enable push notifications
  - Hosted Push Notifications Service (HPNS)
  - Test Push Notifications Service (TPNS)
  - ID-only push notifications
- Push notification server location
- Maximum notifications per channel

Enable rate limiting

Rate limiting prevents your Mattermost server from being overloaded with too many requests, and decreases the risk and impact of third-party applications or malicious attacks on your server.

Configure rate limiting settings by going to **System Console > Environment > Rate Limiting**, or by editing the `config.json` file as described in the following tables. Changes to configuration settings in this section require a server restart before taking effect.

Important

Mattermost rate limiting configuration settings are intended for small deployments of Mattermost up to a few hundred users, and is not intended for larger, Enterprise-scale deployments.

Enable or disable rate limiting to throttle APIs to a specified number of requests per second. • **true**: APIs are throttled at the rate specified by the Maximum queries per second configuration setting. • **false**: **(Default)** API access isn't throttled.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.Enable: false",` • Environment variable: `MM_RATELIMITSETTINGS_ENABLE`

Maximum queries per second

Throttle the API at this number of requests per second when rate limiting is enabled. Numerical input. Default is **10**. Increase this value to accept more requests each second, and decrease this value to allow fewer requests.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.PerSec: 10,` • Environment variable: `MM_RATELIMITSETTINGS_PERSEC`

Maximum burst size

The maximum number of requests allowed beyond the per second query limit when rate limiting is enabled. Numerical input. Default is **100**. Increase this value to allow for more concurrent requests to be handled, and decrease this value to limit this capacity.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.MaxBurst: 100,` • Environment variable: `MM_RATELIMITSETTINGS_MAXBURST`

Memory store size

The maximum number of user sessions connected to the system as determined by vary rate limit settings when rate limiting is enabled. Numerical input. Default is **10000**. Typically set to the number of users in the system. We recommend setting this value to the expected number of users. A higher value may result in underutilized resources, and a lower value may result in user sessions/tokens expiring too frequently.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.MemoryStoreSize: 10000,` • Environment variable: `MM_RATELIMITSETTINGS_MEMORYSTORESIZE`

Vary rate limit by remote address

Configure Mattermost to rate limit API access by IP address when rate limiting is enabled. • **true**: **(Default)** Rate limit API access by IP address. Recommended when using a proxy. • **false**: Rate limiting does not vary by IP address.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByRemoteAddr: true,` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYREMOTEADDR`

Vary rate limit by user

Configure Mattermost to rate limit API access by authentication token or not when rate limiting is enabled. • **true**: Rate limit API access by user authentication token. Recommended when using a proxy. • **false**: **(Default)** Rate limiting does not vary by user authentication token.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByUser: false,` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYUSER`

Vary rate limit by HTTP header

Configure Mattermost to vary rate limiting API access by the HTTP header field specified. Recommended when you're using a proxy. • When configuring NGINX, set this to **X-Real-IP**. • When configuring AmazonELB, set this to **X-Forwarded-For**.	• System Config path: **Environment > Rate Limiting** • `config.json` setting: `".RateLimitSettings.VaryByHeader: "",` • Environment variable: `MM_RATELIMITSETTINGS_VARYBYHEADER`

Reporting configuration settings

View the following statistics for your overall deployment and specific teams, as well as access server logs, in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Reporting**:

- Site statistics
- Team statistics
- Server logs
- Statistics configuration settings

Site statistics

View statistics on a wide variety of activities in Mattermost, including: users, seats, teams, channels, posts, calls, sessions, commands, webhooks, websocket and database connections, and collaborative playbooks.	• System Config path: **Reporting > Site Statistics** • `config.json setting`: N/A • Environment variable: N/A

> Note
>
> - Bots, deactivated users, and synthetic users in Microsoft Teams integrations and connected workspaces users aren't counted towards the total number of activated users.
> - For billing purposes, activated guest accounts do consume a licensed seat, which is returned when the guest account is deactivated. This means that guest accounts count as a paid user in your Mattermost workspace

---

Team statistics

View statistics per team on number of activated users, number of public and private channels, total post count, and count of paid users (self-hosted only).	• System Config path: **Reporting > Team Statistics** • `config.json` setting: N/A • Environment variable: N/A

> Note
>
> Bots, deactivated users, and synthetic users in Microsoft Teams integrations and connected workspaces users aren't counted towards the total number of active users.

Server logs

Note

Available only for self-hosted deployments

View logging of server-side events. Reload data, download the `mattermost.log` file locally, and view full log event details for any log entry.	• System Config path: **Reporting > Server Logs** • `config.json` setting: N/A • Environment variable: N/A

> Tip
>
> From Mattermost v10.9, you can toggle between JSON and plain text server logs in the System Console when console log output is configured as JSON by specifying the log format as **JSON** or **Plain text**. This option is located in the top right corner of the page **Server logs** page.

Statistics configuration settings

Note

Available only for self-hosted deployments

The following configuration setting controls statistics collection behavior. This setting is not available in the System Console and can only be set in the `config.json` file.

Maximum users for statistics

Note

This setting is used to maximize performance for large Enterprise deployments and isn't available in the System Console and can only be set in `config.json`.

Set the maximum number of users on the server before statistics for total messages, total hashtag messages, total file messages, messages per day, and activated users with messages per day are disabled. Numerical input. Default is **2500** users.	• System Config path: N/A • `config.json` setting: `"AnalyticsSettings.MaxUsersForStatistics": 2500` • Environment variable: N/A

On this page

- Reporting configuration settings
  - Site statistics
  - Team statistics
  - Server logs
  - Statistics configuration settings
    - Maximum users for statistics

Self-hosted workspace edition and license settings

Start a trial or manage your self-hosted deployment by selecting the **Product**

menu, selecting **System Console**, and then selecting **About > Edition and License**.

Start a trial

You can start a trial of Mattermost Enterprise for free for 30 days. Select **Start trial** and fill out the form to begin.

> Note
>
> Once you've started a trial, you can't start another one with the same Mattermost instance. If you need more time to evaluate Mattermost Enterprise, contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/).

Manage your self-hosted deployment

You can also review and manage the following aspects of your self-hosted deployment:

- View the edition of your Mattermost self-hosted deployment.
- Manage your product subscription.
- Upload a new license.
- Remove a license to downgrade the server.

- Talk to a Mattermost Expert (https://mattermost.com/contact-sales/) for assistance.

On this page

- Self-hosted workspace edition and license settings
  - Start a trial
  - Manage your self-hosted deployment

Site configuration settings

Review and manage the following site configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **Site Configuration**:

- Customization
- Localization
- Users and Teams
- Notifications
- System-wide notifications
- Emoji
- Posts
- File Sharing and Downloads
- Public Links
- Notices
- Connected Workspaces

> Tip
>
> System admins managing a self-hosted Mattermost deployment can edit the `config.json` file as described in the following tables. Each configuration value below includes a JSON path to access the value programmatically in the `config.json` file using a JSON-aware tool. For example, the `SiteName` value is under `TeamSettings`.
>
> - If using a tool such as jq (https://stedolan.github.io/jq/), you'd enter: `cat config/config.json | jq '.TeamSettings.SiteName'`
> - When working with the `config.json` file manually, look for an object such as `TeamSettings`, then within that object, find the key `SiteName`.

Customization

Access the following configuration settings in the System Console by going to **Site Configuration > Customization**.

Site name

Name of the site shown in login screens and user interface. String input. Maximum 30 characters. Default is `Mattermost`	• System Config path: **Site Configuration > Customization** • `config.json` setting: `TeamSettings` > `SiteName` • Environment variable: `MM_TEAMSETTINGS_SITENAME`

Site description

Text displayed above the login form. When not specified, the phrase "Log in" is displayed. String input.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `TeamSettings` > `CustomDescriptionText` • Environment variable: `MM_TEAMSETTINGS_CUSTOMDESCRIPTIONTEXT`

Enable custom branding

• **true**: Enables the display of a custom image and text on the login page • **false**: **(Default)** Custom branding is disabled See also the custom brand image and custom brand text configuration settings for more branding options.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `TeamSettings` > `EnableCustomBrand` > `false` • Environment variable: `MM_TEAMSETTINGS_ENABLECUSTOMBRAND`

Custom brand image

A JPG image for display on the login page. The image **must** be uploaded through the System Console. There is no `config.json` setting. The file should be **smaller than 2 MB**. Enable custom branding must be set to **true** to display the image.	• System Config path: **Site Configuration > Customization** • `config.json` setting: N/A • Environment variable: N/A

Custom brand text

Text that will be shown below the **Custom brand image** on the login page. You can format this text using the same Markdown formatting as in Mattermost messages. String input. Maximum 500 characters. Enable custom branding must be set to **true** to display the text.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `TeamSettings` > `CustomBrandText` • Environment variable: `MM_TEAMSETTINGS_CUSTOMBRANDTEXT`

Enable Ask Community link

• **true**: **(Default)** A link to the [Mattermost Community (https://mattermost.com/community/)](https://mattermost.com/community/) appears as **Ask the community** under the **Help** menu in the channel header. • **false**: The link does not appear. The link does not display on mobile apps.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `EnableAskCommunityLink` > `true` • Environment variable: `MM_SUPPORTSETTINGS_ENABLEASKCOMMUNITYLINK`

Help link

This field sets the URL for the Help link on the login and sign up pages, as well as the **Help Resources** link under the **Help** menu in the channel header. String input. Default is `https://about.mattermost.com/default-help/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `HelpLink` • Environment variable: `MM_SUPPORTSETTINGS_HELPLINK`

> Note
>
> If this value is empty, the Help link is hidden on the login and sign up pages. However, the **Help Resources** link remains available under the **Help** menu.

Terms of Use link

Note

Available only for self-hosted deployments

This field sets the URL for the Terms of Use of a self-hosted site. A link to the terms appears at the bottom of the sign-up and login pages. The default URL links to a [Terms of Use (https://mattermost.com/terms-of-use/)](https://mattermost.com/terms-of-use/) page hosted on `mattermost.com`. This includes the Mattermost Acceptable Use Policy explaining the terms under which Mattermost software is provided to end users. If you change the default link to add your own terms, the new terms **must include a link** to the default terms so end users are aware of the Mattermost Acceptable Use Policy. String input. Default is `https://about.mattermost.com/default-terms/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `TermsOfServiceLink` • Environment variable: `MM_SUPPORTSETTINGS_TERMSOFSERVICELINK`

> Note
>
> This setting doesn't change the **Terms of Use** link in the **About Mattermost** window.

Privacy Policy link

Note

Available only for self-hosted deployments

This field sets the URL for the Privacy Policy of a self-hosted site. A link to the policy appears at the bottom of the sign-up and login pages. If this field is empty, the link does not appear. String input. Default is `https://about.mattermost.com/default-privacy-policy/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `PrivacyPolicyLink` • Environment variable: `MM_SUPPORTSETTINGS_PRIVACYPOLICYLINK`

> Note
>
> This setting does not change the **Privacy Policy** link in the **About Mattermost** window.

About link

Note

Available only for self-hosted deployments

This field sets the URL for a page containing general information about a self-hosted site. A link to the About page appears at the bottom of the sign-up and login pages. If this field is empty the link does not appear. String input. Default is `https://about.mattermost.com/default-about/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `AboutLink` • Environment variable: `MM_SUPPORTSETTINGS_ABOUTLINK`

Forgot Password custom link

When the **Forgot Password** link is enabled on the Mattermost login page, users are taken to a custom URL to recover or change their password. Leave this field blank to use Mattermost's Password Reset workflow.	• System Config path: **Site Configuration > Forgot password custom link** • `config.json` setting: `SupportSettings` > `ForgetPasswordLink` • Environment variable: `MM_SUPPORTSETTINGS_FORGETPASSWORDLINK`

> Note
>
> This configuration setting applies to all Mattermost clients, including web, desktop app, and mobile app. You can control whether the **Forgot Password** link is visible or hidden in clients by going to **Authentication > Password > Enable Forgot Password Link**. See the configuration documentation for details.

Report a Problem

Note

Available only for self-hosted deployments

Specify how the **Report a Problem** option behaves in the Mattermost app via the **Help** menu:

- **Default link**: Uses the default Mattermost URL to report a problem. For commercial customers, this is the [Mattermost Support Portal (https://support.mattermost.com/hc/en-us/requests/new)](https://support.mattermost.com/hc/en-us/requests/new). Non-commercial customers are directed to [create a new issue on the Mattermost GitHub repository (https://github.com/mattermost/mattermost/issues/new)](https://github.com/mattermost/mattermost/issues/new).

- **Email address**: Enables you to enter an email address that users will be prompted to send a message to when they choose **Report a Problem** in Mattermost.
- **Custom link**: Enables you to enter a URL that users will be directed to when they choose **Report a Problem** in Mattermost.
- **Hide link**: Removes the **Report a Problem** option from Mattermost.

Report a Problem link

Note

Available only for self-hosted deployments

This field sets the URL for the **Report a Problem** link in the channel header **Help** menu. If this field is empty the link does not appear. String input. Default is `https://mattermost.com/pl/report-a-bug`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `ReportAProblemLink` • Environment variable: `MM_SUPPORTSETTINGS_REPORTAPROBLEMLINK`

Report a Problem email address

Note

Available only for self-hosted deployments

This field sets the email address for the **Report a Problem** link in the channel header **Help** menu. String input. Cannot be left blank.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `ReportAProblemMail` • Environment variable: `MM_SUPPORTSETTINGS_REPORTAPROBLMEMAIL`

Allow mobile app log downloads

Note

Available only for self-hosted deployments

Enable users to download mobile app logs for troubleshooting. When the **Report a Problem** link is shown, mobile logs can be downloaded as part of the reporting flow. • **true** (**Default**): Users can download mobile app logs. • **false** Users can't download mobile app logs.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `SupportSettings` > `AllowDownloadLogs` • Environment variable: `MM_SUPPORTSETTINGS_ALLOWDOWNLOADLOGS`

Mattermost apps download page link

Note

Available only for self-hosted deployments

This field sets the URL for the Download Apps link in the **Product** menu. If this field is empty, the link does not appear. If you have an Enterprise App Store, set the link to the appropriate download page for your Mattermost apps. String input. Default is `https://mattermost.com/pl/download-apps`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `NativeAppSettings` > `AppDownloadLink` • Environment variable: `MM_NATIVEAPPSETTINGS_APPDOWNLOADLINK`

Android app download link

Note

Available only for self-hosted deployments

This field sets the URL to download the Mattermost Android app. Users who access the Mattermost site on a mobile browser will be prompted to download the app through this link. If this field is empty, the prompt does not appear. If you have an Enterprise App Store, link to your Android app. String input. Default is `https://mattermost.com/pl/android-app/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `NativeAppSettings` > `AndroidAppDownloadLink` • Environment variable: `MM_NATIVEAPPSETTINGS_ANDROIDAPPDOWNLOADLINK`

iOS app download link

Note

Available only for self-hosted deployments

This field sets the URL to download the Mattermost iOS app. Users who access the site on a mobile browser will be prompted to download the app through this link. If this field is empty, the prompt does not appear. If you use an Enterprise App Store, link to your iOS app. String input. Default is `https://mattermost.com/pl/ios-app/`.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `NativeAppSettings` > `IosAppDownloadLink` • Environment variable: `MM_NATIVEAPPSETTINGS_IOSAPPDOWNLOADLINK`

Enable desktop app landing page

• **true**: **(Default)** Prompts users to use the desktop app. • **false**: Doesn't prompt users to use the desktop app.	• System Config path: **Site Configuration > Customization** • `config.json` setting: `ServiceSettings` > `EnableDesktopLandingPage` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLEDESKTOPLANDINGPAGE`

App custom URL schemes

This setting isn't available in the System Console and can only be set in `config.json`.

Define valid custom URL schemes for redirect links provided by custom-built mobile Mattermost apps. This ensures users are redirected to the custom-built mobile app and not Mattermost's mobile client.

When configured, after OAuth or SAML user authentication is complete, custom URL schemes sent by mobile clients are validated to ensure they don't include default schemes such as `http` or `https`. Mobile users are then redirected back to the mobile app using the custom scheme URL provided by the mobile client. We recommend that you update your mobile client values as well with valid custom URL schemes.

This feature's `config.json` setting is `"NativeAppSettings.AppCustomURLSchemes"` with an array of strings as input separated by spaces.
For example: • `MM_NativeAppSettings_AppCustomURLSchemes = mmauth:// mmauthbeta://` • Via mmctl: `mmctl config set NativeAppSettings.AppCustomURLSchemes "mmauth://" "mmauthbeta://"`

Mobile external browser

Note

Available only for self-hosted deployments

From Mattermost v10.2 and Mobile v2.2.1, this setting configures the mobile app to use an external mobile browser to perform SSO authentication. • **true**: The mobile app uses the default internal mobile browser to perform SSO authentication. • **false**: **(Default)** The mobile app uses an external mobile browser to perform SSO authentication.	• System Config path: N/A • `config.json` setting: `NativeAppSettings.MobileExternalBrowser` • Environment variable: `MM_NATIVEAPPSETTINGS_MOBILEEXTERNALBROWSER`

Enable this configuration setting when there are issues with the mobile app SSO redirect flow.

Localization

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Localization**. Changes to configuration settings in this section require a server restart before taking effect.

Default server language

The default language for system messages and logs. Options: `"bg"`, `"de"`, `"en"`, `"en-AU"`, `"es"`, `"fa"`, `"fr"`, `"hu"`, `"it"`, `"ja"`, `"ko"`, `"nl"`, `"pl"`, `"pt-br"`, `"ro"`, `"ru"`, `"sv"`, `"tr"`, `"uk"`, `"vi"`, `"zh-Hans"`, and `"zh-Hant"`. Default is `"en"`.	• System Config path: **Site Configuration > Localization** • `config.json` setting: `LocalizationSettings` > `DefaultServerLocale` • Environment variable: `MM_LOCALIZATIONSETTINGS_DEFAULTSERVERLOCALE`

> Note
>
> Changing this configuration setting changes the default server language for users who haven't set a language preference via **Settings**. Mattermost applies the user's language preference when specified.

Default client language

The default language for new users and pages where the user isn't logged in. Options: `"bg"`, `"de"`, `"en"`, `"en-AU"`, `"es"`, `"fa"`, `"fr"`, `"hu"`, `"it"`, `"ja"`, `"ko"`, `"nl"`, `"pl"`, `"pt-br"`, `"ro"`, `"ru"`, `"sv"`, `"tr"`, `"uk"`, `"vi"`, `"zh-Hans"`, and `"zh-Hant"`. Default is `"en"`.	• System Config path: **Site Configuration > Localization** • `config.json` setting: `LocalizationSettings` > `DefaultClientLocale` • Environment variable: `MM_LOCALIZATIONSETTINGS_DEFAULTCLIENTLOCALE`

> Note
>
> Changing this configuration setting changes the default client language for users who haven't set a language preference via **Settings**. Mattermost applies the user's language preference when specified.

Available languages

Sets the list of languages users see under **Settings > Display > Language**. If this field is left blank, users see all supported languages. Newly supported languages are added automatically. If this field is not blank, it must contain the **Default client language**, in addition to any other languages. For example, to limit the language choices to US English and Español (es), the string would be `"en,es"`. Options: `"bg"`, `"de"`, `"en"`, `"en-AU"`, `"es"`, `"fa"`, `"fr"`, `"hu"`, `"it"`, `"ja"`, `"ko"`, `"nl"`, `"pl"`, `"pt-br"`, `"ro"`, `"ru"`, `"sv"`, `"tr"`, `"uk"`, `"vi"`, `"zh-Hans"`, and `"zh-Hant"`. Default is `"en"`.	• System Config path: **Site Configuration > Localization** • `config.json` setting: `LocalizationSettings` > `AvailableLocales` • Environment variable: `MM_LOCALIZATIONSETTINGS_AVAILABLELOCALES`

Enable experimental locales

Enable work in progress languages in Mattermost to review translations and identify translation gaps.

• **true**: Work in progress languages are available in Mattermost in addition to officially supported languages. • **false**: **(Default)** Only officially supported languages are available in Mattermost.	• System Config path: **Site Configuration > Localization** • `config.json` setting: `LocalizationSettings` > `EnableExperimentalLocales` > `false` • Environment variable: `MM_LOCALIZATIONETTINGS_ENABLEEXPERIMENTALLOCALES`

Note

- Cloud system admins can request this configuration setting to be enabled for their instance by contacting their Mattermost Account Manager.
- Work in progress languages may be incomplete. Strings missing translations display in US English.
- Currently, only web and desktop app product strings are impacted by this configuration setting. Server and mobile product strings aren't impacted by this setting.
- See the language documentation for details on selecting a language preference in Mattermost.

Users and teams

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Users and Teams**.

Max users per team

The **Max users per team** is the maximum total number of users per team, including activated and deactivated users. In Mattermost, a team of people should be a small organization with a specific goal. In the physical world, a team could sit around a single table. The default maximum (50) should be enough for most teams, but with appropriate [hardware (https://docs.mattermost.com/install/software-hardware-requirements.html)](https://docs.mattermost.com/install/software-hardware-requirements.html), this limit can be increased to thousands of users. Channels are another way of organizing communications within teams on various topics. Numerical input. Default is **50** self-hosted deployments, and **10000** for Cloud deployments.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `MaxUsersPerTeam` > `50` • Environment variable: `MM_TEAMSETTINGS_MAXUSERSPERTEAM`

Max channels per team

The maximum number of channels per team, including both active and archived channels. Numerical input. Default is **2000** for self-hosted deployments, and **10000** for Cloud deployments.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `MaxChannelsPerTeam` > `2000` • Environment variable: `MM_TEAMSETTINGS_MAXCHANNELSPERTEAM`

Enable join/leave messages by default

Specify the default configuration of system messages displayed when users join or leave channels. • **true**: **(Default)** Join/Leave messages are displayed. • **false**: Join/Leave messages are hidden. Users can override this default by going to **Settings > Advanced > Enable Join/Leave Messages**.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `EnableJoinLeaveMessageByDefault` > `true` • Environment variable: `MM_TEAMSETTINGS_ENABLEJOINLEAVEMESSAGEBYDEFAULT`

Enable users to open direct message channels with

This setting determines whether a user can open a direct message channel with anyone on the Mattermost server or only to members of the same team. This setting only affects the options presented in the user interface. It does not affect permissions on the backend server. • **Any user on the Mattermost server**: **(Default)** Users can send a direct message to any user through the **Direct Messages > More** menu. `config.json` setting: `"any"` • **Any member of the team**: The **Direct Messages > More** menu only allows direct messages to users on the same team. Pressing `Ctrl K` on Windows or Linux, or `⌘ K` on Mac, only lists other users on the team currently being viewed. A user who is a member of multiple teams can only send direct messages to the team that is being viewed. However, the user can receive messages from other teams, regardless of the team currently being viewed. `config.json` setting: `"team"`	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `RestrictDirectMessage` • Environment variable: `MM_TEAMSETTINGS_RESTRICTDIRECTMESSAGE`

Teammate name display

This setting determines how names appear in posts and under the **Direct Messages** list. Users can change this setting in their interface under **Settings > Display > Teammate Name Display**, unless this setting is locked by a system admin via the **Lock teammate name display for all users** configuration setting. • **Show username**: **(Default for self-hosted deployments)** Displays usernames. `config.json` option: `"username"`. • **Show nickname if one exists...**: Displays the user's nickname. If the user doesn't have a nickname, their full name is displayed. If the user doesn't have a full name, their username is displayed. `config.json` option: `"nickname_full_name"`. • **Show first and last name**: **(Default for Cloud deployments)** Displays user's full name. If the user doesn't have a full name, their username is displayed. Recommended when using SAML or LDAP if first name and last name attributes are configured. `config.json` option: `"full_name"`.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `TeammateNameDisplay` > `username` • Environment variable: `MM_TEAMSETTINGS_TEAMMATENAMEDISPLAY`

Lock teammate name display for all users

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

This setting controls whether users can change settings under **Settings > Display > Teammate Name Display**. • **true**: Users **cannot** change the Teammate Name Display. • **false**: **(Default)** Users can change the Teammate Name Display setting.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `LockTeammateNameDisplay` > `false` • Environment variable: `MM_TEAMSETTINGS_LOCKTEAMMATENAMEDISPLAY`

Allow users to view archived channels

• **true**: **(Default)** Allows users to access the content of archived channels of which they were a member. • **false**: Users are unable to access content in archived channels.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `ExperimentalViewArchivedChannels` > `true` • Environment variable: `MM_TEAMSETTINGS_EXPERIMENTALVIEWARCHIVEDCHANNELS`

> Note
>
> Cloud admins can't modify this configuration setting.

Show email address

• **true**: **(Default)** All users can see the email addresses of every other user. • **false**: Hides email addresses in the client user interface, except from system admins and the System Roles with read/write access to Compliance, Billing, or User Management (users/teams/channels/groups etc).	• System Config path: **Site Configuration > Users and teams** • `config.json` setting: `PrivacySettings` > `ShowEmailAddress` > `true` • Environment variable: `MM_PRIVACYSETTINGS_SHOWEMAILADDRESS`

Show full name

• **true**: **(Default)** Full names are visible to all users in the client user interface. • **false**: Hides full names from all users, except system admins. Username is shown in place of the full name.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `PrivacySettings` > `ShowFullName` > `true` • Environment variable: `MM_PRIVACYSETTINGS_SHOWFULLNAME`

Enable custom user statuses

• **true**: **(Default)** Users can set status messages and emojis that are visible to all users. • **false**: Users cannot set custom statuses.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `EnableCustomUserStatuses` > `true` • Environment variable: `MM_TEAMSETTINGS_ENABLECUSTOMUSERSTATUSES`

Enable last active time

• **true**: **(Default)** Users can see when deactivated users were last active on a user's profile and in direct message channel headers. • **false**: Users can't see when deactivated users were last online.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `TeamSettings` > `EnableLastActiveTime` > `true` • Environment variable: `MM_TEAMSETTINGS_ENABLELASTACTIVETIME`

Enable custom user groups

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **true**: **(Default)** Users with appropriate permissions can create custom user groups, and users can @mention custom user groups in Mattermost conversations. • **false**: Custom user groups cannot be created.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `ServiceSettings` > `EnableCustomGroups` > `true` • Environment variable: `MM_SERVICESETTINGS.ENABLECUSTOMGROUPS`

User statistics update time

Set the server time for updating the user post statistics, including each user's total message count, and the timestamp of each user's most recently sent message. Must be a 24-hour time stamp in the form `HH:MM` based on the local time of the server. Default is **00:00**.	• System Config path: **Site Configuration > Users and Teams** • `config.json` setting: `ServiceSettings` > `RefreshPostStatsRunTime` > `00:00` • Environment variable: `MM_SERVICESETTINGS.REFRESHPOSTSTATSRUNTIME`

Notifications

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Notifications**.

Show @channel, @all, or @here confirmation dialog

• **true**: **(Default)** Requires users to confirm when posting @channel, @all, @here, or group mentions in channels with more than 5 members. • **false**: No confirmation is required.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `TeamSettings` > `EnableConfirmNotificationsToChannel` > `true` • Environment variable: `MM_TEAMSETTINGS_ENABLECONFIRMNOTIFICATIONSTOCHANNEL`

Enable email notifications

• **true**: **(Default)** Enables automatic email notifications for posts. • **false**: Disables notifications. A developer may choose this option to speed development by skipping email setup (see also the **Enable preview mode banner** setting).	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `SendEmailNotifications` > `ture` • Environment variable: `MM_EMAILSETTINGS_SENDEMAILNOTIFICATIONS`

Note

- Cloud admins can't modify this configuration setting.
- If this setting is **false**, and the SMTP server is set up, account-related emails (such as authentication messages) will be sent regardless of this setting.
- Email invitations and account deactivation emails aren't affected by this setting.
- If you don't plan on configuring Mattermost for email, disabling this configuration setting in larger deployments may improve server performance in the following areas, particularly in high-traffic environments where performance is a key concern:
  - **Reduced Server Load**: Generating and sending emails requires processing power and resources. By disabling email notifications, you reduce the load on the server, which can be reallocated to other tasks.
  - **Decreased I/O Operations**: Sending emails involves input/output (I/O) operations, such as writing to logs and databases, and handling communication with the email server. Reducing these I/O operations can improve overall system efficiency.
  - **Lowered Network Traffic**: Each email sent contributes to network traffic. Disabling email notifications decreases the amount of data being transmitted, which can lead to better performance, especially in environments with limited bandwidth.
  - **Faster Response Times**: With fewer background tasks (like sending emails) to handle, the application can potentially respond to user requests more quickly, improving perceived performance.
  - **Resource Allocation**: Resources like CPU cycles, memory, and network bandwidth that would have been used for sending emails can be used elsewhere, possibly improving the performance of other critical components of the system.
  - However, disabling email notifications can negatively impact user experience, communication efficiency, and overall productivity. It's

> important to balance performance improvements with the needs of your organization and users.

Enable preview mode banner

• **true**: **(Default)** When **Send email notifications** is **false**, users see the Preview Mode banner. This banner alerts users that email notifications are disabled. • **false**: Preview Mode banner does not appear.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `EnablePreviewModeBanner` > `true` • Environment variable: `MM_EMAILSETTINGS_ENABLEPREVIEWMODEBANNER`

> Note
>
> Cloud admins can’t modify this configuration setting.

Enable email batching

• **true**: Multiple email notifications for mentions and direct messages over a given time period are batched into a single email. • **false**: **(Default)** Email notifications are sent for each mention or direct message.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `EnableEmailBatching` > `false` • Environment variable: `MM_EMAILSETTINGS_ENABLEEMAILBATCHING`

Note

- Cloud admins can’t modify this configuration setting.
- The Site Url and SMTP Email Server must be configured to allow email batching.
- Regardless of how this setting is configured, users can disable email-based notifications altogether.
- When email batching is enabled, users can customize how often to receive batched notifications. The default frequency is 15 minutes.
- Email batching in High Availability Mode is planned, but not yet supported.

Email notification contents

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

• **Send full message contents**: **(Default)** Email notifications include the full message contents, along with the name of the sender and the channel. `config.json` setting: `"full"` • **Send generic description with only sender name**: Only the name of the sender and team name are included in email notifications. Use this option if Mattermost contains confidential information and policy dictates it cannot be stored in email. `config.json` setting: `"generic"`	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `EmailNotificationContentsType` • Environment variable: `MM_EMAILSETTINGS_EMAILNOTIFICATIONCONTENTSTYPE`

Notification display name

Display name for email notifications sent from the Mattermost system. String input. No default setting. This field is required when changing settings in the System Console.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `FeedbackName` • Environment variable: `MM_EMAILSETTINGS_FEEDBACKNAME`

Notification from address

Email address for notification emails from the Mattermost system. This address should be monitored by a system admin. String input. Default is `test@example.com`. This field is required when changing settings in the System Console.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `FeedbackEmail` • Environment variable: `MM_EMAILSETTINGS_FEEDBACKEMAIL`

> Note
>
> Cloud admins can't modify this configuration setting.

Support email address

Sets a user support (or feedback) email address that is displayed on email notifications and during the Getting Started tutorial. This address should be monitored by a system admin. If no value is set, email notifications will not contain a way for users to request assistance. String input. Default is `feedback@mattermost.com`. This field is required when changing settings in the System Console.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `SupportSettings` > `SupportEmail` • Environment variable: `MM_SUPPORTSETTINGS_SUPPORTEMAIL`

Notification reply-to address

Email address used in the reply-to header when sending notification emails from the Mattermost system. This address should be monitored by a system admin. String input. Default is `test@example.com`.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `ReplyToAddress` • Environment variable: `MM_EMAILSETTINGS_REPLYTOADDRESS`

Notification footer mailing address

Optional setting to include the organization's name and mailing address in the footer of email notifications. If not set, nothing will appear. String input.	• System Config path: **Site Configuration > Notifications** • `config.json` setting: `EmailSettings` > `FeedbackOrganization` • Environment variable: `MM_EMAILSETTINGS_FEEDBACKORGANIZATION`

Push notification contents

- **Generic description with only sender name**: Push notifications include the sender's name, but not the channel name or message contents. `config.json` setting: `"generic_no_channel"`
- **Generic description with sender and channel names**: **(Default)** Push notifications include the name of the sender and channel, but not the message contents. `config.json` setting: `"generic"`
- **Full message content sent in the notification payload**: Includes the message contents in the push notification payload, which may be sent through [Apple's Push Notification service (https://developer.apple.com/documentation/usernotifications)](https://developer.apple.com/documentation/usernotifications) or Google's Firebase Cloud Messaging . We **highly recommended** this option only be used with an `https` protocol to encrypt the connection and protect confidential information. `config.json` setting: `"full"`
- **Full message content fetched from the server on receipt** (*Available in Mattermost Enterprise*): The

- System Config path: **Site Configuration > Notifications**
- `config.json` setting: `EmailSettings` > `PushNotificationContents`
- Environment variable: `MM_EMAILSETTINGS_PUSHNOTIFICATIONCONTENTS`

notification payload contains no message content. Instead it contains a unique message ID used to fetch message content from the Mattermost server when a push notification is received via a notification service app extension (https://developer.apple.com/documentation/usernotifications/modifying-content-in-newly-delivered-notifications) on iOS or an expandable notification pattern (https://developer.android.com/develop/ui/views/notifications/expanded) on Android.

If the server cannot be reached, a generic push notification is displayed without message content or sender name. For customers who wrap the Mattermost mobile application in a secure container, the container must fetch the message contents using the unique message ID when push notifications are received.

If the container is unable to execute the fetch, the push notification contents cannot be received by the customer's mobile application without passing the message contents through Apple's or Google's notification service.

`config.json` setting: `"id_loaded"`	

Enable notification monitoring

Enable or disable notification metrics data collection. • **true**: **(Default)** Mattermost notification data collection is enabled for client-side web and desktop app users. • **false**: Mattermost notification data collection is disabled.	• System Config path: **Environment > Performance Monitoring** • `config.json` setting: `MetricsSettings` > `EnableNotificationMetrics` > `true` • Environment variable: `MM_METRICSSETTINGS_ENABLENOTIFICATIONMETRICS`

> Note
>
> See the performance monitoring documentation to learn more about Mattermost Notification Health metrics.

System-wide notifications

Available onEnterprise and Professional plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > System-wide notifications**.

Enable system-wide notifications

• **true**: Enable system-wide notifications to display at the top of the Mattermost interface for all users across all teams. • **false**: **(Default)** Disable system-wide notifications.	• System Config path: **Site Configuration > System-wide notifications** • `config.json` setting: `AnnouncementSettings` > `SystemWideNotifications` > `false` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_SYSTEMWIDENOTIFICATIONS`

Banner text

The text of the system-wide notification, when enabled. String input.	• System Config path: **Site Configuration > System-wide notifications** • `config.json` setting: `AnnouncementSettings` > `BannerText` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_BANNERTEXT`

Banner color

The background color of system-wide notifications. String input of a CSS color value.	• System Config path: **Site Configuration > System-wide notifications** • `config.json` setting: `AnnouncementSettings` > `BannerColor` > `"#f2a93b"` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_BANNERCOLOR`

Banner text color

The color of the text in system-wide notifications. String input of a CSS color value.	• System Config path: **Site Configuration > System-wide notifications** • `config.json` setting: `AnnouncementSettings` > `BannerTextColor` > `"#333333"` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_BANNERTEXTCOLOR`

Allow banner dismissal

• **true**: **(Default)** Users can dismiss the system-wide notification. It will re-appear the next time the user logs in, and when the text is updated by an admin, or when an admin disables system-wide notifications and reenables them. • **false**: Users cannot dismiss the banner.	• System Config path: **Site Configuration > System-wide notifications** • `config.json` setting: `AnnouncementSettings` > `AllowBannerDismissal` > `true` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_ALLOWBANNERDISMISSAL`

Emoji

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Emoji**.

Enable emoji picker

• **true**: **(Default)** Enables an emoji picker when composing messages and for message reactions. • **false**: Disables the emoji picker in message composition and reactions.	• System Config path: **Site Configuration > Emoji** • `config.json` setting: `ServiceSettings` > `EnableEmojiPicker` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLEEMOJIPICKER`

Enable custom emoji

• **true**: **(Default)** Allows users to add up to 6000 emojis through a **Custom Emoji** option in the emoji picker. Emojis can be GIF, PNG, or JPG files up to 512 KB in size. • **false**: Disables custom emojis.	• System Config path: **Site Configuration > Emoji** • `config.json` setting: `ServiceSettings` > `EnableCustomEmoji` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLECUSTOMEMOJI`

> Note
>
> While Mattermost supports up to 6000 custom emojis, an increase in custom emojis can slow your server's performance.

Posts

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Posts**.

Automatically follow threads

Note

Available only for self-hosted deployments

• **true**: **(Default)** Enables automatic following for all threads that a user starts, or in which the user participates or is mentioned. A **Threads** table in the database tracks threads and thread participants. A **ThreadMembership** table tracks followed threads for each user and whether the thread is read or unread. • **false**: Disables automatic following of threads.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `ThreadAutoFollow` > `true` • Environment variable: `MM_SERVICESETTINGS_THREADAUTOFOLLOW`

> Note
>
> - This setting **must** be enabled for threaded discussions to function.
> - Enabling this setting does not automatically follow threads based on previous user actions. For example, threads a user participated in prior to enabling this setting won't be automatically followed, unless the user adds a new comment or is mentioned in the thread.

Threaded discussions

Important

Customers upgrading from a legacy Mattermost release prior to v7.0 must review the [administrator's guide to enabling threaded discussions (https://support.mattermost.com/hc/en-us/articles/6880701948564-Administrator-s-](https://support.mattermost.com/hc/en-us/articles/6880701948564-Administrator-s-)

guide-to-enabling-Collapsed-Reply-Threads) (formally known as Collapsed Reply Threads) prior to enabling this functionality.

• **Always On**: **(Default)** Enables threaded discussions on the server and for all users. This is the recommended configuration for optimal user experience and to ensure consistency in how users read and respond to threaded conversations. `config.json` setting: `"always_on"` • **Default On**: Enables threaded discussions on the server and for all users. • **Default Off**: Enables threaded discussions on the server but **not** for users. • **Disabled**: Users cannot enable threaded discussions. `config.json` setting: `"disabled"`	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `CollapsedThreads` • Environment variable: `MM_SERVICESETTINGS_COLLAPSEDTHREADS`

Message priority

> Tip
>
> Mattermost Enterprise or Professional (https://mattermost.com/pricing) customers can additionally request message acknowledgements to track that specific, time-sensitive messages have been seen and actioned. See the message priority documentation to learn more.

• **true**: **(Default)** Enables message priority for all users which enables them to set a visual indiciator for important or urgent root messages. • **false**: Disables the ability to set message priority and request acknowledgements.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `PostPriority` > `true` • Environment variable: `MM_SERVICESETTINGS_POSTPRIORITY`

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas, particularly in environments where performance and responsiveness are critical:

- Simplified Processing: When post priority is enabled, the system has to manage and prioritize posts based on their designated priority levels. This adds additional processing overhead as the system must evaluate and sort posts accordingly. By disabling this feature, all posts are treated equally, which simplifies the processing logic and reduces the computational load.
- Reduced Latency: With post priority enabled, there might be delays introduced while the system determines the priority of each post and processes them in the correct order. Disabling post priority can lead to more consistent and potentially quicker handling of posts because the system processes them on a first-come, first-served basis.
- Lower Resource Utilization: Managing post priorities can consume additional system resources such as CPU and memory. Disabling this feature can free up these resources, allowing the system to allocate them to other tasks, thereby improving overall performance.
- Improved Scalability: In a high-traffic environment, the complexity of managing post priorities can become more pronounced. Disabling this feature simplifies the system's operations, making it easier to scale as the number of users and posts increases.

Persistent notifications

Available onEnterprise and Professional plansCloudandself-hosteddeployments

• **true**: **(Default)** Users can trigger repeating notifications to mentioned recipients of urgent messages. • **false**: Disables the ability to send repeating notifications.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `AllowPersistentNotifications` > `true` • Environment variable: `MM_SERVICESETTINGS_ALLOWPERSISTENTNOTIFICATIONS`

Maximum number of recipients for persistent notifications

Available onEnterprise and Professional plansCloudandself-hosteddeployments

The maximum number of recipients users may send persistent notifications to. Numerical input. Default is **5**.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `PersistentNotificationMaxRecipients` > `5` • Environment variable: `MM_SERVICESETTINGS_PERSISTENTNOTIFICATIONMAXRECIPIENTS`

Frequency of persistent notifications

Available onEnterprise and Professional plansCloudandself-hosteddeployments

The number of minutes between repeated notifications for urgent messages sent with persistent notifications. Numerical input. Default is **5**. Minimum is **2**.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `PersistentNotificationIntervalMinutes` > `5` • Environment variable: `MM_SERVICESETTINGS_PERSISTENTNOTIFICATIONINTERVALMINUTES`

Total number of persistent notifications per post

Available onEnterprise and Professional plansCloudandself-hosteddeployments

The maximum number of times users may receive persistent notifications. Numerical input. Default is **6**.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `PersistentNotificationMaxCount` > `6` • Environment variable: `MM_SERVICESETTINGS_PERSISTENTNOTIFICATIONMAXCOUNT`

Enable website link previews

Important

The server must be connected to the internet to generate previews. This connection can be established through a firewall or outbound proxy if necessary.

• **true**: The server generates a preview of the first website, image, or YouTube video linked in a message. Users can disable website previews, but not image or YouTube previews, under **Settings > Display > Website Link Previews**. • **false**: **(Default)** All previews are disabled and the server does not request metadata for any links contained in messages.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `EnableLinkPreviews` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLELINKPREVIEWS`

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas:

- Reduced Network Requests: When link previews are enabled, the system needs to fetch metadata (such as title, description, or image) from the linked webpage. This requires additional network requests, which can slow down the system.
- Lower Server Load: Creating link previews involves parsing the content of the linked pages. If many users are sharing links, the server will have to perform numerous network requests and process a lot of additional data, increasing the load on the server.
- Less Data Processing: Every link shared needs to be processed to extract the necessary preview information. This processing consumes CPU and memory resources, which can otherwise be reserved for other tasks.
- Decreased Client-Side Rendering Time: On the client side, rendering link previews (adding text, images, and layouts) takes time and resources. Disabling link previews means that clients do not need to render these elements, leading to faster message display.
- Saved Bandwidth: Link previews often include images and other data from the linked content. By disabling them, you save the bandwidth that would be used to download these additional resources.
- However, disabling link previews can negatively impact user experience, communication efficiency, and overall productivity. It’s important to balance performance improvements with the needs of your organization and users.

Disable link previews for specific domains

Use this setting to disable previews of links for specific domains. String input of a comma-separated list of domains, for example: `"mattermost.com, images.example.com"`	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `RestrictLinkPreviews` • Environment variable: `MM_SERVICESETTINGS_RESTRICTLINKPREVIEWS`

Enable message link previews

• **true**: **(Default)** Share links to Mattermost messages will generate a preview for any users that have access to the original message. • **false**: Share links do not generate a preview.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `EnablePermalinkPreviews` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLEPERMALINKPREVIEWS`

Note

Disabling this configuration setting in larger deployments may improve server performance in the following areas, particularly in environments with high message throughput or limited resources:

- Reduced Server Load: When permalink previews are enabled, the server has to generate preview summaries for each shared link. This generates additional requests to fetch metadata and may involve parsing web pages, which increases the processing load on the server.
- Less Data Transfer: Permalink previews include additional metadata such as images, titles, and descriptions. Disabling previews reduces the amount of data that needs to be transferred, which can decrease bandwidth usage and improve message load times, particularly for channels with a high volume of links.
- Faster Message Rendering: On the client-side, rendering messages with multimedia previews takes more time compared to plain text messages. Disabling previews can reduce rendering complexity and improve client performance, especially on devices with limited resources.
- Network Latency: Fetching metadata for link previews may introduce network latency, as the server must reach out to external resources. Disabling this can eliminate these delays, ensuring faster message processing and display.
- Simplified Message Handling: In the absence of previews, messages are simpler and less resource-intensive to store, retrieve, and display. This can contribute to overall improved system responsiveness and efficiency.
- However, disabling permalink previews can negatively impact user experience, communication efficiency, and overall productivity. It’s important to balance performance improvements with the needs of your organization and users.

Enable SVGs

• **true**: Enables previews of SVG files attached to messages. • **false**: **(Default)** Disables previews of SVG files.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `EnableSVGs` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLESVGS`

> Warning
>
> Enabling SVGs is not recommended in environments where not all users are trusted.

Enable LaTeX code block rendering

• **true**: Enables rendering of LaTeX in code blocks. • **false**: **(Default)** Disables rendering in blocks. Instead, LaTeX code is highlighted.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `EnableLatex` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLELATEX`

> Warning
>
> Enabling LaTeX rendering is not recommended in environments where not all users are trusted.

Enable inline LaTeX rendering

• **true**: Enables rendering of LaTeX in message text. • **false**: **(Default)** Disables inline rendering of LaTeX. Instead, LaTeX in message text is highlighted. LaTeX can also be rendered in a code block, if that feature is enabled. See **Enable LaTeX code block rendering**.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `EnableInlineLatex` > `false` • Environment variable: `MM_SERVICESETTINGS_ENABLEINLINELATEX`

> Warning
>
> Enabling LaTeX rendering isn't recommended in environments where not all users are trusted.

Custom URL schemes

A list of URL schemes that will automatically create a link in message text, for example: `["git", "smtp"]`. These schemes always create links: `http`, `https`, `ftp`, `tel`, and `mailto`. `config.json` setting: an array of strings	• System Config path: **Site Configuration > Posts** • `config.json` setting: `DisplaySettings` > `CustomURLSchemes` > `[]` • Environment variable: `MM_DISPLAYSETTINGS_CUSTOMURLSCHEMES`

Maximum Markdown nodes

The maximum number of Markdown elements (such as emojis, links, or table cells), that can be included in a single piece of text in a message. Numerical input. Default is **0** which applies a Mattermost-specified limit.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `DisplaySettings` > `MaxMarkdownNodes` > `0` • Environment variable: `MM_DISPLAYSETTINGS_MAXMARKDOWNNODES`

> Note
>
> This limit applies to all Mattermost clients, including web, desktop app, and mobile app.

Google API key

Note

Available only for self-hosted deployments

If a key is provided in this setting, Mattermost displays titles of embedded YouTube videos and detects if a video is no longer available. Setting a key should also prevent Google from throttling access to embedded videos that receive a high number of views. String input.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `GoogleDeveloperKey` • Environment variable: `MM_SERVICESETTINGS_GOOGLEDEVELOPERKEY`

> Note
>
> This key is used in client-side Javascript, and must have the YouTube Data API added as a service.

Enable server syncing of message drafts

Enable or disable the ability to synchronize draft messages across all supported Mattermost clients. • **true**: **(Default)** Message drafts are saved on the server and may be accessed from different clients. Users may still disable server synchronization of draft messages by going to **Settings > Advanced Settings**. • **false**: Draft messages are stored locally on each device.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `AllowSyncedDrafts` > `true` • Environment variable: `MM_SERVICESETTINGS_ALLOWSYNCEDDRAFTS`

Note

While drafts can be very useful for maintaining work continuity, especially in collaborative environments, disabling draft synchronization across devices can lead to noticeable performance improvements by reducing the computational and data management overhead as follows:

- Reduced Data Synchronization: When drafts are enabled and synchronized across devices, the system needs to handle those data synchronization operations which can consume significant bandwidth and computing resources. Disabling draft syncing reduces the load on servers and networks.
- Lower Storage Usage: Storing drafts requires additional database operations and storage space. Each draft is an extra piece of data that needs to be saved, managed, and retrieved. Without drafts, the system has fewer records to keep, which can streamline database operations.
- Decreased Client Processing: On the client side, draft management involves monitoring changes, saving drafts periodically, and handling conflict resolution if multiple drafts are edited from different devices. Disabling drafts reduces these client-side processes, thus freeing up memory and CPU resources.
- Simplified Architecture: Maintaining synced drafts often requires complex backend logic to ensure consistency and avoid data conflicts. Simplifying this architecture by removing draft syncing can lead to more efficient and faster backend operations.
- Improved User Experience: Users may experience faster load times and reduced latency without the overhead of draft syncing. This can be particularly noticeable in environments with limited or variable internet connectivity.
- However, disabling draft synchronization can negatively impact user experience, communication efficiency, and overall productivity. It’s important to balance performance improvements with the needs of your organization and users.

Unique emoji reaction limit

Limit the number of unique emoji reactions on each message. Increasing this limit can lead to poor client performance. Numerical input. Default is **50**. Maximum is 500.	• System Config path: **Site Configuration > Posts** • `config.json` setting: `ServiceSettings` > `UniqueEmojiReactionLimitPerPost` > `50` • Environment variable: `MM_SERVICESETTINGS_UNIQUEEMOJIREACTIONLIMITPERPOST`

File sharing and downloads

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > File Sharing and Downloads**.

Allow file sharing

• **true**: **(Default)** Allows users to attach files to messages. • **false**: Prevents users from attaching files (including images) to a message. This affects users on all clients and devices, including mobile apps.	• System Config path: **Site Configuration > File Sharing and Downloads** • `config.json` setting: `FileSettings` > `EnableFileAttachments` > `true` • Environment variable: `MM_FILESETTINGS_ENABLEFILEATTACHMENTS`

Allow file uploads on mobile

• **true**: **(Default)** Allows users to attach files to messages from mobile apps. • **false**: Prevents users from attaching files (including images) to messages from mobile apps.	• System Config path: **Site Configuration > File Sharing and Downloads** • `config.json` setting: `FileSettings` > `EnableMobileUpload` > `true` • Environment variable: `MM_FILESETTINGS_ENABLEMOBILEUPLOAD`

Allow file downloads on mobile

• **true**: **(Default)** Enables file downloads on mobile apps. • **false**: Disables file downloads on mobile apps. Users can still download files from a mobile web browser.	• System Config path: **Site Configuration > File sharing and downloads** • `config.json` setting: `FileSettings` > `EnableMobileDownload` > `true` • Environment variable: `MM_FILESETTINGS_ENABLEMOBILEDOWNLOAD`

Public Links

Available onall plansself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Public Links**.

Enable public file links

• **true**: Allows users to create public links to files attached to Mattermost messages. • **false**: **(Default)** Prevents users from creating public links to files and disables all previously created links.	• System Config path: **Site Configuration > Public Links** • `config.json` setting: `FileSettings` > `EnablePublicLink` > `false` • Environment variable: `MM_FILESETTINGS_ENABLEPUBLICLINK`

> Note
>
> When set to **false**, anyone who tries to visit a previously created public link will receive an error message. If the setting is returned to **true**, previously created links will be accessible, unless the **Public link salt** has been regenerated.

Public link salt

32-character salt added to the URL of public file links. Changing this setting will **invalidate** all previously generated links. The salt is randomly generated when Mattermost is installed, and can be regenerated by selecting **Regenerate** in the System Console. String input.	• System Config path: **Site Configuration > Public Links** • `config.json` setting: `FileSettings` > `PublicLinkSalt` • Environment variable: `MM_FILESETTINGS_PUBLICLINKSALT`

Notices

Available onall plansCloudandself-hosteddeployments

Access the following configuration settings in the System Console by going to **Site Configuration > Notices**.

Enable admin notices

• **true**: **(Default)** System admins will receive in-product notices about server upgrades and administration features. • **false**: System admins will not receive specific notices. Admins will still receive notices for all users (see **Enable end user notices**)	• System Config path: **Site Configuration > Notices** - • `config.json` setting: `AnnouncementSettings` > `AdminNoticesEnabled` > `true` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_ADMINNOTICESENABLED`

Enable end user notices

• **true**: **(Default)** All users receive in-product notices about client upgrades and end user features. • **false**: Users will not receive in-product notices.	• System Config path: **Site Configuration > Notices** • `config.json` setting: `AnnouncementSettings` > `UserNoticesEnabled` > `true` • Environment variable: `MM_ANNOUNCEMENTSETTINGS_USERNOTICESENABLED`

Connected workspaces

Available onEnterprise Advanced plansCloudandself-hosteddeployments

The following settings aren't available in the System Console and can only be set in `config.json`.

When connected workspaces are enabled, system admins can create and manage connected workspaces in the System Console by going to **Site Configuration > Connected Workspaces**.

Enable connected workspaces

Enable the ability to establish secure connections between Mattermost instances, and invite secured connections to shared channels where users can participate as they would in any public and private channel.

Connected workspaces requires Mattermost Enterprise servers running v10.2 or later.

By default, both configuration settings are disabled and must be enabled in order to share channels with secure connections. Enabling connected workspace functionality requires a server restart.

This feature's two `config.json` settings include:

- `ConnectedWorkspacesSettings.EnableRemoteClusterService: false` with options `true` and `false`.
- `ConnectedWorkspacesSettings.EnableSharedChannels: false` with options `true` and `false`.

Note

- Neither setting is available in the System Console and can only be set in `config.json` under `ConnectedWorkspacesSettings`.
- System admins for Cloud deployments can submit a request to have these required configuration settings enabled for their Cloud deployment instance.
- Following an upgrade to Mattermost v10.2 or later, existing configuration values for shared channels, including `EnableSharedChannels` and `EnableRemoteClusterService` are automatically converted to connected workspace configuration settings in the `config.json` file. The deprecated shared channels experimental settings remain in the `config.json` file to support backwards compatibility.

Disable shared channel status sync

Disable member status and availability synchronization between connected workspaces.

• **true**: Channel as well as member status and availability isn't synchronized. • **false**: **(Default)** Channel as well as channel member status and availability is synchronized at regular intervals.	• System Config path: N/A • `config.json` setting: `ConnectedWorkspacesSettings` > `DisableSharedChannelsStatusSync` > `false` • Environment variable: N/A

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

Default maximum posts per sync

Define the default maximum number of mesages to synchronize at a time. Default is **50**.	• System Config path: N/A • `config.json` setting: `ConnectedWorkspacesSettings` > `DefaultMaxPostsPerSync` > `50` • Environment variable: N/A

Sync users on connection open

Automatically synchronize users when a new connection between workspaces is established. This ensures that remote users are immediately discoverable for direct and group messages without requiring them to post in a shared channel first.

• **true**: **(Default)** Users are automatically synchronized when a new connection is established. • **false**: Users are not automatically synchronized when a new connection is established.	• System Config path: N/A • `config.json` setting: `ConnectedWorkspacesSettings` > `SyncUsersOnConnectionOpen` > `true` • Environment variable: N/A

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

Global user sync batch size

The number of users to sync in each batch when performing global user synchronization between connected workspaces. Default is **100**.	• System Config path: N/A • `config.json` setting: `ConnectedWorkspacesSettings` > `GlobalUserSyncBatchSize` > `100` • Environment variable: N/A

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

Member sync batch size

The number of channel members to sync in each batch when synchronizing channel membership between connected workspaces. Default is **100**.	• System Config path: N/A • `config.json` setting: `ConnectedWorkspacesSettings` > `MemberSyncBatchSize` > `100` • Environment variable: N/A

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

---

config.json-only settings

Available onall plansself-hosteddeployments

Cross-team search

Disable the ability to search across all teams or a specific team. • **true**: **(Default)** Cross-team search is enabled. Searches can be performed against all channels the user is a member of across all teams, a specific team, or the current team. • **false**: Cross-team search is disabled. Searches are performed on all channels the user is member of within the current team only.	• System Config path: N/A • `config.json` setting: `ServiceSettings.EnableCrossTeamSearch` > `true` • Environment variable: `MM_SERVICESETTINGS_ENABLECROSSTEAMSEARCH`

On this page

- Site configuration settings
  - Customization
    - Site name
    - Site description
    - Enable custom branding
    - Custom brand image
    - Custom brand text
    - Enable Ask Community link
    - Help link
    - Terms of Use link
    - Privacy Policy link
    - About link
    - Forgot Password custom link
    - Report a Problem
    - Report a Problem link

    - Report a Problem email address
    - Allow mobile app log downloads
    - Mattermost apps download page link
    - Android app download link
    - iOS app download link
    - Enable desktop app landing page
    - App custom URL schemes
    - Mobile external browser
  - Localization
    - Default server language
    - Default client language
    - Available languages
    - Enable experimental locales
  - Users and teams
    - Max users per team
    - Max channels per team
    - Enable join/leave messages by default
    - Enable users to open direct message channels with
    - Teammate name display
    - Lock teammate name display for all users
    - Allow users to view archived channels
    - Show email address
    - Show full name
    - Enable custom user statuses
    - Enable last active time
    - Enable custom user groups
    - User statistics update time
  - Notifications
    - Show @channel, @all, or @here confirmation dialog
    - Enable email notifications
    - Enable preview mode banner
    - Enable email batching
    - Email notification contents

    - Notification display name
    - Notification from address
    - Support email address
    - Notification reply-to address
    - Notification footer mailing address
    - Push notification contents
    - Enable notification monitoring
  - System-wide notifications
    - Enable system-wide notifications
    - Banner text
    - Banner color
    - Banner text color
    - Allow banner dismissal
  - Emoji
    - Enable emoji picker
    - Enable custom emoji
  - Posts
    - Automatically follow threads
    - Threaded discussions
    - Message priority
    - Persistent notifications
    - Maximum number of recipients for persistent notifications
    - Frequency of persistent notifications
    - Total number of persistent notifications per post
    - Enable website link previews
    - Disable link previews for specific domains
    - Enable message link previews
    - Enable SVGs
    - Enable LaTeX code block rendering
    - Enable inline LaTeX rendering
    - Custom URL schemes
    - Maximum Markdown nodes
    - Google API key

    - Enable server syncing of message drafts
    - Unique emoji reaction limit
- File sharing and downloads
    - Allow file sharing
    - Allow file uploads on mobile
    - Allow file downloads on mobile
- Public Links
    - Enable public file links
    - Public link salt
- Notices
    - Enable admin notices
    - Enable end user notices
- Connected workspaces
    - Enable connected workspaces
    - Disable shared channel status sync
    - Default maximum posts per sync
    - Sync users on connection open
    - Global user sync batch size
    - Member sync batch size
- config.json-only settings
    - Cross-team search

SMTP email setup

In a production environment, Mattermost requires SMTP email enabled for email notifications and password resets when using email-based authentication.

Set up an SMTP email service

Any SMTP email service can be used. You need a copy of the following information: `Server Name`, `Port`, `SMTP Username`, and `SMTP Password`.

Tip

If you don't have an SMTP service you can set one up with [Amazon Simple Email Service (SES) (https://aws.amazon.com/ses/)](https://aws.amazon.com/ses/):

1. Go to [Amazon SES console (https://console.aws.amazon.com/ses)](https://console.aws.amazon.com/ses) then **SMTP Settings > Create My SMTP Credentials**.
2. Copy the `Server Name`, `Port`, `SMTP Username`, and `SMTP Password` values. You'll need these values to configure Mattermost.
3. From the `Domains` menu, set up and verify a new domain, then enable `Generate DKIM Settings` for the domain. We recommend you set up [Sender Policy Framework (https://en.wikipedia.org/wiki/Sender_Policy_Framework)](https://en.wikipedia.org/wiki/Sender_Policy_Framework) (SPF) and/or [Domain Keys Identified Mail (https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail)](https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail) (DKIM) for your email domain.
4. Choose a sender address like `mattermost@domain.com` and select `Send a Test Email` to verify setup is working correctly.

Alternatively, you can use one of the services listed below, or can set up local `sendmail` by setting **Server Name** `127.0.0.1` with **Port** `25`.

If deploying Mattermost using Docker, the standard docker `172.16.0.0/12` IP range isn't used. Specify the IP range `192.168.0.0/24` to the email service to avoid relay access errors. If using postfix, under `/etc/postfix/main.cf`, specify `mynetworks = 192.168.0.0/24`. This may vary depending on how Mattermost is deployed. Ensure that **Port 25** is open if a firewall is present.

Configure SMTP settings

1. In Mattermost go to **System Console > Authentication > Email**, and set **Allow Sign Up With Email** to **true**.

2. In the System Console, go to **Notifications > Email** and configure Mattermost for your SMTP service. See the SMTP configuration documentation for details.

> - *Set **Send Email Notifications** to **true**.*
> - *Set the **Notification Display Name** for the account sending notifications.*
> - *Set **Notification Email Address** for the email address used to send notifications.*
> - *Enter the **SMTP Username**, **SMTP Password**, **SMTP Server**, and **SMTP Port** you copied from initial setup.*
> - *We recommend setting **Connection Security**: to **TLS (Recommended)** to encrypt communication between Mattermost and your SMTP service.*

3. Select **Save**.
4. Under **Connection Security**, select **Test Connection**. Mattermost will confirm whether a connection to the SMTP service is successful by sending you an email. If the test fails, Mattermost will provide details about why it failed in the System Console. See the check Mattermost logs section below for details.

Sample SMTP settings

Amazon SES

- Set **SMTP Username** to **[YOUR_SMTP_USERNAME]**
- Set **SMTP Password** to **[YOUR_SMTP_PASSWORD]**
- Set **SMTP Server** to **email-smtp.us-east-1.amazonaws.com**
- Set **SMTP Port** to **465**
- Set **Connection Security** to **TLS**

Postfix

Make sure Postfix is installed on the same machine as Mattermost.

- Set **SMTP Username** to **(empty)**
- Set **SMTP Password** to **(empty)**
- Set **SMTP Server** to **localhost**
- Set **SMTP Port** to **25**
- Set **Connection Security** to **(empty)**

Gmail

- Set **SMTP Username** to **your_email@gmail.com**
- Set **SMTP Password** to **your_password**
- Set **SMTP Server** to **smtp.gmail.com**
- Set **SMTP Port** to **587**
- Set **Connection Security** to **STARTTLS**

> Warning
>
> Additional configuration is required in Google to allow SMTP email to relay through their servers. See SMTP relay: Route outgoing non-Gmail messages through Google (https://support.google.com/a/answer/2956491?hl=en) for the required steps.

Hotmail

- Set **SMTP Username** to **your_email@hotmail.com**
- Set **SMTP Password** to **your_password**
- Set **SMTP Server** to **smtp-mail.outlook.com**
- Set **SMTP Port** to **587**
- Set **Connection Security** to **STARTTLS**

Office365/Outlook

- Set **SMTP Username** to **your_email@hotmail.com**
- Set **SMTP Password** to **your_password**
- Set **SMTP Server Name** to **smtp.office365.com**
- Set **SMTP Port** to **587**
- Set **Connection Security** to **STARTTLS**

Troubleshooting SMTP

TLS/STARTTLS requirements

If you fill in **SMTP Username** and **SMTP Password** then you must set **Connection Security** to **TLS** or to **STARTTLS**

Troubleshooting using logs

If you have issues with your SMTP install, from your Mattermost team site go to **System Console > Logs** to look for error messages related to your setup. You can do a search for the error code to narrow down the issue. Sometimes ISPs require nuanced setups for SMTP and error codes can hint at how to make the proper adjustments.

For example, if **System Console > Logs** displays the following error, search for `554 5.7.1 error` and `Client host rejected: Access denied`.

```
Connection unsuccessful: Failed to add to email address - 554 5.7.1
<unknown[IP-ADDRESS]>: Client host rejected: Access denied
```

Checking your Notifications settings

If an SMTP connection test in the System Console for a self-hosted Mattermost deployment fails with the message `Connection unsuccessful: Failed to set the to address: 550 5.7.27 <test@example.com>`, go to **System Console > Site Configuration > Notifications** to verify that notification settings are configured correctly, including **Notification Display Name**, **Notification**

From Address, **Support Email Address**, and **Notification Reply-To Address**. Cloud administrators can't manage **Notification From Address** or **Notification Reply-To Address**.

Checking your SMTP server is reachable

- Attempt to Telnet to the email service to make sure the server is reachable. For additional information, visit https://learn.microsoft.com/en-us/exchange/mail-flow/test-smtp-with-telnet?view=exchserver-2019. If you're using an earlier version than Exchange Server 2019, select your version from the left-hand navigation menu.
- You must run the following commands from the same machine or virtual instance where `mattermost/bin/mattermost` is located.
- Telnet to the email server with `telnet mail.example.com 25`. If the command works you should see something like:

  ```
  Trying 24.121.12.143... Connected to mail.example.com. 220
  mail.example.com NO UCE ESMTP
  ```

  Then type something like `HELO <your mail server domain>`. If the command works you should see something like:

  ```
  250-mail.example.com NO UCE 250-STARTTLS 250-PIPELINING 250 8BITMIME
  ```

> Note
>
> - Telnet isn't included in official Mattermost Docker images, so you either need to use `ping` on those, or install Telnet yourself either directly or by modifying the Dockerfile.
> - For further assistance, review the [Troubleshooting forum (https://forum.mattermost.com/c/trouble-shoot/16)](https://forum.mattermost.com/c/trouble-shoot/16) for previously reported errors, or [join the Mattermost user community for troubleshooting help (https://mattermost.com/community/)](https://mattermost.com/community/).

On this page

- SMTP email setup
- SMTP email setup
  - Set up an SMTP email service
  - Configure SMTP settings
  - Sample SMTP settings
    - Amazon SES
    - Postfix
    - Gmail
    - Hotmail
    - Office365/Outlook
  - Troubleshooting SMTP
    - TLS/STARTTLS requirements
    - Troubleshooting using logs
    - Checking your Notifications settings
    - Checking your SMTP server is reachable
  - Set up an SMTP email service
  - Configure SMTP settings
  - Sample SMTP settings
  - Troubleshooting SMTP
    - TLS/STARTTLS requirements
    - Troubleshooting using logs
    - Checking your Notifications settings
    - Checking your SMTP server is reachable

System Attributes

System attributes configuration settings provide system admins with centralized control over key user account properties.

Review and manage the following system attributes configuration options in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **System Attributes**.

You can define, manage, and enforce specific attributes, including:

- **Custom attributes for user profiles**: Display details such as job titles, departments, or other metadata, on user profiles that align with your organizational structures and workflows. Learn more about managing custom user profile attributes.
- **Granular access controls based on user attributes**: Ensure users have access to only the resources and functionality relevant to their roles, bolstering security and compliance across the organization. Learn more about managing access based on user attributes.

User management configuration settings

Review and manage the following in the System Console by selecting the **Product**

menu, selecting **System Console**, and then selecting **User Management**:

- Users
- Groups
- Teams
- Channels
- Permissions
- System roles

Users

Mattermost system admins can provision and manage user accounts, team membership, roles and permissions, and update user email addresses.

Provision users

Getting people set up with a Mattermost account is typically something that system admins do when deploying and configuring the Mattermost deployment. A Mattermost admin can provision Mattermost users using one or more of the following methods:

- Enable account creation.
- Use mmctl user create or Mattermost [APIs (https://api.mattermost.com/#tag/users)](https://api.mattermost.com/#tag/users) to create user accounts.

- Migrate user accounts from other collaboration systems and bulk load that user data into Mattermost.
- Connect an authentication service to assist with user provisioning, such as AD/LDAP authentication or SAML authentication.

Review user data

From Mattermost v9.6, you can review the following user data in the System Console:

- **Email**: The user's email address.
- **Member since**: The number of days since the user joined the Mattermost server.
- **Last login**: The date of the user's last successful login to the server.
- **Last activity**: The total number of days since the user was last active on the server, which is typically based on the user's availability.
- **Last post**: The total number of days since the user's last sent message on the server.
- **Days active**: (PostgreSQL only) The total number of days in which the user has sent a message in Mattermost.
- **Messages posted**: (PostgreSQL only) The total number of messages the user has sent on the server.

By default, you see all columns of data and data for all time.

- Show or hide data all data columns exccept **User details** and **Actions**, as preferred.
- Filter results to the last 30 days, the previous month, and the last 6 months for **Last post**, **Days active**, and **Messages posted** user data.

Find users

Find a user using the System Console.

1. Go to **System Console > User Management > Users** to access all user accounts.

2. Search for specific users by entering a partial or full username, user ID, first name, last name, or email address in the **Search** field and pressing `Enter`.

Filter user searches

Filter System Console user searches to narrow down results based on the team membership, role, and user status.

1. Go to **System Console > User Management > Users** to access all user accounts.

2. Select **Filters** located to the right of the **Search users** field to access available filter options.

3. Select **Apply** to filter user search results.

Identify a user's ID

Users can be specified in Mattermost by username or user ID. Usernames automatically resolve when a match is detected. Identify a user's ID using the System Console, the Mattermost API, or mmctl.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select a **User** to review their ID in the User Configuration page.

Alternatively, identify a user's ID using the Mattermost API or mmctl:

- Using the Mattermost API, make an HTTP GET request to the following endpoint: `https://your-mattermost-url/api/v4/users/username/username_here`. Replace `your-mattermost-url` with the URL of your Mattermost instance and `username_here` with the username you are looking for. The API response contains a JSON object that includes the user's ID among other details.
- Using mmctl, in a terminal window, use the following command to list all users and their IDs: `mmctl user list` to return a list of user IDs.

Export user data

From Mattermost v9.6, Mattermost Enterprise and Professional customers can export user data as a CSV report.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Filter the user data as needed.
3. Select **Export** located in the top right corner of the System Console interface, and then select **Export data**. You'll receive the report in CSV format as a direct message in Mattermost.

Deactivate users

To delete a user from your Mattermost deployment, you can deactivate the user's account. Deactivated users have an deactivated status, are logged out of Mattermost as soon as they are deactivated, and deactivated users can no longer log back in. You can manage the user's role, password, and email address while a user's account is deactivated.

> Note
>
> From Mattermost v10.10, when a user account is deactivated, the account's availability is automatically set to offline.

> Note
>
> LDAP-managed users must be deactivated through LDAP, and can't be deactivated using the System Console or the API.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select a **User** that you wish to activate or deactivate.
3. If the selected user is currently active, you can find the **Deactivate** button in the **User Configuration** page.
4. Select **Deactivate**, and confirm the deactivation.

> Tip
>
> Re-activate a deactivated user by selecting **Activate**.

What happens to deactivated user integrations?

Important

If you deactivate a Mattermost user who has integrations tied to their user account, consider the following consequences and recommendations based on the integration type:

- **Slash commands** will continue to work after user deactivation. Consider deleting the existing slash command and creating a new slash command associated with a different user account to decouple sensitive token data from the deactivated user account. Alternatively, consider regenerating the token of the existing slash command. Check that the deactivated user doesn't have access to the slash command **Request URL** which is the callback URL to receive the HTTP POST or GET event request when the slash command is run.
- **Outgoing webhooks** will continue to work after user deactivation. Consider regenerating the webhook token and check that the deactivated

user no longer has access to the callback URLs, as having access would result in the deactivating user receiving the outgoing webhooks.

- **Incoming webhooks** will continue to work after user deactivation. Because the URL produced (https://developers.mattermost.com/integrate/webhooks/incoming/#create-an-incoming-webhook) includes `xxx-generatedkey-xxx`, anyone who has the URL can post messages to the Mattermost instance. We recommend removing the incoming webhook and creating a new one associated with a different user account.
- **Bot accounts** won't continue to work after user deactivation when the disable bot accounts when owner is deactivated is enabled. This configuration setting is enabled by default.
- **OAuth apps** won't continue to work after user deactivation, and associated tokens are deleted. Manual action is needed to keep these integrations running.

Manage user's roles

Apply roles to users using the System Console. The current available roles are **System admin** and **Member**.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Under **Actions**, select **Manage roles**.
3. Specify whether the user has the role of **System admin** or **Member**, and then select **Save**.

Manage user's teams

Add or remove users from teams using the System Console.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select the user you want to manage.
3. You can include the user in one new team or a group of teams. Select **Add Team**, select one or more teams, and then select **Add**.

4. To remove the user from a specific team, select **More**

   •••

   next to the team name, and select **Remove from team**.

> Tip
>
> You can also remove the user from a specific team from the **Actions** column on the **Users** page. Select **Manage Teams**, and then select **Remove from Team** for applicable teams.

Manage user's settings

Note

Available only on [Enterprise (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

From Mattermost v9.11, system admins can help end users customize their Mattermost notifications by editing the user's notification settings on the user's behalf within the System Console. Users can view, modify, and override their own settings at any time.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select the user you want to manage.
3. Select **Manage User Settings**.
4. In **Admin Mode**, modify the user's settings as needed. Saved changes take effect immediately in the user's account.

Reset login attempts

If a user, whose account details are synchronized with AD/LDAP, can't access their Mattermost account due to too many failed login attempts, you can reset the number of failed login attempts for their user account in the System Console.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Under **Actions**, select **Reset login attempts**.

> Note
>
> To adjust the maximum login attempts allowed for all users, go to **System Console > Authentication > AD/LDAP > Maximum Login Attempts**. Lowering this configuration setting value below the maximum threshhold allowed on your AD/LDAP server will ensure that your users won't get locked out of AD/LDAP due to failed login attempts in Mattermost.

Update user's email

Update the emails of users using the System Console.

> Note
>
> From Mattermost v10.9, email addresses enclosed in angle brackets (e.g., `<billy@example.com>`) will be rejected. To avoid issues, ensure all user emails comply with the plain address format (e.g., `billy@example.com`). In addition, we strongly recommend taking proactive steps to audit and update Mattermost user data to align with this product change, as impacted users may face issues accessing Mattermost or managing their user profile. You can update these user emails manually using mmctl user email.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select the user, and enter a new email in the **Email** field.
3. Select **Save**.

> Tip
>
> You can also update the email from the **Actions** column on the **Users** page. Select **Update email**, enter the new email for the user, and then select **Save**.

Reset user's password

Reset a user's password using the System Console.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select the user from the list, and then select **Reset Password**.
3. Enter a new password and select **Reset**.

You can also reset the password using the **Actions** column for the specific user on the **Users** page. Select **Reset password** from the **Actions** column dropdown, enter the new password in the pop-up dialog box and select **Reset**.

Revoke a user's session

Revoke the user sessions in case of an emergency to secure the user account using the System Console. This logs the user out of all devices.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Under the **Actions** column for the specific user, select **Remove sessions**.
3. Select **Revoke** to remove all sessions for that user.

Review user profile details

View user profile details using the System Console. Gather information including the user's ID, username, email, authentication method, and team memberships.

1. Go to **System Console > User Management > Users** to access all user accounts.
2. Select the user from the list, and browse user details.

Groups

Manage default teams and channels by linking AD/LDAP groups to Mattermost groups.	• System Config path: **User Management > Groups** • `config.json setting`: N/A • Environment variable: N/A

See the AD/LDAP groups documentation for details.

Teams

Find Teams

You can find existing teams in your Mattermost instance using the System Console.

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Enter the team name in the **Search** box.

Mattermost Teams

> Tip
>
> From Mattermost v9.6, you can search for specific teams by entering a partial or full team name in the **Search** field and pressing `Enter`.

Filter team searches

Filter your team search to narrow down results based on the team management type (anyone can join, invite only, or group sync).

1. Go to **System Console > User Management > Teams** to access all available teams.

2. Select **Filters** located to the right of the **Search** field to access available filter options.

3. Choose any filter and select **Apply** to filter team search results.

Review team configuration

View team configuration details using the System Console. Gather information including the team name, team description, team management options, groups, and members.

1. Go to **System Console > User Management > Teams** to access all available teams.

2. Select the team from the list to view its configuration page.

Manage team membership

Admins can directly add or remove member from the team and customize how members are added to the team using the System Console.

Add members

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Select the team from the list to view its configuration page.
3. In the **Members** section, select **Add Members**.
4. Select an user or available bot from the list or try using the **Search** to find a specific one.
5. Select **Add** to add the user or bot.
6. Select **Save**.

Remove members

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Select the team from the list to view its configuration page.
3. In the **Members** section, select **Remove** for the user or bot that you want to remove.
4. Select **Save**.

Sync group members

When enabled, adding and removing users from groups will add or remove them from this team. The only way of inviting members to this team is by adding the groups they belong to. See the Synchronize teams and channels documentation for further details.

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Select the team from the list to view its configuration page.
3. In the **Team Management** section, enable the **Sync Group Members** option.
4. Select **Save**.

Anyone can join this team

This team can be discovered allowing anyone with an account to join this team.

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Select the team from the list to view its configuration page.
3. In the **Team Management** section, enable the **Anyone can join this team** option.
4. Select **Save**.

Only specific email domains can join this team

Users can only join the team if their email matches one of the specified domains.

1. Go to **System Console > User Management > Teams** to access all available teams.

2. Select the team from the list to view its configuration page.
3. In the **Team Management** section, enable the **Only specific email domains can join this team** option and add the specific domains.
4. Select **Save**.

Synchronize team members

Admins can choose between inviting members to a team manually or synchronizing members automatically from AD/LDAP groups. See the using AD/LDAP synchronized groups documentation for details on managing team or private channel membership.

Archive the team

1. Go to **System Console > User Management > Teams** to access all available teams.
2. Select the team from the list to view its configuration page.
3. In the **Team Profile** section, select **Archive Team**.
4. Select **Save**.

Channels

Find Channels

Find existing channels in your Mattermost instance using the System Console.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Enter the channel name in the **Search** box.

Mattermost Channels

Filter channel searches

Filter your channel search to narrow down results based on the channel type (as public, private or archived), channel management type (group sync or manual invites) or based on the team the channel belongs to.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select **Filters** located to the right of the **Search** field to access available filter options.
3. Choose any filter and select **Apply** to filter channel search results.

Review channel configuration

View channel configuration details using the System Console. Gather information including the channel profile, advanced access controls, channel management options, groups, and members.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select a channel from the list to review its channel configuration details.

Advanced Access Control

Manage the Management actions available to channel members and guests.

Create Posts

The ability for members and guests to create posts in the channel.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select a channel from the list to view its configuration page.
3. In the **Create Posts** section under the **Advanced Access Control** tab, select the option for **Guests**, **Members**, or both to enable those users to post messages in the channel.
4. Select **Save**.

Post Reactions

The ability for members and guests to react with emojis on messages in the channel.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select a channel from the list to view its configuration page.
3. In the **Post Reactions** section under the **Advanced Access Control** tab, select the option for **Guests**, **Members**, or both to enable those users to react with emojis on messages posted to the channel.
4. Select **Save**.

Manage Members

The ability for members to add and remove people from the channels. Guests can't add or remove people from channels.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select a channel from the list to view its configuration page.
3. In the **Manage Members** section under the **Advanced Access Control** tab, select **Members** to enable those users to manage members for the channel.
4. Select **Save**.

Channel Mentions

The ability for members and guests to use channel mentions, including **@all**, **@here**, and **@channel**, in the channel.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Channel Mentions** section under the **Advanced Access Control** tab, select the option for **Guests**, **Members**, or both to enable those users to use channel mentions.
4. Select **Save**.

> Tip
>
> **Guests** and **Members** can't use channel mentions without the ability to **Create Posts**. To enable this permission, these users must have been granted **Create Posts** permission first.

Manage Bookmarks

The ability for members to add, delete, and sort bookmarks. Guests can't add, remove, or sort bookmarks for the channel.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Manage Bookmarks** section under the **Advanced Access Control** tab, select **Members** to enable those users to add, sort, or remove bookmarks for the channel.
4. Select **Save**.

> Tip
>
> The ability to manage bookmarks for the channel is available for **Members** only. **Guests** can't add, remove or sort bookmarks for the channel.

Channel Management

Choose between inviting members manually or sychronizing members automatically from groups.

Sync Group Members

When enabled, adding and removing users from groups will add or remove them from this team. The only way of inviting members to this team is by adding the groups they belong to. See the Synchronize teams and channels documentation for further details.

1. Go to **System Console > User Management > Channels** to access all available channels.

2. Select the channel from the list to view its configuration page.
3. In the **Channel Management** tab, enable the **Sync Group Members** option.
4. Select **Save**.

Public channel or private channel

Public channels are discoverable and any user can join. Private channels require invitations to join.

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Public channel or private channel** section under the **Channel Management** tab, toggle between **Public** or **Private** to change channel visibility.
4. Select **Save**.

> Tip
>
> When Group Sync is enabled, private channels can't be converted to public channels.

Members

Choose between inviting members manually or synchronizing members automatically from groups.

Add members

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Members** section, select **Add Members**.
4. Use the **Search** box to find a specific user or group to add to the channel.
5. Select **Add** to add the user or group as a **Member**.
6. Select **Save**.

Remove members

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Members** section, select **Remove** for the user that you want to remove.
4. Select **Save**.

Archive a channel

1. Go to **System Console > User Management > Channels** to access all available channels.
2. Select the channel from the list to view its configuration page.
3. In the **Channel Profile** section, select **Archive Channel**.
4. Select **Save**.

> Tip
>
> Channels can be deleted with all content, including posts in the database, using the mmctl channel delete tool.

Permissions

Note

Available only on [Enterprise and Professional (https://mattermost.com/pricing/)](https://mattermost.com/pricing/) plans

Restrict actions in Mattermost to authorized users only.	• System Config path: **User Management > Permissions** • `config.json setting`: N/A • Environment variable: N/A

See advanced permissions documentation for details.

System roles

Restrict System Console access to authorized users only.	• System Config path: **User Management > System Roles** • `config.json setting`: N/A • Environment variable: N/A

See the delegated granular administration documentation for details.

On this page

- User management configuration settings
  - Users
    - Provision users
    - Review user data
    - Find users
    - Filter user searches
    - Identify a user's ID
    - Export user data
    - Deactivate users
      - What happens to deactivated user integrations?
    - Manage user's roles
    - Manage user's teams
    - Manage user's settings
    - Reset login attempts
    - Update user's email
    - Reset user's password
    - Revoke a user's session
    - Review user profile details
  - Groups
  - Teams
    - Find Teams
    - Filter team searches
    - Review team configuration
    - Manage team membership
      - Add members
      - Remove members
      - Sync group members
      - Anyone can join this team
      - Only specific email domains can join this team
    - Synchronize team members
    - Archive the team

- Channels
  - Find Channels
  - Filter channel searches
  - Review channel configuration
  - Advanced Access Control
    - Create Posts
    - Post Reactions
    - Manage Members
    - Channel Mentions
    - Manage Bookmarks
  - Channel Management
    - Sync Group Members
    - Public channel or private channel
  - Members
    - Add members
    - Remove members
  - Archive a channel
- Permissions
- System roles

Attribute-Based Access Control

From Mattermost v10.9, system admins in large or complex organizations who require Zero Trust Security when handling with sensitive information can prevent unauthorized access through attribute-based access controls.

Enforcing strict access controls based on user attributes eliminates manual role adjustment processes that can lead to security risks, inefficiencies, or inappropriate access, while maintaining security and compliance by ensuring that only authorized users can access specific Mattermost channels.

Before you begin

Attribute-based access controls require defined user attributes that are either synchronized from an external system (such as LDAP or SAML) or manually configured and enabled on your Mattermost server. You'll need to configure user attributes in the System Console first befopre creating access policies.

Once user attributes are defined, go to **System Console > System Attributes > Attribute-Based Access** to enable attribute-based access controls for your Mattermost instance. This functionality requires a Mattermost Enterprise Advanced license.

Once enabled, you have 2 ways to configure access policies in Mattermost:

- Attribute-based access rules you can assign across multiple channels.
- Channel-specific rules you can assign to individual channels.

Define access control policies

You can add multiple rules to a single policy, and each rule can include multiple attribute values.

1. In the System Console, go to **System Attributes > Attribute-Based Access** and select **Add Policy**.
2. Enter a unique policy name.

3. Choose whether to automatically add users who match your configured attribute values as new members. Automatic synchronization is disabled by default.

> - ***True***: *Automatically maintains channel membership according to the defined rules as user attributes change.*
> - ***False*** *(**Default**): Only removes members and restricts adding them to the channel if they don't match defined rules. Users are not automatically added.*
>
> *Regardless of whether this configuration setting, users who no longer match the configured attribute values in the future will be removed from the channel after the next channel synchronization.*

4. Define the attribute-based access rules to restrict channel membership.

> *Simple Mode*
>
> *Simple Mode is ideal for simple and straightforward access control scenarios. Each row is a single condition that must be met for a user to comply with the policy. Simple Mode only supports simple conditions without nested logic or mixed logical operators. All rules are combined with a logical AND operator* `&&`*.*
>
> 1. *Select **Add attribute** and select the attribute you want to use for access control.*
> 2. *Specify how you want the attribute to match the user profile. You can choose from the following options:*
>
> > - ***Is***: *The attribute must be an exact match of the value.*
> > - ***Is not***: *The attribute must not contain specified value.*
> > - ***In***: *The attribute must match at least one value.*
> > - ***Starts with***: *The attribute value must start with the specified value.*
> > - ***Ends with***: *The attribute value must end with the specified value.*
> > - ***Contains***: *The attribute value must exist somewhere with the specified value.*

> 3. *Specify the attribute values that users must have to be granted access to the channel.*
>
> *Advanced Mode*
>
> *Advanced Mode is ideal for complex access control scenarios that require CEL syntax to combine multiple conditions with logical operators that support rules like* `user.<attribute> == <value>`*, using* `&&` */* `||` *(and/or) for multiple conditions, and* `()` *to group conditions. The CEL Expression Editor provides real-time syntax validation and feedback, as well as context-aware autocomplete for attributes, operators, and attribute values.*
>
> *You can also start defining rules in Simple Mode and then switch to Advanced Mode to refine the rules further as needed. However, you'll be blocked from switching from Advanced back to Simple Mode if one of the following are true:*
>
> - *Mixed logical operators are used between conditions.*
> - *Nested logic/grouping (parentheses) are present.*
> - *Unsupported operators or expressions are detected.*
>
> *The syntax structure is* `user.<attribute> <operator> <value>`*.*
>
> *As you type, autocompletes show available attributes. As you select attributes, autocomplete suggests appropriate CEL operators. After selecting an operator, when attribute values are pre-defined, autocomplete suggests values to choose from. Mattermost will explicitly indicate issues such as missing operators, incorrect syntax, or incomplete conditions.*
>
> *Select the* ***Validate syntax*** *bar to check the syntax of your rule. If the syntax is valid, the bar will turn green and display a message indicating that the syntax is valid. If there are any issues, the bar will turn red and display an error message.*

Test rules

Select **Test access rule** to test the rule against your user base to return how many users would be granted access to the channel based on the current rule. Test your rules to ensure the intended scope and avoid unexpected access changes.

Manage rules

You can apply changes to existing rules or remove rules at any time using either Simple Mode or Advanced Mode. Select **Save** to save your changes.

Assign policies to private channels

Specify the private channel that your access control policy applies to by selecting **Add channels** to search for and select the channels you want. You can assign the policy to multiple channels at once, or you can assign it to individual channels later. Select **Save** to save your changes.

> Note
>
> Private channels with attribute-based access control policies can't have guest users invited to them. Only users who match the defined attribute criteria can be added to ABAC-controlled channels, ensuring strict adherence to access control policies.

Delete policies

To delete a policy, select the **Delete** button next to the policy you want to remove. You can only delete policies that are not currently assigned to any channels. If a policy is assigned to channels, you must first remove it from those channels before you can delete it.

Define access controls per channel

You can assign an existing access control policy to a private channels for more granular control over channel membership. This is useful when you need to apply different rules for different channels.

1. In the System Console, go to **User Management > Channels** to select the private channel you want to configure, and select **Edit**.
2. In the **Channel Management** section, enable the **Enable attribute-based channel access** option.

3. Under **Access policy**, select **Link to a policy** to select an existing policy.

Remove channel policies

Disable the policy for the channel by selecting **Remove Policy**. You can then link the channel to a different policy if preferred.

On this page

- Attribute-Based Access Control
- Attribute-Based Access Control
  - Before you begin
  - Define access control policies
    - Test rules
    - Manage rules
    - Assign policies to private channels
    - Delete policies
  - Define access controls per channel
    - Remove channel policies
  - Before you begin
  - Define access control policies
    - Test rules
    - Manage rules
    - Assign policies to private channels
    - Delete policies
  - Define access controls per channel
    - Remove channel policies

Customize branding

Whether you're customizing the appearance of your workspace, utilizing branding tools, or managing code signing for custom builds, this section of documentation has you covered and provides everything you need to customize the branding of Mattermost to align with your organization's identity. Use the navigation below to access detailed instructions for each customization option.

- Customize Mattermost - Learn how to customize the Mattermost server.
- Custom branding tools - Learn about custom branding tools for Mattermost.
- Code signing custom builds - Learn about code signing custom builds of Mattermost.

Mattermost error codes

Mattermost is designed to deploy in private networks which may be disconnected or "air-gapped" from the internet. In these deployments, links to Mattermost's online documentation may be unavailable.

Therefore, Mattermost error codes are provided to enable administrators to use internet connected machines to look up error messages and resolutions in online resources, including Mattermost documentation and user forums, as well as providing Mattermost support teams a concise way to identify issues.

Mattermost error codes are used in logs, administrative tools, as well as on-screen messages to concisely identify issues and connect them with additional resources.

In advanced deployments, error codes can be overwritten by administrators to reference internal documentation and guides.

ERROR_SAFETY_LIMITS_EXCEEDED

A safety limits exceeded error (`ERROR_SAFETY_LIMITS_EXCEEDED`) displays in the free version of Mattermost, and certain functionality may be limited, when usage grossly exceeds the recommended limit for users in a safe deployment, including more than 2,500 users are registered on the server.

2,500 users represents a "high upper limit" for deployments that are approximately 100 times the recommended size, which is far beyond the intended design of the product.

The free version of Mattermost is intended for approximately 50 users. If your Mattermost materially exceeds this recommended size, system admins should seek to either purchase a commercial license (https://mattermost.com/pricing/), or apply for a non-profit subscription license. Alternatively, admins can deactivate users until the user count falls below the high upper limit.

ERROR_LICENSED_USERS_LIMIT_EXCEEDED

A licensed user limit exceeded error (`ERROR_LICENSED_USERS_LIMIT_EXCEEDED`) displays when attempting to create or activate users on a licensed Mattermost server where the user count would exceed the maximum number of users allowed by the license.

This error occurs when:

- Creating new users would exceed the license user limit.
- Activating deactivated users would exceed the license user limit.

To resolve this error, system administrators can:

- Deactivate users to reduce the active user count below the license limit.
- Contact [Mattermost Sales (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) to discuss license options.

On this page

- Mattermost error codes
  - ERROR_SAFETY_LIMITS_EXCEEDED
  - ERROR_LICENSED_USERS_LIMIT_EXCEEDED

Generate a Support Packet

The Support Packet is used to help customers diagnose and troubleshoot issues. Use the System Console or the mmctl system supportpacket command to generate a zip file that includes configuration information, logs, plugin details, and data on external dependencies across all nodes in a high-availability cluster. Confidential data, such as passwords, are automatically stripped.

Generate

Important

Before generating a Support Packet, go to **System Console > Environment > Logging** and ensure **Output logs to file** is set to **true**, and set **File Log Level** to **DEBUG**.

Web/Desktop

1. Go to the System Console, and select **Commercial Support** from the System Console menu.

2. Select **Download Support Packet**. A zip file is downloaded to the local machine. You'll be notified if any packet files are unavailable during packet generation. See the `warning.txt` file for details.

mmctl

Run the mmctl system supportpacket command to generate and download a Support Packet to share with Mattermost Support.

```
go run ./cmd/mmctl system supportpacket
```

```
Downloading Support Packet Downloaded Support Packet to
mattermost_support_packet_.zip
```

Santitize confidential data

Please sanitize any confidential data you wish to exclude before sharing the packet with Mattermost.

When present, the following information is automatically santized during packet generation: `LdapSettings.BindPassword`, `FileSettings.PublicLinkSalt`, `FileSettings.AmazonS3SecretAccessKey`, `EmailSettings.SMTPPassword`, `GitLabSettings.Secret`, `GoogleSettings.Secret`, `Office365Settings.Secret`, `OpenIdSettings.Secret`, `SqlSettings.DataSource`, `SqlSettings.AtRestEncryptKey`, `ElasticsearchSettings.Password`, `All SqlSettings.DataSourceReplicas`, `All SqlSettings.DataSourceSearchReplicas`, `MessageExportSettings.GlobalRelaySettings.SmtpPassword`, and `ServiceSettings.SplitKey`.

Important

Plugins may not be sanitized during packet generation.

- From Mattermost v10.1, plugins can mark their configuration as hidden. If a plugin marks its configuration as hidden, the configuration is sanitized during packet generation.
- Otherwise, ensure you sanitize any additional confidential details in the `plugin.json` file before sharing it with Mattermost. Replace details with

example strings that contain the same special characters if possible, as special characters are common causes of configuration errors.

Share the packet with Mattermost

Add the generated Support Packet to a [standard support request (https://support.mattermost.com/hc/en-us/requests/new)](https://support.mattermost.com/hc/en-us/requests/new), or share with with the Mattermost team you're working with.

Important

Disable debug logging once you've generated the Support Packet. Debug logging can cause log files to expand substantially, and may adversely impact server performance. We recommend enabling it temporarily, or in development environments, but not production enviornments.

Contents of a Support Packet

A Mattermost Support Packet can contain the following files:

v10.10 and later

> Note
>
> From v10.10, support packet file organization has been improved to make it easier to identify cluster-wide versus cluster-specific files:
>
> - **Cluster-wide files** (identical across all nodes in a high-availability cluster) remain in the root directory of the support packet.
> - **Cluster-specific files** (unique per node) are now organized in subdirectories named after each cluster node.

Cluster-wide files (root directory):

- metadata.yaml
- `plugins.json` (all active and inactive plugins)
- `sanitized_config.json` (sanitized copy of the Mattermost configuration)

- `stats.yaml` (Mattermost usage statistics)
- `jobs.yaml` (last runs of important jobs)
- `diagnostics.yaml` (core plugin diagnostics data)
- `permissions.yaml` (role & scheme information)
- `warning.txt` (present when issues are encountered during packet generation)
- `tsdb_dump.tar.gz` (present when the Metrics plugin is installed and the **Performance metrics** option is selected when generating the Support Packet)

Cluster-specific files (in node subdirectories):

- `<node-id>/mattermost.log` (Mattermost logs for each node)
- `<node-id>/audit.log` (Mattermost audit logs for each node)
- `<node-id>/ldap.log` (AD/LDAP logs for each node)
- `<node-id>/notifications.log` (notifications logs for each node)
- `<node-id>/cpu.prof` (Go performance metrics for each node)
- `<node-id>/heap.prof` (Go performance metrics for each node)
- `<node-id>/goroutines` (Go performance metrics for each node)

The following additional plugin diagnostic data is included in the generated support packet when the plugin is enabled and operational:

- GitHub: `/github/diagnostics.yaml`
- GitLab: `/com.github.manland.mattermost-plugin-gitlab/diagnostics.yaml`
- Jira: `/jira/diagnostics.yaml`
- Calls: `/com.mattermost.calls/diagnostics.yaml`
- Boards: `/focalboard/diagnostics.yaml`
- Playbooks: `/playbooks/diagnostics.yaml`
- MSCalendar: `/com.mattermost.mscalendar/diagnostics.yaml`

- Google Calendar: `/com.mattermost.gcal/diagnostics.yaml`

v10.5 to v10.9

> Note
>
> From v10.5, the following support packet data has changed:
>
> - The `support_packet.yaml` file has been removed and split into `diagnostics.yaml` and `stats.yaml` files.
> - All fields in `diagnostics.yaml` have been moved into their own objects for improved readability.
> - Field names are normalized.
> - New data includes server statistics, logs, permissions, and extended job list details.
> - Mattermost-supported plugin diagnostic data is included where applicable.

- metadata.yaml
- `mattermost.log` (Mattermost logs)
- `audit.log` (Mattermost audit logs)
- `ldap.log` (AD/LDAP logs)
- `notifications.log` (notifications logs)
- `plugins.json` (all active and inactive plugins)
- `sanitized_config.json` (sanitized copy of the Mattermost configuration)
- `stats.yaml` (Mattermost usage statistics)
- `jobs.yaml` (last runs of important jobs)
- `diagnostics.yaml` (core plugin diagnostics data)
- `permissions.yaml` (role & scheme information)

- Go performance metrics, including: `cpu.prof`, `heap.prof`, and `goroutines`
- `warning.txt` (present when issues are encountered during packet generation)
- `tsdb_dump.tar.gz` (present when the Metrics plugin is installed and the **Performance metrics** option is selected when generating the Support Packet)

The following additional plugin diagnostic data is included in the generated support packet when the plugin is enabled and operational:

- GitHub: `/github/diagnostics.yaml`
- GitLab: `/com.github.manland.mattermost-plugin-gitlab/diagnostics.yaml`
- Jira: `/jira/diagnostics.yaml`
- Calls: `/com.mattermost.calls/diagnostics.yaml`
- Boards: `/focalboard/diagnostics.yaml`
- Playbooks: `/playbooks/diagnostics.yaml`
- MSCalendar: `/com.mattermost.mscalendar/diagnostics.yaml`
- Google Calendar: `/com.mattermost.gcal/diagnostics.yaml`

Prior to v10.5

- metadata.yaml
- `mattermost.log`
- `plugins.json`
- `sanitized_config.json`
- `support_packet.yaml`
- `diagnostics.yaml` (core plugin diagnostics data)
- Go performance metrics, including: `cpu.prof`, `heap.prof`, and `goroutines`
- `warning.txt` (present when issues are encountered during packet generation)

Note

- From Mattermost v10.10, support packets from high availability deployments organize cluster-specific files (such as log files) in subdirectories named after each cluster node, while cluster-wide files remain in the root directory.
- Prior to v10.10, each node in the cluster of a high availability deployment has its own `mattermost.log` file and advanced logging files included directly in the support packet.
- LDAP groups are not included during Support Packet generation. Only `LDAP Version` and `LDAP Vendor` are included when present. These values are included in the `support_packet.yaml` file.
- From Mattermost v9.11, `LDAP Vendor` errors are included in the Support Packet. If fetching the LDAP Vendor name fails, the Support Packet generation includes the error in `warning.txt` . If no LDAP Vendor name is found, the Support Packet lists them as `unknown` .
- From Mattermost v10.4, a new `diagnostics.yaml` file includes Mattermost Calls diagostics data, including plugin version, calls and active session counts, as well as average duration and participant counts.

Metadata

From Mattermost v9.11, generated Support Packets include a `metadata.yaml` file that contains the following information.

Field name	Required/ Optional	Description	Example
version	Required	Version of the schema that the current metadata file is compatible with. Current version is 1.	1
type	Required	The type of the packet. Each type of Support Packet can be mapped to a specific component generating the Support Packet.	mattermost
generated_at	Required	The date and time the packet was created. Value is in epoch (ms).	1707473288731
server_version	Required	Version of the server that the Support Packet was	9.1.1

Field name	Required/ Optional	Description	Example
		generated at. Semver is expected.	
server_id	Required	Unique identifier of the server. Expected to be 26 characters or longer.	9qpiszyjr3g8bxda35abcd1234
license_id	Optional	Unique identifier of the current server's license. Expected to be 26 characters or longer. This field is empty when there's no license.	abcdejisd67yigqhmkz4ho1234
customer_id	Optional	The id of the customer, as defined in the license file. Expected to be 26 characters or longer. Empty when	a1b2c3d4qbbr5cpkbpbmef123h

Field name	Required/ Optional	Description	Example
		there's no license.	
extras	Optional	Key/value of any additional information, specific to the plugin/ component that generated the file. Can be useful for identifying the contents of the data. Consider adding plugin (or component) versions in order to set expectation regarding the contents of this object.	
extras.plugin_id	Required for plugins	The ID of the plugin.	
extras.plugin_version	Required for plugins	The version of the plugin.	

For example:

```
version: 1 type: support-packet generated_at: 1622569200 server_version:
9.1.1 server_id: 8fqk9rti13fmpxdd5934a3xsxh license_id:
3g3pqn8in3brzjkozcn1kdidgr customer_id: 74cmws7gf3ykpj31car7zahsny extras:
plugin_version: 0.1.0
```

Go performance metrics

The Support Packet contains 3 go runtime profiling files:

- `cpu.prof` contains a 5-second CPU profile
- `heap.prof` contains a heap profile
- `goroutines` contains a dump of all the running go routines

These files can be read using [pprof (https://golang.google.cn/cmd/pprof/)](https://golang.google.cn/cmd/pprof/).

Use `go tool pprof -web X` to open a visualization of the profile in your browser, replacing `X` with the profile's file name.

Load metric

Mattermost v10.10 introduces an additional diagnostic data point under the **Product Menu > About Mattermost** popup.

The Load Metric field displays monthly active users relative to the total number of licensed users, giving Mattermost support teams a contextual reference point for understanding deployment active usage for troubleshooting and guidance. It isn't a comprehensive performance monitoring tool or health indicator, but serves as a supplementary data point when traditional diagnostic methods aren't available.

On this page

- Generate a Support Packet
- Generate a Support Packet
  - Generate
    - Web/Desktop
    - mmctl

- Santitize confidential data
- Share the packet with Mattermost
- Contents of a Support Packet
  - v10.10 and later
  - v10.5 to v10.9
  - Prior to v10.5
- Metadata
- Go performance metrics
- Load metric
- Generate
- Santitize confidential data
- Share the packet with Mattermost
- Contents of a Support Packet
- Metadata
- Go performance metrics
- Load metric

Install a license key

You can use the System Console or the mmctl tools to add or change a Mattermost license key.

Use System Console

1. Go go **System Console > About > Edition and License**.
2. Upload your license key file.

Once the key is uploaded and installed, the details of your license are displayed.

Use mmctl

Use the mmctl license upload command to upload a new license or replace an existing license file with a new one. When complete, restart the Mattermost server. If you're running in a High Availability environment, the new license file must be updated to every node.

```
mmctl license upload [license] [flags]
```

Note

- Enterprise customers with the Premier Support add-on can request a staging license for testing.
- Removing a Mattermost Enterprise or Professional license key won't remove the configuration for Enterprise settings; however, these features won't function until an Enterprise or Professional license key is applied.
- When you're using High Availability, it's critical to ensure that all servers in the cluster have same Enterprise license properly installed to prevent multi-node clusters from failing. An Enterprise license is required for High Availability to work.
- When you apply an Enterprise license key to a server previously licensed for Professional, Professional features retain their configuration settings in Enterprise.
- When you apply a Professional license to a server previously licensed for Enterprise, Enterprise features retain their configuration but will no longer be accessible for use.

Change an existing license key

You don't need to wait for your current license key to expire before replacing it with a new license from Mattermost. The server only checks for seat count and license end date, not the start date, so you can apply a new license as soon as you receive it. However, ensure your new license is for a seat count that's greater than or equal to your current total number of Mattermost users.

> Tip
>
> To find the total number of users, go to **System Console > Reporting > Site Statistics**. The total number of users is displayed in the **Total Activated Users** field. The license will be rejected if this value is greater than the value allowed by your license key.

Follow these steps to change your license key:

1. Go to **System Console > About > Edition and License**.
2. From Mattermost v6.7, simply upload your new license key file.

> Note
>
> If you're running a Mattermost release older than v6.7, select **Remove Enterprise License and Downgrade Server** to clear the license from the server and refresh the System Console first before uploading the new key.

License key storage

Once you've uploaded your license key to your Mattermost server, it's stored in your SQL database at `mattermost.Licenses`. You can check what keys are on your server by running `select * from mattermost.Licenses;`.

On this page

- Install a license key

- Install a license key
  - Use System Console
  - Use mmctl
  - Change an existing license key
  - License key storage
  - Change an existing license key
  - License key storage

Migration

This Mattermost Migration Guide is organized into sections based on migration scenarios and tools to help you transition smoothly to Mattermost or optimize your current setup.

Whether you're migrating from another platform, upgrading your database, or using bulk tools for data management, this guide provides the resources and instructions you need for a successful migration. Use the navigation below to explore detailed guidance tailored to your migration needs.

- Migrate from MySQL to PostgreSQL - Learn how to migrate from MySQL to PostgreSQL.
- Server migration guide - Learn about about migrating to Mattermost.
- Migrate from Slack - Learn how to migrate from Slack to Mattermost.
- Bulk export tool - Learn about the bulk export tool for Mattermost.
- Bulk loading tool - Learn about the bulk loading tool for Mattermost.
- Migration announcement email template - Use this email template to notify your users that you've migrated to Mattermost.

Monitoring and performance

This Monitoring and Performance Guide is organized into sections to help you effectively monitor, optimize, and manage the performance of your Mattermost installation.

From collecting performance metrics and deploying monitoring tools to configuring health checks and managing notifications, this guide offers comprehensive resources to ensure your Mattermost workspace operates at peak efficiency. Use the navigation below to explore detailed instructions and best practices.

- Optimize your Mattermost workspace - Learn about optimizing your Mattermost workspace.
- Collect performance metrics - Learn about collecting performance metrics for Mattermost.
- Deploy Prometheus and Grafana for performance monitoring - Learn how to deploy Prometheus and Grafana for performance monitoring.
- Performance monitoring metrics - Learn about performance monitoring metrics for Mattermost.
- Push notification health targets - Learn about push notification health targets for Mattermost.
- Performance alerting guide - Learn about performance alerting for Mattermost.
- Ensuring releases perform at scale - Learn how to ensure releases perform at scale for Mattermost.
- Manage user surveys - Learn about managing user surveys for Mattermost.
- User satisfaction surveys - Learn how to send user satisfaction surveys for Mattermost.
- Notify admin - Learn how to notify admins for Mattermost.

- System-wide notifications - Learn about system-wide notifications for Mattermost.
- Statistics - Learn about Mattermost statistics .
- In-product notices - Learn how to use in-product notices for Mattermost.
- Health checks - Learn about health checks for Mattermost.
- Health check probes - Learn how to set up health check probes for Mattermost.
- Product limits - Learn about product limits for Mattermost.

Self-hosted billing

Mattermost Enterprise and Professional plans are offered as an annual subscription service. With your purchase you're provided with a license key. Depending on which version of Mattermost you're on, you may have to download and apply this license manually via the System Console.

When you upgrade to an Enterprise or Professional plan, you can enter the number of users (also referred to as "seats") before you complete your purchase and the license key is generated. The number you enter has to be equal to or greater than your current number of activated users (formerly called active users).

You can add new users to your deployment at any time. Every quarter, during the true-up review (section 3.5 License True-up Review of Software License Agreement, also referenced in docs), we check the license license seat count relative to number of activated users. If there's a difference, we'll provide you with a new invoice which is payable immediately.

Please see the self-hosted subscriptions documentation for more detailed information.

Server configuration

This Server Configuration Guide is organized into sections to provide you with the tools and knowledge necessary to configure your Mattermost server for improved efficiency, scalability, and functionality.

Whether you're setting up email notifications, optimizing search capabilities, enabling high availability, or configuring telemetry, this guide covers all aspects of server setup and management. Use the navigation below to access detailed instructions for each topic.

- Store configuration in your database - Learn how to store configuration information in your Mattermost database rather than as a JSON file.
- Server configuration options - Learn about server configuration options for Mattermost.
- Set up attribute-based access controls - Learn how to set up attribute-based access controls for your Mattermost instance for Zero Trust Security.
- Set up Mattermost Agents - Learn how to enable AI-powered Agents for your Mattermost instance.
- Install Mattermost Boards - Learn how to install and configure the Boards plugin for your Mattermost instance.
- Manage custom user attributes - Learn how to manage custom user attributes in user profiles in Mattermost.
- Environment variables - Learn how to use environment variables for Mattermost configuration.
- Customize the server - Learn about customizing branding for Mattermost server.
- SMTP email setup - Learn how to set up SMTP email for Mattermost.
- Email templates - Learn about customizing email templates for Mattermost.
- Chinese, Japanese, and Korean search - Learn about enabling Chinese, Japanese, and Korean search for Mattermost.

- SSL client certificate setup - Learn how to set up SSL client certificates for Mattermost.
- Connected workspaces - Learn how to connect Mattermost workspaces.
- Telemetry - Learn about Mattermost telemetry and data collection.

Server maintenance

This Server Maintenance Guide is organized into sections that provide the tools and knowledge needed to maintain your Mattermost server effectively, ensuring optimal security, scalability, and reliability.

Whether you’re installing a license key, performing backups, upgrading the server, or using administrative tools like mmctl and the CLI, this guide offers comprehensive instructions to help you manage your server with confidence. Use the navigation below to access detailed information on each topic.

- Install a license key - Learn how to install a license key for Mattermost.
- Generate a support packet - Learn how to generate a support packet for Mattermost.
- Backup and disaster recovery - Learn about backup and disaster recovery for Mattermost.
- Upgrade Mattermost server - Learn how to upgrading Mattermost server.
- Secure Mattermost - Learn about securing Mattermost server.
- Mattermost error codes - Learn about Mattermost error codes and troubleshooting.
- Logging - Learn how to customize logging options based on business practices and needs.
- mmctl - Learn about the mmctl command line tool for Mattermost.
- CLI - Learn about command line tools for Mattermost.
- Feature labels - Learn about Mattermost feature labels and their meanings.

User attributes

From Mattermost v10.8, ensure your teams always have the critical information they need to collaborate effectively by defining and managing organization-specific user profile attributes as system attributes that you can synchronize with your AD/LDAP or SAML identity provider.

System attributes enable you to customize user profile attributes to match your organization's unique needs and streamline collaboration while keeping user data centralized and consistent. The attributes you define, such as position, rank, or location, are attributes users can manage as part of their user profile. These custom profile attributes supplement existing user details visible from the user's profile picture.

Before you begin

Mattermost Enterprise v10.9 or later

From Mattermost Enterprise v10.9, the ability to create and manage system properties as a Mattermost system admin requires no manual setup.

Mattermost Enterprise v10.8

You'll need Mattermost Enterprise v10.8 or later deployment, and be a Mattermost system admin to enable the system properties feature flag, `MM_FEATUREFLAGS_CUSTOMPROFILEATTRIBUTES` to create and manage system properties. See the Mattermost developer documentation for details on enabling feature flags in a self-hosted deployment. Mattermost Cloud customers can request this feature flag be enabled by contacting their Mattermost Account Manager or by creating a support ticket (https://support.mattermost.com/hc/en-us/requests/new?ticket_form_id=11184911962004) request.

If you plan to synchronize system properties with your AD/LDAP or SAML identity provider, ensure AD/LDAP or SAML synchronization is already enabled and configured. See the AD/LDAP groups product documentation or SAML 2.0 configuration settings documentation for details.

Add attributes

You can define and manage up to 20 system attributes using the System Console. Each attribute becomes a user profile option users can populate. Once you reach the maximum of 20 attributes, you can't create new attributes until you delete attributes you no longer need.

1. In the System Console, go to **Site Configuration > System Attributes > User Attributes** and select **Add Attribute**.
2. Enter the following details:

> - ***Attribute name****: Enter a unique name for the attribute. Attribute names can be up to 40 characters long.*
> - ***Type****: Specify the type of attribute as one of the following:*
> - ***Text*** *for text-based profile attributes.*
> - ***Phone*** *for phone number-based profile attributes.*
> - ***URL*** *for web site address-based profile attributes.*
> - ***Select*** *for a list of profile attribute values users can choose from. Specify each value followed by pressing* ***TAB*** *or* ***ENTER****. Values*

> *can be up to 64 characters long, and users can choose a single value.*
>
> - ***Multi-select*** *for a list of profile attribute values users can select from. Specify each value followed by pressing* ***TAB*** *or* ***ENTER****. Values can be up to 64 characters long, and users can choose multiple values.*

3. Specify the attribute's visibility as one of the following:

> - ***Always show****: The attribute is always visible in user profiles.*
> - ***Hide when empty****: The attribute is only visible in user profiles when it has a value.*
> - ***Always hide****: The attribute is never visible in user profiles.*

4. Save your changes.

> Tip
>
> Duplicate existing attributes by selecting **More**
>
> •••
>
> and selecting **Duplicate attribute**. This creates a new attribute with the same name and type as the original attribute. You can then edit the new attribute to change its name, type, and visibility as needed.

Manage attributes

- **Modify**: Select the attribute fields to make inline changes to the attribute's name, type, or values. Select **More**

  •••

  to change a attribute's visibility.

- **Order**: Control the order you want attributes to appear in user profiles by dragging and dropping them in the list.

- **Delete**: Delete attributes you no longer need or want by selecting **More**

  •••

  and selecting **Delete property**.

In cases where multiple system admins manage system attributes, refresh your web browser instance to see real-time updates to system attributes made by other admins.

Sync attributes with your identity provider

1. Synchronize a attribute with AD/LDAP or SAML by selecting **More**

   •••

   and selecting **Link attribute to AD/LDAP** or **Link attribute to SAML**. Mattermost takes you directly to the **AD/LDAP** or **SAML 2.0** configuration settings page in the System Console where you can map the attributes you want to synchronize.

2. Scroll to the **Custom profile attributes sync** section to specify the attribute used to populate the attribute in user profiles.

3. Specify the attribute mapping for the attribute by entering the attribute name in the system attribute's **Attribute** field. The attribute name must match the attribute name in your identity provider.

4. Save your changes.

On this page

- User attributes
- User attributes
  - Before you begin
    - Mattermost Enterprise v10.9 or later
    - Mattermost Enterprise v10.8
  - Add attributes
  - Manage attributes
  - Sync attributes with your identity provider
  - Before you begin

- Add attributes
- Manage attributes
- Sync attributes with your identity provider

User management

Welcome to the Mattermost User Management Guide. This User Management Guide is organized into sections to help you manage users, permissions, and roles effectively in your Mattermost workspace.

Whether you’re configuring team and channel settings, managing guest accounts, or leveraging advanced permissions infrastructure, this guide provides the resources and instructions necessary to tailor user management to your organization’s needs. Use the navigation below to explore detailed guidance for each area.

- Permissions - Learn about permissions in Mattermost.
- Manage team and channel configuration - Learn about managing team and channel configuration in Mattermost.
- Advanced permissions infrastructure - Learn about advanced permissions infrastructure in Mattermost.
- Guest accounts - Learn about guest accounts in Mattermost.

User provisioning

- Corporate directory integrations - Mattermost integrates with all major account providers via Active Directory, SAML, and OAuth.
- Provisioning workflows - Learn about provisioning workflows in Mattermost.
- AD/LDAP setup - Learn how to set up AD/LDAP in Mattermost.
- AD/LDAP manage team or private channel membership - Learn how to manage team or private channel membership using AD/LDAP sync groups in Mattermost.
- GitLab SSO - Learn how to use GitLab SSO in Mattermost.
- OpenID Connect SSO - Learn how to use about OpenID Connect SSO in Mattermost.
- Google SSO - Learn how to use Google SSO in Mattermost.
- Entra ID SSO - Learn how to use Entra ID SSO in Mattermost.
- Convert OAuth 2.0 providers to OpenID - Learn how to convert OAuth 2.0 providers to OpenID in Mattermost.

Bulk export tool

Moving data from one Mattermost instance into another begins with exporting data to a JSONL (https://jsonlines.org) file using the bulk loading feature. This tool is useful if you have created a server for a proof of concept, have created another server for production use, and now want to retain the history from the proof of concept instance.

You can export the following data types:

- Teams
- Channels (public, private, and direct)
- Messages (regular, non-reply, and direct messages)
- Threaded discussions and message replies
- Message reactions
- Users
- Users' preferences, including pinned and saved messages
- Users' team memberships
- Users' channel memberships
- Users' notification preferences
- Custom emoji
- Direct message and group message channels, including read/unread status
- Roles
- Permissions schemes
- Bot users

> Note
>
> Configuration for data types such as exporting specific areas of the server, exporting additional types of posts, file attachments, webhooks, and bot messages is not yet supported. Deleted objects are also not yet supported.
>
> For requests to add additional attributes or objects to our exporter, please add a feature request on our [feature idea forum (https://portal.productboard.com/mattermost/33-what-matters-to-you)](https://portal.productboard.com/mattermost/33-what-matters-to-you).

Bulk export data

Use mmctl

1. Create a full export file including attachments by running the mmctl export create – attachments command. See the Mattermost data migration documentation for details.
2. While the job is running, you can check its status by running the mmctl export job show command.
3. When the export job status is successful:

> *1. Identify the name of the completed export file by running the mmctl export list command.*
>
> *2. Download the export file to your local machine by running the mmctl export download command.*

Use CLI

> Note
>
> From Mattermost v6.0, this command has been deprecated in favor of mmctl export commands as the supported way to export data out of Mattermost.

The export command runs in the CLI. It has permissions to access all information in the Mattermost database.

To run the export command:

1. Navigate to the directory where the Mattermost server is installed. On a default install of Mattermost, the directory is `/opt/mattermost`.
2. Run the following command to extract data from all teams on the server. Note that you can change the file name and specify an absolute or relative path to dictate where the file is exported:

   `sudo -u mattermost bin/mattermost export bulk file.json --all-teams`

   `sudo -u mattermost bin/mattermost export bulk /home/user/bulk_data.json --all-teams`

3. Retrieve your file from the location you specified.

All Mattermost bulk export data files will begin with a `Version` object as the first line of the file. This indicates the version of the Mattermost bulk import file format with which the exported data is compatible.

Version object

Field name	Type	Description
type	string	The string "version"
version	number	The number 1.
info	object	Optional VersionInfo object

VersionInfo object

Field name	Type	Description
generator	string	The name of the tool this export was generated with. Well known tools are: `"mattermost-server"` for the Mattermost Server. `"mmetl"` for the Slack export converter "mmetl".
version	string	The version of the tool this export was generated with. This may contain multiple pieces of version info, separated by spaces. The first one should be a semantic version. `"7.6.0 (29bb1e53ef5a439c73065f47de2972f9bbcb09a4, enterprise: true)"` is an example of such a version string.
created	string	The timestamp of the file creation. This should be formatted as an RFC 3339 timestamp. The nanosecond part is optional. `"2022-11-22T16:40:51.019582328+01:00"`
additional	any	Any additional information the generator wants to include into the file header. May be omitted. Be aware that the size of each line is limited to a few MiB.

Team object

Field name	Type	Description
name	string	The team name.
display_name	string	The display name for the team.
type	string	The type of team. Will have one of the following values: `"O"` for an open team `"I"` for an invite-only team.
description	string	The team description.

allow_open_invite	bool	Whether to allow open invitations. Will have one of the following values:
		`"true"`
		`"false"`

scheme	string	The name of the permissions scheme that applies to this team.

Channel object

Field name	Type	Description
team	string	The name of the team this channel belongs to.
name	string	The name of the channel.
display_name	string	The display name for the channel.
type	string	The type of channel. Will have one the following values: `"O"` for a public channel. `"P"` for a private channel.
header	string	The channel header.
purpose	string	The channel purpose.
scheme	string	The name of the permissions scheme that applies to this team.

User object

Field name	Type	Description
username	string	The user's username. This is the unique identifier for the user.
email	string	The user's email address.
auth_service	string	The authentication service used for this user account. This field will be absent for user/password authentication. `"gitlab"` - GitLab authentication. `"ldap"` - LDAP authentication (Enterprise and Professional) `"saml"` - Generic SAML based authentication (Enterprise) `"google"` - Google OAuth authentication (Enterprise) `"entra id"` - Microsoft Entra ID OAuth Authentication (Enterprise)
auth_data	string	The authentication data if `auth_service` is used. The value depends on the `auth_service` that is specified. The data comes from the following fields for the respective auth_services: `"gitlab"` - The value of the Id attribute provided in the GitLab auth data. `"ldap"` - The value of the LDAP attribute specified as the "ID Attribute" in the Mattermost LDAP configuration. `"saml"` - The value of the SAML email address attribute. `"google"` - The value of the OAuth Id attribute. `"office365"` - The value of the OAuth Id attribute.
nickname	string	The user's nickname.
first_name	string	The user's first name.
last_name	string	The user's last name.
position	string	The user's position.

roles	string	The user's roles.
locale	string	The user's localization configuration.
use_markdown_preview	bool	`"true"` if the user has enabled markdown preview in the message input box.
use_formatting	bool	`"true"` if the user has enabled post formatting for links, emoji, text styles, and line breaks.
show_unread_section	string	`"true"` if the user has enabled showing unread messages at top of channel sidebar.
theme	string	The user's theme. Formatted as a Mattermost theme string.
military_time	string	`"true"` if the user has enabled a 24-hour clock. `"false"` if using a 12-hour clock.
collapse_previews	string	`"true"` if user collapses link preview by default. `"false"` if user expands link previews by default.
message_display	string	The style the user prefers for displayed messages. Options are `"clean"` if the user uses the standard style or `"compact"` if the user uses compact style.
channel_display_mode	string	`"full"` if the users displays channel messages at the full width of the screen or `"centered"` if the user uses a fixed width, centered block.
tutorial_step	string	`"1"`, `"2"`, or `"3"` indicates which specified tutorial step to start with for the user. `"999"` skips the tutorial.
email_interval	string	Email batching interval to use during bulk import.
delete_at	int64	Timestamp of when the user was deactivated.
teams	array	The teams which the user is member of. Is an array of **UserTeamMembership** objects.
notify_props	object	

		The user's notify preferences, as defined by the **UserNotifyProps** object.

UserNotifyProps object

This object is a member of the User object.

Field name	Type	Description
desktop	string	Preference for sending desktop notifications. Will be one of the following values: `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.
desktop_sound	string	Preference for whether desktop notification sound is played. Will be one of the following values: `"true"` - Sound is played. `"false"` - Sound is not played.
email	string	Preference for email notifications. Will be one of the following values: `"true"` - Email notifications are sent immediately. `"false"` - Email notifications are not sent.
mobile	string	Preference for sending mobile push notifications. Will be one of the following values: `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.
mobile_push_status	string	Preference for when push notifications are triggered. Will be one of the following values: `"online"` - When online, away or offline. `"away"` - When away or offline. `"offline"` - When offline.
channel	string	Preference for whether @all, @channel, and @here trigger mentions. Will be one of the following values: `"true"` - Mentions are triggered. `"false"` - Mentions are not triggered.
comments	string	Preference for reply mention notifications. Will be one of the following values: `"any"` - Trigger notifications on messages in reply threads that the user starts or participates in. `"root"` - Trigger notifications on messages in threads that the user starts. `"never"` - Do not trigger notifications on messages in reply threads unless the user is mentioned.

mention_keys	string	Preference for custom non-case sensitive words that trigger mentions. Words are separated by commas.

UserTeamMembership object

Field name	Type	Description
name	string	The name of the team this user is a member of.
roles	string	The roles the user has within this team.
theme	string	The user's theme for this team. Formatted as a Mattermost theme string.
channels	array	The channels within this team that the user is a member of. Listed as an array of **UserChannelMembership** objects.

UserChannelMembership object

Field name	Type	Description
name	string	The name of the channel in the parent team that this user is a member of.
roles	string	The roles the user has within this channel.
notify_props	object	The notify preferences for this user in this channel as defined by the **ChannelNotifyProps** object.
favorite	boolean	Whether the channel is marked as a favorite for this user. Will be one of the following values: `"true"` - Yes. `"false"` - No.
mention_count	int64	The mention preferences for this user in this channel as defined by the **ChannelMentionCount** object.
mention_count_root	int64	The mention preferences for this user in this channel as defined by the **ChannelMentionCountRoot** object.
urgent_mention_count	int64	The mention preferences for this user in this channel as defined by the **UrgendMentionCount** object.
msg_count	int64	The mention preferences for this user in this channel as defined by the **MsgCount** object.
msg_count_root	int64	The mention preferences for this user in this channel as defined by the **MsgCountRoot** object.
last_viewed_at	int64	The mention preferences for this user in this channel as defined by the **LastViewedAt** object.

ChannelNotifyProps object

This object is a member of the ChannelMembership object.

Field name	Type	Description
desktop	string	Preference for sending desktop notifications. Will be one of the following values: `"default"` - Global default. `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.
mobile	string	Preference for sending mobile notifications. Will be one of the following values: `"default"` - Global default. `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.
mark_unread	string	Preference for marking channel as unread. Will be one of the following values: `"all"` - For all unread messages. `"mention"` - Only for mentions.

Post object

Field name	Type	Description
team	string	The name of the team that this post is in.
channel	string	The name of the channel that this post is in.
user	string	The username of the user for this post.
message	string	The message that the post contains.
props	object	The props for a post. Contains additional formatting information used by integrations and bot posts. For a more detailed explanation see the message attachments documentation.
create_at	int	The timestamp for the post, in milliseconds since the Unix epoch.
reactions	array	The emoji reactions to this post. Will be an array of Reaction objects.

Reply object

Field name	Type	Description
user	string	The username of the user for this reply.
message	string	The message that the reply contains.
create_at	int	The timestamp for the reply, in milliseconds since the Unix epoch.

Reaction object

This object is a member of the Post object.

Field name	Type	Description
user	string	The username of the user for this reply.
emoji_name	string	The emoji of the reaction.
create_at	int	The timestamp for the reply, in milliseconds since the Unix epoch.

Emoji object

Field name	Type	Description
name	string	The emoji name.
image	string	The path (either absolute or relative to the current working directory) to the image file for this emoji.

DirectChannel object

Field name	Type	Description
members	array	List of channel members.
favorited_by	array	List of channel members who have favorited the direct channel.
header	string	The channel header.

DirectPost object

Field name	Type	Description
user	string	The username of the user for this post.
message	string	The message that the post contains.
create_at	int	The timestamp for the post, in milliseconds since the Unix epoch.

On this page

- Bulk export tool
- Bulk export tool
  - Bulk export data
    - Use mmctl
    - Use CLI
  - Version object
  - VersionInfo object
  - Team object
  - Channel object
  - User object
  - UserNotifyProps object
  - UserTeamMembership object
  - UserChannelMembership object
    - ChannelNotifyProps object
  - Post object
  - Reply object
  - Reaction object
  - Emoji object
  - DirectChannel object
  - DirectPost object
  - Bulk export data
  - Version object

- VersionInfo object
- Team object
- Channel object
- User object
- UserNotifyProps object
- UserTeamMembership object
- UserChannelMembership object
  - ChannelNotifyProps object
- Post object
- Reply object
- Reaction object
- Emoji object
- DirectChannel object
- DirectPost object

Cloud Dedicated Bring Your Own Key

Bring Your Own Key (BYOK) provides Enterprise Cloud customers with autonomy over their encryption key life cycle. BYOK supports encryption at rest with custom KMS keys that the enterprise provides and maintains.

BYOK requires a subscription to Mattermost Cloud Enterprise Dedicated, which offers enhanced data security and compliance by ensuring that enterprises have full control over their data encryption processes.

In Mattermost Cloud Enterprise Dedicated, you can use KMS keys in 2 ways:

- One KMS key for all services; or,
- Per-service KMS keys (EBS, RDS, S3)
    - Keys do not need to be unique to each service.
    - All services must be encrypted at rest.
    - Selective enablement of this feature can be supported.
    - In cases where a global database is needed, we recommend providing 2 KMS keys (1 per region).

Configure BYOK

1. Enterprise customer provides their AWS KMS ARN to the Mattermost Infrastructure SRE team.
2. Enterprise customer adds the following blocks to their KMS Policy for the AWS KMS ARN provided:

```
{  "Sid": "Allow use of the key",  "Effect": "Allow",  "Principal": {
"AWS": "arn:aws:iam::<MATTERMOST_AWS_ACCOUNT_ID>:user/mattermost-cloud-
<environment>-provisioning-<VPC_ID>"  },  "Action": [  "kms:Encrypt",
"kms:Decrypt",  "kms:ReEncrypt*",  "kms:GenerateDataKey*",
"kms:DescribeKey"  ],  "Resource": "<CUSTOM_CUSTOMER_KMS_ID>" }, {  "Sid":
"Allow use of the key role nodes",  "Effect": "Allow",  "Principal": {
"AWS": "arn:aws:iam::<MATTERMOST_AWS_ACCOUNT_ID>:role/nodes.<CLUSTER_ID>-
kops.k8s.local"  },  "Action": [  "kms:Encrypt",  "kms:Decrypt",
"kms:ReEncrypt*",  "kms:GenerateDataKey*",  "kms:DescribeKey"  ],
"Resource": "<CUSTOM_CUSTOMER_KMS_ID>" },
```

3. The Mattermost Infrastructure SRE team updates the kops cluster and S3, RDS resources after the KMS policy is updated on the customer's end.

Alternatively, the Enterprise customer can provide an external key (non-KMS) to the Mattermost Infrastructure SRE team that Mattermost maintains on behalf of the customer. This path offers less control to customers but simplifies the setup process.

Requirements

- Customers must own their AWS Account. (In the alternative path mentioned above this is delegated to Mattermost.)
- Customers oversee the maintenance life cycle of their custom KMS key.
- A valid AWS KMS ARN for encrypting storage and databases should be provided to the Infrastructure SRE team.
- The customer should incorporate the provided policy blocks from the Infrastructure SRE team into their KMS key policy.

Considerations

- Changing the AWS KMS key in the database necessitates downtime due to AWS Aurora's encryption [limitations. (https://repost.aws/knowledge-center/update-encryption-key-rds)](https://repost.aws/knowledge-center/update-encryption-key-rds)
- Proper communication is essential for setting expectations and scheduling changes.

Conclusion

If you are a large enterprise with compliance requirements, or are working in highly-regulated industries, using Mattermost Cloud Dedicated with BYOK ensures full data control.

For any further assistance or queries, talk to a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/)

On this page

- Cloud Dedicated Bring Your Own Key
  - Configure BYOK
    - Requirements
    - Considerations
  - Conclusion

Mattermost workspace migration

This document outlines the process for migrating an existing Mattermost instance from self-hosted to Cloud, and from Cloud to self-hosted.

> Note
>
> Migrating an existing Mattermost instance to another using the same deployment method follows the same process that's documented below, regardless as to whether the source or destination of the migration is in the Cloud or self-hosted. **These steps will work for both Cloud and self-hosted Mattermost instances**.

Migrate from self-hosted to Cloud

When you migrate from self-hosted to Cloud, you'll need to open a ticket with the Mattermost Support team so they can assist you with the process. The information below describes the migration process. Before you get started, visit our [Support site (https://support.mattermost.com/hc/en-us/requests/new)](https://support.mattermost.com/hc/en-us/requests/new) to open a ticket.

Important

- This migration process is only available to customers using paid Mattermost editions.
- We recommend reviewing the knowledge base article on migrating from self-hosted to Cloud before starting a migration to familiarize yourself with key considerations, required prerequisites, and our guidance on testing your export.

Before you begin your migration, take note of the following information before you begin:

User Authentication

If you are using the email login method, users will need to reset their passwords after the migration has been completed. Other authentication methods, such as LDAP and GitLab SSO, required changes to your infrastructure allowing the specific authentication method to function in Mattermost cloud.

Plugins

If you're using plugins that aren't listed on the Marketplace, they won't be included in the export and you won't have access to them going forward. See the list of supported plugins by visiting the integrations overview documentation.

Data

The migration only includes data from Channels. No Playbooks data is exported.

Migration process

Export from your self-hosted instance

Use your administrator credentials to log into your self-hosted Mattermost server. Once you're logged in, run:

```
mmctl export create --attachments
```

This creates a full export of the server, and includes attached files. If you don't want to export attached files, leave out `--attachments`.

This process can take some time, so `mmctl` will return immediately, and the job will run in the background until the export is fully created. If successful, the command will immediately output a job ID, like this:

```
Export process job successfully created, ID: yfrr9ku5i7fjubeshs1ksrknzc
```

While the job is running, its status can be checked using the ID that was provided when it was created, and when it's done the output will look similar to this:

```
mmctl export job show yfrr9ku5i7fjubeshs1ksrknzc ID:
yfrr9ku5i7fjubeshs1ksrknzc Status: success Created: 2021-11-03 10:44:13
-0500 CDT Started: 2021-11-03 10:44:23 -0500 CDT
```

The completed file will be downloaded to your desktop as a `.zip` file.

> Note
>
> Do not rename the file as the file name is referenced in log files, which are used by the Support team to validate the exported file.

The Support team will provide you with S3 credentials so you can upload the exported file. Once you've uploaded the file, please contact the Support team and let them know.

Create a new workspace on the Mattermost Cloud

In the meantime, you can log into Mattermost Cloud with your Mattermost credentials and create a Cloud workspace.

> Note
>
> Do not create any users in your Mattermost Cloud instance as the migration process performs this task for you.

Importing your data into your Mattermost Cloud instance

Once the export upload to the provided S3 bucket is complete and you've shared your Mattermost Cloud instance name/URL, Support can begin the import step.

Depending on the size of the export this process can take some time. Support will contact you as soon as the import is complete. During this time it is highly recommended you do not use your Mattermost Cloud instance.

Start using your Mattermost Cloud instance

When the export is complete, you can log into your Cloud instance and can invite your users to log in.

> Note
>
> We recommend that you keep your self-hosted Mattermost server until you've been using your Cloud instance for a while and all is verified as is as expected.

Migrate from Cloud to self-hosted

You can migrate your Cloud workspace data to a self-hosted deployment at any time.

How does the process work?

Before you export and migrate your data, you must install Mattermost on the server you'll be using to run Mattermost. The migration is done using the mmctl CLI tool, which is a remote CLI tool for Mattermost that's installed locally and uses the Mattermost API. `mmctl` is pre-installed.

The mmctl usage notes provide some additional context and information which you can reference before and during the process.

You'll be using the mmctl export commands to export your Cloud data for channels, messages, users, etc. The export file is downloaded to a location specified in the export commands. Once the export is complete, you'll import the data into your self-hosted instance.

Alternatively, you can export the data to an Amazon S3 cloud storage location in cases where an export is quite large and challenging to download from the Mattermost server. See the create the export section below for details.

> Note
>
> Prior to migrating your data from Cloud, please ensure you have the appropriate permissions within your organization to carry out the data export which may contain sensitive information. Mattermost is not liable for any actions taken after the data export.
>
> Moreover, the export process doesn't include integrations or any custom data. Other aspects of your instance, such as specific security settings and requirements, are also not included. For assistance with migrating additional data and settings, see our support options: https://mattermost.com/support/.

Once `mmctl` is authenticated, you can generate the export from the source instance.

Authenticate

Authentication is done with either Mattermost login credentials or an authentication token. First, use your administrator credentials to log into the instance with mmctl, replacing *example-source-domain.com* with the network address of the source instance:

```
mmctl auth login https://yourdomain.cloud.mattermost.com
```

You'll be prompted for a username (use your admin user), password, and for a connection name. The connection name can be anything you want, and it's used to identify this set of credentials in the future, for your convenience. Then you will be able to start the export process.

Create the export

Important

If your Mattermost Cloud deployment includes plugins that aren't listed on the Cloud Marketplace, those plugins won't be included in the export, and you

won't have access to those plugins going forward. See the integrations documentation for a list of Cloud-supported integrations.

Once you're logged in, run the following `mmctl` command:

```
mmctl export create
```

Running this command creates a full export of the server, including attached files. Append `--no-attachments` if you do not wish to export attached files from your instance, and append `--with-archived-channels` to include archived channels in the export file. This process can take some time, so `mmctl` will return immediately, and the job will run in the background on the Mattermost instance until the export is fully created. If successful, the command will immediately output a job ID, like this:

```
Export process job successfully created, ID: yfrr9ku5i7fjubeshs1ksrknzc
```

While the job is running, its status can be checked using the ID that was provided when it was created, and when it's done the output will look similar to this:

```
mmctl export job show yfrr9ku5i7fjubeshs1ksrknzc ID:
yfrr9ku5i7fjubeshs1ksrknzc Status: success Created: 2021-11-03 10:44:13
-0500 CDT Started: 2021-11-03 10:44:23 -0500 CDT
```

Once the status is `success`, download the export onto your local machine. First, discover the name of the completed export file with `mmctl export list`:

```
mmctl export list r3kcj8yuwbramdt714doafi3oo_export.zip
```

This will show all of the exports on the server, so be sure to download the latest one and to delete it when you're done to save storage. Generate a link to download the file with a command like the following, but with the filename of the export on your server:

```
mmctl export generate-presigned-url r3kcj8yuwbramdt714doafi3oo_export.zip
```

Tip

- As an alternative to this last step, from your Mattermost Cloud web instance, you can retrieve the file download link to the export by using the Mattermost slash command `/exportlink [job-id|zip file|latest]`. Use the `latest` option to automatically pull the latest export available, or specify the download link by `job-id` or `zip file`.
- Mattermost v8.1.0-RC1 is required to use the `mmctl export generate-presigned-url` command on a self-hosted Mattermost instance. Access the [Mattermost Enterprise v8.1.0-RC1 binary (https://releases.mattermost.com/8.1.0-rc1/mattermost-enterprise-8.1.0-rc1-linux-amd64.tar.gz)](https://releases.mattermost.com/8.1.0-rc1/mattermost-enterprise-8.1.0-rc1-linux-amd64.tar.gz) or the [Mattermost Team Edition v8.1.0-RC1 binary (https://releases.mattermost.com/8.1.0-rc1/mattermost-team-8.1.0-rc1-linux-amd64.tar.gz)](https://releases.mattermost.com/8.1.0-rc1/mattermost-team-8.1.0-rc1-linux-amd64.tar.gz).

Upload the export to the new server

Finally, it's time to take our export from the source server and use it as an import into the destination server. Before proceeding, review and modify the following self-hosted Mattermost configuration settings, where applicable, to ensure a smooth and successful import.

Mattermost configuration setting	**Large file import recommendation**
Maximum Users Per Team	Increase this value to a number that **exceeds** the maximum number of users, per team, in the import file.
Maximum File Size	Temporarily increase this value to be **larger** than the size of the import file. Following a successful import, we strongly recommend reverting this value to a reasonable limit for daily expected usage.
Write Timeout	Temporarily adjust this value based on import file speed and network path to enable the file to upload without timeouts. Start with a value of **3600** and adjust if needed. Following a successful import, we strongly recommend reverting this setting to its initial or previous value.
Read Timeout	Temporarily adjust this value based on import file speed and network path to enable the file to upload without timeouts. Start with a value of **3600** and adjust if needed. Following a successful import, we strongly recommend reverting this setting to its initial or previous value.
Amazon S3 Request Timeout	If using cloud-based file storage, adjust this value to ensure your storage requests don't time out too soon.

Next, log into the destination server using `mmctl` the same way you logged into the source server:

```
mmctl auth login https://localinstance.company.com
```

Use the following command to upload the export to the destination server. The speed of the upload may vary based on connection speed. When the upload is complete the command will return with the ID of the import:

```
mmctl import upload r3kcj8yuwbramdt714doafi3oo_export.zip Upload session
successfully created, ID: cfuq6q9kkjrqfgnph1pew3db4e Import file
successfully uploaded, name: xrzs9wrzufntbfcxpy39mdq9hy
```

Complete the import into the new server

```
mmctl import list available
cfuq6q9kkjrqfgnph1pew3db4e_r3kcj8yuwbramdt714doafi3oo_export.zip
```

Run the import job to process to import the export file into the server. The speed of this process may vary based on connection speed. First, start the import process:

```
mmctl import process
cfuq6q9kkjrqfgnph1pew3db4e_r3kcj8yuwbramdt714doafi3oo_export.zip
```

Once you've marked the file for processing, you can check the status of the job using `mmctl import job list`:

```
mmctl --local import job list ID: f93jxu1hzty79enwa1xy6f1tbr Status:
pending Created: 2021-10-28 13:32:55 +0200 CEST
```

When the job is complete, the `success` status is displayed:

```
mmctl --local import job list ID: f93jxu1hzty79enwa1xy6f1tbr Status:
success Created: 2021-10-28 13:32:55 +0200 CEST Started: 2021-10-28
13:33:05 +0200 CEST
```

Then extract the export file to use it by running the following mmctl command:

```
mmctl extract run [flags].
```

Once your migration is complete and you've imported your data into your self-hosted instance we recommend that you take a few days to validate your data

and ensure everything is working as expected before taking down your Cloud instance.

> Note
>
> If you are using email/password authentication, your users must reset their passwords.

If you encounter any issues or problems, contact our support team (https://mattermost.com/support/), or visit the Mattermost Help Center (https://support.mattermost.com/).

Migrate from self-hosted to Cloud

When you migrate from self-hosted to Cloud, you'll need to open a ticket with the Mattermost Support team so they can assist you with the process. The information below describes the migration process. Before you get started, visit our Support site (https://support.mattermost.com/hc/en-us/requests/new) to open a ticket.

> Note
>
> This migration process is only available to customers using paid Mattermost editions.

Before you begin your migration, take note of the following information before you begin:

User Authentication

If you are using the email login method, users will need to reset their passwords after the migration has been completed. Other authentication methods, such as LDAP and GitLab SSO, required changes to your infrastructure allowing the specific authentication method to function in Mattermost cloud.

Plugins

If you're using plugins that aren't listed on the Marketplace, they won't be included in the export and you won't have access to them going forward. You can view the list of plugins in the Support knowledgebase (https://support.mattermost.com/hc/en-us/articles/5346624843924-Mattermost-Plugin-Marketplace-List-as-of-November-12th-2024).

Data

The migration only includes channel data. No collaborative playbooks data is exported.

Migration process

Export from your self-hosted instance

Use your administrator credentials to log into your self-hosted Mattermost server. Once you're logged in, run:

```
mmctl export create --attachments
```

This creates a full export of the server, and includes attached files. If you don't want to export attached files, leave out `--attachments`.

This process can take some time, so `mmctl` will return immediately, and the job will run in the background until the export is fully created. If successful, the command will immediately output a job ID, like this:

```
Export process job successfully created, ID: yfrr9ku5i7fjubeshs1ksrknzc
```

While the job is running, its status can be checked using the ID that was provided when it was created, and when it's done the output will look similar to this:

```
mmctl export job show yfrr9ku5i7fjubeshs1ksrknzc ID:
yfrr9ku5i7fjubeshs1ksrknzc Status: success Created: 2021-11-03 10:44:13
-0500 CDT Started: 2021-11-03 10:44:23 -0500 CDT
```

The completed file will be downloaded to your desktop as a `.zip` file.

> Note
>
> Do not rename the file as the file name is referenced in log files, which are used by the Support team to validate the exported file.

The Support team will provide you with S3 credentials so you can upload the exported file. Once you've uploaded the file, please [contact our support team (https://mattermost.com/support/)](https://mattermost.com/support/) and let them know.

Create a new workspace on the Mattermost Cloud

In the meantime, you can log into Mattermost Cloud with your Mattermost credentials and create a Cloud workspace.

> Note
>
> Do not create any users in your Mattermost Cloud instance as the migration process performs this task for you.

Importing your data into your Mattermost Cloud instance

Once the export upload to the provided S3 bucket is complete and you've shared your Mattermost Cloud instance name/URL, Support can begin the import step.

Depending on the size of the export this process can take some time. Support will contact you as soon as the import is complete. During this time it is highly recommended you do not use your Mattermost Cloud instance.

Start using your Mattermost Cloud instance

When the export is complete, you can log into your Cloud instance and can invite your users to log in.

> Note
>
> We recommend that you keep your self-hosted Mattermost server until you’ve been using your Cloud instance for a while and all is verified as is as expected.

On this page

- Mattermost workspace migration
  - Migrate from self-hosted to Cloud
    - Migration process
    - Create a new workspace on the Mattermost Cloud
    - Importing your data into your Mattermost Cloud instance
    - Start using your Mattermost Cloud instance
  - Migrate from Cloud to self-hosted
    - How does the process work?
  - Authenticate
  - Create the export
  - Upload the export to the new server
  - Complete the import into the new server
  - Migrate from self-hosted to Cloud
    - Migration process
    - Create a new workspace on the Mattermost Cloud
    - Importing your data into your Mattermost Cloud instance
    - Start using your Mattermost Cloud instance

Mattermost Cloud data residency

Mattermost Cloud resides in the `aws-us-east-1` region, located in Virginia, United States. The following customer data will be stored at rest in this data center when using Mattermost Cloud:

- Channels, messages, replies, and channel membership information
- Collaborative playbooks and event runs
- User accounts and profiles
- Files (e.g., images, docs, etc.) uploaded to the Mattermost Service
- Plugin, app- or bot-generated messages and files
- Data used to measure user count, usage, and revenue
- Data used for analytics and to measure quality of service, e.g., sanitized logs
- IDs generated by Mattermost on behalf of the customer

If you require your data to reside in an area outside of the United States, talk to a Mattermost Expert (https://mattermost.com/contact-sales/), or consider deploying one of our self-hosted options that provides full control of your data. You may also work with one of our European partners (https://mattermost.com/partners) for deploying and hosting your Mattermost server.

Cloud IP Filtering

IP filtering is a powerful security feature that allows system admins to control access to their workspace by defining approved IP ranges. Only users within these specified IP ranges can access the workspace, ensuring enhanced security for your workspace. IP filtering requires a subscription to Mattermost Cloud Enterprise.

Configure IP filtering

1. **Log in as system admin**: As a system admin, access the System Console of your workspace by selecting the **Product**

   menu and selecting **System Console**.

2. **Go to Site Configuration**: Go to the **Site Configuration** section.

3. **Access IP Filtering Settings**: Under **Site Configuration**, select **IP Filtering** to access the IP Filtering settings.

About CIDR notation

CIDR (Classless Inter-Domain Routing) notation is used to specify a range of IP addresses. It consists of an IP address followed by a forward slash and a number indicating the network's prefix length. For example:

- `192.168.0.0/24` represents the IP range from `192.168.0.0` to `192.168.0.255`.
- The `/24` signifies that the first 24 bits are the network address, leaving 8 bits for host addresses.

> Tip
>
> For a more in-depth explanation of CIDR notation, refer to [this article (https://aws.amazon.com/what-is/cidr/)](https://aws.amazon.com/what-is/cidr/).

Configure IP filters

Enable/disable IP filtering

System admins have the option to enable or disable IP filtering:

- **Enable:** To activate IP filtering, ensure at least one IP range is added to the whitelist.
- **Disable:** Temporarily disable IP filtering by removing all IP ranges from the whitelist, or by flipping the global IP Filtering toggle in the System Console.

Add an IP range

To add an IP range to the whitelist, follow these steps:

1. Select the **Add Filter** button within the IP Filtering settings page in the System Console.
2. Enter the IP range using CIDR notation. For example, `192.168.1.0/24`.
3. Provide a descriptive name or label for the IP range to ease identification in the future.
4. Save the changes.

> Note
>
> The System Console will restrict you from saving changes if the IP address you are accessing your workspace on is not within the ranges you have specified at the time you save your changes.
>
>
>

Edit or remove an existing IP range

To edit or remove an existing IP range from the whitelist:

1. Locate the IP range you want to modify within the **IP Filtering** settings.
2. Hover over the rule you'd like to edit or delete, and select the respective edit or delete option beside the IP range.
3. Make necessary changes or confirm the removal of the IP range.
4. Save your changes by selecting **Save**.

Unable to access your workspace?

If you are unable to access your workspace due to previously set IP filters, and you need to regain access to your workspace, [contact our support team (https://mattermost.com/support/)](https://mattermost.com/support/).

> Note
>
> Going through this process will disable **all** existing rules applied to your workspace. This means that any IP address will now be able to access it.

Conclusion

By configuring IP filters using CIDR notation, system admins can effectively manage access to the workspace, enhancing security by allowing access only from specified IP ranges.

For any further assistance or queries, [contact our support team (https://mattermost.com/support/)](https://mattermost.com/support/).

On this page

- Cloud IP Filtering
  - Configure IP filtering
  - About CIDR notation
  - Configure IP filters
    - Enable/disable IP filtering
    - Add an IP range
    - Edit or remove an existing IP range
    - Unable to access your workspace?
  - Conclusion

Code signing custom builds

Code signing is an essential process for ensuring the authenticity and integrity of your custom `Mattermost` builds. This guide provides steps on how to code sign a build using your own certificates for Windows, Mac, and Linux.

Important

Make sure to follow each operating system's guidelines and best practices for signing applications.

Prerequisites

Windows

1. **Code Signing Certificate**: Obtain a certificate from a Certificate Authority (CA) or use a self-signed certificate if suitable.
2. **SignTool**: Available as part of the Windows SDK.

Linux

1. **GPG Key**: Create a GPG key if you don't have one.
2. **GnuPG**: Install [GnuPG (https://gnupg.org/howtos/card-howto/en/ch02.html)](https://gnupg.org/howtos/card-howto/en/ch02.html) if not already installed.

Mac

1. **Developer ID Application Certificate**: Obtain from Apple. It requires an Apple Developer account.
2. **Xcode**: Ensure [Xcode (https://developer.apple.com/xcode/)](https://developer.apple.com/xcode/) is installed.

Process

Windows

1. **Install SignTool**: Install the Windows SDK to access the `SignTool` utility.

2. **Obtain a Code Signing Certificate**: Purchase or create a certificate (`.pfx` file) via a CA.
3. **Import the Certificate**: Open the `.pfx` file and import it into the Windows Certificate Store.
4. **Sign the Executable**
   - Open the command prompt as Administrator.
   - Use `SignTool` to sign your executable:

```
signtool sign /v /s "My" /sha1 <cert hash> /fd SHA256 /tr http://
timestamp.digicert.com /td SHA256 <path-to-your-executable>
```

Linux

1. **Create or Import Your GPG Key**: If you don't have a GPG key, create one:

```
gpg --full-generate-key
```

   Alternatively, import an existing GPG key, if you have one:

```
gpg --import /path/to/your-key.asc
```

2. **Sign the Package**: Use `dpkg-sig` to sign a Debian package:

```
dpkg-sig --sign builder your-package.deb
```

   Use `rpmsign` to sign an RPM package:

```
rpmsign --addsign your-package.rpm
```

3. **Verify the Signature**: Verify the signature of a `.deb` package:

```
dpkg-sig --verify your-package.deb
```

   Verify the signature of an `.rpm` package:

```
rpm --checksig your-package.rpm
```

Mac

1. **Obtain a Code Signing Certificate**: Create a `Developer ID Application` certificate in your Apple Developer account and download it.
2. **Import the Certificate**: Double-click the certificate to import it into the Keychain.
3. **Sign the Application**: Use the `codesign` tool from Xcode to sign your application:

   ```
   codesign --deep --force --verify --verbose --sign "Developer ID
   Application: Your Name (TeamID)" /path/to/your.app
   ```

4. [Optional] **Verify the Signature**: Verify the signature to ensure everything is correctly signed:

   ```
   spctl --assess --verbose=4 /path/to/your.app codesign -dv --verbose=4 /
   path/to/your.app
   ```

Summary

- **Windows**: Use `SignTool` from the Windows SDK with your imported code signing certificate.
- **Mac**: Use `codesign` and `spctl` tools from Xcode with your Apple Developer ID certificate.
- **Linux**: Use `GnuPG` to create/sign with your GPG key, `dpkg-sig` for `.deb` packages, and `rpmsign` for `.rpm` packages.

On this page

- Code signing custom builds
- Code signing custom builds
  - Prerequisites
    - Windows
    - Linux

    - Mac
  - Process
    - Windows
    - Linux
    - Mac
  - Summary
  - Prerequisites
  - Process
  - Summary

Command line tools

In self-managed deployments, a `mattermost` command is available for configuring the system from the directory where the Mattermost server is installed. For an overview of the Mattermost command line interface (CLI), [read this article (https://medium.com/@santosjs/plugging-in-to-the-mattermost-cli-8cdcef2bd1f6)](https://medium.com/@santosjs/plugging-in-to-the-mattermost-cli-8cdcef2bd1f6) from Santos.

Important

From Mattermost v6.0, the majority of these CLI commands have been replaced with equivalents available using the mmctl command line tool. However, mattermost import commands, mattermost export commands, and related subcommands, remain available and fully supported from Mattermost v6.0.

These `mattermost` commands include the following functionality:

Compliance Export

- Export data
- Schedule an export job

Database

- Initialize the database, execute migrations, and load custom defaults
- Migrate the database schema for unapplied migrations
- Reset the database to its initial state
- Return the most recently applied version number
- Roll back database migrations

Server Operations

- Start the Mattermost job server
- Run the Mattermost server
- Display Mattermost version information

Use the CLI

Via Mattermost

To run the CLI commands, you must be in the Mattermost root directory. On a default installation of Mattermost, the root directory is `/opt/mattermost`. If you followed our standard installation process, you must run the commands as the user `mattermost`. The name of the executable is `mattermost`, and it can be found in the `/opt/mattermost/bin` directory.

For example, to get the Mattermost version on a default installation of Mattermost:

```
cd /opt/mattermost/ sudo -u mattermost bin/mattermost version
```

> Note
>
> - Ensure you run the Mattermost binary as the `mattermost` user. Running it as `root` user (for example) may cause complications with permissions as the binary initiates plugins and accesses various files when running CLI commands. Running the server as `root` may result in ownership of the plugins and files to be overwritten as well as other potential permissions errors.
> - When running CLI commands on a Mattermost installation that has the configuration stored in the database, you might need to pass the database connection string as:
>
> ```
> bin/mattermost --config="postgres://mmuser:mostest@localhost:5432/
> mattermost_test?sslmode=disable\u0026connect_timeout=10"
> ```

Via GitLab Omnibus

On GitLab Omnibus, you must be in the following directory when you run CLI commands: `/opt/gitlab/embedded/service/mattermost`. Also, you must run the commands as the user *mattermost* and specify the location of the configuration file. The executable is `/opt/gitlab/embedded/bin/mattermost`.

For example, to get the Mattermost version on GitLab Omnibus:

```
cd /opt/gitlab/embedded/service/mattermost sudo /opt/gitlab/embedded/bin/
chpst -e /opt/gitlab/etc/mattermost/env -P -U mattermost:mattermost -u
mattermost:mattermost /opt/gitlab/embedded/bin/mattermost version
```

> Note
>
> The example commands on this documentation page are for a default installation of Mattermost. You must modify the commands so that they work on GitLab Omnibus.

Via Docker Install

On Docker install, the `/mattermost/bin` directory was added to `PATH`, so you can use the CLI directly with the `docker exec` command. Note that the container name may be `mattermostdocker_app_1` if you installed Mattermost with `docker-compose.yml`.

For example, to get the Mattermost version on a Docker Install:

```
docker exec -it <your-mattermost-container-name> mattermost version
```

Via Docker Preview

The Docker Install tab details and command references below also apply to the [Mattermost docker preview image (https://github.com/mattermost/mattermost-docker-preview)](https://github.com/mattermost/mattermost-docker-preview).

> Note
>
> - The CLI is run in a single node which bypasses the mechanisms that a High Availability environment uses to perform actions across all nodes in the cluster. As a result, when running CLI commands in a High Availability environment, tasks that change configuration settings require a server restart.
> - Parameters in CLI commands are order-specific.
> - If special characters (`!`, `|`, `(`, `)`, `\`, `'`, or `"`) are used, the entire argument needs to be surrounded by single quotes, or the individual characters need to be escaped out.

mattermost CLI commands

Description

Commands for configuring and managing your Mattermost instance and users.

Options

```
-c, --config {string} Configuration file to use. (default
"config.json") --disableconfigwatch {boolean} When true, the
config.json file will not be reloaded automatically when another
process changes it (default "false")
```

Child Commands

- mattermost db - Database commands
- mattermost export - Compliance export commands
- mattermost help - Generate full documentation for the CLI
- mattermost import - Legacy import command
- mattermost jobserver - Start the Mattermost job server
- mattermost server - Run the Mattermost server

- mattermost version - Display version information

mattermost db

Description

Commands related to the database

Child Commands

- mattermost db downgrade - Roll back database migrations. Requires either an update plan to roll back to, or comma-separated version numbers to be rolled back.
- mattermost db init - Initialize the database, execute migrations, and load custom defaults
- mattermost db migrate - Migrate the database schema for unapplied migrations
- mattermost db reset - Reset the database to its initial state
- mattermost db version - Return the most recently applied version number

Options

```
-h, --help help for db -c, --config string Configuration file to use
```

mattermost db downgrade

Description

Rolls back database migrations. Requires either an update plan to roll back to, or comma-separated version numbers to be rolled back.

Format

```
mattermost db downgrade
```

Example

```
mattermost db downgrade 99,100,101
```

Options

```
--<plan-file> string Runs the rollback migrations defined in the plan
file.
```

mattermost db init

Description

Initializes the database for a given data source name (DSN), executes migrations, and loads custom defaults when specified.

Format

```
mattermost db init
```

Examples

Use the `config` flag to pass the DSN:

```
mattermost db init --config postgres://localhost/mattermost
```

Run this command to use the `MM_CONFIG` environment variable:

```
MM_CONFIG=postgres://localhost/mattermost mattermost db init
```

Run this command to set a custom defaults file to be loaded into the database:

```
MM_CUSTOM_DEFAULTS_PATH=custom.json MM_CONFIG=postgres://localhost/
mattermost mattermost db init
```

mattermost db migrate

Description

Migrates the database schema if there are any unapplied migrations.

Child Commands

- mattermost db downgrade - Roll back database migrations.

Format

```
mattermost db migrate
```

Example

```
mattermost db migrate
```

Options

```
--auto-recover bool If the migration plan receives an error during
migrations, this command will try to rollback migrations already
applied within the plan. Not recommended without reviewing migration
plan by combining --save-plan and --dry-run options. --save-plan bool
Saves the plan for the migration into the file store so that it can be
used for reviewing the plan or for downgrading. --dry-run bool Does not
apply the migrations, but it validates how the migration would run
based on the given conditions.
```

mattermost db reset

Description

Resets the database to its initial state. Doesn't start the application server. Only starts the store layer and truncates the tables, excluding the `migrations` table.

Format

```
mattermost db reset
```

Example

```
bin/mattermost db reset
```

mattermost db version

Description

Returns the most recently applied version number.

Format

```
mattermost db version
```

---

mattermost export

Note

Description

Commands for exporting data for compliance and for merging multiple Mattermost instances.

Child Commands

- mattermost export actiance - Deprecated from Mattermost v10.5 in favor of mattermost export schedule.
- mattermost export csv - Deprecated from Mattermost v10.5.
- mattermost export global-relay-zip - Deprecated from Mattermost v10.5.
- mattermost export schedule - Schedule a compliance export job.
- mattermost export bulk - Export data to a file compatible with the Mattermost Bulk Import format. Deprecated in favor of mmctl export commands.

mattermost export actiance

From Mattermost v10.5, this command has been deprecated. It will be added to the mmctl command line tool in a future version. Until then, please use mattermost export schedule.

mattermost export csv

From Mattermost v10.5, this command has been deprecated. It will be added to the mmctl command line tool in a future version.

mattermost export global-relay-zip

From Mattermost v10.5, this command has been deprecated. It will be added to the mmctl command line tool in a future version.

mattermost export schedule

Description

> Schedule an export job in a format suitable for importing into a third-party archive system.

Format

```
mattermost export schedule
```

Example

```
bin/mattermost export schedule --exportFrom=1513102632
```

Options

```
--exportFrom string Unix timestamp (seconds since epoch, UTC) to export
data from. --timeoutSeconds string Set how long the export should run
for before timing out.
```

mattermost export bulk

From Mattermost v6.0, this command has been deprecated in favor of mmctl export commands as the supported way to export data out of Mattermost.

---

mattermost help

Description

Generate full documentation in Markdown format for the Mattermost command line tools.

Format

```
mattermost help {outputdir}
```

---

mattermost import

Description

Import data into Mattermost.

Child Command

- mattermost import bulk - Import a Mattermost Bulk Import File. Deprecated in favor of mmctl import commands.
- mattermost import slack - Import a team from Slack.

mattermost import bulk

From Mattermost v6.0, this command has been deprecated in favor of mmctl import commands as the supported way to import data into Mattermost.

mattermost import slack

See the mmctl import commands documentation as the preferred way to import Slack data into Mattermost.

Description

Import a team from a Slack export zip file.

Format

. code-block:: sh

mattermost import slack {team} {file}

Example

```
bin/mattermost import slack myteam slack_export.zip
```

---

mattermost jobserver

Description

Start the Mattermost job server.

Format

```
mattermost jobserver
```

Example

```
bin/mattermost jobserver
```

---

mattermost server

Description

Runs the Mattermost server.

Format

```
mattermost server
```

---

mattermost version

> Note
>
> From Mattermost v6.5, this CLI command no longer interacts with the database. The mattermost db migrate CLI command has been introduced to trigger schema migrations.

Desription

Displays Mattermost version information.

Format

```
mattermost version
```

Troubleshooting

Executing a command hangs and doesn't complete

If you have Bleve search indexing enabled, temporarily disable it in **System Console > Experimental > Bleve** and run the command again. You can also optionally use the new mmctl Command Line Tool.

Bleve does not support multiple processes opening and manipulating the same index. Therefore, if the Mattermost server is running, an attempt to run the CLI will lock when trying to open the indeces.

If you aren't using the Bleve search indexing, feel free to post in our [Troubleshooting forum (https://forum.mattermost.com/c/trouble-shoot/16)](https://forum.mattermost.com/c/trouble-shoot/16) to get help.

On this page

- Command line tools
- Command line tools

- Use the CLI
  - Via Mattermost
  - Via GitLab Omnibus
  - Via Docker Install
  - Via Docker Preview
- mattermost CLI commands
- mattermost db
  - mattermost db downgrade
  - mattermost db init
  - mattermost db migrate
  - mattermost db reset
  - mattermost db version
- mattermost export
  - mattermost export actiance
  - mattermost export csv
  - mattermost export global-relay-zip
  - mattermost export schedule
  - mattermost export bulk
- mattermost help
- mattermost import
  - mattermost import bulk
  - mattermost import slack
- mattermost jobserver
- mattermost server
- mattermost version
- Troubleshooting
  - Executing a command hangs and doesn’t complete
- Use the CLI
- mattermost CLI commands
- mattermost db
  - mattermost db downgrade
  - mattermost db init
  - mattermost db migrate

    - mattermost db reset
    - mattermost db version
- mattermost export
    - mattermost export actiance
    - mattermost export csv
    - mattermost export global-relay-zip
    - mattermost export schedule
    - mattermost export bulk
- mattermost help
- mattermost import
    - mattermost import bulk
    - mattermost import slack
- mattermost jobserver
- mattermost server
- mattermost version
- Troubleshooting
    - Executing a command hangs and doesn’t complete

Configure server health check probes

This page describes how to configure health check probes for a Mattermost server.

Before you begin, you should have a running Mattermost server. If you don't, you can install Mattermost on various distributions.

> Note
>
> Highly available Mattermost cluster support requires Mattermost Enterprise.

You can perform a health check with the following 2 methods:

- ping the APIv4 endpoint
- ping the Mattermost server

Ping the APIv4 endpoint

You can use the [GET /system/ping APIv4 endpoint (https://api.mattermost.com/#tag/system%2Fpaths%2F~1system~1ping%2Fget)](https://api.mattermost.com/#tag/system%2Fpaths%2F~1system~1ping%2Fget) to check for system health.

A sample request is included below. The endpoint checks if the server is up and healthy based on the configuration setting `GoRoutineHealthThreshold`.

- If `GoRoutineHealthThreshold` and the number of goroutines on the server exceeds that threshold, the server is considered unhealthy.
- If `GoRoutineHealthThreshold` is not set or the number of goroutines is below the threshold the server is considered healthy.

This endpoint can also be provided to schedulers like [Kubernetes (https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/)](https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/).

```
import github.com/mattermost/mattermost/tree/master/server/public/model"
Client := model.NewAPIv4Client("https://your-mattermost-url.com")
Client.Login("email@domain.com", "Password1") // GetPing status, err :=
Client.GetPing()
```

Ping the Mattermost server

The [Mattermost Probe (https://github.com/csduarte/mattermost-probe)](https://github.com/csduarte/mattermost-probe) constantly pings a Mattermost server using a variety of probes.

These probes can be configured to verify core features, including sending and receiving messages, joining channels, pinging a login page, and searching of users and channels.

The project is contributed by the Mattermost open source community. Suggestions and contributions for the project are welcome.

On this page

- Configure server health check probes
  - Ping the APIv4 endpoint
  - Ping the Mattermost server

Mattermost feature labels

Mattermost's feature labels serve as indicators of the status, maturity, and support level of each feature, helping users and system administrators navigate product feature adoption with clarity and confidence. Feature labels communicate the stage of development, level of readiness, and potential risks associated with a feature for customers seeking to adopt new value.

Experimental

Feature is an early Proof of Concept (POC) with an unstable codebase, minimal QA covering a small or specific set of use cases, and potential UI issues. Security reviews are incomplete, making it unsuitable for production. Distribution of the solution is limited, and our Cloud environments are typically not eligible to use experimental features. Data loss can occur as data schemas and configurations may change, and minimal documentation is available. Caution is advised as the solution may be discarded if its value is not proven.

Beta

Feature is in active development towards General Availability. Not fully complete, but reviewed by our security team, adoption is suitable for a small set of customers behind a feature flag. Identified bugs are fixed on a best effort basis, and no major breaking changes are anticipated, full testing is in progress, and detailed documentation may not be available. Beta is a transitional stage, meaning the solution is maturing but requires careful consideration in a full production deployment as scale and client availability may vary. Premier Support (https://mattermost.com/support/) is recommended when using beta features in production environments.

General Availability

Feature has undergone thorough validation and testing and has production-level quality. It is feature-complete, meets quality standards, has successfully passed security reviews, and has detailed product documentation available.

General Availability features are suitable for widespread production deployment and adoption, and are eligible for commercial support, as they offer stability and reliability, with no expected changes that could disrupt functionality or scalability.

Deprecated

Feature is officially marked for removal from the product. It is no longer supported or actively maintained by the development team. If the feature is still in use in your deployed version, we recommend users discontinue its use and migrate to alternative functionalities.

On this page

- Mattermost feature labels
  - Experimental
  - Beta
  - General Availability
  - Deprecated

In-product notices

Mattermost in-product notices keep users and administrators informed of the newest product improvements, features, and releases.

Administrator notices

Administrator notices inform system admins when a new server version is available, when external dependencies are being deprecated, or when server upgrades are recommended due to ending support life cycles. System admins may also receive notices about recommended server configuration options to optimize the user experience of their deployment.

System admins can disable end user notices by going to **System Console > Site Configuration > Notices**.

End user notices

End user notices are used to inform all users when:

- New desktop app versions and feature enhancements are available for users.

- Mattermost is gathering user feedback to improve the product and user experience.

System admins can disable end user notices by going to **System Console > Site Configuration > Notices**.

Frequently asked questions (FAQs)

Are notices enabled by default?

Yes. Notices are enabled by default for all Mattermost users. System admins may choose to disable admin or end user notices in **System Console > Site Configuration > Notices**.

Will I still receive notices if my server is air-gapped?

No, the Mattermost server requires a connection to the internet to receive notices.

How often will users receive notices?

Notices will be used to raise awareness of new features as part of our monthly release cadence. Users will only receive notices that specifically apply to them.

For example, if a user is already running the latest Desktop App version, they will not receive an upgrade notice.

On this page

- In-product notices
  - Administrator notices
  - End user notices
  - Frequently asked questions (FAQs)
    - Are notices enabled by default?
    - Will I still receive notices if my server is air-gapped?
    - How often will users receive notices?

Mattermost logging

Mattermost provides 3 independent logging systems that can be configured separately with separate log files and rotation policies to meet different operational and compliance needs:

Logging Type	Captures	Production Recommendation	Configuration Priority
Log Settings	All general Mattermost server operations, errors, startup/initialization, API calls, and system events.	Always enabled. Console: INFO, File: INFO	**High** - Essential for operations
Audit Log Settings	Security and compliance events, user actions, API access, authentication events, and administrative changes.	Enable if compliance is required	**Medium** - Based on regulatory needs
Notification Log Settings	Notification subsystem events, push notifications, email delivery, mobile notification processing.	Enable for notification troubleshooting	**Low** - Enable when debugging issues

By default, all Mattermost plans write logs to both the console and to the `mattermost.log` file in a machine-readable JSON format for log aggregation tools. Mattermost Enterprise and Professional customers can additionally log directly to syslog and TCP socket destination targets. Audit and notification logging are designed to be asynchronous to minimize performance impact.

System admins can customize the following logging options based on your business practices and needs by going to **System Console > Environment > Logging** or by editing the `config.json` file directly.

Console logs

Console logs feature verbose debug level log messages for general and notification activities that are written to the console using the standard output stream (stdout). You can customize console logs for general and notification activities. See the Logging configuration settings for details.

File logs

File logs feature info level log messages for general and notification activities, including errors and information around startup, and initialization and webhook debug messages. The file is stored in `./logs/mattermost.log`, rotated at 100 MB, and archived to a separate file in the same directory. You can customize file logs for general and notification activities. See the Logging configuration settings for details.

> Tip
>
> You can download the `mattermost.log` file locally by going to **System Console > Reporting > Server Logs**, and selecting **Download Logs**.

You can optionally output log records to any combination of console, local file, syslog, and TCP socket targets, each featuring additional customization. See Advanced Logging for details.

Define logging output

Define logging output for general and notification activities in JSON format in the System Console by going to **Environment > Logging > Advanced Logging** or by editing the `config.json` file directly. You can use the sample JSON below as a starting point.

```
{  "console1": {  "type": "console",  "format": "json",  "levels": [
{"id": 5, "name": "debug", "stacktrace": false},  {"id": 4, "name": "info",
"stacktrace": false, "color": 36},  {"id": 3, "name": "warn", "stacktrace":
false},  {"id": 2, "name": "error", "stacktrace": true, "color": 31},
{"id": 1, "name": "fatal", "stacktrace": true, "color": 31},  {"id": 0,
"name": "panic", "stacktrace": true, "color": 31},  {"id": 10, "name":
"stdlog", "stacktrace": false}  ],  "options": {  "out": "stdout"  },
"maxqueuesize": 1000  },  "file1": {  "type": "file",  "format": "json",
"levels": [  {"id": 5, "name": "debug", "stacktrace": false},  {"id": 4,
"name": "info", "stacktrace": false},  {"id": 3, "name": "warn",
"stacktrace": false},  {"id": 2, "name": "error", "stacktrace": true},
{"id": 1, "name": "fatal", "stacktrace": true},  {"id": 0, "name": "panic",
"stacktrace": true}  ],  "options": {  "filename":
"mattermost_logging.log",  "max_size": 100,  "max_age": 1,  "max_backups":
10,  "compress": true  },  "maxqueuesize": 1000  },  "file2": {  "type":
"file",  "format": "json",  "levels": [  {"id": 2, "name": "error",
"stacktrace": true},  {"id": 1, "name": "fatal", "stacktrace": true},
{"id": 0, "name": "panic", "stacktrace": true}  ],  "options": {
"filename": "mattermost_logging_errors.log",  "max_size": 100,  "max_age":
30,  "max_backups": 10,  "compress": true  },  "maxqueuesize": 1000  } }
```

Audit logging

Note

By default, Mattermost doesn’t write audit logs locally to a file on the server. You can enable and customize experimental audit logging in Mattermost to record activities and events performed within a Mattermost workspace, such as access to the Mattermost REST API or mmctl. The ability to enable audit logging in Mattermost is currently in Beta.

- Audit logs are recorded asynchronously to reduce latency to the caller, and are stored separately from general logging.
- During short spans of inability to write to targets, the audit records buffer in memory with a configurable maximum record cap. Based on typical audit record volumes, it could take many minutes to fill the buffer. After that, the records are dropped, and the record drop event is logged.
- You can customize console logs for general and notification activities. See the Logging configuration settings for details.

- In addition, you can output audit log records to any combination of console, local file, syslog, and TCP socket targets, each featuring additional customization. See Advanced Logging for details.

> Note
>
> - From Mattermost v7.2, experimental audit logging is a breaking change from previous releases in cases where customers looking to parse previous audit logs with the new format.
> - The format and content of an audit log record has changed to become standardized for all events using a standard JSON schema.
> - Existing tools which ingest or parse audit log records may need to be modified.
> - From Mattermost v9.3, you can enable and customize advanced logging for AD/LDAP events separately from other logging.

Advanced logging

Note

System admins can output log and audit records general, audit, and notification activities to any combination of console, local file, syslog, and TCP socket targets. Each output target features additional configuration options you can customize for your Mattermost deployment.

> Tip
>
> - From Mattermost v9.11, system admins can configure advanced logging JSON options using the `mmctl config set` command. See the mmctl config set documentation for an example slash command.
> - From Mattermost v9.3, system admins can configure advanced logging options in the System Console using multi-line JSON by going to **Environment > Logging**.
> - Alternatively, admins can configure advanced logging within the `AdvancedLoggingJSON` section of the `config.json` file using multi-line JSON or escaped JSON as a string.
> - Mattermost Team Edition customers can output audit log records to the console or a file.

Advanced logging options can be configured to:

- Enable trace logging for AD/LDAP to troubleshoot authentication issues.
- Capture errors and panics in a separate, monitored file that triggers alerts.
- Capture production debug and error logs in a separate file with log file rotation to reproduce issues, while enforcing a cap on the amount of disk space the debug logs are allowed to use.
- Audit every API endpoint accessed during a user workflow.

Define advanced log output

Multi-line JSON

In the example below, file output is written to `./logs/audit.log` in plain text and includes all audit log levels & events. Older logs are kept for 1 day, and up to a total of 10 backup log files are kept at a time. Logs are rotated using gzip when the maximum size of the log file reaches 500 MB. A maximum of 1000 audit records can be queued/buffered while writing to the file.

```
"AdvancedLoggingJSON": {  "file_1": {  "type": "file",  "format": "plain",
"levels": [  { "id": 100, "name": "audit-api" },  { "id": 101, "name":
"audit-content" },  { "id": 102, "name": "audit-permissions" },  { "id":
103, "name": "audit-cli" }  ],  "options": {  "compress": true,
"filename": "./logs/audit.log",  "max_age": 1,  "max_backups": 10,
"max_size": 500  },  "maxqueuesize": 1000  } }
```

Filespec

Advanced logging configuration can be pointed to a filespec to another configuration file, rather than multi-line JSON, to keep the config.json file tidy:

```
"AdvancedLoggingJSON": "/path/to/audit_log_config.json"
```

The separate configuration file includes the multi-line JSON instead.

In the example below, the first output is written to the console in plain text and includes all audit log levels, events, and command outputs. A pipe `|` delimiter is placed between fields.

A second output is written to `./logs/audit.log` in plain text in a machine-readable JSON format and includes all audit log levels, events, and command outputs. Older logs are kept for 1 day, and up to a total of 10 backup log files are kept at a time. Logs are rotated using GZIP when the maximum size of the log file reaches 500 MB. A maximum of 1000 audit records can be queued/buffered while writing to the file.

Contents of `audit_log_config.json` file:

```
{  "sample-console": {  "type": "console",  "format": "plain",
"format_options": {  "delim": " | "  },  "levels": [  {"id": 100, "name":
"audit-api"},  {"id": 101, "name": "audit-content"},  {"id": 102, "name":
"audit-permissions"},  {"id": 103, "name": "audit-cli"}  ],  "options": {
"out": "stdout"  },  "maxqueuesize": 1000  },  "sample-file": {  "type":
"file",  "format": "json",  "levels": [  {"id": 100, "name": "audit-api"},
{"id": 101, "name": "audit-content"},  {"id": 102, "name": "audit-
permissions"},  {"id": 103, "name": "audit-cli"}  ],  "options": {
"filename": "./logs/audit.log",  "max_size": 500,  "max_age": 1,
"max_backups": 10,  "compress": true  },  "maxqueuesize": 1000  } }
```

Specify destination targets

Log records can be sent to any combination of console, local file, syslog, and TCP socket targets. Log targets have been chosen based on support for the vast majority of log aggregators and other log analysis tools, without needing additional software installed.

System admins can define multiple log targets to:

- Mirror log output to files and log aggregators for redundancy.
- Log certain entries to specific destinations. For example, all `audit-content` records can be routed to a different destination than the other levels.
- Admins can also temporarily disable log targets by setting its `type` to `none`.

Configure format preferences

System admins can control log formatting per target.

Plain log format configuration options

Key	Type	Description
disable_timestamp	bool	Disables output of the timestamp. Default is `false`.
disable_level	bool	Disables output of the level name. Default is `false`.
disable_msg	bool	Disables output of the message text. Default is `false`.
disable_fields	bool	Disables output of all fields. Default is `false`.
disables_stacktrace	bool	Disables output of stack traces. Default is `false`.
enable_caller	string	Enables output of the file and line number that emitted a log record. Default is `false`.
delim	string	Delimiter placed between fields. Default is single space.
min_level_len	number	Minimum level name length. When level names are less than the minimum, level names are padded with spaces. Default is `0`.
min_msg_len	number	Minimum message length. When message text is less than the minimum, message text is padded with spaces. Default is `0`.
timestamp_format	string	Format for timestamps. Default is [RFC3339 (https://www.rfc-editor.org/rfc/rfc3339)](https://www.rfc-editor.org/rfc/rfc3339).

line_end	string	Alternative end of line character(s). Default is `n`.
enable_color	bool	Enables color for targets that support color output. Default is `false`.

JSON log format configuration options

Key	Type	Description
disable_timestamp	bool	Disables output of the timestamp. Default is `false`.
disable_level	bool	Disables output of the log level display name. Default is `false`.
disable_msg	bool	Disables output of the message text. Default is `false`.
disable_fields	bool	Disables output of all fields. Default is `false`.
disables_stacktrace	bool	Disables output of stack traces. Default is `false`.
enable_caller	string	Enables output of the file and line number that emitted a log record. Default is `false`.
timestamp_format	string	Format for timestamps. Default is [RFC3339 (https://www.rfc-editor.org/rfc/rfc3339)](https://www.rfc-editor.org/rfc/rfc3339).

GELF log format configuration options

Key	Type	Description
hostname	string	Outputs a custom hostname in log records. If omitted, hostname is taken from the operating system.
enable_caller	string	Enables output of the file and line number that emitted a log record. Default is `false`.

Configure log levels and events

Key	Type	Description
id	number	Unique identifier of the log level.
name	string	Name of the log level.
stacktrace	bool	Outputs a stack trace. Default is `false`.
color	number	The ANSI color code used to output parts of the log record. Supported values include: • Black: `30` • Red: `31` • Green: `32` • Yellow: `33` • Blue: `34` • Magenta: `35` • Cyan: `36` • White: `37`

Log levels

The following log levels support audit logs:

ID	Name	Description
100	`audit-api`	API events
101	`audit-content`	Content changes. This log level can generate considerably more records than the other audit log levels.
102	`audit-permissions`	Permission changes
103	`audit-cli`	CLI operations

The following log levels support application logs:

ID	Name	Description
140	`LDAPError`	AD/LDAP authentication errors
141	`LDAPWarn`	AD/LDAP authentication warnings
142	`LDAPInfo`	AD/LDAP authentication information logs
143	`LDAPDebug`	AD/LDAP authentication debug logs
144	`LDAPTrace`	AD/LDAP authentication trace logs. Replaces `LdapSetings.trace` from Mattermost v9.3.

Configure target-specific settings

Console target configuration options

Console targets can be either `stdout` or `stderr`.

- The standard output stream, `stout`, is typically used for command output that prints the results of a command to the user.
- The standard error stream, `sterr`, is typically used to print any errors that occur when a program is running.

File target configuration options

File targets support rotation and compression triggered by size and/or duration.

Key	Type	Description
filename	string	Full path to the output file.
max_size	number	Maximum size, in megabytes (MB), the log file can grow before it gets rotated. Default is `100` MB.
max_age	number	Maximum number of days to retain old log files based on the timestamp encoded in the filename. Default is `0` which disables the removal of old log files.
max_backups	number	Maximum number of old log files to retain. Default is `0` which retains all old log files. **Note**: Configuring `max_age` can result in old log files being deleted regardless of this configuration value.
compress	bool	Compress rotated log files using [gzip (https://www.gnu.org/software/gzip/)](https://www.gnu.org/software/gzip/). Default is `false`.

Syslog target configuration options

Note

Syslog targets support local and remote syslog servers, with or without TLS transport. Syslog target support requires Mattermost Enterprise.

Key	Type	Description
host	string	IP or domain name of the server receiving the log records.
port	number	Port number for the server receiving the log records.
tls	bool	Create a TLS connection to the server receiving the log records. Default is `false`.
cert	string	Path to a cert file (.pem) to be used when establishing a TLS connection to the server.
insecure	bool	Mattermost accepts any certificate presented by the server, and any host name in that certificate. Default is `false`. **Note**: Should only be used in testing environments, and shouldn't be used in production environments.
tag	string	Syslog tag field.

TCP target configuration options

Note

The TCP socket targets can be configured with an IP address or domain name, port, and optional TLS certificate. TCP socket target support requires Mattermost Enterprise. You can download a sample JSON file of the configuration to use as a starting point.

Key	Type	Description
host	string	IP or domain name of the server receiving the log records.
port	number	Port number for the server receiving the log records.
tls	bool	Create a TLS connection to the server receiving the log records. Default is `false`.
cert	string	Path to a cert file (.pem) to be used when establishing a TLS connection to the server.
insecure	bool	Mattermost accepts any certificate presented by the server, and any host name in that certificate. Default is `false`. **Note**: Should only be used in testing environments, and shouldn't be used in production environments.

Frequently asked questions

How do I enable debug logging?

As a Mattermost system admin, go to **System Console > Environment > Logging > File Log Level**, and set it to **DEBUG**. Then save your changes.

Debug logging can cause log files to expand substantially, and may adversely impact server performance. Keep an eye on your server logs, or only enable it temporarily, or in development environments, but not production enviornments.

Does Mattermost have an audit log besides the system `auditd`?

Yes. See the audit logging documentation for details.

When syslog is configured as the target, does it contain the IP address of the emitter of the data (i.e., the Mattermost app node)?

Yes. That is a function of the syslog daemon (receiver). Typically, all the log lines are prefixed with a timestamp and the hostname of the node sending the data. For example, a log line starts with: `Nov 28 10:56:59 tower kernel: [1072437.431123] ....`, where `tower` is the hostname of the server that generated the log line.

Once audit logs are enabled, can audit logging track events where an admin disables or modifies the audit log settings?

Yes, though it depends on how audit logs are configured. Audit log config can be specified via REST API, mmctl, System Console, file on disk, and using environment variables. When changes are made via the REST API or System

Console, there is an audit record. However, the Mattermost server can't capture changes to a configuration file or via environment variables.

Do server logs show any information about a service or similar stopping/updating when auditing would be disabled?

Yes. When updating the audit log configuration via REST API, mmctl, or System Console, the last event of the audit log should be about the admin user updating the config of the server, which helps your security team identify which actions took place in the system and by whom.

How do I omit incoming webhook details from the logs?

See enable-webhook-debugging

How do I adjust the maximum log field size?

See maximum-field-size

How can I configure Advanced logging via environment variables?

The `MM_LOGSETTINGS_ADVANCEDLOGGINGJSON` environment variable is used to configure Advanced logging . You can use `jq` to generate the JSON payload, e.g.

```
export MM_LOGSETTINGS_ADVANCEDLOGGINGJSON=$(jq -n -c '{  "console1": {
"Type": "console",  "Format": "json",  "Levels": [  {"ID": 5, "Name":
"debug", "Stacktrace": false},  {"ID": 4, "Name": "info", "Stacktrace":
false, "color": 36},  {"ID": 3, "Name": "warn", "Stacktrace": false},
{"ID": 2, "Name": "error", "Stacktrace": true, "color": 31},  {"ID": 1,
"Name": "fatal", "Stacktrace": true, "color": 31},  {"ID": 0, "Name":
"panic", "Stacktrace": true, "color": 31},  {"ID": 10, "Name": "stdlog",
"Stacktrace": false}  ],  "Options": {  "Out":
"stdout"  },  "MaxQueueSize": 1000  } }')
```

How can I turn on trace logging for LDAP?

Please use the following JSON configuration as as starting point to enable trace logging for LDAP:

```
{ "ldap-file": { "type": "file", "format": "plain", "levels": [ { "id": 144, "name": "LDAPTrace" }, { "id": 143, "name": "LDAPDebug" }, { "id": 142, "name": "LDAPInfo" }, { "id": 141, "name": "LDAPWarn", "stacktrace": true }, { "id": 140, "name": "LDAPError", "stacktrace": true } ], "options": { "filename": "./logs/ldap.log", "max_size": 100, "max_age": 14, "max_backups": 3, "compress": false }, "maxqueuesize": 1000 } }
```

On this page

- Mattermost logging
- Mattermost logging
  - Console logs
  - File logs
  - Define logging output
  - Audit logging
  - Advanced logging
    - Define advanced log output
      - Multi-line JSON
      - Filespec
    - Specify destination targets
    - Configure format preferences
      - Plain log format configuration options
      - JSON log format configuration options
      - GELF log format configuration options
    - Configure log levels and events
      - Log levels
    - Configure target-specific settings
      - Console target configuration options
      - File target configuration options
      - Syslog target configuration options
      - TCP target configuration options

- Frequently asked questions
  - How do I enable debug logging?
  - Does Mattermost have an audit log besides the system auditd?
  - When syslog is configured as the target, does it contain the IP address of the emitter of the data (i.e., the Mattermost app node)?
  - Once audit logs are enabled, can audit logging track events where an admin disables or modifies the audit log settings?
  - Do server logs show any information about a service or similar stopping/updating when auditing would be disabled?
  - How do I omit incoming webhook details from the logs?
  - How do I adjust the maximum log field size?
  - How can I configure Advanced logging via environment variables?
  - How can I turn on trace logging for LDAP?
- Console logs
- File logs
- Define logging output
- Audit logging
- Advanced logging
  - Define advanced log output
  - Specify destination targets
  - Configure format preferences
    - Plain log format configuration options
    - JSON log format configuration options
    - GELF log format configuration options
  - Configure log levels and events
    - Log levels
  - Configure target-specific settings
    - Console target configuration options
    - File target configuration options
    - Syslog target configuration options
    - TCP target configuration options
- Frequently asked questions
  - How do I enable debug logging?
  - Does Mattermost have an audit log besides the system auditd?

- When syslog is configured as the target, does it contain the IP address of the emitter of the data (i.e., the Mattermost app node)?
- Once audit logs are enabled, can audit logging track events where an admin disables or modifies the audit log settings?
- Do server logs show any information about a service or similar stopping/updating when auditing would be disabled?
- How do I omit incoming webhook details from the logs?
- How do I adjust the maximum log field size?
- How can I configure Advanced logging via environment variables?
- How can I turn on trace logging for LDAP?

mmctl command line tool

The mmctl is a CLI tool for the Mattermost server which is installed locally and uses the Mattermost API, but may also be used remotely. Authentication is done with either login credentials or an authentication token. This mmctl tool is included and replaces the CLI. The mmctl can currently be used alongside the Mattermost CLI tool. The Mattermost CLI tool will be deprecated in a future release.

Being installed locally enables system admins for both self-hosted and Cloud Mattermost instances to run CLI commands even in instances where there's no access to the server (e.g., via SSH).

This feature was developed to a large extent by community contributions and we'd like to extend our gratitude to the contributors who have worked on this project. We are currently accepting pull requests for Help Wanted issues in the [mattermost-server (https://github.com/mattermost/mattermost/issues?q=is%3Aissue+is%3Aopen+label%3A%22Help+Wanted%22+label%3AArea%2Fmmctl)](https://github.com/mattermost/mattermost/issues?q=is%3Aissue+is%3Aopen+label%3A%22Help+Wanted%22+label%3AArea%2Fmmctl) repo. You can learn more about the unit test coverage campaign for mmctl in the [Unit testing mmctl commands (https://mattermost.com/blog/unit-testing-mmctl-commands/)](https://mattermost.com/blog/unit-testing-mmctl-commands/) blog post.

mmctl usage notes

- System admins have two ways to run `mmctl` commands: by downloading `mmctl` from the release URLs, which you can find in the installation instructions, or by building it directly, for which you can check the build instructions below. The source code lives in the [server/cmd/mmctl directory within the mattermost repository (https://github.com/mattermost/mattermost/tree/master/server/cmd/mmctl)](https://github.com/mattermost/mattermost/tree/master/server/cmd/mmctl).
- `mmctl` also comes bundled with the Mattermost distribution, and is located in the `bin` folder of the installation, next to the `CLI`.
  - We recommend you add the path to the Mattermost `bin` folder into your `$PATH` environment variable. This ensures that you can run mmctl commands locally regardless of your current directory location.

  - If the `bin` directory is not added to the `$PATH` environment variable, each time you use mmctl you must be in the `bin` directory to run mmctl commands, and the commands must be prefixed with `./`. If you're working from a different directory, make sure you specify the full path to mmctl when running mmctl commands.

- Parameters in mmctl commands are order-specific.
- You can use the `--local` flag with mmctl commands to run them without authentication by allowing communicating with the server through a Unix socket. See the local mode documentation for activation and usage details.
- If special characters (`!`, `|`, `(`, `)`, `\`, `'`, and `"`) are used, the entire argument needs to be surrounded by single quotes (e.g. `-password 'mypassword!'`, or the individual characters need to be escaped out (e.g. `password mypassword\!`).
- Team name and channel name refer to the handles, not the display names. So in the URL `https://community.mattermost.com/core/channels/town-square` team name would be `core` and channel name would be `town-square`.

mmctl commands

- mmctl auth - Authentication Management
- mmctl bot - Bot Management
- mmctl channel - Channel Management
- mmctl command - Command Management
- mmctl completion - Generate autocompletion scripts for bash and zsh
- mmctl config - Configuration Management
- mmctl docs - Generate mmctl documentation
- mmctl export - Exports Management
- mmctl group - Group Management
- mmctl group channel - Channel Group Management
- mmctl group team - Team Group Management

- mmctl group user - Custom User Group Management
- mmctl import - Import Management
- mmctl extract - Content Extraction Job Management
- mmctl integrity - (Deprecated) Database Record Integrity
- mmctl job - Job Management
- mmctl ldap - LDAP Management
- mmctl license - License Management
- mmctl logs - Log Management
- mmctl oauth - OAuth2 application management
- mmctl permissions - Permissions Management
- mmctl plugin - Plugin Management
- mmctl post - Post Management
- mmctl roles - Roles Management
- mmctl saml - SAML Management
- mmctl sampledata - Generate sample data
- mmctl system - System Management
- mmctl team - Team Management
- mmctl team users - Team User Management
- mmctl token - Token Management
- mmctl user - User Management
- mmctl user preference - Manage User Preferences
- mmctl version - Version Management
- mmctl webhook - Webhook Management
- mmctl websocket - Websocket Management

Options

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output -h, --help help for
mmctl --insecure-sha1-intermediate allows to use insecure TLS protocols,
such as SHA-1 --insecure-tls-version allows to use TLS versions 1.0 and 1.1
--json the output format will be in json format --local allows
communicating with the server through a unix socket --quiet prevent mmctl
to generate output for the commands --strict will only run commands if the
mmctl version matches the server one --suppress-warnings disables printing
warning messages
```

Install mmctl

The mmctl tool comes bundled with Mattermost package. For customers that want to setup it independently from the package, the following methods are available to install mmctl.

Release package

If you’re using a release page for Linux, macOS, or Windows, we recommend downloading the mmctl builds through their release URL: `https://releases.mattermost.com/mmctl/${MATTERMOST_VERSION}/${PLATFORM}_${ARCHITECTURE}.tar` (for windows, substitute the `.tar` suffix with `.zip`)

E.g. to download version `v10.7.0` of the mmctl amd64 build for linux, you can run the following:

```
curl -vfsSL -O https://releases.mattermost.com/mmctl/v10.7.0/
linux_amd64.tar
```

Supported platforms, and corresponding supported architectures include linux (amd64 and arm64), darwin (amd64 and arm64), windows (amd64 only).

Local git clone

You can build mmctl from a local Git clone to work directly with the source code, use the latest features, and experiment with custom changes. This workflow is ideal if you need a tailored version of mmctl, want to fix bugs, or test new updates before they’re officially released.

```
git clone https://github.com/mattermost/mattermost.git cd mattermost/server
go install ./cmd/mmctl
```

Homebrew

On Linux or macOS, this isn't an officially supported method This installation channel is managed by the community. Please refer to the [homebrew/homebrew-core repo (https://github.com/Homebrew/homebrew-core)](https://github.com/Homebrew/homebrew-core) for reporting issues.

Use this option on Linux and macOS if you have Homebrew installed.

```
brew install mmctl
```

Build mmctl

The `mmctl` tool uses `go` modules to manage dependencies, so you need to have installed `go` 1.19 or greater on your machine.

After checking out the [mattermost repository (https://github.com/mattermost/mattermost)](https://github.com/mattermost/mattermost) locally to your machine, from the root directory of the project, you can compile the mmctl binary by running:

```
make -C server mmctl-build
```

Local mode

Note

Available only for self-hosted deployments

Local mode allows platform administrators with access to the Mattermost server to run mmctl commands against the API without needing to have a user registered. To ensure secure usage of this API, the server exposes a local socket that only a user with access to the server's file system can access. The

requests coming from the socket are treated as authorized, so they can reach the handlers without requiring a user session.

The API that the socket exposes follows the same specification that can be found in the API documentation, so mmctl is able to interact with it without needing any modifications. When a request comes in through the socket, it is flagged as local by the server, and this flag is taken into account when checking for session permissions to correctly authorize the sessions.

Activating local mode

To use local mode, the Mattermost server first needs to have local mode enabled. When local mode is enabled, a socket is created at `/var/tmp/mattermost_local.socket` by default.

> Tip
>
> When trying to use local mode with mmctl, ensure you're using the same user when running the server and mmctl, or clean up the socket file before switching to a new user. If you encounter an error like `socket file "/var/tmp/mattermost_local.socket" doesn't exists, please check the server configuration for local mode`, this can be resolved by setting this configuration setting to `true`.

Using local mode

From Mattermost v10.8, when no authentication credentials are found in the authentication configuration, mmctl automatically assumes local mode, eliminating the need to manually append the `--local` flag to the command you want to use. When valid credentials exist, Mattermost will continue to validate them as expected.

Prior to Mattermost v10.8, you must append `--local` to the command you want to use, or set the environment variable as `MMCTL_LOCAL=true`.

To use a socket file other than the default, you need to set the environment variable to `MMCTL_LOCAL_SOCKET_PATH`. This file must match the server configuration setting.

Running mmctl tests

Note

Available only for self-hosted deployments

mmctl has two types of tests: unit tests and end to end tests.

To execute them, you can run the following commands from the mattermost project root directory:

```
# For the unit tests make -C server test-mmctl-unit # For the end to end
tests make -C server test-mmctl-e2e
```

mmctl auth

Description

Manage the credentials and authentication methods of remote Mattermost instances.

> *Child Commands*
>
> - *mmctl auth clean - Clean credentials*
> - *mmctl auth current - Display current credentials*
> - *mmctl auth delete - Delete authentication details*
> - *mmctl auth list - List registered credentials*
> - *mmctl auth login - Log into Mattermost instance*
> - *mmctl auth renew - Renew login credentials*
> - *mmctl auth set - Set login credentials*

Options

```
-h, --help help for auth
```

mmctl auth clean

Description

Clean the credentials associated with a Mattermost instance.

Format

```
mmctl auth clean [flags]
```

Examples

```
mmctl auth clean
```

Options

```
-h, --help help for clean
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl auth current

Description

Show the currently stored user credentials.

Format

```
mmctl auth current [flags]
```

Examples

```
mmctl auth current
```

Options

```
-h, --help help for current
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl auth delete

Description

Delete a named credential.

Format

```
mmctl auth delete [server name] [flags]
```

Examples

```
mmctl auth delete local-server
```

Options

```
-h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl auth list

Description

Print a list of registered credentials.

Format

```
mmctl auth list [flags]
```

Examples

```
mmctl auth list
```

Options

```
-h, --help help for auth list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl auth login

Description

Log in to an instance and store credentials. You can log in with a personal access token (https://developers.mattermost.com/integrate/reference/personal-access-token) instead of using username/password. See access tokens for details.

Format

```
mmctl auth login [instance url] --name [server name] --username [username]
--password-file [password-file] [flags]
```

Examples

```
mmctl auth login https://mattermost.example.com mmctl auth login https://
mattermost.example.com --name local-server --username sysadmin --password-
file mysupersecret.txt mmctl auth login https://mattermost.example.com --
name local-server --username sysadmin --password-file mysupersecret.txt --
mfa-token 123456 mmctl auth login https://mattermost.example.com --name
local-server --access-token myaccesstoken
```

Options

```
-t, --access-token-file string Access token file to be read to use instead
of username/password -h, --help help for login -m, --mfa-token string MFA
token for the credentials -n, --name string Name for the credentials --no-
activate If present, it won't activate the credentials after login -f, --
password-file string Password file to be read for the credentials -u, --
username string Username for the credentials
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

Password

```
$ mmctl auth login https://community.mattermost.com --name community --
username my-username --password-file mysupersecret
```

The `login` command can also work interactively, so if you leave any required flag empty, `mmctl` will ask you for it interactively:

```
$ mmctl auth login https://community.mattermost.com Connection name:
community Username: my-username Password File:
```

MFA

To log in with MFA, use the `--mfa-token` flag:

```
$ mmctl auth login https://community.mattermost.com --name community --
username my-username --password-file mysupersecret --mfa-token 123456
```

Access tokens

You can generate and use a personal access token to authenticate with a server, instead of using username and password to log in:

```
$ mmctl auth login https://community.mattermost.com --name community --
access-token MY_ACCESS_TOKEN
```

Alternatively, you can log in to your Mattermost server with a username and password:

```
$ mmctl auth login https://my-instance.example.com --name my-instance --
username john.doe --password-file mysupersecret credentials for my-
instance: john.doe@https://my-instance.example.com stored
```

We can check the currently stored credentials with:

```
$ mmctl auth list | Active | Name | Username | InstanceUrl |
|--------|-------------|----------|--------------------------------| | * |
my-instance | john.doe | https://my-instance.example.com |
```

And now we can run commands normally:

```
$ mmctl user search john.doe id: qykfw3t933y38k57ubct77iu9c username:
john.doe nickname: position: first_name: John last_name: Doe email:
john.doe@example.com auth_service:
```

Installing shell completions

To install the shell completions for bash, add the following line to your `~/.bashrc` or `~/.profile` file:

```
source <(mmctl completion bash)
```

For zsh, add the following line to your `~/.zshrc` file:

```
source <(mmctl completion zsh)
```

mmctl auth renew

Description

Renew the credentials for a given server.

Format

```
mmctl auth renew [flags]
```

Examples

```
mmctl auth renew local-server
```

Options

```
-t, --access-token-file string Access token file to be read to use instead
of username/password -h, --help help for renew -m, --mfa-token string MFA
token for the credentials -f, --password-file string Password file to be
read for the credentials
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl auth set

Description

Set credentials to use in the following commands.

Format

```
mmctl auth set [server name] [flags]
```

Examples

```
mmctl auth set local-server
```

Options

```
-h, --help help for set
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

Authenticate to a server (e.g. >mmctl auth login `https://test.mattermost.com)`, then enter your username and password (and MFA token if MFA is enabled on the account).

mmctl bot

Manage bots.

> *Child Commands*
>
> - *mmctl bot assign - Assign bot ownership*
> - *mmctl bot create - Create a new bot*
> - *mmctl bot disable - Disable a bot*
> - *mmctl bot enable - Enable a bot*
> - *mmctl bot list - List all bots*
> - *mmctl bot update - Update bot configuration*

Options

```
-h, --help help for bot
```

mmctl bot assign

Description

Assign the ownership of a bot to another user.

Format

```
mmctl bot assign [bot-username] [new-owner-username] [flags]
```

Examples

```
mmctl bot assign testbot user2
```

Options

```
-h, --help help for assign
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl bot create

Description

Create a bot.

Format

```
mmctl bot create [username] [flags]
```

Examples

```
mmctl bot create testbot
```

Options

```
--description string Optional. The description text for the new bot. --
display-name string Optional. The display name for the new bot. -h, --help
help for create --with-token Optional. Auto genreate access token for the
bot.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --quiet prevent mmctl to generate output for the
commands --strict will only run commands if the mmctl version matches the
server one --suppress-warnings disables printing warning messages
```

mmctl bot disable

Description

Disable an enabled bot.

Format

```
mmctl bot disable [username] [flags]
```

Examples

```
mmctl bot disable testbot
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl bot enable

Description

Enable a disabled bot.

Format

```
mmctl bot enable [username] [flags]
```

Examples

```
mmctl bot enable testbot
```

Options

```
-h, --help help for enable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl bot list

Description

List the bot's users.

Format

```
mmctl bot list [flags]
```

Examples

```
mmctl bot list
```

Options

```
--all Optional. Show all bots (including deleleted and orphaned) -h, --help
help for list --orphaned Optional. Only show orphaned bots
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl bot update

Description

Update bot information.

Format

```
mmctl bot update [username] [flags]
```

Examples

```
mmctl bot update testbot --username newbotusername
```

Options

```
--description string Optional. The new description text for the bot --
display-name string Optional. The new display name for the bot -h, --help
help for update --username string Optional. The new username for the bot
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel

Manage channels.

> *Child Commands*
>
> - *mmctl channel archive - Archive a channel*
> - *mmctl channel create - Create a channel*
> - *mmctl channel delete - Delete a channel*
> - *mmctl channel list - List all channels on specified teams*
> - *mmctl channel make-private - (Deprecated) Set a channel's type to "private"*
> - *mmctl channel modify - Modify a channel's type (private/public)*
> - *mmctl channel move - Move channels to the specified team*

> - *mmctl channel rename - Rename a channel*
> - *mmctl channel restore - (Deprecated) Restore a channel from the archive*
> - *mmctl channel search - Search a channel by name*
> - *mmctl channel unarchive - Unarchive a channel*
> - *mmctl channel users - Manage channel users*
> - *mmctl channel users add - Add a user to a channel*
> - *mmctl channel users remove - Remove a user from a channel*

Options

```
-h, --help help for channel
```

mmctl channel archive

Description

Archive channels along with all related information including posts from the database. Channels can be specified by `[team]:[channel]` (i.e., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel archive [channels] [flags]
```

Examples

```
mmctl channel archive myteam:mychannel
```

Options

```
-h, --help help for archive
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel create

Description

Create a channel.

Format

```
mmctl channel create [flags]
```

Examples

```
mmctl channel create --team myteam --name mynewchannel --display-name "My
New Channel" mmctl channel create --team myteam --name mynewprivatechannel
--display-name "My New Private Channel" --private
```

Options

```
--display-name string Channel Display Name --header string Channel header
-h, --help help for create --name string Channel Name --private Create a
private channel --purpose string Channel purpose --team string Team name or
ID
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel delete

Description

Permanently delete channels along with all related information including posts from the database.

> Note
>
> Requires the Enable API Channel Deletion configuration setting to be set to `true`. If this configuration setting is set to `false`, attempting to delete the channel using mmctl fails.

Format

```
mmctl channel delete [channels] [flags]
```

Examples

```
mmctl channel delete myteam:mychannel
```

Options

```
--confirm Confirm you really want to delete the channel and a database
backup has been performed. -h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel list

Description

List all Public, Private, and archived channels on specified teams. Archived channels are appended with `(archived)`. Private channels the user is a member of, or has access to, are appended with `(private)`.

Format

```
mmctl channel list [teams] [flags]
```

Examples

```
mmctl channel list myteam
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel make-private

This command is deprecated in favour of mmctl channel modify and the `--private` flag.

Description

Set the type of a channel from Public to Private. Channel can be specified by `[team]:[channel]` (e.g., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel make-private [channel] [flags]
```

Examples

```
mmctl channel make-private myteam:mychannel
```

Options

```
-h, --help help for make-private
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel modify

Description

Change the Public/Private type of a channel. Channel can be specified by `[team]:[channel]` (e.g., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel modify [channel] [flags]
```

Examples

```
mmctl channel modify myteam:mychannel --private mmctl channel modify
channelId --public
```

Options

```
-h, --help help for modify --private Convert the channel to a private
channel --public Convert the channel to a public channel
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel move

Description

Move the provided channels to the specified team. Validate that all users in the channel belong to the target team. Incoming/outgoing webhooks are moved along with the channel. Channels can be specified by `[team]:[channel]` (e.g., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel move [team] [channels] [flags]
```

Examples

```
mmctl channel move newteam oldteam:mychannel
```

Options

```
-h, --help help for move --force Remove users that are not members of
target team before moving the channel.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel rename

Description

Rename an existing channel.

Format

```
mmctl channel rename [channel] [flags]
```

Examples

```
mmctl channel rename myteam:oldchannel --name 'new-channel' --display_name
'New Display Name' mmctl channel rename myteam:oldchannel --name 'new-
channel' mmctl channel rename myteam:oldchannel --display_name 'New Display
Name'
```

Options

```
--display_name string Channel Display Name -h, --help help for rename --
name string Channel Name
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel restore

This command is deprecated in favor of mmctl channel unarchive. Not used in Mattermost Server version v5.26 and later.

Description

Restore a previously deleted channel. Channels can be specified by `[team]:[channel]` (e.g., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel restore [channels] [flags]
```

Examples

```
mmctl channel restore myteam:mychannel
```

Options

```
-h, --help help for restore
```

Options inherited from parent commands

```
--format string the format of the command output [plain, json] (default
"plain") --insecure-sha1-intermediate allows the use of insecure TLS
protocols, such as SHA-1 --local allows communicating with the server
through a unix socket --strict will only run commands if the mmctl version
matches the server one
```

mmctl channel search

Description

Search for channel details by channel name. Channels can be specified by team (e.g., `--team myteam mychannel`), or by team ID. Returns channel name, display name, and channel ID.

Format

```
mmctl channel search [channel] mmctl search --team [team] [channel] [flags]
```

Examples

```
mmctl channel search mychannel mmctl channel search --team myteam mychannel
```

Options

```
-h, --help help for search --team string team name or ID
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel unarchive

Description

Unarchive a previously archived channel. Channels can be specified by `[team]:[channel]` (e.g., `myteam:mychannel`), or by channel ID.

Format

```
mmctl channel unarchive [channels] [flags]
```

Examples

```
mmctl channel unarchive myteam:mychannel
```

Options

```
-h, --help help for unarchive
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel users

Description

Manage channel users.

Options

```
-h, --help help for users
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel users add

Description

Add users to a channel.

Format

```
mmctl channel users add [channel] [users] [flags]
```

Examples

```
mmctl channel users add myteam:mychannel user@example.com username
```

Options

```
-h, --help help for add
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl channel users remove

Description

Remove users from a channel.

Format

```
mmctl channel users remove [channel] [users] [flags]
```

Examples

```
mmctl channel users remove myteam:mychannel user@example.com username mmctl
channel users remove myteam:mychannel --all-users
```

Options

```
--all-users Remove all users from the indicated channel -h, --help help for
remove
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command

Manage slash commands.

> *Child Commands*
>
> - *mmctl command archive - Archive a slash command*
> - *mmctl command create - Create a custom command*

> - *mmctl command delete - (Deprecated) Delete a specified slash command*
> - *mmctl command list - List slash commands on specified teams*
> - *mmctl command modify - Modify a slash command*
> - *mmctl command move - Move a slash command to a different team*
> - *mmctl command show - Show a custom slash command*

Options

```
-h, --help help for command
```

mmctl command archive

Description

Archive a slash command. Commands can be specified by command ID.

Format

```
mmctl command archive [commandID] [flags]
```

Examples

```
mmctl command archive commandID
```

Options

```
-h, --help help for archive
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command create

Description

Create a custom slash command for the specified team.

Format

```
mmctl command create [team] [flags]
```

Examples

```
mmctl command create myteam --title MyCommand --description "My Command
Description" --trigger-word mycommand --url http://localhost:8000/my-slash-
handler --creator myusername --response-username my-bot-username --icon
http://localhost:8000/my-slash-handler-bot-icon.png --autocomplete --post
```

Options

```
--autocomplete Show Command in autocomplete list --autocompleteDesc string
Short Command Description for autocomplete list --autocompleteHint string
Command Arguments displayed as help in autocomplete list --creator string
Command Creator's Username (required) --description string Command
Description -h, --help help for create --icon string Command Icon URL --post
Use POST method for Callback URL --response-username string Command Response
Username --title string Command Title --trigger-word string Command Trigger
Word (required) --url string Command Callback URL (required)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command delete

This command is deprecated in favor of mmctl command archive.

Description

Delete a slash command. Commands can be specified by command ID.

Format

```
mmctl command delete [flags]
```

Examples

```
mmctl command delete commandID
```

Options

```
-h, --help help for delete
```

Options inherited from parent commands

```
--format string the format of the command output [plain, json] (default
"plain") --insecure-sha1-intermediate allows the use of insecure TLS
protocols, such as SHA-1 --local allows communicating with the server
through a unix socket --strict will only run commands if the mmctl version
matches the server one
```

mmctl command list

Description

List all commands on specified teams.

Format

```
mmctl command list [teams] [flags]
```

Examples

```
mmctl command list myteam
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command modify

Description

Modify a slash command. Commands can be specified by command ID.

Format

```
mmctl command modify [commandID] [flags]
```

Examples

```
mmctl command modify commandID --title MyModifiedCommand --description "My
Modified Command Description" --trigger-word mycommand --url http://
localhost:8000/my-slash-handler --creator myusername --response-username my-
bot-username --icon http://localhost:8000/my-slash-handler-bot-icon.png --
autocomplete --post
```

Options

```
--autocomplete Show Command in autocomplete list --autocompleteDesc string
Short Command Description for autocomplete list --autocompleteHint string
Command Arguments displayed as help in autocomplete list --creator string
Command Creator's username, email or id (required) --description string
Command Description -h, --help help for modify --icon string Command Icon
URL --post Use POST method for Callback URL --response-username string
Command Response Username --title string Command Title --trigger-word string
Command Trigger Word (required) --url string Command Callback URL
(required)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command move

Description

Move a slash command to a different team. Commands can be specified by command ID.

Format

```
mmctl command move [team] [commandID] [flags]
```

Examples

```
mmctl command move newteam commandID
```

Options

```
-h, --help help for move
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl command show

Description

Show a custom slash command. Commands can be specified by command ID. Returns command ID, team ID, trigger word, display name, and creator username.

Format

```
mmctl command show [commandID] [flags]
```

Examples

```
mmctl command show commandID
```

Options

```
-h, --help help for show
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl completion

Generate autocompletion scripts for `bash` and `zsh`.

> *Child Commands*
>
> - *mmctl completion bash - Edit the configuration settings*
> - *mmctl completion zsh - Get the value of a configuration setting*

Options

```
-h, --help help for completion
```

mmctl completion bash

Description

Generate the `bash` autocompletion scripts.

To load completion, run:

```
. <(mmctl completion bash)
```

To configure your `bash` shell to load completions for each session, add the above line to your `~/.bashrc`.

Format

```
mmctl completion bash [flags]
```

Options

```
-h, --help help for bash
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl completion zsh

Description

Generate the `zsh` autocompletion scripts.

To load completion, run:

```
. <(mmctl completion zsh)
```

To configure your `zsh` shell to load completions for each session, add the above line to your `~/.zshrc`.

Format

```
mmctl completion zsh [flags]
```

Options

```
-h, --help help for zsh
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config

Configuration settings.

> *Child Commands*
>
> - *mmctl config edit - Edit the configuration settings*
> - *mmctl config export - Export the server configuration*
> - *mmctl config get - Get the value of a configuration setting*
> - *mmctl config migrate - Migrate existing configuration between backends*
> - *mmctl config patch - Patch the configuration*
> - *mmctl config reload - Reload the server configuration*
> - *mmctl config reset - Reset the configuration*
> - *mmctl config set - Set the value of a configuration*
> - *mmctl config show - Write the server configuration to STDOUT*
> - *mmctl config subpath - Update client asset loading to use the configured subpath*

Options

```
-h, --help help for config
```

mmctl config edit

Description

Open the editor defined in the EDITOR environment variable to modify the server's configuration. Once complete, save the file, then upload it to your server.

Format

```
mmctl config edit [flags]
```

Examples

```
mmctl config edit
```

Options

```
-h, --help help for edit
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config export

Description

Use this command to export the server configuration, which can then be imported into another server or environment. Masked values are only exported when the tool is running in local mode.

Format

```
mmctl config export [flags]
```

Examples

```
mmctl config export --remove-masked --remove-defaults
```

Options

```
-h, --help help for export  --remove-defaults remove default values from
the exported configuration  --remove-masked remove masked values from the
exported configuration (default true)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config get

Description

Get the value of a configuration setting by its name in dot notation.

Format

```
mmctl config get [flags]
```

Examples

```
mmctl config get SqlSettings.DriverName
```

Options

```
-h, --help help for get
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config migrate

Description

Migrate a file-based configuration to (or from) a database-based configuration. Point the Mattermost server at the target configuration to start using it. This command only migrates the configuration data from one type to another.

> Note
>
> - To change the store type to use the database, a system admin needs to set a `MM_CONFIG` environment variable and restart the Mattermost server.
> - The `migrate` function requires local mode to be enabled. To do this, add the following line to your Mattermost Environment file:
>
>  ```
>  MM_SERVICESETTINGS_ENABLELOCALMODE=true
>  ```

Format

```
mmctl config migrate [from_config] [to_config] [flags]
```

Examples

```
mmctl config migrate path/to/config.json "postgres://
mmuser:mostest@localhost:5432/mattermost_test?
sslmode=disable&connect_timeout=10" --local
```

Options

```
-h, --help help for migrate
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config patch

Description

Patch the server configuration with the specified configuration file.

Format

```
mmctl config patch <config-file> [flags]
```

Examples

```
mmctl config patch /path/to/config.json
```

Options

```
-h, --help help for reload
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config reload

Description

Reload the server configuration and apply new settings.

Format

```
mmctl config reload [flags]
```

Examples

```
mmctl config reload
```

Options

```
-h, --help help for reload
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config reset

Description

Reset the value of a configuration setting by its name in dot notation or a setting section. Accepts multiple values for array settings.

Format

```
mmctl config reset [flags]
```

Examples

```
mmctl config reset SqlSettings.DriverName LogSettings
```

Options

```
--confirm Confirm you really want to reset all configuration settings to
the default value -h, --help help for reset
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config set

Description

Set the value of a config setting by its name in dot notation. Accepts multiple values for array settings.

> Note
>
> Mattermost plugin uploads can’t be enabled through the API or using mmctl in local mode.

Format

```
mmctl config set [flags]
```

4 Examples

```
mmctl config set SqlSettings.DriverName postgres mmctl config set
SqlSettings.DataSourceReplicas "replica1" "replica2" mmctl config set
PluginSettings.Plugins.com.mattermost.calls.rtcdserviceurl "http://
mattermost-rtcd" mmctl config set LogSettings.AdvancedLoggingJSON
'{"console1":{"Type":"console","Format":"json","Levels":
[{"ID":5,"Name":"debug","Stacktrace":false},
{"ID":4,"Name":"info","Stacktrace":false,"color":36},
{"ID":3,"Name":"warn","Stacktrace":false},
{"ID":2,"Name":"error","Stacktrace":true,"color":31},
{"ID":1,"Name":"fatal","Stacktrace":true,"color":31},
{"ID":0,"Name":"panic","Stacktrace":true,"color":31},
{"ID":10,"Name":"stdlog","Stacktrace":false}],"Options":
{"Out":"stdout"},"MaxQueueSize":1000}}'
```

Options

```
-h, --help help for set
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config show

Description

Print the server configuration and write to STDOUT in JSON format.

Format

```
mmctl config show [flags]
```

Examples

```
mmctl config show
```

Options

```
-h, --help help for show
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl config subpath

Description

Update the hard-coded production client asset paths to take into account Mattermost running on a subpath. This command needs access to the Mattermost `assets` directory to be able to rewrite the paths.

Format

```
mmctl config subpath [flags]
```

Examples

```
# you can rewrite the assets to use a subpath mmctl config subpath --assets-
dir /opt/mattermost/client --path /mattermost # the subpath can have
multiple steps mmctl config subpath --assets-dir /opt/mattermost/client --
path /my/custom/subpath # or you can fallback to the root path passing /
mmctl config subpath --assets-dir /opt/mattermost/client --path /
```

Options

```
-a, --assets-dir string directory of the Mattermost assets in the local
filesystem -h, --help help for subpath -p, --path string path to update the
assets with
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl docs

Description

Generate mmctl documentation.

Format

```
mmctl docs [flags]
```

Options

```
-d, --directory string The directory where the docs would be generated in.
(default "docs") -h, --help help for docs
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export

Manage exports.

> *Child Commands*
>
> - *mmctl export create - Create an export file*
> - *mmctl export delete - Delete an export file*
> - *mmctl export download - Download export files*
> - *mmctl export generate-presigned-url - Generate a pre-signed URL for an export file*
> - *mmctl export job - List, show, and cancel export jobs*
> - *mmctl export job cancel - Cancel export job*
> - *mmctl export job list - List export jobs*
> - *mmctl export job show - Show export job*
> - *mmctl export list - List export files*

Options

```
-h, --help help for group
```

mmctl export create

Description

Create an export file including message attachments.

Format

```
mmctl export create [flags]
```

Options

```
--no-attachments Omit file attachments in the export file. --include-
archived-channels Include archived channels in the export file. -h, --help
help for create
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export delete

Description

Delete an export file.

Format

```
mmctl export delete [exportname] [flags]
```

Example

```
mmctl export delete export_file.zip
```

Options

```
-h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export download

Description

Download export files.

Format

```
mmctl export download [exportname] [filepath] [flags]
```

Example

```
# You can indicate the name of the export and its destination path $ mmctl
export download samplename sample_export.zip # If you only indicate the
name, the path will match it $ mmctl export download sample_export.zip
```

Options

```
-h, --help help for download --num-retries int Number of retries to resume
a download. (Default is 5)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export generate-presigned-url

Description

Generate a pre-signed URL for an export file in cases where a Mattermost Cloud export is large and challenging to download from the Mattermost server.

Requires the `EnableExportDirectDownload` feature flag to be set to `true`.

Format

```
mmctl export generate-presigned-url [exportname] [flags]
```

Options

```
-h, --help help for generate-presigned-url
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export job

Description

List, show, and export jobs.

Options

```
-h, --help help for job
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export job cancel

Description

Cancel an export job.

Format

```
mmctl export job cancel [exportJobID] [flags]
```

Example

```
export job cancel o98rj3ur83dp5dppfyk5yk6osy
```

Options

```
-h, --help help for download
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export job list

Description

List export jobs. Export files include the Job ID in the file name.

Format

```
mmctl export job list [flags]
```

Example

```
mmctl export job list
```

Options

```
--all Fetch all export jobs. ``--page`` flag will be ignored if provided
-h, --help help for list --page int Page number to fetch for the list of
export jobs --per-page int Number of export jobs to be fetched (maximum
200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export job show

Description

Show export job.

Format

```
mmctl export job show [exportJobID] [flags]
```

Example

```
mmctl export job show
```

Options

```
-h, --help help for show
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl export list

Description

List export files. Export files include the job ID in the file name.

Format

```
mmctl export list [flags]
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group

Manage channel and team groups.

> *Child Commands*
>
> - *mmctl group channel - Manage channel groups*
> - *mmctl group list-ldap - List LDAP groups*
> - *mmctl group team - Manage team groups*
> - *mmctl group user - Manage custom user groups*

mmctl group channel

Management of channel groups

> *Child Commands*
>
> - *mmctl group channel disable - Disable group channel constrains*
> - *mmctl group channel enable - Enable group channel constrains*
> - *mmctl group channel list - List channel groups*
> - *mmctl group channel status - Check group status*

Options

```
-h, --help help for group
```

mmctl group channel disable

Description

Disable group constrains in the specified channel.

Format

```
mmctl group channel disable [team]:[channel] [flags]
```

Examples

```
mmctl group channel disable myteam:mychannel
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group channel enable

Description

Enable group constrains in the specified channel.

Format

```
mmctl group channel enable [team]:[channel] [flags]
```

Examples

```
mmctl group channel enable myteam:mychannel
```

Options

```
-h, --help help for enable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group channel list

Description

List the groups associated with a channel.

Format

```
mmctl group channel list [team]:[channel] [flags]
```

Examples

```
mmctl group channel list myteam:mychannel
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group channel status

Description

Show the group constrain status for the specified channel.

Format

```
mmctl group channel status [team]:[channel] [flags]
```

Examples

```
mmctl group channel status myteam:mychannel
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group list-ldap

Description

List LDAP groups.

Format

```
mmctl group list-ldap [flags]
```

Examples

```
mmctl group list-ldap
```

Options

```
-h, --help help for list-ldap
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group team

Manage team groups.

> *Child Commands*
>
> - *mmctl group team disable - Disable group team constrains*
> - *mmctl group team enable - Enable group team constrains*
> - *mmctl group team list - List team groups*
> - *mmctl group team status - Check group constrain status*

Options

```
-h, --help help for group
```

mmctl group team disable

Description

Disable group constrains in the specified team.

Format

```
mmctl group team disable [team] [flags]
```

Examples

```
mmctl group team disable myteam
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group team enable

Description

Enable group constrains in the specified team.

Format

```
mmctl group team enable [team] [flags]
```

Examples

```
mmctl group team enable myteam
```

Options

```
-h, --help help for enable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group team list

Description

List the groups associated with a team.

Format

```
mmctl group team list [team] [flags]
```

Examples

```
mmctl group team list myteam
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group team status

Description

Show the group constrain status for the specified team.

Format

```
mmctl group team status [team] [flags]
```

Examples

```
mmctl group channel status myteam
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl group user

Manage custom user groups.

> *Child Commands*
>
> - *mmctl group user restore - Restore custom user group*

Options

```
-h, --help help for group
```

mmctl group user restore

Description

Restore custom user group.

Format

```
mmctl group user restore [groupname] [flags]
```

Examples

```
mmctl group user restore examplegroup
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl extract

Manage content extraction jobs.

> *Child Commands*
>
> - *mmctl extract job - List and show content extraction jobs*
> - *mmctl extract run - Start a content extraction job*

Options

```
-h, --help help for list
```

mmctl extract job

List and show content extraction jobs.

> *Child Commands*
>
> - *mmctl extract job list - List content extraction jobs*

> • *mmctl extract job show - Show extract job*

mmctl extract job list

Description

List content extraction jobs.

Format

```
mmctl extract job list [flags]
```

Examples

```
mmctl extract job list
```

Options

```
--all Fetch all export jobs. --page flag will be ignore if provided -h, --
help help for list --page int Page number to fetch for the list of export
jobs --per-page int Number of export jobs to be fetched (maximum 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl extract job show

Description

Show extract job.

Format

```
mmctl extract job show [extractJobID] [flags]
```

Examples

```
mmctl extract job show f3d68qkkm7n8xgsfxwuo498rah
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl extract run

Description

Start a content extraction job.

Format

```
mmctl extract run [flags]
```

Examples

```
mmctl extract run
```

Options

```
--from int The timestamp of the earliest file to extract, expressed in
seconds since the unix epoch. -h, --help help for run --to int The
timestamp of the latest file to extract, expressed in seconds since the
unix epoch. Defaults to the current time.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import

Description

Manage imports.

> *Child Commands*
>
> - *mmctl import delete - Delete import files*
> - *mmctl import job - List and show import jobs*
> - *mmctl import job list - List import jobs*
> - *mmctl import job show - Show import job*
> - *mmctl import list - List all import files*
> - *mmctl import list available - List available import files*
> - *mmctl import list incomplete - List incomplete import files uploads*
> - *mmctl import process - Start an import job*
> - *mmctl import upload - Upload import files*
> - *mmctl import validate - Validate an import file*

Options

```
-h, --help help for import
```

mmctl import delete

Description

Delete import files from the server.

Format

```
mmctl import delete [import-name] [flags]
```

Examples

```
mmctl import delete 35uy6cwrqfnhdx3genrhqqznxc_import.zip mmctl import
delete import1.zip import2.zip
```

Options

```
--confirm Confirm you really want to delete the import file(s) -h, --help
help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import job

Description

List and show import jobs.

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import job list

Description

List import jobs

Format

```
mmctl import job list [flags]
```

Examples

```
mmctl import job list
```

Options

```
--all Fetch all import jobs. --page flag will be ignore if provided -h, --
help help for list --page int Page number to fetch for the list of import
jobs --per-page int Number of import jobs to be fetched (maximum 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import job show

Description

Show import job.

Format

```
mmctl import job show [importJobID] [flags]
```

Examples

```
mmctl import job show f3d68qkkm7n8xgsfxwuo498rah
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import list

Description

List all import files.

Examples

```
mmctl import list
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import list available

Description

List available import files.

Format

```
mmctl import list available [flags]
```

Examples

```
mmctl import list available
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import list incomplete

Description

List incomplete import files uploads.

Format

```
mmctl import list incomplete [flags]
```

Examples

```
mmctl import list incomplete
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import process

Description

Start an import job.

Format

```
mmctl import process [importname] [flags]
```

Examples

```
mmctl import process 35uy6cwrqfnhdx3genrhqqznxc_import.zip
```

Options

```
-h, --help help for status --bypass-upload File is read directly from the
filesystem, instead of being processed from the server. Supported in --
local mode only. --extract-content Document attachments will be extracted
and indexed during the import process. We recommend disabling this to
improve performance.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import upload

Description

Upload import files.

Format

```
mmctl import upload [filepath] [flags]
```

Examples

```
mmctl import upload import_file.zip
```

Options

```
-h, --help help for upload --resume Set to true to resume an incomplete
import upload. --upload string The ID of the import upload to resume.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl import validate

Description

Validate an import file.

Format

```
mmctl import validate [filepath] [flags]
```

Examples

```
import validate import_file.zip --team myteam --team myotherteam
```

Options

```
--check-missing-teams Check for teams that are not defined but referenced
in the archive --check-server-duplicates Set to false to ignore teams,
channels, and users already present on the server (Default true) -h, --help
help for validate --ignore-attachments Do not check if the attached files
are present in the archive --team stringarray Predefined teams to assume as
already present on the destination server. Implies --check-missing-teams.
The flag can be repeated
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl integrity

Description

This command is deprecated from Mattermost v9.3. Performs a relational integrity check which returns information about any orphaned record found. This command can only be run using local mode.

Format

```
mmctl integrity [flags]
```

Options

```
--confirm Confirm you really want to run a complete integrity check that
may temporarily harm system performance -h, --help help for integrity -v, --
verbose Show detailed information on integrity check results
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl job

Management of jobs.

> *Child Commands*
>
> - *mmctl job list - List the latest jobs*
> - *mmctl job update - Update the status of a job*

Options

```
-h, --help help for ldap
```

mmctl job list

Description

List the latest jobs.

Format

```
mmctl job list [flags]
```

Examples

```
job list  job list --ids jobID1,jobID2  job list --type ldap_sync --status
success  job list --type ldap_sync --status success --page 0 --per-page 10
```

Options

```
--all Fetch all import jobs. --page flag will be ignored if provided -h, --
help help for list --ids strings Comma-separated list of job IDs to which
the operation will be applied. All other flags are ignored --page int Page
number to fetch for the list of import jobs --per-page int Number of import
jobs to be fetched (default 5) --status string Filter by job status --type
string Filter by job type
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl job update

Description

Update the status of a job.

Important

The following status rules are permitted:

- `in_progress` can be changed to `pending`
- `in_progress` or `pending` can be changed to `cancel_requested`
- `cancel_requested` can be changed to `canceled`

These restrictions can be bypassed with `--force=true`. Bypassing restrictions can have unexpected consequences on your Mattermost server and should be used with caution.

Format

```
mmctl job update [job] [status] [flags]
```

Examples

```
job update myJobID pending  job update myJobID pending --force true  job
update myJobID canceled --force true
```

Options

```
--force Setting a job status is restricted to certain statuses. You can
overwrite these restrictions by using --force. Use this option with
caution. -h, --help help for update
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl ldap

LDAP-related utilities.

> *Child Commands*
>
> - *mmctl ldap idmigrate - Migrate LDAP IdAttribute to a new value*
> - *mmctl ldap job - List and show LDAP synchronization jobs*
> - *mmctl ldap sync - Sync all LDAP users and groups*

Options

```
-h, --help help for ldap
```

mmctl ldap idmigrate

Description

Migrate LDAP `IdAttribute` to a new value. Run this utility to change the value of your ID Attribute without your users losing their accounts. After running the command, you can change the ID Attribute to the new value in the System Console. For example, if your current ID Attribute was `sAMAccountName` and you wanted to change it to `objectGUID`, you would:

1. Wait for an off-peak time when your users won't be impacted by a server restart.
2. Run the command `mmctl ldap idmigrate objectGUID`.
3. Update the config within the System Console to the new value `objectGUID`.
4. Restart the Mattermost server.

Format

```
mmctl ldap idmigrate <objectGUID> [flags]
```

Examples

```
mmctl ldap idmigrate objectGUID
```

Options

```
-h, --help help for idmigrate
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl ldap job

List and show LDAP synchronization jobs.

Child Commands

- mmctl ldap job list - LIst LDAP synchronization jobs
- mmctl ldap job show - Show LDAP synchronization jobs

Options

```
-h, --help help for ldap
```

mmctl ldap job list

Description

List LDAP synchronization jobs.

Format

```
mmctl ldap job list [flags]
```

Examples

```
mmctl ldap job list
```

Options

```
--all Fetch all import jobs. The ``--page`` flag will be ignored if
provided. -h, --help help for list --page int Page number to fetch for the
list of import jobs --per-page int Number of import jobs to be fetched
(maximum 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl ldap job show

Description

Show LDAP synchronization jobs.

Format

```
mmctl ldap job show [ldapJobID] [flags]
```

Examples

```
mmctl ldap show f3d68qkkm7n8xgsfxwuo498rah
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl ldap sync

Description

Synchronize all LDAP users and groups now.

Format

```
mmctl ldap sync [flags]
```

Examples

```
mmctl ldap sync
```

Options

```
-h, --help help for sync --include-removed-members Include members who left
or were removed from a group-synced team/channel. Useful in cases where
synchronized groups are unlinked/re-linked for testing purposes, when LDAP
users are deactivated and reactivated, or when a user leaves a team in
error.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl license

Manage the Mattermost license.

> *Child Commands*
>
> - *mmctl license remove - Remove the current license*
> - *mmctl license upload - Upload a new license*
> - *mmctl license upload-string - Upload a license from a string*

Options

```
-h, --help help for license
```

mmctl license remove

Description

Remove the current license and use Mattermost Team Edition.

Format

```
mmctl license remove [flags]
```

Examples

```
mmctl license remove
```

Options

```
-h, --help help for remove
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl license upload

Description

Upload a license and replace the current license.

Format

```
mmctl license upload [license] [flags]
```

Examples

```
mmctl license upload /path/to/license/mylicensefile.mattermost-license
```

Options

```
-h, --help help for upload
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl license upload-string

Description

Upload a license from a string. Replaces the current license.

Format

```
mmctl license upload-string [license] [flags]
```

Examples

```
license upload-string "mylicensestring"
```

Options

```
-h, --help help for upload
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl logs

Description

Display logs in a human-readable format. As the log format depends on the server, the `--format` flag cannot be used with this command.

Format

```
mmctl logs [flags]
```

Options

```
-h, --help help for logs -l, --logrus Use logrus for formatting -n, --
number int Number of log lines to retrieve (default 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl oauth

Manage registered OAuth2 applications.

> *Child Commands*
>
> - *mmctl oauth list - Remove permissions from a role*

Options

```
-h, --help help for permissions
```

mmctl oauth list

Description

List all registered OAuth2 applications.

Format

```
mmctl oauth list [flags]
```

Example

```
mmctl list
```

Options

```
-h, --help help for add --page int Page number to fetch for the list of
OAuth2 apps --per-page int Number of OAuth2 apps to be fetched per page
(maximum 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions

Manage permissions and roles.

> *Child Commands*
>
> - *mmctl permissions add - Add permissions to a role*
> - *mmctl permissions remove - Remove permissions from a role*

> - *mmctl permissions reset - Reset default permissions for a role*
> - *mmctl permissions role assign - Assign users to role*
> - *mmctl permissions role show - Show the role information*
> - *mmctl permissions role unassign - Unassign users from a role*

Options

```
-h, --help help for permissions
```

mmctl permissions add

Note

Description

Add one or more permissions to an existing role.

Format

```
mmctl permissions add <role> <permission...> [flags]
```

Examples

```
mmctl permissions add system_user list_open_teams mmctl permissions add
system_manager sysconsole_read_user_management_channels
```

Options

```
-h, --help help for add
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions remove

Note

Description

Remove one or more permissions from an existing role. For classified projects, this command can be used to ensure that team members don't know about members outside of their project, and @mentions don't disclose the names of people outside of a classified project.

Format

```
mmctl permissions remove <role> <permission...> [flags]
```

Examples

```
mmctl permissions remove system_user list_open_teams mmctl permissions
remove system_manager sysconsole_read_user_management_channels mmctl
permissions remove system_user view_member
```

Options

```
-h, --help help for remove
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions reset

Note

Description

Reset the given role's permissions to the default settings and overwrite custom settings.

Format

```
mmctl permissions reset <role_name> [flags]
```

Examples

```
# Reset the permissions of the 'system_read_only_admin' role. $ mmctl
permissions reset system_read_only_admin
```

Options

```
-h, --help help for reset
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions role assign

Note

Description

Assign users to a role by username.

Format

```
mmctl permissions role assign <role_name> <username...> [flags]
```

Examples

```
# Assign users with usernames 'john.doe' and 'jane.doe' to the role named
'system_admin'. mmctl permissions role assign system_admin john.doe
jane.doe # Examples using other system roles mmctl permissions role assign
system_manager john.doe jane.doe mmctl permissions role assign
system_user_manager john.doe jane.doe mmctl permissions role assign
system_read_only_admin john.doe jane.doe
```

Options

```
-h, --help help for assign
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions role show

Description

Show all the information about a role.

Format

```
mmctl permissions role show <role_name> [flags]
```

Examples

```
mmctl permissions show system_user
```

Options

```
-h, --help help for show
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl permissions role unassign

Description

Unassign users from a role by username. Available in Mattermost Professional and Mattermost Enterprise.

Format

```
mmctl permissions role unassign <role_name> <username...> [flags]
```

Examples

```
# Unassign users with usernames 'john.doe' and 'jane.doe' from the role
named 'system_admin'. mmctl permissions unassign system_admin john.doe
jane.doe # Examples using other system roles mmctl permissions unassign
system_manager john.doe jane.doe mmctl permissions unassign
system_user_manager john.doe jane.doe mmctl permissions unassign
system_read_only_admin john.doe jane.doe
```

Options

```
-h, --help help for unassign
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin

Manage plugins.

> *Child Commands*
>
> - *mmctl plugin add - Add plugins*
> - *mmctl plugin delete - Remove plugins*
> - *mmctl plugin disable - Disable plugins*
> - *mmctl plugin enable - Enable plugins*
> - *mmctl plugin install-url - Install plugin from URL*
> - *mmctl plugin list - List plugins*
> - *mmctl plugin marketplace - Manage Marketplace plugins*

Options

```
-h, --help help for plugin
```

mmctl plugin add

Description

Add plugins to your Mattermost server.

Format

```
mmctl plugin add [plugins] [flags]
```

Examples

```
mmctl plugin add hovercardexample.tar.gz pluginexample.tar.gz
```

Options

```
-f, --force overwrite a previously installed plugin with the same ID, if
any -h, --help help for add
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin delete

Description

Delete previously uploaded plugins from your Mattermost server.

Format

```
mmctl plugin delete [plugins] [flags]
```

Examples

```
mmctl plugin delete hovercardexample pluginexample
```

Options

```
-h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin disable

Description

Disable plugins. Disabled plugins are immediately removed from the user interface and logged out of all sessions.

Format

```
mmctl plugin disable [plugins] [flags]
```

Examples

```
mmctl plugin disable hovercardexample pluginexample
```

Options

```
-h, --help help for disable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin enable

Description

Enable plugins for use on your Mattermost server.

Format

```
mmctl plugin enable [plugins] [flags]
```

Examples

```
mmctl plugin enable hovercardexample pluginexample
```

Options

```
-h, --help help for enable
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin install-url

Description

Supply URLs to plugins compressed in a `.tar.gz` file. Plugins must be enabled in the server's config settings.

Format

```
mmctl plugin install-url <url>... [flags]
```

Examples

```
# You can install one plugin $ mmctl plugin install-url https://example.com/
mattermost-plugin.tar.gz # Or install multiple plugins at a time $ mmctl
plugin install-url https://example.com/mattermost-plugin-one.tar.gz https://
example.com/mattermost-plugin-two.tar.gz
```

Options

```
-f, --force overwrite a previously installed plugin with the same ID, if
any -h, --help help for install-url
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin list

Description

List all enabled and disabled plugins installed on your Mattermost server.

Format

```
mmctl plugin list [flags]
```

Examples

```
mmctl plugin list
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin marketplace

Manage Marketplace plugins.

> *Child Commands*
>
> - *mmctl plugin marketplace install - Install a plugin from the Plugin Marketplace*
> - *mmctl plugin marketplace list - List plugins on the Plugin Marketplace*

Options

```
-h, --help help for marketplace
```

mmctl plugin marketplace install

Description

Install a plugin available on the Plugin Marketplace server. The latest version of the plugin will be installed.

Format

```
mmctl plugin marketplace install <id> [flags]
```

Examples

```
$ mmctl plugin marketplace install jitsi
```

Options

```
-h, --help help for install
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl plugin marketplace list

Description

Get all plugins available from the Plugin Marketplace server, merging data from locally installed plugins as well as prepackaged plugins shipped with the server.

Format

```
mmctl plugin marketplace list [flags]
```

Examples

```
# You can list all available plugins $ mmctl plugin marketplace list --all
# Pagination options can be used $ mmctl plugin marketplace list --page 2 --
per-page 10 # Filtering narrows down the search $ mmctl plugin marketplace
list --filter jit # You can retrieve only local plugins $ mmctl plugin
marketplace list --local-only
```

Options

```
--all Fetch all plugins. --page flag will be ignore if provided --filter
string Filter plugins by ID, name or description -h, --help help for list --
local-only Only retrieve local plugins --page int Page number to fetch for
the list of users --per-page int Number of users to be fetched (maximum
200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl post

Manage posts.

> *Child Commands*
>
> - *mmctl post create - Create a post*
> - *mmctl post delete - Delete a post*
> - *mmctl post list - List posts for a channel*

Options

```
-h, --help help for post
```

mmctl post create

Description

Create a post.

Format

```
mmctl post create [flags]
```

Examples

```
mmctl post create myteam:mychannel --message "some text for the post"
```

Options

```
-h, --help help for create -m, --message string Message for the post -r, --
reply-to string Post id to reply to
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl post delete

Description

Mark a post as deleted and remove it and all attachments from the client without permanently deleting it from the database. Permanently delete a post and all attachments using the `--permanent` flag.

Format

```
mmctl post delete [posts] [flags]
```

Examples

Mark post as deleted:

```
mmctl post delete udjmt396tjghi8wnsk3a1qs1sw
```

Permanently delete a post and its file contents from the database and filestore:

```
mmctl post delete udjmt396tjghi8wnsk3a1qs1sw --permanent
```

Permanently delete multiple posts and their file contents from the database and filestore:

```
mmctl post delete udjmt396tjghi8wnsk3a1qs1sw 7jgcjt7tyjyyu83qz81wo84w6o --
permanent
```

Options

```
--confirm Confirm you really want to delete the post and a DB backup has
been performed -h, --help help for delete --permanent Permanently delete
the post and its contents from the database
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl post list

Description

List posts for a channel.

Format

```
mmctl post list [flags]
```

Examples

```
mmctl post list myteam:mychannel mmctl post list myteam:mychannel --number
20
```

Options

```
-f, --follow Output appended data as new messages are posted to the channel
-h, --help help for list -n, --number int Number of messages to list
(default 20) -i, --show-ids Show posts ids -s, --since string List messages
posted after a certain time (ISO 8601)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl roles

Description

Promote users to the system admin role, or remove system admin privileges from users.

Format

Promote users to the system admin role:

```
mmctl roles system_admin [users] [flags]
```

Remove system admin privileges:

```
mmctl roles member [users] [flags]
```

Examples

Promote a user to the system admin role:

```
mmctl roles system_admin john_doe
```

Promote multiple users to the system admin role:

```
mmctl roles system_admin john_doe jane_doe
```

Remove system admin privileges from a user:

```
mmctl roles member john_doe
```

Remove system admin privileges from multiple users:

```
mmctl roles member john_doe jane_doe
```

Options

```
-h, --help help for roles
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl saml

SAML-related utilities.

> *Child Commands*
>
> - *mmctl saml auth-data-reset - Reset AuthData field to Email*

Options

```
-h, --help help for system
```

mmctl saml auth-data-reset

Description

Resets the AuthData field for SAML users to their email. Run this utility after setting the 'id' SAML attribute to an empty value.

Format

```
mmctl saml auth-data-reset [flags]
```

Examples

```
# Reset all SAML users' AuthData field to their email, including deleted
users $ mmctl saml auth-data-reset --include-deleted # Show how many users
would be affected by the reset $ mmctl saml auth-data-reset --dry-run
# Skip confirmation for resetting the AuthData $ mmctl saml auth-data-reset
-y # Only reset the AuthData for the following SAML users $ mmctl saml auth-
data-reset --users userid1,userid2
```

Options

```
--dry-run Dry run only -h, --help help for auth-data-reset --include-
deleted Include deleted users --users strings Comma-separated list of user
IDs to which the operation will be applied -y, --yes Skip confirmation
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl sampledata

Description

Generate a sample data file and store it locally, or directly import it to the remote server.

Format

```
mmctl sampledata [flags]
```

Examples

```
# You can create a sampledata file and store it locally $ mmctl sampledata
--bulk sampledata-file.jsonl # Or you can simply print it to the STDOUT $
mmctl sampledata --bulk - # You can customize the amount of entities to
create $ mmctl sampledata -t 7 -u 20 -g 4 # The sampledata file can be
directly imported into the remote server by not specifying a ``--bulk```
flag $ mmctl sampledata # Sample users can be created with profile pictures
$ mmctl sampledata --profile-images ./images/profiles
```

Options

```
-b, --bulk string Optional. Path to write a JSONL bulk file instead of
uploading into the remote server. --channel-memberships int The number of
sample channel memberships per user in a team. (default 5) --channels-per-
team int The number of sample channels per team. (default 10) --deactivated-
users int The number of deactivated users. --direct-channels int The number
of sample direct message channels. (default 30) --group-channels int The
number of sample group message channels. (default 15) -g, --guests int The
number of sample guests. (default 1) -h, --help help for sampledata --posts-
per-channel int The number of sample post per channel. (default 100) --
posts-per-direct-channel int The number of sample posts per direct message
channel. (default 15) --posts-per-group-channel int The number of sample
posts per group message channel. (default 30) --profile-images string
Optional. Path to folder with images to randomly pick as user profile
image. -s, --seed int Seed used for generating the random data (Different
seeds generate different data). (default 1) --team-memberships int The
number of sample team memberships per user. (default 2) -t, --teams int The
number of sample teams. (default 2) -u, --users int The number of sample
users. (default 15)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system

System management commands for interacting with the server state and configuration.

> *Child Commands*
>
> - *mmctl system clearbusy - Clear the busy state*
> - *mmctl system getbusy - Get the current busy state*
> - *mmctl system setbusy - Set the busy state to* `true`

> - *mmctl system status - Print the status of the server*
> - *mmctl system supportpacket - Download a Support Packet*
> - *mmctl system version - Print the remote server version build number*

Options

```
-h, --help help for system
```

mmctl system clearbusy

Description

Clear the busy state which re-enables non-critical services.

Format

```
mmctl system clearbusy [flags]
```

Examples

```
mmctl system clearbusy
```

Options

```
-h, --help help for clearbusy
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system getbusy

Description

Get the server busy state (high load) and timestamp corresponding to when the server busy flag will be automatically cleared.

Format

```
mmctl system getbusy [flags]
```

Examples

```
mmctl system getbusy
```

Options

```
-h, --help help for getbusy
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system setbusy

Description

Set the busy state to `true` for the specified number of seconds which disables non-critical services.

Format

```
mmctl system setbusy -s [seconds] [flags]
```

Examples

```
mmctl system setbusy -s 3600
```

Options

```
-h, --help help for setbusy -s, --seconds uint Number of seconds until
server is automatically marked as not busy (default 3600)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system status

Description

Print the server status which is calculated using several basic server healthchecks.

Format

```
mmctl system status [flags]
```

Examples

```
mmctl system status
```

Options

```
-h, --help help for status
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system supportpacket

Description

Generate and download a Support Packet of the Mattermost server to share with Mattermost Support.

Format

```
mmctl system supportpacket [flags]
```

Examples

```
mmctl system supportpacket
```

Options

```
-h, --help help for version -o, --output-file string Output file name.
Default is ``mattermost_support_packet_YYYY-MM-DD-HH-MM.zip``.
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl system version

Description

Print the server version build number of the currently connected Mattermost instance.

Format

```
mmctl system version [flags]
```

Examples

```
mmctl system version
```

Options

```
-h, --help help for version
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team

Manage teams.

Important

When specifying team names within mmctl commands, you must use the `team-URL` version of the team name, rather than the display name you see in the channel sidebar. Your `team-URL` does not contain spaces. Run the mmctl team list command to return a list of all teams on the server in `team-URL` format. See the team name and URL selection documentation for details.

Child Commands

- mmctl team archive - Archive some teams
- mmctl team create - Create teams
- mmctl team delete - Delete teams
- mmctl team list - List teams
- mmctl team modify - Modify teams
- mmctl team rename - Rename teams
- mmctl team restore - Restore teams
- mmctl team search - Search teams
- mmctl team users - Manage team users

Options

```
-h, --help help for team
```

mmctl team archive

Description

Archive a team along with all related information including posts from the database.

Format

```
mmctl team archive [teams] [flags]
```

Examples

```
mmctl team archive myteam
```

Options

```
--confirm Confirm you really want to archive the team and a database backup
has been performed -h, --help help for archive
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team create

Description

Create a team.

Format

```
mmctl team create [flags]
```

Examples

```
mmctl team create --name mynewteam --display_name "My New Team" mmctl team
create --name private --display_name "My New Private Team" --private
```

Options

```
--display_name string Team Display Name --email string Administrator Email
(anyone with this email is automatically a team admin) -h, --help help for
create --name string Team Name --private Create a private team
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team delete

Description

Permanently delete a team along with all related information including posts from the database.

> Note
>
> Requires the Enable API Team Deletion configuration setting to be set to `true`. If this configuration setting is set to `false`, attempting to delete the team using mmctl fails.

Format

```
mmctl team delete [teams] [flags]
```

Examples

```
mmctl team delete myteam
```

Options

```
--confirm Confirm you really want to delete the team and a database backup
has been performed -h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team list

Description

List all teams on the server.

Format

```
mmctl team list [flags]
```

Examples

```
mmctl team list
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team modify

Description

Modify a team's privacy setting to public or private.

Format

```
mmctl team modify [teams] [flag] [flags]
```

Examples

```
mmctl team modify myteam --private
```

Options

```
-h, --help help for modify --private Modify team to be private --public
Modify team to be public
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team rename

Description

Rename an existing team.

Format

```
mmctl team rename [team] [flags]
```

Examples

```
mmctl team rename old-team --display_name 'New Display Name'
```

Options

```
--display_name string Team Display Name -h, --help help for rename
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team restore

Description

Restore archived teams.

Format

```
mmctl team restore [teams] [flags]
```

Examples

```
mmctl team restore myteam
```

Options

```
-h, --help help for restore
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team search

Description

Search for teams based on name.

Format

```
mmctl team search [teams] [flags]
```

Examples

```
mmctl team search team1
```

Options

```
-h, --help help for search
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team users

Manage team users.

> *Child Commands*
>
> - *mmctl team users add - Add users to a team*
> - *mmctl team users remove - Remove users from a team*

Options

```
-h, --help help for token
```

mmctl team users add

Description

Add specified users to a team.

Format

```
mmctl team users add [team] [users] [flags]
```

Examples

```
mmctl team users add myteam user@example.com username
```

Options

```
-h, --help help for add
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl team users remove

Description

Remove specified users from a team.

Format

```
mmctl team users remove [team] [users] [flags]
```

Examples

```
mmctl team users remove myteam user@example.com username
```

Options

```
-h, --help help for remove
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl token

Manage users' access tokens.

> *Child Commands*
>
> - *mmctl token generate - Generate token for a user*
> - *mmctl token list - List users' tokens*
> - *mmctl token revoke - Revoke tokens for a user*

Options

```
-h, --help help for token
```

mmctl token generate

Description

Generate token for a user.

Format

```
mmctl token generate [user] [description] [flags]
```

Examples

```
mmctl token generate testuser test-token
```

Options

```
-h, --help help for generate
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl token list

Description

List the tokens belonging to a user.

Format

```
mmctl token list [user] [flags]
```

Examples

```
mmctl user tokens testuser
```

Options

```
--active List only active tokens (default true) --all Fetch all tokens. --
page flag will be ignore if provided -h, --help help for list --inactive
List only inactive tokens --page int Page number to fetch for the list of
users --per-page int Number of users to be fetched (maximum 200)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl token revoke

Description

Revoke tokens for a user.

Format

```
mmctl token revoke [token-ids] [flags]
```

Examples

```
mmctl revoke testuser test-token-id
```

Options

```
-h, --help help for revoke
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user

Manage users.

> *Child Commands*
>
> - *mmctl user activate - Activate a user*
> - *mmctl user change-password - Change a user's password*
> - *mmctl user convert - Convert users to bots or convert bots to users*
> - *mmctl user create - Create user*
> - *mmctl user deactivate - Deactivate user*
> - *mmctl user delete - Delete users*
> - *mmctl user deleteall - Delete all users and all posts (local command only)*
> - *mmctl user demote - Demote users to guests*
> - *mmctl user email - Set user email*
> - *mmctl user invite - Invite user*
> - *mmctl user list - List users*
> - *mmctl user migrate-auth - Bulk migrate user accounts authentication type*
> - *mmctl user preference - Manage user preferences*
> - *mmctl user promote - Promote guests to users*
> - *mmctl user reset-password - Reset user password*
> - *mmctl user resetmfa - Reset a user's MFA token*
> - *mmctl user search - Search for a user*
> - *mmctl user username - Change username of the user*
> - *mmctl user verify - Mark user's email as verified*

Options

```
-h, --help help for user
```

mmctl user activate

Description

Activate users that have been deactivated.

Format

```
mmctl user activate [emails, usernames, userIds] [flags]
```

Examples

```
mmctl user activate user@example.com mmctl user activate username
```

Options

```
-h, --help help for activate
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user change-password

Description

Change the password of a user to the one provided. If the user is changing their own password, the flag `--current` must indicate the current password.

The flag `--hashed` can be used to indicate that the new password has been introduced as already hashed.

Format

```
mmctl user change-password <user> [flags]
```

Examples

```
# If you have system permissions, you can change other user's passwords $
mmctl user change-password john_doe --password new-password # If you are
changing your own password, you need to provide the current one $ mmctl
user change-password my-username --current current-password --password new-
password # You can ommit these flags to introduce them interactively $
mmctl user change-password my-username Are you changing your own password?
(YES/NO): YES Current password: New password: # If you have system
permissions, you can update the password with the already hashed new #
password. The hashing method should be the same that the server uses
internally. $ mmctl user change-password john_doe --password
HASHED_PASSWORD --hashed
```

Options

```
-c, --current string The current password of the user. Use only if changing
your own password --hashed The supplied password is already hashed -h, --
help help for change-password -p, --password string The new password for
the user
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user convert

Description

Convert user accounts to bots, or convert bots to user accounts.

Format

```
mmctl user convert (--bot [emails] [usernames] [userIds] | --user
<username> --password PASSWORD [--email EMAIL]) [flags]
```

Examples

```
# You can convert a user to a bot providing an email, an ID, or a username
$ mmctl user convert user@example.com --bot # Or you can convert multiple
users at a time $ mmctl user convert user@example.com anotherUser --bot #
You can convert a bot to a user and specify the email and password that the
user will have after conversion $ mmctl user convert botusername --email
new.email@email.com --password password --user
```

Options

```
--bot If supplied, convert users to bots --email string The email address
for the converted user account. Required when the "bot" flag is set --
firstname string The first name for the converted user account. Required
when the "bot" flag is set -h, --help help for convert --lastname string
The last name for the converted user account. Required when the "bot" flag
is set --locale string The locale (e.g., EN, FR) for the converted new user
account. Required when the "bot" flag is set --nickname string The nickname
for the converted user account. Required when the "bot" flag is set --
password string The password for converted new user account. Required when
"user" flag is set --system_admin If supplied, the converted user will be a
system admin. Defaults to false. Required when the "bot" flag is set --user
If supplied, convert a bot to a user --username string Username for the
converted user account. Required when the "bot" flag is set
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user create

Description

Create a user.

Format

```
mmctl user create [flags]
```

Examples

```
# You can create a user $ mmctl user create --email user@example.com --
username userexample --password Password1 # You can define optional fields
like first name, last name, and nickname $ mmctl user create --email
user@example.com --username userexample --password Password1 --firstname
User --lastname Example --nickname userex # You can also create the user as
a system sdmin $ mmctl user create --email user@example.com --username
userexample --password Password1 --system-admin # You can verify user on
creation if you have the correct permissions $ mmctl user create --email
user@example.com --username userexample --password Password1 --system-admin
--email-verified
```

Options

```
--disable-welcome-email Optional. If supplied, the new user will not
receive a welcome email. Defaults to false --email string Required. The
email address for the new user account --email-verified Optional. If
supplied, the new user will have the email verified. Defaults to false --
firstname string Optional. The first name for the new user account --guest
Optional. If supplied, the new user will be a guest. Defaults to false -h,
--help help for create --lastname string Optional. The last name for the
new user account --locale string Optional. The locale (ex: en, fr) for the
new user account --nickname string Optional. The nickname for the new user
account --password string Required. The password for the new user account --
system-admin Optional. If supplied, the new user will be a system
administrator. Defaults to false --username string Required. Username for
the new user account
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user deactivate

Description

Deactivate users. Deactivated users are immediately logged out of all sessions and are unable to log back in.

Format

```
mmctl user deactivate [emails, usernames, userIds] [flags]
```

Examples

```
mmctl user deactivate user@example.com mmctl user deactivate username
```

Options

```
-h, --help help for deactivate
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user delete

Description

Permanently delete users along with all related information including posts from the database.

> Note
>
> Requires the Enable API User Deletion configuration setting to be set to `true`. If this configuration setting is set to `false`, attempting to delete the user using mmctl fails.

Format

```
mmctl user delete [users] [flags]
```

Examples

```
mmctl user delete user@example.com
```

Options

```
--confirm Confirm you really want to delete the user and a database backup
has been performed -h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user deleteall

Description

Permanently delete all users and all related information including posts. This command can only be run in local mode.

Format

```
mmctl user deleteall [flags]
```

Examples

```
mmctl user deleteall
```

Options

```
--confirm Confirm you really want to delete the user and a database backup
has been performed -h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user demote

Description

Demote a user to a guest.

Format

```
mmctl user demote [users] [flags]
```

Examples

```
mmctl user demote user1 user2
```

Options

```
-h, --help help for demote
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user email

Description

Change the email address associated with a user.

> Note
>
> From Mattermost v10.9, email addresses enclosed in angle brackets (e.g., `<billy@example.com>`) will be rejected. To avoid issues, ensure all user emails comply with the plain address format (e.g., `billy@example.com`). In addition, we strongly recommend taking proactive steps to audit and update Mattermost user data to align with this product change, as impacted users may face issues accessing Mattermost or managing their user profile. You can update these user emails manually using this mmctl command: `mmctl user email "<affecteduser@domain.com>" affecteduser@domain.com`.

Format

```
mmctl user email [user] [new email] [flags]
```

Examples

```
mmctl user email testuser user@example.com
```

Options

```
-h, --help help for email
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user invite

Description

Send an email invite to a user to join a team. You can invite a user to multiple teams by listing them. You can specify teams by name or ID.

Format

```
mmctl user invite [email] [teams] [flags]
```

Examples

```
mmctl user invite user@example.com myteam mmctl user invite
user@example.com myteam1 myteam2
```

Options

```
-h, --help help for invite
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user list

Description

List all users.

Format

```
mmctl user list [flags]
```

Examples

```
mmctl user list
```

Options

```
--all Fetch all users. --page flag will be ignore if provided -h, --help
help for list --page int Page number to fetch for the list of users --per-
page int Number of users to be fetched (maximum 200) --team string If
supplied, only users belonging to this team will be listed
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user migrate-auth

Description

Migrate accounts from one authentication provider to another. For example, you can upgrade your authentication provider from email to LDAP.

Format

```
mmctl user migrate-auth [from_auth] [to_auth] [migration-options] [flags]
```

Arguments

> ***from_auth:***
>
> *The authentication service to migrate users accounts from. Supported options: email, gitlab, google, ldap, office365, saml.*
>
> ***to_auth:***
>
> *The authentication service to migrate users to. Supported options: ldap, saml.*
>
> ***migration-options (ldap):***
> *match_field:*
>
> *The field that is guaranteed to be the same in both authentication services. For example, if the users emails are consistent set to email. Supported options: email, username.*
>
> ***migration-options (saml):***
> ***users_file:***
>
> *The path of a json file with the usernames and emails of all users to migrate to SAML. The username and email must be the same that the SAML service provider store. And the email must match with the email in mattermost database.*
>
> ```
> Example json content: {  "usr1@email.com": "usr.one",
> "usr2@email.com": "usr.two" }
> ```

Examples

```
mmctl user migrate-auth email saml users.json
```

user.json Example

```
[ { "email": "user1@example.com", "auth_data": { "saml": { "idp_id":
"saml_idp_1", "saml_user_id": "user123" } } }, { "email":
"user2@example.com", "auth_data": { "saml": { "idp_id": "saml_idp_2",
"saml_user_id": "user456" } } } ]
```

Options

```
--auto Automatically migrate all users. Assumes the usernames and emails
are identical between Mattermost and SAML services. (saml only) --confirm
Confirm you really want to proceed with auto migration. (saml only) --force
Force the migration to occur even if there are duplicates on the LDAP
server. Duplicates will not be migrated. (ldap only) -h, --help help for
migrate-auth
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user preference

Description

Manage user preferences.

> *Child Commands*
>
> - *mmctl user preference get - Get a specific user preference*
> - *mmctl user preference list - List user preferences*
> - *mmctl user preference set - Set a specific user preference*
> - *mmctl user preference delete - Delete a specific user preference*

Options

```
-h, --help help for auth
```

mmctl user preference get

Description

Get a specific user preference.

Format

```
mmctl user preference get --category [category] --name [name] [users]
[flags]
```

Examples

```
preference get --category display_settings --name use_military_time
user@example.com
```

Options

```
-c, --category string The category of the preference -h, --help help for
get -n, --name string The name of the preference
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user preference list

Description

List user preferences.

Format

```
mmctl user preference list [--category category] [users] [flags]
```

Examples

```
preference list user@example.com
```

Options

```
-c, --category string The optional category by which to filter -h, --help
help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user preference set

Description

Set a specific user preference.

Format

```
mmctl user preference set --category [category] --name [name] --value
[value] [users] [flags]
```

Examples

```
preference set --category display_settings --name use_military_time --value
true user@example.com
```

Options

```
-c, --category string The category of the preference -h, --help help for
set -n, --name string The name of the preference -v, --value string The
value of the preference
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user preference delete

Description

Delete a specific user preference.

Format

```
mmctl user preference delete --category [category] --name [name] [users]
[flags]
```

Examples

```
preference delete --category display_settings --name use_military_time
user@example.com
```

Options

```
-c, --category string The category of the preference -h, --help help for
delete -n, --name string The name of the preference
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user promote

Description

Promote a guest to a user.

Format

```
mmctl user promote [guests] [flags]
```

Examples

```
mmctl user promote guest1 guest2
```

Options

```
-h, --help help for promote
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user reset-password

Description

Send users an email to reset their password.

Format

```
mmctl user reset-password [users] [flags]
```

Examples

```
mmctl user reset-password user@example.com
```

Options

```
-h, --help help for reset-password
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user resetmfa

Description

Turn off multi-factor authentication for a user. If MFA enforcement is enabled, the user will be forced to re-enable MFA as soon as they log in.

Format

```
mmctl user resetmfa [users] [flags]
```

Examples

```
mmctl user resetmfa user@example.com
```

Options

```
-h, --help help for resetmfa
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user search

Description

Search for users based on username, email, or user ID. The command returns user information including usernames, email addresses, first and last names, and account status. From Mattermost v10.10, output also includes the user's deactivation status to help system admins identify inactive accounts.

Format

```
mmctl user search [users] [flags]
```

Examples

```
mmctl user search user1@mail.com user2@mail.com
```

Options

```
-h, --help help for search
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user username

Description

Change the username of the user.

Format

```
mmctl user username [user] [new username] [flags]
```

Examples

```
mmctl user username testuser newusername
```

Options

```
-h, --help help for version
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl user verify

Description

Mark user's email as verified without requiring user to complete email verification path.

Format

```
mmctl user verify [users] [flags]
```

Examples

```
mmctl user verify user1
```

Options

```
-h, --help help for version
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl version

Description

Print the version of mmctl.

Format

```
mmctl version [flags]
```

Options

```
-h, --help help for version
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook

Description

Manage webhooks.

> *Child Commands*
>
> - *mmctl webhook create-incoming - Create an incoming webhook*
> - *mmctl webhook create-outgoing - Create an outgoing webhook*
> - *mmctl webhook delete - Delete webhooks*
> - *mmctl webhook list - List webhooks*
> - *mmctl webhook modify-incoming - Modify an incoming webhook*

> - *mmctl webhook modify-outgoing - Modify an outgoing webhook*
> - *mmctl webhook show - Show a webhook*

Options

```
-h, --help help for webhook
```

mmctl webhook create-incoming

Description

Create an incoming webhook to allow external posting of messages to a specific channel.

Format

```
mmctl webhook create-incoming [flags]
```

Examples

```
mmctl webhook create-incoming --channel [channelID] --user [userID] --
display-name [displayName] --description [webhookDescription] --lock-to-
channel --icon [iconURL]
```

Options

```
--channel string Channel ID (required) --description string Incoming
webhook description --display-name string Incoming webhook display name -h,
--help help for create-incoming --icon string Icon URL --lock-to-channel
Lock to channel --user string User ID (required)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook create-outgoing

Description

Create an outgoing webhook to allow external posting of messages from a specific channel.

Format

```
mmctl webhook create-outgoing [flags]
```

Examples

```
mmctl webhook create-outgoing --team myteam --user myusername --display-
name mywebhook --trigger-word "build" --trigger-word "test" --url http://
localhost:8000/my-webhook-handler  mmctl webhook create-outgoing --team
myteam --channel mychannel --user myusername --display-name mywebhook --
description "My cool webhook" --trigger-when start --trigger-word build --
trigger-word test --icon http://localhost:8000/my-slash-handler-bot-
icon.png --url http://localhost:8000/my-webhook-handler --content-type
"application/json"
```

Options

```
--channel string Channel name or ID --content-type string Content-type --
description string Outgoing webhook description --display-name string
Outgoing webhook display name -h, --help help for create-outgoing --icon
string Icon URL --team string Team name or ID (required) --trigger-when
string When to trigger webhook (exact: for first word matches a trigger
word exactly, start: for first word starts with a trigger word) (default
"exact") --trigger-word stringArray Word to trigger webhook (required) --
url stringArray Callback URL (required) --user string The username, email,
or ID of the user that the webhook should post as (required)
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook delete

Description

Delete a webhook with a given ID.

Format

```
mmctl webhook delete [flags]
```

Examples

```
mmctl webhook delete [webhookID]
```

Options

```
-h, --help help for delete
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook list

Description

Print a list of all webhooks.

Format

```
mmctl webhook list [flags]
```

Examples

```
mmctl webhook list myteam
```

Options

```
-h, --help help for list
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook modify-incoming

Description

Modify an existing incoming webhook by changing its title, description, channel, or icon URL.

Format

```
mmctl webhook modify-incoming [flags]
```

Examples

```
mmctl webhook modify-incoming [webhookID] --channel [channelID] --display-
name [displayName] --description [webhookDescription] --lock-to-channel --
icon [iconURL]
```

Options

```
--channel string Channel ID --description string Incoming webhook
description --display-name string Incoming webhook display name -h, --help
help for modify-incoming --icon string Icon URL --lock-to-channel Lock to
channel
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook modify-outgoing

Description

Modify an existing outgoing webhook by changing its title, description, channel, icon, url, content-type, or triggers.

Format

```
mmctl webhook modify-outgoing [flags]
```

Examples

```
mmctl webhook modify-outgoing [webhookId] --channel [channelId] --display-
name [displayName] --description "New webhook description" --icon http://
localhost:8000/my-slash-handler-bot-icon.png --url http://localhost:8000/my-
webhook-handler --content-type "application/json" --trigger-word test --
trigger-when start
```

Options

```
--channel string Channel name or ID --content-type string Content-type --
description string Outgoing webhook description --display-name string
Outgoing webhook display name -h, --help help for modify-outgoing --icon
string Icon URL --trigger-when string When to trigger webhook (exact: for
first word matches a trigger word exactly, start: for first word starts
with a trigger word) --trigger-word stringArray Word to trigger webhook --
url stringArray Callback URL
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl webhook show

Description

Show the webhook specified by `[webhookId]`.

Format

```
mmctl webhook show [webhookId] [flags]
```

Examples

```
mmctl webhook show w16zb5tu3n1zkqo18goqry1je
```

Options

```
-h, --help help for show
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

mmctl websocket

Description

Display websocket in a human-readable format.

Format

```
mmctl websocket [flags]
```

Options

```
-h, --help help for websocket
```

Options inherited from parent commands

```
--config string path to the configuration file (default "$XDG_CONFIG_HOME/
mmctl/config") --disable-pager disables paged output --insecure-sha1-
intermediate allows to use insecure TLS protocols, such as SHA-1 --insecure-
tls-version allows to use TLS versions 1.0 and 1.1 --json the output format
will be in json format --local allows communicating with the server through
a unix socket --quiet prevent mmctl to generate output for the commands --
strict will only run commands if the mmctl version matches the server one --
suppress-warnings disables printing warning messages
```

On this page

- mmctl command line tool
- mmctl command line tool
  - mmctl usage notes
  - mmctl commands
  - Install mmctl
    - Release package
    - Local git clone
    - Homebrew
  - Build mmctl
  - Local mode
    - Activating local mode
    - Using local mode
  - Running mmctl tests
  - mmctl auth
    - mmctl auth clean
    - mmctl auth current
    - mmctl auth delete
    - mmctl auth list
    - mmctl auth login
      - Password
      - MFA
      - Access tokens
      - Installing shell completions
    - mmctl auth renew
    - mmctl auth set

- mmctl bot
  - mmctl bot assign
  - mmctl bot create
  - mmctl bot disable
  - mmctl bot enable
  - mmctl bot list
  - mmctl bot update
- mmctl channel
  - mmctl channel archive
  - mmctl channel create
  - mmctl channel delete
  - mmctl channel list
  - mmctl channel make-private
  - mmctl channel modify
  - mmctl channel move
  - mmctl channel rename
  - mmctl channel restore
  - mmctl channel search
  - mmctl channel unarchive
  - mmctl channel users
  - mmctl channel users add
  - mmctl channel users remove
- mmctl command
  - mmctl command archive
  - mmctl command create
  - mmctl command delete
  - mmctl command list
  - mmctl command modify
  - mmctl command move
  - mmctl command show
- mmctl completion
  - mmctl completion bash
  - mmctl completion zsh

- mmctl config
  - mmctl config edit
  - mmctl config export
  - mmctl config get
  - mmctl config migrate
  - mmctl config patch
  - mmctl config reload
  - mmctl config reset
  - mmctl config set
  - mmctl config show
  - mmctl config subpath
- mmctl docs
- mmctl export
  - mmctl export create
  - mmctl export delete
  - mmctl export download
  - mmctl export generate-presigned-url
  - mmctl export job
  - mmctl export job cancel
  - mmctl export job list
  - mmctl export job show
  - mmctl export list
- mmctl group
- mmctl group channel
  - mmctl group channel disable
  - mmctl group channel enable
  - mmctl group channel list
  - mmctl group channel status
  - mmctl group list-ldap
- mmctl group team
  - mmctl group team disable
  - mmctl group team enable
  - mmctl group team list

    - mmctl group team status
  - mmctl group user
    - mmctl group user restore
  - mmctl extract
    - mmctl extract job
    - mmctl extract job list
    - mmctl extract job show
    - mmctl extract run
  - mmctl import
    - mmctl import delete
    - mmctl import job
    - mmctl import job list
    - mmctl import job show
    - mmctl import list
    - mmctl import list available
    - mmctl import list incomplete
    - mmctl import process
    - mmctl import upload
    - mmctl import validate
  - mmctl integrity
  - mmctl job
    - mmctl job list
    - mmctl job update
  - mmctl ldap
    - mmctl ldap idmigrate
  - mmctl ldap job
    - mmctl ldap job list
    - mmctl ldap job show
  - mmctl ldap sync
  - mmctl license
    - mmctl license remove
    - mmctl license upload
    - mmctl license upload-string

- mmctl logs
- mmctl oauth
  - mmctl oauth list
- mmctl permissions
  - mmctl permissions add
  - mmctl permissions remove
  - mmctl permissions reset
  - mmctl permissions role assign
  - mmctl permissions role show
  - mmctl permissions role unassign
- mmctl plugin
  - mmctl plugin add
  - mmctl plugin delete
  - mmctl plugin disable
  - mmctl plugin enable
  - mmctl plugin install-url
  - mmctl plugin list
- mmctl plugin marketplace
  - mmctl plugin marketplace install
  - mmctl plugin marketplace list
- mmctl post
  - mmctl post create
  - mmctl post delete
  - mmctl post list
- mmctl roles
- mmctl saml
  - mmctl saml auth-data-reset
- mmctl sampledata
- mmctl system
  - mmctl system clearbusy
  - mmctl system getbusy
  - mmctl system setbusy
  - mmctl system status

    - mmctl system supportpacket
    - mmctl system version
- mmctl team
    - mmctl team archive
    - mmctl team create
    - mmctl team delete
    - mmctl team list
    - mmctl team modify
    - mmctl team rename
    - mmctl team restore
    - mmctl team search
- mmctl team users
    - mmctl team users add
    - mmctl team users remove
- mmctl token
    - mmctl token generate
    - mmctl token list
    - mmctl token revoke
- mmctl user
    - mmctl user activate
    - mmctl user change-password
    - mmctl user convert
    - mmctl user create
    - mmctl user deactivate
    - mmctl user delete
    - mmctl user deleteall
        - mmctl user demote
    - mmctl user email
    - mmctl user invite
    - mmctl user list
    - mmctl user migrate-auth
- mmctl user preference
    - mmctl user preference get

    - mmctl user preference list
    - mmctl user preference set
    - mmctl user preference delete
        - mmctl user promote
    - mmctl user reset-password
    - mmctl user resetmfa
    - mmctl user search
    - mmctl user username
    - mmctl user verify
- mmctl version
- mmctl webhook
    - mmctl webhook create-incoming
    - mmctl webhook create-outgoing
    - mmctl webhook delete
    - mmctl webhook list
    - mmctl webhook modify-incoming
    - mmctl webhook modify-outgoing
    - mmctl webhook show
- mmctl websocket
- mmctl usage notes
- mmctl commands
- Install mmctl
- Build mmctl
- Local mode
    - Activating local mode
    - Using local mode
- Running mmctl tests
- mmctl auth
    - mmctl auth clean
    - mmctl auth current
    - mmctl auth delete
    - mmctl auth list

    - mmctl auth login
        - Password
        - MFA
        - Access tokens
        - Installing shell completions
    - mmctl auth renew
    - mmctl auth set
- mmctl bot
    - mmctl bot assign
    - mmctl bot create
    - mmctl bot disable
    - mmctl bot enable
    - mmctl bot list
    - mmctl bot update
- mmctl channel
    - mmctl channel archive
    - mmctl channel create
    - mmctl channel delete
    - mmctl channel list
    - mmctl channel make-private
    - mmctl channel modify
    - mmctl channel move
    - mmctl channel rename
    - mmctl channel restore
    - mmctl channel search
    - mmctl channel unarchive
    - mmctl channel users
    - mmctl channel users add
    - mmctl channel users remove
- mmctl command
    - mmctl command archive
    - mmctl command create
    - mmctl command delete

    - mmctl command list
    - mmctl command modify
    - mmctl command move
    - mmctl command show
  - mmctl completion
    - mmctl completion bash
    - mmctl completion zsh
  - mmctl config
    - mmctl config edit
    - mmctl config export
    - mmctl config get
    - mmctl config migrate
    - mmctl config patch
    - mmctl config reload
    - mmctl config reset
    - mmctl config set
    - mmctl config show
    - mmctl config subpath
  - mmctl docs
  - mmctl export
    - mmctl export create
    - mmctl export delete
    - mmctl export download
    - mmctl export generate-presigned-url
    - mmctl export job
    - mmctl export job cancel
    - mmctl export job list
    - mmctl export job show
    - mmctl export list
  - mmctl group
  - mmctl group channel
    - mmctl group channel disable
    - mmctl group channel enable

    - mmctl group channel list
    - mmctl group channel status
    - mmctl group list-ldap
  - mmctl group team
    - mmctl group team disable
    - mmctl group team enable
    - mmctl group team list
    - mmctl group team status
  - mmctl group user
    - mmctl group user restore
  - mmctl extract
    - mmctl extract job
    - mmctl extract job list
    - mmctl extract job show
    - mmctl extract run
  - mmctl import
    - mmctl import delete
    - mmctl import job
    - mmctl import job list
    - mmctl import job show
    - mmctl import list
    - mmctl import list available
    - mmctl import list incomplete
    - mmctl import process
    - mmctl import upload
    - mmctl import validate
  - mmctl integrity
  - mmctl job
    - mmctl job list
    - mmctl job update
  - mmctl ldap
    - mmctl ldap idmigrate

- mmctl ldap job
  - mmctl ldap job list
  - mmctl ldap job show
- mmctl ldap sync
- mmctl license
  - mmctl license remove
  - mmctl license upload
  - mmctl license upload-string
- mmctl logs
- mmctl oauth
  - mmctl oauth list
- mmctl permissions
  - mmctl permissions add
  - mmctl permissions remove
  - mmctl permissions reset
  - mmctl permissions role assign
  - mmctl permissions role show
  - mmctl permissions role unassign
- mmctl plugin
  - mmctl plugin add
  - mmctl plugin delete
  - mmctl plugin disable
  - mmctl plugin enable
  - mmctl plugin install-url
  - mmctl plugin list
- mmctl plugin marketplace
  - mmctl plugin marketplace install
  - mmctl plugin marketplace list
- mmctl post
  - mmctl post create
  - mmctl post delete
  - mmctl post list
- mmctl roles

- mmctl saml
  - mmctl saml auth-data-reset
- mmctl sampledata
- mmctl system
  - mmctl system clearbusy
  - mmctl system getbusy
  - mmctl system setbusy
  - mmctl system status
  - mmctl system supportpacket
  - mmctl system version
- mmctl team
  - mmctl team archive
  - mmctl team create
  - mmctl team delete
  - mmctl team list
  - mmctl team modify
  - mmctl team rename
  - mmctl team restore
  - mmctl team search
- mmctl team users
  - mmctl team users add
  - mmctl team users remove
- mmctl token
  - mmctl token generate
  - mmctl token list
  - mmctl token revoke
- mmctl user
  - mmctl user activate
  - mmctl user change-password
  - mmctl user convert
  - mmctl user create
  - mmctl user deactivate
  - mmctl user delete

    - mmctl user deleteall
        - mmctl user demote
    - mmctl user email
    - mmctl user invite
    - mmctl user list
    - mmctl user migrate-auth
  - mmctl user preference
    - mmctl user preference get
    - mmctl user preference list
    - mmctl user preference set
    - mmctl user preference delete
        - mmctl user promote
    - mmctl user reset-password
    - mmctl user resetmfa
    - mmctl user search
    - mmctl user username
    - mmctl user verify
  - mmctl version
  - mmctl webhook
    - mmctl webhook create-incoming
    - mmctl webhook create-outgoing
    - mmctl webhook delete
    - mmctl webhook list
    - mmctl webhook modify-incoming
    - mmctl webhook modify-outgoing
    - mmctl webhook show
  - mmctl websocket

Product limits

This page describes some of the product limits that apply to Mattermost, including hard and recommended limits.

Feature	Hard Limit	Recommended Limit
Group messages	8 participants	If 8 isn't enough, create a private channel
Call participants	Unlimited	50
Custom brand text	500	
Custom site description	1024	
Advanced log output	500 MB file size, 1000 audit records	
Custom status	100 characters	
Channel name	64 characters	
Channel purpose	250 characters	
Channel header	1024 characters	
Bookmarked links & files	50	
Message acknowledgement response time	5 minutes	
Custom user group notifications	256 users per group	

Feature	Hard Limit	Recommended Limit
Emoji reactions per message	50	
Custom emoji uploads	6000	
Team description	50 characters	
SSO user session duration	30 days	
Mobile user session duration	30 days	
Attachments per message (files or images)	10	
Attachment file size	100 MB	Configurable
Attachment image file size	253 MB	Configurable
Attachment dimensions	7680 pixels x 4320 pixels	Configurable
Attachment image resolution size	33 MP (mega pixels) or 8K resolution	Configurable
User nickname	64 characters	
Message length	16,383 characters	

Feature	Hard Limit	Recommended Limit
Teams per deployment	10,000	
Users per team	No enforced limit	Dependent on infrastructure

Request a server Health Check

The Mattermost Health Check is a comprehensive evaluation of your system's current operational health available to Mattermost Enterprise customers.

This evaluation provides you with insights based on your system's configuration, user activity, and log events that helps you proactively identify issues, enhance collaboration security, optimize resource utilization and scalability which leads to minimized downtime, improved performance, and significant cost savings for your Enterprise organization.

> Tip
>
> Want to configure your own server health check probes for your self-hosted Mattermost deployment? See the configure server health check probes documentation for details.

Why request a Mattermost Health Check?

Regularly assessing the health of your Mattermost environment ensures that:

- System performance remains optimal, with minimal delays or downtime.
- User experience is consistently reliable, preventing missed messages or slow load times.
- Security and compliance settings are up-to-date and effective.
- Scalability aligns with your evolving usage patterns, supporting growth without issues.

Get started

Getting started with a Mattermost Health Check is simple and involves 3 steps:

1. Generate a Support Packet: The Mattermost Support Packet contains critical information about your Mattermost environment, including logs, configurations, and usage data.
2. Submit Your Support Packet: Once you've generated the Support Packet, submit it through our Support System as a standard support request (https://support.mattermost.com/hc/en-us/requests/new). Please include "Health Check Provided" in the subject line.
3. Receive Your Health Check Report: Within 2 weeks of submitting your Support Packet, you will receive a comprehensive performance report. This report will include:

- Detailed analysis of your current system health.
- Actionable recommendations to enhance performance, reliability, and security.
- Guidance on best practices for future system maintenance.

Get help

If you have any questions or need assistance, contact your Customer Success Manager or Technical Account Manager, the Mattermost Support Team (https://mattermost.com/support/), or visit the Mattermost Help Center (https://support.mattermost.com/).

On this page

- Request a server Health Check
  - Why request a Mattermost Health Check?
  - Get started
  - Get help

Statistics

Statistics on users, posts, and channels are tracked for each system and team. Enterprise Editions have access to advanced system statistics.

> Note
>
> To maximize performance for large Enterprise deployments, statistics for total messages, total hashtag messages, total file messages, messages per day, and activated users with messages per day is configurable by changing the `MaxUsersForStatistics` value in config.json.

For advanced metrics for Enterprise deployments, see performance monitoring documentation to learn more.

Site statistics

System statistics are viewable under **System Console > Reporting**. The data shown here is a cumulative sum across all teams on the system.

Total Users

: The total number of active accounts created on your system. Excludes deactivated accounts.

Total Teams

: The total number of teams created on your system.

Total Channels

: The total number of public channels and private channels created in all the teams on your system, including deleted channels. Doesn't include direct message channels.

Total Posts

The total number of posts made in all the teams on your system, including deleted posts and posts made using automation.

Daily Active Users

The total number of users who viewed the Mattermost site in the last 24 hours. Excludes bot users.

Monthly Active Users

The total number of users who viewed the Mattermost site in the last 30 days. Excludes bot users.

Total Posts (graph)

The total number of posts made on a certain day in all the teams on your system, including deleted posts and posts made using automation.

Total Posts from Bots (graph)

The total number of posts made by a bot account (https://developers.mattermost.com/integrate/admin-guide/admin-bot-accounts/) on a certain day in all the teams on your system, including deleted posts and posts made using automation.

Active Users with Posts (graph)

Users who made a post on a certain day in all the teams on your system, including system messages posted from the user's account.

Advanced system statistics

Note

Available only for self-hosted deployments

Mattermost Enterprise includes additional system statistics.

Total Sessions

The number of active user sessions connected to your system. Expired sessions are not counted.

Total Commands

The number of active slash commands currently set up on your system. Slash commands that are created and then removed in the **Integrations** menu are not counted.

Incoming Webhooks

The number of active incoming webhooks currently setup on your system. Incoming webhooks that are created and then removed in the **Integrations** menu are not counted.

Outgoing Webhooks

The number of active outgoing webhooks currently set up on your system. Outgoing webhooks that are created and then removed in the **Integrations** menu are not counted.

WebSocket Conns

The number of active WebSocket connections currently on your server.

Master DB Conns

The number of active connections currently on your master database.

Replica DB Conns

The number of active connections currently on one or more of your read replica databases.

Total Playbooks

The total number of collaborative playbooks on this server.

Total Playbook Runs

The total number of runs (active and complete) on this server.

Channel Types

This chart displays the number of public channels and private channels in a visual format, including channels that might have been deleted.

Posts, Files and Hashtags

This chart displays the number of posts containing files, hashtags, or only text. Posts containing both files and hashtags are counted in both categories, and deleted posts are included.

Team statistics

Team Statistics are viewable under **System Console > Team Statistics**. The data shown here is a cumulative sum across this team only, and excludes posts made in Direct Message channels, which are not tied to a team.

Total Users

> The total number of active accounts on this team. Excludes deactivated accounts.

Public Channels

> The number of public channels created in this team. Excludes deleted channels.

Private Channels

> The number of private channels created in this team. Excludes deleted channels.

Total Posts

> The total number of posts made in this team, including deleted posts and posts made using automation. Excludes posts made in Direct Message channels, which are not tied to a team.

Total Posts (graph)

> The total number of posts made on a certain day in this team, including deleted posts and posts made using automation.

Active Users with Posts (graph)

> Users who made a post on a certain day in this team, including system messages posted from the user's account.

Recent Active Users

> Twenty most recent users who have logged in and had recent browser activity in Mattermost.

Newly Created Users

> Most recent users who have joined the team.

Troubleshooting/FAQ

I see an error: “Not enough data for a meaningful representation”

If the statistics page is loading endlessly and you get an error message saying “Not enough data for a meaningful representation”, check whether you’re using an ad blocker. An ad blocker can prevent this page from loading data. To test this, temporarily disable your ad blocker, or view the page in a browser without an ad blocker installed.

Can team admins review their own team’s statistics?

Note

Available only for self-hosted deployments

Yes. In Mattermost Enterprise, you can enable team admins to see their team’s statistics by modifying available delegated granular administration system roles. See the delegated granular administration documentation to learn more about these admin roles, including how to manage privileges and assign roles.

To enable team admins to access their team’s statistics:

1. Go to **System Console > User Management > System Roles**, then edit the **Viewer** system admin role.

2. Under **Privileges**, expand the **Reporting** section, set **Team Statistics** to **Read only**, then set **Site Statistics** and **Server Logs** to **No access**.

3. Set all other privileges to **No access** to restrict all users with the **Viewer** role to access only the **Team Statistics** page in the System Console.

4. Under **Assigned People**, select **Add People** to assign team admins to the **Viewer** role, and enable them to access their team's statistics.

5. Select **Save**.

> Note
>
> System admins must manually add people to or remove people from the **Viewer** system admin role to address team admin changes, such as promotions or demotions.

On this page

- Statistics
  - Site statistics
    - Advanced system statistics
  - Team statistics
  - Troubleshooting/FAQ
    - I see an error: “Not enough data for a meaningful representation”
    - Can team admins review their own team’s statistics?

System-wide notifications

System admins can configure system-wide notifications visibile in all channels for all users across all teams.

1. Enable system-wide notifications by going to **System Console > Site Configuration > System-wide notifications**, and setting **Enable system-wide notifications** to **true**.
2. In the **Banner Text** field, enter the notification text you want to share in channels.
3. Specify the background and text colors for the banner.
4. By default, users can dismiss the banner until they log in again or until you update the banner. To prevent users from dismissing the banner, set **Allow Banner Dismissal** to **false**.
5. Select **Save**.

Update the banner by either changing the text of the banner or by re-enabling the banner after it's been disabled.

To disable the banner, set **Enable Announcement Banner** to **false**.

Manage team and channel members

System admins can manage channel configuration in the System Console, including:

- **Management:** Manage synchronization, moderation, and membership settings.
- **Promoting/demoting Team and Channel Admins:** Team admins and channel admins can be demoted via the System Console.
- **View channel members:** System admins can view members of a channel without having to join the channel.
- **Member count:** Total member count in a team or channel.
- **Archive channels:** System admins can archive channels without having to join the channel.

Teams

To view and manage team information, navigate to **System Console > Teams**. Teams can be managed by **Group Sync** or **Invite Only**. The type of management in use may affect the management options available for that team.

Select a team to view its configuration options.

Team profile

The name and description of the team.

System admins can archive or unarchive the team from within **User Management > Teams > Team Management**. Archiving a team makes its contents inaccessible for all users. All related information is archived, including posts from the database. Before archiving a team, we recommend that you perform a database backup.

Archive a team

Select **Archive Team**, then select **Save**. Select **Archive** when prompted to confirm the team archive.

Alternatively, system admins can use the mmctl `mmctl team archive` to archive teams. See the mmctl product documentation for details.

Unarchive a team

Select **Unarchive Team**, then select **Save**.

Alternatively, system admins can use the mmctl `mmctl team restore` to archive teams. See the mmctl product documentation for details.

Team management

Note

- When **Sync Group Members** is enabled, the **Synced Groups** list is visible and additional groups can be added.
- When **Sync Group Members** is not enabled, invitation limitations can be selected.

Groups

Note

You can add and remove groups, as well as promote or demote group members to team admin/member roles.

Members

A list of all members in a channel is visible to system admins. Members can be added and removed from the team members and be promoted or demoted to a team admin/member role. Use the role **Filter** to refine your search results. You can use one filter, or combine filters to search by multiple roles:

- Guest
- Member

- Team admin
- System admin

Channels

To view and manage channel information, navigate to **System Console > Channels**. Channels can be managed by **Group Sync** or **Manual Invites**. The type of management in use may affect the management options available for that channel. Use the **Filter** to refine your search results. You can use one filter, or combine filters to search by channel and management type:

- Public channels
- Private channels
- Archived channels
- Channels that are synchronized to an AD/LDAP Group
- Channels with users who were invited manually (not synced from LDAP)

Select a channel to view its configuration options.

Profile

The name and description of the channel. To archive the channel, select **Archive Channel > Save**. The channel is still searchable in the **Channels** list. To unarchive the channel, select **Unarchive Channel** and **Save**.

Channel management

Note

- When **Sync Group Members** is enabled, the **Synced Groups** list is visible and additional groups can be added.
- When **Sync Group Members** is not enabled, you can select whether the channel is **Private** or **Public**.

Advanced access controls

Note

Advanced access control settings enable system admins to restrict actions within specific channels. These actions include:

- **Make channel read-only:** Read-only channels enable system admins to turn off posting in specified channels.
- **Restrict reactions:** Turn off the ability for members and guests to post reactions.
- **Restrict channel mentions:** Turn off the ability for users to post channel wide mentions (@all/@channel/@here) in specified channels.
- **Channel member management:** Only admins have the ability to add and remove channel members in the specified channels.
- **Bookmarks management:** Only channel and system admins can manage channel bookmarks.

These settings are modified in **System Console > User Management > Channels**.

> Note
>
> These settings are applicable only to Guests and Members. System, team, and channel admins aren't affected. If you wish to grant posting ability to a specific member, you must first promote that member to channel admin.

The availability of advanced access control settings can also be affected by existing system and team permissions configurations. If there are existing configurations that override the channel settings you would like to apply, it will be indicated in the user interface. These settings can then be adjusted in the relevant panel in the **Permissions** section of the System Console.

Configure a channel so that members can post/reply/react but guests can only read and react.

1. Navigate to **System Console > User Management > Channels**.
2. Select **Edit** next to the name of the channel you want to configure.
3. In the **Create Posts** panel, uncheck **Guests**.
4. In the **Post Reactions** panel, uncheck **Guests** if required.
5. Select **Save**.

The channel is available for all members and guests to access, but guests can only read messages and react to them.

Create an Announcement Channel where only Channel Admins are able to post (read-only).

1. Create a new channel (either Public or Private).
2. Navigate to **System Console > User Management > Channels**.
3. Select **Edit** next to the name of the channel you just created (you may need to search for it).
4. In the **Create Posts** panel, uncheck **Guests** and **Members**.
5. In the **Post Reactions** panel, uncheck **Guests** and **Members**.
6. Select **Save**.

The channel is available for all members and guests to access but only Admins can post to the channel.

Groups

Note

You can add and remove groups, as well as promote or demote group members to team admin/member roles.

Members

A list of all members in a channel is visible to system admins. Members can be added and removed from the team members and be promoted or demoted to a team admin/member role. Use the role **Filter** to refine your search results. You can use one filter, or combine filters to search by multiple roles:

- Guest
- Member
- Channel admin
- System admin

On this page

- Manage team and channel members
  - Teams
    - Team profile
    - Archive a team
    - Unarchive a team
    - Team management
    - Groups
    - Members
  - Channels
    - Profile
    - Channel management
    - Advanced access controls
    - Groups
    - Members

Telemetry

As described in the privacy policy in each Mattermost server, telemetry data optionally shared from your Mattermost servers is used to identify security and reliability issues, to analyze and fix software problems, to help improve the quality of Mattermost software and related services, and to make design decisions for future releases.

Telemetry data is encrypted in transit, does not include personally identifiable information or message contents, and details of how the information is used and processed is available in our Privacy Policy (https://mattermost.com/privacy-policy/).

We use the data for the following purposes to:

- Identify security and reliability issues.
- Analyze and fix software problems.
- Help improve the quality of Mattermost software and related services.
- Make design decisions for future releases.

> Note
>
> Telemetry data collection is enabled by default for all Mattermost deployments. Self-hosted system admins can opt out of sharing telemetry data from Mattermost self-hosted servers within the System Console. Cloud system admins can't disable telemetry for Mattermost Cloud deployments.

Security update check feature

New threats to system security constantly arise. To alert you of relevant, high priority security updates, Mattermost servers are configured to share

diagnostic information with Mattermost Inc. so that we can provide appropriate alerts.

The following data is collected once every 24 hours:

- Mattermost server build number and version
- Type of build (Enterprise Edition or Team Edition)
- Server operating system
- The server diagnostic ID (same as the ID accessing the push notification proxy, and is used to prevent double-counting of telemetry data)
- Database type
- Database version
- Number of teams
- Number of users
- Number of activated users
- Whether or not the unit tests have been run
- Date and time of the last check for security updates
- The location of the Amazon Cloudfront server used for telemetry data

Opt out

To opt out, you can disable this security update check feature for self-hosted deployments in the System Console by going to **Environment > SMTP > Enable Security Alerts**. See the enable security alerts documentation for details. When this feature is disabled, you will not receive any security alerts.

Error and diagnostics reporting feature

Mattermost error and diagnostic data is collected for the following purposes:

- To add improvements that are specific to your usage, upgrade, and deployment patterns, including identifying security and reliability issues.

- To analyze and fix software problems.
- To help improve the quality of Mattermost software and related services.
- To make design decisions for future releases.

> Note
>
> Error and diagnostic reporting is sent by Mattermost to the endpoint `https://pdat.matterlytics.com`, a custom Rudder domain. When this feature is enabled, any 500 errors will automatically be sent to the Mattermost-hosted [Sentry (https://sentry.io/welcome/)](https://sentry.io/welcome/) endpoint.

Opt out

To opt out, you can disable the error and diagnostics reporting feature for self-hosted deployments in the System Console by going to **Environment > Logging > Enable Diagnostics and Error Reporting**. See the enable diagnostics and error reporting documentation for details.

Deployment and server configuration data

Reporting frequency

- When starting the server for the first time: Every 10 minutes for the first hour, then every hour for the first 12 hours.
- At the 24 hour mark and every 24 hours thereafter.

Deployment configuration information

> *Basic information including Mattermost server version, database and operating system type and version, and count of system admin accounts.*

Deployment type

- Manual install (includes `wget` installs)
- Docker
- Mattermost Omnibus

- Kubernetes operator
- GitLab Omnibus

Server Configuration Settings

Non-personally identifiable data from configuration settings file (`config.json`) in the form of `type` ("enumerated integer" or "enumerated boolean") values, `true/false` ("boolean"), and `count` ("integer"). Specifically these include:

Type values (enumerated integer and enumerated boolean)

ServiceSettings: enum WebserverMode, bool EnableSecurityFixAlert, bool EnableInsecureOutgoingConnections, bool EnableIncomingWebhooks, bool EnableOutgoingWebhooks, bool EnableCommands, bool EnableDeveloper, bool EnableOnlyAdminIntegrations, bool EnablePostUsernameOverride, bool EnablePostIconOverride, bool EnableCustomEmoji, enum RestrictCustomEmojiCreation, bool EnableTesting, bool DeveloperFlags, bool EnableClientPerformanceDebugging, bool EnableMultifactorAuthentication, bool EnableOAuthServiceProvider, enum OutgoingIntegrationRequestsDefaultTimeout, enum ConnectionSecurity, bool UseLetsEncrypt, bool Forward80To443, enum ConnectionSecurity, bool TLSStrictTransport, bool EnforceMultifactorAuthentication, bool EnableUserTypingMessages, bool TimeBetweenUserTypingUpdatesMilliseconds, bool EnablePostSearch, bool EnableUserStatuses, bool EnableChannelViewMessages, bool EnableEmojiPicker, bool EnableGifPicker, bool EnableAuthenticationTransfer, enum TeammateNameDisplay, bool EnableUserAccessTokens, enum MaximumLoginAttempts, bool ExtendSessionLengthWithActivity, enum SessionLengthWebInHours, enum SessionLengthMobileInHours, enum SessionLengthSSOInHours, int SessionCacheInMinutes, enum SessionIdleTimeoutInMinutes, enum TimeBetweenUserTypingUpdatesMilliseconds, enum ClusterLogTimeoutMilliseconds, bool CloseUnusedDirectMessages, bool EnablePreviewFeatures, bool EnableTutorial, bool EnableOnboarding, bool ExperimentalEnableDefaultChannelLeaveJoinMessages, bool ExperimentalGroupUnreadChannels, bool AllowCookiesForSubdomains, bool EnableAPITeamDeletion, bool EnableAPITriggerAdminNotifications, bool EnableAPIUserDeletion, bool EnableAPIChannelDeletion, bool

ExperimentalEnableHardenedMode, bool DisableLegacyMFA, bool ExperimentalStrictCSRFEnforcement, bool EnableEmailInvitations, bool ExperimentalChannelOrganization, bool EnableLegacySidebar, bool CorsAllowCredentials, bool CorsDebug, bool DisableBotsWhenOwnerIsDeactivated, bool EnableBotAccountCreation, bool RestrictLinkPreviews, bool EnablePermalinkPreviews, bool EnableSVGs, bool EnableLatex, bool EnableInlineLatex, bool Directory, bool RetentionDays, bool EnableLocalMode; **TeamSettings**: bool EnableUserCreation, bool EnableTeamCreation, bool RestrictTeamNames, bool EnableOpenServer, bool EnableUserDeactivation, bool EnableCustomBrand, bool RestrictDirectMessage, enum MaxNotificationsPerChannel, bool EnableConfirmNotificationsToChannel; enum MaxUsersPerTeam, enum MaxChannelsPerTeam, bool EnableJoinLeaveMessageByDefault, bool EnableCustomUserStatuses, bool EnableLastActiveTime, bool RefreshPostStatsRunTime, bool ExperimentalTownSquareIsReadOnly, bool ExperimentalHideTownSquareinLHS, bool EnableXToLeaveChannelsFromLHS, bool ExperimentalEnableAutomaticReplies, bool ExperimentalViewArchivedChannels, bool LockTeammateNameDisplay, bool MaxFieldSize; **ClientRequirementSettings**: enum AndroidLatestVersion; **GuestAccountsSettings**: bool Enable, bool AllowEmailAccounts, bool EnforceMultifactorAuthentication; **SqlSettings**: enum DriverName, bool Trace, enum ConnMaxIdleTimeMilliseconds, bool ConnMaxLifetimeMilliseconds; enum MaxOpenConns, enum QueryTimeout, bool DisableDatabaseSearch; **LogSettings**: bool EnableConsole, enum ConsoleLevel, bool ConsoleJson, bool EnableFile, enum FileLevel, bool FileJson, bool EnableWebhookDebugging; **NotificationLogSettings**: bool EnableConsole, bool ConsoleLevel, bool ConsoleJson, bool EnableFile, bool FileLevel, bool FileJson **PasswordSettings**: bool Lowercase, bool Number, bool Uppercase, bool Symbol, enum MinimumLength; **FileSettings**: bool EnablePublicLink, enum DriverName, enum MaxFileSize, enum FileSettings.MaxImageResolution, enum MaxImageDecoderConcurrency, bool FileSettings.ExtractContent, bool FileSettings.ArchiveRecursion, bool AmazonS3SSL, bool AmazonS3SignV2, bool AmazonS3SSE, bool AmazonS3Trace, bool MaximumPayloadSizeBytes, bool MaximumPayloadSizeBytes, bool EnableFileAttachments, bool EnableMobileUpload, bool EnableMobileDownload; **EmailSettings**: bool

EnableSignUpWithEmail, bool EnableSignInWithEmail, bool EnableSignInWithUsername, bool RequireEmailVerification, bool SendEmailNotifications, bool UseChannelInEmailNotifications, bool EmailNotificationContentsType, bool EnableSMTPAuth, enum ConnectionSecurity, bool SendPushNotifications, enum PushNotificationContents, bool EnableEmailBatching, bool SkipServerCertificateVerification, enum EmailBatchingBufferSize, enum EmailBatchingInterval, bool EnablePreviewModeBanner, enum SMTPServerTimeout; **MessageExportSettings**: bool DownloadExportResults; **RateLimitSettings**: bool EnableRateLimiter, bool VaryByRemoteAddr, bool VaryByUser, enum PerSec, enum MaxBurst, enum MemoryStoreSize; **PrivacySettings**: bool ShowEmailAddress, bool ShowFullName; **ThemeSettings**: bool EnableThemeSelection, bool AllowCustomThemes; **GitLabSettings**: bool Enable; **GoogleSettings**: bool Enable; **Office365Settings**: bool Enable; **SupportSettings**: bool CustomTermsOfServiceEnabled, enum CustomTermsOfServiceReAcceptancePeriod, enum ReportAProblemType; **LdapSettings**: bool Enable, bool EnableSync, enum ConnectionSecurity, bool SkipCertificateVerification, enum SyncIntervalMinutes, enum QueryTimeout, enum MaxPageSize, bool EnableAdminFilter; **ComplianceSettings**: bool Enable, bool EnableDaily; **LocalizationSettings**: enum DefaultServerLocale, enum DefaultClientLocale, enum AvailableLocales; **SamlSettings**: bool Enable, bool EnableSyncWithLdap, bool IgnoreGuestsLdapSync, bool EnableSyncWithLdapIncludeAuth, bool Verify, bool Encrypt, bool SignRequest, bool EnableAdminFilter; **ClusterSettings**: bool Enable, bool UseIpAddress, bool ReadOnlyConfig, bool EnableExperimentalGossipEncryption, bool EnableGossipCompression; **MetricsSettings**: bool Enable, bool EnableClientMetrics, bool EnableNotificationMetrics, enum BlockProfileRate; **WebrtcSettings** (only in v5.5 and earlier): bool Enable; **ExperimentalSettings** bool ClientSideCertEnable, bool EnablePostMetadata, bool LinkMetadataTimeoutMilliseconds, bool EnableClickToReply, bool RestrictSystemAdmin, bool CloudBilling, bool AllowSyncedDrafts, bool YoutubeReferrerPolicy; **AnnouncementSettings**: bool EnableBanner, bool AllowBannerDismissal, bool AdminNoticesEnabled, bool UserNoticesEnabled; **ElasticsearchSettings**: bool EnableIndexing, bool EnableSearching, bool Sniff, enum PostIndexReplicas, enum PostIndexShards, enum LiveIndexingBatchSize, enum BatchSize, bool

SkipTLSVerification, bool Trace; **PluginSettings**: bool Enable, bool EnableUploads, bool EnableHealthCheck, bool EnableMarketplace, bool EnableRemoteMarketplace, bool AutomaticPrepackagedPlugins, bool RequirePluginSignature; **DataRetentionSettings**: bool EnableMessageDeletion, bool MessageRetentionHours, bool AllowInsecureDownloadUrl, bool EnableFileDeletion, bool FileRetentionHours, enum DeletionJobStartTime; **MessageExportSettings**: bool EnableExport, enum ExportFormat, enum DailyRunTime, enum ExportFromTimestamp, enum BatchSize, enum GlobalRelaySettings.CustomerType; **ExperimentalAuditSettings**: bool SysLogEnabled, bool SysLogInsecure, enum SysLogMaxQueueSize, bool FileEnabled, enum FileMaxSizeMB, enum FileMaxAgeDays, bool FileMaxBackups, bool FileCompress, enum FileMaxQueueSize; **BleveSettings**: bool EnableIndexing, bool EnableSearching, bool EnableAutocomplete, enum BatchSize; bool FeatureFlags

Counts (integer)

> ***SqlSettings****: int DataSourceReplicas, int DataSourceSearchReplicas, int ReplicaLagSettings;* ***ThemeSettings****: int AllowedThemes;* ***PluginSettings****: int SignaturePublicKeyFiles*

True/false (boolean) value whether setting remains default (true) or non-default (false). **NOTE: No input data is used**:

> ***ServiceSettings****: bool SiteURL, bool WebsocketURL, bool TLSCertFile, bool TLSKeyFile, bool ReadTimeout, bool WriteTimeout,bool IdleTimeout, bool GoogleDeveloperKey, bool AllowCorsFrom, bool CorsExposedHeaders, bool AllowedUntrustedInternalConnections, bool ManagedResourcePaths, bool CollapsedThreads, bool PostPriority, bool AllowPersistentNotifications, bool PersistentNotificationMaxCount, bool PersistentNotificationIntervalMinutes, bool PersistentNotificationMaxRecipients;* ***TeamSettings****: bool SiteName, bool CustomBrandText, bool CustomDescriptionText, bool UserStatusAwayTimeout, bool ExperimentalPrimaryTeam;* ***DisplaySettings****: bool CustomUrlSchemes, bool MaxMarkdownNodes;* ***GuestAccountSettings****: bool RestrictCreationToDomains, bool EnforceMultifactorAuthentication, bool HideTags;* ***LogSettings****: bool FileLocation;* ***NotificationLogSettings****: bool FileLocation;* ***EmailSettings****: bool FeedbackName, bool FeedbackEmail, bool*

> *FeedbackOrganization, bool LoginButtonColor, bool LoginButtonBorderColor, bool LoginButtonTextColor, bool ImageProxyType, bool ImageProxyURL, bool ImageProxyOptions;* ***RateLimitSettings****: bool VaryByHeader;* ***SupportSettings****: bool TermsOfServiceLink, bool PrivacyPolicyLink, bool AboutLink, bool HelpLink, bool ReportAProblemLink, bool AllowDownloadLogs, bool AppCustomURLSchemes, bool MobileExternalBrowser bool SupportEmail;* ***ThemeSettings****: bool DefaultTheme;* ***LdapSettings****: bool FirstNameAttribute, bool LastNameAttribute, bool EmailAttribute, bool UserNameAttribute, bool NicknameAttribute, bool IdAttribute, bool PositionAttribute, bool LoginFieldName, bool LoginButtonColor, bool LoginButtonBorderColor, bool LoginButtonTextColor, bool GroupFilter, bool GroupDisplayNameAttribute, bool GroupIdAttribute, bool GuestFilter, bool AdminFilter;* ***SamlSettings****: bool SignatureAlgorithm, bool CanonicalAlgorithm, bool ScopingIDPProviderId, bool ScopingIDPName, bool IdAttribute, bool GuestAttribute, bool FirstNameAttribute, bool LastNameAttribute, bool EmailAttribute, bool UserNameAttribute, bool NicknameAttribute, bool LocaleAttribute, bool PositionAttribute, bool LoginIdAttribute, bool LoginButtonText, bool LoginButtonColor, bool LoginButtonBorderColor, bool LoginButtonTextColor, bool AdminFilter;* ***NativeAppSettings****: bool AppDownloadLink, bool AndroidAppDownloadLink, bool IosAppDownloadLink;* ***WebrtcSettings*** *(only in v5.5 and earlier): bool StunURI, bool TurnURI;* ***ClusterSettings****: bool NetworkInterface, bool BindAddress, bool AdvertiseAddress;* ***MetricsSettings****: bool BlockProfileRate;* ***AnalyticsSettings****: bool MaxUsersForStatistics;* ***ExperimentalSettings*** *bool ClientSideCertCheck;* ***AnnouncementSettings****: bool BannerColor, bool BannerTextColor;* ***ElasticsearchSettings****: bool ConnectionUrl, bool Username, bool Password, bool IndexPrefix;* ***PluginSettings****: bool MarketplaceUrl, bool SignaturePublicKeyFiles, bool ChimeraOAuthProxyUrl;* ***MessageExportSettings****: bool GlobalRelaySettings.SmtpUsername, bool GlobalRelaySettings.SmtpPassword, bool GlobalRelaySettings.EmailAddress;* ***ConnectedWorkspacesSettings****: bool EnableSharedChannels, bool EnableRemoteClusterService, bool DisableSharedChannelsStatusSync, bool DefaultMaxPostsPerSync.*

Commercial License Information (Enterprise Edition only)

Information about commercial license key purchased or trial license key used for Enterprise Edition servers: Company ID, license ID, license issue date, license start date, license expiry date, number of licensed users, license name, list of unlocked subscription features.

Advanced Access Controls Configuration Information (Enterprise Edition only)

Information related to channel moderation, including number of channel schemes, number of channels with posting messages disabled for users or guests, number of channels with emoji reactions disabled for users or guests, number of channels with managing members disabled, number of channels with channel mentions disabled for users or guests.

Channel Member Management Information (Enterprise Edition only)

Information related to bulk user management and team and channel filtering, including number of users added, number of users removed, number of users promoted, number of users demoted, number of times archive and unarchive is used from any channel configuration page, and number of times channel search or team search filters are used.

Groups Configuration Information (Enterprise Edition only)

Information related to AD/LDAP groups, including number of groups synced to Mattermost, teams and channels associated to groups, teams and channels synced with groups, and number of group members.

Plugin Configuration Information

Basic information including number of active and inactive plugins, which are using webapp or backend portions, which Mattermost plugins are enabled along with their versions, and core plugins disabled count. Some plugins may send summary data such as number of authenticated users of the plugin. The list of plugins is obtained from the Marketplace. If the Marketplace can’t be reached, the list of known plugins is used instead.

Permissions Configuration Information (Enterprise Edition only)

Permissions configured for each role for the System Scheme and each Team Override Scheme created in the system. Scheme ID; team admin permissions; team user permissions; channel admin permissions; channel user permissions; number of teams the scheme is associated with; number of users assigned to each admin role; Number of admin

roles not using default privileges; Changes to default privileges of each admin role.

Aggregated Usage Statistics

Non-personally identifiable summations of basic usage statistics: Number of enabled and disabled accounts, number of user logins in the last 24 hours and the last 30 days, number of users active in the last day/month, whether APIv3 endpoints were used in the last 24 hours, number of posts, channels, teams, guest accounts, bots, and file storage.

True Up Diagnostics

Requested help from sales with license true up; attempted to download true up packet.

Event data

Reporting Frequency

- Immediately after the specific event occurs.

> Note
>
> The majority of these events have been disabled. Refer to the source file for the current list of events sent via telemetry (https://github.com/mattermost/mattermost-redux/blob/master/src/client/client4.ts#L3069).

Non-personally Identifiable Error Information, distinguished by end users and system admins

Boolean when the following events occur:

- *Sign-in Error*: Email login error, AD/LDAP login error, SAML login error

Boolean when the following events occur, including the error message, recently dispatched Redux actions, and non-identifiable information of the device, operating system, and the app:

- *Mobile App Errors*: App crashes caused by type errors, exceptions, and failed logins

Non-personally Identifiable Diagnostic Information, distinguished by end users and system admins

Boolean when the following events occur:

- *Team and Account Setup Diagnostics:* Account creation via email, invite or UI, account creation page view, account creation completion; tutorial step and tip completion or opt out, team creation page view, team name and URL entry, team creation completion, clicks on all form elements, buttons, textboxes and links on sign up page, team selection page, and team creation pages
- *Sign-in Diagnostics:* Login succeeded or failed for email, LDAP, or SAML/SSO; logout succeeded; switched authentication method from email to LDAP or SAML/SSO or vice versa; reset password; updated password
- *Navigation Discovery Diagnostics:* Joined a channel from the "More" list, through an invite or by clicking a public link; created a channel, direct, or group direct message conversation; renamed, joined, left or deleted an existing channel; updated header or purpose; added or removed members; updated channel notification preferences; loaded more messages in a channel; switched a channel or a team; opened the "More" modal for channels or direct message conversations; updated team name; invited members; updated profile and Channels settings
- *Core Feature Discovery Diagnostics:* Created, edited or deleted a message; posted a message containing a hashtag, link, mention or file attachment; searched for a term; searched for saved posts or recent mentions
- *Advanced Feature Discovery Diagnostics:* Reacted to a message; favorited or unfavorited a channel; saved or unsaved a message; pinned or unpinned a message; replied to a message; expanded the right-hand sidebar; started or finished a WebRTC video call (only in v5.5 and earlier); created or deleted a personal access token; added or removed post:all or post:channels permission; created a category in the sidebar

- *Integration Discovery Diagnostics:* Created or triggered a webhook or slash command; created, authorized or deleted an OAuth 2.0 app; created, posted, or deleted a custom emoji
- *Plugin Discovery Diagnostics:* Number of installed plugins containing either server or webapp portions, or both; number of those plugins being activated
- *Plugin Marketplace Diagnostics:* Plugin ID, current version, and target version for all install and update events. Only sent when the default Marketplace is configured
- *Plugin telemetry:* Search terms used in Marketplace on cloud workspaces will be recorded
- *Commercial License Diagnostics (Enterprise Edition only):* Uploaded an Enterprise license key to the server
- *Mobile Performance Diagnostics:* Load times for starting the app, switching channels, and switching teams
- *Permissions Discovery Diagnostics (Enterprise Edition only):* Provides all the permissions configured for each role for the System Scheme and each Team Override Scheme created in the system. Scheme ID; team admin permissions; Team user permissions; channel admin permissions; Channel user permissions; Number of teams the scheme is associated with
- *Group Discovery Diagnostics:* Provides information related to AD/LDAP (Enterprise Edition only) and custom groups (Enterprise and Professional Edition only), including number of unique users in groups, number of groups synchronized to Mattermost, teams and channels associated to groups, teams and channels synchronized with groups, number of group members, custom group @mentions, changes to existing custom groups, and new custom groups created.
- *System Console Menu Discovery Diagnostics:* Clicks on the hamburger menu items of the System Console, including Administrator's Guide, Troubleshooting Forum, Commercial Support, About Mattermost, and clicks on the left-hand side navigation menu items
- *In Product Notices Diagnostics:* Notices viewed, and the notices on which an action button was clicked.

- *Threaded discussions:* Clicks to reply to a thread, reply using the footer element, filter threads by unread, mark as read, access to global threads section.
- *Custom Groups:* Invite people to a channel by using a custom group, mention a custom group, and modify a custom group.
- *Read-Only Channels:* Navigate to a read-only channel, post a message to a read-only channel, and open a read-only channel.
- *Shared Workspaces:* Navigate to a shared channel, post a message in a shared channel, and mention a remote user.
- *Guest Accounts:* Mention a guest account, directly message a guest, and add a guest to a channel.
- *Passive Keyword Tracking*: Update passive keywords highlighted in channels and threads.

Playbooks telemetry

Collaborative playbooks metadata is collected and sent every 24 hours. Visit the [playbooks telemetry file (https://github.com/mattermost/mattermost-plugin-playbooks/blob/master/server/telemetry/rudder.go)](https://github.com/mattermost/mattermost-plugin-playbooks/blob/master/server/telemetry/rudder.go) for details about the types of metadata collected.

Android Mobile App performance monitoring

To improve Android app performance, we are collecting trace events and device information, collectively known as metrics, to identify slow performing key areas. Those metrics will be sent only from users using the Android app Beta build starting in version v1.20, who are logged in to servers that allow sending diagnostic information.

Trace events

> Includes duration on how long the action took place like startup, team/channel switch, posts loading/update and channel drawer open/close. The naming convention is interpreted as `[start observation]:[end observation]`, e.g. `start:overall` as from app start until fully rendered or

`post_list:thread` as on press of post at post list until thread is opened. Complete list of trace events are the following:

- start:overall
- start:process_packages
- start:content_appeared
- start:select_server_screen
- start:channel_screen
- team:switch
- channel:loading
- channel:switch_loaded
- channel:switch_initial
- channel:close_drawer
- channel:open_drawer
- posts:loading
- post_list:thread
- post_list:permalink

Device information

The information being collected is non-personally identifiable. Except for system_version, device information is based from [react-native-device-info (https://github.com/mattermost/react-native-device-info#react-native-device-info)](https://github.com/mattermost/react-native-device-info#react-native-device-info) library. Refer to the linked documentation to learn more. Complete list of device information are the following:

- api_level
- build_number
- bundle_id
- brand

- country
- device_id
- device_locale
- device_type
- device_unique_id
- height
- is_emulator
- is_tablet
- manufacturer
- max_memory
- model
- server_version
- system_name
- system_version
- timezone
- version
- width

On this page

- Telemetry
  - Security update check feature
    - Opt out
  - Error and diagnostics reporting feature
    - Opt out
    - Deployment and server configuration data
    - Event data
  - Playbooks telemetry

- Android Mobile App performance monitoring

User satisfaction surveys

Feedback is used to measure user satisfaction and improve product quality by hearing directly from users. Please refer to our privacy policy (https://mattermost.com/privacy-policy/) for more information on the collection and use of information received through our services.

> Tip
>
> If you're looking for survey functionality for your self-hosted Mattermost instance that offers more customization options, stores results data locally on your Mattermost server, and doesn't send telemetry data back to Mattermost, we recommend using the User Survey integration instead, and disabling the user satisfaction survey

Administration

Is the survey enabled by default?

The user satisfaction survey is a pre-packaged plugin, and surveys are enabled by default on all servers. However, the plugin will not be activated on any servers that have Error and Diagnostic Reporting disabled, meaning no surveys or data collection occurs.

How can surveys be disabled?

Disabling the **User Satisfaction Surveys** plugin from **System Console > Plugins > Plugin Management** will disable surveys and all data collection by the plugin.

> Note
>
> - If surveys have been disabled from the plugin configuration in **System Console > Plugins > User Satisfaction Surveys**, but the plugin itself is still enabled, surveys won't be scheduled but users can still send written feedback by messaging Surveybot.
> - When the plugin or surveys in the plugin configuration are disabled, they remain disabled for subsequent server upgrades.

When is the survey scheduled?

Users will receive surveys 21 days after every server upgrade, assuming the following conditions are true:

- User Satisfaction Surveys plugin is enabled in **System Console > Plugins > Plugin Management**.
- Surveys are enabled in the plugin configuration in **System Console > Plugins > User Satisfaction Surveys**.
- User account is greater than 21 days old.
- User has not completed a survey in the last 90 days.
- User has not been sent a survey in the last 90 days.
- Current server version is greater than the server version of the last survey, excluding dot releases.

The above conditions mean that at maximum frequency a user will receive a survey every 90 days, assuming the server is upgraded within that time period.

How will I be notified when a survey is scheduled?

System admins will receive an email notification and in-product direct message from "Surveybot" mentioning the scheduled date the survey will be triggered.

Survey Data

How is the survey received?

Once the survey is triggered on the server, all users will receive an in-product direct messages from "Surveybot" on their next login or page refresh in Mattermost.

Users can optionally select a 0-10 score on how likely they are to recommend Mattermost and then provide written feedback about their experience. Selecting a score and providing feedback are optional, and the survey can be ignored without interrupting usage of Mattermost.

What data is collected?

Data is only collected when a user selects a score or provides written feedback in response to survey questions. Please refer to our [privacy policy (https://mattermost.com/privacy-policy/)](https://mattermost.com/privacy-policy/) for more information on the collection and use of information received through our services. The following **non-personally identifiable information** is collected:

- Survey information:
    - Score (0-10) submitted by the user
    - Written feedback submitted by the user (if applicable)

    - Timestamp of the survey submission
- Server information:
    - Server/Web App version the survey was submitted on
    - Installation date of the server
    - Diagnostic ID used for error and diagnostics reporting
    - License ID used for error and diagnostics reporting (if applicable)
    - Enterprise or Professional (if applicable)
- User information:
    - User role (system admin, team admin, or member)
    - Account creation timestamp
    - User ID of the surveyed user

Will this data be sent through my firewall?

Note

Available only for self-hosted deployments

If Mattermost is hosted in a private network with firewall then data from the User Satisfaction Surveys plugin is not sent unless outbound connections are allowed or specifically configured for this plugin.

On this page

- User satisfaction surveys
    - Administration
        - Is the survey enabled by default?
        - How can surveys be disabled?
        - When is the survey scheduled?
        - How will I be notified when a survey is scheduled?
    - Survey Data
        - How is the survey received?

- What data is collected?
- Will this data be sent through my firewall?

AD/LDAP groups

Overview

The groups feature is useful for organizations that have many new users to onboard or that onboard users frequently and want to ensure users are added to default teams and channels that are pertinent to them. The group feature currently supports:

- Creating groups by synchronization with your AD/LDAP system groups.
- Syncing groups to pre-defined roles in Mattermost.
- AD/LDAP nested groups.
- Using synchronized groups to manage membership of teams and Private channels.

For a technical overview of the feature by Martin Kraft, who led the development of the feature, please see [this blog post (https://developers.mattermost.com/blog/ldap-nested-groups-modelling-and-representation-in-code)](https://developers.mattermost.com/blog/ldap-nested-groups-modelling-and-representation-in-code).

You can also watch a video overview about adding users to Mattermost with AD/LDAP on [YouTube (https://www.youtube.com/watch?v=zyku2ibsG0M)](https://www.youtube.com/watch?v=zyku2ibsG0M).

Pre-installation notes

If you have enabled synchronization with AD/LDAP, all groups matching the default filter `(|(objectClass=group)(objectClass=groupOfNames)`

`(objectClass=groupOfUniqueNames))` will be available to be linked in the groups list view at **System Console > User Management > Groups**.

The group filter is an optional configuration setting available in the **User Filters** section of the AD/LDAP wizard (**System Console > Authentication > AD/LDAP**) and allows you to specify the groups that should have access in Mattermost. The **Group** filter is independent of the **User** filter; however, it does leverage the Base DN attribute. You may need to adjust your Base DN to ensure group objects can be searched in your AD/LDAP tree.

The synchronization of groups happens with the synchronization of users, during which Mattermost queries AD/LDAP for updated account information. Please see the Active Directory/LDAP Set up documentation. for more information. The group feature has no effect on users' authentication to Mattermost.

Enable AD/LDAP group synchronization

To enable this feature, go to **System Console > User Management > Groups**.

Synchronize AD/LDAP groups to Mattermost

To synchronize specific AD/LDAP groups to Mattermost, specify the `Group ID Attribute` and the `Group Display Name Attribute` (e.g., "cn" for Common Name) in the **Group Synchronization** section of the AD/LDAP wizard (**System Console > Authentication > AD/LDAP**).

Additionally, you can specify the **Group** filter used to retrieve groups. If the **Group** filter configuration is left blank, then all groups matching the default filter `(|(objectClass=group)(objectClass=groupOfNames)(objectClass=groupOfUniqueNames))` are returned. Attribute values for **Group ID** and **Group Display Name** are case-sensitive.

Group synchronization occurs after user synchronization and results for group synchronization are available on the synchonization status table (located at the bottom of the **AD/LDAP** configuration page). After the AD/LDAP groups have been synchronized, go to **System Console > User Management > Groups** to link and configure Mattermost groups.

> Note
>
> The synchronization process doesn't create Mattermost groups. Mattermost groups are created when you "link" the AD/LDAP group as outlined in the next section **Linking AD/LDAP groups to Mattermost groups**. Existing AD/LDAP users are added to the Mattermost groups on the next synchronization and new users are added on their first login.

On subsequent synchronizations and once groups are linked:

> - *Users that have been added to an AD/LDAP group will be added to the linked Mattermost group and to teams and channels configured for that group.*

> - *Mattermost groups that are linked to AD/LDAP groups no longer included in your filter are deleted.*
> - *Users removed from an AD/LDAP group are removed from the linked Mattermost group, but their channel and team membership is only revoked when the channel or team is synchronized to an AD/LDAP group.*

LDAP Group and GroupMember sync.

```
mattermost_test=# \d groups
                Table "public.groups"
   Column    |          Type           | Modifiers
-------------+-------------------------+-----------
 id          | character varying(26)   | not null
 name        | character varying(64)   |
 displayname | character varying(128)  |
 description | character varying(1024) |
 type        | character varying(64)   |
 remoteid    | character varying(48)   |
 createat    | bigint                  |
 updateat    | bigint                  |
 deleteat    | bigint                  |
Indexes:
    "groups_pkey" PRIMARY KEY, btree (id)
    "groups_name_key" UNIQUE CONSTRAINT, btree (name)
    "groups_type_remoteid_key" UNIQUE CONSTRAINT, btree (type, remoteid)
    "idx_groups_remote_id" btree (remoteid)
```

```
mattermost_test=# \d groupmembers
         Table "public.groupmembers"
  Column  |          Type          | Modifiers
----------+------------------------+-----------
 groupid  | character varying(26) | not null
 userid   | character varying(26) | not null
 createat | bigint                |
 deleteat | bigint                |
Indexes:
    "groupmembers_pkey" PRIMARY KEY, btree (groupid, userid)
    "idx_groupmembers_create_at" btree (createat)
```

Link AD/LDAP groups to Mattermost groups

Groups that have been returned from the default filter or your AD/LDAP group filter will be available in a list view on the Groups page. The link action will create Mattermost groups corresponding to the AD/LDAP group. AD/LDAP groups linked to a Mattermost group will display the **Linked** icon. AD/LDAP groups that have not been linked to a Mattermost group will display the **Not Linked** icon. An AD/LDAP group that is not linked does not create a Mattermost group.

You can link groups individually by the inline **Linked** button and use the checkbox next to the group name to select multiple groups and choose **Link Selected Groups**. When selecting multiple groups with a mix of **Linked** and **Not Linked** states, the bulk action of the button will be **Link Selected Groups** until all selected are marked **Linked**. Using the bulk action speeds the process of creating Mattermost groups from your AD/LDAP Groups.

If you see a **Link Failed** message, either select the message, or check the box alongside the group name to expose the inline link message and try again.

Configure the linked group

AD/LDAP groups that have been linked to Mattermost groups can be configured to add team and channels. To configure the group, select **Configure > Group Configuration** and view the group profile which includes the group name. This name is automatically mapped from the AD/LDAP group common name attribute and is read-only.

Add default teams or channels for the group

To add the teams and channels that you want the group members to default in, select either **Add Team** or **Add Channel** from the **Add Team or Channel** button. You can assign roles to group members using the options provided in the **Assigned Roles** column. Roles are updated on the next scheduled AD/LDAP synchronization.

Channels are nested below the team they belong to in the team and channel list. The following table describes the icons available on this page and what they indicate:

Icon	Description
	Team is open for anyone to join.
	Team isn't open for anyone to join.
	Public channel.
	Private channel.

> Note
>
> - When a team is added, the `Town Square` and `Off-Topic` channels will also be created automatically, as well as any default channels set in the ExperimentalDefaultChannels config setting.
> - When a channel is added without setting the team explicitly, the team will be shown in the **Team and Channel Membership** listing, but it won't be added to the group specifically. Because of this dependency, when the channel is removed, the team will also be removed. Teams are listed in parentheses after the channel name in the channel selector.

Synchronize teams and channels

For new users, default teams and channels will be added when they log in for the first time. For existing users, default teams and channels will be added after the next scheduled AD/LDAP sychronization.

It may take a few seconds to load all team and channel memberships for a user depending on the number of teams and channels the group is defaulted to. In our testing, it took six seconds for an organization with 200,000 users and 30,000 linked groups.

> Note
>
> Users aren't removed from the team or channel on subsequent synchronizations of the AD/LDAP groups. Users need to be manually removed from the team or channel per the existing functionality. They won't be automatically re-added if they were manually removed or removed themselves. To manage a team or Private channel membership with synchronized groups, please see the section below on **Disable and re-activate AD/LDAP users** for details.

Remove configured teams and channels from a group

To remove a team or channel configured for a group, select **Remove** to the right of the team or channel name. Users already part of the team and channel won't be removed from that channel by this action.

View users belonging to the group

Users who have logged in and accessed Mattermost will be visible in the members list on the group object. Members are read-only at this time and new members can be added through management in your AD/LDAP system.

Users can be removed from the Mattermost group on subsequent synchronizations. However, they won't be removed from teams and channels unless the team or channel is group-synchronized.

> Note
>
> When a member removes themselves manually from a channel, that action is tracked in the **Channel Member History** table. If a system admin manually forces all members in a specific LDAP group to join the teams and channels synchronized to that group, members can potentially be re-added to channels from which they were previously removed.

Disable and re-activate AD/LDAP users

If a member is removed from an AD/LDAP group, deactivated in AD/LDAP, or filtered from the AD/LDAP user filter, that member loses access to Mattermost.

From Mattermost v7.7, system admins can add members back to all of the member's default teams and channels in the System Console by going to **User Management > Users**, selecting the user's role, then selecting the **Re-sync user via LDAP groups** option.

> Tip
>
> Using the Mattermost API, system admins can manually re-add all group members back into synchronized teams or channels by forcing members in an LDAP group to join the teams and channels synchronized to that group, even if members left on their own, were removed, were filtered out, or were deactivated. See our Mattermost API documentation for details on synchronizing user attribute changes in the configured AD/LDAP server with Mattermost.

Manage groups

Once a group has been configured, the default teams and channels can be changed via the **Edit** option on the group list view.

Delete groups

Mattermost groups can be deleted by adjusting your AD/LDAP group filter to remove the group or by unlinking the group on the Groups listing page. If you add the group back by re-adjusting the AD/LDAP group filter and link the group again on the group configuration page, the previous team and channel configurations will be available.

Use AD/LDAP synchronized groups to manage team or private channel membership

Mattermost groups created with synchronized AD/LDAP groups can be used to manage the membership of private teams and private channels. When a team or private channel is managed by synchronized groups, member users will be added and removed based on their membership to the synchronized AD/LDAP group.

> Note
>
> It is not possible to add guests to teams and channels that are managed using groups.

For instance, you may have an AD/LDAP group that contains your development team that you want to synchronize to a developer team. By using this feature, new developers will get added to the team when they are added to the synchronized AD/LDAP group and they will be removed from the team when removed from the AD/LDAP group.

Similarly, you may have an AD/LDAP group that contains your leadership team that you want to synchronize to a private channel for coordination and updates.

This feature helps control the membership of the channel so that guests and member users outside of the synchronized group are prevented from being added to the channel mistakenly.

On teams that are managed by synchronized groups, guests, and member users outside of the group are restricted from:

> - *Invitation through a team invite link.*
> - *Invitations through an email invite.*

Similarly on private channels that are managed by synchronized groups, guests and member users outside of the group are restricted from:

> - *Invitation through a mention.*
> - *Invitation through the* `/invite` *slash command.*
> - *Being added to the channel via the* ***Add Members*** *menu option.*

Users can remove themselves from teams and Private channels managed by synchronized groups.

Manage membership of a team or channel with synchronized groups

To manage membership of a private team with synchronized groups:

1. Go to **System Console > User Management > Teams**. Select the team you want to manage with group synchronization.
2. Under **Team Management**, enable **Sync Group Members**. If **Anyone can join this team** is enabled or if specific email domains are set, they will be disabled by the Sync Group Members feature.
3. Add one or more groups to the team. If there are groups already associated with default users in the team, they'll be listed.
4. Review the notice in the footer of the screen for any users that are not part of groups who will be removed from the team on the next synchronization.
5. Select **Save**. Members will be updated on the next scheduled AD/LDAP synchronization.

Alternatively, you can use the mmctl tools to set the team to be managed by groups:

1. Ensure there is at least one group already associated to the team. You can view and add default teams to a group via **System Console > User Management > Groups > Group Configuration**. Please see more information on adding default teams and channels here. Additionally, you can use the mmctl to confirm if there is already a group associated to the team by running the mmctl group team list command.
2. Ensure **Team Settings > General > Allow any user with an account on this server to join this team** is set to **No**.
3. Convert the team to have its membership managed by synchronized groups by running the mmctl group team enable command.

To manage membership of a private channel with synchronized groups:

1. Go to **System Console > User Management > Channels**. Select the channel you want to manage with group synchronization.

2. Under **Channel Management**, enable **Sync Group Members**. Please ensure the channel is set to **private**.
3. Add one or more groups to the channel. If there are groups already associated with default users in the team, they'll be listed.
4. Review the notice in the footer of the screen for any users that are not part of groups who will be removed from the channel on the next synchronization.
5. Select **Save**.

Members will be updated on the next scheduled AD/LDAP synchronization. Alternatively, you can use the mmctl to set a private channel to be managed by groups:

1. Ensure there is at least one group already associated to the channel. You can view and add default channels to a group via **System Console > User Management > Groups > Group Configuration**. Please see more information on adding default teams and channels here. Additionally, you can use the mmctl to view if there is already a group associated to the channel by running the mmctl group channel list command.
2. Convert the team to have its membership managed by synchronized groups by running the mmctl group channel enable command.

Assign roles to group members

Group members can be assigned predefined roles by system admins, which are applied across the group during the scheduled sychronization. The roles are:

- Member (default)
- Team admin (in teams)
- Channel admin (in channels)

The permissions for each role can be viewed and modified in **System Console > Permissions**.

To set the Team admin role in a synced group

1. Go to **System Console > User Management > Teams**.

2. Select **Edit** next to the team you want to configure.
3. Ensure that **Sync Group Members** is enabled.
4. Choose **Add Group** to add one or more groups to the team. If there are groups already associated to default users into the team, they will already be present.
5. Select the arrow next to the current role in the **Roles** column to display and select the **Team Admin** option.
6. Repeat as needed for any other synced groups you have added.
7. Select **Save**.

Roles are updated on the next scheduled AD/LDAP synchronization.

To set the Channel Admin role in a synced group

1. Go to **System Console > User Management > Channels**.
2. Select **Edit** next to the team you want to configure.
3. Ensure that **Sync Group Members** is enabled.
4. Choose **Add Group** to add one or more groups to the team. If there are groups already associated with default users in the team, they'll be listed.
5. Select the arrow next to the current role in the **Roles** column to display and select the **Channel Admin** option.
6. Repeat as needed for any other synced groups you have added.
7. Select **Save**.

Roles are updated on the next scheduled AD/LDAP synchronization.

> Note
>
> Members who have been synced as part of a group cannot have their role changed via **View Members** in Mattermost.

Add or remove groups from teams

Once team management is converted to use synchronized groups, team admins and system admins can add additional groups from **Main Menu > Add Groups to Team**. This will add users to the next AD/LDAP synchronization, and any new users to the group will be added to the team on subsequent synchronizations. Team admins will be prevented from converting the team to a public space by enabling **Team Settings > Allow any user with an account on this server to join this team**.

Team admins and system admins can also remove groups from a team from **Main Menu > Manage Groups**. This will disassociate the group from the team. Users are removed on the next AD/LDAP synchronization.

The system admin can remove groups from **System Console > User Management > Teams > Team Configuration > Synced Groups**.

Add or remove groups from private channels

Once the management of the channel is converted to be managed by synchronized groups, team admins and system admins can add additional groups from **Channel Menu > Add Groups to Channel**. This will add users on the next AD/LDAP synchronization and any new users to the group will be added to the channel on subsequent synchronizations.

Team admins and system admins can also remove groups from a team from **Main Menu > Manage Groups**. This will disassociate the group from the team. Users are removed on the next AD/LDAP synchronization.

The system admin can remove groups from **System Console > User Management > Channels > Channel Configuration > Synced Groups**.

Manage members

Users are automatically removed from the team or private channel when removed from a synchronized AD/LDAP group that is managing the membership of that team or channel. Additionally, users who are not in the synchronized groups are prevented from being added through the `/invite` and mention flows within a channel.

A user can remove themselves from the team or from the private channel when it is managed by synchronized groups. They can be added back by users who have permission to manage members for a team or private channel by using the `/invite` slash command or by mentioning the user in a channel.

If the user is removed from a synchronized group and later re-added to the group, they can be manually added back to the team or Private channel as noted above.

> Note
>
> Users won't be added back by the AD/LDAP synchronization automatically once they remove themselves or are removed by the LDAP synchronized group.

Disable group-synchronized management of teams and private channels

To remove the management of members by synchronized groups in a team, disable **Sync Group Members** under **System Console > User Management > Teams > Team Management**. Alternatively, you can also run the mmctl group team disable command.

To remove the management of members by synchronized groups in a channel, disable **Sync Group Members** under **System Console > User Management > Channels > Channel Management**. Alternatively, you can also run the mmctl group channel disable command.

Frequently asked questions

Why do my LDAP users and groups exist in Mattermost, but my groups have no members?

In order for Mattermost to detect group membership correctly, and to automatically add users to the group configured in the System Console, you must use one of the following AD/LDAP attributes to represent group members

in Mattermost: `member` or `uniqueMember`. These attributes use a `Distinguished Name` as the value on groups.

> Note
>
> LDAP installations that use `memberUid` to indicate group membership are not supported because `memberUid` is an attribute of an object class `posixGroup` that does not use `Distinguished Names` as the value on groups.

Why can't my existing users see the teams and channels they have been synced to?

Existing Mattermost users that are members of linked Mattermost groups will be added to teams and channels on the next scheduled synchronization job that is run after teams and channels are added to the Mattermost group. You can manually initiate a synchronization from the **Sync History** section of the AD/LDAP wizard (**System Console > Authentication > AD/LDAP > AD/LDAP Synchronize Now**).

How do nested groups work with AD/LDAP Group Sync?

Users within nested groups are included as members of parent groups. The group filter that you specify can include any type of AD/LDAP group on your system. The `member` AD/LDAP attribute is used to determine nested groups that belong to a parent group.

How do I manage a team or private channel membership with synchronized groups?

You can do this by setting the team or channel management to synced groups instead defaulting a group to a team or channel. See the section above on synchronizing teams and channels to learn more.

How do I use AD/LDAP group sync with SAML?

You can use AD/LDAP group sync with SAML by enabling SAML Synchronization with AD/LDAP. You do not need to enable sign-in with LDAP for this feature to work.

However, it's critical that the unique Mattermost ID identifier that you have chosen as your attribute in your directory service (AD/LDAP) is the same for both the SAML and AD/LDAP configurations.

For instance, if `ObjectGUID` has been chosen as the Mattermost ID in your AD/LDAP configuration, then an attribute that has the same value should also be mapped to the ID attribute in your SAML assertion. We also recommend that the ID attribute you select is unique and unchanging (such as a `GUID`).

Why aren't public channels supported with synchronized groups?

Public channels are available to all members to discover and join. Managing membership with synchronized groups removes the ability for Public channels to be accessible to users on the team. Private channels typically require more controlled membership management, which is why this feature applies to Private channels. Groups can be assigned to public teams and public channels as described in this documentation.

Does a team with its membership managed by groups have any effect on public channel access?

Only users that are members of groups synchronized to team are able to discover and join public channels. Private channels can also be managed by synchronized groups when a team is managed by synchronized groups.

Why don't users get readded to teams or channels once they have been removed from and then later re-added to the LDAP group?

The implementation of group removals does not currently differentiate between users who have removed themselves or have been removed by the LDAP synchronization process. Our design optimizes for users who have

removed themselves from a team or channel. In the future, we may add the ability for Admins to re-add users who have been removed and even prevent users from leaving a team or channel.

Additionally, LDAP users who are not accessible to Mattermost based on filters will be removed from the groups and from group synced teams and channels. If they were removed from teams and channels then they would not be re-added to those teams and channels upon becoming subsequently reaccessible to Mattermost.

How can I use LDAP attributes or Groups with OpenID?

At this time, LDAP data isn't compatible with OpenID. If you currently rely on LDAP to manage your users' teams, channels, groups, or attributes, you won't be able to do this automatically with users who have logged in with OpenID. If you need LDAP synced to each user, we suggest using SAML or LDAP as the login provider. Some OpenID providers can use SAML instead, like Keycloak.

On this page

- AD/LDAP groups
  - Overview
  - Pre-installation notes
  - Enable AD/LDAP group synchronization
  - Synchronize AD/LDAP groups to Mattermost
  - Link AD/LDAP groups to Mattermost groups
  - Configure the linked group
  - Add default teams or channels for the group
  - Synchronize teams and channels
  - Remove configured teams and channels from a group
  - View users belonging to the group
  - Disable and re-activate AD/LDAP users
  - Manage groups
  - Delete groups
  - Use AD/LDAP synchronized groups to manage team or private channel membership
    - Manage membership of a team or channel with synchronized groups
    - Assign roles to group members

    - Add or remove groups from teams
    - Add or remove groups from private channels
    - Manage members
    - Disable group-synchronized management of teams and private channels
  - Frequently asked questions
    - Why do my LDAP users and groups exist in Mattermost, but my groups have no members?
    - Why can’t my existing users see the teams and channels they have been synced to?
    - How do nested groups work with AD/LDAP Group Sync?
    - How do I manage a team or private channel membership with synchronized groups?
    - How do I use AD/LDAP group sync with SAML?
    - Why aren’t public channels supported with synchronized groups?
    - Does a team with its membership managed by groups have any effect on public channel access?
    - Why don’t users get readded to teams or channels once they have been removed from and then later re-added to the LDAP group?
    - How can I use LDAP attributes or Groups with OpenID?

AD/LDAP setup

Overview

Mattermost offers “Same Sign-On” with Microsoft AD/LDAP (formally known as Active Directory/LDAP). Enable the same credentials used in on-prem AD/LDAP deployments to be reused in Mattermost, with optional multi-factor authentication.

AD/LDAP is a service that stores authentication and authorization details of users on your organization’s network. When you integrate your AD/LDAP system with Mattermost, users can log into Mattermost without having to create new credentials. User accounts are managed in AD/LDAP, and changes are synchronized with Mattermost.

Mattermost provides a step-by-step AD/LDAP setup wizard in the System Console that guides you through the configuration process with sections and incremental testing to ensure each part of your setup works correctly before proceeding to the next step.

Benefits of integrating AD/LDAP with Mattermost include:

- **Single sign-on.** Users can log in to Mattermost with their AD/LDAP credentials.
- **Centralized identity management.** Mattermost accounts can display user information from AD/LDAP, such as first and last name, email, and username.
- **Automatic account provisioning.** A Mattermost user account is automatically created the first time a user signs in with their AD/LDAP credentials.
- **Sync groups to predefined roles in Mattermost.** Assign team and channel roles to groups via AD/LDAP Group Sync.
- **Compliance alignment with administrator management.** Manage Administrator access to Mattermost in the System Console using AD/LDAP filters.

Pre-installation notes

If you're using AD/LDAP with **nested security groups** you need to write a PowerShell script, or similar, to flatten and aggregate the tree into a single security group to map into Mattermost.

We strongly recommend the following as you prepare to set up AD/LDAP:

Attribute Naming and Case Sensitivity:

- Attribute names in both AD/LDAP and Mattermost configurations are **case-sensitive**.
- Ensure that the attribute names in the AD/LDAP claim rules **exactly match** the expected attribute names in Mattermost. Case deviations will result in issues.

Essential attributes:

- The `NameID` element is required for user identification in SAML assertions.
- All required attributes (e.g., `Email`, `Username`, `FirstName`, and `LastName`) must be included and correctly mapped.

How to choose a stable unique identifier for `NameID`:

- Using a stable and unique identifier (`EmployeeID` or `ObjectGUID`) for the `NameID` helps prevent issues in cases where user details could change over time (e.g., `LastName` or `Email`).
- If stable, unique attributes aren't available in AD, using attributes that might change over time can result in future issues.

Getting started

There are two ways to set up AD/LDAP:

1. **Configure AD/LDAP using the System Console setup wizard**

> - *Log in to your workspace and create a new account using email and password. This is assigned the system admin role as the first user created.*

> - *Next, use the AD/LDAP setup wizard to configure AD/LDAP step-by-step, testing each section as you go, and then convert your system admin account to use the AD/LDAP login method.*

2. **Configure AD/LDAP by editing ``config.json``**

> - *Edit `config.json` to enable AD/LDAP based on the AD/LDAP settings documentation. When you log in to Mattermost the first user to log in with valid AD/LDAP credentials will be assigned the system admin role.*

Configure AD/LDAP login

1. **Create a system admin account using email authentication.**

> - *Create a new workspace and create an account using email and password, which is automatically assigned the **system admin** role since it is the first account created. You may also assign the role to another account.*

2. **Configure AD/LDAP using the setup wizard.**

> - *Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP setup wizard. The wizard is organized into sections that you can navigate through using the sidebar:*
> - ***Connection Settings**: Configure server connection details*
> - ***User Filters**: Set up user identification and filtering*
> - ***Account sync**: Map AD/LDAP attributes to Mattermost user fields*
> - ***Group Synchronization**: Configure group settings and group attributes (if using LDAP groups)*
> - ***Sync Performance**: Adjust synchronization timing and performance settings*
> - ***Sync History**: View synchronization status and manually trigger syncs*

> - *Each section includes a test button that allows you to verify your configuration before proceeding to the next step. This incremental testing helps identify and resolve issues early in the setup process.*

3. **Confirm that AD/LDAP sign-on is enabled.**

> - *After configuring AD/LDAP through the wizard, confirm that users can log in using AD/LDAP credentials.*

4. **Switch your system admin account from email to AD/LDAP authentication.**

> - *Navigate to your profile, and select **Security > Sign-in Method > Switch to AD/LDAP** and log in with your AD/LDAP credentials to complete the switch.*

5. **(Optional) Restrict authentication to AD/LDAP.**

> - *Go to **System Console > Authentication > Email** and set **Enable sign-in with email** to **false** and **Enable sign-in with username** to **false**.*
> - *Then choose **Save** to save the changes. This should leave AD/LDAP as the only login option.*

6. **(Optional) If you configured `First Name Attribute` and `Last Name Attribute` in the System Console.**

> - *Navigate to **System Console > Site Configuration > Users and Teams** and set **Teammate Name Display** to **Show first and last name**. This is recommended for a better user experience.*

> Note
>
> If you've made a mistake and lock yourself out of the system somehow, you can set an existing account to system admin using the mmctl roles command.

Configure AD/LDAP synchronization

In addition to configuring AD/LDAP sign-in, you can also configure AD/LDAP synchronization. When synchronizing, Mattermost queries AD/LDAP for relevant account information and updates Mattermost accounts based on changes to attributes (first name, last name, and nickname). When accounts are disabled in AD/LDAP users are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes the updated attributes.

The AD/LDAP synchronization depends on email. Make sure all users on your AD/LDAP server have an email address, or ensure their account is deactivated in Mattermost.

When Mattermost is configured to use AD/LDAP for user authentication, the following user attribute changes can't be made through the API: first name, last name, position, nickname, email, profile image, or username. LDAP must be the authoritative source for these user attributes.

To configure AD/LDAP synchronization with AD/LDAP sign-in:

1. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard and navigate to the **Connection Settings** section. Set **Enable Synchronization with AD/LDAP** to **true**.
2. Navigate to the **Sync Performance** section and configure the **Synchronization Interval (minutes)** to specify how often Mattermost accounts synchronize attributes with AD/LDAP. The default setting is 60 minutes. The profile picture attribute is only synchronized when the user logs in.

> - *If you want to synchronize immediately after disabling an account, use the **AD/LDAP Synchronize Now** button in the **Sync History** section of the wizard.*
> - *To configure AD/LDAP synchronization with SAML sign-in, see the SAML documentation.*

> Note
>
> - Ensure at least one AD/LDAP user is in Mattermost or the sync won't complete.
> - Synchronization with AD/LDAP settings in the System Console can be used to determine the connectivity and availability of arbitrary hosts. System admins concerned about this can use custom admin roles to limit access to modifying these settings. See the delegated granular administration documentation for details.

3. From Mattermost v10.9, you can configure Mattermost to automatically re-add members of an LDAP group to group-synchronized teams or channels during LDAP synchronization, even if those members were previously removed. This option enables you to maintain uninterrupted collaboration and address specific organizational needs, ensuring users who were unintentionally removed due to changes in LDAP group membership, synchronization errors, or exceptions to the standard group sync rules can be seamlessly restored.

> > *Note*
> >
> > *The mmctl ldap sync command takes precedence over this server configuration setting. If you have this setting disabled, and run the mmctl command with the* `--include-removed-members` *flag, removed members will be re-added during LDAP synchronization.*

Configure AD/LDAP sign-in using filters

Using filters assigns roles to specified users on login. To access AD/LDAP filter settings, navigate to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard and go to the **User Filters** section.

User filter

(Optional) Enter an AD/LDAP filter to use when searching for user objects. Only the users selected by the query will be able to access Mattermost. For AD/LDAP, the query to filter out disabled users is `(&(objectCategory=Person)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))`.

1. Navigate to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard.
2. Go to the **User Filters** section and complete the **User Filter** field.
3. Use the **Test Filters** button to verify your filter works correctly.
4. Choose **Save**.

When the user accesses Mattermost, they log in with same username and password that they use for organizational logins.

Filters can also be used for excluding users who belong to certain groups. For AD/LDAP, the query to filter out groups is `(&(memberof=cn=ACME_ALL,ou=Users,dc=sademo,dc=com)(!(memberof=cn=DEV_OPS,ou=Users,dc=sademo,dc=com)))`.

Guest filter

(Optional) When enabled, the Guest Filter in Mattermost identifies external users whose AD/LDAP role is guest and who are invited to join your Mattermost workspace. These users will have the Guest role applied immediately upon first login instead of the default member user role. This eliminates having to manually assign the role in the System Console.

If this filter is removed/changed, active guests will not be promoted to a member and will retain their Guest role. Guests can be promoted in **System Console > User Management**.

1. Navigate to **System Console > Authentication > Guest Access** and set Guest Access to `true`.
2. Navigate to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard.
3. Go to the **User Filters** section and expand **Configure additional filters**.

4. Complete the **Guest Filter** field.
5. Use the **Test Filters** button to verify your filter works correctly.
6. Choose **Save**.

When a guest logs in for the first time they are presented with a default landing page until they are added to channels.

See the Guest Accounts documentation for more information about this feature.

Admin filter

(Optional) Enter an AD/LDAP filter to use for designating system admins. The users selected by the query will have access to your Mattermost workspace as system admins. By default, system admins have complete access to the Mattermost System Console. Existing members that are identified by this attribute will be promoted from member to system admin upon next login.

The next login is based upon Session lengths set in **System Console > Session Lengths**. It is recommended that users are demoted to members manually in **System Console > User Management** to ensure access is restricted immediately.

1. Navigate to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard.
2. Go to the **User Filters** section and expand **Configure additional filters**.
3. Set **Enable Admin Filter** to **true**.
4. Complete the **Admin Filter** field.
5. Use the **Test Filters** button to verify your filter works correctly.
6. Choose **Save**.

> Note
>
> If the Admin Filter is set to `false`, the member's role as system admin is retained. However if this filter is removed/changed, system admins that were promoted via this filter will be demoted to members and won't retain access to the System Console.

When this filter isn't in use, members can be manually promoted/demoted via **System Console > User Management**.

Configure AD/LDAP deployments with multiple domains

Organizations using multiple domains can integrate with Mattermost using a "Forest" configuration to bring together multiple domains. Please see Forests as Collections of Domain Controllers that Trust Each Other (https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759073(v=ws.10)?redirectedfrom=MSDN) for more information.

For forest configurations that contain multiple domains which do NOT share a common root, you can search across all of the domains using the Global Catalog. To do so, update your `config.json` as follows:

- Set the LdapPort to 3268 (instead of 389)
- Set the BaseDN to " " (A single space character)

See Global Catalog and LDAP Searches (https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc978012(v=technet.10)?redirectedfrom=MSDN) for additional details.

Troubleshooting/FAQ

The following are frequently asked questions and troubleshooting suggestions on common error messages and issues. It is recommended that you check your logs for errors as they can provide an idea of what the issue is.

If the Test Connection button fails, how can I troubleshoot the connection?

Check that your AD/LDAP connection settings are correct by running an AD/LDAP user query in an external system. See [LDAP Connection Test Example (https://ldaptool.sourceforge.net/)](https://ldaptool.sourceforge.net/). If the AD/LDAP connection is verified to be working outside of Mattermost, try the following:

- Check your AD/LDAP system to verify your `Bind Username` format.
- Check your **AD/LDAP Port** and **Connection Security** settings in the System Console. (**AD/LDAP Port** set to 389 typically uses **Connection Security** set to `None`. **AD/LDAP Port** set to 636 typically ties to **Connection Security** set to **TLS**).
- If you're seeing `x509: certificate signed by unknown authority` in your logs, try installing an intermediate SSL certificate or have your LDAP server send the complete certificate chain.

If these options don't work, please [contact our support team (https://mattermost.com/support/)](https://mattermost.com/support/).

When I first set up and synchronize AD/LDAP, are the users automatically created in Mattermost?

No, each user is created on their first login.

When I try to synchronize AD/LDAP, why does the status show as `Pending` and not complete?

Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, and make sure that the **Enable Synchronization with AD/LDAP** setting is set to **true**.

If the issue persists, try selecting the **Test Filters** button to test that the User Filter is correctly formatted. Refer to this document for guidance on setting a correct syntax format.

Make sure that you also have at least one AD/LDAP user in Mattermost or the synchronization will not complete.

What's the difference between the Username Attribute, ID Attribute, and Login ID Attribute?

There are three AD/LDAP attributes that apear to be similar but serve a different purpose:

1. **Username Attribute:** Used within the Mattermost user interface to identify and mention users. For example, if **Username Attribute** is set to `john.smith`, a user typing `@john` will see `@john.smith` in their autocomplete options and posting a message with `@john.smith` will send a notification to that user that they've been mentioned.

2. **ID Attribute:** Used as the unique identifier in Mattermost. It should be an AD/LDAP attribute with a value that does not change, such as `ObjectGUID`. If a user's ID attribute changes, it will create a new Mattermost account unassociated with their old one. If you need to change this field after users have already logged in, use the mattermost ldap idmigrate mmctl tool.

3. **Login ID Attribute:** The attribute in the AD/LDAP server used to log in to Mattermost. Normally this attribute is the same as the **Username Attribute** field above, or another field that users can easily remember.

How do I deactivate users?

If a user has logged into Mattermost through AD/LDAP or SAML, you can choose how they are deactivated, whether manually or automatically.

There are three main ways to do this:

1. **User deletion:** If the user is completely removed from the AD/LDAP server, they will be deactivated in Mattermost on the next synchronization.

2. **User filter:** Set the user filter to only select the subset of AD/LDAP users you want to have access to Mattermost. When someone is removed from the selected group, they will be deactivated in Mattermost on the next synchronization.

3. **Manually deactivate**: Go to **System Console > User Management > Users**, select a user's role, and select **Deactivate**. When you manually deactivate a user, they can reactivate themselves by logging back in.

For AD/LDAP, to filter out deactivated users you must set the user filter to:

`(&(objectCategory=Person)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))`

Filters can also be used for excluding users who belong to certain groups. For AD/LDAP, the query to filter out groups is:

`(&(memberof=cn=ACME_ALL,ou=Users,dc=sademo,dc=com)`

`(!(memberof=cn=DEV_OPS,ou=Users,dc=sademo,dc=com)))`

When a user is deactivated in Mattermost via options one or two above, all the user's current sessions are revoked and they will be unable to log in or access Mattermost.

Can I connect to multiple AD/LDAP servers?

There is currently no built-in way to connect to multiple AD/LDAP servers. You will need to connect the instances in a forest before connecting to Mattermost. Consider upvoting the [feature request (https://portal.productboard.com/mattermost/33-what-matters-to-you)](https://portal.productboard.com/mattermost/33-what-matters-to-you) on our forum.

When trying to log in, I see the error `AD/LDAP not available on this server`

This indicates that there is a problem somewhere with your configuration. We recommend that you check your Mattermost configuration settings to ensure that AD/LDAP is enabled, and the settings are correct.

If you're still having issues, you can [contact support (https://mattermost.com/support/)](https://mattermost.com/support/) for additional troubleshooting.

I see the error `User not registered on AD/LDAP server`

This means the query sent back to the AD/LDAP server returned no results. We recommend that you:

1. Check that the user credentials were entered properly - you should log in with the field set as the *ID Attribute*.
2. Check that the user account exists in the AD/LDAP server.
3. Check the AD/LDAP configuration settings are correct.

If you're still having issues, you can contact Mattermost Support (https://mattermost.com/support/) for additional troubleshooting.

I updated a user account in AD/LDAP, and they can no longer log in to Mattermost

If the user can no longer log in to Mattermost with their AD/LDAP credentials - for example, they get an error message `An account with that email already exists`, or a new Mattermost account is created when they try to log in - this means the **ID Attribute** for their account has changed.

The issue can be fixed by changing the value of the field used for the **ID Attribute** back to the old value. If you're currently using a field that sometimes changes for an **ID Attribute** (e.g. username, email that changes when someone gets married), we recommend you switch to using a non-changing field such as a GUID.

To do this, you can set the Login ID Attribute to whatever you would like users to log in with (e.g. username or email).

> Note
>
> Currently the value is case sensitive. If the **ID Attribute** is set to the username and the username changes from `John.Smith` to `john.smith`, the user will experience problems logging in.

I see the log error `LDAP Result Code 4 "Size Limit Exceeded"`

This indicates that your AD/LDAP server configuration has a maximum page size set and the query coming from Mattermost is returning a result set in excess of that limit.

To address this issue you can set the max page size in your Mattermost configuration to match the limit on your AD/LDAP server. This will return a sequence of result sets that do not exceed the max page size, rather than returning all results in a single query. A max page size setting of 1500 is recommended.

If the error is still occurring, it is likely that no AD/LDAP users have logged into Mattermost yet. Ensure that at least one AD/LDAP user has logged into Mattermost and re-run the synchronization. The error should disappear at that point.

I see the log error `Missing NameID Element`

This indicates that the AD/LDAP server configuration doesn't include the `NameID` element in the SAML assertion. The `NameID` element is required for user identification in SAML assertions. Ensure the `NameID` is mapped to a unique user identifier, such as the user's email address or another stable attribute that isn't subject to change over time.

I see the log error `Username Attribute is Missing`

The `Username` attribute in the SAML assertion was either missing or is incorrectly named. Verify that all required attributes are included in the SAML assertion. Attribute names are case-sensitive and must match exactly what Mattermost expects. Update the claim rules in AD/LDAP to correctly map LDAP attributes to the expected outgoing claim types, ensuring proper casing (e.g., `Username` instead of `UserName`).

Can the Enter ID User Filter read security groups?

Yes it can, but make sure that:

- Permissions are correctly configured on the service account you are using.
- Each user object is a direct member of the security group.

How do I know if an AD/LDAP sync job fails?

Mattermost provides the status of each AD/LDAP sync job in the **Sync History** section of the AD/LDAP wizard (**System Console > Authentication > AD/LDAP**). Here you can see the number of users updated and if the job succeeded or failed.

On this page

- AD/LDAP setup
  - Overview

- Pre-installation notes
- Getting started
- Configure AD/LDAP login
- Configure AD/LDAP synchronization
- Configure AD/LDAP sign-in using filters
  - User filter
  - Guest filter
  - Admin filter
- Configure AD/LDAP deployments with multiple domains
- Troubleshooting/FAQ
  - If the Test Connection button fails, how can I troubleshoot the connection?
  - When I first set up and synchronize AD/LDAP, are the users automatically created in Mattermost?
  - When I try to synchronize AD/LDAP, why does the status show as Pending and not complete?
  - What’s the difference between the Username Attribute, ID Attribute, and Login ID Attribute?
  - How do I deactivate users?
  - Can I connect to multiple AD/LDAP servers?
  - When trying to log in, I see the error AD/LDAP not available on this server
  - I see the error User not registered on AD/LDAP server
  - I updated a user account in AD/LDAP, and they can no longer log in to Mattermost
  - I see the log error LDAP Result Code 4 "Size Limit Exceeded"
  - I see the log error Missing NameID Element
  - I see the log error Username Attribute is Missing
  - Can the Enter ID User Filter read security groups?
  - How do I know if an AD/LDAP sync job fails?

Advanced permissions: backend infrastructure

This document outlines the backend server infrastructure for permissions in Mattermost and is recommended only for technical Admins or developers looking to make modifications to their installation.

Entity definitions

Permissions

A **permission** describes a permitted action which may be carried out on an object. It describes the action that users may perform in the context in which they have been assigned the role granting the permission.

Roles

A **role** is something to which permissions are granted, that is then assigned to users in contexts in order to grant them the assigned permissions in that context. One user may end up with different sets of permissions granted by different roles in different contexts.

Scope

Permissions live within a given scope. There are three scopes in the Mattermost system: System, Team and Channel. Permissions cascade down the scopes from the context in which they are applied. For example, if a "Channel" scoped permission is applied to a "Team" context, the permission applies to any channels within that team. A permission is considered,

- **System scope** if it makes sense only on the system level. For example, `manage_oauth`.
- **Team scope** if it makes sense at the team level and system level. For example, `create_public_channel`.

- **Channel scope** if it makes sense at channel, team and system level. For example, `manage_public_channel_properties`.

Context

A **context** is an instance of a scope. For example, a channel called "Developers Hangout" is an instance of channel scope. Contexts have hierarchical relationships between them that reflect the hierarchical ordering of scopes. Each context has one parent, and may have multiple children, with the ultimate parent context being the system context:

- A channel context has a parent team context, whose parent is the system context. For example, the "Developers Hangout" channel is the channel context, with parent team context "Contributors Team", with parent system context.

- A team context has a parent system context and child channel contexts. For example, the "Contributors Team" is the team context, with parent system context, and with children channel contexts such as "Developers Hangout", "Reception" and "Marketing".

When determining whether a user is allowed to carry out a given action in a given context, the union of the permissions of all roles that user has been assigned in the current context and its parent contexts is calculated. This enables permissions to cascade down the scope hierarchy. For example, if a users is granted the `manage_public_channel_properties` permission in a role in the system context, then the user has permissions to manage public channel properties in all channels, in all teams, of which they are a member.

Schemes

Schemes describe the default roles applied to users in a context, and all child contexts. Schemes are either defined specifically for a context, or if they are not specified, the relevant parts of the parent context's scheme are applied, ultimately climbing the hierarchy to the System Scheme, which serves the purpose of providing the system-wide defaults. For example, if Team A does not have a team-scoped scheme defined, the System Scheme will provide the defaults for all contexts in Team A.

Additionally, the lowest-scoped scheme always takes precedence in the context. For example, if Team B has a team-scoped scheme, that scheme takes precedence over the System Scheme defaults for all contexts in Team B.

Data structure

Permissions

Permissions in Mattermost are a property of the server code base and are not created or modified dynamically. The current set of permissions are as described in the table below.

Mattermost permissions

Name (i18n)	Scope	Description
invite_user	team	Invite users to the team using Send Email Invite or Get Team Invite Link.
add_user_to_team	team	Add existing server users to the current team.
manage_slash_commands	system	Create, edit, and delete your own slash commands.
manage_others_slash_commands	system	Edit or delete other users' slash commands.
create_public_channel	team	Create public channels.
create_private_channel	team	Create private channels.
manage_public_channel_members	channel	Manage public channel members.
manage_private_channel_members	channel	Manage private channel members.

Name (i18n)	Scope	Description
assign_system_admin_role	system	Grant other users the system admin role.
manage_roles	system	Manage other users' system-wide roles.
manage_team_roles	team	Add and remove team members.
manage_channel_roles	channel	Add and remove channel members.
manage_system	system	Access to System Console.
sysconsole_read_usermanagement_system_roles	system	View system roles.
sysconsole_write_usermanagement_system_roles	system	Add, remove, and assign system roles.
create_direct_channel	system	Open Direct Message channels.
create_group_channel	system	Open Group Message channels.
manage_public_channel_properties	channel	Edit public channel name, header, and purpose.

Name (i18n)	Scope	Description
manage_private_channel_properties	channel	Edit private channel name, header, and purpose.
list_public_teams	system	View public teams listed in the “Join Another Team” menu accessed from the main menu.
join_public_teams	system	Join public teams listed in the “Join Another Team” menu accessed from the main menu.
list_private_teams	system	View private teams listed in the “Join Another Team” menu accessed from the main menu.
join_private_teams	system	Join private teams listed in the “Join Another Team” menu accessed from the main menu.
list_team_channels	team	List public channels in a team.
join_public_channels	team	Join public channels.

Name (i18n)	Scope	Description
delete_public_channel	channel	Archive public channels.
delete_private_channel	channel	Archive private channels.
edit_other_users	system	Edit values on the *user* object of other users.
read_channel	channel	View posts in a channel.
read_channel_contents	channel	Read the contents of a channel.
read_public_channel	team	View and access public channels on a team.
add_reaction	channel	Add emoji reactions to posts.
remove_reaction	channel	Remove emoji reactions from posts.
remove_others_reactions	channel	Remove other users emoji reactions from posts.

Name (i18n)	Scope	Description
permanent_delete_user (deprecated)	system	Permanently delete other users.
upload_file	channel	Upload file attachments to posts.
get_public_link	system	Get permalink for posts.
manage_incoming_webhooks	team	Create, edit, and delete your own incoming webhooks.
manage_outgoing_webhooks	team	Create, edit, and delete your own outgoing webhooks.
manage_others_webhooks(deprecated)	team	Edit and delete other users' incoming or outgoing webhooks.
manage_others_incoming_webhooks	team	Edit and delete other users' incoming webhooks.
manage_others_outgoing_webhooks	team	Edit and delete other users' outgoing webhooks.
manage_oauth	system	

Name (i18n)	Scope	Description
		Create, edit, and delete your own OAuth 2.0 apps.
manage_system_wide_oauth	system	Edit or delete other users' OAuth 2.0 apps.
create_post	channel	Post in channels.
create_post_public	channel	Create a system message within a channel.
create_post_ephermal	channel	Create an ephemeral message within a channel.
edit_post	channel	Authors edit their own posts.
edit_others_posts	channel	Edit other users' posts.
delete_post	channel	Authors delete their own posts.
delete_others_posts	channel	Delete other users' posts.

Name (i18n)	Scope	Description
remove_user_from_team	team	Remove users from team.
create_team	system	Create teams.
manage_team	team	Access Team Settings.
import_team	system	Import teams in Team Settings.
view_team	team	Read the Team object.
create_bot	team	Create bot accounts.
assign_bot	team	Assign bots to users other than who created the bot.
read_bot	team	View own bots created.
read_others_bots	team	View bots created by others.
manage_bots	team	Edit and delete own bots.

Name (i18n)	Scope	Description
manage_others_bots	team	Edit and delete bots created by others.
view_members	team	List all members on the team.
list_users_without_team	system	List users without a team.
create_user_access_token	system	Create user access tokens.
read_user_access_token	system	Read user access tokens by ID.
revoke_user_access_token	system	Revoke user access tokens.
manage_jobs	system	Create and cancel jobs.
create_emojis	team	Create custom emoji.
delete_emojis	team	Delete own custom emoji.
delete_others_emojis	team	Delete custom emoji created by others.

Name (i18n)	Scope	Description
invite_guest	system	Invite guest users via email invite or add existing guests to teams.
promote_guest	system	Promote guests to member users.
demote_to_guest	system	Demote member users to guests.
manage_remote_clusters	system	Add, remove, and view remote clusters for shared channels.
manage_post_bleve_indexes_job	system	Manage the status of a Bleve post indexing job.
manage_data_retention_job	system	Manage the status of a data retention job.
manage_compliance_export_job	system	Manage the status of a compliance export job.
manage_elasticsearch_post_indexing_job	system	Manage the status of an Elasticsearch post indexing job.
manage_elasticsearch_post_aggregation_job	system	

Name (i18n)	Scope	Description
		Manage the status of an Elasticsearch post aggregation job.
manage_ldap_sync_job	system	Manage the status of an LDAP synchronization job.
add_bookmark_public_channel	channel	Add bookmarks to a public channel.
add_bookmark_private_channel	channel	Add bookmarks to a private channel.
edit_bookmark_public_channel	channel	Make changes to bookmarks in a public channel.
edit_bookmark_private_channel	channel	Make changes to bookmarks in a private channel.
delete_bookmark_public_channel	channel	Delete bookmarks in a public channel.
delete_bookmark_private_channel	channel	Delete bookmarks in a private channel.
order_bookmark_public_channel	channel	Reorder bookmarks in a public channel.

Name (i18n)	Scope	Description
order_bookmark_private_channel	channel	Reorder bookmarks in a private channel.
manage_channel_banner	channel	Manage channel banners.

`Roles` field

Roles are applied to objects that represents that user's membership in a context. These are referenced in the `Roles` field of the `User`, `TeamMember`, `ChannelMember` and `Schemes` Tables.

In the `TeamMember` and `ChannelMember` tables, it's the `Roles` field that contains custom roles and the `SchemeAdmin` and `SchemeUser` booleans that indicate the member object should inherit the respective roles from the relevant scheme, either the default or custom scheme assigned to the relevant team.

`Roles` table

Roles are dynamic and user configurable, necessitating a database table with the following fields:

- `Id` (Autoincrement, Primary Key)
- `Name` (Unique String with Character Constraints, e.g. "team_user").
- `Display Name` (String)
- `Description` (String)
- `Permissions` (String): Space-separated permissions names
- `Scheme Managed` (bool): Indicates whether this role is managed as part of a scheme.
- `BuiltIn` (bool): Indicates if this role is built-in to the Mattermost system and not removable by the user.

Built-in roles

The System Scheme is built-in to the product, and its roles are defined as `BuiltIn: true` in the `Roles` table. You can use the Mattermost API to [retrieve a list of permissions by role name (https://api.mattermost.com/#tag/roles/paths/~1roles~1names/post)](https://api.mattermost.com/#tag/roles/paths/~1roles~1names/post).

The following built-in roles with default permissions are available:

channel_admin

- manage_private_channel_members
- read_public_channel_groups
- use_channel_mentions
- create_post
- use_group_mentions
- add_reaction
- read_private_channel_groups
- remove_reaction
- manage_public_channel_members
- manage_channel_roles
- add_bookmark_public_channel
- edit_bookmark_public_channel
- delete_bookmark_public_channel
- order_bookmark_public_channel
- add_bookmark_private_channel
- edit_bookmark_private_channel
- delete_bookmark_private_channel
- order_bookmark_private_channel

- manage_channel_banner

channel_guest

- read_channel
- read_channel_contents
- add_reaction
- remove_reaction
- upload_file
- edit_post
- create_post
- use_channel_mentions

channel_user

- manage_public_channel_properties
- use_group_mentions
- add_reaction
- delete_private_channel
- manage_private_channel_members
- read_private_channel_groups
- delete_public_channel
- read_public_channel_groups
- use_channel_mentions
- read_channel
- read_channel_contents
- delete_post
- get_public_link

- remove_reaction
- manage_public_channel_members
- upload_file
- manage_private_channel_properties
- create_post
- edit_post
- add_bookmark_public_channel
- edit_bookmark_public_channel
- delete_bookmark_public_channel
- order_bookmark_public_channel
- add_bookmark_private_channel
- edit_bookmark_private_channel
- delete_bookmark_private_channel
- order_bookmark_private_channel

system_admin

- manage_others_slash_commands
- sysconsole_write_user_management_permissions
- edit_brand
- remove_reaction
- manage_incoming_webhooks
- sysconsole_write_user_management_groups
- create_public_channel
- manage_private_channel_members
- sysconsole_write_authentication

- join_private_teams
- create_post_ephemeral
- list_users_without_team
- sysconsole_write_reporting
- join_public_channels
- invite_guest
- list_private_teams
- sysconsole_write_user_management_channels
- manage_others_bots
- read_user_access_token
- add_user_to_team
- view_members
- edit_post
- demote_to_guest
- delete_others_posts
- sysconsole_write_plugins
- delete_private_channel
- sysconsole_read_user_management_system_roles
- sysconsole_read_user_management_users
- revoke_user_access_token
- read_others_bots
- read_public_channel_groups
- sysconsole_write_user_management_teams
- sysconsole_write_billing

- convert_public_channel_to_private
- remove_user_from_team
- manage_team
- add_reaction
- manage_oauth
- list_team_channels
- create_team
- read_jobs
- invite_user
- manage_shared_channels
- remove_others_reactions
- manage_remote_clusters
- sysconsole_write_user_management_users
- sysconsole_read_experimental
- sysconsole_write_compliance
- edit_others_posts
- assign_bot
- manage_bots
- manage_others_outgoing_webhooks
- manage_system_wide_oauth
- delete_others_emojis
- manage_others_incoming_webhooks
- promote_guest
- sysconsole_write_experimental

- sysconsole_read_plugins
- create_group_channel
- sysconsole_read_environment
- manage_roles
- use_channel_mentions
- manage_public_channel_properties
- manage_channel_roles
- get_public_link
- sysconsole_read_billing
- sysconsole_write_integrations
- download_compliance_export_result
- manage_slash_commands
- assign_system_admin_role
- create_post
- delete_post
- create_direct_channel
- list_public_teams
- create_post_public
- read_private_channel_groups
- sysconsole_read_integrations
- read_other_users_teams
- manage_jobs
- sysconsole_read_site
- manage_outgoing_webhooks

- sysconsole_write_environment
- manage_system
- sysconsole_read_user_management_permissions
- manage_public_channel_members
- sysconsole_write_about
- import_team
- sysconsole_write_user_management_system_roles
- sysconsole_read_reporting
- upload_file
- read_channel
- read_channel_contents
- sysconsole_read_user_management_teams
- delete_emojis
- manage_private_channel_properties
- view_team
- sysconsole_read_user_management_groups
- create_private_channel
- create_bot
- join_public_teams
- delete_public_channel
- read_public_channel
- sysconsole_read_about
- read_bots
- sysconsole_read_authentication

- edit_other_users
- sysconsole_read_user_management_channels
- convert_private_channel_to_public
- use_group_mentions
- create_user_access_token
- sysconsole_write_site
- manage_team_roles
- sysconsole_read_compliance
- create_emojis
- manage_post_bleve_indexes_job
- manage_data_retention_job
- manage_compliance_export_job
- manage_elasticsearch_post_indexing_job
- manage_elasticsearch_post_aggregation_job
- manage_ldap_sync_job
- add_bookmark_public_channel
- edit_bookmark_public_channel
- delete_bookmark_public_channel
- order_bookmark_public_channel
- add_bookmark_private_channel
- edit_bookmark_private_channel
- delete_bookmark_private_channel
- order_bookmark_private_channel
- manage_channel_banner

system_custom_group_admin

- create
- edit
- delete
- manage members
- restore

system_guest

- create_group_channel
- create_direct_channel

system_manager

- sysconsole_write_user_management_permissions
- sysconsole_read_about
- sysconsole_read_user_management_channels
- join_private_teams
- delete_private_channel
- view_team
- read_jobs
- sysconsole_read_user_management_teams
- sysconsole_read_plugins
- manage_channel_roles
- manage_public_channel_members
- remove_user_from_team
- sysconsole_read_environment
- list_private_teams

- manage_private_channel_members
- manage_private_channel_properties
- edit_brand
- add_user_to_team
- convert_public_channel_to_private
- read_private_channel_groups
- sysconsole_write_environment
- manage_jobs
- sysconsole_read_reporting
- read_public_channel
- manage_team
- read_channel
- sysconsole_read_integration
- read_public_channel_groups
- list_public_teams
- manage_team_roles
- sysconsole_read_user_management_groups
- manage_public_channel_properties
- sysconsole_write_user_management_groups
- sysconsole_read_user_management_permissions
- sysconsole_write_site
- sysconsole_read_site
- sysconsole_write_user_management_channels
- sysconsole_write_integrations

- delete_public_channel
- sysconsole_write_user_management_teams
- join_public_teams

system_post_all

- create_post
- use_channel_mentions
- use_group_mentions

system_post_all_public

- create_post_public
- use_group_mentions
- use_channel_mentions

system_read_only_admin

- sysconsole_read_compliance
- read_other_users_teams
- sysconsole_read_reporting
- list_private_teams
- sysconsole_read_experimental
- read_jobs
- read_public_channel
- view_team
- sysconsole_read_user_management_users
- sysconsole_read_plugins
- sysconsole_read_user_management_teams
- read_public_channel_groups

- sysconsole_read_user_management_channels
- sysconsole_read_user_management_permissions
- sysconsole_read_about
- download_compliance_export_result
- read_channel
- sysconsole_read_authentication
- sysconsole_read_site
- list_public_teams
- sysconsole_read_integrations
- read_private_channel_groups
- sysconsole_read_environment
- sysconsole_read_user_management_groups

system_user

- list_public_teams
- join_public_teams
- create_direct_channel
- create_group_channel
- view_members
- create_team
- create_emojis
- delete_emojis

system_user_access_token

- create_user_access_token
- read_user_access_token

- revoke_user_access_token

system_user_manager

- manage_public_channel_members
- sysconsole_write_user_management_groups
- manage_private_channel_properties
- read_channel
- sysconsole_read_authentication
- manage_private_channel_members
- read_jobs
- view_team
- sysconsole_read_user_management_groups
- list_private_teams
- join_public_teams
- manage_team
- list_public_teams
- add_user_to_team
- sysconsole_read_user_management_channels
- sysconsole_write_user_management_teams
- read_public_channel
- sysconsole_read_user_management_permissions
- manage_public_channel_properties
- join_private_teams
- convert_public_channel_to_private
- manage_channel_roles

- sysconsole_read_user_management_teams
- read_public_channel_groups
- delete_public_channel
- remove_user_from_team
- manage_team_roles
- delete_private_channel
- sysconsole_write_user_management_channels
- read_private_channel_groups

team_admin

- remove_user_from_team
- manage_others_slash_commands
- manage_team_roles
- manage_public_channel_members
- use_group_mentions
- manage_others_outgoing_webhooks
- manage_slash_commands
- manage_team
- manage_others_incoming_webhooks
- manage_channel_roles
- read_public_channel_groups
- remove_reaction
- delete_post
- manage_outgoing_webhooks
- use_channel_mentions

- manage_incoming_webhooks
- delete_others_posts
- read_private_channel_groups
- create_post
- manage_private_channel_members
- convert_public_channel_to_private
- add_reaction
- import_team
- add_bookmark_public_channel
- edit_bookmark_public_channel
- delete_bookmark_public_channel
- order_bookmark_public_channel
- add_bookmark_private_channel
- edit_bookmark_private_channel
- delete_bookmark_private_channel
- order_bookmark_private_channel
- manage_channel_banner

team_guest

- view_team

team_post_all

- create_post
- use_group_mentions
- use_channel_mentions

team_post_all_public

- use_group_mentions
- create_post_public
- use_channel_mentions

team_user

- invite_user
- add_user_to_team
- list_team_channels
- join_public_channels
- read_public_channel
- view_team
- create_public_channel
- create_private_channel

`Schemes` table

Schemes are dynamic and user configurable, necessitating a database table with the following fields:

- `Id` (Autoincrement, Primary Key)
- `Name` (Unique String with Character Constraints, e.g. "corporate_scheme")
- `Display` Name
- `Description` (String)
- `Scope` (String): Team or Channel
- `Team Admin Role` (String): Empty if Channel Scope
- `Team User Role` (String): Empty if Channel Scope
- `Team Guest Role` (String): Empty if Channel Scope

- `Channel Admin Role` (String): Always provided
- `Channel User Role` (String): Always provided
- `Channel Guest Role` (String): Always provided

On this page

- Advanced permissions: backend infrastructure
  - Entity definitions
    - Permissions
    - Roles
    - Scope
    - Context
    - Schemes
  - Data structure
    - Permissions
    - Roles field
    - Roles table
    - Built-in roles
    - Schemes table

Advanced permissions

Mattermost system admins using Mattermost Cloud or Mattermost Server can use Advanced Permissions to customize which users can perform specific actions, such as creating teams, managing channels, and configuring webhooks. The Mattermost permission system is based on a modified RBAC (role-based access control) architecture, using roles to determine which users have the ability to perform various actions.

Two permission schemes are provided in Mattermost:

- **System Scheme**: Applies permissions universally across all teams and channels.
- **Team Override Schemes**: Allow admins to customize permissions for each team.

This document describes the types of permissions that can be given to users of Mattermost using schemes as well as channel settings and roles. The permissions backend documentation provides additional technical details around permissions.

Permissions structure

The Mattermost System Console provides a number of elements for Admins to control the permissions in their system.

System scheme

You can set the default permissions granted to system admins, team admins, channel admins, guests (if enabled), and all members. The permissions granted in the System Scheme apply system-wide, meaning:

- **Guests:** If Guest Accounts are enabled, permissions apply to guest users in all channels, in all teams.
- **All Members:** Permissions apply to all members, including admins, in all channels, in all teams.

- **Channel Administrators:** Permissions apply to all channel admins in all channels, in all teams.
- **Team Administrators:** Permissions apply to all team admins, in all teams.

To override the System Scheme default permissions in a specific team, you must set up a Team Override Scheme.

You can access the System Scheme interface by going to **System Console > User Management > Permissions > System Scheme**.

System Scheme

Configure the default permissions for Team Admins, Channel Admins and other members. This scheme is inherited by all teams unless a Team Override Scheme is applied in specific teams.

Guests

Permissions granted to guest users.

Permission	Description
[ ] Create Channels	Create new private channels.
[x] Edit Own Posts	Anytime after posting, allow users to edit their own posts.
[ ] Delete Own Posts	Author's own posts can be deleted.
[x] Create Posts	Allow users to create posts.
[x] Post Reactions	Add and delete reactions on posts.
[x] Channel Mentions	Notify channel members with @all, @channel and @here
[ ] Group Mentions	Notify group members with a group mention

All Members

Permissions granted to all members, including administrators and newly created users.

Permission	Description
[x] Teams	Create teams and manage members.
[x] Add Team Members	Add team members, send email invites and share team invite link.
[x] Create Teams	Create new teams.
[x] Invite guests	Invite guests to teams and send guest email invites.
[-] Manage Public Channels	Join, create and archive channels, manage settings and members.
[x] Create Channels	Create new public channels.
[x] Manage Channel Settings	Update public channel names, headers and purposes.
[x] Manage Channel Members	Add and remove public channel members.
[x] Archive Channels	Archive public channels.
[ ] Convert Channels	Convert public channels to private
[x] Manage Bookmarks	Add, edit, delete and sort bookmarks
[x] Manage Private Channels	Create and archive channels, manage settings and members.
[x] Create Channels	Create new private channels.
[x] Manage Channel Settings	Update private channel names, headers and purposes.
[x] Manage Channel Members	Add and remove private channel members.
[x] Archive Channels	Archive private channels.
[x] Manage Bookmarks	Add, edit, delete and sort bookmarks
[-] Manage Public Playbooks	Manage public playbooks.
[x] Create Public Playbook	Create new public playbooks.
[x] Manage Playbook Configurations	Prescribe checklists, actions, and templates.
[x] Manage Playbook Members	Add and remove public playbook members.
[ ] Convert Playbooks	Convert public playbooks to private.
[-] Manage Private Playbooks	Manage private playbooks.
[x] Create Private Playbook	Create new private playbooks.
[x] Manage Playbook Configurations	Prescribe checklists, actions, and templates.
[x] Manage Playbook Members	Add and remove private playbook members.
[ ] Convert Playbooks	Convert private playbooks to public

Team override scheme

On systems with multiple Mattermost teams, each team may operate and collaborate in a unique way. Team Override Schemes give Admins the flexibility to tailor permissions to the needs of each team.

When you use this permission scheme:

- The permissions granted in a Team Override Scheme apply only in the teams which are assigned to the scheme.
- The System Scheme does not apply to teams that are added to a Team Override Scheme.
- Teams can only belong to one Team Override Scheme.

You can access the Team Override Scheme interface by going to **System Console > User Management > Permissions > Team Override Schemes**.

Team Scheme

Team Override Schemes set the permissions for Team Admins, Channel Admins and other members in specific teams. Use a Team Override Scheme when specific teams need permission exceptions to the System Scheme.

Scheme Details

Set the name and description for this scheme.

Scheme Name: Team Scheme

Scheme Description: Scheme Description

Select teams to override permissions

Select teams where permission exceptions are required.

Add Teams

No team selected. Please add teams to this list.

Guests

Permissions granted to guest users.

Permission	Description
☐ Create Channels	Create new private channels.
☑ Edit Own Posts	Anytime after posting, allow users to edit their own posts.
☐ Delete Own Posts	Author's own posts can be deleted.
☑ Create Posts	Allow users to create posts.
☑ Post Reactions	Add and delete reactions on posts.
☑ Channel Mentions	Notify channel members with @all, @channel and @here
☐ Group Mentions	Notify group members with a group mention

All Members

Permissions granted to all members, including administrators and newly created users.

Permission	Description
☑ Teams	Manage team members.
☑ Add Team Members	Add team members, send email invites and share team invite link.
☑ Invite guests	Invite guests to teams and send guest email invites.
▬ Manage Public Channels	Join, create and archive channels, manage settings and members.
☑ Create Channels	Create new public channels.
☑ Manage Channel Settings	Update public channel names, headers and purposes.
☑ Manage Channel Members	Add and remove public channel members.
☑ Archive Channels	Archive public channels.
☐ Convert Channels	Convert public channels to private
☑ Manage Bookmarks	Add, edit, delete and sort bookmarks
☑ Manage Private Channels	Create and archive channels, manage settings and members.
☑ Create Channels	Create new private channels.

Channel permissions

The channel permissions interface is accessed in **System Console > User Management > Channels**.

Advanced access controls

See the team and channel management documentation for details.

Supplementary roles

Being considered for a future release of Mattermost Enterprise

Allows admins to grant additional permissions to specific users or to a group of users based on AD/LDAP group membership. Permissions can be granted within the scope of channels, teams, or system level.

Recipes

This section provides some examples of common permissions use cases for team management, channel management, and overall permissions.

Team management

Ensure users only see each other when in the same team or channel

Example: A classified organization wants to use Mattermost teams for classified projects. In each project, team members can't know about members outside of their project, and @mentions can't disclose the names of people outside of a classified project.

Use the mmctl permissions remove command to revoke the `view_member` permission from the `system_user` role: `mmctl permissions remove system_user view_member`.

Only allow admins, in a specific team, to add members

Example: In Team A, only allow system and team admins to add new team members. As the default for all other teams, allow all users to add and invite new members.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members > Teams** panel, check the box for **Add Team Members**. This sets the system default for all teams.
4. Select **Save**.
5. Select the back arrow to return to the **Permission Schemes** menu.
6. Select **New Team Override Scheme**.

> 1. *Name and describe the scheme. For example,* `Authorized Personnel Only` *with description* `Restrict adding team members to eam and System Admins.`
> 2. *Select **Add Teams** to add Team B to the **Select teams to override permissions** list, locate Team B, then select **Add**.*
> 3. *In the **All Members** panel, uncheck the box for **Add Team Members**.*
> 4. *In the **Team Administrators** panel, check the box for **Add Team Members**.*

7. Select **Save**.
8. Select the back arrow to return to the **Permission Schemes** menu.

Public and private channel management

Restrict who can rename & edit channels

Example: As the default for the entire system, restrict renaming channels and editing headers and purposes to admins only.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.

3. In the **All Members** panel, uncheck the box for **Manage Public Channels > Manage Channel Settings**.

The **Manage Channel Settings** option is now only available to channel admins, team admins, and system admins.

Restrict who can create channels in specific teams

Example: In Team C, restrict public channel creation to admins. As the default for all other teams, allow everyone to create public channels.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members** panel, in the **Manage Public Channels** menu, check the box for **Create Channels**. This sets the system default to allow creation of public channels on all teams.
4. Select **Save**.
5. Select the arrow to return to the **Permission Schemes** interface.
6. Select **New Team Override Scheme**.

> *1. Name and describe the scheme. For example, `Contractor Scheme` with description `Restrict public channel creation to Admins only`.*
>
> *2. Select **Add Teams** to add Team B to the **Select teams to override permissions** list, locate Team B, then select **Add**.*
>
> *3. In the **All Members** panel, in the **Manage Public Channels** section, uncheck the box for **Create Channels**.*
>
> *4. In the **Team Administrators** panel, in the **Manage Public Channels** section, check the box for **Create Channels**.*

Convert public channels to private channels

Allow anyone to convert public channels to private channels

Example: Set the default setting to allow all members, team admins, and channel admins to convert public channels to private.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members** panel, uncheck the box for **Manage Public Channels > Convert Channels**.

This permission is applied to all other roles (excluding the Guest role). When this permission is not enabled for all members, it must be manually applied to team admins and channel admins if required.

Read-only channels

Note

Members can participate but guests can only read and react

1. Go to **System Console > User Management > Channels**.
2. Select **Edit** next to the name of the channel you want to configure.
3. In the **Create Posts** panel, uncheck **Guests**.
4. In the **Post Reactions** panel, uncheck **Guests** if required.
5. Select **Save**.

The channel is available for all members and guests to access, but guests can only read messages and react to them.

Create an announcement channel where only channel admins can post

1. Create a new channel (either public or private).
2. Navigate to **System Console > User Management > Channels**.

3. Select **Edit** next to the name of the channel you just created (you may need to search for it).
4. In the **Create Posts** panel, uncheck **Guests** and **Members**.
5. In the **Post Reactions** panel, uncheck **Guests** and **Members**.
6. Select **Save**.

The channel is available for all members and guests to access but only admins can post.

Message management

Restrict who can delete messages

Example: As the default for the entire system, restrict deleting posts to only system and team admins.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members** and **Channel Administrators** panels, in the **Delete Posts** section, uncheck the boxes for **Delete Own Posts** and **Delete Others' Posts**.
4. In the **Channel Administrators** and **Team Administrators** panels, in the **Delete Posts** section, check the boxes for **Delete Own Posts** and **Delete Others' Posts**.

Restrict who can edit messages

Example: As the default for the entire system, only allow users to edit their own posts for five minutes after posting.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members**, **Channel Administrators**, and **Team Administrators** panels, in the **Manage Posts** section, check the box for **Edit Posts**.

4. From any panel, select the gear button to set the global time limit to `300` seconds.

Integration management

Restrict who can manage webhooks and slash commands

Example: As the default for the entire system, only allow system admins to create, edit and delete integrations.

1. Go to **System Console > User Management > Permissions**.
2. Select **Edit Scheme**.
3. In the **All Members** and **Team Administrators** panels, in the **Integrations & Customizations** section, uncheck the boxes for **Manage Incoming Webhooks**, **Manage Outgoing Webhooks**, and **Manage Slash Commands**.

Administration tools

There are a number of API and mmctl tools available for admins to help in configuring and troubleshooting the permissions system:

1. Reset all permissions to the default on new installations using the mmctl permissions reset command.
2. Use the [GetAllRoles (https://api.mattermost.com/#tag/roles/operation/GetAllRoles)](https://api.mattermost.com/#tag/roles/operation/GetAllRoles) API endpoint to get a list of all roles.
3. Add permissions to a role using the mmctl permissions add command.

Backend infrastructure

Technical admins or developers looking for a deeper understanding of the permissions backend can refer to our permissions backend documentation.

Glossary

- **Permission:** The ability to execute certain actions. Permissions are granted to roles.
- **Roles:** A set of permissions. Users or groups are assigned to roles.
- **Group:** A set of users, usually synced from AD/LDAP. Groups are assigned to roles in the context of teams, channels, or system-wide.
- **Default Roles:** All Members, Guests (if enabled), channel admins, team admins, system admins.
- **System Scheme:** A set of default roles that apply system-wide.
- **Team Override Scheme:** A set of default roles that apply only in the team specified. Permissions granted to roles in a Team Override Scheme overrides roles in the System Scheme.
- **System-wide:** Applies across the entire system, including all teams of which the user is a member.
- **Team-wide:** Applies in a specific team only.

On this page

- Advanced permissions
  - Permissions structure
    - System scheme
    - Team override scheme
  - Channel permissions
    - Advanced access controls
    - Supplementary roles
  - Recipes
    - Team management
      - Ensure users only see each other when in the same team or channel
      - Only allow admins, in a specific team, to add members
    - Public and private channel management
      - Restrict who can rename & edit channels

        - Restrict who can create channels in specific teams
    - Convert public channels to private channels
        - Allow anyone to convert public channels to private channels
    - Read-only channels
        - Members can participate but guests can only read and react
        - Create an announcement channel where only channel admins can post
    - Message management
        - Restrict who can delete messages
        - Restrict who can edit messages
    - Integration management
        - Restrict who can manage webhooks and slash commands
- Administration tools
- Backend infrastructure
- Glossary

Bulk loading data

Large quantities of data can be imported from a JSONL (https://jsonlines.org) file into Mattermost at the command line using the bulk loading feature. This feature is most suitable for migrating data from an existing system, or for pre-populating a new installation with data.

You can import the following data types:

- Teams
- Threaded discussions
- Channels (public and private)
- Users
- Users' team memberships
- Users' channel memberships
- Users' notification preferences
- Users' custom status
- Posts (regular, non-reply posts)
- Posts' replies
- Posts' reactions
- Posts' file attachments
- Direct message and group message channels
- Direct message and group message channels' read/unread status
- Direct messages and group messages
- Direct messages from a user to themselves
- Permissions schemes
- Custom emoji

- Bot users

Importing additional types of posts is not yet supported.

About the bulk loading command

The bulk loading command is interruptible and idempotent

If the import is interrupted for any reason, it continues from where it left off the next time you run it. You can run the command repeatedly with the same data file, and the data is imported only once. Posts with matching timestamps to incoming posts will have their attachments replaced by the incoming data. Prior to v5.20 any updates to posts with matching timestamps were appended to older posts.

You can run the bulk loading command on a live system

Although you don't need to shut down Mattermost to run the command, changes made by users of the system between runs can be overwritten if the corresponding fields exist in the data file.

Some data fields are optional

Not all fields are mandatory. If an optional field is missing from the object that is being imported, the field's current value in the database is not changed.

The bulk loading command is not a synchronization tool

You cannot use the bulk loading command to remove any objects or their fields from the Mattermost database. The command only creates or overwrites fields.

Important

The bulk loading command runs in the mmctl and operates in the security context of the mmctl. This means it has full permissions to access and alter everything in the Mattermost database.

Bulk load data

Before running the bulk loading command, you must first create a JSONL (https://jsonlines.org) file that contains the data that you want to import in

your Mattermost directory. The file can have any name, but in this example it's called `data.jsonl`. The format of the file is described in the data-format section.

Next, zip it by running the `zip -r data.zip data.jsonl` command.

Using mmctl local mode

From Mattermost v9.5, the mmctl bulk import process command in local mode supports processing an import file without uploading it to the server.

Run `mmctl import process --bypass-upload <file>.zip --local` to start your import and enable the Mattermost server to read from the file directly.

Not using mmctl local mode

If you're not running mmctl commands in local mode:

1. Upload the ZIP file to the database by running the mmctl import upload command. For example: `mmctl import upload data.zip`. After uploading, two IDs are returned: the first line contains the upload session ID, and the second line contains the filename.
2. Confirm that the file is uploaded and ready for use by running the mmctl import list available command.
3. Import your uploaded file by running the mmctl import process command using the upload session ID (not the filename). For example: `mmctl import process <upload_session_id>_data.zip` where `<upload_session_id>` is the upload session ID returned from the upload command.

Data format

The input data file must be a valid [JSONL (https://jsonlines.org)](https://jsonlines.org) file with the following objects, each on its own line in the file. The objects must occur in the file in the order listed.

Version

Mandatory. The Version object must be the first line in the file, and must occur only once. The version is the version of the bulk importer tool, which is currently `1`.

Scheme

Optional. If present, Scheme objects must occur after the Version object but before any Team objects.

Emoji

Optional. If present, Emoji objects must occur after the Version objects but before any Team objects.

Team

Optional. If present, Team objects must occur after any Scheme objects and before any Channel objects.

Channel

Optional. If present, Channel objects must occur after all Team objects and before any User objects.

User

Optional. If present, User objects must occur after the Team and Channel objects in the file and before any Post objects. Each User object defines the teams and channels that the user is a member of. If the corresponding teams and channels are not in the data file, then they must exist in the Mattermost database.

Post

Optional. If present, Post objects must occur after the last User object but before any DirectChannel objects. Each Post object defines the team, the channel, and the username of the user who posted the message. If the corresponding team, channel, or user are not in the data file, then they must exist in the Mattermost database.

DirectChannel

Optional. If present, DirectChannel objects must occur after all Post objects in the file and before any DirectPost objects.

DirectPost

Optional. If present, DirectPost objects must occur after all other objects in the file. Each DirectPost object defines the usernames of the channel members and the username of the user who posted the message. If the corresponding usernames are not in the data file, then they must exist in the Mattermost database.

With the exception of the Version object, each object has a field or a combination of fields that is used as the unique identifier of that object. The bulk loader uses the unique identifier to determine if the object being imported is a new object or an update to an existing object.

The identifiers for each object are listed in the following table:

Objects and their unique identifiers

Object	Unique Identifier
Version	Not Applicable
Scheme	*name*
Role	*name*
Emoji	*name*
Team	*name*
Channel	*name*, *team*
User	*username*
UserNotifyProps	*username*
UserTeamMembership	*team*, *username*
UserChannelMembership	*team*, *channel*, *username*
Post	*channel*, *message*, *create_at*
Reply	*post*, *message*, *create_at*
Reaction	*post*, *emoji_name*, *create_at*

Object	Unique Identifier
Attachment	*path*
DirectChannel	*members*
DirectPost	*channel_members*, *user*, *message*, *create_at*

The following fragment is from a file that imports two teams, each with two channels, many users, and many posts.

```
 1{ "type": "version", ... }  2{ "type": "team", "team": { "name":
"TeamA", ...} }  3{ "type": "team", "team": { "name": "TeamB", ...} }  4{
"type": "channel", "channel": { "team": "TeamA", "name":
"channel_a1", ...} }  5{ "type": "channel", "channel": { "team": "TeamA",
"name": "channel_a2", ...} }  6{ "type": "channel", "channel": { "team":
"TeamB", "name": "channel_b1", ...} }  7{ "type": "channel", "channel": {
"team": "TeamB", "name": "channel_b2", ...} }  8{ "type": "user", "user": {
"username": "user001", ...} }  9{ "type": "user", "user": { "username":
"user002", ...} } 10{ "type": "user", "user": { "username":
"user003", ...} } 11{ "type": "user", ... } 12{ "type": "user", ... } 13{
"type": "user", ... } 14. 15. 16. 17{ "type": "post", { "team": "TeamA",
"name": "channel_a1", "user": "user001", ...} } 18{ "type": "post", {
"team": "TeamA", "name": "channel_a1", "user": "user001", ...} } 19{
"type": "post", { "team": "TeamA", "name": "channel_a1", "user":
"user001", ...} } 20. 21. 22.
```

Version object

The Version object must be the first object in the data file, and can appear only once. The version represents the version of the bulk import tool and currently is `1`.

Example Version object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "version",  "version": 1 }
```

Fields of the Version object

Field name	Type	Description	Validated	Mandatory
type	string	Must be the string "version"	Yes	Yes
version	number	Must be the number 1.	Yes	Yes

Scheme object

Scheme objects represent Permissions Schemes in the Mattermost permissions system. If present, Scheme objects must occur after the Version object and before any Team objects.

Example Scheme object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "scheme",  "scheme": {  "name": "custom_scheme_name",
"display_name": "Custom Scheme Name",  "description": "This is a custom
override scheme.",  "scope": "team",  "default_team_admin_role": {  "name":
"custom_scheme_team_admin_role",  "display_name": "Custom Scheme Team Admin
Role",  "description": "This is the default team admin role for the custom
scheme.",  "permissions": ["add_user_to_team", "manage_team_roles"],  },
"default_team_user_role": {  "name": "custom_scheme_team_user_role",
"display_name": "Custom Scheme Team User Role",  "description":
"This is the default team user role for the custom scheme.",
"permissions": ["create_public_channel", "create_private_channel"],  },
"default_channel_admin_role": {  "name":
"custom_scheme_channel_admin_role",  "display_name": "Custom Scheme Channel
Admin Role",  "description":
"This is the default channel admin role for the custom scheme.",
"permissions": ["manage_private_channel_members",
"manage_channel_roles"],  },  "default_channel_user_role": {  "name":
"custom_scheme_channel_user_role",  "display_name": "Custom Scheme Channel
User Role",  "description": "This is the default channel user role for the
custom scheme.",  "permissions": ["manage_public_channel_members",
"manage_public_channel_properties"],  },  } }
```

Fields of the Scheme object

Field name	Type	Description	Validated	Mandatory
name	string	The scheme name. Must start with and contain only lowercase letters `([a-z0-9])` or _, and must be between 2-64 characters in length.	Yes	Yes
display_name	string	The display name for the scheme.	Yes	Yes
scope	string	The scope for the scheme. Must be either "team" or "channel".	Yes	Yes
description	string	The description of the scheme.	Yes	No
default_team_admin_role	**Role** object	The default role applied to team admins in teams using this scheme. This field is mandatory if the scheme scope is set to "team", otherwise must **not** be present.	Yes	No
default_team_user_role	**Role** object	The default role applied to Team Users in teams using this scheme. This field is mandatory if the	Yes	No

		scheme scope is set to "team", otherwise must **not** be present.		
default_channel_admin_role	**Role** object	The default role applied to channel admins in channels using this scheme. This field is mandatory for both "team" and "channel" scope schemes.	Yes	Yes
default_channel_user_role	**Role** object	The default role applied to Channel Users in channels using this scheme. This field is mandatory for both "team" and "channel" scope schemes.	Yes	Yes

Fields of the Role object

This object is a member of the Scheme object.

Field name	Type	Description	Validated	Mandatory
name	string	The scheme name.	Yes	Yes
display_name	string	The display name for the scheme.	Yes	Yes
description	string	The description of the scheme.	Yes	No
permissions	array	The permissions the role should grant. This is an array of strings where the strings are the names of individual permissions in the Mattermost permissions system.	Yes	No

Emoji object

Emoji objects represent custom Emoji. If present, Emoji objects must occur after the Version object and before any Team objects.

Example Emoji object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{ "type": "emoji", "emoji": {  "name": "custom-emoji-troll",  "image":
"bulkdata/emoji/trollolol.png"  } }
```

Fields of the Emoji object

Field name	Type	Description	Validated	Mandatory
name	string	The emoji name.	Yes	Yes
image	string	The path (either absolute or relative to the current working directory) to the image file for this emoji.	No	Yes

Team object

If present, Team objects must occur after the Version object and before any Channel objects.

Example Team object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{ "type": "team", "team": {  "name": "team-name",  "display_name": "Team
Display Name",  "type": "O",  "description": "The Team Description",
"allow_open_invite": true  } }
```

Fields of the Team object

Field name	Type	Description	Validated	Mandatory
name	string	The team name.	Yes	Yes
display_name	string	The display name for the team.	Yes	Yes
type	string	The type of team. Can have one the following values: `O` for an open team `I` for an invite-only team.	Yes	Yes
description	string	The team description.	Yes	No

allow_open_invite	bool	Whether to allow open invitations. Must have one of the following values:
		`true`
		`false`
			Yes	No

scheme	string	The name of the Scheme that should be applied to this team.	No [1]	No
[1] Not validated, but an error occurs if no such scheme exists when running in apply mode.				

Channel object

If present, Channel objects must occur after all Team objects and before any User objects.

Example Channel object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "channel",  "channel": {  "team": "team-name",  "name": "channel-name",  "display_name": "Channel Name",  "type": "O",  "header": "The Channel Header",  "purpose": "The Channel Purpose",  } }
```

Fields of the Channel object

Field name	Type	Description	Validated	Mandatory
team	string	The name of the team this channel belongs to.	No [1]	Yes
name	string	The name of the channel. Must start with and contain only lowercase letters `([a-z0-9])` or `-` or _.	Yes	Yes
display_name	string	The display name for the channel.	Yes	yes
type	string	The type of channel. Can have one the following values: `O` for a public channel. `P` for a private channel.	Yes	Yes
header	string	The channel header.	Yes	No
purpose	string	The channel purpose.	Yes	No
scheme	string	The name of the Scheme that should be applied to this team.	No [1]	No
[1] Not validated, but an error occurs if no such team/scheme exists when running in apply mode.				

User object

If present, User objects must occur after the Team and Channel objects in the file and before any Post objects.

Example User object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "user",  "user": {  "profile_image": "profile-picture.png",
"username": "username",  "email": "email@example.com",  "auth_service":
"",  "password": "passw0rd",  "nickname": "bobuser",  "first_name": "Bob",
"last_name": "User",  "position": "Senior Developer",  "roles":
"system_user",  "locale": "pt_BR",  "teams": [  {  "name": "team-name",
"theme": "{  \"awayIndicator\":\"#DBBD4E\",  \"buttonBg\":\"#23A1FF\",
\"buttonColor\":\"#FFFFFF\",  \"centerChannelBg\":\"#ffffff\",
\"centerChannelColor\":\"#333333\",  \"codeTheme\":\"github\",
\"linkColor\":\"#2389d7\",  \"mentionBg\":\"#2389d7\",  \"mentionColor\":
\"#ffffff\",  \"mentionHighlightBg\":\"#fff2bb\",  \"mentionHighlightLink\":
\"#2f81b7\",  \"newMessageSeparator\":\"#FF8800\",  \"onlineIndicator\":
\"#7DBE00\",  \"sidebarBg\":\"#fafafa\",  \"sidebarHeaderBg\":\"#3481B9\",
\"sidebarHeaderTextColor\":\"#ffffff\",  \"sidebarText\":\"#333333\",
\"sidebarTextActiveBorder\":\"#378FD2\",  \"sidebarTextActiveColor\":
\"#111111\",  \"sidebarTextHoverBg\":\"#e6f2fa\",  \"sidebarUnreadText\":
\"#333333\",  }",  "roles": "team_user team_admin",  "channels": [  {
"name": "channel-name",  "roles": "channel_user",  "notify_props": {
"desktop": "default",  "mark_unread": "all"  }  }  ]  }  ]  } }
```

Fields of the User object

Field name	Type	Description	Validated	Mandatory
profile_image	string	The user's profile image. This must be an existing file path.	Yes	No
username	string	The user's username. This is the unique identifier for the user.	Yes	Yes
email	string	The user's email address.	Yes	Yes
auth_service	string	The authentication service to use for this user account. If not provided, it defaults to password-based authentication. Must be one of the following values: `""` or not provided - password authentication. `"gitlab"` - GitLab authentication. `"ldap"` - LDAP authentication (Enterprise and Professional) `"saml"` - Generic SAML based authentication (Enterprise) `"google"` - Google OAuth authentication (Enterprise) `"entra id"` - Microsoft Entra ID OAuth Authentication (Enterprise)	No	No
auth_data	string	The authentication data if `auth_service` is used.	No	No

		The value depends on the `auth_service` that is specified. The data comes from the following fields for the respective auth_services: `""` or not provided - must be omitted. `"gitlab"` - The value of the Id attribute provided in the GitLab auth data. `"ldap"` - The value of the LDAP attribute specified as the "ID Attribute" in the Mattermost LDAP configuration. `"saml"` - The value of the SAML Email address attribute. `"google"` - The value of the OAuth Id attribute. `"office365"` - The value of the OAuth Id attribute.		
password	string	A password for the user. Can be present only when password-based authentication is used. When password-based authentication is used and the password is not present, the bulk loader generates a password.	Yes	No
nickname	string	The user’s nickname.	Yes	No
first_name	string	The user’s first name.	Yes	No
last_name	string	The user’s last name.	Yes	No
position	string	The user’s position.	Yes	No

roles	string	The user's roles. Must be one of the following values: `"system_user"` `"system_admin system_user"`	Yes	No
locale	string	The user's locale. This must be a valid locale for which Mattermost has been localised.	No	No
delete_at	int64	Timestamp for when the user was deactivated.	No	No
teams	array	The teams which the user will be made a member of. Must be an array of **UserTeamMembership** objects.	Yes	No
theme	string	The user's theme. Formatted as a Mattermost theme string.	No	No
military_time	string	How times should be displayed to this user. Must be one of the following values: `"true"` - Use 24 hour clock. `"false"` - Use 12 hour clock.	No	No
collapse_previews	string	Whether to collapse or expand link previews by default. Must be one of the following values: `"true"` - Collapsed by default.	No	No

		`"false"` - Expanded by default.		
message_display	string	Which style to use for displayed messages. Must be one of the following values: `"clean"` - Use the standard style. `"compact"` - Use the compact style.	No	No
channel_display_mode	string	How to display channel messages. Must be one of the following values: `"full"` - Use the full width of the screen. `"centered"` - Use a fixed width, centered block.	No	No
tutorial_step	string	Where to start the user tutorial. Must be one of the following values: `"1"`, `"2"` or `"3"` - Start from the specified tutorial step. `"999"` - Skip the user tutorial.	No	No

use_markdown_preview	bool	Enable preview of message markdown formatting. Can have one the following values:
		`"True"`
		`"False"`
			Yes	Yes

use_formatting	bool	Enable post formatting for links, emoji, text styles and line breaks. Can have one the following values:	Yes	Yes

		`"True"` `"False"`		

show_unread_section	bool	Enable showing unread messages at top of channel sidebar. Can have one the following values:
		`"True"`
		`"False"`
			Yes	Yes

email_interval	string	Specify an email batching interval during bulk import. Can have one of the following values: `"immediately"` - Emails are sent immediately. `"fifteen"` - Emails are batched and sent every 15 minutes. `"hour"` - Emails are batched and sent every hour.	Yes	Yes
notify_props	**UserNotifyProps** object	The user's notify preferences, as defined by the **UserNotifyProps** object.	Yes	No

Fields of the UserNotifyProps object

This object is a member of the User object.

Field name	Type	Description	Validated	Mandatory
desktop	string	Preference for sending desktop notifications. Must be one of the following values: `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.	Yes	No
desktop_sound	string	Preference for whether desktop notification sound is played. Must be one of the following values: `"true"` - Sound is played. `"false"` - Sound is not played.	Yes	No
email	string	Preference for email notifications. Must be one of the following values: `"true"` - Email notifications are sent based on the email_interval setting `"false"` - Email notifications are not sent.	No	No
mobile	string	Preference for sending mobile push notifications. Must be one of the following values: `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.	Yes	No
mobile_push_status	string	Preference for when push notifications are	Yes	No
[1] Not validated, but an error occurs if no such team exists when running in apply mode.				

		triggered. Must be one of the following values: `"online"` - When online, away or offline. `"away"` - When away or offline. `"offline"` - When offline.		
channel	string	Whether @all, @channel and @here trigger mentions. Must be one of the following values: `"true"` - Mentions are triggered. `"false"` - Mentions are not triggered.	Yes	No
comments	string	Preference for reply mention notifications. Must be one of the following values: `"any"` - Trigger notifications on messages in reply threads that the user starts or participates in. `"root"` - Trigger notifications on messages in threads that the user starts. `"never"` - Do not trigger notifications on messages in reply threads unless the user is mentioned.	Yes	No
mention_keys	string	Preference for custom non-case sensitive words that trigger mentions. Words must be separated by commas.	No	No
[1] Not validated, but an error occurs if no such team exists when running in apply mode.				

Fields of the UserTeamMembership object

This object is a member of the User object.

Field name	Type	Description	Validated	Mandatory
name	string	The name of the team this user should be a member of.	No [1]	Yes
theme	string	The user's theme for the specified team. Formatted as a Mattermost theme string.	Yes	No

roles	string	The roles the user should have within this team. Must be one of the following values:
		`"team_user"`
		`"team_admin team_user"`
			Yes	No

channels	array	The channels within this team that the user should be made a member of. Must be an array of **UserChannelMembership** objects.	Yes	No
[1] Not validated, but an error occurs if no such team exists when running in apply mode.				

Fields of the UserChannelMembership object

This object is a member of the TeamMembership object.

Field name	Type	Description	Validated	Mandatory
name	string	The name of the channel in the parent team that this user should be a member of.	No [1]	Yes
roles	string	The roles the user should have within this channel. Must be one of the following values: `"channel_user"` `"channel_user channel_admin"`	Yes	No
notify_props	object	The notify preferences for this user in this channel. Must be a **ChannelNotifyProps** object	Yes	No
favorite	boolean	Whether to favorite the channel. Must be one of the following values: `"true"` - Yes. `"false"` - No.	No	No
[1] Not validated, but an error occurs if the parent channel does not exist when running in apply mode.				

Fields of the ChannelNotifyProps object

This object is a member of the ChannelMembership object.

Field name	Type	Description	Validated	Mandatory
desktop	string	Preference for sending desktop notifications. Must be one of the following values: `"default"` - Global default. `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.	Yes	No
mobile	string	Preference for sending mobile notifications. Must be one of the following values: `"default"` - Global default. `"all"` - For all activity. `"mention"` - Only for mentions. `"none"` - Never.	Yes	No
mark_unread	string	Preference for marking channel as unread. Must be one of the following values: `"all"` - For all unread messages. `"mention"` - Only for mentions.	Yes	No

Post object

If present, Post objects must occur after the last User object in the file, but before any DirectChannel objects.

Example Post object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "post",  "post": {  "team": "team-name",  "channel": "channel-
name",  "user": "username",  "message": "The post message",  "props": {
"attachments": [{  "pretext": "This is the attachment pretext.",  "text":
"This is the attachment text."  }]  },  "create_at": 140012340013,
"flagged_by": [  "username1",  "username2",  "username3"  ],  "replies":
[{  "user": "username4",  "message": "The reply message",  "create_at":
140012352049,  "attachments": [{  "path": "/some/valid/file/path/
1"  }],  }, {  "user": "username5",  "message": "Other reply message",
"create_at": 140012353057,  }],  "reactions": [{  "user": "username6",
"emoji_name": "+1",  "create_at": 140012356032,  }, {  "user":
"username7",  "emoji_name": "heart",  "create_at": 140012359034,  }],
"attachments": [{  "path": "/some/valid/file/path/1"  }, {  "path": "/some/
valid/file/path/2"  }]  } }
```

Fields of the Post object

Field name	Type	Description	Validated	Mandatory
team	string	The name of the team that this post is in.	No [1]	Yes
channel	string	The name of the channel that this post is in.	No [2]	Yes
user	string	The username of the user for this post.	No [3]	Yes
message	string	The message that the post contains.	Yes	Yes
props	object	The props for a post. Contains additional formatting information used by integrations and bot posts. For a more detailed explanation see the message attachments documentation.	Yes	Yes
create_at	int	The timestamp for the post, in milliseconds since the Unix epoch.	Yes	Yes
flagged_by	array	Must contain a list of members who have flagged the post.	No	No
replies	array	The posts in reply to this post. Must be an array of Reply objects.	Yes	No
reactions	array	The emoji reactions to this post. Must be an array of Reaction objects.	Yes	No
attachments	array		Yes	No

[1] Not validated, but an error occurs if the team does not exist when running in apply mode.
[2] Not validated, but an error occurs if the channel does not exist in the corresponding team when running in apply mode.
[3] Not validated, but an error occurs if the user does not exist when running in apply mode.

		File attachments associated with this post. Must be an array of Attachment objects.		
[1] Not validated, but an error occurs if the team does not exist when running in apply mode. [2] Not validated, but an error occurs if the channel does not exist in the corresponding team when running in apply mode. [3] Not validated, but an error occurs if the user does not exist when running in apply mode.				

Fields of the Reply object

This object is a member of the Post/DirectPost object.

Field name	Type	Description	Validated	Mandatory
user	string	The username of the user for this reply.	No [3]	Yes
message	string	The message that the reply contains.	Yes	Yes
create_at	int	The timestamp for the reply, in milliseconds since the Unix epoch.	Yes	Yes
flagged_by	array	Must contain a list of members who have flagged the post.	No	No
reactions	array	The emoji reactions to this post. Must be an array of Reaction objects.	Yes	No
attachments	array	The file attachments to this post. Must be an array of Attachment objects.	Yes	No

Fields of the Reaction object

This object is a member of the Post/DirectPost object.

Field name	Type	Description	Validated	Mandatory
user	string	The username of the user for this reply.	No [3]	Yes
emoji_name	string	The emoji of the reaction.	Yes	Yes
create_at	int	The timestamp for the reply, in milliseconds since the Unix epoch.	Yes	Yes

Fields of the Attachment object

This object is a member of the Post/DirectPost object.

Field name	Type	Description	Validated	Mandatory
path	string	The path to the file to be attached to the post.	No [1]	Yes
[1] Not validated, but an error occurs if the file path is not found or accessible when running in apply mode.				

DirectChannel object

A direct channel can have from two to eight users as members of the channel. If there are only two members, Mattermost treats it as a Direct Message channel. If there are three or more members, Mattermost treats it as a Group Message channel.

Example DirectChannel object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "direct_channel",  "direct_channel": {  "members": [
"username1",  "username2",  "username3"  ],  "header": "The Channel
Header",  "favorited_by": [  "username1",  "username2",
"username3"  ]  } }
```

Fields of the DirectChannel object

Field name	Type	Description	Validated	Mandatory
members	array	Must contain a list of members, with a minimum of two usernames and a maximum of eight usernames.	No [1]	Yes
header	string	The channel header.	Yes	No
favorited_by	array	Must contain a list of members who have favorited the channel.	No	No
[1] Not validated, but an error occurs if one or more of the users don't exist when running in apply mode.				

DirectPost object

DirectPost objects must occur after all other objects in the file.

Example DirectPost object

For clarity, the object is shown using regular JSON formatting, but in the data file it cannot be spread across several lines. It must be all on one line.

```
{  "type": "direct_post",  "direct_post": {  "channel_members": [
"username1",  "username2",  "username3",  ],  "user": "username2",
"message": "Hello Group Channel",  "create_at": 140012340013,
"flagged_by": [  "username1",  "username2",  "username3"  ],  "replies":
[{  "user": "username4",  "message": "The reply message",  "create_at":
140012352049,  }, {  "user": "username5",  "message": "Other reply
message",  "create_at": 140012353057,  }],  "reactions": [{  "user":
"username6",  "emoji_name": "+1",  "create_at": 140012356032,  }, {
"user": "username7",  "emoji_name": "heart",  "create_at":
140012359034,  }]  } }
```

Fields of the DirectPost object

Field name	Type	Description	Validated	Mandatory
channel_members	array	Must contain a list of members, with a minimum of two usernames and a maximum of eight usernames.	No [1]	Yes
user	string	The username of the user for this post.	No [2]	Yes
message	string	The message that the post contains.	Yes	Yes
create_at	int	The timestamp for the post, in milliseconds since the Unix epoch.	Yes	Yes
flagged_by	array	Must contain a list of members who have flagged the post.	No	No
replies	array	The posts in reply to this direct post. Must be an array of Reply objects.	Yes	No
reactions	array	The emoji reactions to this direct post. Must be an array of Reaction objects.	Yes	No
attachments	array	The attachments to this direct post. Must be an array of Attachment objects.	Yes	No
[1] Not validated, but an error occurs if no channels contain an identical list when running in apply mode. [2] Not validated, but an error occurs if the user does not exist when running in apply mode.				

Common issues

Run the bulk import command as the *mattermost* user. Running it as *root* or any other user will cause issues with file permissions on imported attachments.

Ensure that file attachments are enabled, that you have enough free space in your file storage system to support the incoming attachments, and that your maximum file size is appropriate.

Make sure you have enough free space for logs on the Mattermost server as well as free space on the database server for both the database itself and transaction logs.

Disable anti-virus or any other plugins that might interfere with attachment uploading. They could potentially block uploading of attachments and cause the import to fail if configured incorrectly. If you need anti-virus scanning, scan the attachment folder before the import.

Troubleshooting

Running bulk loading tool hangs and doesn’t complete

If you have Bleve search indexing enabled, temporarily disable it in **System Console > Experimental > Bleve** and run the command again.

Bleve does not support multiple processes opening and manipulating the same index. Therefore, if the Mattermost server is running, an attempt to run the bulk loading tool will lock when trying to open the indeces.

On this page

- Bulk loading data
  - About the bulk loading command
  - Bulk load data
    - Using mmctl local mode
    - Not using mmctl local mode
  - Data format
  - Version object
    - Example Version object

    - Fields of the Version object
  - Scheme object
    - Example Scheme object
    - Fields of the Scheme object
    - Fields of the Role object
  - Emoji object
    - Example Emoji object
    - Fields of the Emoji object
  - Team object
    - Example Team object
    - Fields of the Team object
  - Channel object
    - Example Channel object
    - Fields of the Channel object
  - User object
    - Example User object
    - Fields of the User object
    - Fields of the UserNotifyProps object
    - Fields of the UserTeamMembership object
    - Fields of the UserChannelMembership object
    - Fields of the ChannelNotifyProps object
  - Post object
    - Example Post object
    - Fields of the Post object
    - Fields of the Reply object
    - Fields of the Reaction object
    - Fields of the Attachment object
  - DirectChannel object
    - Example DirectChannel object
    - Fields of the DirectChannel object
  - DirectPost object
    - Example DirectPost object

    - Fields of the DirectPost object
- Common issues
- Troubleshooting
    - Running bulk loading tool hangs and doesn’t complete

Certificate-based authentication (Experimental)

Certificate-based authentication (CBA) is available as an experimental feature to identify a user or a device before granting access to Mattermost and provide an additional layer of security to access the system.

Follow these steps to configure user CBA for your browser and Mattermost Desktop Apps. Support for the Mattermost iOS and Android Apps is planned. It is expected that you can manage certificate distribution for each personal device (BYOD) and their life cycle management with a service like OpenSSL (https://www.openssl.org/).

Before you begin, follow the official guides to install Mattermost on your system, including NGINX configuration as a proxy with SSL and HTTP/2, and a valid SSL certificate such as Let's Encrypt.

Set up mutual TLS authentication for the Web App

This is the first step for setting up certificate-based authentication. If you haven't set up mutual TLS authentication yet, see our documentation to learn more.

Set up Mattermost server to log in with a client certificate

1. Make sure your Mattermost server is licensed with a valid Enterprise license.
2. In `ExperimentalSettings` of the `config.json` file, set `ClientSideCertEnable` to `true` and `ClientSideCertCheck` to one of the following values:

- `primary` - After the client side certificate is verified, user's email is retrieved from the certificate and is used to log in without a password.

- `secondary` - After the client side certificate is verified, user's email is retrieved from the certificate and matched against the one supplied by the user. If they match, the user logs in with regular email/password credentials.

The `config.json` file should then have the following lines

```
"ExperimentalSettings": { "ClientSideCertEnable": true,
"ClientSideCertCheck": "secondary" },
```

3. Restart the Mattermost server.

On Ubuntu 14.04 and RHEL 6:

```
sudo restart mattermost
```

On Ubuntu 16.04, Debian Stretch, and RHEL 7:

```
sudo systemctl restart mattermost
```

4. Go to `https://example.mattermost.com` and try to log in. The server should require the x.509 cert to have an `emailAddress` equal to the Mattermost user's email.

On this page

- Certificate-based authentication (Experimental)
  - Set up mutual TLS authentication for the Web App
  - Set up Mattermost server to log in with a client certificate

- Mattermost documentation

Using the System Console, Mattermost Enterprise and Professional customers can migrate OAuth configuration to the OpenID Connect standard. A one-time, one-click conversion tool is available within the existing **OAuth 2.0** page. You can also go to **System Console > Authentication > OpenID Connect**.

Select **Convert to OpenID Connect** to migrate your active service provider configuration to the new standard. No further changes are required.

> Note
>
> - Inactive service providers configured for OAuth 2.0 are also migrated to the new OpenID Connect standard.

Connected workspaces

Communicate across organizations, as well as external partners and vendors using Mattermost by synchronizing messages, emoji reactions, and file sharing in real-time through secured, connected Mattermost workspaces.

Connected workspaces in Mattermost behave like regular public and private channels and offer the same user experience and functionality. All members using secure connections, including local members and remote members, can send and receive channel messages, use emojis to react to messages, share files, and search message history. Content is synchronized across all participating Mattermost instances.

A channel's permissions and access continues to be governed by each server separately. Advanced access control permissions can be applied to a shared channel, and be in effect on the local Mattermost server while not being in effect on a remote Mattermost server.

Set up connected workspaces

The process of connecting Mattermost workspaces involves the following 5 steps:

1. Ensure that all Mattermost Enterprise servers are running v10.2 or later.
2. Enable the connected workflows functionality for each Mattermost Enterprise instance you want to connect.
3. System admins must create a secure and trusted connection with other Mattermost Enterprise instances using the System Console or slash commands. This process involves creating a password-protected, encrypted invitation, creating a strong decryption password, then sending the invitation and password to the system admin of a remote Mattermost instance.
4. When a remote system admin receives the invitation, they must accept the invitation using the System Console or slash commands.

5. Once a trusted relationship is established between 2 Mattermost servers, system admins can share specific public or private channels with secure connections.

> Note
>
> - System admins can only create secure connections with other Mattermost Enterprise instances, and can only share channels with secured connections.
> - System admins must use Mattermost to generate a password-protected encrypted invitation code. However, sending secure connection invitations is not completed using Mattermost. System admins must have an independent way to extend the secure connection invitation, such as by email.
> - A channel shared by a host organization cannot be shared from the receiving organization to another organization. Organizations can't share a channel originating from another organization.

Enable connected workflows

System admins must enable connected workspaces functionality for their Mattermost instance. Ensure the following configuration settings are set to `true` in `config.json`:

- `ConnectedWorkspacesSettings.EnableRemoteClusterService = true`
- `ConnectedWorkspacesSettings.EnableSharedChannels = true`

See the Site Configuration Settings documentation for details.

> Note
>
> Following an upgrade to Mattermost v10.2 or later, existing configuration values for shared channels, including `EnableSharedChannels` and `EnableRemoteClusterService` are automatically converted to connected workspace configuration settings in the `config.json` file. The deprecated shared channels experimental settings remain in the `config.json` file to support backwards compatibility.

Create a secure connection

System Console

Only system admins can create workspace connections using the System Console.

1. Go to **Site Configuration > Connected Workspaces**.
2. Under **Connected Workspaces**, select **Add a connection**, and then select **Create a connection**.
3. Specify the **Organization Name** for this connection. The remote system admin must specify this name when accepting a connection invitation.
4. Select the **Destination Team** as the default team where shared channels will be added. This team will be used as the default location for organizing shared channels from this connected workspace.
5. Select **Save**.

An invitation consisting of a password-protected AES 256-bit encrypted code blob is generated. The connection is labeled as **Connection Pending** until the remote system admin accepts the invitation.

Slash Commands

By default, only system admins can use slash commands to create workspace connections. You can grant the ability to **Manage Shared Channels** and **Managed Secure Connections** to Mattermost users by modifying permissions of the system scheme or team override scheme.

System admins can run the following slash command to create a secure connection invitation:

`/secure-connection create --name <--displayname> --password`

For example:

`/secure-connection create --name AcmeUS --displayname "AcmeUSA" --password examplepassword`

This slash command creates an invitation consisting of a password-protected AES 256-bit encrypted code blob for a remote Mattermost entity known locally as `AcmeUS` with a password of `examplepassword`. Within Mattermost, this shared connection displays to the local system admin based on the `name` and `displayname` provided.

Extend the invitation

You must use a system, other than Mattermost, to share invitation codes and passwords. We strongly recommend sharing invitation codes separately from passwords to ensure that no one has all of the data necessary to take action if the message were compromised. Ensure the remote Mattermost instance can access your Mattermost workspace URL.

System Console

Once you've created a connection in the System Console, you're prompted to share the invitation code and password with the system admin of the remote Mattermost server you want to connect with. Copy both the invitation code and password to a safe location, then select **Done**.

Slash Commands

Copy the invitation code blob in the System message, then share the code blob and the decryption password to the remote Mattermost system admin you want to securely connect with.

Accept a connection invitation

System Console

1. Go to **Site Configuration > Connected Workspaces**.

2. Under **Connected Workspaces**, select **Add a connection**, and then select **Accept an invitation**.
3. Specify the **Organization Name** for this invitation. This must be the same name specified when creating the connection.
4. Select the **Destination Team** where shared channels will be added. This team will serve as the default location for organizing shared channels from this connected workspace.
5. Paste the encrypted invitation code and password you've been provided to connect with the remote workspace.
6. Select **Accept**.

The system admin who accepts the connection invitation is automatically added to all shared channels.

Slash Commands

Run the following slash command to accept a secure connection invitation from a remote Mattermost instance:

`/secure-connection accept --name --displayname --password --invite [code blob]`

For example:

`/secure-connection accept --name AcmeUS --displayname "AcmeUSA" --password examplepassword --invite [code-blob]`

This slash command accepts a secure connection invitation from `AcmeUS`.

Share channels with secure connections

Once a connection is established between two Mattermost servers, system admins can share channels across secured workspaces.

System Console

1. Under **Shared Channels**, select **Add channels**.
2. Specify the channels you want to share between Mattermost servers.

Shared channels and members of those shared channels display a shared

icon to distinguish them from channels and channel members of the local server.

Slash Commands

Run the following slash command to specify the public or private channels to share:

```
/share-channel invite --connectionID <--readonly>
```

You can extend an invitation that permits remote members to participate in the channel based on their channel and member permissions.

Alternatively, you can extend a read-only invitation to a secure connection by appending the optional `--readonly` parameter to this command. Remote members can't post or reply to messages within shared read-only channels.

> Tip
>
> To convert a read-only shared channel to a participation channel, remove the original secured connection from the channel, then re-extend an invitation to that secure connection while omitting the optional `--readonly` parameter. For example:
>
> ```
> /share-channel invite --connectionID
> ```
>
> This slash command invites the shared connection to the current channel based on its `connectionID`.
>
> See Reviewing Secure Connection Status to find the `connectionID` for a shared connection.

Direct message delivery

From Mattermost v10.10, creating a direct or group message with remote users across connected workspaces is only available when the feature flag `EnableSharedChannelsDMs` is enabled.

When `EnableSharedChannelsDMs` is disabled, the direct message option on a user's profile is disabled and unavailable to ensure users can't attempt direct messages with remote users when connected workspaces isn't enabled on the other Mattermost instance.

Plugin component interaction

From Mattermost v10.10, plugin interactions such as slash commands, interactive buttons, and other plugin-generated components aren't displayed or accessible in shared channels by default to ensure a consistent experience across different Mattermost instances.

System admins can enable the `EnableSharedChannelsPlugins` feature flag to enable these plugin interactions in shared channels. When plugin components are enabled in shared channels, we recommend ensuring that all connected Mattermost instances have the same plugins installed and configured to avoid inconsistent user experiences. Plugin behaviors can vary between instances if different plugin versions or configurations are used.

Remote user discovery

From Mattermost v10.10, remote users across connected workspaces can be discovered for direct or group messages when the feature flag `EnableSyncAllUsersForRemoteCluster` is enabled. This feature implements global user synchronization between connected Mattermost instances, making remote users discoverable without requiring them to post in a shared channel first. When enabled, Mattermost uses cursor-based synchronization to efficiently sync user information between connected instances. The system:

- Synchronizes users in configurable batch sizes to prevent timeouts and reduce memory usage
- Tracks synchronization progress using timestamps to enable efficient resumption
- Filters out users from their original cluster to prevent syncing users back to their home instance
- Only syncs users that have been updated since the last synchronization

The feature includes configuration options for automatically syncing users when connections are established and controlling batch processing sizes for optimal performance.

When `EnableSyncAllUsersForRemoteCluster` is disabled, remote users are only discoverable in the DM/GM creation modal after they have participated in a shared channel.

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

Channel membership synchronization

From Mattermost v10.10, channel membership synchronization between connected workspaces is controlled by the feature flag `EnableSharedChannelsMemberSync`. When this feature flag is enabled, channel membership changes are automatically synchronized across all connected workspaces that share the same channel.

When `EnableSharedChannelsMemberSync` is enabled:

- Users added to a shared channel on one workspace are automatically added to the corresponding shared channel on all connected workspaces
- Users removed from a shared channel on one workspace are automatically removed from the corresponding shared channel on all connected workspaces
- Membership changes are processed in configurable batch sizes to optimize performance and prevent timeouts
- The system uses cursor-based synchronization to efficiently track and sync membership changes

The feature includes a configuration option for controlling batch processing sizes for member synchronization to ensure optimal performance during large membership changes.

When `EnableSharedChannelsMemberSync` is disabled, channel membership changes are not synchronized between connected workspaces, and users must be manually added or removed from shared channels on each workspace.

> Note
>
> Enabling these features can increase the load on your Mattermost server's CPU, memory, and database due to frequent updates, database queries, and API communication. Excessive sync frequency and retries can overwhelm system resources, potentially causing performance degradation or instability. Monitor your system carefully when enabling these features.

Manage connections and invitations

System admins can edit or delete a connected workspace, and review connection status, and regenerate invitation codes and passwords for pending connections.

Edit a connected workspace

In the System Console, system admins can change the **Organization Name**, the **Destination Team** (which serves as the default location for organizing shared channels from this connected workspace), or channels shared with a remote Mattermost instance as well as channels shared with your local Mattermost instance.

1. Under **Connected Workspaces**, identify the connected workspace you want to change.
2. Select the **More**

   •••

   icon to the right of the connected workspace, and then select **Edit**.

Remove all connections from the current channel

Run the following slash command to remove all secure connections from the current channel:

`/share-channel unshare`

This slash command removes all secure connections from the current channel. A System message notifies you that the channel is no longer shared. Secure connections may continue to be invited to other shared channels.

Delete a connected workspace

Deleting a connected server severs the trust relationship between the local Mattermost server and the remote Mattermost server.

From Mattermost v10.10, removing a shared channel from a connected workspace removes the channel from all connected workspaces. The channel is deleted from both the local and remote Mattermost servers. Prior to Mattermost v10.10, removing a shared channel from a connected workspace stops synchronizing the channel with the remote Mattermost server; however, the channel continues to function for local users.

System Console

1. Under **Connected Workspaces**, identify the connected workspace you want to remove.
2. Select the **More**

   •••

   icon to the right of the connected workspace, and then select **Delete**.

Slash Commands

Using slash commands, you can uninvite or delete a secure connection from your Mattermost instance.

Run the following slash command to uninvite a secure connection:

`/share-channel uninvite --connectionID`

This slash command removes a secure connection from the current channel based on its `connectionID`. The secure connection may continue to be invited to other shared channels.

Run the following slash command to delete a secure connection:

```
/secure-connection remove --connectionID
```

For example:

```
/secure-connection remove --connectionID
```

This slash command severs the trust relationship between the local Mattermost server and a remote Mattermost server based on its `connectionID` and removes the secure connection from all shared Mattermost channels.

Review connection status

System Console

Under **Connected Workspaces**, you can review all connected workspaces and their current status as one of: **Connected**, **Offline**, or **Connection Pending**.

Slash Commands

Run the following slash command to review the current status of all secure connections established for your Mattermost instance:

```
/secure-connection status
```

Status details include:

- Connection ID
- Connection URL
- Description
- Invite accepted (Yes/No)
- Online (Yes/No)
- Last ping timestamp (UTC)
- Deleted

Regenerate invitation codes for pending connections

When using the System Console to manage connected workspaces, system admins can re-generate invitation codes and passwords for pending connections.

1. Under **Connected Workspaces**, identify the pending connection whose invitation and password you want to regenerate.
2. Select the **More**

   •••

   icon to the right of the connected workspace, and then select **Regenerate invitation code**.

> Note
>
> Regenerating doesn't invalidate the existing password, and the existing password can continue to be used in addition to the newly-generated password. Once a connection invitation is accepted and the workspace displays a status of **Connected**, invitation codes and passwords can't be regenerated.

Frequently Asked Questions

Are special characters supported in secure connection names?

No. When using slash commands, `--name` can include periods, hyphens, and/or underscores. You must surround `--name` using quotation marks (") when the value contains spaces.

What happens if two Mattermost instances contain different emojis?

In cases where one Mattermost instance has different emojis than another instance, emoji text displays in place of a missing emoji image.

Is a Display Name required for all secure connections?

No. When using slash commands, `--displayname` is optional. When omitted, `--name` is displayed and used instead.

What information is synchronized between connected workspaces?

By default, member status and availability for all members of shared channels is synchronized between connected workspaces.

When a user is added to a shared channel, member status is synchronized within a few seconds of the member's status changing. Status updates aren't immediate and don't necessarily display in real-time.

When using Mattermost in a web browser, Mattermost polls the server every minute. Refreshing the browser page triggers immediate synchronization.

By default, a maximum of 50 messages are synchronized at a time, and this value is configurable.

Channel as well as member status and availability synchronization can be disabled.

From Mattermost v10.10, channel membership can be synchronized between connected workspaces when the feature flag `EnableSharedChannelsMemberSync` is enabled. When a user is added to or removed from a shared channel on one workspace, that membership change is automatically applied to the corresponding shared channel on all connected workspaces. This ensures consistent channel membership across all participating Mattermost instances. Additionally, connected workspaces also synchronize message priority, message acknowledgements, and persistent notifications between connected servers. This ensures that important message indicators and user interactions are consistently reflected across all connected workspace instances.

Do connection interruptions affect message synchronization?

Yes. A System message is posted in the channel visible to all channel members when message synchronization is interrupted for more than 5

minutes. Once connectivity is restored, a full synchronization will happen for all missed messages, including direct messages and channel links.

What happens if two secure connections share the same usernames?

In cases where members share the same usernames across Mattermost secure connections, usernames on the local server instance are appended with the secure connection name of the remote server.

For example, if multiple members named John Smith exist after two Mattermost instances establish a secure connection with one another, all remote John Smith members include their Secure Connection ID following their username to help differentiate members across multiple Mattermost instances.

On this page

- Connected workspaces
- Connected workspaces
  - Set up connected workspaces
  - Enable connected workflows
  - Create a secure connection
    - System Console
    - Slash Commands
    - Extend the invitation
      - System Console
      - Slash Commands
  - Accept a connection invitation
    - System Console
    - Slash Commands
  - Share channels with secure connections
    - System Console
    - Slash Commands
  - Direct message delivery
  - Plugin component interaction
  - Remote user discovery

- Channel membership synchronization
- Manage connections and invitations
  - Edit a connected workspace
  - Remove all connections from the current channel
  - Delete a connected workspace
    - System Console
    - Slash Commands
  - Review connection status
    - System Console
    - Slash Commands
  - Regenerate invitation codes for pending connections
- Frequently Asked Questions
  - Are special characters supported in secure connection names?
  - What happens if two Mattermost instances contain different emojis?
  - Is a Display Name required for all secure connections?
  - What information is synchronized between connected workspaces?
  - Do connection interruptions affect message synchronization?
  - What happens if two secure connections share the same usernames?
- Set up connected workspaces
- Enable connected workflows
- Create a secure connection
  - Extend the invitation
- Accept a connection invitation
- Share channels with secure connections
- Direct message delivery
- Plugin component interaction
- Remote user discovery
- Channel membership synchronization
- Manage connections and invitations
  - Edit a connected workspace
  - Remove all connections from the current channel
  - Delete a connected workspace
  - Review connection status

    - Regenerate invitation codes for pending connections
- Frequently Asked Questions
    - Are special characters supported in secure connection names?
    - What happens if two Mattermost instances contain different emojis?
    - Is a Display Name required for all secure connections?
    - What information is synchronized between connected workspaces?
    - Do connection interruptions affect message synchronization?
    - What happens if two secure connections share the same usernames?

Converting OAuth 2.0 Service Providers to OpenID Connect

Using the System Console, Mattermost Enterprise and Professional customers can migrate OAuth configuration to the OpenID Connect standard. A one-time, one-click conversion tool is available within the existing **OAuth 2.0** page. You can also go to **System Console > Authentication > OpenID Connect**.

Select **Convert to OpenID Connect** to migrate your active service provider configuration to the new standard. No further changes are required.

> Note
>
> - Inactive service providers configured for OAuth 2.0 are also migrated to the new OpenID Connect standard.

Configuring OpenID Connect Single Sign-On

Note

Available only for self-hosted deployments

For details on configuring Mattermost to use a service provider as a Single Sign-on (SSO) service for team creation, account creation, and user sign-in using OpenID Connect, refer to the following documentation:

- OpenID Connect Single Sign-On
- GitLab Single Sign-On
- Google Apps Single Sign-On
- Entra ID Single Sign-On

On this page

- Converting OAuth 2.0 Service Providers to OpenID Connect
  - Configuring OpenID Connect Single Sign-On

Delegated granular administration

Mattermost supports the creation and customization of system administration roles with specific granular permissions and System Console access. This allows senior administrators in large organizations to delegate and de-centralize specialized administration and administrative tasks with specific admin roles.

These admin roles permit granular access to specific areas of the System Console and related API endpoints. These roles enable users to perform certain administrative tasks without requiring access to all system administration areas. These system roles never supersede the user's original role or the user's permissions configured by the Permissions scheme.

> Warning
>
> Even when a role is set to **No Access** or **Read Only** for a System Console page, granting **Can Edit** on any System Console page enables access to the underlying configuration endpoint (`PUT /api/v4/config/patch`). This means a user with write access in one area can modify configuration values across all areas. Administrators should assign **Can Edit** permissions with caution.

Available roles

A system admin can configure the following delegated granular administration roles in the System Console. Each role has a set of default permissions, which can be adjusted as needed.

- **System Manager:** This role can be configured to have read/write permissions in different management areas.
- **User Manager:** This role can be configured to have read/write to all the user management areas and to authentication

- **Custom Group Manager** This role has permissions to create, edit, restore, and delete custom user groups. This role can be used to assign individual users the ability to manage custom groups when **Custom Groups** permissions are removed for **All Members** via **System Console > Permissions > Edit Scheme > Custom Groups**.
- **Viewer:** The Viewer role can view all areas of the System Console, and can be configured with write access where needed.

When a user is assigned a system role, they have role-based access to the System Console and the underlying API endpoints. Each role has a different set of default permissions, and what users can access or view depends on the role they've been assigned.

The table below lists the default permissions for each role. Admins should carefully review and configure these settings to align with their organization's needs. Particular caution should be exercised with Permissions write access, as it enables modifications to the permissions of any role, except for the delegated granular administrator roles.

System role	Read/Write access	Read Only access
System Manager	• User Management - Groups - Teams - Channels - Permissions • Environment • Site Configuration • Integrations	• Edition/License • Reporting • Authentication • Plugins
User Manager	• User Management - Groups - Teams - Channels	• (User Management) Permissions • Authentication
Custom Group Manager	Custom User Groups	N/A
Viewer	N/A	• All pages within the System Console

Assign admin roles

There are two ways to assign roles:

1. In the System Console under **User Management > Delegated Granular Administration**.
2. Using the mmctl tool. This can be done either locally or remotely.

You want to	Using the System Console	Using mmctl
Assign roles	Go to **System Console > User Management > Delegated Granular Administration > Assigned People**	`mmctl permissions role assign [role_name] [username...]`
Grant the System Manager role to a user	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **System Manager** role. 2. Under **Assigned People**, select **Add People**. 3. Search for and select the user name, then select **Add** to grant the System Manager role to that user.	`mmctl permissions role assign system_manager user-name`
Grant the User Manager role to two users	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **User Manager** role. 2. Under **Assigned People**, select **Add People**. 3. Search for and select the two users, then select **Add** to grant the User Manager role to those users.	`mmctl permissions role assign system_user_manager user-name1 user-name2`
Grant the Viewer role to a user	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **Viewer** role.	`mmctl permissions role assign system_read_only_admin user-name`

	2. Under **Assigned People**, select **Add People**. 3. Search for and select the user name, then select **Add** to grant the Viewer role to that user.	
Grant the Custom Group Manager role to two users	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **Custom Group Manager** role. 2. Under **Assigned People**, select **Add People**. 3. Search for and select the two users, then select **Add** to grant the Custom Group Manager role to those users. 4. All users have the ability to create custom user groups by default. When you assign users to the Custom Group Manager role, you must manually remove these permissions from all users by going to **System Console > User Management > Permissions > Edit Scheme**. Under **All Members**, clear all of the **Custom Groups** permissions, including **Create**, **Manage members**, **Edit**, and **Delete**.	`mmctl permissions role assign system_custom_group_admin user-name1 user-name2`
Remove the System Manager role from a single user	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **System Manager** role. 2. Under **Assigned People**, search for the user, then select **Remove**.	`mmctl permissions role unassign system_manager bob-smith`

Edit privileges of admin roles (advanced)

System admins can grant read/write access to other areas of the System Console, as well as remove read/write access (including default access), for all system roles except the Custom Group Manager role.

There are two ways to assign roles:

1. In the System Console under **User Management > Delegated Granular Administration**.
2. Using the mmctl tool. This can be done either locally or remotely.

You want to	Using the System Console	Using mmctl
Edit role privileges	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **System Manager**, **User Manager**, or **Viewer** role. 2. For each set of privileges, select the access level as **Can edit**, **Read only**, or **No access**. Note If you set privilege subsections to different access levels, then the privilege access level displays as **Mixed Access**.	`mmctl permissions add [role_name] [permission...]` `mmctl permissions reset system_read_only_admin`
Grant write access to the Authentication section of the System Console for all users with the User Manager role	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **User Manager** role.	`mmctl permissions add system_user_manager sysconsole_write_authentication`

	2. Under **Privileges > Authentication** select **Can edit**, then select **Save**.	
Grant read-only access to the Authentication section of the System Console for all users with the User Manager role	1. Go to **System Console > User Management > Delegated Granular Administration**. then select the **User Manager** role. 2. Under **Privileges > Authentication** select **Read only**, then select **Save**.	`mmctl permissions remove system_user_manager sysconsole_read_authentication`
Remove write access to the Authentication section of the System Console for all users with the User Manager role	1. Go to **System Console > User Management > Delegated Granular Administration**, then select the **User Manager** role. 2. Under **Privileges > Authentication** select **No access**, then choose **Save**.	`mmctl permissions remove system_user_manager sysconsole_write_authentication`
Reset a role to its default set of permissions	This is completed using the mmctl tool only.	`mmctl permissions reset [role_name]` For example, to reset the permissions of the `system_read_only_admin` role: `mmctl permissions reset system_read_only_admin`

Admin roles and privileges

Roles

- `system_manager`
- `system_user_manager`
- `system_custom_group_admin`
- `system_read_only_admin`

Privileges

System Console section	Permissions
About	• PERMISSION_SYSCONSOLE_READ_ABOUT_EDITION_AND_LICENSE • PERMISSION_SYSCONSOLE_WRITE_ABOUT_EDITION_AND_LICENSE
Reporting	**Site Statistics** • PERMISSION_SYSCONSOLE_READ_REPORTING_SITE_STATISTICS • PERMISSION_SYSCONSOLE_WRITE_REPORTING_SITE_STATISTICS **Team Statistics** • PERMISSION_SYSCONSOLE_READ_REPORTING_TEAM_STATISTICS • PERMISSION_SYSCONSOLE_WRITE_REPORTING_TEAM_STATISTICS **Server Logs** • PERMISSION_SYSCONSOLE_READ_REPORTING_SERVER_LOGS • PERMISSION_SYSCONSOLE_WRITE_REPORTING_SERVER_LOGS
User Management	**Users** • PERMISSION_SYSCONSOLE_READ_USERMANAGEMENT_USERS • PERMISSION_SYSCONSOLE_WRITE_USERMANAGEMENT_USERS **Groups** • PERMISSION_SYSCONSOLE_READ_USERMANAGEMENT_GROUPS • PERMISSION_SYSCONSOLE_WRITE_USERMANAGEMENT_GROUPS **Teams** • PERMISSION_SYSCONSOLE_READ_USERMANAGEMENT_TEAMS • PERMISSION_SYSCONSOLE_WRITE_USERMANAGEMENT_TEAMS

System Console section	Permissions
	Channels • PERMISSION_SYSCONSOLE_READ_USERMANAGEMENT_CHANNELS • PERMISSION_SYSCONSOLE_WRITE_USERMANAGEMENT_CHANNELS **Permissions** • PERMISSION_SYSCONSOLE_READ_USERMANAGEMENT_PERMISSIONS • PERMISSION_SYSCONSOLE_WRITE_USERMANAGEMENT_PERMISSIONS
Environment	**Web Server** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_WEB_SERVER • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_WEB_SERVER **Database** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_DATABASE • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_DATABASE **Elasticsearch** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_ELASTICSEARCH • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_ELASTICSEARCH **File Storage** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_FILE_STORAGE • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_FILE_STORAGE **Image Proxy** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_IMAGE_PROXY • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_IMAGE_PROXY **SMTP** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_SMTP

System Console section	Permissions
	• PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_SMTP **Push Notification Server** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_PUSH_NOTIFICATION_ • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_PUSH_NOTIFICATION_ **High Availability** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_HIGH_AVAILABILITY • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_HIGH_AVAILABILITY **Rate Limiting** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_RATE_LIMITING • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_RATE_LIMITING **Logging** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_LOGGING • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_LOGGING **Session Lengths** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_SESSION_LENGTHS • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_SESSION_LENGTHS **Performance Monitoring** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_PERFORMANCE_MONIT • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_PERFORMANCE_MONI **Developer** • PERMISSION_SYSCONSOLE_READ_ENVIRONMENT_DEVELOPER • PERMISSION_SYSCONSOLE_WRITE_ENVIRONMENT_DEVELOPER

System Console section	Permissions
Site Configuration	**Customization** • PERMISSION_SYSCONSOLE_READ_SITE_CUSTOMIZATION • PERMISSION_SYSCONSOLE_WRITE_SITE_CUSTOMIZATION **Localization** • PERMISSION_SYSCONSOLE_READ_SITE_LOCALIZATION • PERMISSION_SYSCONSOLE_WRITE_SITE_LOCALIZATION **Users and Teams** • PERMISSION_SYSCONSOLE_READ_SITE_USERS_AND_TEAMS • PERMISSION_SYSCONSOLE_WRITE_SITE_USERS_AND_TEAMS **Notifications** • PERMISSION_SYSCONSOLE_READ_SITE_NOTIFICATIONS • PERMISSION_SYSCONSOLE_WRITE_SITE_NOTIFICATIONS **Announcement Banner** • PERMISSION_SYSCONSOLE_READ_SITE_ANNOUNCEMENT_BANNER • PERMISSION_SYSCONSOLE_WRITE_SITE_ANNOUNCEMENT_BANNER **Emoji** • PERMISSION_SYSCONSOLE_READ_SITE_EMOJI • PERMISSION_SYSCONSOLE_WRITE_SITE_EMOJI **Posts** • PERMISSION_SYSCONSOLE_READ_SITE_POSTS • PERMISSION_SYSCONSOLE_WRITE_SITE_POSTS **File Sharing and Downloads** • PERMISSION_SYSCONSOLE_READ_SITE_FILE_SHARING_AND_DOWNLOAD

System Console section	Permissions
	• PERMISSION_SYSCONSOLE_WRITE_SITE_FILE_SHARING_AND_DOWNLOA **Public Links** • PERMISSION_SYSCONSOLE_READ_SITE_PUBLIC_LINKS • PERMISSION_SYSCONSOLE_WRITE_SITE_PUBLIC_LINKS **Notices** • PERMISSION_SYSCONSOLE_READ_SITE_NOTICES • PERMISSION_SYSCONSOLE_WRITE_SITE_NOTICES
Authentication	**Signup** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_SIGNUP • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_SIGNUP **Email** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_EMAIL • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_EMAIL **Password** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_PASSWORD • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_PASSWORD **MFA** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_MFA • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_MFA **AD/LDAP** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_MFA • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_MFA

System Console section	Permissions
	SAML 2.0 • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_SAML • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_SAML **OpenID Connect** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_OPENID • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_OPENID **Guest Access** • PERMISSION_SYSCONSOLE_READ_AUTHENTICATION_GUEST_ACCESS • PERMISSION_SYSCONSOLE_WRITE_AUTHENTICATION_GUEST_ACCESS
Plugin	• PERMISSION_SYSCONSOLE_READ_PLUGINS • PERMISSION_SYSCONSOLE_WRITE_PLUGINS
Integrations	**Integration Management** • PERMISSION_SYSCONSOLE_READ_INTEGRATIONS_INTEGRATION_MANAG • PERMISSION_SYSCONSOLE_WRITE_INTEGRATIONS_INTEGRATION_MANA **Bot Accounts** • PERMISSION_SYSCONSOLE_READ_INTEGRATIONS_BOT_ACCOUNTS • PERMISSION_SYSCONSOLE_WRITE_INTEGRATIONS_BOT_ACCOUNTS **GIF (Beta)** • PERMISSION_SYSCONSOLE_READ_INTEGRATIONS_GIF • PERMISSION_SYSCONSOLE_WRITE_INTEGRATIONS_GIF **CORS** • PERMISSION_SYSCONSOLE_READ_INTEGRATIONS_CORS

System Console section	Permissions
	• PERMISSION_SYSCONSOLE_WRITE_INTEGRATIONS_CORS
Compliance	**Data Retention Policy** • PERMISSION_SYSCONSOLE_READ_COMPLIANCE_DATA_RETENTION_POLI • PERMISSION_SYSCONSOLE_WRITE_COMPLIANCE_DATA_RETENTION_POL **Compliance Export** • PERMISSION_SYSCONSOLE_READ_COMPLIANCE_COMPLIANCE_EXPORT • PERMISSION_SYSCONSOLE_WRITE_COMPLIANCE_COMPLIANCE_EXPORT **Compliance Monitoring** • PERMISSION_SYSCONSOLE_READ_COMPLIANCE_COMPLIANCE_MONITORI • PERMISSION_SYSCONSOLE_WRITE_COMPLIANCE_COMPLIANCE_MONITOR **Custom Terms of Service** • PERMISSION_SYSCONSOLE_READ_COMPLIANCE_CUSTOM_TERMS_OF_SE • PERMISSION_SYSCONSOLE_WRITE_COMPLIANCE_CUSTOM_TERMS_OF_S

System Console section	Permissions
Experimental	**Features** • PERMISSION_SYSCONSOLE_READ_EXPERIMENTAL_FEATURES • PERMISSION_SYSCONSOLE_WRITE_EXPERIMENTAL_FEATURES **Feature Flags** • PERMISSION_SYSCONSOLE_READ_EXPERIMENTAL_FEATURE_FLAGS • PERMISSION_SYSCONSOLE_WRITE_EXPERIMENTAL_FEATURE_FLAGS **Bleve** • PERMISSION_SYSCONSOLE_READ_EXPERIMENTAL_BLEVE • PERMISSION_SYSCONSOLE_WRITE_EXPERIMENTAL_BLEVE

Frequently Asked Questions

Can a User Manager or System Manager reset an administrator's email or password without their knowledge?

This is not possible with the default privileges of these roles. The ability to reset passwords or email addresses of administrators is limited to system admins.

Can a User Manager or System Manager access the configuration file?

Yes. However, they will only have access to read actual values and modify values in accordance with their permissions. If appropriate read permissions do not exist, the default key values will be displayed.

Are all actions of admin roles logged?

Every change made by any admin is included in the audit log.

Can a System Manager change their own permissions or elevate their role?

No. System Managers can't elevate their role, and aren't able to elevate other members' roles.

Can any of the new roles view API keys/passwords or other sensitive information within the System Console (such as SMTP, AWS, Elastic Search)?

No, password information is only visible to system admins and is obfuscated for other roles.

If download links for compliance exports are enabled in the System Console, can a Read Only Admin download the reports?

Only roles that are explicitly granted access to **System Console > Compliance** have access to download compliance reports.

Can any of the new roles force-join Private channels?

Yes at this time they can, however, we will be improving on this behavior in the future with a prompt that lets them know they are entering a private channel. We are also planning on adding a permission which would remove the ability to access Private channels.

Can I create a new role or clone an existing role?

No, but we are actively seeking feedback on this capability.

Can I use an LDAP filter to assign these roles?

No, but we are considering this functionality for a future enhancement.

Can I rename the roles?

This is being considered for future development.

Can a System Manager or User Manager demote or deactivate another Admin or Manager?

A System or User Manager can demote or deactivate another System or User Manager, but can't demote or deactivate a system admin.

Can a System Manager or User Manager assign or unassign admin roles?

Only the system admin has access to edit system roles.

On this page

- Delegated granular administration
  - Available roles
  - Assign admin roles
  - Edit privileges of admin roles (advanced)
  - Admin roles and privileges
    - Roles
    - Privileges
  - Frequently Asked Questions
    - Can a User Manager or System Manager reset an administrator's email or password without their knowledge?
    - Can a User Manager or System Manager access the configuration file?
    - Are all actions of admin roles logged?
    - Can a System Manager change their own permissions or elevate their role?
    - Can any of the new roles view API keys/passwords or other sensitive information within the System Console (such as SMTP, AWS, Elastic Search)?
    - If download links for compliance exports are enabled in the System Console, can a Read Only Admin download the reports?
    - Can any of the new roles force-join Private channels?

- Can I create a new role or clone an existing role?
- Can I use an LDAP filter to assign these roles?
- Can I rename the roles?
- Can a System Manager or User Manager demote or deactivate another Admin or Manager?
- Can a System Manager or User Manager assign or unassign admin roles?

Guest accounts

Guest accounts in Mattermost are a way to collaborate with individuals, such as vendors and contractors, outside of your organization by controlling their access to channels and team members. For example, guest accounts can be used to collaborate with customers on a support issue or work on a website project with resources from an external design firm.

Important

- A system admin must enable guest access before guests can be invited.
- Mattermost Enterprise and Professional customers can control who can invite guests in their organization. By default, only system admins can invite guests.
- Guest accounts count as a paid user in your Mattermost workspace. However, guests aren't automatically added to the default **Town-square** channel when they log in. You must invite guests to individual teams and channels manually. Deactivating a guest account reduces your licensed seat count.
- You'll identify guest users in Mattermost based on their **GUEST** badge next to their name and profile picture. Channels that contain guests also display the message ***This channel has guests** in the channel header.

Guests account limits

Guests can:

- Pin messages to channels
- Use slash commands (excluding restricted commands such as invite members, rename channels, change headers, etc)
- Favorite channels
- Mute channels

- Update their profile
- Use different authentication methods than other users
- Leave channels to which they were added (including Town Square).

Guests cannot:

- Discover public channels
- Join open teams
- Create direct messages or group messages with members who aren't within the same channel
- Invite people

Guest authentication

Guests can access the Mattermost server via email invitation, and be authenticated using AD/LDAP or SAML 2.0.

Before you proceed, ensure that the authentication method you wish to use is correctly configured on your server and enabled in Mattermost. For configuration steps and technical documentation, see Active Directory/LDAP setup and SAML Single-Sign-On.

Converting a member user to a guest won't change the channels they are in. However, they will be restricted from discovering additional channels and are unable to direct message/group message users outside of the channels they are in. They can be added to channels by system admins and other roles that have the correct permissions to invite guests.

Configure AD/LDAP authentication

When enabled, the **Guest Filter** in Mattermost identifies external users whose AD/LDAP role is `guest` and who are invited to join your Mattermost server. These users will have the `guest` role applied immediately upon first login instead of the default member user role. This eliminates having to manually assign the role in the System Console.

1. Go to **System Console > Authentication > Guest Access** to enable guest access.

2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **User Filters** section, then expand **Configure additional filters**.

3. Complete the **Guest Filter** field.

4. Select **Save**.

If a Mattermost guest user has the `guest` role removed in the AD/LDAP system, the synchronization process will not automatically promote them to a member user role. This is done manually via **System Console > User Management**. If a member user has the **Guest Attribute** added, the synchronization processes will automatically demote the member user to the guest role.

When a guest logs in without having any channels assigned to their account, they're advised to talk to a Mattermost system admin.

Configure SAML 2.0 authentication

When enabled, the **Guest Attribute** in Mattermost identifies external users whose SAML assertion is guest and who are invited to join your Mattermost server. These users will have the `guest` role applied immediately upon first login instead of the default member user role. This eliminates having to manually assign the role in the System Console.

If a Mattermost guest user has the guest role removed in the SAML system, the synchronization processes will not automatically promote them to a member user role. This is done manually via **System Console > User Management**. If a member user has the **Guest Attribute** added, the synchronization processes will automatically demote the member user to the guest role.

1. Go to **System Console > Guest Access** to enable guest access.

2. Go to **System Console > Authentication > SAML 2.0**.

3. Complete the **Guest Attribute** field.

4. Select **Save**.

When a guest logs in without having any channels assigned to their account, they're advised to talk to a Mattermost system admin.

Manage guests

See the following documentation to learn more about managing guests:

- invite people to teams and channels
- manage channel members
- remove people from teams

> Tip
>
> You can also use the `/kick` or `/remove` slash commands to remove a guest from a channel.

When a guest has been removed from all channels within a team, and if they belong to other teams, they will default into the last channel on the last team they have accessed. If they are removed from all channels on all teams, they'll be taken to a screen letting them know they have no channels assigned.

Promote and demote user roles

System admins can demote a user from a member to a guest by updating the user's role in **System Console > User Management > Users**. Select the member, then select **Demote to Guest**. All system and custom roles assigned to the demoted user are removed. System admins should also purge all of the demoted guest's sessions by selecting the guest user, then selecting **Revoke Sessions**.

The demoted guest user retains their existing channel and team memberships, but is restricted from discovering public channels and collaborating with users outside of the channels they're in. This is useful if you're already collaborating with external contractors, and want to restrict their abilities within Mattermost.

System admins can also promote a guest to member by updating their role in **System Console > User Management > Users**. Select the guest, then select **Promote to Member**.

> Note
>
> You can filter the list in **System Console > User Management > Users** to view all guests in the system.

Disable guest accounts

To disable the guest accounts feature, go to **System Console > Authentication > Guest Access**, then set **Enable Guest Access** to **False**. To deactivate individual guest accounts, go to **System Console > User Management > Users**. Select a user, then select **Deactivate**. You can re-activate individual guest accounts by selecting **Activate**.

- When a single guest account is deactivated or the guest account feature is disabled, guests are marked as `deactivated`, are logged out of Mattermost, and all guest sessions are revoked. In Mattermost Server versions prior to 5.18, disabling the guest account feature leaves current guest accounts as activated until they are manually deactivated.
- If you're using AD/LDAP and the guest access setting is disabled, the `guest` filter and existing guest users in System Console are deactivated. Additionally, no new guests can be invited or added using the filter as an authentication method. If a previous guest's credentials match the user filter (the only filter which is active when guest access is disabled), they will be reactivated and promoted to a member user upon their next login.
- Similarly, for SAML, when the guest access setting is disabled, the `guest` attribute and existing guest users in System Console are deactivated. Additionally, no new guests can be invited or added using the attribute as an authentication method. If a previous guest's credentials match the user attribute (the only attribute which is active when guest access is disabled), they will be reactivated and promoted to a member user upon their next login.

You can disable individual guest accounts in **System Console > User Management** via **Manage Members**. When a single guest account is disabled or the feature is disabled, the guest will be marked as `deactivated`, be logged out of Mattermost, and all their sessions will be revoked.

Reinstate guest accounts

When guest access is re-enabled for AD/LDAP, the `guest` filter is reinstated.

New users matching the `guest` filter will be authenticated as new guest users on login.

Previous guest users will be activated with the next synchronization. If their credentials still match the `guest` filter, they will retain their guest status. If they no longer match the `guest` filter but do match the `user` filter, they will be not be promoted to member user automatically on login - this must be done manually. If a previous guest was reactivated as a member user when guest access was disabled, and now are identified by the `guest` filter once again, they will automatically be demoted to Guest upon their login.

Similarly, for SAML, when guest access is re-enabled, the SAML `guest` attribute is reinstated. New users matching the `guest` attribute will be authenticated as new guest users on login.

Previous guest users will be activated with the next synchronization. If their credentials still match the `guest` attribute, they will retain their guest status. If they no longer match the `guest` attribute but do match the `user` filter, they will be not be promoted to member user automatically on login - this must be done manually. If a previous guest was reactivated as a member user when guest access was disabled, and now are identified by the `guest` attribute once again, they will automatically be demoted to guest upon their login.

Frequently asked questions

How am I charged for guest accounts?

Guests are charged as a user seat.

Why doesn’t Mattermost have single-channel guests?

We wanted to support collaboration with external guests for the broadest use cases without limiting guests’ access to channels. In the future, we may consider adding single-channel guests.

Can I set an expiration date for guests?

Currently, you cannot. This feature may be added at a later stage.

Can MFA be applied selectively?

If MFA is enforced for your users, it can be applied to guest accounts. Guests can configure MFA in by going to their profile picture and selecting **Profile > Security**. If MFA is not enforced for your users, it can't be applied to guest accounts.

Has the guest accounts feature been reviewed by an external security firm?

The guest account feature was reviewed by the Mattermost security team. We do not have an external firm review scheduled but will include this feature in future reviews.

How can I validate my guests' identity?

Guests can be authenticated via SAML and/or AD/LDAP to ensure that only the named guest can log in. Alternatively, you can whitelist domains via **System Console > Authentication > Guest Access > Whitelisted Guest Domains**.

Can I restrict guests' ability to upload content?

It is not currently possible to selectively disable upload/download functionality as it is a server-wide configuration.

On this page

- Guest accounts
  - Guests account limits
  - Guest authentication
    - Configure AD/LDAP authentication
    - Configure SAML 2.0 authentication
  - Manage guests
    - Promote and demote user roles

    - Disable guest accounts
    - Reinstate guest accounts
  - Frequently asked questions
    - How am I charged for guest accounts?
    - Why doesn’t Mattermost have single-channel guests?
    - Can I set an expiration date for guests?
    - Can MFA be applied selectively?
    - Has the guest accounts feature been reviewed by an external security firm?
    - How can I validate my guests’ identity?
    - Can I restrict guests’ ability to upload content?

Using AD/LDAP synchronized groups to manage team or private channel membership

Mattermost groups created with synchronized AD/LDAP groups can be used to manage the membership of private teams and private channels. When a team or private channel is managed by synchronized groups, users will be added and removed based on their membership to the synchronized AD/LDAP group.

For instance, you may have an AD/LDAP group that contains your development team that you want to synchronize to a developer team. By using this feature, new developers will get added to the team when they are added to the synchronized AD/LDAP group and they will be removed from the team when removed from the AD/LDAP group.

Similarly, you may have an AD/LDAP group that contains your leadership team that you want to synchronize to a private channel for coordination and updates. This feature will help control the membership of the channel so that users outside of the synchronized group are prevented from being added to the channel mistakenly.

On teams that are managed by synchronized groups, users outside of the group are restricted from:

> - *Invitation through a team invite link*
> - *Invitation through an email invite*

Similarly on private channels that are managed by synchronized groups, users outside of the group are restricted from:

> - *Invitation through a mention*
> - *Invitation through the* `/invite` *slash command*
> - *Being added to the channel with* ***Add members***

Users can remove themselves from teams and private channels managed by synchronized groups.

Managing membership of a team or channel with synchronized groups

To manage membership of a private team with synchronized groups:

1. Navigate to **System Console > User Management > Teams**.
2. Select the team you want to manage with group synchronization.
3. Under **Team Management**, enable **Sync Group Members**. If **Anyone can join this team** is enabled or if specific email domains are set, they will be disabled by the sync group members feature.
4. Add one or more groups to the team. If there are groups already associated to default users into the team, they will already be present.
5. Review the notice in the footer of the screen for any users that are not part of groups who will be removed from the team on the next synchronization.
6. Select **Save**. Members will be updated on the next scheduled AD/LDAP synchronization.

Alternatively you can use the mmctl tools to set the team to be managed by groups:

1. Ensure there's at least one group already associated to the team. You can view and add default teams to a group via **System Console > User Management > Groups > Group Configuration**. See the mmctl group team list documentation for more information on adding default teams and channels and confirming whether if there is already a group associated to the team.
2. Ensure **Team Settings > General > Allow any user with an account on this server to join this team** is set to `No`.
3. Convert the team to have its membership managed by synchronized groups by running the mmctl group team enable command.

To manage membership of a private channel with synchronized groups:

1. Navigate to **System Console > User Management > Channels**.
2. Select the channel you want to manage with group synchronization.
3. Under **Channel Management**, enable **Sync Group Members**. Please ensure the channel is set to `private`.
4. Add one or more groups to the channel. If there are groups already associated to default users in the channel, they'll already be present.
5. Review the notice in the footer of the screen for any users that are not part of groups who will be removed from the channel on the next synchronization.
6. Select **Save**. Members will be updated on the next scheduled AD/LDAP synchronization.

Alternatively you can use the mmctl tool to set a private channel to be managed by groups:

1. Ensure there's at least one group already associated to the channel. You can view and add default channels to a group via **System Console > User Management > Groups > Group Configuration**. See our AD/LDAP documentation for more information on adding default teams and channels. Additionally, you can use the mmctl to view if there is already a group associated to the channel by running the mmctl group channel list command.
2. Convert the team to have its membership managed by synchronized groups by running the mmctl group channel enable command.

Add or remove groups from teams

Once the management of the team is converted to be managed by synchronized groups, a team admin or system admin can add additional groups from **Main Menu > Add Groups to Team**. This will add users on the next AD/LDAP synchronization and any new users to the group will be added to the team on subsequent synchronizations. Team admins will be prevented from changing the team to public by enabling **Team Settings > Allow any user with an account on this server to join this team**.

Team admins or system admins can also remove groups from a team from **Main Menu > Manage Groups**. This will disassociate the group from the team. Users are removed on the next AD/LDAP synchronization.

The system admin can also remove groups from **System Console > User Management > Teams > Team Configuration > Synced Groups**.

Add or remove groups from private channels

Once the management of the channel is converted to be managed by synchronized groups, a team admin or system admin can add additional groups from **Channel Menu > Add Groups to Channel**. This will add users on the next AD/LDAP synchronization and any new users to the group will be added to the channel on subsequent synchronizations.

Team admins and system admins can also remove groups from a team from **Main Menu > Manage Groups**. This will disassociate the group from the team. Users are removed on the next AD/LDAP synchronization.

The system admin can also remove groups from **System Console > User Management > Channels > Channel Configuration > Synced Groups**.

Managing members

Users are automatically removed from the team or private channel when removed from a synchronized AD/LDAP group that is managing the membership of that team or channel. Additionally, users who are not in the synchronized groups are prevented from being added through the `/invite` and mention flows within a channel.

A user can remove themselves from the team or from the private channel when it is managed by synchronized groups. They can be added back by users who have permission to manage members for a team or private channel by using the `/invite` slash command or by mentioning the user in a channel.

If the user is removed from a synchronized group and later readded to the group, they can be manually added back to the team or private channel as noted above.

> Note
>
> Users will not be automatically added back by the AD/LDAP synchronization once they remove themselves or are removed by the LDAP synchronized group.

Disabling group synchronized management of teams and private channels

To remove the management of members by synchronized groups in a team, disable **Sync Group Members** under **System Console > User Management > Teams > Team Management**, or run the mmctl group team disable command.

To remove the management of members by synchronized groups in a channel, disable **Sync Group Members** under **System Console > User Management > Channels > Channel Management**, or run the mmctl group channel disable command.

Frequently asked questions

Why aren't public channels supported with this feature?

Public channels are available to all members to discover and join. Managing membership with synchronized groups removes the ability for public channels to be accessible to users on the team. Private channels typically require a more controlled membership management, which is why this feature applies to Private channels. Groups can be assigned to public teams and public channels as described in this documentation.

Does a team with its membership managed by groups have any effect on Public channel access?

Only users that are members of groups synchronized to team are able to discover and join public channels. Private channels can also be managed by synchronized groups when a team is managed by synchronized groups.

Why don't users get readded to teams or channels once they have been removed from and then later re-added to the LDAP group?

The implementation of group removals does not currently differentiate between users who have removed themselves or have been removed by the LDAP synchronization process. Our design optimizes for users who have removed themselves from a team or channel. In the future, we may add the ability for admins to re-add users who have been removed, and even prevent users from leaving a team or channel.

Additionally, LDAP users who aren't accessible to Mattermost based on filters will be removed from the groups and from group-synced teams and channels. If they were removed from teams and channels then they will not be re-added to those teams and channels upon becoming subsequently re-accessible to Mattermost.

On this page

- Using AD/LDAP synchronized groups to manage team or private channel membership
  - Managing membership of a team or channel with synchronized groups
  - Add or remove groups from teams
  - Add or remove groups from private channels
  - Managing members
  - Disabling group synchronized management of teams and private channels
  - Frequently asked questions

Migrate from Slack

> Note
>
> Slack's data control policies may change at any time (https://x.com/Austen/status/1752064934970896626). Also, Slack may choose to change its export capabilities, and/or charge fees to customers for exporting data stored in Slack. Support for negotiating export of customer IP from Slack Enterprise can be requested by contacting a Mattermost Expert (https://mattermost.com/contact-sales/).

Migrating from Slack to Mattermost involves the following steps:

1. Prepare your Mattermost server
2. Generate a Slack import
3. Download file attachments and email addresses
4. Convert the Slack import to Mattermost's format
5. Import data into Mattermost

1. Prepare your Mattermost server

During the import process, we advise to create a new team for importing the Slack workspace data. If merging multiple Slack workspaces into a single team is the desired end-result, we recommend completing the import to separate teams, validating the results, then using mmctl to move channels between teams.

Also, system administrator roles will be overwritten if the usernames match and the user isn't an admin on the Slack workspace.

Make sure you are running on a supported version of Mattermost to benefit from the most up-to-date functionality and fixes.

2. Generate a Slack import

Slack offers two ways to [export your data from their product (https://slack.com/help/articles/201658943-Export-your-workspace-data)](https://slack.com/help/articles/201658943-Export-your-workspace-data).

1. Regular export - Contains only public channel posts. This does not include private channels, DMs, or group conversations. This can be generated from **Slack > Administration > Workspace settings > Import/Export Data > Export > Start Export**.

2. Corporate export - Contains all posts. This includes public channels, private channels, DMs and group messages. You must [request this export type from Slack directly (https://slack.com/help/articles/1500001548241-Request-to-export-additional-data-from-your-workspace-or-Enterprise-Grid-org)](https://slack.com/help/articles/1500001548241-Request-to-export-additional-data-from-your-workspace-or-Enterprise-Grid-org).

You will receive a zip file containing the following contents:

- Channels (`channels.json`)
- Users (`users.json`)
- Direct messages (`dms.json`) (Corporate export)
- Private channels (`groups.json`) (Corporate export)
- Group direct messages (`mpims.json`) (Corporate export)
- App activity logs (`integration_logs.json`)
- Folders containing posts for every public channel
- Folders containing posts for every private channel (Corporate exports)

> Note
>
> - Refer to the Slack help article (https://slack.com/help/articles/220556107-How-to-read-Slack-data-exports) for additioanl details on zip file contents.
> - As a proprietary SaaS service, Slack is able to change its export format quickly and without notice. If you encounter issues not mentioned in the following documentation, please let the Mattermost Product Team know by filing an issue (https://handbook.mattermost.com/contributors/contributors/ways-to-contribute).

3. Download file attachments and email addresses

When you download your Slack export zip file, some data will be missing from the zip file:

- User emails
- Uploaded attachment contents

We need to create a Slack app in order to gather these contents. Follow these steps to create a Slack app:

1. Go to https://api.slack.com/apps.
2. Select **Create New App**.
3. Select **From scratch**.
4. Name the app **Slack Advanced Exporter** and select the workspace. You'll have to do this for every workspace. Then select **Create App**.
5. Select **OAuth & Permissions** on the left-hand side of the screen. Then scroll down to **Scopes**.

6. Under **Bot Token Scopes** type in and select the following scopes:

> - `users:read`
> - `users:read.email`

7. Scroll up and select **Install to Workspace**.
8. Grant the app permissions when prompted.
9. Copy the **Bot User OAuth Token** and save it somewhere convenient.

We'll now use **Bot User OAuth Token** with the `slack-advanced-exporter` tool to download emails and attachments. Download the latest release of `slack-advanced-exporter` for your OS and architecture here (https://github.com/grundleborg/slack-advanced-exporter/releases/) and extract the executable from the download.

Once you have the program downloaded locally, run the commands below to fetch the emails, and then fetch file attachments. Replace `<SLACK TOKEN>` with the Slack token you generated earlier and `<SLACK EXPORT FILE>` with the path (https://www.geeksforgeeks.org/absolute-relative-pathnames-unix/) to your export file.

```
./slack-advanced-exporter --input-archive <SLACK EXPORT FILE> --output-archive export-with-emails.zip fetch-emails --api-token <SLACK TOKEN> ./slack-advanced-exporter --input-archive export-with-emails.zip --output-archive export-with-emails-and-attachments.zip fetch-attachments --api-token <SLACK TOKEN>
```

> Note
>
> - The first command collects all of the user emails and creates the file `export-with-emails.zip`. The second command fetches attachments and creates the file `export-with-emails-and-attachments.zip`, which we will use going forward.
> - The second command can take a long time if you have a large number of file uploads. If it's interrupted, delete the file generated (if any), and start again.

The file `export-with-emails-and-attachments.zip` now contains all the information necessary to be imported into Mattermost.

It's preferable to fetch e-mails first to avoid copying large attachments around. Make sure you choose different file names at each stage, as the tool does not support in-place modifications.

Important

Avoid unzipping and rezipping the Slack export. Doing so can modify the directory structure of the archive which could cause issues with the import process.

4. Convert Slack import to Mattermost's bulk export format

Now that you have a Slack export file with emails and attachments, let's convert this information into Mattermost's bulk import format using the import preparation tool `mmetl`.

Download the latest release of mmetl for your OS and architecture. Run `mmetl help` to learn more about using the tool.

Next, run the command below to create a Mattermost bulk import file. Replace `<TEAM-NAME>` with the name of your team in Mattermost. Note that the name needs to be one word and lowercase (i.e. if you named your team `My Team`, `<TEAM-NAME>` would be `my-team`).

```
./mmetl transform slack --team <TEAM-NAME> --file export-with-emails-and-
attachments.zip --output mattermost_import.jsonl
```

The tool outputs a [.jsonl (https://jsonlines.org/examples)](https://jsonlines.org/examples) file containing all of your users, channels, and posts. It also creates a `data` folder that contains all of your attachments. Users' default authentication method will be configured as email/password. See the section below for instructions on migrating the auth method to another choice.

Important

It doesn't matter what you name the `.jsonl` file. You can name it what you want with the *–output* flag as shown above. It just needs to be a `.jsonl` file.

5. Import data into Mattermost

You can upload the export through Mattermost's API from the server or from another computer using mmctl commands. The server will save the import in its file store before running the import (e.g. AWS S3), so there will be time spent uploading/downloading the file in this case.

The migration is idempotent, meaning that you can run multiple imports that contain the same posts, and there won't be duplicated created posts in Mattermost. Each post is imported with the correct user/author and `created_at` value from your Slack instance. Threads are kept intact with the import.

Ensure you have the Mattermost command line tool `mmctl` installed. This allows you to perform different tasks that communicate to Mattermost's API. You'll also want to configure authentication for the tool.

To prepare our files to be uploaded to the server, we need to put both the `.jsonl` file and `data` folder together into a zip file.

```
zip -r mattermost-bulk-import.zip data mattermost_import.jsonl
```

Then we can upload the zip file to our Mattermost server:

```
mmctl import upload ./mattermost-bulk-import.zip
```

Run this command to list the available imports:

```
mmctl import list available
```

Run this command to process the import. Replace `<IMPORT FILE NAME>` with the name you got from the `mmctl import list available` command:

```
mmctl import process <IMPORT FILE NAME>
```

Finally, run this command to view the status of the import process job. If the job status shows as `pending`, then wait before running the command again. The `--json` flag is required to view the possible error message. Replace `<JOB ID>` with the id you got from the `mmctl import list process` command:

```
mmctl import job show <JOB ID> --json
```

Debug imports

You can use the `mmctl import job show` command to view any relevant errors that may have occurred.

Important

Note that if you are part of the user group being imported from Slack, your Mattermost profile must have a matching username and email to the corresponding `user` line of the `jsonl` file. You can manually edit the file to ensure it matches your Mattermost user.

Additional tools

- mm-emoji (https://github.com/users/maxwellpower/packages/container/package/mm-emoji) - Designed to smoothly transition emojis from a Slack instance to Mattermost.
- mm-importjs (https://github.com/mickmister/mmimportjs) - Breaks up large import files into smaller ones, as well as automatically remove null characters in post content when importing data to Mattermost.
- slack-migrate-pinned-posts (https://github.com/svelle/slack-migrate-pinned-posts) - Migrates pinned posts from Slack to Mattermost.

Use the imported team

- During the import process, the emails and usernames from Slack are used to create new Mattermost accounts. If emails are not present in the Slack export archive, then placeholder values will be generated and the system admin will need to update these manually. We recommend administrators search the final import `jsonl` file for `user` lines for with `@example.com` in the email property to address and resolve the missing information prior to import.
- Slack users can activate their new Mattermost accounts by using Mattermost's **Password Reset** screen with their email addresses from Slack to set new passwords for their Mattermost accounts.

- Once logged in, Mattermost users will have access to previous Slack messages in the public channels imported from Slack.
- Instructions on how to migrate user authenticatation to LDAP or SAML can be found here.

On this page

- Migrate from Slack
  - 1. Prepare your Mattermost server
  - 2. Generate a Slack import
  - 3. Download file attachments and email addresses
  - 4. Convert Slack import to Mattermost's bulk export format
  - 5. Import data into Mattermost
    - Debug imports
    - Additional tools
    - Use the imported team

Migrate from HipChat to Mattermost

You can migrate HipChat users and message histories to Mattermost using the following guidelines.

Step 1: Set up your Mattermost instance

- Go to Mattermost download page to install Mattermost in your environment using one of the installation guides for Linux binary install, Docker install or various orchestrated installations.

Questions? Please visit our troubleshooting forum (https://forum.mattermost.com/t/how-to-use-the-troubleshooting-forum/150) for help.

Step 2: Export your data from HipChat Server or HipChat Data Center

HipChat Server/HipChat Data Center

If you're able to upgrade HipChat Server or HipChat Data Center to the latest version, we recommend using Group Export Dashboard to export your data. If you're unable to upgrade, see Command Line Interface procedure below.

Using the Group Export Dashboard

1. Log in to your Hipchat Server (e.g. hipchat.yourcompany.com).
2. Click on **Server Admin > Export**.
3. Select the data to export.
4. In the **Password** and **Confirm Password** fields, create a password to protect your archive files (Store this password as it is not saved anywhere else).

5. Click **Export**. Once the export is done, you will receive an email with a link to download the file.

If you're unable to use the Group Export Dashboard, use the command line interface to export

1. Go to **CLI**.
2. Enter `hipchat export --export -p your_password`.
3. Once the export is done, you will receive an email with a link to download the file.

Step 3: Import your data into Mattermost

1. Follow the Mattermost Bulk Load Tool guide to import your data into Mattermost. Files exported from HipChat will need to be converted to the format required by Mattermost. Talk to a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) if you require assistance in the conversion.
2. You are also encouraged to use the HipChat import tool created by Herzum: https://github.com/herzum/HC2MM.

If you're interested in contributing or testing any community contributed solution, please let us know at info@mattermost.com, Twitter or Mattermost forums at https://forum.mattermost.com.

3. Alternatively, talk to a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) for partner recommendations for your region to assist in your import.

Step 4: Onboard your users into Mattermost

After importing users, you can send out an announcement via email or via your old system (or both) to let users know how to log into Mattermost with their old accounts or how to create new accounts.

Announcing Mattermost onboarding in your previous messaging system:

Use the following message template to alert users of the migration:

```
@all, we're moving communications to a new Mattermost server. You can start
your new account by going to the [your new location, e.g. ``https://
yourcompany.com/mattermost``], selecting **I forgot my password**, and
entering the email you used on this system in the Reset Password page to set
up new credentials. Your message history and channels should carry over from
this system into Mattermost. Any questions? Please let us know.
```

Announcing Mattermost onboarding using email using username/password:

1. Get a list of email addresses of people in the new system by running a database query on Mattermost. Run `SELECT Email FROM Users` from either PostgreSQL or MySQL databases.
2. Adapt the migration announcement email template to let users know how to reclaim their old accounts or start new ones.

Onboard users using SSO in Mattermost

Alternatively, you can choose to set up SSO (Single Sign-on) with Mattermost if you are using an Enterprise version.

1. Configure Active Directory/LDAP or SAML Single Sign-on from the **System Console**.
2. Adjust the messaging templates above to remove "password reset" references and indicate which SSO system credentials Mattermost has configured.

On this page

- Migrate from HipChat to Mattermost
  - Step 1: Set up your Mattermost instance
  - Step 2: Export your data from HipChat Server or HipChat Data Center
    - HipChat Server/HipChat Data Center
  - Step 3: Import your data into Mattermost
  - Step 4: Onboard your users into Mattermost
    - Onboard users using SSO in Mattermost

Migration guide

Thousands of organizations are moving to Mattermost for powerful, flexible, and easy-to-manage workplace collaboration. Mattermost deploys as a single Linux binary with PostgreSQL, and can scale from dozens to tens of thousands of users in a single channel.

This guide summarizes different approaches to migrating to Mattermost from other tools, including Slack, HipChat, Jabber, Pidgin, Bitnami, and other bespoke messaging solutions, as well as migrating Mattermost server to another server instance.

Migrate from Slack

See the Migrate from Slack documentation for details on migrating from Slack to Mattermost.

Migrate from HipChat

See the Migrate from HipChat documentation for details on migrating from HipChat Server and HipChat Data Center to Mattermost.

Migrate from Jabber

BrightScout helped a major U.S. Federal Agency rapidly migrate from Jabber to Mattermost and open sourced their Extract, Transform and Load (ETL) tool at https://github.com/Brightscout/mattermost-etl. Read more about their case study (https://mattermost.com/blog/u-s-federal-agency-migrates-from-jabber-to-mattermost-the-open-source-way/) online.

Migrate from Pidgin

In some cases, people are using Pidgin clients with different backends to communicate. To continue using Pidgin with a Mattermost backend, consider using Mattermost ETL tool (https://github.com/Brightscout/mattermost-etl),

created by BrightScout, to migrate data from your existing backend into Mattermost.

Then use the [Pidgin-Mattermost plugin (https://github.com/EionRobb/purple-mattermost)](https://github.com/EionRobb/purple-mattermost) (complete with an installer for end user machines) to continue to support legacy Pidgin users while offering a whole new Mattermost experience on web, mobile, and PC.

Migrate from Bitnami

Bitnami uses MySQL, and renames the Mattermost database tables by converting the names to all lower case. For example, in non-Bitnami installations, the Users table is named `Users`, but in Bitnami, the table is `users` (with a lowercase `u`). As a result, when you migrate your data from Bitnami to a non-Bitnami installation, you must modify the MySQL startup script so that it starts MySQL in lowercase table mode.

You can modify the script by adding the `--lower-case-table-names=1` switch to the MySQL start command. The location of the start-up script generally depends on how you installed MySQL, whether by using the package manager for the operating system, or by manually installing MySQL. You must modify the start-up script before migrating the data.

For more information about letter case in MySQL table names and the `--lower-case-table-names` switch, see the [Identifier Case Sensitivity (https://dev.mysql.com/doc/refman/5.7/en/identifier-case-sensitivity.html)](https://dev.mysql.com/doc/refman/5.7/en/identifier-case-sensitivity.html) topic in the MySQL documentation.

Migrate from bespoke messaging solutions

Mattermost is often selected to replace bespoke solutions by IT and DevOps teams as a stable, enterprise-grade, commercially-supported solution on an open source platform that meets and exceeds the flexibility and innovation of bespoke solutions.

Migrating from bespoke messengers to Mattermost can be challenging. Because of the difficulty of upgrading and maintaining bespoke solutions, the format for storing data is unpredictable, and the community around any single legacy release is small.

If your data in the bespoke messenger is vital, consider:

1. Mattermost bulk load tool: Use the Mattermost bulk load tool to ETL from your bespoke system to Mattermost.
2. Mattermost ETL framework from BrightScout (https://github.com/Brightscout/mattermost-etl): Consider the Mattermost ETL framework from BrightScout to custom-configure an adapter to plug in to the Bulk Load tool mentioned above.
3. **Legacy Slack import:** If you only recently switched from Slack to a bespoke tool, consider going back to import the data and users from the old Slack instance directly into Mattermost, leveraging the extensive support for Slack-import provided.
4. **Export to Slack, then import to Mattermost:** Export HipChat, Flowdock, Campfire, Chatwork, Hall, or CSV files to Slack (https://slack.com/help/articles/217872578-Import-data-from-one-Slack-workspace-to-another) and then export to a Slack export file and import the file into Mattermost.

If your data in the bespoke messenger is not vital, consider:

1. **Parallel systems:** Running Mattermost in parallel with your bespoke system until the majority of workflow and collaboration has moved to Mattermost
2. **Hard switch:** Announce a "hard switch" to Mattermost after a period of time of running both systems in parallel. Often this has been done due to security concerns in bespoke products or products nearing end-of-life.

Sometimes systems running in parallel turn into a hard switch migration when a bespoke or deprecated system experiences a major outage or a security exploit. In 2017, this was experienced by many companies using Mattermost and HipChat.com in parallel when HipChat suffered a major security breach where customer data was stolen by an unknown attacker (https://thenextweb.com/news/hipchat-hacked-weekend-bad#.tnw_lAotA9OV).

Migrate Mattermost from one server to another

The following instructions migrate Mattermost from one server to another by backing up and restoring the Mattermost database and `config.json` file. For

these instructions SOURCE refers to the Mattermost server *from which* your system will be migrated and DESTINATION refers to the Mattermost server *to which* your system will be migrated.

1. Back up your SOURCE Mattermost server. See Backup and Disaster Recovery documentation.
2. Upgrade your SOURCE Mattermost server to the latest major build version. See Upgrading Mattermost Server documentation.
3. Install the latest major build of Mattermost server as your DESTINATION.

> - *Make sure your new instance is properly configured and tested. The database type (MySQL or PostgreSQL) and version of SOURCE and DESTINATION deployments need to match.*
> - *Stop the DESTINATION server using* `sudo stop mattermost`*, then back up the database and* `config.json` *file.*

4. Migrate database from SOURCE to DESTINATION. Backup the database from the SOURCE Mattermost server and restore it in place of the database to which the DESTINATION server is connected.
5. Migrate `config.json` from SOURCE to DESTINATION. Copy `config.json` file from SOURCE deployment to DESTINATION.
6. If you use local storage (`FileSettings.DriverName` is set to `local`), migrate `./data` from SOURCE to DESTINATION.

> - *Copy the* `./data` *directory from SOURCE deployment to DESTINATION.*
> - *If you use a directory other than* `./data`*, copy that directory instead.*

7. Start the DESTINATION deployment by running `sudo start mattermost`. Then go to the **System Console**, make a minor change, and save it to upgrade your `config.json` schema to the latest version using default values for any new settings added.
8. Test that the system is working by going to the URL of an existing team. You may need to refresh your Mattermost browser page in order to get the latest updates from the upgrade.

Once your migration is complete and verified, you can optionally upgrade the Team Edition of Mattermost to Enterprise Edition using the upgrade guide.

On this page

- Migration guide
  - Migrate from Slack
  - Migrate from HipChat
  - Migrate from Jabber
  - Migrate from Pidgin
  - Migrate from Bitnami
  - Migrate from bespoke messaging solutions
  - Migrate Mattermost from one server to another

Migration announcement email

To notify your end users of your migration to Mattermost, we created a sample email template that you can use.

Remember to replace all the items in bold with your information.

Email template

To: End users

Subject: Moving to Mattermost

Hi all,

As some of you already know, we are moving from **[System]** to Mattermost as our collaboration platform. We have imported our existing users and data into Mattermost so we can continue to communicate, share files and collaborate.

To get started:

1. Open a browser on your computer, go to **[Mattermost URL]** and select "I forgot my password" to set your new password for Mattermost using the email address you used on the previous system.
2. Once you reset your password, use your email and new password to log in.
3. Download the Mattermost apps for desktop and mobile. Refer to the Use Mattermost end user documentation on how to get up and running quickly.
4. Remember to bookmark the URL so you can use it to log in next time!

Questions?

If you have any questions, feel free to post in the **[~Mattermost channel]** or email us at **[Support email]**.

Happy collaborating!

[Sign-off]

On this page

- Migration announcement email
  - Email template

Multi-factor authentication

Multi-factor authentication (MFA) is a secondary layer of user authentication applied to your Mattermost server that applies to all users on all teams within your Mattermost workspace.

With MFA enabled, users need to provide a secure one-time code in addition to their username and password in order to log in to Mattermost. MFA is useful in organizations that have specific security and compliance policies. It can also be used in organizations where Mattermost is not behind a firewall and doesn't have access to existing MFA infrastructure.

Mattermost offers a smartphone-based MFA check in addition to email-password or Active Directory/LDAP authentication to log in to the Mattermost server.

Supported devices include iOS, Android, or other devices that are able to install a software-based authenticator such as FreeOTP, Google Authenticator, Microsoft Authenticator, 1Password, LastPass, or a similar app. After the app is installed, the device does not require internet access.

> Note
>
> As the MFA implementation relies on Time-based One-time passwords (TOTP) ensure that your server system time is accurate. If there is a discrepancy, users may not be able to log in successfully.

Enabling MFA

System admins can enable this option by going to **System Console > Authentication > MFA**, then setting **Enable Multi-factor Authentication** to **true**.

Once enabled, users can choose to set up multi-factor authentication on their account by selecting **Profile > Security > Multi-factor Authentication** from their profile picture.

Disabling MFA

System admins can disable this option by going to **System Console > Authentication > MFA**. When **Enable Multi-factor Authentication** is set to **false**, users can't opt to use or disable multi-factor authentication on their account via **Profile > Security > Multi-factor Authentication**.

MFA can be disabled for user accounts using the API. See the Mattermost API Reference (https://api.mattermost.com/#tag/users/paths/~1users~1{user_id}~1mfa/put) for details.

Enforcing MFA

Note

Admins can fulfill Multi-Factor Authentication (MFA) compliance requirements by enforcing an MFA requirement for login with email and LDAP accounts. Go to **System Console > Authentication > MFA**, then set **Enforce Multi-factor Authentication** to **true**.

When MFA enforcement is set to **true**, users with email or LDAP authentication who don't have MFA set up will be directed to the MFA setup page when they log in to Mattermost. They will not be able to access the site until MFA setup is complete. Any new users will be required to set up MFA during the sign up process.

Users will not be able to remove MFA from their account while enforcement is on.

> Note
>
> Turning on MFA enforcement prevents users from accessing the site until set up is complete. We recommended that you turn on enforcement during non-peak hours when people are less likely to be using Mattermost.

On this page

- Multi-factor authentication
  - Enabling MFA
    - Disabling MFA
  - Enforcing MFA

SSL client certificate setup

Follow these steps to configure SSL client certificates for your browser and the Mattermost desktop apps on Windows, macOS, and Linux. SSL client certificates are not yet supported on the Mattermost mobile apps.

Before you begin, follow the official guides to install Mattermost on your system, including NGINX configuration as a proxy with SSL and HTTP/2, and a valid SSL certificate such as Let's Encrypt.

For the purposes of this guide, the Mattermost server domain name is `example.mattermost.com`, and the user account is `mmuser` with email `mmuser@mattermost.com` and password `mmuser-password`.

Important

- Generating the client certificates in this section is optional if you have already generated them before.
- We strongly recommend configuring an SSL certificate (or a self-signed certificate) for security, privacy, compliance, as well as to avoid browser limitations that can prevent Mattermost product features from working that copy data using the user's local clipboard, including sharing message links and inviting people.

Set up mutual TLS authentication for the web app

1. Create a [certificate authority (CA) key (https://en.wikipedia.org/wiki/Certificate_authority)](https://en.wikipedia.org/wiki/Certificate_authority) and a certificate for signing the client certificate. When establishing a TLS connection, the NGINX proxy server requests and validates a client certificate provided by the web app.

```
openssl genrsa -des3 -out ca.mattermost.key 4096
```

```
pass phrase: capassword
```

```
openssl req -new -x509 -days 365 -key ca.mattermost.key -out
ca.mattermost.crt
```

```
Country Name: US State: Maryland Locality Name: Meade Organization Name:
Mattermost Organization Unit: Smarttotem Common Name: example.mattermost.com
Email Address: admin@mattermost.com
```

2. Create the client side key for `mmuser` with a passphrase, and the certificate signing request:

```
openssl genrsa -des3 -out mmuser-mattermost.key 1024
```

```
passphrase: mmuser-passphrase
```

```
openssl req -new -key mmuser-mattermost.key -out mmuser-mattermost.csr
```

```
Country Name: US State: Maryland Locality Name: Meade Organization Name:
Mattermost Organization Unit: Smarttotem Common Name: mmuser Email Address:
mmuser@mattermost.com Challenge password: mmuser-passphrase
```

3. Sign the user's client certificate with the previously created CA certificate:

```
openssl x509 -req -days 365 -in mmuser-mattermost.csr -CA ca.mattermost.crt
-CAkey ca.mattermost.key -set_serial 01 -out mmuser-mattermost.crt
```

4. Check the newly generated client certificate for `mmuser`:

```
openssl x509 -in mmuser-mattermost.crt -text -noout
```

5. Open the file `/etc/nginx/sites-available/mattermost` and modify the following lines, so that the NGINX proxy server requests and verifies the client certificate:

```
ssl on; ssl_certificate /etc/letsencrypt/live/example.mattermost.com/
fullchain.pem; ssl_certificate_key /etc/letsencrypt/live/
example.mattermost.com/privkey.pem; ssl_client_certificate /opt/mattermost/
config/ca.mattermost.crt; ssl_verify_client on;  ... location ~ /api/
v[0-9]+/(users/)?websocket$ {  proxy_set_header X-SSL-Client-Cert
$ssl_client_cert;  proxy_set_header X-SSL-Client-Cert-Subject-DN
$ssl_client_s_dn;  ... location / {  proxy_set_header X-SSL-Client-Cert
$ssl_client_cert;  proxy_set_header X-SSL-Client-Cert-Subject-DN
$ssl_client_s_dn;  ...
```

6. Confirm the CA key for `mmuser` works by the following curl command to the proxy:

```
curl -v -s -k --key mmuser-mattermost.key --cert mmuser-
mattermost.crt:mmuser-passphrase https://example.mattermost.com
```

You should see the Mattermost login page. If you see:

> - `No required SSL certificate was sent`, *something went wrong. Review the above steps and try again.*
> - `Error reading X.509 key or certificate file: Decryption has failed.`, *make sure the passphrase is included together with the certificate, because curl doesn't prompt for it separately.*

7. Generate a PKCS12 file from the CA key and certificate, to install the certificate into your client machine for your browser to use:

```
openssl pkcs12 -export -out mmuser-mattermost.p12 -inkey mmuser-
mattermost.key -in mmuser-mattermost.crt -certfile ca.mattermost.crt
```

```
Enter Export Password: mmuser-passphrase
```

8. Repeat steps 2-7 above for other users as needed.

9. Import the generated `.p12` file in step 7 into your key chain. In the Chrome browser on macOS:

> 1. *Go to **Settings > Advanced > Privacy and security > Manage certificates**. This opens the Keychain Access app.*

> *2. Go to **File > Import Items** and select the `mmuser-mattermost.p12` file.*

10. Go to `https://example.mattermost.com`. You should see a popup for the client certificate request.

On this page

- SSL client certificate setup
  - Set up mutual TLS authentication for the web app

Entra ID Single Sign-On

Configuring EntraID as a Single Sign-On (SSO) service

Follow these steps to configure Mattermost to use your Entra ID logon credentials and Azure Active Directory account as a Single Sign-on (SSO) service for team creation, account creation, and sign-in.

> Note
>
> The system must be using SSL as Microsoft only allows OAuth redirect URIs that are SSL-enabled.

Step 1: Register an application in Azure Portal

1. Log in to the [Azure Portal (https://portal.azure.com/)](https://portal.azure.com/) with the account that relates to the Azure Active Directory tenant where you want to register the application. You can confirm the tenant in the top right corner of the portal.
2. In the left-hand navigation pane, select the **Azure Active Directory service**, then select **App registrations > New registration**.
3. Give your new registration a **Name**.
4. Define which **Supported account types** can access the application. For example, if this is to be only accessed from your enterprise's Azure AD accounts, then select **Accounts in this organizational directory only**.
5. Define the **Redirect URI** as Web client, then input the URL with the host name that will be specific to your Mattermost service followed by `/signup/office365/complete`. An example below is: `https://your.mattermost.com/signup/office365/complete`

Once the App Registration has been created, you can configure it further. See the standard Azure AD documentation for reference.

Step 2: Generate a new client secret in Azure Portal

1. In the Azure Portal, select **Certificates and Secrets** from the menu, then select the button to generate a **New Client secret**.

2. Provide a description, define the expiry for the token, then select **Add**.

3. In Azure Portal, select **Overview** from the menu, then copy and paste both the Application (client) ID and the Directory (tenant) ID to a temporary location. You will enter these values as an **Application ID** and as part of an **Auth Endpoint** and **Token Endpoint** URL in the Mattermost System Console.

Step 3: Configure Mattermost for Entra ID SSO

1. Log in to Mattermost, then go to **System Console > Authentication > OpenID Connect**.

2. Select **Entra ID** as the service provider.

3. Paste the **Directory (tenant) ID** from the Azure Portal as the **Directory (tenant) ID** in Mattermost.

4. The **Discovery Endpoint** for OpenID Connect with Entra ID is prepopulated with `https://login.microsoftonline.com/common/v2.0/.well-known/openid-configuration`.

5. Paste the **Application (client) ID** from the Azure Portal as the **Client ID** in Mattermost.
6. Paste the **client secret** value from the Azure Portal as the **Client Secret** in Mattermost.
7. Select **Save**.

> Note
>
> When Mattermost is configured to use OpenID Connect or OAuth 2.0 for user authentication, the following user attribute changes can't be made through the Mattermost API: first name, last name, or username. OpenID Connect or OAuth 2.0 must be the authoritative source for these user attributes.

Note about Microsoft Active Directory Tenants

A Microsoft Active Directory (AD) tenant is a dedicated instance of Azure Active Directory (Azure AD) that you own and would have received when signing up for a Microsoft cloud service, such as Azure or Entra ID. Tenants are commonly used by organizations who want to store information about their users, such as passwords, user profile data, and permissions. See the Microsoft Entra ID <https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-create-new-tenant>`_ documentation to learn more about getting an Azure AD tenant.

To allow your Azure AD users to log in to Mattermost using Entra ID SSO, you must register Mattermost in the Microsoft Azure AD tenant that contains the users' information. The registration can be done from the [Microsoft Azure portal (https://portal.azure.com)](https://portal.azure.com). The steps to register the Mattermost account in the tenant should be similar to those provided above, and you can find more information about [integrating apps with Azure AD here (https://learn.microsoft.com/en-us/azure/active-directory/develop/quickstart-create-new-tenant)](https://learn.microsoft.com/en-us/azure/active-directory/develop/quickstart-create-new-tenant).

If you don't register Mattermost in the Microsoft Azure AD tenant your organization uses, Entra ID SSO will likely fail for your users.

> Note
>
> If you do not use Azure Active Directory, you may register Mattermost with your Entra ID or Azure account (a personal, work, or school account), then set up Entra ID SSO with Mattermost using the steps provided above.

Configure Mattermost `config.json` for Entra ID SSO

Instead of using the System Console, you can add the Entra ID settings directly to the `config.json` file on your Mattermost server.

1. Open `config.json` as *root* in a text editor. It's usually in `/opt/mattermost/config` but it might be elsewhere on your system.
2. Locate the `Office365Settings` section, then add or update the following information:

```
"Office365Settings": { "Enable": false, "Secret":
"i.hddd6Pu3--5dg~cRddddqOrBdd1a", "Id":
"28ddd714-1f2f-4f9c-9486-90b8dddd27", "Scope": "profile openid email",
"AuthEndpoint": "", "TokenEndpoint": "", "UserApiEndpoint": "",
"DiscoveryEndpoint": "https://login.microsoftonline.com/common/v2.0/.well-
known/openid-configuration", "DirectoryId": "common" }
```

3. Save your changes, then restart your Mattermost server. After the server restarts, users must change their login method before they can log in with Entra ID.

Frequently Asked Questions

How can I use LDAP attributes or Groups with OpenID?

At this time, LDAP data isn't compatible with OpenID. If you currently rely on LDAP to manage your users' teams, channels, groups, or attributes, you won't be able to do this automatically with users who have logged in with OpenID. If you need LDAP synced to each user, we suggest using SAML or LDAP as the login provider. Some OpenID providers can use SAML instead, like Keycloak.

On this page

- Entra ID Single Sign-On
  - Configuring EntraID as a Single Sign-On (SSO) service
    - Step 1: Register an application in Azure Portal
    - Step 2: Generate a new client secret in Azure Portal
    - Step 3: Configure Mattermost for Entra ID SSO
    - Note about Microsoft Active Directory Tenants
    - Configure Mattermost config.json for Entra ID SSO
  - Frequently Asked Questions
    - How can I use LDAP attributes or Groups with OpenID?

GitLab Single Sign-On

Configuring GitLab as a Single Sign-On (SSO) service

Follow these steps to configure Mattermost to use GitLab as a Single Sign-on (SSO) service for team creation, account creation, and user login.

Important

- Only the default GitLab SSO is officially supported.
- "Double SSO", where GitLab SSO is chained to other SSO solutions, is not supported. It may be possible to connect GitLab SSO with AD, LDAP, SAML, or MFA add-ons in some cases, but because of the special logic required, they're not officially supported, and they're known not to work in some cases.
- Mattermost Free (self-hosted only) supports the OAuth 2.0 standard.
- Mattermost Professional and Mattermost Enterprise support the OpenID Connect standard.

Step 1: Add a Mattermost application to your GitLab account

1. Log in to your GitLab account, then go to `https://{gitlab-site-name}/profile/applications`. For *{gitlab-site-name}* use the name of your GitLab instance. If you're using GitLab itself as your service provider, use `gitlab.com`.
2. Add a new application:

> *1. In the **Name** field, enter `Mattermost`.*
>
> *2. In the **Redirect URI** field, add the following two lines using your own value for {mattermost-site-name}.*

> > > ```
> > > https://{mattermost-site-name}/login/gitlab/complete https://{mattermost-site-name}/signup/gitlab/complete
> > > ```
> >
> > *If your GitLab instance is not set up to use SSL, your URIs must begin with `http://` instead of `https://`.*
>
> *3. Select scopes.*
>
> - *For Mattermost Team Edition, select `read_user`.*
> - *For Mattermost Enterprise, select `openid`, `profile`, and `email`.*

3. Select **Save application**.
4. Keep the GitLab window open. You need the *Application Id* and *Application Secret Key* when you configure Mattermost.

Step 2: Configure Mattermost for GitLab SSO

1. Log in to Mattermost, then go to **System Console > Authentication > OpenID Connect**.
2. Select **GitLab** as the service provider.
3. Enter the **GitLab Site URL** of your GitLab instance. If your GitLab instance is not set up to use SSL, start the URL with `http://` instead of `https://`. If you are using GitLab itself as your provider, use `gitlab.com`.
4. The **Discovery Endpoint** for OpenID Connect with GitLab is prepopulated with `https://gitlab.com/.well-known/openid-configuration`.
5. Paste the **Application ID** from GitLab as the **Client ID** in Mattermost.
6. Paste the **Application Secret Key** from GitLab as the **Client Secret** in Mattermost.
7. Update the `config.json` file and specify the scopes you selected in GitLab under the `GitLabSettings` property. At a minimum, `openid` is a required scope for Mattermost Enterprise and Professional, and `read_user` is a required scope for Mattermost Team Edition. Mattermost Team Edition does not work with scopes other than `read_user`. Changes to this setting require a server restart before taking effect.

8. Select **Save**.

> Note
>
> - When Mattermost is configured to use OpenID Connect or OAuth 2.0 for user authentication, the following user attribute changes can't be made through the Mattermost API: first name, last name, or username. OpenID Connect or OAuth 2.0 must be the authoritative source for these user attributes.
> - If you are using Mattermost behind a load balancer and you have SSL configured, you may need to set *X-Forwarded-Proto* header to https at your load balancer.

(Optional) Step 3: Force users to sign up using SSO only

To force all users to sign-up with SSO only, set **System Console > Authentication > Email > Enable sign-in with email** to `false` Users must change their login method before they can log in to Mattermost with GitLab.

Frequently Asked Questions

How can I use LDAP attributes or Groups with OpenID?

At this time, LDAP data isn't compatible with OpenID. If you currently rely on LDAP to manage your users' teams, channels, groups, or attributes, you won't be able to do this automatically with users who have logged in with OpenID. If you need LDAP synced to each user, we suggest using SAML or LDAP as the login provider. Some OpenID providers can use SAML instead, like Keycloak.

On this page

- GitLab Single Sign-On
  - Configuring GitLab as a Single Sign-On (SSO) service
    - Step 1: Add a Mattermost application to your GitLab account
    - Step 2: Configure Mattermost for GitLab SSO

    - (Optional) Step 3: Force users to sign up using SSO only
  - Frequently Asked Questions
    - How can I use LDAP attributes or Groups with OpenID?

Google Single Sign-On

Configuring Google Apps as a Single Sign-On (SSO) service

Follow these steps to configure Mattermost to use Google as a Single Sign-on (SSO) service for team creation, account creation, and login.

> Note
>
> The Google People API (https://developers.google.com/people) has replaced the Google+ API, which was deprecated by Google as of March 7th, 2019 per their notice (https://developers.google.com/+/api-shutdown).

Step 1: Create OpenID Connect project in Google API Manager

1. Go to Google Cloud Platform.
2. Select **Credentials** in the left-hand sidebar.
3. Select **Create Credentials**, then select **OAuth client ID**.
4. Select the **Web application** as the application type.
5. Enter `Mattermost-<your-company-name>` as the **Name**, replacing `<your-company-name>` with the name of your organization.
6. Under **Authorized redirect URIs**, select **Add URL**, then enter `{your-mattermost-url}/signup/google/complete`. For example: `http://localhost:8065/signup/google/complete`.
7. Select **Create**.

8. Copy and paste the **Your Client ID** and **Your Client Secret** values to a temporary location. You will enter these values in the Mattermost System Console.

Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information.

Application type *
Web application

Learn more about OAuth client types

Name *
Mattermost - your-company-name

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Authorized redirect URIs

For use with requests from a web server

URIs

http://localhost:8065/signup/google/complete

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to users within your organization unless the OAuth consent screen is published and verified.

Your Client ID
128998788295-5rm0nromg9jri68u5bn7rv3rvlrlvajn.apps.gc

Your Client Secret
0Vp9WIQiUccQHVb2JU2mpsJJ

Step 2: Enable Google People API

Go to the [Google People API (https://developers.google.com/people)](https://developers.google.com/people), then select **Enable** in the header. This might take a few minutes to propagate through Google's systems.

Step 3: Configure Mattermost for Google Apps SSO

1. Log in to Mattermost, then go to **System Console > Authentication > OpenID Connect**.
2. Select **Google Apps** as the service provider.
3. The **Discovery Endpoint** for OpenID Connect with Google Apps is prepopulated with `https://accounts.google.com/.well-known/openid-configuration`.
4. Paste in the **Client ID** from Google in Mattermost.
5. Paste in the **Client Secret** from Google in Mattermost.
6. Select **Save**.

> Note
>
> When Mattermost is configured to use OpenID Connect or OAuth 2.0 for user authentication, the following user attribute changes can't be made through the Mattermost API: first name, last name, or username. OpenID Connect or OAuth 2.0 must be the authoritative source for these user attributes.

Configure Mattermost `config.json` for Google Apps SSO

Note

Available only for self-hosted deployments

Instead of using the System Console, you can add the Google settings directly to the `config.json` file directly on your Mattermost server.

1. Open `config.json` as *root* in a text editor. It's usually in `/opt/mattermost/config`, but it might be elsewhere on your system.
2. Locate the `GoogleSettings` section, then add or update the following information:

```
"GoogleSettings": { "Enable": true, "Secret": "P-k9R-7E7ayX9LdddddWdXVg",
"Id": "1022ddddd5846-
bkddddd4a1ddddd9d88j1kb6eqc.apps.googleusercontent.com", "Scope": "profile
openid email", "AuthEndpoint": "", "TokenEndpoint": "", "UserApiEndpoint":
"", "DiscoveryEndpoint": "https://accounts.google.com/.well-known/openid-
configuration", "ButtonText": "", "ButtonColor": "" }
```

3. Save your changes, then restart your Mattermost server. After the server restarts, users must change their login method before they can log in with Google Apps.

Frequently Asked Questions

How can I use LDAP attributes or Groups with OpenID?

At this time, LDAP data isn't compatible with OpenID. If you currently rely on LDAP to manage your users' teams, channels, groups, or attributes, you won't be able to do this automatically with users who have logged in with OpenID. If you need LDAP synced to each user, we suggest using SAML or LDAP as the login provider. Some OpenID providers can use SAML instead, like Keycloak.

On this page

- Google Single Sign-On
  - Configuring Google Apps as a Single Sign-On (SSO) service
    - Step 1: Create OpenID Connect project in Google API Manager
    - Step 2: Enable Google People API
    - Step 3: Configure Mattermost for Google Apps SSO
    - Configure Mattermost config.json for Google Apps SSO
  - Frequently Asked Questions
    - How can I use LDAP attributes or Groups with OpenID?

OpenID Connect Single Sign-On

Mattermost provides OpenID Connect support for GitLab, Google Apps, and Entra ID. With OpenID Connect, users can also use their login to Keycloak, Atlassian Crowd, Apple, Microsoft, Salesforce, Auth0, Ory.sh, Facebook, Okta, OneLogin, and Azure AD, as well as others, as a Single Sign-on (SSO) service for team creation, account creation, and user login.

Follow these steps to configure a service provider using OpenID Connect.

Step 1: Create an OpenID Connect Application

1. Follow service provider documentation for creating an OpenID Connect application. Most OpenID Connect service providers require authorization of all redirect URIs.
2. In the appropriate field, enter `{your-mattermost-url}/signup/openid/complete` For example: `http://domain.com/signup/openid/complete`
3. Copy and paste values for the **Discovery Endpoint**, **Client ID**, and **Client Secret** values to a temporary location. You will enter these values when you configure Mattermost.

Step 2: Configure Mattermost for an OpenID Connect SSO

1. Log in to Mattermost, then go to **System Console > Authentication > OpenID Connect**.
2. Select **OpenID Connect (Other)** as the service provider.
3. Enter the **Discovery Endpoint**.
4. Enter the **Client ID**.
5. Enter the **Client Secret**.

6. Specify a **Button Name** and **Button Color** for the OpenID Connect option on the Mattermost login page.
7. Select **Save**.
8. Restart your Mattermost server to see the changes take effect.

> Note
>
> - When Mattermost is configured to use OpenID Connect for user authentication, the following user attribute changes can't be made through the Mattermost API: first name, last name, or username. OpenID Connect must be the authoritative source for these user attributes.
> - The **Discovery Endpoint** setting can be used to determine the connectivity and availability of arbitrary hosts. System admins concerned about this can use custom admin roles to limit access to modifying these settings. See the delegated granular administration documentation for details.

Frequently Asked Questions

How can I use LDAP attributes or Groups with OpenID?

At this time, LDAP data isn't compatible with OpenID. If you currently rely on LDAP to manage your users' teams, channels, groups, or attributes, you won't be able to do this automatically with users who have logged in with OpenID. If you need LDAP synced to each user, we suggest using SAML or LDAP as the login provider. Some OpenID providers can use SAML instead, like Keycloak.

On this page

- OpenID Connect Single Sign-On
  - Step 1: Create an OpenID Connect Application
  - Step 2: Configure Mattermost for an OpenID Connect SSO
  - Frequently Asked Questions
    - How can I use LDAP attributes or Groups with OpenID?

Configure SAML with Microsoft ADFS using Microsoft Windows Server 2016

This document provides steps to configure SAML 2.0 with Microsoft ADFS for Mattermost and Microsoft Windows Server 2016.

Before you begin

Before you begin, you need to generate encryption certificates for encrypting the SAML connection.

1. You can use the [Bash script (https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates)](https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates) from the `mattermost/docs` repository on GitHub, or any other suitable method. See the generate self-signed certificates documentation for details on generating a self-signed x509v3 certificate for use with multiple URLs / IPs.

2. Save the two files that are generated. They are the private key and the public key. In the System Console, they are referred to as the **Service Provider Private Key** and the **Service Provider Public Certificate** respectively.

Prerequisites

- An Active Directory instance where all users have specified email and username attributes. For Mattermost servers running 3.3 and earlier, users must also have their first name and last name attributes specified.

- A running Microsoft Server. The screenshots used in this guide are from Microsoft Server 2012R2, but similar steps should work for other versions.

- An SSL certificate to sign your ADFS login page.

- ADFS installed on your Microsoft Server. You can find a detailed guide for deploying and configuring ADFS in [this Microsoft article (https://

learn.microsoft.com/en-us/previous-versions/dynamicscrm-2016/deployment-administrators-guide/gg188612(v=crm.8)?redirectedfrom=MSDN).

On your ADFS installation, open the ADFS console. Select **Service**, then select **Endpoints**. In the **Type** column, search for `SAML 2.0/WS-Federation` and note down the value of **URL Path** column. This is also known as the **SAML SSO URL Endpoint** in this guide. If you chose the defaults for the installation, this will be `/adfs/ls`.

Add a relying party trust

1. Open the ADFS management snap-in, then select **AD FS > Relying Party Trusts > Add Relying Party Trust** from the right sidebar. You can also right-click **Relying Party Trusts**, then select **Add Relying Party Trust** from the context menu.

2. On the **Welcome** screen of the configuration wizard, select **Claims aware**, then select **Start**.

3. On the **Select Data Source** screen, select **Enter data about the relying party manually**.

4. On the **Specify Display Name** screen, enter a **Display Name** (e.g., `Mattermost`). You can add optional notes.

5. On the **Configure Certificate** screen, leave the certificate settings at their default values.

If you would like to set up encryption for your SAML connection, select **Browse**, then upload your Service Provider Public Certificate.

6. On the **Configure URL** screen, select **Enable Support for the SAML 2.0 WebSSO protocol**, then enter the **SAML 2.0 SSO service URL** in the following format: `https://<your-mattermost-url>/login/sso/saml` where `<your-mattermost-url>` should typically match the Mattermost Site URL.

7. On the **Configure Identifiers** screen, enter the **Relying party trust identifier**. This identifies the claims being requested. The **SAML 2.0 SSO service URL** format should be `https://<your-mattermost-url>/login/sso/saml` where `<your-mattermost-url>` matches your Mattermost Site URL. Then choose **Next**.

Add Relying Party Trust Wizard
Configure Identifiers
Steps
Welcome
Select Data Source
Specify Display Name
Configure Certificate
Configure URL
Configure Identifiers
Choose Access Control Policy
Ready to Add Trust
Finish
Relying parties may be identified by one or more unique identifier strings. Specify the identifiers for this relying party trust.
Relying party trust identifier:
https://<your-idp-url>/adfs/services/trust
Add
Example: https://fs.contoso.com/adfs/services/trust
Relying party trust identifiers:
Remove
< Previous
Next >
Cancel

Add Relying Party Trust Wizard
Configure Identifiers
Steps
Welcome
Select Data Source
Specify Display Name
Configure Certificate
Configure URL
Configure Identifiers
Choose Access Control Policy
Ready to Add Trust
Finish
Relying parties may be identified by one or more unique identifier strings. Specify the identifiers for this relying party trust.
Relying party trust identifier:
Add
Example: https://fs.contoso.com/adfs/services/trust
Relying party trust identifiers:
https://<your-idp-url>/adfs/services/trust
Remove
< Previous
Next >
Cancel

This string must match the **Service Provider Identifier** string. For more information about the Relying party trust identifier and how prefix matching is applied see this documentation.

Add your **SAML 2.0 SSO service URL** using this same process.

8. On the **Choose Access Control Policy** screen, select the access control policy suitable for your environment. This guide assumes the default values **Permit everyone** and an unchecked box.

9. On the **Ready to Add Trust** screen, review your settings.

10. On the **Finish** screen, select **Configure claims issuance policy for this application**, then select **Close**.

Create claim rules

1. In the **Issuance Transform Rules** tab of the **Claim Rules** editor, select **Add Rule...**.

2. On the **Choose Rule Type** screen, select **Send LDAP Attributes as Claims** from the drop-down menu, then select **Next**.

3. On the **Configure Claim Rule** screen, enter a **Claim Rule Name** of your choice, select **Active Directory** as the **Attribute Store**, then add the following mapping:

> - *From the **LDAP Attribute column**, select* `E-Mail-Addresses`*. From the **Outgoing Claim Type**, type* `Email`*.*
> - *From the **LDAP Attribute column**, select* `E-Mail-Addresses`*. From the **Outgoing Claim Type**, type* `Name ID`*.*
> - *From the **LDAP Attribute column**, select* `Given-Name`*. From the **Outgoing Claim Type**, type* `FirstName`*.*
> - *From the **LDAP Attribute column**, select* `Surname`*. From the **Outgoing Claim Type**, type* `LastName`*.*
> - *From the **LDAP Attribute column**, select* `SAM-Account-Name`*. From the **Outgoing Claim Type**, type* `Username`*.*

The `FirstName` and `LastName` attributes are optional.

Select **Finish** to add the rule.

The entries in the **Outgoing Claim Type** column can be modified. The entries may contain dashes but no spaces. They are used to map the corresponding fields in Mattermost.

4. Select **Add Rule** to create another new rule.

5. On the **Choose Rule Type** screen, select **Transform an Incoming Claim** from the drop-down menu, then select **Next**.

6. On the **Configure Claim Rule** screen, enter a **Claim Rule Name** of your choice, then:

> - *Select **Name ID** for the **Incoming claim type***
> - *Select **Unspecified** for the **Incoming name ID format***
> - *Select **E-Mail Address** for the **Outgoing claim type***

Select **Pass through all claim values**, then select **Finish**.

7. Select **Finish** to create the claim rule, then select **OK** to finish creating rules.

8. Open Windows PowerShell as an administrator, then run the following command:

```
Set-ADFSRelyingPartyTrust -TargetName <display-name> -SamlResponseSignature "MessageAndAssertion"
```

where `<display-name>``is the name you specified in step 4 when you added a relying party trust. In this example, ``<display-name>` would be `mattermost`.

This action adds the signature to SAML messages, making verification successful.

Export identity provider certificate

Next, export the identity provider certificate, which will be later uploaded to Mattermost to finish SAML configuration.

1. Open the ADFS management snap-in, select **AD FS > Service > Certificates**, then double-click on the certificate under **Token-signing**. You can also right-click the field, then select **View Certificate** in the context menu.

2. On the **Certificate** screen, open the **Details** tab, select **Copy to File**, then select **OK**.

3. On the **Certificate Export Wizard** screen, select **Next**.

4. Select **Base-64 encoded X.509 (.CER)**, then select **Next** again.

5. On the **Certificate Export Wizard** screen, select **Browse** to specify the location where you want the Identity Provider Certificate to be exported, then specify the file name.

6. Select **Save**. On the **Certificate Export Wizard** screen, verify the file path is correct, then select **Next**.

7. In the **Completing the Certificate Export Wizard**, select **Finish**, then select **OK** to confirm the export was successful.

Configure SAML Sign-On for Mattermost

Create a metadata URL by appending "FederationMetadata/2007-06/FederationMetadata.xml" to the root URL of the ADFS server, for example: `https://<adfs.domain.com>/federationmetadata/2007-06/FederationMetadata.xml>`.

Next, start the Mattermost server, then log in to Mattermost as a system admin. Go to **System Console > Authentication > SAML**, paste the metadata URL in the **Identity Provider Metadata URL** field, then select **Get SAML Metadata from IdP**.

This populates the **SAML SSO URL** and the **Identity Provider Issuer URL** fields automatically. The Identity Provider Public Certificate is also downloaded from the server and set locally.

The following fields can be selected:

- Set **Enable Login With SAML 2.0** to **true**.
- Set **Enable Synchronizing SAML Accounts With AD/LDAP** to suit your environment.
- Set **Override SAML bind data with AD/LDAP information** to suit your environment.

If you don't plan to use a metadata URL, you can manually enter the following fields:

- For **SAML SSO URL** use the `SAML 2.0/W-Federation URL ADFS Endpoint` you copied at the beginning of the process.
- For **Identity Provider Issuer URL** use the `Relying party trust identifier` from ADFS.
- For **Identity Provider Public Certificate** use the``X.509 Public Certificate``.

2. Configure Mattermost to verify the signature.

> - *Set **Verify Signature** to `true`.*
> - *For **Service Provider Login URL** use the `SAML 2.0 SSO service URL` you specified in ADFS.*
>
>
>

3. Enable encryption.

> - *Set **Enable Encryption** to `true`.*
> - *For **Service Provider Private Key** use the Service Provider Private Key generated at the start of this process.*
> - *For **Service Provider Public Certificate** use the Service Provider Public Certificate you generated at the start of this process.*

> - *Set **Sign Request** to suit your environment.*
>
> > Enable Encryption: True False
> > When false, Mattermost will not decrypt SAML Assertions encrypted with your Service Provider Public Certificate. Disabling encryption is not recommended for production environments.
> >
> > Service Provider Private Key: Choose File Upload
> > No file uploaded
> > The private key used to decrypt SAML Assertions from the Identity Provider.
> >
> > Service Provider Public Certificate: Choose File Upload
> > No file uploaded
> > The certificate used to generate the signature on a SAML request to the Identity Provider for a service provider initiated SAML login, when Mattermost is the Service Provider.

4. Set attributes for the SAML Assertions, which will be used to update user information in Mattermost. Attributes for email and username are required and should match the values you entered in ADFS earlier. See documentation on SAML configuration settings for more detail.

For Mattermost servers running 3.3 and earlier, the first name and last name attributes are also required fields.

5. Select **Save**.

6. (Optional) If you configured `First Name` Attribute and the `Last Name` Attribute, go to **System Console > Site Configuration > Users and Teams**, then set **Teammate Name Display** to **Show first and last name**. This is recommended for a better user experience.

You're done! If you'd like to confirm SAML SSO is successfully enabled, switch your system admin account from email to SAML-based authentication from your profile picture via **Profile > Security > Sign-in Method > Switch to SAML SSO**, then log in with your SAML credentials to complete the switch.

We also recommend that you post an announcement about how the migration will work to your users.

You may also configure SAML for ADFS by editing the `config.json` file to enable SAML based on SAML configuration settings. You must restart the Mattermost server for the changes to take effect.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.

- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See "How do I deactivate users?" for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn't deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.
3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.

4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.

2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.

3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and

comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

On this page

- Configure SAML with Microsoft ADFS using Microsoft Windows Server 2016
  - Before you begin
  - Prerequisites
  - Add a relying party trust
  - Create claim rules
  - Export identity provider certificate
  - Configure SAML Sign-On for Mattermost
  - Frequently Asked Questions
    - What encryption options are supported for SAML?
    - How to bind authentication to Id attribute instead of email
    - Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?
    - Can I provision and deprovision users who log in via SAML?
    - How do I migrate users from one authentication method (e.g. email) to SAML?
    - How is SAML different from OAuth 2.0 and OpenId Connect?
  - Configure SAML synchronization with AD/LDAP
    - Override SAML data with AD/LDAP data

Configure SAML with Microsoft ADFS for Windows Server 2012

The following process provides steps to configure SAML 2.0 with Microsoft ADFS for Mattermost.

Before you begin

Before you begin, you need to generate encryption certificates for encrypting the SAML connection.

1. You can use the [Bash script (https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates)](https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates) from the `mattermost/docs` repository on GitHub, or any other suitable method. See the generate self-signed certificates documentation for details on generating a self-signed x509v3 certificate for use with multiple URLs / IPs.

2. Save the two files that are generated. They are the private key and the public key. In the System Console, they are referred to as the **Service Provider Private Key** and the **Service Provider Public Certificate** respectively.

The following are basic requirements to use ADFS for Mattermost:

- An Active Directory instance where all users have a specified email and username attributes. For Mattermost servers running 3.3 and earlier, users must also have their first name and last name attributes specified.

- A Microsoft Server running. The screenshots used in this guide are from Microsoft Server 2012R2, but similar steps should work for other versions.

- An SSL certificate to sign your ADFS login page.

- ADFS installed on your Microsoft Server. You can find a detailed guide for deploying and configuring ADFS in [this Microsoft article (https://learn.microsoft.com/en-us/previous-versions/dynamicscrm-2016/](https://learn.microsoft.com/en-us/previous-versions/dynamicscrm-2016/)

deployment-administrators-guide/gg188612(v=crm.8)?redirectedfrom=MSDN).

On your ADFS installation, note down the value of the **SAML 2.0/W-Federation URL** in ADFS Endpoints section, also known as the **SAML SSO URL Endpoint** in this guide. If you chose the defaults for the installation, this will be `/adfs/ls/`.

Add a relying party trust

1. In the ADFS management sidebar, go to **AD FS > Trust Relationships > Relying Party Trusts**, then select **Add Relying Party Trust**. A configuration wizard opens for adding a new relying party trust.

2. On the **Welcome** screen, select **Start**.

3. On the **Select Data Source** screen, select **Enter data about the relying party manually**.

4. On the **Specify Display Name** screen, enter a **Display Name** to recognize the trust, such as `Mattermost`, then add any notes you want to make.

5. On the **Choose Profile** screen, select **AD FS profile**.

6. On the **Configure Certificate** screen, leave the certificate settings at their default values.

However, if you would like to set up encryption for your SAML connection, select **Browse**, then upload your Service Provider Public Certificate.

7. On the **Configure URL** screen, select **Enable Support for the SAML 2.0 WebSSO protocol**, then enter the **SAML 2.0 SSO service URL**, similar to `https://<your-mattermost-url>/login/sso/saml` where `<your-mattermost-url>` should typically match the Mattermost Site URL.

8. On the **Configure Identifiers** screen, enter the **Relying party trust identifier** (also known as the **Identity Provider Issuer URL**) of the form `https://<your-idp-url>/adfs/services/trust`, then click **Add**.

9. On the **Configure Multi-factor Authentication Now** screen, you may enable multi-factor authentication. This is beyond the scope of this documentation.

10. On the **Choose Issuance Authorization Rules** screen, select **Permit all users to access this relying party**.

11. On the **Ready to Add Trust** screen, review your settings.

12. On the **Finish** screen, select **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes**, then select **Close**. You exit the configuration wizard, and a **Claim Rules** editor opens.

Create claim rules

1. In the **Issuance Transform Rules** section of the **Claim Rules** editor, select **Add Rule...** to open an **Add Transform Claim Rule Wizard**.

2. On the **Choose Rule Type** screen, select **Send LDAP Attributes as Claims** from the drop-down menu, then select **Next**.

3. In the **Configure Claim Rule** screen, enter a **Claim Rule Name** of your choice, select **Active Directory** as the **Attribute Store**, then complete the following:

> - *From the **LDAP Attribute column**, select* `E-Mail-Addresses`*. From the **Outgoing Claim Type**, type* `Email`*.*
> - *From the **LDAP Attribute column**, select* `E-Mail-Addresses`*. From the **Outgoing Claim Type**, type* `Name ID`*.*
> - *From the **LDAP Attribute column**, select* `Given-Name`*. From the **Outgoing Claim Type**, type* `FirstName`*.*
> - *From the **LDAP Attribute column**, select* `Surname`*. From the **Outgoing Claim Type**, type* `LastName`*.*
> - *From the **LDAP Attribute column**, select* `SAM-Account-Name`*. From the **Outgoing Claim Type**, type* `Username`*.*

The `FirstName` and `LastName` attributes are optional.

Select **Finish** to add the rule.

Note that the entries in the **Outgoing Claim Type** column can be chosen to be something else. They can contain dashes but no spaces. They will be used to map the corresponding fields in Mattermost later.

4. Create another new rule by selecting **Add Rule**.

5. On the **Choose Rule Type** screen, select **Transform an Incoming Claim** from the drop-down menu, then select **Next**.

6. On the **Configure Claim Rule** screen, enter a **Claim Rule Name** of your choice, then:

> - *Select **Name ID** for the **Incoming claim type**.*
> - *Select **Unspecified** for the **Incoming name ID format**.*
> - *Select **E-Mail Address** for the **Outgoing claim type**.*

Select **Pass through all claim values**, then select **Finish**.

7. Select **Finish** to create the claim rule, then select **OK** to finish creating rules.

8. Open Windows PowerShell as an administrator, then run the following command:

```
Set-ADFSRelyingPartyTrust -TargetName <display-name> -SamlResponseSignature "MessageAndAssertion"
```

where `<display-name>` is the name you specified in step 4 when adding a relying party trust. In this example, `<display-name>` would be `mattermost`.

This action adds the signature to SAML messages, making verification successful.

Export identity provider certificate

Next, export the identity provider certificate, which will be later uploaded to Mattermost to finish SAML configuration.

1. In the ADFS management sidebar, go to **AD FS > Service > Certificates**, then double click on the certificate under **Token-signing**. Alternatively, you can right-click on the field, then select **View Certificate**.

2. On the **Certificate** screen, go to the **Details** tab, then select **Copy to File**, followed by **OK**. This opens a **Certificate Export Wizard**.

3. On the **Certificate Export Wizard** screen, select **Next**, then, select the option **Base-64 encoded X.509 (.CER)**, and select **Next** again.

4. On the **Certificate Export Wizard** screen, select **Browse** to specify the location where you want the Identity Provider Certificate to be exported, then specify the file name.

5. Select **Save**. In the **Certificate Export Wizard** screen, verify the file path is correct, then select **Next**.

6. In the **Completing the Certificate Export Wizard**, select **Finish**, then select **OK** to confirm the export was successful.

Configure SAML Sign-On for Mattermost

Create a metadata URL by appending "FederationMetadata/2007-06/FederationMetadata.xml" to the root URL of the ADFS server, for example: `https://<adfs.domain.com>/federationmetadata/2007-06/FederationMetadata.xml>`.

Next, start the Mattermost server and log in to Mattermost as a system admin. Go to **System Console > Authentication > SAML**, paste the metadata URL in the **Identity Provider Metadata URL** field, then select **Get SAML Metadata from IdP**.

This populates the **SAML SSO URL** and the **Identity Provider Issuer URL** fields automatically. The Identity Provider Public Certificate is also downloaded from the server and set locally.

Alternatively you can enter the following fields manually:

- **SAML SSO URL**: **SAML 2.0/W-Federation URL** ADFS Endpoint you copied earlier.
- **Identity Provider Issuer URL**: `Relying party trust identifier` from ADFS you specified earlier.
- **Identity Provider Public Certificate**: `X.509 Public Certificate` you downloaded earlier.

2. Configure Mattermost to verify the signature. The **Service Provider Login URL** is the SAML 2.0 SSO service URL you specified in ADFS earlier.

3. Enable encryption by uploading the Service Provider Private Key and Service Provider Public Certificate you generated earlier.

4. Configure Mattermost to sign SAML requests using the Service Provider Private Key.

5. Set attributes for the SAML Assertions, which will be used to update user information in Mattermost. Attributes for email and username are required and should match the values you entered in ADFS earlier. See documentation on SAML configuration settings for more detail.

For Mattermost servers running 3.3 and earlier, the `FirstName` and `LastName` attributes are also required fields.

Email Attribute:

E.g.: "Email" or "PrimaryEmail"

The attribute in the SAML Assertion that will be used to populate the email addresses of users in Mattermost.

Username Attribute:

E.g.: "Username"

The attribute in the SAML Assertion that will be used to populate the username field in Mattermost.

Id Attribute:

E.g.: "Id"

(Optional) The attribute in the SAML Assertion that will be used to bind users from SAML to users in Mattermost.

Guest Attribute:

E.g.: "usertype=Guest" or "isGuest=true"

(Optional) Requires Guest Access to be enabled before being applied. The attribute in the SAML Assertion that will be used to apply a guest role to users in Mattermost. Guests are prevented from accessing teams or channels upon logging in until they are assigned a team and at least one channel.

Note: If this attribute is removed/changed from your guest user in SAML and the user is still active, they will not be promoted to a member and will retain their Guest role. Guests can be promoted in **System Console** > **User Management**.

Existing members that are identified by this attribute as a guest will be demoted from a member to a guest when they are asked to login next. The next login is based upon Session lengths set in **System Console** > **Session Lengths**. It is highly recommend to manually demote users to guests in **System Console** > **User Management** to ensure access is restricted immediately.

Enable Admin Attribute:

○ True ◉ False

Admin Attribute:

E.g.: "usertype=Admin" or "isAdmin=true"

(Optional) The attribute in the SAML Assertion for designating System Admins. The users selected by the query will have access to your Mattermost server as System Admins. By default, System Admins have complete access to the Mattermost System Console.

Existing members that are identified by this attribute will be promoted from member to System Admin upon next login. The next login is based upon Session lengths set in **System Console** > **Session Lengths**. It is highly recommend to manually demote users to members in **System Console** > **User Management** to ensure access is restricted immediately.

Note: If this filter is removed/changed, System Admins that

6. (Optional) Customize the login button text.

> **Login Button Text:** SAML
>
> (Optional) The text that appears in the login button on the login page. Defaults to "SAML".

7. Select **Save**.

8. (Optional) If you configured a `FirstName` and `LastName` Attribute, go to **System Console > Site Configuration > Users and Teams**, then set **Teammate Name Display** to **Show first and last name**. This is recommended for a better user experience.

If you'd like to confirm SAML SSO is successfully enabled, switch your system admin account from email to SAML-based authentication from your profile picture via **Profile > Security > Sign-in Method > Switch to SAML SSO**, then log in with your SAML credentials to complete the switch.

We recommend that you post an announcement about how the migration will work for your users.

You may also configure SAML for ADFS by editing the `config.json` file to enable SAML based on SAML configuration settings. You must restart the Mattermost server for the changes to take effect.

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.

2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.

3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.

4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.

2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.

3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend

using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This

configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.
- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See “How do I deactivate users?” for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn’t deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

On this page

- Configure SAML with Microsoft ADFS for Windows Server 2012
  - Before you begin
  - Add a relying party trust
  - Create claim rules
  - Export identity provider certificate
  - Configure SAML Sign-On for Mattermost
  - Configure SAML synchronization with AD/LDAP
    - Override SAML data with AD/LDAP data
  - Frequently Asked Questions
    - What encryption options are supported for SAML?

- How to bind authentication to Id attribute instead of email
- Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?
- Can I provision and deprovision users who log in via SAML?
- How do I migrate users from one authentication method (e.g. email) to SAML?
- How is SAML different from OAuth 2.0 and OpenId Connect?

Before you begin

Before you begin, you need to generate encryption certificates for encrypting the SAML connection.

1. You can use the [Bash script (https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates)](https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates) from the `mattermost/docs` repository on GitHub, or any other suitable method. See the generate self-signed certificates documentation for details on generating a self-signed x509v3 certificate for use with multiple URLs / IPs.

2. Save the two files that are generated. They are the private key and the public key. In the System Console, they are referred to as the **Service Provider Private Key** and the **Service Provider Public Certificate** respectively.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.

- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See "How do I deactivate users?" for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn't deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

On this page

- Frequently Asked Questions
  - What encryption options are supported for SAML?
  - How to bind authentication to Id attribute instead of email
  - Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?
  - Can I provision and deprovision users who log in via SAML?
  - How do I migrate users from one authentication method (e.g. email) to SAML?
  - How is SAML different from OAuth 2.0 and OpenId Connect?

Configure SAML with Keycloak

The following process provides steps to configure SAML with Keycloak for Mattermost.

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Set up a connection app for Mattermost SSO

> Note
>
> This was tested with Keycloak v22.0.5.
>
> Additionally, you don't typically need to make a realm dedicated to Mattermost. We recommend adding Mattermost as a client to your primary realm.

1. Log in to Keycloak as an administrator.
2. Select **Clients**, then **Create client**. You'll use this client ID in a later step.
   - ***Client type***: `SAML`
   - ***Client ID***: `mattermost`
3. Click **Next** and update the following values:
   - ***Home URL***: `https://<<siteURL>>/login/sso/saml`
   - ***Valid redirect URIs***: `https://<<siteURL>>/login/sso/saml`
4. **Save** the client.
5. Under the **Settings** tab, update the following values:
   - ***Enabled***: ***On***
   - ***Name ID format***: `email`

- ***Force Name ID format**: On*
- ***Sign Documents**: Off*

6. Navigate to the **Keys** tab.

- ***Client signature required**: Off*
- ***Encrypt Assertions**: On*
  - *Click **Generate***
  - *Download the **private.key** file.*
  - *Click **Confirm***
- *Click **Export** and update the following values:*
  - ***Archive Format**: `PKCS12`*
  - ***Key Alias**: `mattermost`*
  - ***Key Password**: `mattermost`*

> - ***Store Password***: `mattermost`
>
> > *Note*
> >
> > *In the image below, we used Mattermost for the Realm Certificate Alias. You should use your Realm Name here, as it's designed to identify what this certificate is within your Keycloak realm.*
>
>
>

7. Navigate to the **Client scopes** tab.

> 1. *Select **mattermost-dedicated***
> 2. *Click **Add predefined mapper***
> 3. *Select the **X500 email**, **X500 givenName**, and **X500 surname** attributes.*
> 4. *Click **Add**.*

8. Add the username and ID attribute.

> *If you're planning to sync your SAML users with LDAP within Mattermost, the ID value used here must match with your Mattermost LDAP* `ID Attribute`*.*
>
> 1. *Select* ***Add Mapper*** *->* ***By Configuration*** *-> User Property.*
> 2. *Set* ***Name*** *to* `Username`*.*
> 3. *Set* ***Property*** *to* `username` *(This is case sensitive and must be lowercase).*
> 4. *Set* ***SAML Attribute Name*** *to* `username`*.*
> 5. *Select* ***Save****.*
> 6. *Repeat the above steps and use the property of* `id` *to create the ID Attribute.*

Clients > Client details > Dedicated scopes > Mapper details

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type	User Property
Name *	ID
Property	id
Friendly Name	
SAML Attribute Name	id
SAML Attribute NameFormat	Basic

Save Cancel

Once done your Mappers should look like this:

Clients > Client details > Dedicated scopes

mattermost

This is a client scope which includes the dedicated mappers and scope

Mappers Scope

Search for mapper Add mapper 1 - 5

Name	Category	Type	Priority
Username	AttributeStatement Mapper	User Property	0
X500 givenName	AttributeStatement Mapper	User Property	0
X500 surname	AttributeStatement Mapper	User Property	0
X500 email	AttributeStatement Mapper	User Property	0
ID	AttributeStatement Mapper	User Property	0

1 - 5

9. Get the metadata URL from Keycloak:

> 1. *Within your Realm, select **Realm Settings**.*
>
> 2. *At the bottom of the **General** tab you should see a **SAML 2.0 Identity Provider Metadata** endpoint. Right-click and copy this URL. Store for the next step.*

Configure SAML for Mattermost

1. Start the Mattermost server and log in to Mattermost as a system admin. Go to **System Console > Authentication > SAML**.

2. Set the **Identity Provider Metadata URL** to the value you copied from the step above and select **Get SAML Metadata from IdP**. The metadata import will populate fields related to your Keycloak configuration.

> *If you have any issues with this import, you can check the `mattermost.log` file for more information. Enable debug logging and try again.*

3. Set the below fields:

> - ***Verify Signature**: false*
> - ***Service Provider Login URL**:* `http://<<siteURL>>/login/sso/saml`
> - ***Service Provider Identifier**:* `mattermost`
>
> *The Service Provider Identifier will match the **Client ID** that you configured in the second Keycloak step.*

4. Configure the Encryption using the key you downloaded in step 6 of the Keycloak config.

> 1. *Generate the* `.crt` *file from the* `.p12` *file.*
>
> > `openssl pkcs12 -in keystore.p12 -out mattermost.crt -nodes`
>
> 2. *Generate the* `.key` *file from the* `.p12` *file.*
>
> > `openssl pkcs12 -in keystore.p12 -out mattermost.key -nodes -nocerts`
>
> 3. *Upload both of these files within the Mattermost System Console. Make sure to select **Upload**.*
>
> > - ***Service Provider Private Key**:* `mattermost.key`
> > - ***Service Provider Private Certificate**:* `mattermost.crt`

>
>

5. (Optional) Set up request signing with the below parameters.

>
>

6. Set attributes for the SAML Assertions, which will update user information in Mattermost.

> *The attributes below are from steps 7 and 8 above. These values must be the **SAML Attribute Name** within Keycloak. See documentation on SAML configuration settings for more details.*
>
> - ***Email Attribute***: `email`
> - ***Username Attribute***: `username`
> - ***Id Attribute***: `id`

7. Select **Save**.

You're done! If you'd like to confirm SAML SSO is successfully enabled, switch your system admin account from email to SAML-based authentication via **Profile > Security > Sign-in Method > Switch to SAML SSO** and log in with your SAML credentials to complete the switch.

It's also recommended to post an announcement about how the migration will work for users.

You may also configure SAML for Keycloak by editing `config.json`. Before starting the Mattermost server, edit `config.json` to enable SAML based on SAML configuration settings. You must restart the Mattermost server for the changes to take effect.

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.

2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.

3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.

4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.

2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.

3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend

using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This

configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.
- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See “How do I deactivate users?” for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn’t deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

On this page

- Configure SAML with Keycloak
  - Set up a connection app for Mattermost SSO
  - Configure SAML for Mattermost
  - Configure SAML synchronization with AD/LDAP
    - Override SAML data with AD/LDAP data
  - Frequently Asked Questions
    - What encryption options are supported for SAML?
    - How to bind authentication to Id attribute instead of email
    - Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

- Can I provision and deprovision users who log in via SAML?
- How do I migrate users from one authentication method (e.g. email) to SAML?
- How is SAML different from OAuth 2.0 and OpenId Connect?

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.
3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.
4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.
2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.
3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.
4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.

4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

On this page

- Configure SAML synchronization with AD/LDAP
  - Override SAML data with AD/LDAP data

Configure SAML with Okta

The following process provides steps to configure SAML 2.0 with Okta for Mattermost.

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Before you begin

Before you begin, you need to generate encryption certificates for encrypting the SAML connection.

1. You can use the [Bash script (https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates)](https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates) from the `mattermost/docs` repository on GitHub, or any other suitable method. See the generate self-signed certificates documentation for details on generating a self-signed x509v3 certificate for use with multiple URLs / IPs.

2. Save the two files that are generated. They are the private key and the public key. In the System Console, they are referred to as the **Service Provider Private Key** and the **Service Provider Public Certificate** respectively.

Set Up a connection app for Mattermost Single Sign-On

1. Log in to Okta as an administrator.

2. Switch to the **Classic UI**, using the drop-down in the upper left.

3. Go to **Admin Dashboard > Applications > Add Application**.

4. Select **Create New App**, then choose **SAML 2.0** as the Sign on method.

5. Enter **General Settings** for the application, including **App name** and **App logo** (optional). It's recommended to display the application icon to users, including in the Okta Mobile app. If you'd like to use a Mattermost logo for the application, you can download one [from our page (https://handbook.mattermost.com/operations/operations/publishing/publishing-guidelines/brand-and-visual-design-guidelines)](https://handbook.mattermost.com/operations/operations/publishing/publishing-guidelines/brand-and-visual-design-guidelines).

6. Enter **SAML Settings**, including:

> - ***Single sign on URL:*** `https://<your-mattermost-url>/login/sso/saml` *where* `https://<your-mattermost-url>` *should typically match the Mattermost Site URL.*
>
> - ***Audience URI:*** *For instance,* `mattermost`

- ***Name ID format:*** `unspecified`
- ***Application username:*** `Email`

7. To set up encryption for your SAML connection, select **Show Advanced Settings**.

8. Set **Assertion Encryption** as **Encrypted**, then upload the Service Provider Public Certificate you generated earlier to the **Encryption Certificate** field.

9. Enter attribute statements used to map attributes between Okta and Mattermost. For more information on which attributes are configurable, see our documentation on SAML configuration settings. Email and username attributes are required. For SAML with Okta, an ID attribute is also required, and that ID must be mapped to `user.id`.

10. Select **Next**. Then, set Okta support parameters for the application. Recommended settings:

    - ***I'm an Okta customer adding an internal app***
    - ***This is an internal app that we have created***

11. Select **Finish**.

12. In the Mattermost System Console, go to **Authentication > SAML 2.0**, then set **Override SAML bind data with AD/LDAP information** to **false** if currently set to **true**. You can re-enable this configuration setting later when once setup is complete.

13. On the next screen, select the **Sign On** tab, then select **View Setup Instructions**.

14. Select the **Identity Provider metadata** link, then copy the link from the browser URL field. This will be used during the SAML configuration steps in the next section.

15. Take note of **Identity Provider Single Sign-On URL** (also known as **SAML SSO URL**), and the Identity Provider Issuer, as both may be needed to configure SAML for Mattermost.

16. Download the X.509 Certificate file and save it. You may need to upload it to Mattermost in a later step.

Configure SAML Sign-On for Mattermost

Start the Mattermost server and log in to Mattermost as a system admin. Go to **System Console > Authentication > SAML 2.0**, then paste the copied Identity Provider Metadata URL in the **Identity Provider Metadata URL** field and select **Get SAML Metadata from IdP**.

This populates the **SAML SSO URL** and the **Identity Provider Issuer URL** fields automatically. The Identity Provider Public Certificate is also downloaded from the server and set locally.

Alternatively you can enter the following fields manually:

- **SAML SSO URL:** `Identity Provider Single Sign-On URL` from Okta, specified earlier.
- **Identity Provider Issuer URL:** `Identity Provider Issuer` from Okta, specified earlier.
- **Identity Provider Public Certificate:** X.509 Public Certificate file you downloaded from Okta earlier.

2. Configure Mattermost to verify the signature. The **Service Provider Login URL** is the `Single sign on URL` you specified in Okta earlier.

3. Enable encryption based on the parameters provided earlier.

4. Configure Mattermost to sign SAML requests using the Service Provider Private Key.
5. Set attributes for the SAML Assertions used to update user information in Mattermost.
    - Attributes for Email, Username, and Id are required and should match the values you entered in Okta earlier.

6. (Optional) Customize the login button text.

7. Select **Save**.

8. (Optional) If you configured `First Name` Attribute and `Last Name` Attribute, go to **System Console > Site Configuration > Users and Teams**, then set **Teammate Name Display** to **Show first and last name**. This is recommended for a better user experience.

Once complete, and to confirm SAML SSO is successfully enabled, switch your system admin account from email to SAML-based authentication from your profile picture via **Profile > Security > Sign-in Method > Switch to SAML SSO**, then log in with your SAML credentials to complete the switch.

We also recommend that you post an announcement for your users to explain how the migration will work.

You may also configure SAML for Okta by editing the `config.json` file to enable SAML based on SAML configuration settings. You must restart the Mattermost server for the changes to take effect.

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.
3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.

4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.

2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.

3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and

comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.
- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See “How do I deactivate users?” for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn’t deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

On this page

- Configure SAML with Okta
  - Before you begin
  - Set Up a connection app for Mattermost Single Sign-On
  - Configure SAML Sign-On for Mattermost
  - Configure SAML synchronization with AD/LDAP
    - Override SAML data with AD/LDAP data
  - Frequently Asked Questions
    - What encryption options are supported for SAML?
    - How to bind authentication to Id attribute instead of email

- Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?
- Can I provision and deprovision users who log in via SAML?
- How do I migrate users from one authentication method (e.g. email) to SAML?
- How is SAML different from OAuth 2.0 and OpenId Connect?

Configure SAML with OneLogin

The following process provides steps to configure SAML 2.0 with OneLogin for Mattermost.

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Before you begin

Before you begin, you need to generate encryption certificates for encrypting the SAML connection.

1. You can use the [Bash script (https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates)](https://github.com/mattermost/docs/tree/master/source/scripts/generate-certificates) from the `mattermost/docs` repository on GitHub, or any other suitable method. See the generate self-signed certificates documentation for details on generating a self-signed x509v3 certificate for use with multiple URLs / IPs.

2. Save the two files that are generated. They are the private key and the public key. In the System Console, they are referred to as the **Service Provider Private Key** and the **Service Provider Public Certificate** respectively.

Create a OneLogin connection app for Mattermost SSO

1. Add a SAML test connector app.

> *1. Log in to OneLogin as an administrator.*
>
> *2. Go to **Apps > Add Apps**.*
>
> *3. Search for "SAML Test Connector", then select **SAML Test Connector (Advanced)**.*

4. *In the **Display Name** field, enter a name for the application, then optionally upload an app icon. You can use the Mattermost logo for the icon, which you can download from [Branding Guidelines (https://handbook.mattermost.com/operations/operations/publishing/publishing-guidelines/brand-and-visual-design-guidelines)](https://handbook.mattermost.com/operations/operations/publishing/publishing-guidelines/brand-and-visual-design-guidelines) page.*

5. *Make sure that the **Visible in portal** option is enabled.*

> 6. *Select **Save**.*

2. Configure the app.

> 1. *Select the **Configuration** tab, then enter the following values:*
>
> - ***RelayState**: leave blank*
> - ***Audience**: leave blank*
> - ***Recipient**:* `https://<your-mattermost-url>/login/sso/saml` *where* `https://<your-mattermost-url>` *should typically match the Mattermost Site URL.*
> - ***ACS (Consumer) URL Validator**:* `https:\/\/<your-mattermost-url>\/login\/sso\/saml`
> - ***ACS (Consumer) URL**:* `https://<your-mattermost-url>/login/sso/saml`

> 2. *In System Console, enable encryption, then select **Save**. You're redirected to the **Info** tab. From there, select the **Configuration** tab to access the **SAML Encryption** field.*
>
> 3. *Paste the Public Key that you generated earlier into the **SAML Encryption** field at the bottom of the page. This field displays in OneLogin only when encryption is enabled in Mattermost.*

*4. Select **Save**.*

3. Enter the attribute parameters. The following attributes are recommended:

Name	**Value**
ID	UUID
Email	Email
FirstName	First Name
LastName	Last Name
Name ID value	Email
Username	Username

Attribute parameters map attributes between OneLogin and Mattermost. For more information on which attributes are configurable, see our documentation on SAML configuration settings.

Email attributes are required.

*1. Select the **Parameters** tab.*

*2. Select **Add Parameter**.*

3. *In the **Field name** field, enter an attribute parameter such as* `Email`.

4. *Select the **Include in SAML assertion** checkbox.*

5. *Select **Save**.*

6. *Select **Edit**.*

7. *In the **Value** field, select the OneLogin value that corresponds to the attribute parameter.*

8. *Repeat the steps above to add any other attributes that you need. After you've added all the attributes you want to use, the parameter list should look similar to the following image:*

4. Copy the SSO information.

1. *Select the **SSO** tab.*

2. *Copy the values in the **Issuer URL** and **SAML 2.0 Endpoint (HTTP)** fields, then save them for later use.*

*3. Select **View Details** to view the X.509 certificate.*

*4. Make sure that the **X.509 PEM** option is selected in the drop-down.*

*5. Select **DOWNLOAD**, then save the file in a convenient location for later use.*

5. Save all your changes.

Configure SAML Sign-On for Mattermost

1. Start the Mattermost server, then log in to Mattermost as a system admin. Go to **System Console > Authentication > SAML**.

> 1. *Enter the **OneLogin Issuer URL** into the **Identity Provider Metadata URL** field.*
>
> 2. *Select **Get SAML Metadata from IdP** to download the metadata.*

2. Configure Mattermost to verify the signature.

> 1. *In the **Verify Signature** field, select **True**.*
>
> 2. *In the **Service Provider Login URL**, enter* `https//<your-mattermost-url>/login/sso/saml`.

3. Configure Mattermost to sign SAML requests using the Service Provider Private Key.

4. Enable encryption.

> 1. *In the **Enable Encryption** field, select **True**.*
>
> 2. *In the **Service Provider Private Key** field, upload the private key that you generated earlier.*
>
> 3. *In the **Service Provider Public Certificate** field, upload the public key that you generated earlier.*

5. Set attributes for the SAML Assertions, which are used for updating user information in Mattermost.

> *The **Email Atttribute** field and the **Username Attribute** field are required, and should match the values that you entered earlier when you configured the SAML Test Connector on OneLogin.*

6. (Optional) Customize the login button text.

7. Select **Save**.

8. (Optional) If you configured `First Name` Attribute and `Last Name` Attribute, go to **System Console > Site Configuration > Users and Teams**, then

set **Teammate Name Display** to **Show first and last name**. This is recommended for a better user experience.

To confirm that SAML SSO is successfully enabled, switch your system admin account from email to SAML-based authentication from your profile picture via **Profile > Security > Sign-in Method > Switch to SAML SSO**, then log in with your SAML credentials to complete the switch.

We also recommend that you post an announcement to your users detailing how the migration will work.

You can also configure SAML for OneLogin by editing the `config.json` file to enable SAML based on SAML configuration settings. You must restart the Mattermost server for the changes to take effect.

Configure SAML synchronization with AD/LDAP

In addition to configuring SAML sign-in, you can optionally configure synchronizing SAML accounts with AD/LDAP. When configured:

> - *Mattermost queries AD/LDAP for relevant account information and updates SAML accounts based on changes to attributes (first name, last name, and nickname)*
> - *Accounts disabled in AD/LDAP are deactivated in Mattermost, and their active sessions are revoked once Mattermost synchronizes attributes.*

To configure SAML synchronization with AD/LDAP:

1. Go to **System Console > Authentication > SAML 2.0**, then set **Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
2. Go to **System Console > Authentication > AD/LDAP** to open the AD/LDAP wizard, navigate to the **Connection Settings** section, then set **Enable Synchronization with AD/LDAP** to **true**.
3. To ignore guest users when sychronizing, go to **System Console > Authentication > SAML 2.0**, then set **Ignore Guest Users when Synchronizing with AD/LDAP** to **true**.

4. Set the rest of the AD/LDAP settings based on configuration settings documentation to connect Mattermost with your AD/LDAP server.

> - *If you don't want to enable AD/LDAP sign-in, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then set **Enable sign-in with AD/LDAP** to **false**.*

5. To specify how often Mattermost synchronizes SAML user accounts with AD/LDAP, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Sync Performance** section, then set a **Synchronization Interval** in minutes. The default setting is 60 minutes. If you want to synchronize immediately after disabling an account, go to the **Sync History** section and select **AD/LDAP Synchronize Now**.

6. To confirm that Mattermost can successfully connect to your AD/LDAP server, go to **System Console > Authentication > AD/LDAP** wizard, navigate to the **Connection Settings** section, then select **Test Connection**.

Once the synchronization with AD/LDAP is enabled, user attributes are synchronized with AD/LDAP based on their email address. If a user with a given email address doesn't have an AD/LDAP account, they will be deactivated in Mattermost on the next AD/LDAP sync.

To re-activate the account:

1. Add the user to your AD/LDAP server.

2. Purge all caches in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches**.

3. Run AD/LDAP synchronization by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

4. Purge all caches again in Mattermost by going to **System Console > Environment > Web Server**, then select **Purge All Caches** again. This re-activates the account in Mattermost.

> *Note*
>
> *If a user is deactivated from AD/LDAP, they will be deactivated in Mattermost on the next sync. They will be shown as "Deactivated" in the System Console users list, all of their sessions will expire and they won't be able to log back in to Mattermost.*
>
> *If a user is deactivated from SAML, their session won't expire until they're deactivated from AD/LDAP. However, they won't be able to log back in to Mattermost.*

> *Note*
>
> *SAML synchronization with AD/LDAP is designed to pull user attributes such as first name and last name from your AD/LDAP, not to control authentication.*
>
> *In particular, the user filter cannot be used to control who can log in to Mattermost, this should be controlled by your SAML service provider's group permissions.*

See technical description of SAML synchronization with AD/LDAP for more details.

Override SAML data with AD/LDAP data

Alternatively, you can choose to override SAML bind data with AD/LDAP information. For more information on binding a user with the SAML ID Attribute, please refer to this documentation.

This process overrides SAML email address with AD/LDAP email address data or SAML Id Attribute with AD/LDAP Id Attribute if configured. We recommend using this configuration with the SAML ID Attribute to help ensure new users are not created when the email address changes for a user.

To ensure existing user accounts do not get disabled in this process, ensure the SAML IDs match the LDAP IDs by exporting data from both systems and

comparing the ID data. Mapping ID Attributes for both AD/LDAP and SAML within Mattermost to fields that hold the same data will ensure the IDs match as well.

1. Set the SAML `Id Attribute` by going to **System Console > Authentication > SAML 2.0 > Id Attribute**.
2. Set **System Console > Authentication > SAML 2.0 > Override SAML bind data with AD/LDAP information** to **true**.
3. Set **System Console > Authentication > SAML 2.0 > Enable Synchronizing SAML Accounts With AD/LDAP** to **true**.
4. Run AD/LDAP sync by going to **System Console > Authentication > AD/LDAP** wizard, navigating to the **Sync History** section, then select **AD/LDAP Synchronize Now**.

Frequently Asked Questions

What encryption options are supported for SAML?

See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

How to bind authentication to Id attribute instead of email

Alternatively, you can use an `Id` attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

Configuring with an `Id` attribute allows you to reuse an email address for a new user without the old user's information being exposed. For instance, if a user with an email address joe.smith@mattermost.com was once an employee, a new employee named Joe Smith can use the same email. This configuration is also useful when a user's name changes and their email needs to be updated.

This process was designed with backwards compatibility to email binding. Here is the process applied to new account creations and to accounts logging in after the configuration:

- A user authenticated with SAML is bound to the SAML service user using the Id Attribute (as long as it has been configured) or bound by email using the email received from SAML.
- When the user tries to login and the SAML server responds with a valid authentication, then the server uses the "Id" field of the SAML authentication to search the user.
- If a user bound to that ID already exists, it logs in as that user.
- If a user bound to that ID does not exist, it will search base on the email.
- If a user bound to the email exists, it logs in with email and updates the autentication data to the ID, instead of the email.
- If a user bound to the ID or email does not exist, it will create a new Mattermost account bound to the SAML account by ID and will allow the user to log in.

> Note
>
> Existing accounts won't update until they log in to the server.

Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?

Yes. IWA is supported on the browser, with support added to iOS and Android mobile apps in Q2/2019 (mobile apps v1.18 and later).

However, IWA is not supported on the Mattermost Desktop Apps due to a limitation in Electron. As a workaround you may create a browser desktop shortcut for quick access to Mattermost, just like a Desktop App.

Can I provision and deprovision users who log in via SAML?

Yes, but this relies on AD/LDAP to do so. Currently, we do not support SCIM. See “How do I deactivate users?” for more information.

How do I migrate users from one authentication method (e.g. email) to SAML?

See the mmctl user migrate-auth command documentation for details.

How is SAML different from OAuth 2.0 and OpenId Connect?

OAuth 2.0 was primarily intended for delegated authorization, where an app is authorized to access resources, such as Google contact list. It doesn’t deal with authentication.

OpenID Connect is built on top of OAuth 2.0, which supports authentication and thus direct SSO.

SAML is like OpenID Connect, except typically used in enterprise settings. OpenID Connect is more common in consumer websites and web/mobile apps.

Learn more at https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba.

On this page

- Configure SAML with OneLogin
  - Before you begin
  - Create a OneLogin connection app for Mattermost SSO
  - Configure SAML Sign-On for Mattermost
  - Configure SAML synchronization with AD/LDAP
    - Override SAML data with AD/LDAP data
  - Frequently Asked Questions
    - What encryption options are supported for SAML?
    - How to bind authentication to Id attribute instead of email

- Can SAML via Microsoft ADFS be configured with Integrated Windows Authentication (IWA)?
- Can I provision and deprovision users who log in via SAML?
- How do I migrate users from one authentication method (e.g. email) to SAML?
- How is SAML different from OAuth 2.0 and OpenId Connect?

SAML Single Sign-On: technical documentation

Security Assertion Markup Language (SAML) is an open standard that allows identity providers (IdP), like OneLogin, to pass authorization credentials to service providers (SP), like Mattermost.

In simpler terms, it means you can use one set of credentials to log in to many different sites. With a SAML identity provider account, you can log in to Mattermost and other sites securely with the same account.

The main benefit is that it helps administrators centralize user management by controlling which sites users have access to with their SAML identity provider credentials.

Mattermost supports using a single metadata URL to retrieve configuration information for the Identity Provider using your single sign-on URL to generate an IdP metadata URL. The IdP metadata XML file contains the IdP certificate, the entity ID, the redirect URL, and the logout URL.

Using this URL populates the SAML SSO URL and the Identity Provider Issuer URL fields in the configuration process automatically and the Identity Provider Public Certificate is also downloaded from the server and set locally.

This is currently supported for Okta and Microsoft ADFS server 2012 and 2016. OneLogin support will be added in a future release.

For detailed steps, view the Configure SAML with Okta, Configure SAML with Microsoft ADFS for Windows Server 2012, and Configure SAML with Microsoft ADFS using Microsoft Windows Server 2016 documentation. See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

SAML providers

Identity Providers (IdP): An identity provider performs the authentication. When a user clicks to log in, the identity provider confirms who the user is,

and sends data to the service provider with the proper authorization to access the site.

Examples: OneLogin, Okta, Microsoft Active Directory (ADFS) or Azure.

Service Providers (SP): A service provider gets authentication and authorization information from an IdP. Once received, it gives the user access to the system and logs the user in.

Examples: Mattermost, Zendesk, Zoom, Salesforce.

SAML request (AuthNRequest)

When Mattermost initiates an SP-initiated SAML request flow, it generates a **HTTP-Redirect** binding request to the IdP that contains an XML payload as a base64 string


```
bM441nuRIzAjKeMM8RhegMFjZ4L4xPBHhAfHYqgnYDQnSxC+
+Qn5IocWuzuBGz7JQmT9C57nxjxgbFIatiqUCQN17aYrLn/mWE09C5mJMYlcV68ibEkbR/
JKUQ+2u/N+mSD4/C/QvFvuB6BcJaXaz0h7NwGhHROUte6MoGJKMPE=
```


AuthNRequests can also be signed by Mattermost, in which case the XML payload is similar to:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:samlsig="https://
www.w3.org/2000/09/xmldsig#" ID="_u5mpjadp1fdozfih4cj8ap4brh" Version="2.0"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
AssertionConsumerServiceURL="http://localhost:8065/login/sso/saml"
IssueInstant="2019-06-08T16:00:31Z">  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://www.okta.com/
exkoxukx1D8OIfY03356</saml:Issuer>  <samlsig:Signature Id="Signature1">
<samlsig:SignedInfo>  <samlsig:CanonicalizationMethod Algorithm="https://
www.w3.org/2001/10/xml-exc-c14n#"></samlsig:CanonicalizationMethod>
<samlsig:SignatureMethod Algorithm="https://www.w3.org/2000/09/xmldsig#rsa-
sha1"></samlsig:SignatureMethod>  <samlsig:Reference
URI="#_u5mpjadp1fdozfih4cj8ap4brh">  <samlsig:Transforms>
<samlsig:Transform Algorithm="https://www.w3.org/2000/09/xmldsig#enveloped-
signature"></samlsig:Transform>  </samlsig:Transforms>
<samlsig:DigestMethod Algorithm="https://www.w3.org/2000/09/xmldsig#sha1"></
samlsig:DigestMethod>  <samlsig:DigestValue></samlsig:DigestValue>  </
samlsig:Reference>  </samlsig:SignedInfo>  <samlsig:SignatureValue></
samlsig:SignatureValue>  <samlsig:KeyInfo>  <samlsig:X509Data>
<samlsig:X509Certificate>MIIFmzCCA4OgAwIBAgIJAIusvV3gZIwiMA0GCSqGSIb3DQEBCwUAMGIxCzAJBgNVBAYTA
3PB1I0CG3TNzgllgy5CwRGHLKn5/
t8rPsJoWLKOUGenzVdXWuoVi3jyl5FZ1N60CBbXfmSWk20dSIkcYCcYEgs1BTBqKKYFw2dV4M0op
pzNtlq7A0Glpg/gpaR/2TXEPAhOsUfORC2qAdJt9ev0AQjp5V0TkIKMZz8oo33Coi38TG5r/
LG+ihRbpWzO7j9Rc2S5I6bczvG4wOg7nVKG+B5XBvuU9PjSqgxpd/F/
fYf+ggAEru58E+VM4veCRV8vSPbBqDG4FMPV6DiA+tH/
70n6zuPCS3soxX00kjKtP80QD6UgzvM2NN7PHiNlf0Zj6VCU3VdjEnypg7dzlHJuyyaAaTD5nSfk
ecamEoJpq7kaUB7uTmBHELRUhOOy24f54HnP72vnxicZL8cWsOkJwQAqIGzBxQ7J0uX4Os71WrV2
YIur8QVk6KN6MBPxfiCh3xO/
R+cycgx0aMrWZoyzOzP7NCTM5MNE41C48xeGviyCtUID4xiBow+xo6IDUaiCoUVJhz579ore8ic7
0a19DD0qHy4SpBvrUwCO54kvkgn6HjYlLC/k8nFM9F9W9wVAQD/
QwIjd7EtLZLGgbU61Jv3q4kZxxq270hogCRY0lmI3RxkedGHhetF7kaizrikW5zJQEeido/
ir37HhX5AgMBAAGjVDBSMBIGA1UdEwEB/
wQIMAYBAf8CAQAwPAYDVR0RBDUwM4IQbG9ncy5leGFtcGxlLmNvbYITbWV0cmljcy5leGFtcGxlL
mNvbYcEwKgAAYcEfwAAATANBgkqhkiG9w0BAQsFAAOCAgEAH1O91BABHXZjrU7v1OwG+GbU/
4TYZqBXXNxax++OFSRCkEEoNKGg49R6J7lY4lrm12zBlw+oGSyjIOerzi39/
dcxDkKpzyhGvEN4mExbDlybmdCVrHPeWgZl7uwqn4Bj1xiu97M6eMthgxJE7KVNDGDHthGL0/
fTlONIh3qS7Far33hLHJJKy3+lC1MDB8cNltV3mf/ctHCx5Wa0bfZId0MJgd/
seP0WU1HCf3kIxhnhsnOYs32xu7EGiM4/lgnquVd/q/f99ueSaDSHrep373/
w2ce9iF3U0qcLd2iP8ayF/
daGeW1dVPL9R10Oe4BpRjMkjlLwhZdjJeKSg9GBa2GXUEn1Ru9vpSw/C10no3Qx/
6ZHweYbSmJ6hBg4T0nDBp6iVS1eQULNXxDuDWb26U0ESOO5jK8ATywuc45o0bqdvD1XOrGYGfGno
fx7ofRWwKHWfltvxurnbsyo2vH6nM6K41K2DpVdyQOKAGvKe/
oCWfdi+WyBQJGWcIp2OTC1XyWHv7JsY3lo04+islkHEcqJyd8Rf8GWmRHdXz0WzGiZbxWzAuvRRW
nzM31VAws8kQBHTBwIJlJoGX4AXfEvPi+NTxkntf8cQdJucK9ZZbP4ycXHULO4LneyJoJ9Q7nxX1
1xWv7BDWxxclOXy6tyUkg9Fjb7pQ/HCVvGhRzilVU=</samlsig:X509Certificate>  </
```

```
samlsig:X509Data>  </samlsig:KeyInfo>  </samlsig:Signature> </
samlp:AuthnRequest>
```

SAML responses

There are different types of SAML responses sent by the IdP to the SP. The response contains the Assertion with the NameID and attributes of a user.

Below is a table of the different types of responses. Each response type is fully supported except when the SAML assertion is signed while the SAML response itself is not:

Signed SAML Response	Signed SAML Assertion	Encrypted SAML Assertion	Supported by Mattermost
Yes	Yes	Yes	Yes
Yes	Yes	No	Yes
Yes	No	Yes	Yes
Yes	No	No	Yes
No	Yes	Yes	Partially, validation of assertion signature not supported
No	Yes	No	Partially, validation of assertion signature not supported
No	No	Yes	Yes
No	No	No	Yes

For example XML responses for each type, see the [OneLogin SAML response examples (https://www.samltool.com/generic_sso_res.php)](https://www.samltool.com/generic_sso_res.php).

Technical description of SAML synchronization with AD/LDAP

When enabled, SAML synchronization with AD/LDAP occurs in phases:

1. Get all the current LDAP users from the Mattermost database who have `Users.AuthService` set to `ldap`. This is a SQL query issued against the Mattermost database: `SELECT * FROM Users WHERE AuthService = 'ldap'`.

2. Get all the current SAML users from the Mattermost database who have `Users.AuthService` set to `saml`. This is a SQL query issued against the Mattermost database: `SELECT * FROM Users WHERE AuthService = 'saml'`.

3. Get all the current LDAP users from the LDAP server as defined by `LdapSettings.UserFilter`. This is an [LDAP query (https://github.com/mattermost/mattermost/blob/master/server/scripts/ldap-check.sh)](https://github.com/mattermost/mattermost/blob/master/server/scripts/ldap-check.sh) issued against the LDAP server. Users are retrieved in batches as defined by `LdapSettings.MaxPageSize`.

4. Update LDAP attributes. For each existing Mattermost user retrieved in step 1, attempt to find a match against the list of LDAP users from step 3. To find matches, `Users.AuthData` field of the Mattermost user is compared against the `LdapSettings.IdAttribute` LDAP setting.

> - *If any attribute of the user has changed, that attribute is copied from the LDAP server and the user is marked as updated.*
> - *If the corresponding `LdapSettings.IdAttribute` is not found, the user is assumed to be deleted from the LDAP server, and deactivated from Mattermost by setting the `Users.DeleteAt` field to a valid timestamp.*

5. Update SAML attributes. For each existing Mattermost user retrieved in step 2, attempt to find a match against the list of LDAP users from step 3. To find matches, `SamlSettings.Email` is compared against the `LdapSettings.EmailAttribute` LDAP setting.

> - *If any attribute of the user has changed, that attribute is copied from the LDAP server and the user is marked as updated.*
> - *If the corresponding `LdapSettings.EmailAttribute` is not found, the user is assumed to be deleted from the LDAP server, and deactivated from Mattermost by setting the `Users.DeleteAt` field to a valid timestamp.*

Frequently Asked Questions

How can I obtain a SAML metadata XML file consumed by Mattermost?

You can obtain the XML file by calling the Mattermost RESTful API endpoint at `/api/v4/saml/metadata`.

For other useful SAML API calls, see the [API reference (https://api.mattermost.com/#tag/SAML)](https://api.mattermost.com/#tag/SAML).

Can I update user attributes using the API?

No. When Mattermost is configured to use SAML for user authentication, the following user attribute changes can't be made through the API: first name, last name, position, nickname, email, profile image, or username. SAML must be the authoritative source for these user attributes.

Why does the ObjectGUID of a user in Mattermost differ from what I'm seeing in ADFS?

The Active Directory Object-Guid attribute (LDAP display name `objectGUID`) is a 16 byte array which can be displayed in different ways. However, only Microsoft changes the encoding of the ObjectGUID. All the others keep it the same except for the different base (octal, decimal, hex), as follows:

1. The `ldapsearch` linux command displays it as base 64: `Hrz/HqNKnU+lCNTYHx9Ycw==`. This is also the format used in LDIF files.
2. The [LDAP Golang package Mattermost uses (https://github.com/go-ldap/ldap)](https://github.com/go-ldap/ldap) emits the value as hexidecimal (base 16) array, with each byte separated by a backslash: `\1e\bc\ff\1e\a3\4a\9d\4f\a5\08\d4\d8\1f\1f\58\73`

> - *You can remove the backslashes (`1ebcff1ea34a9d4fa508d4d81f1f5873`) and parse it with [Golang like this (https://go.dev/play/p/9b8iDPuz0Nm)](https://go.dev/play/p/9b8iDPuz0Nm). The snippets prints the base 10 representation of each value: `[30 188 255 30 163 74 157 79 165 8 212 216 31 31 88 115]`*

3. Windows Powershell displays the value like this: `1effbc1e-4aa3-4f9d-a508-d4d81f1f5873`

Does `relaystate` need to be passed back to the client from the identity provider?

Yes. If you're integrating Mattermost with a provider that doesn't already do so by default, ensure that `relaystate` is enabled. See the PingIdentity SAML RelayState (https://support.pingidentity.com/s/article/SAML-RelayState-and-PingFederate#:~:text=RelayState%20is%20an%20optional%20parameter,the%20SAML%20 documentation for details.

How can I troubleshoot the SAML logon process?

The most useful tool that we've found for troubleshooting SAML connection issues is the SAML Chrome Panel (https://chromewebstore.google.com/detail/paijfdbeoenhembfhkhllainmocckace?utm_source=item-share-cp) Chrome extension. We're not aware of similar tools for other browsers, but this tool justifies installing Chrome!

On this page

- SAML Single Sign-On: technical documentation
  - SAML providers
  - SAML request (AuthNRequest)
  - SAML responses
  - Technical description of SAML synchronization with AD/LDAP
  - Frequently Asked Questions
    - How can I obtain a SAML metadata XML file consumed by Mattermost?
    - Can I update user attributes using the API?
    - Why does the ObjectGUID of a user in Mattermost differ from what I'm seeing in ADFS?
    - Does relaystate need to be passed back to the client from the identity provider?
    - How can I troubleshoot the SAML logon process?

SAML Single Sign-On

Single sign-on (SSO) is a way for users to log into multiple applications with a single user ID and password without having to re-enter their credentials. The SAML standard allows identity providers to pass credentials to service providers. Mattermost can be configured to act as a SAML 2.0 Service Provider.

Mattermost can be configured to act as a SAML 2.0 Service Provider, and Mattermost officially supports Okta, OneLogin, and Microsoft ADFS as the identity providers (IDPs), please see links below for more details on how to configure SAML with these providers.

In addition to the officially supported identity providers, you can also configure SAML for a custom IdP. For instance, customers have successfully set up miniOrange, Azure AD, DUO, PingFederate, Keycloak, and SimpleSAMLphp as custom IdPs. Because we do not test against these identity providers, it is important that you test new versions of Mattermost in a staging environment to confirm it will work with your identity provider. You can also set up MFA on top of your SAML provider for additional security.

The SAML Single sign-on integration offers the following benefits:

- **Single sign-on.** Users can log in to Mattermost with their SAML credentials.
- **Centralized identity management.** Mattermost accounts automatically pull user attributes from SAML upon login, such as full name, email, and username.
- **Automatic account provisioning.** Mattermost user accounts are automatically created the first time a user signs in with their SAML credentials on the Mattermost server.
- **Sync groups to predefined roles in Mattermost.** Assign team and channel roles to groups via LDAP Group Sync.
- **Compliance alignment with administrator management.** Manage Administrator access to Mattermost in the System Console using SAML attributes.

Important

- SAML Single sign-on itself does not support periodic updates of user attributes nor automatic deprovisioning. However, SAML with AD/LDAP sync can be configured to support these use cases.
- Account creation and updates is influenced by the attributes configured for identification.
  - If the email address is used as the unique identifier and it changes, Mattermost might interpret this as a new user. If an ID attribute is not set, or if it's set to use the email address, a new account may be created in Mattermost when the email address changes.
  - If an ID attribute that doesn't change, such as employeeID, is configured, then Mattermost can use this consistent ID to recognize and match the user account. This allows for updates to other attributes, such as email, without creating a new account. The user details are updated based on the non-changing ID attribute, ensuring continuity and correct user identification without duplicate accounts.
- If configuring Mattermost to use the EU-Login system for authentication, please be aware that their *issuerURI* field is what Mattermost calls "Service Provider Identifier".
- For more information about SAML, see this article from Varonis (https://www.varonis.com/blog/what-is-saml/), and this conceptual example from DUO (https://duo.com/blog/the-beer-drinkers-guide-to-saml).
- See the encryption options documentation for details on what encryption methods Mattermost supports for SAML.

Using SAML attributes to apply roles

You can use attributes to assign roles to specified users on login. To access the SAML attribute settings navigate to **System Console > SAML 2.0**.

Username attribute

(Optional) Enter a SAML assertion filter to use when searching for users.

1. Navigate to **System Console > Authentication > SAML 2.0** (or **System Console > SAML** in versions prior to 5.12).
2. Complete the **Username Attribute** field.
3. Choose **Save**.

When the user accesses the Mattermost URL, they log in with same username and password that they use for organizational logins.

Guest attribute

When enabled, the `guest` attribute in Mattermost identifies external users whose SAML assertion is guest and who are invited to join your Mattermost server. These users will have the guest role applied immediately upon first login instead of the default member user role. This eliminates having to manually assign the role in the System Console.

If a Mattermost guest user has the guest role removed in the SAML system, the synchronization processes will not automatically promote them to a member user role. This is done manually via **System Console > User Management**. If a member user has the `guest` attribute added, the synchronization processes will automatically demote the member user to the guest role.

1. Enable Guest Access via **System Console > SAML 2.0**.
2. Navigate to **System Console > Authentication > SAML 2.0**.
3. Complete the Guest Attribute field.
4. Choose **Save**.

When a guest logs in for the first time they are presented with a default landing page until they are added to channels.

See the Guest accounts documentation for more information about this feature.

Admin attribute

(Optional) The attribute in the SAML Assertion for designating system admins. The users selected by the query will have access to your Mattermost server as system admins. By default, system admins have complete access to the Mattermost System Console.

Existing members that are identified by this attribute will be promoted from member to system admin upon next login. The next login is based upon Session lengths set in **System Console > Session Lengths**. It is recommended that users are manually demoted to members in **System Console > User Management** to ensure access is restricted immediately.

1. Navigate to **System Console > Authentication > SAML 2.0**.
2. Set **Enable Admin Attribute** to `true`.
3. Complete the **Admin Attribute** field.
4. Choose **Save**.

> Note
>
> If the `admin` attribute is set to `false` the member's role as system admin is retained. However if the attribute is removed/changed, system admins that were promoted via the attribute will be demoted to members and will not retain access to the System Console. When this attribute is not in use, system admins can be manually promoted/demoted in **System Console > User Management**.

Configuration assistance

We are open to providing assistance when configuring your custom IdP by answering Mattermost technical configuration questions and working with your IdP provider in support of resolving issues as they relate to Mattermost SAML configuration settings. However, we cannot guarantee your connection will work with Mattermost.

For technical documentation on SAML, see SAML Single Sign-On: technical documentation.

To assist with the process of getting a user file for your custom IdP, see this documentation.

Please note that we may not be able to guarantee that your connection will work with Mattermost, however we will consider improvements to our feature as we are able. You can see more information on getting support here (https://mattermost.com/support/) and submit requests for official support of a particular provider on our feature idea portal (https://portal.productboard.com/mattermost/33-what-matters-to-you).

On this page

- SAML Single Sign-On
  - Using SAML attributes to apply roles
    - Username attribute
    - Guest attribute
    - Admin attribute
  - Configuration assistance

Provisioning workflows

This document provides an overview of user provisioning and deprovisioning workflows in Mattermost.

There are currently 3 recommended user provisioning workflows in Mattermost:

1. **On demand:** If user accounts are not pre-provisioned using one of the methods described below, then a new user account will be provisioned when the user first logs in. When the user logs in, they are asked to select a public team to join (all users must belong to at least one team) and then they are added automatically to the Town Square channel. Mattermost also has a default channel setting that enables system admins to add everyone to additional channels specified by the organization.

2. **Pre-provisioned via bulk import:** Mattermost features a bulk data loading tool that can be used for pre-provisioning new users by adding them to teams and channels before their first login to Mattermost. This tool automates the creation of Teams, Channels, Users, and Posts (with file attachments). It can also be used to migrate users and content from an existing system.

3. **Mattermost API:** The Mattermost RESTful API (https://api.mattermost.com) can be used to pre-provision new user accounts as well as add and remove them from teams and channels. This model is commonly used by enterprises that have central account provisioning applications.

Mattermost user identifier

The user identifier used to uniquely identify users in Mattermost depends on the method of authentication:

- **Email/password**: Email address or username. Created by the user or previously described account pre-provisioning process.

- **AD/LDAP**: Configure a unique and unchanging ID Attribute from the AD/LDAP server to be used in Mattermost. If a user's ID Attribute changes, it will create a new Mattermost account unassociated with their old one.
- **SAML 2.0**: Selected during configuration. You may use an Id Attribute instead of email to bind the user. We recommend choosing an ID that is unique and will not change over time.

User deprovisioning

Users in Mattermost can be deactivated in the following ways:

- **AD/LDAP Synchronization**: AD/LDAP users can be deactivated in Mattermost based on their status in the directory server via synchronization. Learn more in AD/LDAP documentation.
- **System Console**: User management screen in **System Console > Users** allows system admins to deactiveate users with email/password login. See the Deactivate users documentation for details.
- **RESTful API** The Mattermost API can be used to deactivate users. See API documentation to learn more.
- **Command Line Interface**: You can use the Mattermost mmctl user deactivate command to deactivate users.

Once deactivated, users still exist in the Mattermost database and their messages can still be viewed in Mattermost. You can use the mmctl tools to delete a user and all of their content.

From Mattermost v10.10, when a user is deactivated, the account's availability is automatically set to offline.

> Note
>
> As long as the ID attribute selected via AD/LDAP or SAML 2.0 is a unique identifier, the user's email address or name (along with other attributes) can be updated without breaking the user's account.

On this page

- Provisioning workflows
  - Mattermost user identifier
    - User deprovisioning

- Mattermost documentation

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

Backing storage benchmarks

Understanding the performance characteristics of your backing storage is critical for optimizing your Mattermost deployment. These benchmarks offer insights into the relative performance of each storage option, helping you make informed decisions based on your use case and infrastructure needs.

This page provides detailed write and read benchmark results for supported storage options including local file system (EBS, gp3), network file system (EFS), and object storage (S3).

Write operations

File Size	Local (EBS)	EFS	S3
100KB	0.57 ms (±0.26 ms, p50: 0.63 ms)	34.47 ms (±6.33 ms, p50: 32.57 ms)	181.59 ms (±39.05 ms, p50: 174.83 ms)
1MB	2.08 ms (±0.92 ms, p50: 2.28 ms)	57.43 ms (±7.32 ms, p50: 58.16 ms)	277.13 ms (±49.95 ms, p50: 279.36 ms)
10MB	18.15 ms (±5.31 ms, p50: 18.23 ms)	180.58 ms (±97.95 ms, p50: 150.35 ms)	469.29 ms (±166.32 ms, p50: 462.12 ms)
100MB	585.14 ms (±274.38 ms, p50: 795.78 ms)	390.87 ms (±166.32 ms, p50: 339.20 ms)	1.09 s (±0.04 s, p50: 1.09 s)
1GB	7.42 s (±2.13 s, p50: 7.99 s)	3.05 s (±1.09 s, p50: 2.93 s)	10.65 s (±0.82 s, p50: 10.44 s)
10GB	70.66 s (±7.05 s, p50: 80.50 s)	26.54 s (±2.47 s, p50: 26.86 s)	109.05 s (±12.47 s, p50: 105.13 s)

Note

Write Performance: Local EBS storage is fastest for small files (100KB - 10MB), while EFS performs better for larger files (100MB - 10GB). S3 is consistently the slowest for writes across all file sizes.

Read operations

File Size	Local (EBS)	EFS	S3
100KB	0.05 ms (±0.02 ms, p50: 0.04 ms)	0.66 ms (±0.17 ms, p50: 0.60 ms)	28.55 ms (±4.12 ms, p50: 27.63 ms)
1MB	0.18 ms (±0.06 ms, p50: 0.18 ms)	0.82 ms (±0.19 ms, p50: 0.72 ms)	27.03 ms (±4.28 ms, p50: 27.54 ms)
10MB	1.27 ms (±0.27 ms, p50: 1.16 ms)	7.83 ms (±18.12 ms, p50: 1.68 ms)	108.24 ms (±0.56 ms, p50: 108.18 ms)
100MB	16.50 ms (±0.95 ms, p50: 16.16 ms)	73.53 ms (±182.32 ms, p50: 16.91 ms)	1.05 s (±0.00 s, p50: 1.05 s)
1GB	167.89 ms (±2.19 ms, p50: 167.72 ms)	773.41 ms (±1768.89 ms p50: 168.17 ms)	10.49 s (±0.02 s, p50: 10.49 s)
10GB	1.57 s (±0.03 s, p50: 1.56 s)	1.56 s (±0.02 s, p50: 1.55 s)	150.66 s (±33.47 s, p50: 146.51 s)

Note

- **Read Performance**: Local and EFS have comparable read performance for very large files (10GB), but local storage outperforms for smaller files. S3 read performance is significantly slower, especially for large files (150.66s versus ~1.56s for 10GB files).
- **Consistency**: Local storage shows the most consistent performance (lower standard deviations) for small files, while EFS shows more consistent performance for large files. S3 generally has higher variability across all metrics.
- **Median vs Average**: The median (p50) values generally align with the averages, but in some cases (particularly EFS read operations), the median reveals that outliers significantly affect the average. For example, EFS read performance for 100MB and 1GB files shows much better median values than averages.

Write Operations Performance Comparison
Local (EBS)
EFS
S3
Time (ms)
File Size
100KB
1MB
10MB
100MB
1GB
10GB
0.57ms
34.5ms
181.6ms
2.1ms
57.4ms
277.1ms
18.1ms
180.6ms
469.3ms
585.1ms
390.9ms
1.1s
7.4s
3.0s
10.7s
70.7s
26.5s
109.9s

Read Operations Performance Comparison
Local (EBS)
EFS
S3
Time (ms)
File Size
100KB
1MB
10MB
100MB
1GB
10GB
0.05ms
0.66ms
28.6ms
0.18ms
0.82ms
27.0ms
1.3ms
7.8ms
108.2ms
16.5ms
73.5ms
1.1s
167.9ms
773.4ms
10.5s
1.6s
1.6s
150.7s

Testing notes

- For S3 tests, Amazon S3 exported upload part size was set to the default value (100MB).
- Local EBS storage is the stock gp3 (3000 IOPS) provided by EC2 instances.
- Both EBS and EFS solutions tested are considered `local` storage options from the application's perspective, where the file storage system is set to `local` in both cases. EFS is essentially AWS's managed NFS, which enables it to serve as a potential alternative to S3 by allowing multiple Mattermost nodes in a high-availability (HA) deployment to share a common file system. In such HA scenarios, the standard local file storage (e.g., an EBS volume attached to a single instance) is not suitable, as it can't be shared across multiple nodes. EFS is a good alternative in this case, but EFS is not a block storage solution like EBS.

Supported storage options

Local

The local file system of the Mattermost server. If running in a cloud environment like AWS, this often utilizes Elastic Block Storage (EBS), which provides block-level storage attached directly to the instances. This setup typically offers high performance with low latency since the storage is local to the server. This option is often the fastest for small files due to low latency.

EFS

Amazon Elastic File System, a managed file storage solution that supports shared access across multiple instances. EFS supports shared access across multiple instances but adds network overhead, which can impact performance compared to local storage options. EFS is generally slower than local storage but can be beneficial for certain use cases requiring shared access.

S3

Amazon Simple Storage Service, an object storage solution that provides high durability and scalability. While S3 is great for storing large amounts of data

reliably, it introduces higher latency and slower performance due to network-based access and its nature as object storage rather than block or file storage.

On this page

- Backing storage benchmarks
  - Write operations
  - Read operations
  - Testing notes
  - Supported storage options
    - Local
    - EFS
    - S3

Collect performance metrics

System admins can collect and store the same performance monitoring metrics as Prometheus, without having to deploy these third-party tools. Data is collected every minute and is stored in a path you configure. The data is synchronized to either a cloud-based or local file store every hour, and retained for 15 days.

Download and share the collected data with Mattermost to understand application performance, troubleshoot system stability and performance, as well as inform route cause analysis.

> Tip
>
> Already have Prometheus and Grafana deployed? You can use these tools to monitor performance of your Mattermost deployment.

Mattermost configuration

> Note
>
> For Mattermost Cloud deployments, no setup is required. See the usage section below for details on collecting performance metrics.

For a self-hosted Mattermost deployment, a Mattermost system admin must perform the following steps in Mattermost.

1. Log in to your Mattermost workspace as a system admin.
2. From Mattermost v10.1, you can install the Metrics plugin from the in-product Mattermost Marketplace by selecting the **Product**

icon and selecting **App Marketplace**. Search for **Metrics** and select **Install**.

3. Go to **System Console > Plugins > Plugin Management**. In the **Installed Plugins** section, scroll to **Mattermost Metrics Plugin**, and select **Enable Plugin**.

4. Specify the path of the time-series database, and select **Save**.

5. Go to **System Console > Environment > Performance Monitoring**, and set **Enable Performance Monitoring** to **true**. Select **Save**.

> Note
>
> For Mattermost deployments prior to v10.1, you must download the latest version of the plugin binary release (https://github.com/mattermost/mattermost-plugin-metrics/releases), compatible with Mattermost v8.0.1 and later. Go to **System Console > Plugins > Plugin Management > Upload Plugin**, and upload the plugin binary you downloaded.

Upgrade

We recommend upgrading this feature as new versions are released. Generally, updates are seamless and don't interrupt the user experience in Mattermost. Visit the Releases page (https://github.com/mattermost/mattermost-plugin-metrics/releases) for the latest release, available releases, and compatibiilty considerations.

Usage

You need to be a Mattermost system admin to collect performance metrics. Select **Create Dump** to generate dump files.

To use the generated dump file, you can simply clone the Dockprom (https://github.com/stefanprodan/dockprom) repository. Change the Prometheus data volume to point to the dump that you just downloaded. The downloaded file is compressed, so to be able to use it, you need to decompress it first.

The volume configuration for Prometheus should look like the code below in the `docker-compose.yml` file:

```
volumes:  - ./prometheus:/etc/prometheus  - /Path/To/Dump/Directory:/
prometheus/data
```

Once you set this up, run `docker-compose` as described in [Dockprom Repository (https://github.com/stefanprodan/dockprom?tab=readme-ov-file#install)](https://github.com/stefanprodan/dockprom?tab=readme-ov-file#install).

You can also use our [Mattermost Performance Monitoring v2 (https://grafana.com/grafana/dashend-user-guide/boards/15582-mattermost-performance-monitoring-v2/)](https://grafana.com/grafana/dashend-user-guide/boards/15582-mattermost-performance-monitoring-v2/) dashboard by simply importing it into Grafana.

1. Open Grafana (`<localhost>:3000` by default) and then log into it.
2. Once you log in, go to the **Plus**

   +

   icon on the left sidebar, and then select **Import**.
3. Enter the dashboard ID (`15582`) in the **Grafana.com Dashboard** field, and then select **Load** to fetch the dashboard.

What's collected?

Mattermost provides custom metrics and standard Go metrics that can be used to monitor your system's performance. Additionally Enterprise customers can use the Metrics plugin to collect host/system metrics from [node exporter (https://github.com/prometheus/node_exporter)](https://github.com/prometheus/node_exporter) targets to monitor network-related panels for Mattermost Calls.

On this page

- Collect performance metrics
  - Mattermost configuration
  - Upgrade
  - Usage
  - What's collected?

Set server connection details

1. (Optional) Enter **Server Username** used to access the enterprise search server.
2. (Optional) Enter **Server Password** associated with the username.
3. Set **Enable Cluster Sniffing** (Optional). Sniffing finds and connects to all data nodes in your cluster automatically.
4. Optional CA and client certificate configuration settings are available for use with basic authentication credentials or to replace them. See the Enterprise search configuration settings documentation for details.
5. Select **Test Connection** and then select **Save**. If the server connection is unsuccessful you won't be able to save the configuration or enable searching with Elasticsearch or AWS OpenSearch.

Build the post index of existing messages

Select **Index Now**. This process can take up to a few hours depending on the size of the post database and number of messages. The progress percentage can be seen as the index is created. To avoid downtime, set **Enable Elasticsearch for search queries** to `false` so that database search is available during the indexing process.

Enable enterprise search

Ensure bulk indexing is complete before enabling enterprise search, otherwise search results will be incomplete.

Set **Enable Elasticsearch for search queries** to `true`, and setting **Enable Elasticsearch for autocomplete** to `true`. Save your configuration updates and restart the Mattermost server.

> Note
>
> For high post volume deployments, we strongly encourage you to read and properly configure the Mattermost LiveIndexingBatchSize configuration setting.

Once the configuration is saved, new posts made to the database are automatically indexed on the Elasticsearch or AWS OpenSearch server.

Enterprise search limitations

1. Elasticsearch and AWS OpenSearch uses a standard selection of "stop words" to keep search results relevant. Results for the following words will not be returned: "a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", and "with".
2. Searching stop words in quotes returns more results than just the searched terms (ticket (https://mattermost.atlassian.net/browse/MM-7216)).
3. Search results are limited to a user's team and channel membership. This is enforced by the Mattermost server. The entities are indexed in Elasticsearch or AWS OpenSearch in a way that allows Mattermost to filter them when querying, so the Mattermost server narrows down the results on every Elasticsearch or AWS OpenSearch request applying those filters.

Deploy Prometheus and Grafana for performance monitoring

Performance monitoring support enables admins to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard.

Admins can collect and store various data points from the Mattermost application in an OpenMetrics (https://openmetrics.io) format by deploying Prometheus and Grafana.

> Tip
>
> Don't want to deploy Prometheus and Grafana? You can also collect performance metrics using the Mattermost Metrics plugin.

Install Prometheus

Important

While Prometheus and Grafana may be installed on the same server as Mattermost, we recommend installing these integrations on separate servers, and configure Prometheus to pull all metrics from Mattermost and other connected servers.

1. Download a precompiled binary for Prometheus. Binaries are provided for many popular distributions, including Darwin, Linux, and Windows. For installation instructions, see the Prometheus install guides (https://prometheus.io/docs/prometheus/latest/getting_started/).

2. The following settings are recommended in the Prometheus configuration file named `prometheus.yml`:

```
# my global config global:  scrape_interval: 60s # By default, scrape
targets every 15 seconds.  evaluation_interval: 60s # By default, scrape
targets every 15 seconds.  # scrape_timeout is set to the global default
(10s).  # Attach these labels to any time series or alerts when
communicating with  # external systems (federation, remote storage,
Alertmanager).  external_labels:  monitor: 'mattermost-monitor' # Load
rules once and periodically evaluate them according to the global
'evaluation_interval'. rule_files:  # - "first.rules"  # -
"second.rules" # A scrape configuration containing exactly one endpoint
to scrape: # Here it's Prometheus itself. scrape_configs:
# The job name is added as a label `job=<job_name>` to any timeseries
scraped from this config.  - job_name: 'prometheus'  # Override the
global default and scrape targets from this job every five seconds.  #
scrape_interval: 5s  # metrics_path defaults to '/metrics'  # scheme
defaults to 'http'.  static_configs:  - targets: ["<hostname1>:<port>",
"<hostname2>:<port>"]
```

3. Replace the `<hostname1>:<port>` parameter with your Mattermost host IP address and port to scrape the data. It connects to `/metrics` using HTTP.

4. In the Mattermost System Console, go to **Environment > Performance Monitoring** to set **Enable Performance Monitoring** to **true**, then specify the **Listen Address** and select **Save**. See our Configuration Settings documentation for details.

5. To test that the server is running, go to `<ip>:<port>/metrics`.

> Note
>
> A Mattermost Enterprise license is required to connect to `/metrics` using HTTP.

6. Finally, run `vi prometheus.yml` to finish configuring Prometheus. For starting the Prometheus service, read the comprehensive guides provided by Prometheus (https://prometheus.io/docs/prometheus/latest/getting_started/).
7. Once the service has started, you can access the data in `<localhost>:<port>/graph`. While you can use the Prometheus service to create graphs, we'll focus on creating metric and analytics dashboards in Grafana.

> Tip
>
> For troubleshooting advice, check the Prometheus FAQ page (https://prometheus.io/docs/introduction/faq/).

Install Grafana

Important

While Prometheus and Grafana may be installed on the same server as Mattermost, we recommend installing these integrations on separate servers, and configure Prometheus to pull all metrics from Mattermost and other connected servers.

1. Download a precompiled binary for Grafana on Ubuntu or Debian. Binaries are also available for other distributions, including Redhat, Windows and Mac. For install instructions, see Grafana install guides (https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/)

2. The Grafana package is installed as a service, so it is easy to start the server. See their [install guides (https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/)](https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/) to learn more.

3. The default HTTP port is `3000` and default username and password are `admin`.

4. Add a Mattermost data source with the following settings as defined in the screenshot below

> Tip
>
> - For troubleshooting advice, check the Grafana Troubleshooting page (https://grafana.com/docs/grafana/latest/troubleshooting/).
> - For user guides and tutorials, check the Grafana documentation to learn more.

Getting started

To help you get started, you can download three sample dashboards shared in Grafana:

> Tip
>
> See this Grafana guide (https://grafana.com/docs/grafana/v7.5/dashend-user-guide/boards/export-import/) to learn how to import Grafana dashboards either from the UI or from the HTTP API.

- Mattermost Performance Monitoring v2 (https://grafana.com/grafana/dashend-user-guide/boards/15582-mattermost-performance-monitoring-v2/), which contains detailed charts for performance monitoring including application, cluster, job server, and system metrics.
- Mattermost Notification Health Monitoring (https://grafana.com/grafana/dashend-user-guide/boards/21305-mattermost-notification-health), which can be used to track different types of notifications sent from Mattermost. Accessing and enabling Mattermost Notification Health Monitoring requires the feature flag `NotificationMonitoring` to be set to `true`. System admins can disable notification monitoring data collection through the System Console.
- Mattermost Web App Performance Metrics (https://grafana.com/grafana/dashend-user-guide/boards/21460-web-app-metrics/), which contains detailed metrics for client-side performance, including web vitals and Mattermost-specifc metrics.

- Mattermost Desktop App Performance Metrics (https://grafana.com/grafana/dashend-user-guide/boards/22736-desktop-app-metrics/), which contains detailed metrics for client-side desktop performance, including CPU and memory usage metrics.
- Mattermost Mobile App Performance Metrics (https://grafana.com/grafana/dashend-user-guide/boards/21695-mobile-performance-metrics/), which contains detailed metrics for client-side mobile performance, including web vitals and Mattermost-specifc metrics.
- Mattermost Threaded Discussion Metrics (https://grafana.com/grafana/dashend-user-guide/boards/15581-collapsed-reply-threads-performance/), which contains detailed metrics on the queries involved in our threaded discussions feature.
- Mattermost Performance KPI Metrics (https://grafana.com/grafana/dashend-user-guide/boards/2539-mattermost-performance-kpi-metrics/), which contains key metrics for monitoring performance and system health.
- Mattermost Performance Monitoring (Bonus Metrics) (https://grafana.com/grafana/dashend-user-guide/boards/2545-mattermost-performance-monitoring-bonus-metrics/), which contains additional metrics such as emails sent or files uploaded, which may be important to monitor in some deployments.

What's collected?

Mattermost provides custom metrics and standard Go metrics that can be used to monitor your system's performance.

On this page

- Deploy Prometheus and Grafana for performance monitoring
  - Install Prometheus
  - Install Grafana
  - Getting started
  - What's collected?

Elasticsearch server setup

Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. The indexing process can be managed from the System Console after setting up and connecting an Elasticsearch server. The post index is stored on the Elasticsearch server and updated constantly after new posts are made. In order to index existing posts, a bulk index of the entire post database must be generated.

Deploying Elasticsearch includes the following two steps: setting up Elasticsearch, and configuring Mattermost.

Set up Elasticsearch

We highly recommend that you set up Elasticsearch server on a dedicated machine separate from the Mattermost Server.

1. Download and install the latest release of [Elasticsearch v8 (https://www.elastic.co/guide/en/elasticsearch/reference/8.15/install-elasticsearch.html)](https://www.elastic.co/guide/en/elasticsearch/reference/8.15/install-elasticsearch.html), or [Elasticsearch v7.17+ (https://www.elastic.co/guide/en/elasticsearch/reference/7.17/install-elasticsearch.html)](https://www.elastic.co/guide/en/elasticsearch/reference/7.17/install-elasticsearch.html). See the Elasticsearch documentation for installation details.

2. Set up Elasticsearch with `systemd` by running the following commands:

```
sudo /bin/systemctl daemon-reload sudo /bin/systemctl enable
elasticsearch.service sudo systemctl start elasticsearch.service
```

3. Confirm Elasticsearch is working on the server by running the following command:

```
curl localhost:9200
```

4. Get your network interface name by running the following command:

```
ip addr
```

5. Edit the Elasticsearch configuration file in `vi` by running the following command:

```
vi /etc/elasticsearch/elasticsearch.yml
```

6. In this file, replace the `network.host` value of `_eth0_` with your network interface name, and save your changes.

7. When using Elasticsearch v8, ensure you set `action.destructive_requires_name` to `false` in `elasticsearch.yml` to allow for wildcard operations to work.

8. Restart Elasticsearch by running the following commands:

```
sudo systemctl stop elasticsearch sudo systemctl start elasticsearch
```

9. Confirm the ports are listenings by running the following command:

```
netstat -plnt
```

You should see the following ports, including the ones listening on ports 9200 and 9300. Confirm these are listening on your server's IP address.

10. Create an Elasticsearch directory and give it the proper permissions.

11. Install the [icu-analyzer plugin (https://www.elastic.co/guide/en/elasticsearch/plugins/current/analysis-icu.html)](https://www.elastic.co/guide/en/elasticsearch/plugins/current/analysis-icu.html) to the `/usr/share/elasticsearch/plugins` directory by running the following command:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install analysis-icu
```

12. Test the connection from Mattermost to Elasticsearch by running the following command:

```
curl 172.31.80.220:9200
```

Configure Mattermost

Follow these steps to configure Mattermost to use your Elasticsearch server and to generate the post index:

1. Go to **System Console > Environment > Elasticsearch**.
2. Set **Enable Elasticsearch Indexing** to `true` to enable the other the settings on the page. Once the configuration is saved, new posts made to the database are automatically indexed on the Elasticsearch server.
3. Ensure **Backend type** is set to `elasticsearch`.

Set server connection details

1. (Optional) Enter **Server Username** used to access the enterprise search server.
2. (Optional) Enter **Server Password** associated with the username.
3. Set **Enable Cluster Sniffing** (Optional). Sniffing finds and connects to all data nodes in your cluster automatically.
4. Optional CA and client certificate configuration settings are available for use with basic authentication credentials or to replace them. See the Enterprise search configuration settings documentation for details.
5. Select **Test Connection** and then select **Save**. If the server connection is unsuccessful you won't be able to save the configuration or enable searching with Elasticsearch or AWS OpenSearch.

Build the post index of existing messages

Select **Index Now**. This process can take up to a few hours depending on the size of the post database and number of messages. The progress percentage can be seen as the index is created. To avoid downtime, set **Enable Elasticsearch for search queries** to `false` so that database search is available during the indexing process.

Enable enterprise search

Ensure bulk indexing is complete before enabling enterprise search, otherwise search results will be incomplete.

Set **Enable Elasticsearch for search queries** to `true`, and setting **Enable Elasticsearch for autocomplete** to `true`. Save your configuration updates and restart the Mattermost server.

> Note
>
> For high post volume deployments, we strongly encourage you to read and properly configure the Mattermost LiveIndexingBatchSize configuration setting.

Once the configuration is saved, new posts made to the database are automatically indexed on the Elasticsearch or AWS OpenSearch server.

Enterprise search limitations

1. Elasticsearch and AWS OpenSearch uses a standard selection of "stop words" to keep search results relevant. Results for the following words will not be returned: "a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", and "with".

2. Searching stop words in quotes returns more results than just the searched terms (ticket (https://mattermost.atlassian.net/browse/MM-7216)).

3. Search results are limited to a user's team and channel membership. This is enforced by the Mattermost server. The entities are indexed in Elasticsearch or AWS OpenSearch in a way that allows Mattermost to filter them when querying, so the Mattermost server narrows down the results on every Elasticsearch or AWS OpenSearch request applying those filters.

On this page

- Elasticsearch server setup
  - Set up Elasticsearch

- Configure Mattermost
  - Set server connection details
  - Build the post index of existing messages
  - Enable enterprise search
- Enterprise search limitations

Ensuring Releases Perform at Scale

To ensure each release of Mattermost upholds our high standards for performance at scale, the Mattermost Engineering team performs thorough load testing, develops features with scale in mind, and follows strict guidelines for database schema migrations.

Monthly release load tests

Each month, before being approved for distribution, a release candidate of Mattermost is load tested via a comprehensive and mature set of load test tooling (https://github.com/mattermost/mattermost-load-test-ng) using simulated data that matches real-world, high-scale usage patterns.

Multiple tests covering different configurations are run against highly-available deployments of Mattermost with thousands of users and millions of posts. Both PostgreSQL and MySQL are tested, although MySQL tests will be dropped when MySQL goes out of support with Mattermost v11.

The load tests generate a report detailing average API request times, database I/O, memory usage, concurrency, requests per second, and more. This performance report on the release candidate is then compared to the report of the latest previous stable version. Any deviations are investigated and remedied before the release candidate is promoted to the final release.

Each report, along with analysis, is posted into the public Developers: Performance (https://community.mattermost.com/core/channels/developers-performance) channel.

Developing scalable features and systems

Scale is a major consideration during the development of new features and systems, and it's included in technical specifications from the beginning of the software design process.

As a part of implementation, the load test agent is updated to include coverage for the feature or system. A load test is then run with results being compared to baselines.

The code changes are then reviewed by at least two developers and a SDET/QA analyst before being merged.

Once merged, a new build is created and deployed the next day to the Mattermost Community Server (https://community.mattermost.com) where any impacts on performance are monitored for 3-4 weeks before the changes are included in a release candidate.

Database schema changes

Database schema changes are kept to a minimum to reduce risk on upgrades and impact on performance. When a change in schema is required, Mattermost follows strict migration guidelines (https://developers.mattermost.com/contribute/more-info/server/schema-migration-guide/) that minimize risk and prevents performance impact during and after migrations.

If a more involved migration is required, detailed analysis is performed and published with guidance. An example analysis is the Mattermost v6.0 schema migration (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055).

Additionally, all database schema changes are load tested as a part of our monthly release process.

Monitoring post-release

After a new version of Mattermost is released, it is rolled out to Mattermost Cloud customers over a multi-week period. During roll out, performance and error rate metrics are monitored for any unexpected changes. If any user-impacting changes are observed, the release is reverted and the deviations are investigated with any fixes being delivered as a part of patch release.

In addition, Mattermost channels, user forums, and support tickets are closely monitored for reports of any issues. Any reports are investigated and resolved appropriately.

On this page

- Ensuring Releases Perform at Scale
  - Monthly release load tests
  - Developing scalable features and systems
  - Database schema changes
  - Monitoring post-release

Enterprise search

Mattermost database search starts to show performance degradation at around 2 million posts, on a server with 32 GB RAM and 4 CPUs. If you anticipate your Mattermost server reaching more than 2.5 million posts, we recommend enabling Elasticsearch or AWS OpenSearch Service for optimum search performance **before** reaching 3 million posts. Both tools are highly capable and can handle enterprise-scale workloads. The choice between them depends on the following factors:

- **Licensing and cost**: AWS OpenSearch may be preferable for organizations avoiding proprietary licensing or opting for cost-effective solutions.
- **Feature requirements**: Elasticsearch's proprietary features (e.g., advanced analytics, security suites) may be preferred by organizations needing powerful out-of-the-box functionality.
- **Infrastructure alignment**: AWS OpenSearch aligns well with AWS-centric environments, while Elasticsearch offers broader integrations.

Elasticsearch

Elasticsearch is a well-established and widely used search engine with a large ecosystem and community support that provides enterprise-scale deployments with optimized search performance, dedicated indexing, and usage resourcing via cluster support for fast, predicable search results.

Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Learn more about setting up and configuring Mattermost for an Elasticsearch server.

AWS OpenSearch Service

AWS OpenSearch Service is the official path forward from Elasticsearch v7.10.x for AWS customers. It's a fully managed service that makes it easy to

deploy, operate, and scale OpenSearch clusters in the AWS Cloud to provide a simple and cost-effective way to search, analyze, and visualize data in real time.

The AWS OpenSearch Service is built on the open-source OpenSearch project, which is a community-driven fork of Elasticsearch. Learn more about setting up and configuring Mattermost for an OpenSearch server.

Supported paths

Review the following support paths for enterprise search based on the version you're using:

Elasticsearch v8

Elasticsearch v8 (https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html) is supported from Mattermost v9.11. While Mattermost supports Elasticsearch v7.17+, we recommend upgrading your Elasticsearch v7 instance to v8.x. See the Elasticsearch upgrade (https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-upgrade.html) documentation for upgrade details, and see the Elasticsearch setup documentation for details on configuring your Mattermost deployment to use Elasticsearch.

AWS OpenSearch Service

AWS OpenSearch Service is the official path forward from Elasticsearch v7.10.x for AWS customers to provide a simple and cost-effective way to search, analyze, and visual data in real time. It's essentially a continuation of Elasticsearch v7.10.x but maintained as open source by AWS. It provides long-term support, active development, and compatibility with AWS clients, libraries, and managed services.

See the **AWS Elasticsearch v7.10.x** tab on this page for details on upgrading to AWS OpenSearch, and see the AWS OpenSearch setup documentation for details on configuring your Mattermost deployment to use AWS OpenSearch.

AWS Elasticsearch v7.10.x

If you're using Elasticsearch v7.10.x under AWS's managed services, you can't use newer Elasticsearch clients like the v8 client without changing backend

infrastructure. If you're using AWS Elasticsearch v7.10.x, you must [upgrade to AWS OpenSearch (https://docs.aws.amazon.com/opensearch-service/latest/developerguide/version-migration.html)](https://docs.aws.amazon.com/opensearch-service/latest/developerguide/version-migration.html) for future compatibility.

The migration path from Elasticsearch v7.10.x to OpenSearch has been designed to be straightforward, minimizing effort:

1. Disable "compatibility mode" in OpenSearch.
2. Upgrade Mattermost server.
3. Update the Mattermost `ElasticsearchSettings.Backend` configuration setting value from `elasticsearch` to `` `opensearch` `` manually or using mmctl. This value cannot be changed using the System Console. See the Mattermost search backend type configuration setting documentation for additional details.
4. Restart the Mattermost server.

Frequently asked questions (FAQ)

Do I need to use Elasticsearch or AWS OpenSearch?

No. Enterprise search engines are designed for large Enterprise deployments to run highly efficient database searches in a cluster environment. The default Mattermost database search starts to show performance degradation at around 2.5 million posts, depending on the specifications for the database server. If you expect your Mattermost server to have more than 2.5 million posts, we recommend using Elasticsearch or AWS OpenSearch for optimum search performance.

Should I install Elasticsearch or OpenSearch on the same machine as Mattermost Server?

No. We strongly recommend that you install Elasticsearch or AWS OpenSearch on a dedicated machine that's separate from the Mattermost server.

What types of indexes are created?

Mattermost creates three types of indexes: users, channels, and posts. Users and channels have one index each. Posts are aggregated by date, into multiple indexes.

Can I pause an search indexing job?

Yes. From Mattermost v6.7, the search indexing job is resumable. Stopping a server while the search indexing job is running puts the job in pending status. The job resumes when the server restarts. System admins can cancel an indexing job through the System Console.

Can an index rollover policy be defined?

The AggregatePostsAfterDays configuration setting defines a cutoff value. All posts preceding this value are reindexed and aggregated into new and bigger indexes. The default setting is 365 days.

Are there any new search features offered with Elasticsearch?

The current implementation of Elasticsearch matches the search features currently available with database search. The Mattermost team is working on extending the Elasticsearch feature set with file name and content search, date filters, and operators and modifiers.

Are my files stored in Elasticsearch or OpenSearch?

No, files and attachments are not stored.

How do I monitor system health of an Elasticsearch server?

You can use this Prometheus exporter to monitor various metrics (https://github.com/justwatchcom/elasticsearch_exporter#metrics) about Elasticsearch: justwatchcom/elasticsearch_exporter (https://github.com/justwatchcom/elasticsearch_exporter).

You can also refer to this article about Elasticsearch performance monitoring (https://www.datadoghq.com/blog/monitor-elasticsearch-performance-metrics/#key-elasticsearch-performance-metrics-to-monitor). It's not written specifically for Prometheus, which Mattermost's performance monitoring system uses, but has several tips and best practices.

What form of data is sent to Elasticsearch or OpenSearch?

Mattermost communicates with Elasticsearch or OpenSearch through its REST API using JSON messages for indexing and querying entities.

How much data is sent to Elasticsearch or OpenSearch and when?

Every time a message is published, a channel is created, or a user changes, (either because their properties change e.g.: change of the first name or because they join/leave a channel), the data associated with that event is sent to Elasticsearch or OpenSearch.

If search via Elasticsearch or OpenSearch is enabled, every search will generate a query. If autocompletion is enabled, every user or channel autocompletion associated with writing a message or user search will generate a query.

How do I know if a search job fails?

Mattermost provides the status of each Elasticsearch or OpenSearch indexing job in **System Console > Environment > Elasticsearch**. Here you can see if the job succeeded or failed, including the details of the error.

Failures are returned in the server logs. The error log begins with the string `Failed job` and includes a job_id key/value pair. Search job failures are identified with worker name `EnterpriseElasticsearchAggregator` and `EnterpriseElasticsearchIndexer`. You can optionally create a script that programmatically queries for such failures and notifies the appropriate system.

My search indexes won't complete, what should I do?

If you have an search indexing job that's paused, it's likely your Elasticsearch or OpenSearch server has restarted. If you restart that server, you must also restart Mattermost to ensure jobs are completed. If restarting the Mattermost server does not resolve the issue, commercial customers can contact Mattermost Support (https://mattermost.com/support/) for assistance.

Do I need to purge first then bulk index each time?

Yes. Every time you choose to remove old data (purge), you must follow it up with a bulk index operation to repopulate them with new data. When you purge (delete) indices, you are removing the existing data from Elasticsearch. After purging, if you want to repopulate the indices with fresh data, you perform a bulk index operation to rebuild the indices efficiently. The bulk API allows for indexing, deleting, or updating multiple documents in a single request, which is faster and more efficient than processing documents one by one.

Required Permissions For Mattermost Service Account

In "least privilege" environments you may need to further constrain the service account permissions to limit the access your Elasticsearch or AWS OpenSearch service account has.

The following JSON provides an example of a "least privilege" permission set that allows Mattermost to operate correctly with Elasticsearch or AWS OpenSearch:

```
{ "cluster_permissions": [ "cluster:monitor/*", "indices:admin/
template/put", "indices:data/write/bulk" ], "index_permissions": [
{ "index_patterns": [ "\<IndexPrefix\>*" ], "allowed_actions": [
"indices:admin/get", "indices:admin/create", "indices:admin/delete",
"indices:admin/mapping/put", "indices:admin/mappings/fields/get*",
"indices:data/read*", "indices:data/write*" ] } ] }
```

A simpler, more flexible, and resilient variant of the above would be:

```
{ "cluster_permissions": [ "cluster:monitor/*", "indices:admin/template/put", "indices:data/write/bulk" ], "index_permissions": [ { "index_patterns": [ "\<IndexPrefix\>*" ], "allowed_actions": [ "indices:*" ] } ] }
```

On this page

- Enterprise search
- Enterprise search
  - Elasticsearch
  - AWS OpenSearch Service
  - Supported paths
    - Elasticsearch v8
    - AWS OpenSearch Service
    - AWS Elasticsearch v7.10.x
  - Frequently asked questions (FAQ)
    - Do I need to use Elasticsearch or AWS OpenSearch?
    - Should I install Elasticsearch or OpenSearch on the same machine as Mattermost Server?
    - What types of indexes are created?
    - Can I pause an search indexing job?
    - Can an index rollover policy be defined?
    - Are there any new search features offered with Elasticsearch?
    - Are my files stored in Elasticsearch or OpenSearch?
    - How do I monitor system health of an Elasticsearch server?
    - What form of data is sent to Elasticsearch or OpenSearch?
    - How much data is sent to Elasticsearch or OpenSearch and when?
    - How do I know if a search job fails?
    - My search indexes won’t complete, what should I do?
    - Do I need to purge first then bulk index each time?
  - Required Permissions For Mattermost Service Account
  - Elasticsearch
  - AWS OpenSearch Service
  - Supported paths

- Frequently asked questions (FAQ)
  - Do I need to use Elasticsearch or AWS OpenSearch?
  - Should I install Elasticsearch or OpenSearch on the same machine as Mattermost Server?
  - What types of indexes are created?
  - Can I pause an search indexing job?
  - Can an index rollover policy be defined?
  - Are there any new search features offered with Elasticsearch?
  - Are my files stored in Elasticsearch or OpenSearch?
  - How do I monitor system health of an Elasticsearch server?
  - What form of data is sent to Elasticsearch or OpenSearch?
  - How much data is sent to Elasticsearch or OpenSearch and when?
  - How do I know if a search job fails?
  - My search indexes won’t complete, what should I do?
  - Do I need to purge first then bulk index each time?
- Required Permissions For Mattermost Service Account

- Mattermost documentation

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

High availability cluster-based deployment

A high availability cluster-based deployment enables a Mattermost system to maintain service during outages and hardware failures through the use of redundant infrastructure.

High availability in Mattermost consists of running redundant Mattermost application servers, redundant database servers, and redundant load balancers. The failure of any one of these components does not interrupt operation of the system.

Mattermost Enterprise supports:

1. Clustered Mattermost servers, which minimize latency by:

- Storing static assets over a global CDN.
- Deploying multiple Mattermost servers to host API communication closer to the location of end users.

They can also be used to handle scale and failure handoffs in disaster recovery scenarios.

2. Database read replicas, where replicas can be:

- Configured as a redundant backup to the active database server.
- Used to scale up the number of concurrent users.
- Deployed closer to the location of end users, reducing latency.

Moreover, search replicas are also supported to handle search queries.

Deployment guide

Set up and maintain a high availability cluster-based deployment on your Mattermost servers. This document doesn't cover the configuration of databases in terms of disaster recovery, however, you can refer to the frequently asked questions (FAQ) section for our recommendations.

To ensure your instance and configuration are compatible with a high availability cluster-based deployment, please review the configuration and compatibility section.

> Note
>
> Back up your Mattermost database and file storage locations before configuring high availability. For more information about backing up, see Backup and disaster recovery.

1. Set up a new Mattermost server by following one of our **Install Guides**. This server must use an identical copy of the configuration file, `config.json`. Verify the servers are functioning by hitting each independent server through its private IP address.
2. Modify the `config.json` files on both servers to add `ClusterSettings`. See the high availability cluster-based deployment configuration settings documentation for details.
3. Verify the configuration files are identical on both servers then restart each machine in the cluster.
4. Modify your NGINX setup so that it proxies to both servers. For more information about this, see proxy server configuration.
5. Open **System Console > Environment > High Availability** to verify that each machine in the cluster is communicating as expected with green status indicators. If not, investigate the log files for any extra information.

Add a server to the cluster

1. Back up your Mattermost database and the file storage location. See the backup documentation for details.
2. Set up a new Mattermost server. This server must use an identical copy of the configuration file, `config.json`. Verify the server is functioning by hitting the private IP address.
3. Modify your NGINX setup to add the new server.
4. Open **System Console > Environment > High Availability** to verify that all the machines in the cluster are communicating as expected with green status indicators. If not, investigate the log files for any extra information.

Remove a server from the cluster

1. Back up your Mattermost database and the file storage location. See the backup documentation for details.
2. Modify your NGINX setup to remove the server. For information about this, see proxy server configuration documentation for details.
3. Open **System Console > Environment > High Availability** to verify that all the machines remaining in the cluster are communicating as expected with green status indicators. If not, investigate the log files for any extra information.

Configuration and compatibility

Details on configuring your system for High Availability.

Mattermost Server configuration

Configuration settings

1. High availability is configured in the `ClusterSettings` section of `config.json` and the settings are viewable in the System Console. When high availability is enabled, the System Console is set to read-only mode to ensure all the `config.json` files on the Mattermost servers are always identical. However, for testing and validating a high availability setup, you can set `ReadOnlyConfig` to `false`, which allows changes made in the System Console to be saved back to the configuration file.

> ```
> "ClusterSettings": {  "Enable": false,  "ClusterName": "production",
> "OverrideHostname": "",  "UseIpAddress": true,  "ReadOnlyConfig": true,
> "GossipPort": 8074 }
> ```
>
> *For more details on these settings, see the high availability configuration settings documentation.*

2. Change the process limit to 8192 and the maximum number of open files to 65536.

> *Modify* `/etc/security/limits.conf` *on each machine that hosts a Mattermost server by adding the following lines:*
>
> ```
> * soft nofile 65536 * hard nofile 65536 * soft nproc 8192 * hard nproc 8192
> ```

3. Increase the number of WebSocket connections:

> *Modify* `/etc/sysctl.conf` *on each machine that hosts a Mattermost server by adding the following lines:*
>
> ```
> # Extending default port range to handle lots of concurrent connections. net.ipv4.ip_local_port_range = 1025 65000 # Lowering the timeout to faster recycle connections in the FIN-WAIT-2 state. net.ipv4.tcp_fin_timeout = 30 # Reuse TIME-WAIT sockets for new outgoing connections. net.ipv4.tcp_tw_reuse = 1 # Bumping the limit of a listen() backlog. # This is maximum number of established sockets (with an ACK) # waiting to be accepted by the listening process. net.core.somaxconn = 4096 # Increasing the maximum number of connection requests which have # not received an acknowledgment from the client. # This is helpful to handle sudden bursts of new incoming connections. net.ipv4.tcp_max_syn_backlog = 8192 # This is tuned to be 2% of the available memory. vm.min_free_kbytes = 167772 # Disabling slow start helps increasing overall throughput # and performance of persistent single connections. net.ipv4.tcp_slow_start_after_idle = 0 # These show a good performance improvement over defaults. # More info at https://blog.cloudflare.com/http-2-prioritization-with-nginx/ net.ipv4.tcp_congestion_control = bbr net.core.default_qdisc = fq net.ipv4.tcp_notsent_lowat = 16384 # TCP buffer sizes are tuned for 10Gbit/s bandwidth and 0.5ms RTT (as measured intra EC2 cluster). # This gives a BDP (bandwidth-delay-product) of 625000 bytes. net.ipv4.tcp_rmem = 4096 156250 625000 net.ipv4.tcp_wmem = 4096 156250 625000 net.core.rmem_max = 312500 net.core.wmem_max = 312500 net.core.rmem_default = 312500 net.core.wmem_default = 312500 net.ipv4.tcp_mem = 1638400 1638400 1638400
> ```

You can do the same for the proxy server.

Cluster discovery

If you have non-standard (i.e. complex) network configurations, then you may need to use the Override Hostname setting to help the cluster nodes discover each other. The cluster settings in the config are removed from the config file

hash for this reason, meaning you can have `config.json` files that are slightly different in high availability mode. The Override Hostname is intended to be different for each clustered node in `config.json` if you need to force discovery.

If `UseIpAddress` is set to `true`, it attempts to obtain the IP address by searching for the first non-local IP address (non-loop-back, non-localunicast, non-localmulticast network interface). It enumerates the network interfaces using the built-in go function [net.InterfaceAddrs() (https://pkg.go.dev/net#InterfaceAddrs)](https://pkg.go.dev/net#InterfaceAddrs). Otherwise it tries to get the hostname using the [os.Hostname() (https://pkg.go.dev/os#Hostname)](https://pkg.go.dev/os#Hostname) built-in go function.

You can also run `SELECT * FROM ClusterDiscovery` against your database to see how it has filled in the **Hostname** field. That field will be the hostname or IP address the server will use to attempt contact with other nodes in the cluster. We attempt to make a connection to the `url Hostname:Port` and `Hostname:PortGossipPort`. You must also make sure you have all the correct ports open so the cluster can gossip correctly. These ports are under `ClusterSettings` in your configuration.

In short, you should use:

1. IP address discovery if the first non-local address can be seen from the other machines.
2. Override Hostname on the operating system so that it's a proper discoverable name for the other nodes in the cluster.
3. Override Hostname in `config.json` if the above steps do not work. You can put an IP address in this field if needed. The `config.json` will be different for each cluster node.

Time synchronization

Each server in the cluster must have the Network Time Protocol daemon `ntpd` running so that messages are posted in the correct order.

State

The Mattermost server is designed to have very little state to allow for horizontal scaling. The items in state considered for scaling Mattermost are listed below:

- In memory session cache for quick validation and channel access.
- In memory online/offline cache for quick response.
- System configuration file that is loaded and stored in memory.
- WebSocket connections from clients used to send messages.

When the Mattermost server is configured for high availability, the servers use an inter-node communication protocol on a different listening address to keep the state in sync. When a state changes it is written back to the database and an inter-node message is sent to notify the other servers of the state change. The true state of the items can always be read from the database. Mattermost also uses inter-node communication to forward WebSocket messages to the other servers in the cluster for real-time messages such as "[User X] is typing."

Proxy server configuration

The proxy server exposes the cluster of Mattermost servers to the outside world. The Mattermost servers are designed for use with a proxy server such as NGINX, a hardware load balancer, or a cloud service like Amazon Elastic Load Balancer.

If you want to monitor the server with a health check you can use `http://10.10.10.2/api/v4/system/ping` and check the response for `Status 200`, indicating success. Use this health check route to mark the server *in-service* or *out-of-service*.

A sample configuration for NGINX is provided below. It assumes that you have two Mattermost servers running on private IP addresses of `10.10.10.2` and `10.10.10.4`.

```
upstream backend { server 10.10.10.2:8065; server 10.10.10.4:8065; keepalive
256; } proxy_cache_path /var/cache/nginx levels=1:2
keys_zone=mattermost_cache:50m max_size=16g inactive=60m use_temp_path=off;
server { listen 80 reuseport; server_name mattermost.example.com; location
~ /api/v[0-9]+/(users/)?websocket$ { proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade"; client_max_body_size 100M;
proxy_set_header Host $http_host; proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme; proxy_set_header X-Frame-Options
SAMEORIGIN; proxy_buffers 256 16k; proxy_buffer_size 16k;
client_body_timeout 60s; send_timeout 300s; lingering_timeout 5s;
proxy_connect_timeout 30s; proxy_send_timeout 90s; proxy_read_timeout 90s;
proxy_http_version 1.1; proxy_pass http://backend; } location /
{ proxy_set_header Connection ""; client_max_body_size 100M;
proxy_set_header Host $http_host; proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme; proxy_set_header X-Frame-Options
SAMEORIGIN; proxy_buffers 256 16k; proxy_buffer_size 16k;
client_body_timeout 60s; send_timeout 300s; lingering_timeout 5s;
proxy_connect_timeout 30s; proxy_send_timeout 90s; proxy_read_timeout 90s;
proxy_http_version 1.1; proxy_pass http://backend; proxy_cache
mattermost_cache; proxy_cache_revalidate on; proxy_cache_min_uses 2;
proxy_cache_use_stale timeout; proxy_cache_lock on; } }
```

You can use multiple proxy servers to limit a single point of failure, but that is beyond the scope of this documentation.

File storage configuration

> Note
>
> 1. File storage is assumed to be shared between all the machines that are using services such as NAS or Amazon S3.
> 2. If `"DriverName": "local"` is used then the directory at `"FileSettings": "Directory": "./data/"` is expected to be a NAS location mapped as a local directory, otherwise high availability will not function correctly and may corrupt your file storage.
> 3. If you’re using Amazon S3 or MinIO for file storage then no other configuration is required.

If you're using the Compliance Reports feature in Mattermost Enterprise, you need to configure the `"ComplianceSettings": "Directory": "./data/",` to share between all machines or the reports will only be available from the System Console on the local Mattermost server.

Migrating to NAS or S3 from local storage is beyond the scope of this document.

Database configuration

> Tip
>
> Specifying configuration setting values using Mattermost environment variables ensure that they always take precedent over any other configuration settings.

For an AWS High Availability RDS cluster deployment, point the datasource configuration setting to the write/read endpoint at the **cluster** level to benefit from the AWS failover handling. AWS takes care of promoting different database nodes to be the writer node. Mattermost doesn't need to manage this.

Use the read replica feature to scale the database. The Mattermost server can be set up to use one master database and one or more read replica databases.

> Note
>
> For an AWS High Availability RDS cluster deployment, don't hard-code the IP addresses. Point this configuration setting to the write/read endpoint at the **cluster** level. This will benefit from the AWS failover handling where AWS takes care of promoting different database nodes to be the writer node. Mattermost doesn't need to manage this.

On large deployments, also consider using the search replicas feature to isolate search queries onto one or more search replicas. A search replica is similar to a read replica, but is used only for handling search queries.

> Note
>
> For an AWS High Availability RDS cluster deployment, don't hard-code the IP addresses. Point this configuration setting directly to the underlying read-only node endpoints within the RDS cluster. We recommend circumventing the failover/load balancing that AWS/RDS takes care of (except for the write traffic), and populating the `DataSourceReplicas` array with the RDS reader node endpoints. Mattermost has its own method of balancing the read-only connections, and can also balance those queries to the DataSource/write+read connection should those nodes fail.

Mattermost distributes queries as follows:

- All write requests, and some specific read requests, are sent to the master.
- All other read requests (excluding those specific queries that go to the master) are distributed among the available read replicas. If no read replicas are available, these are sent to the master instead.
- Search requests are distributed among the available search replicas. If no search replicas are available, these are sent to the read replicas instead (or, if no read replicas are available, to the master).

Size the databases

For information about sizing database servers, see Hardware requirements for enterprise deployments (multi-server).

In a master/slave environment, make sure to size the slave machine to take 100% of the load in the event that the master machine goes down and you need to fail over.

Deploy a multi-database configuration

To configure a multi-database Mattermost server:

1. Update the `DataSource` setting in `config.json` with a connection string to your master database server.

2. Update the `DataSourceReplicas` setting in `config.json` with a series of connection strings to your database read replica servers in the format `["readreplica1", "readreplica2"]`. Each connection should also be compatible with the `DriverName` setting.

Here's an example `SqlSettings` block for one master and two read replicas:

```
"SqlSettings": {  "DriverName": "mysql",  "DataSource":
"master_user:master_password@tcp(master.server)/mattermost?
charset=utf8mb4,utf8\u0026readTimeout=30s\u0026writeTimeout=30s",
"DataSourceReplicas": ["slave_user:slave_password@tcp(replica1.server)/
mattermost?
charset=utf8mb4,utf8\u0026readTimeout=30s\u0026writeTimeout=30s","slave_user:slave_password@tc
mattermost?
charset=utf8mb4,utf8\u0026readTimeout=30s\u0026writeTimeout=30s"],
"DataSourceSearchReplicas": [],  "MaxIdleConns": 20,  "MaxOpenConns": 300,
"Trace": false,  "AtRestEncryptKey": "",  "QueryTimeout": 30  }
```

The new settings can be applied by either stopping and starting the server, or by loading the configuration settings as described in the next section.

Once loaded, database write requests are sent to the master database and read requests are distributed among the other databases in the list.

Load a multi-database configuration onto an active server

After a multi-database configuration has been defined in `config.json`, the following procedure can be used to apply the settings without shutting down the Mattermost server:

1. Go to **System Console > Environment > Web Server**, then select **Reload Configuration from Disk** to reload configuration settings for the Mattermost server from `config.json`.

2. Go to **System Console > Environment > Database**, then select **Recycle Database Connections** to take down existing database connections and set up new connections in the multi-database configuration.

While the connection settings are changing, there might be a brief moment when writes to the master database are unsuccessful. The process waits for all existing connections to finish and starts serving new requests with the new connections. End users attempting to send messages while the switch is

happening will have an experience similar to losing connection to the Mattermost server.

Manual failover for master database

If the need arises to switch from the current master database - for example, if it is running out of disk space, or requires maintenance updates, or for other reasons - you can switch Mattermost server to use one of its read replicas as a master database by updating `DataSource` in `config.json`.

To apply the settings without shutting down the Mattermost server:

1. Go to **System Console > Environment > Web Server**, then select **Reload Configuration from Disk** to reload configuration settings for the Mattermost server from `config.json`.
2. Go to **System Console > Environment > Database**, then select **Recycle Database Connections** to take down existing database connections and set up new connections in the multi-database configuration.

While the connection settings are changing, there might be a brief moment when writes to the master database are unsuccessful. The process waits for all existing connections to finish and starts serving new requests with the new connections. End users attempting to send messages while the switch is happening can have an experience similar to losing connection to the Mattermost server.

Transparent failover

The database can be configured for high availability and transparent failover use the existing database technologies. We recommend PostgreSQL Clustering or Amazon Aurora. Database transparent failover is beyond the scope of this documentation.

Recommended configuration settings for PostgreSQL

For the Postgres service we recommend the following configuration optimizations on your Mattermost server. These configurations were tested on an AWS Aurora r5.xlarge instance. There are also some general optimizations mentioned which requires servers with higher specifications.

Config for Postgres Primary or Writer node

```
# If the instance is lower capacity than r5.xlarge, then set it to a lower
number. # Also tune the "MaxOpenConns" setting under the "SqlSettings" of
the Mattermost app accordingly. # Note that "MaxOpenConns" on Mattermost is
per data source name. max_connections = 1024 # Set it to 1.1, unless the DB
is using spinning disks. random_page_cost = 1.1 # This should be 32MB if
using read replicas, or 16MB if using a single PostgreSQL instance.
# If the instance is of a lower capacity than r5.xlarge, then set it to a
lower number. work_mem = 32MB # Set both of the below settings to 65% of
total memory. For a 32 GB instance, it should be 21 GB. # If on a smaller
server, set this to 20% or less total RAM. # ex: 512MB would work for a 4GB
RAM server effective_cache_size = 21GB shared_buffers = 21GB # If you are
using pgbouncer, or any similar connection pooling proxy,
# in front of your DB, then apply the keepalive settings to the proxy
instead, # and revert the keepalive settings for the DB back to defaults.
tcp_keepalives_idle = 5 tcp_keepalives_interval = 1 tcp_keepalives_count =
5 # 1GB (reduce this to 512MB if your server has less than 32GB of RAM)
maintenance_work_mem = 512MB autovacuum_max_workers = 4
autovacuum_vacuum_cost_limit = 500 # If you have more than 32 CPUs on your
database server, please set the following options to utilize more CPU for
your server: max_worker_processes = 12 max_parallel_workers_per_gather = 4
max_parallel_workers = 12 max_parallel_maintenance_workers = 4
```

Config for Postgres Replica node

Copy all the above settings to the read replica, and modify or add only the below.

```
# If the instance is lower capacity than r5.xlarge, then set it to a lower
number. # Also tune the "MaxOpenConns" setting under the "SqlSettings" of
the Mattermost app accordingly. # Note that "MaxOpenConns" on Mattermost is
per data source name. max_connections = 1024 # This setting should be 16MB
on read nodes, and 32MB on writer nodes work_mem = 16MB
# The below settings allow the reader to return query results even when the
primary has a write process running, a query conflict. # This is set to on
because of the high volume of write traffic that can prevent the reader from
returning query results within the timeout. # https://www.postgresql.org/
docs/current/hot-standby.html#HOT-STANDBY-CONFLICT hot_standby = on
hot_standby_feedback = on
```

A note on vacuuming

Performance of a Postgres database is particularly sensitive to [vacuuming and analyzing (https://www.postgresql.org/docs/current/sql-vacuum.html)](https://www.postgresql.org/docs/current/sql-vacuum.html). A good way to check how frequently tables are vacuumed is with this query:

```
SELECT relname, n_tup_ins as inserts,n_tup_upd as updates,n_tup_del as
deletes, n_live_tup as live_tuples, n_dead_tup as dead_tuples,
n_mod_since_analyze, last_autovacuum, last_autoanalyze, autovacuum_count,
autoanalyze_count FROM pg_stat_user_tables order by dead_tuples desc LIMIT
10;
```

The output of this query will indicate which tables have accumulated the most dead tuples. You can also look at the `last_autovacuum` and `last_autoanalyze` columns to see when the last autovacuum or autoanalyze ran.

Depending on those values, you can choose to tune table-specific values for autovacuum or autoanalyze thresholds. For example, if you see more than 50,000 dead tuples on a table, and it hasn't been vacuumed or analyzed in the last 6 months, there's a good chance that it would benefit from more aggressive vacuuming. In that case, you can run this to tune your tables:

```
ALTER TABLE <table> SET (  autovacuum_vacuum_scale_factor = 0.1, -- default
is 0.2  autovacuum_analyze_scale_factor = 0.05, -- default is 0.1
autovacuum_vacuum_cost_limit = 1000 -- default is 200 );
```

Feel free to choose different values as necessary. Refer to https://www.postgresql.org/docs/current/routine-vacuuming.html#AUTOVACUUM for more information on how does Postgres calculate when to run vacuuming. Re-run the initial SQL query from time to time and adjust values accordingly.

Leader election

A cluster leader election process assigns any scheduled task such as LDAP sync to run on a single node in a multi-node cluster environment.

The process is based on a widely used [bully leader election algorithm (https://en.wikipedia.org/wiki/Bully_algorithm)](https://en.wikipedia.org/wiki/Bully_algorithm) where the process with the lowest node ID number from amongst the non-failed processes is selected as the leader.

Job server

Mattermost runs periodic tasks via the job server. These tasks include:

- LDAP sync
- Data retention
- Compliance exports
- Elasticsearch indexing

Make sure you have set `JobSettings.RunScheduler` to `true` in `config.json` for all app and job servers in the cluster. The cluster leader will then be responsible for scheduling recurring jobs.

> Note
>
> It is strongly recommended not to change this setting from the default setting of `true` as this prevents the `ClusterLeader` from being able to run the scheduler. As a result, recurring jobs such as LDAP sync, Compliance Export, and data retention will no longer be scheduled.

In previous Mattermost Server versions, and this documentation, the instructions stated to run the Job Server with `RunScheduler: false`. The cluster design has evolved and this is no longer the case.

Plugins and High Availability

When you install or upgrade a plugin, it's propagated across the servers in the cluster automatically. File storage is assumed to be shared between all the servers, using services such as NAS or Amazon S3.

If `"DriverName": "local"` is used then the directory at `"FileSettings": "Directory": "./data/"` is expected to be a NAS location mapped as a local directory. If this is not the case High Availability will not function correctly and may corrupt your file storage.

When you reinstall a plugin in v5.14, the previous **Enabled** or **Disabled** state is retained. As of v5.15, a reinstalled plugin's initial state is **Disabled**.

CLI and High Availability

The CLI is run in a single node which bypasses the mechanisms that a high availability environment uses to perform actions across all nodes in the cluster. As a result, when running CLI commands in a High Availability environment, tasks such as updating and deleting users or changing configuration settings require a server restart.

We recommend using mmctl in a high availability environment instead since a server restart is not required. These changes are made through the API layer, so the node receiving the change request notifies all other nodes in the cluster.

Upgrade guide

An update is an incremental change to Mattermost server that fixes bugs or performance issues. An upgrade adds new or improved functionality to the server.

> Tip
>
> To learn how to safely upgrade your deployment in Kubernetes for High Availability and Active/Active support, see the Upgrading Mattermost in Kubernetes and High Availability Environments documenation.

Update configuration changes while operating continuously

A service interruption is not required for most configuration updates. See the section below for details on upgrades requiring service interruption. You can apply updates during a period of low load, but if your high availability cluster-based deployment is sized correctly, you can do it at any time. The system downtime is brief, and depends on the number of Mattermost servers in your cluster. Note that you are not restarting the machines, only the Mattermost server applications. A Mattermost server restart generally takes about five seconds.

> Note
>
> Don't modify configuration settings through the System Console, otherwise you'll have two servers with different `config.json` files in a high availability cluster-based deployment causing a refresh every time a user connects to a different app server.

1. Make a backup of your existing `config.json` file.
2. For one of the Mattermost servers, make the configuration changes to `config.json` and save the file. Do not reload the file yet.
3. Copy the `config.json` file to the other servers.
4. Shut down Mattermost on all but one server.
5. Reload the configuration file on the server that is still running. Go to **System Console > Environment > Web Server**, then select **Reload Configuration from Disk**.
6. Start the other servers.

Update the Server version while operating continuously

A service interruption is not required for security patch dot releases of Mattermost Server. You can apply updates during a period when the anticipated load is small enough that one server can carry the full load of the system during the update.

> Note
>
> Mattermost supports one minor version difference between the server versions when performing a rolling upgrade (for example v5.27.1 + v5.27.2 or v5.26.4 + v5.27.1 is supported, whereas v5.25.5 + v5.27.0 is not supported). Running two different versions of Mattermost in your cluster should not be done outside of an upgrade scenario.

When restarting, you aren't restarting the machines, only the Mattermost server applications. A Mattermost server restart generally takes about five seconds.

1. Review the upgrade procedure in the *Upgrade Enterprise Edition* section of Upgrade Mattermost Server.
2. Make a backup of your existing `config.json` file.
3. Set your proxy to move all new requests to a single server. If you are using NGINX and it's configured with an upstream backend section in `/etc/nginx/sites-available/mattermost` then comment out all but the one server that you intend to update first, and reload NGINX.
4. Shut down Mattermost on each server except the one that you are updating first.
5. Update each Mattermost instance that is shut down.
6. On each server, replace the new `config.json` file with your backed up copy.
7. Start the Mattermost servers.
8. Repeat the update procedure for the server that was left running.

Server upgrades requiring service interruption

A service interruption is required when the upgrade includes a change to the database schema or when a change to `config.json` requires a server restart, such as when making the following changes:

- Default server language
- Rate limiting
- Webserver mode
- Database
- High availability

If the upgrade includes a change to the database schema, the database is upgraded by the first server that starts.

Apply upgrades during a period of low load. The system downtime is brief, and depends on the number of Mattermost servers in your cluster. Note that you are not restarting the machines, only the Mattermost server applications.

1. Review the upgrade procedure in the *Upgrade Enterprise Edition* section of Upgrade Mattermost Server.
2. Make a backup of your existing `config.json` file.
3. Stop NGINX.
4. Upgrade each Mattermost instance.
5. On each server, replace the new `config.json` file with your backed up copy.
6. Start one of the Mattermost servers.
7. When the server is running, start the other servers.
8. Restart NGINX.

All cluster nodes must use a single protocol

All cluster traffic uses the gossip protocol. Gossip clustering can no longer be disabled.

When upgrading a high availability cluster-based deployment, you can't upgrade other nodes in the cluster when one node isn't using the gossip protocol. You must use gossip to complete this type of upgrade. Alternatively you can shut down all nodes and bring them all up individually following an upgrade.

Requirements for continuous operation

To enable continuous operation at all times, including during server updates and server upgrades, you must make sure that the redundant components are properly sized and that you follow the correct sequence for updating each of the system's components.

Redundancy at anticipated scale

> Upon failure of one component, the remaining application servers, database servers, and load balancers must be sized and configured to

carry the full load of the system. If this requirement is not met, an outage of one component can result in an overload of the remaining components, causing a complete system outage.

Update sequence for continuous operation

You can apply most configuration changes and dot release security updates without interrupting service, provided that you update the system components in the correct sequence. See the upgrade guide for instructions on how to do this.

Exception: Changes to configuration settings that require a server restart, and server version upgrades that involve a change to the database schema, require a short period of downtime. Downtime for a server restart is around five seconds. For a database schema update, downtime can be up to 30 seconds.

Important

Mattermost does not support high availability deployments spanning multiple datacenters. All nodes in a high availability cluster must reside within the same datacenter to ensure proper functionality and performance.

Frequently asked questions (FAQ)

Does Mattermost support multi-region high availability cluster-based deployment?

Yes. Although not officially tested, you can set up a cluster across AWS regions, for example, and it should work without issues.

What does Mattermost recommend for disaster recovery of the databases?

When deploying Mattermost in a high availability configuration, we recommend using a database load balancer between Mattermost and your database. Depending on your deployment this needs more or less consideration.

For example, if you’re deploying Mattermost on AWS with Amazon Aurora we recommend utilizing multiple Availability Zones. If you’re deploying

Mattermost on your own cluster please consult with your IT team for a solution best suited for your existing architecture.

How to find the hostname of the connected websocket?

From Mattermost v10.4, Enterprise customers running self-hosted deployments can go to the **Product** menu

and select **About Mattermost** to see the hostname of the node in the cluster running Mattermost.

Troubleshooting

Capture high availability troubleshooting data

When deploying Mattermost in a high availability configuration, we recommend that you keep Prometheus and Grafana metrics as well as cluster server logs for as long as possible - and at minimum two weeks.

You may be asked to provide this data to Mattermost for analysis and troubleshooting purposes.

> Note
>
> - Ensure that server log files are being created. You can find more on working with Mattermost logs here.
> - When investigating and replicating issues, we recommend opening **System Console > Environment > Logging** and setting **File Log Level** to **DEBUG** for more complete logs. Make sure to revert to **INFO** after troubleshooting to save disk space.
> - Each server has its own server log file, so make sure to provide server logs for all servers in your High Availability cluster-based deployment.

Red server status

When high availability mode is enabled, the System Console displays the server status as red or green, indicating if the servers are communicating correctly with the cluster. The servers use inter-node communication to ping the other machines in the cluster, and once a ping is established the servers exchange information, such as server version and configuration files.

A server status of red can occur for the following reasons:

- **Configuration file mismatch:** Mattermost will still attempt the inter-node communication, but the System Console will show a red status for the server since the high availability mode feature assumes the same configuration file to function properly.
- **Server version mismatch:** Mattermost will still attempt the inter-node communication, but the System Console will show a red status for the server since the high availability mode feature assumes the same version of Mattermost is installed on each server in the cluster. It is recommended to use the latest version of Mattermost (https://mattermost.com/download/) on all servers. Follow the upgrade procedure in Upgrade Mattermost Server for any server that needs to be upgraded.
- **Server is down:** If an inter-node communication fails to send a message it makes another attempt in 15 seconds. If the second attempt fails, the server is assumed to be down. An error message is written to the logs and the System Console shows a status of red for that server. The inter-node communication continues to ping the down server in 15 second intervals. When the server comes back up, any new messages are sent to it.

WebSocket disconnect

When a client WebSocket receives a disconnect it will automatically attempt to re-establish a connection every three seconds with a backoff. After the connection is established, the client attempts to receive any messages that were sent while it was disconnected.

App refreshes continuously

When configuration settings are modified through the System Console, the client refreshes every time a user connects to a different app server. This

occurs because the servers have different `config.json` files in a high availability cluster-based deployment.

Modify configuration settings directly through `config.json` following these steps.

Messages do not post until after reloading

When running in high availability mode, make sure all Mattermost application servers are running the same version of Mattermost. If they are running different versions, it can lead to a state where the lower version app server cannot handle a request and the request will not be sent until the frontend application is refreshed and sent to a server with a valid Mattermost version. Symptoms to look for include requests failing seemingly at random or a single application server having a drastic rise in goroutines and API errors.

On this page

- High availability cluster-based deployment
  - Deployment guide
    - Add a server to the cluster
    - Remove a server from the cluster
  - Configuration and compatibility
    - Mattermost Server configuration
      - Configuration settings
      - Cluster discovery
      - Time synchronization
      - State
      - Proxy server configuration
      - File storage configuration
      - Database configuration
        - Size the databases
        - Deploy a multi-database configuration
        - Load a multi-database configuration onto an active server
        - Manual failover for master database
        - Transparent failover
        - Recommended configuration settings for PostgreSQL

        - Leader election
        - Job server
        - Plugins and High Availability
        - CLI and High Availability
    - Upgrade guide
        - Update configuration changes while operating continuously
        - Update the Server version while operating continuously
        - Server upgrades requiring service interruption
        - All cluster nodes must use a single protocol
    - Requirements for continuous operation
    - Frequently asked questions (FAQ)
        - Does Mattermost support multi-region high availability cluster-based deployment?
        - What does Mattermost recommend for disaster recovery of the databases?
        - How to find the hostname of the connected websocket?
    - Troubleshooting
        - Capture high availability troubleshooting data
        - Red server status
        - WebSocket disconnect
        - App refreshes continuously
        - Messages do not post until after reloading

- Mattermost documentation

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

AWS OpenSearch server setup

AWS OpenSearch Service allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. The indexing process can be managed from the System Console after setting up and connecting an OpenSearch server. The post index is stored on the OpenSearch server and updated constantly after new posts are made. In order to index existing posts, a bulk index of the entire post database must be generated.

Deploying AWS OpenSearch includes the following two steps: setting up AWS OpenSearch, and configuring Mattermost.

Set up AWS OpenSearch

From Mattermost v9.11, beta support is available for [AWS OpenSearch v1.x and v2.x (https://opensearch.org/)](https://opensearch.org/). This document covers both on-premises and AWS OpenSearch setup, including manual steps and Terraform examples.

We highly recommend that you set up an AWS OpenSearch server on a separate machine from the Mattermost server.

On-Premises OpenSearch

1. To install on-premise OpenSearch, provision a dedicated server (e.g. Ubuntu 22.04 LTS).

2. Install Java (OpenSearch requires Java 11+):

```
sudo apt update sudo apt install -y openjdk-11-jdk java -version
```

3. Download & extract OpenSearch 2.x:

```
wget https://artifacts.opensearch.org/releases/bundle/opensearch/2.9.0/
opensearch-2.9.0-linux-x64.tar.gz tar -xzf opensearch-2.9.0-linux-
x64.tar.gz sudo mv opensearch-2.9.0 /usr/share/opensearch
```

4. Create a dedicated user & set permissions:

```
sudo useradd --no-create-home --shell /bin/false opensearch sudo chown
-R opensearch:opensearch /usr/share/opensearch
```

5. Configure systemd:

```
[Unit]  Description=OpenSearch  Wants=network-online.target
After=network-online.target [Service]  Type=notify  User=opensearch
Group=opensearch  ExecStart=/usr/share/opensearch/bin/opensearch
Restart=on-failure  LimitNOFILE=65536  LimitNPROC=4096 [Install]
WantedBy=multi-user.target
```

6. Edit `opensearch.yml` to include the following:

```
cluster.name: mattermost-cluster node.name: node-1 path.data: /var/lib/
opensearch path.logs: /var/log/opensearch network.host: 0.0.0.0
discovery.seed_hosts: ["<other-node-ip>"] cluster.initial_master_nodes:
["node-1", "node-2"]
```

7. Enable & start OpenSearch:

```
sudo systemctl daemon-reload sudo systemctl enable opensearch sudo
systemctl start opensearch sudo systemctl status opensearch
```

Terraform (Docker) Example

```
provider "docker" {  host = "unix:///var/run/docker.sock" } resource
"docker_image" "opensearch" {  name = "opensearchproject/
opensearch:2.9.0" } resource "docker_container" "opensearch" {  name =
"opensearch"  image = docker_image.opensearch.latest  ports {  internal =
9200  external = 9200  }  ports {  internal = 9600  external = 9600  }  env
= [  "cluster.name=mattermost-cluster",  "network.host=0.0.0.0",
"discovery.type=single-node", # remove for multi-node  ]  restart = "unless-
stopped" }
```

AWS OpenSearch Console Setup

1. To install AWS OpenSearch, open the **AWS Console > OpenSearch Service**.
2. Create a domain, where:

> - *Domain name:* `mattermost-os`
> - *Engine version:* `OpenSearch 2.x`.

3. Configure the cluster, where:

> - *instance type:* `r6g.xlarge.search`
> - *data nodes: 2*
> - *master nodes: 2*
> - *storage: EBS gp3 (1536 GiB, 4608 IOPS, 250 MiB/s)*

4. Specify the network for: VPC with 2 subnets, and a security group allowing Mattermost IPs on port `443`.

Note

> *Port 9200 is commonly used for local or on-premise OpenSearch. The AWS OpenSearch domain only supports HTTPS over port 443.*

5. Configure the access policy (JSON):

```
{  "Version": "2012-10-17",  "Statement": [{  "Effect": "Allow",
"Principal": { "AWS": "arn:aws:iam::123456789012:role/
MattermostAppRole" },  "Action": "es:*",  "Resource":
"arn:aws:es:us-east-1:123456789012:domain/mattermost-os/*" }] }
```

6. Configure the following advanced settings (JSON):

```
{  "action.destructive_requires_name": "false",
"rest.action.multi.allow_explicit_index": "true",
"indices.query.bool.max_clause_count": "1024",
"indices.fielddata.cache.size": "20" }
```

7. Configure the automated snapshot start hour as 23 (UTC), enforce HTTPS, then review & create.

8. To test, run the following command:

```
curl https://mattermost-os-xxxxxxxxxxx.us-east-1.es.amazonaws.com
```

AWS Terraform Example

```
provider "aws" {  region = "us-east-1" } resource "aws_iam_role"
"os_service_role" {  name = "OSServiceRole"  assume_role_policy = <<EOF
{  "Version": "2012-10-17",  "Statement": [{  "Action":
"sts:AssumeRole",  "Effect": "Allow",  "Principal": { "Service":
"es.amazonaws.com" }  }] } EOF } resource "aws_opensearch_domain"
"mattermost" {  domain_name = "mattermost-os"  engine_version =
"OpenSearch_2.9"  cluster_config {  instance_type = "r6g.xlarge.search"
instance_count = 2  dedicated_master_enabled = true
dedicated_master_type = "r6g.xlarge.search"  dedicated_master_count = 2
zone_awareness_enabled = true  }  ebs_options {  ebs_enabled = true
volume_type = "gp3"  volume_size = 1536  iops = 4608  }  vpc_options {
subnet_ids = ["subnet-blah1", "subnet-blah2"]  security_group_ids =
["sg-1234567890"]  }  advanced_options = {
"rest.action.multi.allow_explicit_index" = "true"
"indices.query.bool.max_clause_count" = "1024"
"indices.fielddata.cache.size" = "20"
"action.destructive_requires_name" = "false"  }  access_policies =
<<POLICY {  "Version": "2012-10-17",  "Statement": [{  "Effect":
"Allow",  "Principal": {  "AWS": "arn:aws:iam::123456789012:role/
MattermostAppRole"  },  "Action": "es:*",  "Resource": "arn:aws:es:us-
east-1:123456789012:domain/mattermost-os/*"  }]  }  POLICY
service_software_options {  automated_snapshot_start_hour = 23  }
domain_endpoint_options {  enforce_https = true  } }
```

Configure Mattermost

Follow these steps to configure Mattermost to use your AWS OpenSearch server and to generate the post index:

1. Go to **System Console > Environment > Elasticsearch**.
2. Set **Enable Elasticsearch Indexing** to `true` to enable the other the settings on the page.
3. Ensure **Backend type** is set to `opensearch`.
4. Set the **Server Connection Address** to your Elasticsearch or OpenSearch cluster endpoint.
5. Monitor cluster health: `curl https://mattermost-os-xxxxx.us-east-1.es.amazonaws.com/_cluster/health`

Set server connection details

1. (Optional) Enter **Server Username** used to access the enterprise search server.
2. (Optional) Enter **Server Password** associated with the username.
3. Set **Enable Cluster Sniffing** (Optional). Sniffing finds and connects to all data nodes in your cluster automatically.
4. Optional CA and client certificate configuration settings are available for use with basic authentication credentials or to replace them. See the Enterprise search configuration settings documentation for details.
5. Select **Test Connection** and then select **Save**. If the server connection is unsuccessful you won't be able to save the configuration or enable searching with Elasticsearch or AWS OpenSearch.

Build the post index of existing messages

Select **Index Now**. This process can take up to a few hours depending on the size of the post database and number of messages. The progress percentage can be seen as the index is created. To avoid downtime, set **Enable Elasticsearch for search queries** to `false` so that database search is available during the indexing process.

Enable enterprise search

Ensure bulk indexing is complete before enabling enterprise search, otherwise search results will be incomplete.

Set **Enable Elasticsearch for search queries** to `true`, and setting **Enable Elasticsearch for autocomplete** to `true`. Save your configuration updates and restart the Mattermost server.

> Note
>
> For high post volume deployments, we strongly encourage you to read and properly configure the Mattermost LiveIndexingBatchSize configuration setting.

Once the configuration is saved, new posts made to the database are automatically indexed on the Elasticsearch or AWS OpenSearch server.

Enterprise search limitations

1. Elasticsearch and AWS OpenSearch uses a standard selection of "stop words" to keep search results relevant. Results for the following words will not be returned: "a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", and "with".

2. Searching stop words in quotes returns more results than just the searched terms (ticket (https://mattermost.atlassian.net/browse/MM-7216)).

3. Search results are limited to a user's team and channel membership. This is enforced by the Mattermost server. The entities are indexed in Elasticsearch or AWS OpenSearch in a way that allows Mattermost to filter them when querying, so the Mattermost server narrows down the results on every Elasticsearch or AWS OpenSearch request applying those filters.

On this page

- AWS OpenSearch server setup
- AWS OpenSearch server setup

- Set up AWS OpenSearch
  - On-Premises OpenSearch
    - Terraform (Docker) Example
  - AWS OpenSearch Console Setup
- AWS Terraform Example
- Configure Mattermost
  - Set server connection details
  - Build the post index of existing messages
  - Enable enterprise search
- Enterprise search limitations

- Set up AWS OpenSearch
- AWS Terraform Example
- Configure Mattermost
  - Set server connection details
  - Build the post index of existing messages
  - Enable enterprise search
- Enterprise search limitations

Mattermost performance alerting guide

Mattermost recommends using Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/) to track performance metrics of the Mattermost application servers. The purpose of this guide is to help you set up alerts on your Grafana dashboard once you've set up system health tracking.

> Note
>
> We highly recommend setting up performance alerting for deployments above 5,000 users, where additional servers have been added for performance load-balancing.

Prerequisites

Set up performance monitoring for Mattermost. See our Performance Monitoring documentation to learn more.

To get alerts, first set up a Notification Channel in Grafana. Here's how you can set it up to automatically post alerts in Mattermost:

1. In Mattermost:

> 1. *Create an Alerts channel.*
> 2. *Create an incoming webhook (https://developers.mattermost.com/integrate/webhooks/incoming/) for the Alerts channel and copy the URL.*

2. In Grafana:

> 1. *Under the alert icon in the sidebar, select **Contact points**.*
> 2. *Select **Create contact point**.*
> 3. *Enter **Mattermost Alerts Channel** as the name.*

> *4. For type, select **Slack**.*
>
> *5. Paste your webhook URL into the URL field.*
>
> *6. Include an @ mention in the mention field, if you want to send mentions when an alert is posted to Mattermost.*
>
> *7. Press **Send Test** to test the alert.*

If you would also like to get email alerts, you can follow [these instructions (https://grafana.com/docs/grafana/latest/alerting/)](https://grafana.com/docs/grafana/latest/alerting/) to set that up.

Configure alerts

The [Mattermost dashboards (https://grafana.com/grafana/dashend-user-guide/boards/?search=mattermost)](https://grafana.com/grafana/dashend-user-guide/boards/?search=mattermost) for Grafana come with some partially pre-configured alerts on the following charts:

- CPU Utilization Rate
- Memory Usage
- Number of Goroutines
- Number of API Errors per Second
- Mean API Request Time

To configure alerts, set an appropriate threshold and enable notifications. Enabling notifications is the same for each chart, but setting the correct threshold can have some variances that are better handled on a per-chart basis.

1. For each chart, select the chart name, then select **Edit**:

2. Select the **Alert** tab:

3. The alert threshold, which will be discussed in the sections below, is the last field under **Conditions** (the one set to 600 in the screenshot above).

See the sections below for how to set the threshold for each individual chart. If you would like to add your own custom alert conditions, configure them here.

4. To enable the notifications for any alerts, select the **Notification** tab on the left, then select **Mattermost Alerts Channel** under **Send to**:

Enter a message if you would like to add more context to the alert.

By default, the alerts are configured to check the average of a chart over the last minute to see if that value is above a threshold. If it's above the threshold, the alert will be triggered. Since it's an average over the last minute, small spikes that go past the threshold won't necessarily cause an alert. This helps prevent false positives that would result from natural spikes in usage. The alert state for each chart is evaluated every minute.

Available charts

The sections below describe each chart in more detail.

CPU utilization rate

CPU Utilization Rate is fairly straightforward. CPU Utilization Rate tracks the CPU usage of the app servers as a percentage. The maximum percentage is based on the number of CPU cores or vCPUs your app server has. For example, if you have four CPU cores and your app server was at 100% utilization rate on all four cores, the graph would show 400% for that app server.

It's best to set the alert threshold based on your average CPU utilization and how many cores/vCPUs your app servers have. Take a look at the chart over the last seven days. You'll want to set the threshold somewhere between your maximum CPU usage (cores * 100) and the CPU usage you see. The lower you

set the threshold, the more alerts, and therefore the more false positives, you will get. Set the threshold too high and you may not receive an alert for an incident until it's gotten worse. The same principle applies to all alerts, regardless of the chart.

For example, on our community server, we have the threshold set to 15%:

This value is below our maximum CPU usage and above our average usage at peak times. Therefore, we will get alerts if we begin experiencing unusually high CPU usage.

Memory usage

Memory Usage tracks the megabytes of RAM that your app servers are using. Set the threshold similar to the CPU Utilization Rate: below maximum available memory and above your average usage during peak times.

Here's how we have the alert set on our Community server:

Number of Goroutines

Goroutines are functions or methods that run concurrently with other functions and methods. Goroutines are like lightweight threads with low-creation costs. A rising number of goroutines can be a good measure of the performance of your app servers. A continuous increase indicates your app server can't keep up and is creating goroutines faster than they can complete their tasks and stop.

Set the threshold somewhere above the average number of goroutines you see during peak load times. Small spikes are usually nothing to worry about. It's the uncontrolled climbing of goroutines that you want to watch out for.

Here's how we have it set on our Community server:

Number of API errors per second

Any 4xx or 5xx HTTP response status codes are counted as a REST API error. API errors themselves are not necessarily a problem. There are many legitimate reasons for an API error to occur, such as users' sessions expiring or clients requesting to see if a resource exists and is being given a `404 Not Found` response. It is normal to have some API errors that scale with your installation base.

That said, errors against the REST API can be indicative of deployment and other issues. For example, if one of your app servers did not deploy correctly for whatever reason, it may begin returning a high number of API errors. Another example would be a rogue bot spamming the API with bad requests. Alerts on API errors per second would help catch these and other issues.

Here's how it's set on our Community server:

Mean API request time

The Mean API Request Time is the average amount of time a REST API request to the Mattermost app server takes to complete. If an app server starts to perform poorly, you'll likely see a rise in the mean request time as it takes

longer to complete requests. This could also happen if your database can't sustain the load from the app servers. It may also be indicative of an issue between the app servers and your proxy.

You'll want to set the alert threshold a little above what the mean request time is during your peak load times.

Here's how it's set on our community server:

Plugin hooks

You can trace hooks and plugin API calls with Prometheus. Below are some examples of hooks and API Prometheus metrics that you may want to be aware of when troubleshooting or monitoring your server's performance.

```
# HELP mattermost_plugin_hook_time Time to execute plugin hook handler in
seconds. # TYPE mattermost_plugin_hook_time histogram
mattermost_plugin_hook_time_bucket{hook_name="ChannelHasBeenCreated",plugin_
id="com.mattermost.demo-plugin",success="true",le="0.005"} 0
mattermost_plugin_hook_time_bucket{hook_name="ChannelHasBeenCreated",plugin_
id="com.mattermost.demo-plugin",success="true",le="0.01"} 0
```

```
# HELP mattermost_plugin_multi_hook_time Time to execute multiple plugin
hook handler in seconds. # TYPE mattermost_plugin_multi_hook_time histogram
mattermost_plugin_multi_hook_time_bucket{plugin_id="com.mattermost.custom-
attributes",le="0.005"} 100
mattermost_plugin_multi_hook_time_bucket{plugin_id="com.mattermost.custom-
attributes",le="0.01"} 100
```

```
# HELP mattermost_plugin_multi_hook_server_time Time for the server to
execute multiple plugin hook handlers in seconds. # TYPE
mattermost_plugin_multi_hook_server_time histogram
mattermost_plugin_multi_hook_server_time_bucket{le="0.005"} 1043
```

```
# HELP mattermost_plugin_api_time Time to execute plugin API handlers in
seconds. # TYPE mattermost_plugin_api_time histogram
mattermost_plugin_api_time_bucket{api_name="AddUserToChannel",plugin_id="com
.mattermost.plugin-incident-response",success="true",le="0.005"} 0
mattermost_plugin_api_time_bucket{api_name="AddUserToChannel",plugin_id="com
.mattermost.plugin-incident-response",success="true",le="0.01"} 0
```

Other alerts

If you want more alerts, you can set them up on any of the Grafana charts you'd like. We recommend reviewing custom metrics listed on our Performance Monitoring feature documentation.

On this page

- Mattermost performance alerting guide
  - Prerequisites
  - Configure alerts
  - Available charts
    - CPU utilization rate
    - Memory usage
    - Number of Goroutines
    - Number of API errors per second
    - Mean API request time
    - Plugin hooks
  - Other alerts

Performance monitoring metrics

Mattermost provides the following performance monitoring statistics to integrate with Prometheus and Grafana.

Custom Mattermost metrics

The following is a list of custom Mattermost metrics that can be used to monitor your system's performance:

API metrics

- `mattermost_api_time`: The total time in seconds to execute a given API handler.

Caching metrics

- `mattermost_cache_etag_hit_total`: The total number of ETag cache hits for a specific cache.
- `mattermost_cache_etag_miss_total`: The total number of ETag cache misses for an API call.
- `mattermost_cache_mem_hit_total`: The total number of memory cache hits for a specific cache.
- `mattermost_cache_mem_invalidation_total`: The total number of memory cache invalidations for a specific cache.
- `mattermost_cache_mem_miss_total`: The total number of cache misses for a specific cache.

The above metrics can be used to calculate ETag and memory cache hit rates over time.

Cluster metrics

- `mattermost_cluster_cluster_request_duration_seconds`: The total duration in seconds of the inter-node cluster requests.
- `mattermost_cluster_cluster_health_score`: A score that gives an idea of how well it is meeting the soft-real time requirements of the gossip protocol.
- `mattermost_cluster_cluster_requests_total`: The total number of inter-node requests.
- `mattermost_cluster_cluster_event_type_totals`: The total number of cluster requests sent for any type.

Database metrics

- `mattermost_db_active_users`: The total number of active users.
- `mattermost_db_cache_time`: Time to execute the cache handler.
- `mattermost_db_master_connections_total`: The total number of connections to the master database.
- `mattermost_db_read_replica_connections_total`: The total number of connections to all the read replica databases.
- `mattermost_db_search_replica_connections_total`: The total number of connections to all the search replica databases.
- `mattermost_db_store_time`: The total time in seconds to execute a given database store method.

- `mattermost_db_replica_lag_abs`: Absolute lag time based on binlog distance/transaction queue length.
- `mattermost_db_replica_lag_time`: The time taken for the replica to catch up.

Database connection metrics

- `go_sql_max_open_connections`: Maximum number of open connections to the database.
- `go_sql_open_connections`: The number of established connections both in use and idle.
- `go_sql_in_use_connections`: The number of connections currently in use.
- `go_sql_idle_connections`: The number of idle connections.
- `go_sql_wait_count_total`: The total number of connections waited for.
- `go_sql_wait_duration_seconds_total`: The total time blocked waiting for a new connection.
- `go_sql_max_idle_closed_total`: The total number of connections closed due to SetMaxIdleConns.
- `go_sql_max_idle_time_closed_total`: The total number of connections closed due to SetConnMaxIdleTime.
- `go_sql_max_lifetime_closed_total`: The total number of connections closed due to SetConnMaxLifetime.

HTTP metrics

- `mattermost_http_errors_total`: The total number of http API errors.
- `mattermost_http_requests_total`: The total number of http API requests.
- `mattermost_http_websockets_total`: The total number of websocket connections to this server.

> Note
>
> From Mattermost version v9.9, this value includes any potentially unauthenticated connections. Furthermore, this metric comes with an `origin_client` label that can be used to see the distribution of connections from different client types (i.e. web, mobile, and desktop).

Login and session metrics

- `mattermost_login_logins_fail_total` : The total number of failed logins.
- `mattermost_login_logins_total` : The total number of successful logins.

Mattermost channels metrics

- `mattermost_post_broadcasts_total` : The total number of websocket broadcasts sent because a post was created.
- `mattermost_post_emails_sent_total` : The total number of emails sent because a post was created.
- `mattermost_post_file_attachments_total` : The total number of file attachments created because a post was created.

- `mattermost_post_pushes_sent_total`: The total number of mobile push notifications sent because a post was created.
- `mattermost_post_total`: The total number of posts created.
- `mattermost_post_webhooks_total`: Total number of webhook posts created.

Process metrics

- `mattermost_process_cpu_seconds_total`: Total user and system CPU time spent in seconds.
- `mattermost_process_max_fds`: Maximum number of open file descriptors.
- `mattermost_process_open_fds`: Number of open file descriptors.
- `mattermost_process_resident_memory_bytes`: Resident memory size in bytes.
- `mattermost_process_start_time_seconds`: Start time of the process since unix epoch in seconds.
- `mattermost_process_virtual_memory_bytes`: Virtual memory size in bytes.
- `mattermost_process_virtual_memory_max_bytes`: Maximum amount of virtual memory available in bytes.

Search metrics

- `mattermost_search_posts_searches_duration_seconds`: The total duration in seconds of post searches.
- `mattermost_search_channel_index_total`: The total number of channel indexes carried out.
- `mattermost_search_file_index_total`: The total number of files indexes carried out.
- `mattermost_search_files_searches_duration_seconds`: The total duration in seconds of file searches.
- `mattermost_search_files_searches_total`: The total number of file searches carried out.
- `mattermost_search_post_index_total`: The total number of posts indexes carried out.
- `mattermost_search_posts_searches_total`: The total number of post searches carried out.
- `mattermost_search_user_index_total`: The total number of user indexes carried out.

WebSocket metrics

- `mattermost_websocket_broadcast_buffer_size`: Number of events in the websocket broadcasts buffer waiting to be processed.
- `mattermost_websocket_broadcast_buffer_users_registered`: Number of users registered in a broadcast buffer hub.
- `mattermost_websocket_broadcasts_total`: The total number of websocket broadcasts sent for any type.
- `mattermost_websocket_event_total`: Total number of websocket events.
- `mattermost_websocket_reconnects_total`: Total number of websocket reconnect attempts.

Logging metrics

- `mattermost_logging_logger_queue_used`: Number of records in log target queue.
- `mattermost_logging_logger_logged_total`: The total number of records logged.
- `mattermost_logging_logger_error_total`: The total number of logger errors.
- `mattermost_logging_logger_dropped_total`: The total number of dropped log records.
- `mattermost_logging_logger_blocked_total`: The total number of log records that were blocked/delayed.

Debugging metrics - system

- `mattermost_system_server_start_time`: The time the server started.

Use `mattermost_system_server_start_time` to dynamically add an annotation corresponding to the event.

Debugging metrics - jobs

- `mattermost_jobs_active`: Number of active jobs.

Use `mattermost_jobs_active` to display an active jobs chart.

Or, use `mattermost_jobs_active` to dynamically add a range annotation corresponding to jobs being active.

Use annotations to streamline analysis when a job is long running, such as an LDAP synchronization job.

> Note
>
> Jobs where the runtime is less than the Prometheus polling interval are unlikely to be visible because Grafana is performing range queries over the raw Prometheus timeseries data, and rendering an event each time the value changes.

Plugin metrics

- `mattermost_plugin_api_time`: Time to execute plugin API handlers in seconds.
- `mattermost_plugin_hook_time`: Time to execute plugin hook handler in seconds.
- `mattermost_plugin_multi_hook_server_time`: Time for the server to execute multiple plugin hook handlers in seconds.
- `mattermost_plugin_multi_hook_time`: Time to execute multiple plugin hook handler in seconds.

Shared metrics

- `mattermost_shared_channels_sync_collection_duration_seconds`: Duration tasks spend collecting sync data (seconds).
- `mattermost_shared_channels_sync_collection_step_duration_seconds`: Duration tasks spend in each step collecting data (seconds).
- `mattermost_shared_channels_sync_count`: Count of sync events processed for each remote.
- `mattermost_shared_channels_sync_send_duration_seconds`: Duration tasks spend sending sync data (seconds).
- `mattermost_shared_channels_sync_send_step_duration_seconds`: Duration tasks spend in each step sending data (seconds).
- `mattermost_shared_channels_task_in_queue_duration_seconds`: Duration tasks spend in queue (seconds).
- `mattermost_shared_channels_task_queue_size`: Current number of tasks in queue.

Remote cluster metrics

- `mattermost_remote_cluster_clock_skew`: An approximated value for clock skew between clusters.

- `mattermost_remote_cluster_conn_state_change_total`: Total number of connection state changes.
- `mattermost_remote_cluster_msg_errors_total`: Total number of message errors.
- `mattermost_remote_cluster_msg_received_total`: Total number of messages received from the remote cluster.
- `mattermost_remote_cluster_msg_sent_total`: Total number of messages sent to the remote cluster.
- `mattermost_remote_cluster_ping_time`: The ping roundtrip times to the remote cluster.

Notification metrics

- `mattermost_notifications_error`: Total number of errors that stop the notification flow.
- `mattermost_notifications_not_sent`: Total number of notifications the system deliberately did not send.
- `mattermost_notifications_success`: Total number of successfully sent notifications.
- `mattermost_notifications_total`: Total number of notification events.
- `mattermost_notifications_total_ack`: Total number of notification events acknowledged.
- `mattermost_notifications_unsupported`: Total number of untrackable notifications due to an unsupported app version.

Mobile app metrics

- `mattermost_mobileapp_mobile_channel_switch`: Duration of the time taken from when a user clicks on a channel name, and the full channel sreen is loaded (seconds).
- `mattermost_mobileapp_mobile_load`: Duration of the time taken from when a user opens the app and the app finally loads all relevant information (seconds).

- `mattermost_mobileapp_mobile_team_switch`: Duration of the time taken from when a user clicks on a team, and the full categories screen is loaded (seconds).

Web app metrics

- `mattermost_webapp_channel_switch`: Duration of the time taken from when a user clicks on a channel in the LHS to when posts in that channel become visible (seconds).
- `mattermost_webapp_cumulative_layout_shift`: Measure of how much a page's content shifts unexpectedly.
- `mattermost_webapp_first_contentful_paint`: Duration of how long it takes for any content to be displayed on screen to a user (seconds).
- `mattermost_webapp_global_threads_load`: Duration of the time taken from when a user clicks to open Threads in the LHS until when the global threads view becomes visible (milliseconds).
- `mattermost_webapp_interaction_to_next_paint`: Measure of how long it takes for a user to see the effects of clicking with a mouse, tapping with a touchscreen, or pressing a key on the keyboard (seconds).
- `mattermost_webapp_largest_contentful_paint`: Duration of how long it takes for large content to be displayed on screen to a user (seconds).
- `mattermost_webapp_long_tasks`: Counter of the number of times that the browser's main UI thread is blocked for more than 50ms by a single task.
- `mattermost_webapp_page_load`: The amount of time from when the browser starts loading the web app until when the web app's load event has finished (seconds).
- `mattermost_webapp_rhs_load`: Duration of the time taken from when a user clicks to open a thread in the RHS until when posts in that thread become visible (seconds).
- `mattermost_webapp_team_switch`: Duration of the time taken from when a user clicks on a team in the LHS to when posts in that team become visible (seconds).

- `mattermost_webapp_time_to_first_byte`: Duration from when a browser starts to request a page from a server until when it starts to receive data in response (seconds).

Standard Go metrics

Cloudandself-hosteddeployments

The performance monitoring feature provides standard Go metrics for HTTP server runtime profiling data and system monitoring, such as:

- `go_memstats_alloc_bytes` for memory usage
- `go_goroutines` for number of goroutines
- `go_gc_duration_seconds` for garbage collection duration
- `go_memstats_heap_objects` for object tracking on the heap

To learn how to set up runtime profiling, see the pprof package Go documentation. You can also visit the `ip:port` page for a complete list of metrics with descriptions.

> Note
>
> A Mattermost Enterprise license is required to connect to `/metrics` using HTTP.

If enabled, you can run the profiler by

```
go tool pprof http://localhost:<port>/debug/pprof/profile?seconds=<duration>
```

where you can replace `localhost` with the server name. The profiling reports are available at `<ip>:<port>`, which include:

- `/debug/pprof/profile?seconds=30` for CPU profiling
- `/debug/pprof/cmdline` for command line profiling
- `/debug/pprof/symbol` for symbol profiling
- `/debug/pprof/trace` for trace profiling

- `/debug/pprof/goroutine` for Go routine profiling
- `/debug/pprof/heap` for heap profiling
- `/debug/pprof/threadcreate` for threads profiling
- `/debug/pprof/block` for block profiling
- `/debug/pprof/allocs` for past memory allocations
- `/debug/pprof/mutex` for logs of the holders of contended mutexes

Host/system metrics

Available onEnterprise plansCloudandself-hosteddeployments

From Metrics Plugin v0.7.0, you can pull metrics from [node exporter (https://github.com/prometheus/node_exporter)](https://github.com/prometheus/node_exporter) targets to access network-related panels for Mattermost Calls.

Frequently asked questions

Why are chart labels difficult to distinguish?

The chart labels used in server filters and legends are based on the hostname of your machines. If the hostnames are similar, then it will be difficult to distinguish the labels.

You can either set more descriptive hostnames for your machines or change the display name with a `relabel_config` in [Prometheus configuration (https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config)](https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config).

On this page

- Performance monitoring metrics
  - Custom Mattermost metrics
    - API metrics
    - Caching metrics
    - Cluster metrics
    - Database metrics
    - Database connection metrics
    - HTTP metrics
    - Login and session metrics
    - Mattermost channels metrics
    - Process metrics
    - Search metrics
    - WebSocket metrics
    - Logging metrics
    - Debugging metrics - system
    - Debugging metrics - jobs
    - Plugin metrics
    - Shared metrics
    - Remote cluster metrics
    - Notification metrics
    - Mobile app metrics
    - Web app metrics

- Standard Go metrics
- Host/system metrics
- Frequently asked questions
  - Why are chart labels difficult to distinguish?

Push notification health targets

When using the [Mattermost Notification Health (https://grafana.com/grafana/dashend-user-guide/boards/21305-mattermost-notification-health)](https://grafana.com/grafana/dashend-user-guide/boards/21305-mattermost-notification-health) Grafana dashboard to track different types of notifications sent from Mattermost, we recommend adhering to the following mobile push notification health targets to ensure a performant production deployment of Mattermost.

> Note
>
> - Accessing and enabling Mattermost Notification Health Monitoring requires the feature flag `NotificationMonitoring` to be set to `true` .
> - System admins can disable notification monitoring data collection through the System Console.

Push Proxy Delivery Rate

The **Push Proxy Delivery Rate** panel indicates the percentage of push notifications that have been successfully delivered to the Push Notification Service (i.e., Apple Push Notification service (APNs) or Firebase Cloud Messaging (FCM)) out of the total number of push notifications sent.

Monitor this delivery rate to ensure that users are receiving their notifications in a timely manner. This value should always be 100%. Anything below 99% should be investigated further to identify issues with the push notification configuration, connectivity problems, or possible service interruptions with the push proxy provider.

Push Notifications Errors by Reason

The **Push Notifications Errors by Reason** panel is a critical tool for maintaining the reliability and effectiveness of the push notification system. These values provide a breakdown of the errors encountered when sending push notifications, categorized by the specific reason for each error.

This information helps admins to pinpoint common problems affecting push notification delivery, diagnose and fix issues more efficiently, detect patterns and trends, identify intermittent issues before they become more serious, and enhance the overall user experience with the Mattermost platform.

Missing profile errors are generally safe to ignore. However, if you see occurences of this error increase, verify logs for `push_proxy_send_error`.

Push Notifications Not Sent Distribution

The **Push Notifications Not Sent Distribution** panel helps admins ensure the robustness and reliability of the push notification system by providing detailed insights into the specific reasons why notifications weren't sent.

This information helps admins understand the reasons why push notifications may not be sent, (i.e., the channel is muted), identify underlying system or configuration problems that need attention, prioritize areas for troubleshooting and improvement by focusing on the most significant issues affecting notification delivery rates, and monitor any trends or spikes in push notifications not sent, resulting in proactive maintenance and quicker resolution of issues as they arise.

Push Notifications Stats

The **Push Notifications Stats** panel provides a high-level view of the number of notifications sent and acknowledged.

An acknowledged push notification indicates successful communication between the service and the device because the recipient's mobile device successfully received the push notification, and sent back a confirmation of delivery to the push notification service.

When a push notification isn't acknowledged, this indicates that the notification wasn't received, or that the device didn't send back a confirmation to the push notification service.

Closer alignment between the send and acknowledge lines in this graph indicates better performance.

Total Acked Push Notifications

The **Total Acked Push Notifications** value indicates the total number of push notifications that have been acknowledged by mobile devices. This metric helps admins understand how many push notifications sent from the Mattermost server were successfully received and acknowledged by mobile devices. It's an important indicator of the reliability and effectiveness of the push notification system, as it ensures that messages are being delivered and received as intended. Any value over 80% is considered normal.

There are several scenarios where notifications may not be acknowledged which cover the remaining ~10-20% of notifications, as follows:

Servers removed without users logging out

- When a server is removed from the mobile app, and you can no longer contact the server using the mobile app, the server never logs the user session out. Therefore, Mattermost continues to send notifications to the mobile device. These sent notifications, can't be acknowledged because Mattermost would be missing the information about the server URL.

iOS notifications are disabled

- When notifications are disabled on iOS devices, notifications aren't processed by the mobile app. Therefore, we cannot acknowledge those notifications. Mattermost tracks whether the device has notifications disabled, but this information is only updated when the app starts. So any change without re-starting the app can lead to notifications not being acknowledged.

Notifications are dropped at the Push Notification Service

- APNs and FCM can drop notifications in several circumstances, especially when the mobile device doesn't have a connection. The notifications are

marked as sent to the Push Proxy, but are never sent to the devices from the Push Notification Service. Therefore, the mobile device cannot acknowledge the notification.

Extreme battery save mode

- On Android, users can activate an extreme battery save mode. In this mode, notifications are not acknowledged.

On this page

- Push notification health targets
  - Push Proxy Delivery Rate
  - Push Notifications Errors by Reason
  - Push Notifications Not Sent Distribution
  - Push Notifications Stats
  - Total Acked Push Notifications

Redis

Redis is an open-source, in-memory data structure store used as a database, cache, message broker, and streaming engine. Mattermost uses Redis as an external cache to improve performance at scale. When properly configured, Redis can help support Mattermost installations with more than 100,000 users by providing improved performance through efficient caching.

Deployment guide

Deploying Redis with Mattermost includes the following 2 steps: setting up the Redis server, and configuring Redis in Mattermost.

Set up a Redis server

1. Download and install the latest release of [Redis 7.x (https://redis.io/download/)](https://redis.io/download/). See the Redis documentation for installation details specific to your operating system.
2. Configure Redis appropriately for your environment. Ensure that Redis is secured and accessible only by trusted systems.
3. For high availability deployments, consider setting up Redis in a cluster configuration for improved reliability and performance.

Configure Redis in Mattermost

Follow these steps to configure Mattermost to use your Redis server:

1. Go to **System Console > Environment > Cache**.
2. Set **Cache Type** to `redis` to enable the Redis-specific settings.
3. Set the Redis server connection details:

> *1. Enter **Redis Address** for the Redis server you set up earlier (e.g., `redis.example.com:6379`).*

> 2. *(Optional) Enter **Redis Password** if your Redis server requires authentication.*
> 3. *(Optional) Enter **Redis Database** to specify which Redis database to use (-1 by default, which uses Redis's default database).*

4. Save your configuration and restart the Mattermost server.

Alternatively, you can configure Redis in the `config.json` file:

```
"CacheSettings": {  "CacheType": "redis",  "RedisAddress":
"redis.example.com:6379",  "RedisPassword": "",  "RedisCachePrefix": "",
"RedisDB": -1,  "DisableClientCache": false }
```

Deploy with AWS ElastiCache

For enterprise-scale deployments, AWS ElastiCache for Redis provides a fully managed Redis service that simplifies deployment, operation, and scaling.

When deploying Mattermost with AWS ElastiCache:

1. Create an ElastiCache for Redis instance, choosing Redis OSS version 7.1 or later.
2. Based on load testing results, we recommend starting with a `cache.m7g.2xlarge` instance for deployments supporting over 100,000 users.
3. Configure your Mattermost server to connect to the ElastiCache endpoint.
4. Monitor CPU utilization closely, as Redis is single-threaded and performance bottlenecks are typically CPU-bound rather than memory-bound.

Performance considerations

When using Redis with Mattermost, consider the following performance insights:

- Current load tests indicate that memory usage of Redis is relatively low, with CPU being the primary bottleneck because Redis is single-threaded.
- When planning your Redis deployment, use CPU utilization as the primary metric for scaling decisions.

Implementation details

Mattermost uses the [rueidis (https://github.com/redis/rueidis)](https://github.com/redis/rueidis) library instead of the more popular [go-redis (https://github.com/redis/go-redis)](https://github.com/redis/go-redis), primarily because rueidis offers more performance optimizations out of the box, including:

- Automatic pipelining of concurrent queries
- Client-side caching
- Additional performance tuning options like MaxFlushDelay

The primary implementation files include:

- `platform/services/cache/provider.go` - Contains the library initialization code
- `platform/services/cache/redis.go` - Contains the cache interface implementation

Frequently asked questions (FAQ)

Do I need to use Redis?

While Redis is not required for smaller Mattermost deployments, it is highly recommended for large-scale installations. - If your installation has more than 100K users, Redis is highly recommended. Redis significantly improves performance by reducing database load and optimizing common operations. - If your installation has less than 100K users, Redis is not recommended.

It is suggested to keep an eye on the `sum(rate(mattermost_cache_mem_invalidation_total[5m]))` metric which indicates the rate of cache invalidations happening. A single cache invalidation will send a message across your HA cluster to all nodes, leading to database calls from all of them. So the metric, combined with the number of nodes in your cluster should be used to determine when is the right time to use Redis.

For example, if you have more than 10 nodes in your cluster, and the rate of invalidations goes beyond 10, then you should definitely start considering adding Redis.

Should I install Redis on the same machine as Mattermost Server?

For production deployments, we recommend running Redis on a separate machine from the Mattermost server. This separation allows for better resource allocation and improved system reliability.

What Redis metrics should I monitor?

When running Redis with Mattermost, monitor the following metrics:

- CPU utilization (primary bottleneck)
- Memory usage
- Connected clients
- Cache hit ratio
- Operation latency
- Mattermost's `mattermost_db_cache_time` Grafana metric with labels of `cache_name` and `operation`, which can be used to further monitor the performance of Redis

On this page

- Redis
  - Deployment guide
    - Set up a Redis server
    - Configure Redis in Mattermost
  - Deploy with AWS ElastiCache
    - Performance considerations
  - Implementation details
  - Frequently asked questions (FAQ)
    - Do I need to use Redis?
    - Should I install Redis on the same machine as Mattermost Server?
    - What Redis metrics should I monitor?

Scale Mattermost up to 100000 users

This page describes the Mattermost reference architecture designed for the load of up to 100000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - From Mattermost v10.4, Mattermost Enterprise customers can configure Redis (https://redis.io/) (Remote Dictionary Server) as an alternative cache backend. Using Redis can help ensure that Mattermost remains performant and efficient, even under heavy usage. See the Redis cache backend configuration settings documentation for details.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	6	16/32	c7i.4xlarge	F16s v2
RDS Writer	1	16/128	db.r7g.4xlarge	E16as v6
RDS Reader	5	16/128	db.r7g.4xlarge	E16as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based*

> *document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 100000-person team with medium usage (with a safety factor of 2x) would require between 10.56TB [1] and 52.8TB [2] of free space per annum.

[1] 100000 users * 5 MB * 12 months * 2x safety factor

[2] 100000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 100000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 15000 users

This page describes the Mattermost reference architecture designed for the load of up to 15000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, reader

> Note
>
> Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	2	4/8	c7i.xlarge	F4s v2
RDS Writer	1	4/32	db.r7g.xlarge	E4as v6
RDS Reader	1	4/32	db.r7g.xlarge	E4as v6
Elasticsearch Node	2	4/32	r6g.xlarge.search	E4ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 15000-person team with medium usage (with a safety factor of 2x) would require between 1.8TB [1] and 9TB [2] of free space per annum.

[1] 15000 users * 5 MB * 12 months * 2x safety factor

[2] 15000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 15000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 200 users

This page describes the Mattermost reference architecture designed for the load of up to 200 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Not required
- **Database Configuration**: Single

> Note
>
> Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	1	2/4	c7i.large	F2s v2
RDS Writer	1	2/16	db.r7g.large	E2as v6
RDS Reader	0	2/16	db.r7g.large	E2as v6
Elasticsearch Node	0	4/32	r6g.xlarge.search	E4ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 200-person team with medium usage (with a safety factor of 2x) would require between 12GB [1] and 60GB [2] of free space per annum.

[1] 200 users * 5 MB * 12 months * 2x safety factor

[2] 200 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Smaller deployments will need an increase in resources due to the fact the database is hosted on the same server as the Mattermost application.

On this page

- Scale Mattermost up to 200 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 2000 users

This page describes the Mattermost reference architecture designed for a minimum load of 100 concurrent users and up to 2000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, reader

> Note
>
> Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	2	2/4	c7i.large	F2s v2
RDS Writer	1	2/16	db.r7g.large	E2as v6
RDS Reader	1	2/16	db.r7g.large	E2as v6
Elasticsearch Node	2	4/32	r6g.xlarge.search	E4ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 2000-person team with medium usage (with a safety factor of 2x) would require between 120GB [1] and 600GB [2] of free space per annum.

[1] 2000 users * 5 MB * 12 months * 2x safety factor

[2] 2000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 2000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 200000 users

This page describes the Mattermost reference architecture designed for the load of up to 200000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - From Mattermost v10.4, Mattermost Enterprise customers can configure Redis (https://redis.io/) (Remote Dictionary Server) as an alternative cache backend. Using Redis can help ensure that Mattermost remains performant and efficient, even under heavy usage. See the Redis cache backend configuration settings documentation for details.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

User login scalability

Tests across all architectures were conducted at a rate of 4 user logins per second (14,400 users per hour). For this architecture, we performed additional testing at higher rates, reaching up to 30 user logins per second.

Results show that this architecture supports logging in up to 150,000 users within 1.5 hours at the higher rate. Beyond this 150,000-user mark, the only available data is at the base rate of 4 logins per second, and no conclusive performance data exists at larger rates.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	14	16/32	c7i.4xlarge	F16s v2
RDS Writer	1	16/128	db.r7g.4xlarge	E16as v6
RDS Reader	6	16/128	db.r7g.4xlarge	E16as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	4	32/128	m6in.8xlarge	D32s v6
Redis	1	8/32	cache.m7g.2xlarge	Azure Cache Redis P3

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year

- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 200000-person team with medium usage (with a safety factor of 2x) would require between 21.12TB [1] and 105.6TB [2] of free space per annum.

[1] 200000 users * 5 MB * 12 months * 2x safety factor

[2] 200000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 200000 users
  - User login scalability
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 30000 users

This page describes the Mattermost reference architecture designed for the load of up to 30000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	2	8/16	c7i.2xlarge	F8s v2
RDS Writer	1	8/64	db.r7g.2xlarge	E8as v6
RDS Reader	1	8/64	db.r7g.2xlarge	E8as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based*

> *document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 30000-person team with medium usage (with a safety factor of 2x) would require between 3TB [1] and 15TB [2] of free space per annum.

[1] 30000 users * 5 MB * 12 months * 2x safety factor

[2] 30000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 30000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 50000 users

This page describes the Mattermost reference architecture designed for the load of up to 50000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	5	8/16	c7i.2xlarge	F8s v2
RDS Writer	1	8/64	db.r7g.2xlarge	E8as v6
RDS Reader	4	8/64	db.r7g.2xlarge	E8as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based*

> *document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 50000-person team with medium usage (with a safety factor of 2x) would require between 6TB [1] and 30TB [2] of free space per annum.

[1] 50000 users * 5 MB * 12 months * 2x safety factor

[2] 50000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 50000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 80000 users

This page describes the Mattermost reference architecture designed for the load of up to 80000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	4	16/32	c7i.4xlarge	F16s v2
RDS Writer	1	16/128	db.r7g.4xlarge	E16as v6
RDS Reader	3	16/128	db.r7g.4xlarge	E16as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based*

> *document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 80000-person team with medium usage (with a safety factor of 2x) would require between 10.56TB [1] and 52.8TB [2] of free space per annum.

[1] 80000 users * 5 MB * 12 months * 2x safety factor

[2] 80000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 80000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scale Mattermost up to 90000 users

This page describes the Mattermost reference architecture designed for the load of up to 90000 concurrent users. Unsure which reference architecture to use? See the scaling for enterprise documentation for details.

- **High Availability**: Required
- **Database Configuration**: writer, multiple readers

> Note
>
> - Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.
> - While the following Elasticsearch specifications may be more than sufficient for some use cases, we have not extensively tested configurations with lower resource allocations for this user scale. If cost optimization is a priority, admins may choose to experiment with smaller configurations, but we recommend starting with the tested specifications to ensure system stability and performance. Keep in mind that under-provisioning can lead to degraded user experience and additional troubleshooting effort.

Requirements

Resource Type	Nodes	vCPU/ Memory (GiB)	AWS Instance	Azure Instance
Mattermost Application	5	16/32	c7i.4xlarge	F16s v2
RDS Writer	1	16/128	db.r7g.4xlarge	E16as v6
RDS Reader	4	16/128	db.r7g.4xlarge	E16as v6
Elasticsearch cluster	4	8/64	r6g.2xlarge.search	E8ads v6
Proxy	1	16/64	m7i.4xlarge	D16s v6

Lifetime storage

To forecast your own storage usage, begin with a Mattermost server approximately 600 MB to 800 MB in size including operating system and database, then add the multiplied product of:

- Estimated storage per user per month (see below), multiplied by 12 months in a year
- Estimated mean average number of users in a year
- A 1-2x safety factor

Estimated storage per user, per month

File usage per user varies significantly across industries. The below benchmarks are recommended:

- **Low usage teams** (1-5 MB/user/month)

> - *Primarily using text messages and links to communicate. Examples would include software development teams that heavily use web-based*

> *document creation and management tools, and therefore rarely upload files to the server.*

- **Medium usage teams** (5-25 MB/user/month)

> - *Using a mix of text messages as well as shared documents and images to communicate. Examples might include business teams that may commonly drag and drop screenshots, PDFs and Microsoft Office documents into Mattermost for sharing and review.*

- **High usage teams** (25-100 MB/user/month)

> - *Heaviest utilization comes from teams uploading a high number of large files into Mattermost on a regular basis. Examples include creative teams who share and store artwork and media with tags and commentary in a pipeline production process.*

Example

A 90000-person team with medium usage (with a safety factor of 2x) would require between 10.56TB [1] and 52.8TB [2] of free space per annum.

[1] 90000 users * 5 MB * 12 months * 2x safety factor

[2] 90000 users * 25 MB * 12 months * 2x safety factor

We strongly recommend that you review storage utilization at least quarterly to ensure adequate free space is available.

Additional considerations

Elasticsearch (https://www.elastic.co) provides enterprise-scale deployments with optimized search performance and prevents performance degradation and timeouts. Elasticsearch allows you to search large volumes of data quickly, in near real-time, by creating and managing an index of post data. Mattermost's implementation uses Elasticsearch (https://www.elastic.co) as a distributed, RESTful search engine supporting highly efficient database searches in a cluster environment. Visit the Mattermost Elasticsearch product documentation for deployment and configuration details.

Performance monitoring support enables a Mattermost server to track system health for large Enterprise deployments through integrations with Prometheus (https://prometheus.io/) and Grafana (https://grafana.com/). These integrations support data collection from several Mattermost servers, which is particularly useful if you're running Mattermost in high availability mode. Once you're tracking system health, you can set up performance alerts on your Grafana dashboard. Visit the Mattermost Performance Monitoring product documentation for installation details.

On this page

- Scale Mattermost up to 90000 users
  - Requirements
  - Lifetime storage
    - Estimated storage per user, per month
    - Example
  - Additional considerations

Scaling for Enterprise

Mattermost is designed to scale from small teams hosted on a single server to large enterprises running in cluster-based, highly available deployment configurations.

- Mattermost supports any 64-bit x86 processor architecture
- **Databases supported**: PostgreSQL, Amazon RDS for PostgreSQL
- **Storage supported**: Amazon S3 or a local filestore

Server requirements vary based on usage and we highly recommend that you run a pilot before an enterprise-wide deployment in order to estimate full scale usage based on your specific organizational needs.

Backing storage

Review detailed write and read storage benchmark results for supported storage options including local file system (EBS, gp3), network file system (EFS), and object storage (S3) to make informed decisions based on your use case and infrastructure needs.

Enterprise search

We highly recommend a dedicated server for large enterprise deployments to run highly efficient database searches in a cluster environment.

For deployments with over 5 million posts, Enterprise search using Elasticsearch or AWS OpenSearch Service is required for optimized search performance, dedicated indexing and usage resourcing via cluster support without performance degradation and timeouts, resulting in faster, more predicable search results.

High availability

A high availability cluster-based deployment enables a Mattermost system to maintain service during outages and hardware failures through the use of redundant infrastructure.

Redis

Redis is an in-memory data structure store that can be used as a database, cache, and message broker. Mattermost uses Redis as an external cache to improve performance at scale. When properly configured, Redis can help support Mattermost installations with more than 100,000 users by providing improved performance through efficient caching.

Available reference architectures

The following reference architectures are available as recommended starting points for your self-hosted Mattermost deployment, where user counts refer to the number of concurrent users for a given deployment. The number of concurrent numbers is commonly lower than the total number of user accounts.

- Scale up to 200 users - Learn how to scale Mattermost to up to 200 users.
- Scale up to 2000 users - Learn how to scale Mattermost to up to 2000 users.
- Scale up to 15000 users - Learn how to scale Mattermost to up to 15000 users.
- Scale up to 30000 users - Learn how to scale Mattermost to up to 30000 users.
- Scale up to 50000 users - Learn how to scale Mattermost to up to 50000 users.
- Scale up to 80000 users - Learn how to scale Mattermost to up to 80000 users.
- Scale up to 90000 users - Learn how to scale Mattermost to up to 90000 users.

- Scale up to 100000 users - Learn how to scale Mattermost to up to 100000 users.
- Scale up to 200000 users - Learn how to scale Mattermost to up to 200000 users.

Important

Due to constraints in testing, the proxy instance specifications were fixed for all the scaling tests from which we derived these reference architectures. This was done to avoid a combinatorial explosion of variables in tests, but it resulted in minor gaps in our understandings of certain aspects of the reference architectures. In particular, the proxy instance is overspecified for the smaller user counts.

Testing methodology and updates

All tests were executed with the custom load test tool built by the Mattermost development teams to determine supported users for each deployment size. Over time, this guide will be updated with new deployment sizes, deployment architectures, and newer versions of the Mattermost Server will be tested using an ESR.

At a high level, each deployment size was fixed (Mattermost server node count/sizing, database reader/writer count/sizing), and unbounded tests were used to report the maximum numbers of concurrent users the deployment can support. Each test included populated PostgreSQL v14 databases and a post table history of 100 million posts, ~200000 users, 20 teams, and ~720000 channels to provide a test simulation of a production Mattermost deployment.

Tests were defined by configuration of the actions executed by each simulated user (and the frequency of these actions) where the coordinator metrics define a health system under load. Tests were performed using the Mattermost v9.5 Extended Support Release (ESR). Job servers weren’t used. All tests with more than a single app node had an NGINX proxy running in front of them.

For the last test of 200K users, further infrastructure changes were made. Elasticsearch nodes were added. A Redis instance was added, and multiple NGINX proxies were used to distribute traffic evenly across all nodes in the cluster. More details can be found on the scale to 200000 users documentation page.

Full testing methodology, configuration, and setup is available, incluidng a fixed database dump with 100 million posts (https://us-east-1.console.aws.amazon.com/backup/home?region=us-east-1#/resources/arn%3Aaws%3Ards%3Aus-east-1%3A729462591288%3Acluster%3Adb-pg-100m-posts-v9-5-5). Visit the Mattermost Community (https://community.mattermost.com/) and join the Developers: Performance channel (https://community.mattermost.com/core/channels/developers-performance) for details.

The performance specifications and user capacities provided in these reference architectures are based on extensive testing conducted in AWS environments.

For Azure deployments, the instance specifications have been extrapolated using equivalent configurations, drawing upon AWS performance data and industry expertise. While these configurations are designed to meet the stated requirements, we have not yet conducted formal testing in Azure environments to validate performance under load. Azure Blob Storage support is in active development, and not yet natively supported by Mattermost.

Mattermost load testing tools

Mattermost provides a set of tools written in Go to help profiling Mattermost under heavy load, simulating real-world usage of a server installation at scale. The Mattermost Load Test Tool (https://github.com/mattermost/mattermost-load-test-ng) estimates the maximum number of concurrently active users the target system supports, and enables you to control the load to generate.

Visit the Mattermost Load Test Tool (https://github.com/mattermost/mattermost-load-test-ng/tree/master/docs) documentation on GitHub for details on getting started with the tools, and visit the Go documentation for code-specific documentation details.

Important

- The Mattermost Load Test Tool was designed by and is used by our performance engineers to compare and benchmark the performance of the service from month to month to prepare for new releases. It's also used extensively in developing our recommended hardware sizing.
- We recommend deploying Prometheus and Grafana with our dashboards for ongoing monitoring and scale guidance.

- If you encounter performance concerns, we recommend collecting performance metrics and sharing them with us as a first troubleshooting step.

On this page

- Scaling for Enterprise
  - Backing storage
  - Enterprise search
  - High availability
  - Redis
  - Available reference architectures
  - Testing methodology and updates
  - Mattermost load testing tools

Administrator onboarding tasks

This document provides instructions for common administrator tasks, including some recommendations on tasks to prepare your Mattermost deployment to onboard users.

Getting started tasks

1. Once you've installed and deployed Mattermost, ensure all configuration settings are appropriately set under **System Console > Environment** including:

> - *Web server*
> - *Database*
> - *File storage*
> - *SMTP*
> - *Push Notification server*

These settings can also be set in the `config.json` file. Please see our configuration settings documentation for a full listing of all configuration settings.

2. Adjust settings under **System Console > Site Configuration** to brand and customize how users will interact with the site. Be sure to update the Support Email and Help Link in Mattermost under **System Console > Site Configuration > Customization** to provide your users a resource for password resets or questions on their Mattermost account.

> - *The Support email is used on email notifications and during tutorial for users to ask support questions.*
> - *The Help Link is on the Mattermost login page, sign-up pages, and Main Menu and can be used to to link to your help desk ticketing system.*

These settings can also be set in the `config.json` file. Please see our configuration settings documentation for a full listing of all configuration settings.

3. Begin to onboard users by enabling account creation or by connecting an authentication service to assist with user provisioning.

- Users can be pre-provisioned with migration and bulk loading data processes based on prior collaboration systems. Please see our migration guide and bulk loading documentation for additional details.
- AD/LDAP authentication and SAML authentication are available for some subscription plans, providing identity management, single sign-on, and automatic account provisioning.

If your organization requires more structure and project management artifacts for the implementation of Mattermost, please see our Enterprise roll out checklist.

Important administration notes

DO NOT manipulate the Mattermost database

- In particular, DO NOT manually delete data from the database directly. Mattermost is designed as a continuous archive and cannot be supported after manual manipulation.
- If you need to permanently delete a team or user, use the mmctl user delete command or the mmctl user deletall command.

Common tasks

Creating System admin account from the command line

- If the System admin leaves the organization or is otherwise unavailable, you can use the mmctl roles commands to assign the *system_admin* role to an existing user.
- The user needs to log out and log back in before the *system_admin* role is applied.

Migrating to AD/LDAP or SAML from email-based authentication

- If you have a Mattermost Enterprise or Professional plan, you can migrate from email authentication to Active Directory/LDAP or to SAML Single Sign-on. To set up Active Directory/LDAP, see Active Directory/LDAP Setup. To set up SAML Single Sign-on, see SAML Single-Sign-On.
- After the new authentication method is enabled, existing users cannot use the new method until they go to **Settings > Security > Sign-in method** and select **Switch to using AD/LDAP** or **Switch to using SAML Single Sign-on**. After they have switched, they can no longer use their email and password to log in.

Deactivating a user

- System admins can go to **System Console > Users** for a list of all users on the server. The list can be searched and filtered to make finding the user easier. Click the user's role and in the menu that opens, click **Deactivate**.
- To preserve audit history, users are typically never deleted from the system. If permanently deleting a user is necessary (e.g. for the purposes of GDPR (https://gdpr-info.eu/)), an mmctl command can be used to do so.
- Note that AD/LDAP user accounts cannot be deactivated from Mattermost; they must be deactivated from your Active Directory.

Checking for a valid license in Enterprise Edition without logging in

- Open the log file `mattermost.log`. It's usually in the `mattermost/logs/` directory but might be elsewhere on your system. Find the last occurrence of a log entry that starts with the text `[INFO] License key`. If the license key is valid, the complete line should be similar to the following example:

```
[2017/05/19 16:51:40 UTC] [INFO] License key valid unlocking enterprise
features.
```

User experience optimizations

We highly recommend the following best practices, configuration options, and features for an optimal Mattermost user experience.

1. Upgrade your Mattermost server

When you upgrade your Mattermost server frequently, your users can access new features, improved user experiences, bug fixes, security fixes, and mobile app compatibility.

Mattermost releases regular updates to Mattermost Team Edition (https://mattermost.com/) and Mattermost Enterprise Edition (https://mattermost.com/pricing/). See the release life cycle documentation for component life cycle details details.

Upgrading your Mattermost server only takes a few minutes. See the Upgrade Guide for step-by-step instructions.

2. Install plugins

You can enable plugins and integrations to connect your team's workflows and toolsets into Mattermost. Plugins and integrations customize and extend the Mattermost platform.

Install and manage plugins

To enable and manage plugins, go to **System Console > Plugins**. Next, install plugins from **Product menu > Marketplace**. See the Marketplace (https://developers.mattermost.com/integrate/admin-guide/admin-plugins-beta/#marketplace) documentation for details.

Consider installing, configuring, and enabling the following community integrations for your users:

- Create polls with Matterpoll (https://mattermost.com/marketplace/matterpoll/).
- Share GIFs with GIF Commands (https://mattermost.com/marketplace/giphy-plugin/).
- Create and share memes with Memes (https://mattermost.com/marketplace/memes-plugin/).

- Set personal reminders with Remind (https://mattermost.com/marketplace/remind-plugin/).
- Create and share to do items with Todo (https://github.com/mattermost/mattermost-plugin-todo).
- Customize welcome messages for new users with WelcomeBot (https://mattermost.com/marketplace/welcomebot-plugin/).

Explore all plugins and integrations available in the Mattermost Marketplace (https://mattermost.com/marketplace/).

Enable and manage integrations

To enable integrations such as webhooks, slash commands, OAuth2.0, and bots, to go **System Console > Integrations**. More information on these integrations can be found here (https://developers.mattermost.com/integrate/other-integrations/).

3. Enable automatically extended sessions

Keep your desktop and mobile users logged in and extend user sessions automatically (https://mattermost.com/blog/session-expiry-experience/) by setting **System Console > Sessions > Extend session length with activity** to **true**. See the Extend session length with activity configuration settings documentation for details.

4. Enable full content push notifications

Enable push notifications on mobile devices to deliver messages in real time by setting **System Console > Push Notification Server > Enable Push Notifications** to **Use TPNS**. See the Push notification server configuration settings documentation for details.

Enable full content push notifications, including the sender's name, the channel name, and the message text, by setting **System Console > Notifications > Push Notification Contents** to **Full message contents**. See the Push notification contents configuration settings documentation for details.

> Note
>
> - Mattermost subscription plans allow you to enable HPNS that includes production-level uptime SLAs.
> - Mattermost Enterprise customers can enable ID-Only push notifications so push notification content is not passed through Apple Push Notification Service (APNS) or Google Firebase Cloud Messaging (FCM) before reaching the device. The ID-only push notification setting offers a high level of privacy (https://mattermost.com/blog/id-only-push-notifications/) while allowing team members to benefit from mobile push notifications.

5. Enable custom emoji

Emojis enable users to express concepts such as emotions and physical gestures in messages. Enable the emoji picker by setting **System Console > Emoji > Enable Emoji Picker** to **true**. See the Enable emoji picker configuration settings documentation for details.

Empower users to create and share their own custom emojis by setting **System Console > Emoji > Enable Custom Emoji** to **true**. See the Enable custom emoji configuration settings documentation for details.

6. Enable GIF picker

GIFs are animated images that can make messaging more fun and engaging. Enable users to access the Mattermost GIF picker from the message draft area by setting **System Console > GIF (Beta) > Enable GIF Picker** to **true**. See the Enable GIF picker configuration settings documentation for details.

7. Enable link previews

Link previews provide a visual glimpse of relevant content for links shared in messages. Enable link previews by setting **System Console > Posts > Enable Link Previews** to **true**. See the Enable link previews configuration settings documentation for details.

8. Enable batched email notifications

Email notifications can be batched together so users don't get overwhelmed with too many emails.

Enable email notifications first by setting **System Console > Notifications > Enable Email Notifications** to **true**. See the Enable email notifications configuration settings documentation for details. Note that email notifications require an SMTP email server to be configured.

Then, enable batched email notifications by setting **System Console > Notifications > Enable Email Batching** to **true**. See the Enable email batching configuration settings documentation for details. Note that email batching is not available if you are running your deployment in High Availability.

9. Enable Elasticsearch

Mattermost Enterprise customers can enable enterprise search for optimized search performance at enterprise-scale. Both Elasticsearch and AWS OpenSearch solve many known issues with full text database search, such as dots, dashes, and email addresses returning unexpected results.

Enable Elasticsearch by setting **System Console > Elasticsearch > Enable Indexing** to **true**. See the Elasticsearch configuration settings documentation for details. Enabling Elasticsearch requires setting up an Elasticsearch server.

On this page

- Administrator onboarding tasks
  - Getting started tasks
  - Important administration notes
  - Common tasks
  - User experience optimizations

Communicate scheduled maintenance best practices

Performing scheduled maintenance on a Mattermost server with 1,000 or more users requires advanced planning and a clear communication strategy to ensure minimal disruption and maximum transparency.

This guide provides best practices for notifying users via email and Mattermost channels, updating the load balancer's error message, and configuring a dismissable banner to inform users of the upcoming maintenance.

Communication strategy

A well-defined communication strategy is essential for informing users before, during, and after maintenance. The key components of this strategy are:

- Define a clear maintenance window in which the self-hosted Mattermost server will be unavailable.

  > - *Mattermost Cloud deployments have predefined service windows scheduled from 8:00-10:00 UTC on Saturdays only (when applicable) unless an exception has been made and communicated to impacted customers.*

- Email notifications: Send structured and consistent emails to users at intervals of 7 days, 3 days, and 1 day before the scheduled maintenance window.
- Channel-based reminders: Send messages similar to the emails in relevant Mattermost channels at the same intervals as the email notifications.
- Mattermost Banner notification: Set a system-wide notification to display at the top of the Mattermost instance ahead of the maintenance window and outage.
- Display a load balancer message: Update the load balancer to show a maintenance message during the scheduled maintenance window of downtime.

Notification templates

Email Templates

7-Day notice email

Email subject: Scheduled Maintenance Notification: [Date and Time]

```
Dear Mattermost Users, This is a notification that our Mattermost server
will undergo scheduled maintenance on [Date] from [Start Time] to [End Time]
[Time Zone]. During this time, the Mattermost instance will be unavailable.
We apologize for any inconvenience this may cause and appreciate your
understanding as we work to improve our service. If you have any questions
or concerns, please contact our support team at [Support Email]. Thank you
for your cooperation. Best regards, [Your Name] [Your Position]
```

3-Day notice email

Email subject: Reminder: Scheduled Maintenance on [Date and Time]

```
Dear Mattermost Users, This is a reminder that our Mattermost server will
undergo scheduled maintenance on [Date] from [Start Time] to [End Time]
[Time Zone]. The Mattermost instance will be unavailable during this period.
If you have any questions or concerns, please contact our support team at
[Support Email]. Thank you for your cooperation. Best regards, [Your Name]
[Your Position]
```

1-Day notice email

Email subject: Final Reminder: Scheduled Maintenance Tomorrow on [Date and Time]

```
Dear Mattermost Users, This is a final reminder that our Mattermost server
will undergo scheduled maintenance tomorrow, [Date], from [Start Time] to
[End Time] [Time Zone]. The Mattermost instance will be unavailable during
this period. If you have any questions or concerns, please contact our
support team at [Support Email]. Thank you for your cooperation. Best
regards, [Your Name] [Your Position]
```

Channel reminder templates

7-Day channel reminder

```
@all Please be advised that our Mattermost server will undergo scheduled
maintenance on [Date], from [Start Time] to [End Time] [Time Zone]. The
instance will be unavailable during this time. We appreciate your
understanding.
```

3-Day channel reminder

```
@all This is a reminder that our Mattermost server will undergo scheduled
maintenance on [Date], from [Start Time] to [End Time] [Time Zone]. Please
plan accordingly.
```

1-Day channel reminder

```
@all Final reminder: Our Mattermost server will undergo scheduled
maintenance tomorrow, [Date], from [Start Time] to [End Time] [Time Zone].
Thank you for your cooperation.
```

Banner notification

Sample message:

```
Heads up! Scheduled maintenance is planned for [Date], between [Start Time]
and [End Time] [Time Zone]. The Mattermost instance will be unavailable
during this time.
```

Users can dismiss the banner until they log in again, or until you update the banner.

Display load balancer message

Configure the load balancer to display a notification message during the scheduled maintenance window of downtime.

1. Identify the load balancer you are using (e.g., AWS, HAProxy).

2. Edit configuration. Adjust paths and configurations according to your specific environment.

> - *For AWS, navigate to the Load Balancer configurations in the EC2 console.*
> - *For HAProxy, edit the* `haproxy.cfg` *file.*

Important

We strongly recommend adding headers, where needed in your infrastructure, to avoid outdated information being communicated to your users during and following server maintenance.

See the sample message HTML below to use as a starting point:

```
HTML Template <!DOCTYPE html> <html> <head> <title>Maintenance in Progress</
title> <style>  body {  text-align: center;  padding: 50px;  font-family:
"Arial", sans-serif;  background-color: #f2f2f2;  }  .container {  margin:
auto;  width: 50%;  padding: 20px;  background-color: #fff;  border-radius:
8px;  box-shadow: 0px 0px 10px 0px #0000001a;  }  </style> </head> <body>
<div class="container"> <h1>Maintenance in Progress</h1> <p>Our Mattermost
server is currently undergoing scheduled maintenance.</p> <p>Estimated
downtime: [Start Time] to [End Time] [Time Zone]</p>
<p>We apologize for any inconvenience and thank you for your
understanding.</p> <p>If you have any questions, please contact our support
team at <a href="mailto:[Support Email]">[Support Email]</a>.</p> </div> </
body> </html>
```

On this page

- Communicate scheduled maintenance best practices
  - Communication strategy
  - Notification templates
    - Email Templates
      - 7-Day notice email
      - 3-Day notice email
      - 1-Day notice email
    - Channel reminder templates
      - 7-Day channel reminder
      - 3-Day channel reminder
      - 1-Day channel reminder

- Banner notification
- Display load balancer message

Downgrade Mattermost Server

In most cases you can downgrade Mattermost Server using the same steps as Upgrade Mattermost Server. Server binaries can be found in the Mattermost server version archive documentation.

Important

- We don't recommend downgrading more than one major version back from your current installation.
- We strongly recommend testing the downgrade in a staging environment first to identify any potential issues.
- Ensure that your plugins and integrations are compatible with the downgraded version you're moving to.

Prepare for downgrade

Before downgrading the Mattermost server, we strongly recommend the following preparation steps.

1. Back up your data: Ensure you have a full backup of your database and Mattermost application files. This is crucial in case you need to revert any changes.

> 1. *Back up your database using your organization's standard procedures for backing up the database.*
> 2. *Back up your application by copying into an archive folder (e.g.* `mattermost-back-YYYY-MM-DD-HH-mm`*). Ensure to copy your Mattermost configuration files and any other necessary application files.*

2. Carefully review the Mattermost changelog for the version you are downgrading to in order to understand any potential issues or incompatibilities.
3. Verify the current schema version of your database using the mattermost db version –all command. Also, if you aren't sure about the target schema,

you can verify the target schema version (i.e., applied migrations) by checking the public [GitHub repository (https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/migrations.list)](https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/migrations.list) (Select the tag for desired version).

Perform the downgrade

1. Stop the Mattermost service to ensure that no data is being written to the database during the downgrade process.
2. If the database schema has changed between versions, you must downgrade the schema. Use the newer mattermost binary to perform the downgrade using the mattermost db downgrade command. For example: `mattermost db downgrade 128,127,126`

> Tip
>
> You can review downgrade changes before committing them if you have used the `--save-plan` option while upgrading Mattermost. It has both forwards and backwards SQL scripts. This option allows you to avoid specifying migrations to be downgraded, and allows you to use older versions of Mattermost to perform the downgrade. For example: `mattermost db downgrade migration_plan_128_127.json`.

3. There may be changes in configuration settings between versions. Revert any necessary configuration changes in the `config.json` file to match the downgraded version's expectations and support.

After the downgrade

Replace the current Mattermost application binary with the version you want to downgrade to. Make sure to use the binary of the target version. The newer Mattermost binary contains the downgrade SQL for the migrations to be rolled back. The newer binary version is used to perform the downgrade; then you start using the application binary of the version you want to downgrade to.

1. Restart the Mattermost Server after completing the downgrade.

2. Check the logs and test the application to ensure that everything is functioning correctly.
3. Inform your users about the downgrade and any potential changes they might experience.

On this page

- Downgrade Mattermost Server
  - Prepare for downgrade
  - Perform the downgrade
  - After the downgrade

Enterprise install and upgrade

Mattermost Enterprise Edition is free to use in "team mode" without Enterprise features enabled. Mattermost offers two subscription plans, including Enterprise and Professional. You can compare their features online (https://mattermost.com/pricing/).

Once you've downloaded and installed Mattermost, start an Enterprise trial via **Product menu > System Console > Edition and License > Start trial**. View pricing and feature information online, and talk to a Mattermost Expert (https://mattermost.com/contact-sales/) to purchase a license key.

Installing Enterprise Edition

See the Deploy Mattermost page to install Mattermost Enterprise Edition.

Upgrading to Mattermost Enterprise Edition

Checking your Mattermost edition and version

To check your edition and version from the web or desktop interface, open **Product menu > About Mattermost**.

- **Mattermost Enterprise Edition** indicates you can apply a license key to enable Enterprise features.
- **Mattermost Team Edition** indicates you're using the open source version and need to upgrade (per upgrade instructions below) before a license key can be applied.

Upgrading to Enterprise Edition from Mattermost Team Edition

The open source Mattermost Team Edition is functionally identical to the commercial Mattermost Enterprise Edition in its free "team mode", but there is no ability to unlock Enterprise features.

To enable unlocking Enterprise features, replace the Mattermost Team Edition binary with a Mattermost Enterprise Edition binary via an upgrade procedure.

If you'd like to back up Mattermost prior to upgrading, the migration guide outlines the process required to back up and restore your database.

Upgrading to Enterprise Edition in GitLab Omnibus

GitLab Omnibus runs the open source Mattermost Team Edition. To upgrade to Mattermost Enterprise Edition, follow these steps:

1. Disable the built-in Mattermost instance on GitLab Omnibus:

- Go to `/etc/gitlab/gitlab.rb` and set the following line to false:

```
mattermost['enable'] = false
```

- Then run the following command to apply the updated setting:

```
sudo gitlab-ctl reconfigure
```

2. Install Mattermost using one of the guides above.

3. Migrate the database used by GitLab Mattermost for your new Enterprise Edition instance.

4. (Optional) Set up [GitLab slash command integration (https://docs.gitlab.com/ee/user/project/integrations/mattermost_slash_commands.html)](https://docs.gitlab.com/ee/user/project/integrations/mattermost_slash_commands.html) with your Mattermost instance.

If you need to migrate Team Edition prior to install, follow the migration guide.

Upgrading Enterprise Edition to a newer version

Upgrade a previous version of Mattermost Enterprise Edition to a later version by following the upgrade procedure.

Converting Team Edition to Enterprise Edition

If you're running a Linux system with x86-64 architecture, you can convert Team Edition to Enterprise in the System Console using a built-in conversion utility. This tool is recommended if you'd like to run an Enterprise trial or want to upgrade to Enterprise or Professional on standalone servers.

> Note
>
> - If you're using Mattermost in a managed environment, such as GitLab Omnibus, and want to start an Enterprise Edition trial, you can use this tool. However if you plan to upgrade permanently and scale your production environment, we strongly recommend installing a new standalone server and following the appropriate migration process.
> - If you're using a modified version of Mattermost, using this tool will overwrite your changes and replace them with the official Enterprise Edition binary.
> - For versions prior to v5.27, please follow these upgrade instructions.

Navigate to **Product menu > System Console > Edition and License** and select **Upgrade to Enterprise Edition**.

During the upgrade process, the Mattermost Enterprise Edition binary file that matches your current server version is downloaded, decompressed, and extracted. The Team Edition binary is then replaced by the Enterprise Edition version. Once this process is complete, you're prompted to restart your server.

The Mattermost version listed in **Product menu > System Console > Edition and License** will change from **Team Edition** to **Enterprise Edition**, and you can now activate an Enterprise Edition trial or upload a license.

Permissions and limitations

If you're using a package manager, such as GitLab Omnibus, to manage your Mattermost installation the Mattermost system user won't have sufficient permissions to perform the upgrade. If this is the case, you'll need to change the file permissions manually.

Changing the permissions in this way doesn't affect your Mattermost deployment or impact any data. The permission change is done solely for the upgrade.

To change the permissions using the command line on the Mattermost server, you need access to the command line tool as *mattermost* user. Open the command line tool on the Mattermost server and `cd` to the Mattermost installation directory. Run the following commands (replacing `<PathToBinaryFile>` with the appropriate path - typically `/opt/mattermost/bin/mattermost`) to change the ownership of the binary file to *mattermost* user and grant write access:

```
chown mattermost <PathToBinaryFile> chmod +w <PathToBinaryFile>
```

In the Mattermost System Console, retry the upgrade. When the upgrade is complete, return to the command prompt on the Mattermost server and run the following command to restore the file permissions, replacing `<OriginalFileOwner>` with the appropriate value:

```
chown <OriginalFileOwner> <PathToBinaryFile> chmod -w <PathToBinaryFile>
```

Note that automated updates or actions performed by other system admins after the conversion can overwrite the changes, possibly reverting the system to Team Edition. This includes the `run gitlab-ctl configure` command to update Mattermost within GitLab Omnibus. If this occurs, you won't be able to use Enterprise Edition features until converting the server back, but none of your data will be affected.

Troubleshooting

Mattermost has reverted to Team Edition

If you converted Team Edition to Enterprise Edition on a managed deployment and then upgraded, the upgrade will have overwritten Enterprise Edition with the latest version of Team Edition.

You can convert to Enterprise Edition again by following the steps above. If you plan to use Mattermost Enterprise Edition permanently, we recommend migrating your server to a self-hosted one.

The manual process reset my file permissions

If you manually changed your file permissions, you can change them back.

1. Open the command line tool on the Mattermost server.
2. `cd` to the Mattermost installation directory (typically `/opt/mattermost/bin/mattermost`).
3. Enter:

```
chown <OriginalFileOwner> <PathToBinaryFile> chmod -w <PathToBinaryFile>
```

File permissions error

If your Mattermost deployment is part of a managed package you may receive file permissions errors and the upgrade will fail. You can edit the permissions settings manually:

1. Open the command line tool on the Mattermost server.
2. `cd` to the Mattermost installation directory (typically `/opt/mattermost/bin/mattermost`).
3. Enter:

```
chown <OriginalFileOwner> <PathToBinaryFile> chmod -w <PathToBinaryFile>
```

Incompatible system architecture

This tool is currently only compatible with Linux systems using x86-64 architecture. If you’re running Mattermost on a different architecture, please follow the manual upgrade process.

Can’t retrieve Enterprise Edition binary file

If the upgrade fails due to file retrieval failure, unavailable binary, or connectivity error, please check your proxy settings and try again. If the problem persists, follow the manual upgrade process instead.

On this page

- Enterprise install and upgrade
  - Installing Enterprise Edition
  - Upgrading to Mattermost Enterprise Edition
    - Checking your Mattermost edition and version
    - Upgrading to Enterprise Edition from Mattermost Team Edition
    - Upgrading to Enterprise Edition in GitLab Omnibus
    - Upgrading Enterprise Edition to a newer version
  - Converting Team Edition to Enterprise Edition
    - Permissions and limitations
    - Troubleshooting
      - Mattermost has reverted to Team Edition
      - The manual process reset my file permissions
      - File permissions error
      - Incompatible system architecture
      - Can’t retrieve Enterprise Edition binary file

Enterprise roll out checklist

This checklist is intended to serve as a guide to Enterprises who are rolling out Mattermost to thousands of users.

Checklist overview

Prepare for the roll out

- 1. Define the roll out project
- 2. Validate essential security and compliance requirements
- 3. Create development, staging, and production environments
- 4. Configure and customize your Mattermost site
- 5. Test production performance and redundancy

Roll out Mattermost

- 1. Define your team and channel strategy
- 2. Enable key integrations
- 3. Prepare for user onboarding
- 4. Deploy client apps
- 5. Roll out to groups of users
- 6. Drive adoption

Review the roll out

- 1. Review project charter success metrics
- 2. Review and analyze usage
- 3. Analyze system performance

- 4. Harden security
- 5. Perform maintenance tasks

Checklist details

Prepare for the roll out

Much of the preparation work is focused on ensuring the environment is deployed and secured prior to onboarding users.

1. Define the roll out project

- Define key stakeholders and project team members

> - *Example project team members: Project Manager, Network Administrator, Database Administrator, Corporate Directory Administrator, Security & Compliance Administrator(s), User Support, User Champions, User Trainers*

- Define use cases and requirements for teams, their workflows and their integrations

> - *Resource: https://mattermost.com/blog/27-things-enterprises-can-learn-startups-increase-productivity/*

- Define success criteria, goals and metrics to measure success
- Create a Project Charter to document goals, tasks, deliverables, and decisions

> - *Get buy-in from project team members and key stakeholders on the project charter*

2. Validate essential security and compliance requirements

- Review Mattermost security features

> - *Resource: https://docs.mattermost.com/about/security.html*

- Determine monitoring requirements

> - *Database, network, storage, log integrity*
> - *Identify fields for log management tools (e.g. Splunk Enterprise event data)*

- Determine environment access policies

> - *Network access, physical access, group controlled access*

- Determine encryption policies

> - *Resource: https://docs.mattermost.com/deployment-guide/encryption-options.html*
> - *Resource: https://docs.mattermost.com/install/transport-encryption.html*

- Determine system administration access policies

> - *Identify the list of users or groups who need administrative access for Mattermost System Console, Command Line Tools, and API privileges*

- Define and configure authentication policies

> - *Resource: https://docs.mattermost.com/about/corporate-directory-integration.html*

- Determine requirements for multi-factor authentication

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/multi-factor-authentication.html*

- Configure and test SSO or Corporate Directory integration (SAML or AD/LDAP)

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/sso-saml.html*

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/ad-ldap.html*

- Define your mobile usage policy

> - *Resource: https://docs.mattermost.com/deployment-guide/mobile-overview.html*

- Evaluate external network access requirements

> - *The [Mattermost Marketplace (https://mattermost.com/marketplace)](https://mattermost.com/marketplace) is a service hosted by Mattermost that functions as a central place to store the current versions of available Mattermost integrations. See Enable Remote Marketplace documentation for details about required external network access.*
> - *Mattermost supports external GIF providers. See GIF Commands configuration documentation for details about required external network access.*

3. Create development, staging, and production environments

- Finalize production environment design basing hardware on expected usage and requirements for high availability

> - *Resource: https://docs.mattermost.com/deployment-guide/application-architecture.html*
> - *Resource: https://docs.mattermost.com/deployment-guide/index.html*
> - *Resource: https://docs.mattermost.com/administration-guide/scale/scaling-for-enterprise.html*
> - *Resource: https://docs.mattermost.com/administration-guide/scale/high-availability-cluster-based-deployment.html*

- Create development and staging environments

> - *Recommend using to test early configurations for database, authentication, file storage, Elasticsearch, prior to setting up high availability and load balancing*

- *Recommend configuring staging to be an identical replication of your production environment*

- Create a production environment

  - *Install Mattermost*

    - *Install the number of nodes based on your high availability requirements outlined in your production environment design*
    - *Recommendation: Use Kubernetes and the Mattermost Operator, with external supported external database and file storage solutions. This will also provide blue/green deployment, rolling upgrades, and canary builds*

      - *Resource: https://docs.mattermost.com/deployment-guide/server/deploy-kubernetes.html*

  - *Install and configure database*

    - *Install the number of read and search replicas based on your high availability requirements outlined in your production environment design*

      - *Resource: https://docs.mattermost.com/deployment-guide/server/server-architecture.html#database-with-virtual-ips*

    - *(Optional) Set up configuration management via the database instead of a config file for high available environments*

      - *Resource: https://docs.mattermost.com/administration-guide/configure/configuration-in-your-database.html*

  - *Install and configure File Storage*

    - *Resource: https://docs.mattermost.com/deployment-guide/server/preparations.html#file-storage-preparation*

  - *Install and configure proxy or load balancers*

    - *Note: If running Kubernetes and the Mattermost Operator, proxies will be created automatically.*

- *Add SSL Cert*
  - *Resource: https://docs.mattermost.com/administration-guide/onboard/ssl-client-certificate.html*
  - *Resource: https://docs.mattermost.com/administration-guide/scale/high-availability-cluster-based-deployment.html#proxy-server-configuration*
- *(Optional) Set up certificate-based authentication (CBA) for user or device-based authentication with a digital certificate*
  - *Resource: https://docs.mattermost.com/administration-guide/onboard/certificate-based-authentication.html*
- *Configure SMTP for email notifications*
  - *Resource: https://docs.mattermost.com/install/smtp-email-setup.html*
- *Set up Elasticsearch (highly recommended if your organization anticipates over two million posts)*
  - *Resource: https://docs.mattermost.com/administration-guide/scale/elasticsearch.html*

- Document network configuration
  - *Example: https://docs.mattermost.com/getting-started/architecture-overview.html#reference-architectures*

4. Configure and customize your Mattermost site

- Login to Mattermost and access the System Console to connect your environment to Mattermost
  - *Resource: https://docs.mattermost.com/administration-guide/configure/configuration-settings.html#environment-variables*
  - *Upload your valid Enterprise License under Edition and License*
  - *Ensure site URL is set appropriately for your production, dev and staging environments*

> - *Add your database configuration to **System Console > Environment > Database***
> - *Add your Elasticsearch or AWS OpenSearch configuration to **System Console > Environment > Elasticsearch***
> - *Add your file storage system configuration to **System Console > Environment > File Storage***
> - *Add your proxy configuration to **System Console > Environment > Image Proxy***
> - *Add your SMTP configuration to **System Console > Environment > SMTP***
> - *Enable Push Notifications by adding your server to **System Console > Environment > Push Notification Server***
> - *Add your cluster configuration to **System Console > Environment > High Availability***

- Configure your site within the System Console

> - *Resource: https://docs.mattermost.com/administration-guide/configure/configuration-settings.html#site-configuration*

- Set site access policies including permissions for roles and guest access

> - *Permissions Resource: https://docs.mattermost.com/administration-guide/onboard/advanced-permissions.html*
> - *Guest Access Resource: https://docs.mattermost.com/administration-guide/onboard/guest-accounts.html*

5. Test production performance and redundancy

- Define and test disaster recovery policy and processes

> - *Resource: https://docs.mattermost.com/install/install-kubernetes.html#using-mattermost-operator-functionality*

> - *Resource: https://docs.mattermost.com/administration-guide/scale/high-availability-cluster-based-deployment.html#upgrade-guide*

- Performance test production environment

> - *Load test production environment to verify that it will handle anticipated user load*
>
> > - *Resource: https://github.com/mattermost/mattermost-load-test*
>
> - *Set up Prometheus and Grafana to monitor performance*
>
> > - *Resource: https://docs.mattermost.com/administration-guide/scale/deploy-prometheus-grafana-for-performance-monitoring.html*
>
> - *Set up alerts in Grafana*
>
> > - *Resource: https://docs.mattermost.com/administration-guide/scale/deploy-prometheus-grafana-for-performance-monitoring.html*

Roll out Mattermost

Now that you have an environment in place, we recommend working through the following items in an iterative process. You may need to cycle through each of these topics multiple times to make adjustments to suit your organization as you onboard groups of users.

1. Define your team and channel strategy

- Determine and create a team structure for your environment

> - *Recommendation: Start with fewer teams in your early roll out*

> - *Resource: https://docs.mattermost.com/messaging/organizing-channels.html*

- Determine and create key channels to support your users. A default Town Square channel is created automatically and available on every team.

> - *Recommendation: Add a "Support" channel for your users to escalate questions*

- (Optional) Migrate messages and channels from legacy systems

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/migrating-to-mattermost.html*

2. Enable key integrations

- Build the list of key integrations and tools used by your teams

> - *Resource: https://developers.mattermost.com/integrate/getting-started/*

- Define use cases and requirements for plugins, bots, webhooks, slash commands

> - *Resource: https://developers.mattermost.com/integrate/other-integrations/*

- Set up key integrations (or migrate from POC environments)

> - *Resource: https://mattermost.com/marketplace/*

- Understand Mattermost API capabilities

> - *Resource: https://api.mattermost.com/*

3. Prepare for user onboarding

- Onboard champion users
- Onboard trainers and support team

- Create a training plan

> - *Resource: https://academy.mattermost.com/*

- Define user escalation and support processes

> - *Ensure you have set the site's support URL to your own support team under **System Console > Site Configuration > Customization***

- Notify users in advance of roll out

> - *Sample email: https://docs.mattermost.com/getting-started/welcome-email-to-end-users.html*

4. Deploy client apps

- Roll out desktop app

> - *Resource: https://docs.mattermost.com/end-user-guide/collaborate/install-desktop-app.html*
> - *Resource: https://docs.mattermost.com/deployment-guide/desktop-app.html*
> - *(Optional) Use the MSI installer to install on Windows machines*
>
> > - *Resource: https://docs.mattermost.com/deployment-guide/desktop/desktop-msi-installer-and-group-policy-install.html*

- Roll out mobile app

> - *Resource: https://docs.mattermost.com/deployment-guide/mobile-overview.html*
> - *(Optional) Use an EMM provider*
>
> > - *Resource: https://docs.mattermost.com/deployment-guide/mobile/deploy-mobile-apps-using-emm-provider.html*

5. Roll out to groups of users

- Provision user accounts

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/user-provisioning-workflows.html*

- (Optional) Bulk Load users

> - *Resource: https://docs.mattermost.com/administration-guide/onboard/bulk-loading-data.html*

- Onboard users to teams and channels

> - *Recommendation: Use LDAP Group Sync to automate this process*
>
> > - *Resource: https://docs.mattermost.com/administration-guide/onboard/ad-ldap-groups-synchronization.html*

- Implement your training plan to end users on how to use Mattermost

> - *Train on using Mattermost*
> - *Train on how to use integrations*

6. Drive adoption

- Incrementally roll out to additional user groups

> - *See "Roll Out to Groups of Users"*

- Manage support requests and product requests from your end users

> - *Resource: https://mattermost.com/support/*
> - *See the process below for escalating to Mattermost*

- Enable additional integrations and plugins to support user workflows

> - *Resource: https://mattermost.com/marketplace/*

- Understand management tools available to support users

> - *mmctl Command Line Tool Resource: https://docs.mattermost.com/administration-guide/manage/mmctl-command-line-tool.html*

> - *Command Line Tools Resource: https://docs.mattermost.c../administration-guide/manage/command-line-tools.html*

Review the roll out

We recommend that you review your rollout on a cadence that matches your iterative approach to rolling out to users. Below are some areas to consider.

1. Review project charter success metrics

- Perform end-user surveys and measure satisfaction

> - *Optional resource within Mattermost: https://mattermost.com/marketplace/matterpoll/*

- Verify use case fulfillment from original requirements gathering
- Measure your response time and resolution rate for user support issues
- Identify usage gaps and address or create a plan for addressing

2. Review and analyze usage

- Review Project Charter success metrics - identify usage gaps and address or create a plan for addressing
- Monitor site and team statistics

> - *Resource: https://docs.mattermost.com/administration-guide/manage/statistics.html*
> - *Review: Total posts, total teams, total channels, total group chats, total direct chats, top channels, top teams*

- Analyze usage by lines of business and peak usage times

3. Analyze system performance

- Monitor trends in CPU/memory usage
- Review trends in database connections
- Review trends in Go routines

- Review trends in concurrent sessions

4. Harden security

- Harden security controls around the web, desktop, and mobile security
- Harden configuration management
- Harden network security

> - *Identify additional tests and scans*
> - *(Optional) Enable Compliance Reporting*
> > - *Resource: https://docs.mattermost.com/administration-guide/comply/compliance-export.html*

5. Perform maintenance tasks

- Monitor for security updates (or sign up for email updates)

> - *Resource: https://mattermost.com/security-updates/*

- Perform the first upgrade

> - *Resource: https://docs.mattermost.c../administration-guide/upgrade/upgrading-mattermost-server.html*

- Determine upgrade schedule based on Mattermost release schedules and life cycle

> - *Resource: https://docs.mattermost.com/about/releases-lifecycle.html*

- Run System checks and either address or set address-by date

On this page

- Enterprise roll out checklist
  - Checklist overview
    - Prepare for the roll out
    - Roll out Mattermost
    - Review the roll out

- Checklist details
  - Prepare for the roll out
    - 1. Define the roll out project
    - 2. Validate essential security and compliance requirements
    - 3. Create development, staging, and production environments
    - 4. Configure and customize your Mattermost site
    - 5. Test production performance and redundancy
  - Roll out Mattermost
    - 1. Define your team and channel strategy
    - 2. Enable key integrations
    - 3. Prepare for user onboarding
    - 4. Deploy client apps
    - 5. Roll out to groups of users
    - 6. Drive adoption
  - Review the roll out
    - 1. Review project charter success metrics
    - 2. Review and analyze usage
    - 3. Analyze system performance
    - 4. Harden security
    - 5. Perform maintenance tasks

Important Upgrade Notes

Important

Support for Mattermost Server v9.11 Extended Support Release has come to the end of its life cycle on May 15, 2025. Upgrading to Mattermost Server v10.5 or later is required.

We recommend reviewing the additional upgrade notes at the bottom of this page.

If you're upgrading from a version earlier than...	Then...
v10.10	Added a new column `DefaultCategoryName` to the `Channels` table. This is nullable and stores a category name to be added/created when new users join a channel. This is only used if the `ExperimentalChannelCategorySetting` is enabled. The migrations are fully backwards-compatible and no table locks or existing operations on the table are impacted by this upgrade. Zero downtime is expected when upgrading to this release. The SQL queries included are: PostgreSQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000138_add_default_category_name_to_channel.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000138_add_default_category_name_to_channel.up.sql MySQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000138_add_default_category_name_to_channel.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000138_add_default_category_name_to_channel.up.sql
	Added new columns `RemoteId` and `ChannelId` to the `PostAcknowledgements` table. This is nullable, and stores the remote ID (if available) and channel ID to which the post was made when acknowledgements to posts were set. The migrations are fully backwards-compatible and no table locks or existing operations on the table are impacted by this upgrade. Zero downtime is expected when upgrading to this release. The SQL queries included are: PostgreSQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000141_add_remoteid_channelid_to_post_acknowledgements.down.sql and https://github.com/mattermost/mattermost/blob/master/server/

If you're upgrading from a version earlier than...	Then...
	channels/db/migrations/postgres/000141_add_remoteid_channelid_to_post_acknowledgements.up.sql MySQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000141_add_remoteid_channelid_to_post_acknowledgements.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000141_add_remoteid_channelid_to_post_acknowledgements.up.sql

If you're upgrading from a version earlier than...	Then...
	Added a new column `LastMembersSyncAt` to the `SharedChannelRemotes` table and added `LastMembershipSyncAt` to `SharedChannelUsers`. This is nullable, and stores the `LastMembersSyncAt` timestamp (if available) which tracks the last time channel membership data was synchronized between the local cluster and each remote cluster. `LastMembershipSyncAt` timestamp tracks the last time a specific user's channel membership status was synchronized with a specific remote cluster. The migrations are fully backwards-compatible and no table locks or existing operations on the table are impacted by this upgrade. Zero downtime is expected when upgrading to this release. The SQL queries included are: PostgreSQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000140_add_lastmemberssyncat_to_sharedchannelremotes.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000140_add_lastmemberssyncat_to_sharedchannelremotes.up.sql MySQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000140_add_lastmemberssyncat_to_sharedchannelremotes.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000140_add_lastmemberssyncat_to_sharedchannelremotes.up.sql
	Added a new column `LastGlobalUserSyncAt` to the `RemoteClusters` table. This is nullable, and `LastGlobalUserSyncAt` tracks the timestamp of the last successful global user sync for each remote cluster, enabling incremental syncing. The migrations are fully backwards-compatible and no table locks or existing operations on the table are impacted by this upgrade. Zero downtime is expected when upgrading to this release. The SQL queries included are:

If you're upgrading from a version earlier than...	Then...
	PostgreSQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000139_remoteclusters_add_last_global_user_sync_at.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000139_remoteclusters_add_last_global_user_sync_at.up.sql MySQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000139_remoteclusters_add_last_global_user_sync_at.down.sql and https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000139_remoteclusters_add_last_global_user_sync_at.up.sql
v10.9	A new index to the `CategoryId` column in `SidebarChannels` table was added to improve query performance. No database downtime is expected for this upgrade. For PostgreSQL, it takes around 2s to add the index on a table with 1.2M rows with an instance size of 8 cores and 16GB RAM. For MySQL, it takes around 5s on a table with 300K rows on an instance size of 8 cores and 16GB RAM. The migrations are fully backwards-compatible and no table locks or existing operations on the table are impacted by this upgrade. Zero downtime is expected when upgrading to this release. The SQL queries included are: **PostgreSQL**: `CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_sidebarchannels_categoryid ON sidebarchannels(categoryid);` **MySQL**:

If you're upgrading from a version earlier than...	Then...
	`CREATE INDEX idx_sidebarchannels_categoryid ON SidebarChannels(CategoryId);`
	Schema changes in the form of a new materialized view (`AttributeView`) was added that aggregates user attributes into a separate table. No database downtime is expected for this upgrade. A previous version of Mattermost can run with the new schema changes. The SQL queries included in the schema changes are below: PostgreSQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/postgres/000137_update_attribute_view.up.sql MySQL: https://github.com/mattermost/mattermost/blob/master/server/channels/db/migrations/mysql/000136_create_attribute_view.up.sql The objective of showing the SQL queries is not to reduce downtime, but to let the Admin make the schema changes without requiring to upgrade Mattermost. Then when they actually upgrade Mattermost, it becomes instantaneous since the changes are already made.
v10.8	New tables `AccessControlPolicies` and `AccessControlPolicyHistory` will be created. The migration is fully backwards-compatible, non-locking, and zero downtime is expected.
	Support for legacy SKUs E10 and E20 licenses was removed. If you need assistance, talk to a Mattermost expert (https://mattermost.com/contact-sales/).

If you're upgrading from a version earlier than...	Then...

v10.7	Added new column `BannerInfo` in the `Channels` table for storing metadata for an upcoming licensed feature. The migration is fully backwards-compatible, non-locking, and zero downtime is possible. Below are the SQL queries included in the schema changes:
	
	PostgreSQL:
	
	`ALTER TABLE channels ADD COLUMN IF NOT EXISTS bannerinfo jsonb;`
	
	MySQL:
	
	`SET @preparedStatement = (SELECT IF(NOT EXISTS(SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS WHERE table_name = 'Channels' AND table_schema = DATABASE() AND column_name = 'BannerInfo'), 'ALTER TABLE Channels ADD COLUMN BannerInfo json;', 'SELECT 1;')); PREPARE addColumnIfNotExists FROM @preparedStatement; EXECUTE addColumnIfNotExists; DEALLOCATE PREPARE addColumnIfNotExists;`

	Added support for cursor-based pagination on the property architecture tables, including SQL migration to create indices. The migration is fully backwards-compatible, non-locking, and zero downtime is possible. Below are the SQL queries included in the schema changes: MySQL:

If you're upgrading from a version earlier than...	Then...
	`SET @preparedStatement = (SELECT IF((SELECT COUNT(*) FROM INFORMATION_SCHEMA.STATISTICS WHERE table_name = 'PropertyValues' AND table_schema = DATABASE() AND index_name = 'idx_propertyvalues_create_at_id') > 0, 'SELECT 1', 'CREATE INDEX idx_propertyvalues_create_at_id ON PropertyValues(CreateAt, ID);')); PREPARE createIndexIfNotExists FROM @preparedStatement; EXECUTE createIndexIfNotExists; DEALLOCATE PREPARE createIndexIfNotExists;`  `SET @preparedStatement = (SELECT IF((SELECT COUNT(*) FROM INFORMATION_SCHEMA.STATISTICS WHERE table_name = 'PropertyFields' AND table_schema = DATABASE() AND index_name = 'idx_propertyfields_create_at_id') > 0, 'SELECT 1', 'CREATE INDEX idx_propertyfields_create_at_id ON PropertyFields(CreateAt, ID);')); PREPARE createIndexIfNotExists FROM @preparedStatement; EXECUTE createIndexIfNotExists; DEALLOCATE PREPARE createIndexIfNotExists;`  PostgreSQL:  `CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_propertyvalues_create_at_id ON PropertyValues(CreateAt, ID)`  `CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_propertyfields_create_at_id ON PropertyFields(CreateAt, ID)`
v10.6	

If you're upgrading from a version earlier than...	Then...
	Support for PostgreSQL v11 and v12 have been removed. The new minimum PostgreSQL version is v13+. See the minimum supported PostgreSQL version policy documentation for details.
	Migration times: On a system with 12M posts, and 1M fileinfo entries, the migration takes 15s. This migration is non-locking. Note that there is no migration for MySQL deployments because this optimization is only applicable for PostgreSQL.
v10.5	Mattermost versions v10.5.0 and v10.5.1 include a bundled Jira plugin (v4.2.0) that contains a bug which may cause plugin configuration settings to disappear. We strongly advise against upgrading to these versions to avoid potential disruption. If you have already upgraded to v10.5.0 or v10.5.1, we recommend updating the Jira plugin to version v4.2.1, or preferably, upgrading both Mattermost and the Jira plugin to their latest available versions.
	The internal workings of the *PluginLinkComponent* in the web app have been changed to unmount link tooltips from the DOM by default, significantly improving performance. Plugins that register link tooltips using *registerLinkTooltipComponent* will experience changes in how tooltip components are managed—they are now only mounted when a link is hovered over or focused. As a result, plugins may need to update their components to properly handle mounting and unmounting scenarios. For example, changes were made in mattermost-plugin-jira (https://github.com/mattermost/mattermost-plugin-jira/pull/1145), where componentDidUpdate lifecycle hook was replaced with componentDidMount. If your plugin's tooltip component is a functional React component, there is a high chance that this behavior will be handled automatically, as it would be managed by useEffect with an empty dependency array.

If you're upgrading from a version earlier than...	Then...
	The Mattermost server has stopped supporting manual plugin deployment. Plugins were deployed manually when an administrator or some deployment automation copies the contents of a plugin bundle into the server's working directory. If a manual or automated deployment workflow is still required, administrators can instead prepackage the plugin bundles. See this forum post (https://forum.mattermost.com/t/deprecation-notice-manual-plugin-deployment/21192) for details.
	Mattermost has stopped official Mattermost server builds for the Microsoft Windows operating system. Administrators should migrate existing Mattermost server installations to use the official Linux builds. See more details in this forum post (https://forum.mattermost.com/t/deprecation-notice-server-builds-for-microsoft-windows/21498).
	v10.5 introduces updates to the Compliance Export functionality, which will modify how exported data is structured, stored and processed. These changes primarily affect System Administrators and the main changes are outlined below. See more details in the Compliance Export documentation. Output files and directories have changed - Previously we were exporting a single zip containing all the batch directories. Now we will export a single directory, and under that directory each batch will be its own zip. Compliance exports performance improvements - Compliance exports should now be at least 50% faster, and possibly more. Logic improvements - We've made improvements and fixed bugs that we found. Changes specific to each Export Type - The export output formats have been changed. Some fields' semantic meaning has been clarified, and

If you're upgrading from a version earlier than...	Then...
	there are a number of new fields. Our goal was to maintain backwards compatibility while fixing the logic bugs. See the compliance export product documentation for details.
	As part of the Property System Architecture feature, Mattermost v10.5 is going to run a set of migrations to add new tables to the schema. This migration only creates new tables and indexes, so there is no impact on preexisting data. New tables `PropertyGroups`, `PropertyFields` and `PropertyValues` are going to be created. In the case of PostgreSQL, a new enum type `property_field_type` is going to be created, to be used in the `Type` column of the `PropertyFields` table. The `PropertyFields` and `PropertyValues` tables have a unique constraint that will generate an index in MySQL, and in the case of PostgreSQL, they directly use a `UNIQUE INDEX` to enforce this constraint. A new index `idx_propertyvalues_targetid_groupid` will be created for the `PropertyValues` table. The migration is only creating new tables with no data. The migration is backwards-compatible, and a previous version of Mattermost can run with the new schema changes. No table locks or existing operations on the table are impacted by this upgrade. Zero downtime is possible when upgrading to this release. Below are the SQL queries included in the schema changes: **MySQL**:

If you're upgrading from a version earlier than...	Then...
	`CREATE TABLE IF NOT EXISTS PropertyGroups (ID varchar(26) PRIMARY KEY, Name varchar(64) NOT NULL, UNIQUE(Name)); CREATE TABLE IF NOT EXISTS PropertyFields (ID varchar(26) PRIMARY KEY, GroupID varchar(26) NOT NULL, Name varchar(255) NOT NULL, Type enum('text', 'select', 'multiselect', 'date', 'user', 'multiuser'), Attrs json, TargetID varchar(255), TargetType varchar(255), CreateAt bigint(20), UpdateAt bigint(20), DeleteAt bigint(20), UNIQUE(GroupID, TargetID, Name, DeleteAt)); CREATE TABLE IF NOT EXISTS PropertyValues (ID varchar(26) PRIMARY KEY, TargetID varchar(255) NOT NULL, TargetType varchar(255) NOT NULL, GroupID varchar(26) NOT NULL, FieldID varchar(26) NOT NULL, Value json, CreateAt bigint(20), UpdateAt bigint(20), DeleteAt bigint(20), UNIQUE(GroupID, TargetID, FieldID, DeleteAt)); SET @preparedStatement = (SELECT IF((SELECT COUNT(*) FROM INFORMATION_SCHEMA.STATISTICS WHERE table_name = 'PropertyValues' AND table_schema = DATABASE() AND index_name = 'idx_propertyvalues_targetid_groupid') > 0, 'SELECT 1', 'CREATE INDEX idx_propertyvalues_targetid_groupid ON PropertyValues (TargetID, GroupID);')); PREPARE createIndexIfNotExists FROM @preparedStatement; EXECUTE createIndexIfNotExists; DEALLOCATE PREPARE createIndexIfNotExists;` **PostgreSQL**:

If you're upgrading from a version earlier than...	Then...
	```CREATE TABLE IF NOT EXISTS PropertyGroups (ID varchar(26) PRIMARY KEY, Name varchar(64) NOT NULL, UNIQUE(Name)); DO BEGIN IF NOT EXISTS (SELECT * FROM pg_type typ INNER JOIN pg_namespace nsp ON nsp.oid = typ.typnamespace WHERE nsp.nspname = current_schema() AND typ.typname = 'property_field_type') THEN CREATE TYPE property_field_type AS ENUM ('text', 'select', 'multiselect', 'date', 'user', 'multiuser'); END IF; END; LANGUAGE plpgsql; CREATE TABLE IF NOT EXISTS PropertyFields (ID varchar(26) PRIMARY KEY, GroupID varchar(26) NOT NULL, Name varchar(255) NOT NULL, Type property_field_type, Attrs jsonb, TargetID varchar(255), TargetType varchar(255), CreateAt bigint NOT NULL, UpdateAt bigint NOT NULL, DeleteAt bigint NOT NULL); CREATE UNIQUE INDEX IF NOT EXISTS idx_propertyfields_unique ON PropertyFields (GroupID, TargetID, Name) WHERE DeleteAt = 0; CREATE TABLE IF NOT EXISTS PropertyValues (ID varchar(26) PRIMARY KEY, TargetID varchar(255) NOT NULL, TargetType varchar(255) NOT NULL, GroupID varchar(26) NOT NULL, FieldID varchar(26) NOT NULL, Value jsonb NOT NULL, CreateAt bigint NOT NULL, UpdateAt bigint NOT NULL, DeleteAt bigint NOT NULL); CREATE UNIQUE INDEX IF NOT EXISTS idx_propertyvalues_unique ON PropertyValues (GroupID, TargetID, FieldID) WHERE DeleteAt = 0; CREATE INDEX IF NOT EXISTS idx_propertyvalues_targetid_groupid ON PropertyValues (TargetID, GroupID); ```
v10.3	The Classic Mobile App has been phased out. Please download the new v2 Mobile App from the Apple App Store (https://apps.apple.com/us/app/mattermost/id1257222717) or Google Play Store (https://play.google.com/store/apps/details?id=com.mattermost.rn). See more details in the classic mobile app deprecation (https://forum.mattermost.com/t/classic-mobile-app-deprecation/18703) Mattermost forum post.

If you're upgrading from a version earlier than...	Then...
v10.2	Docker Content Trust (DCT) for signing Docker image artifacts has been replaced by Sigstore Cosign in v10.2 (November, 2024). If you rely on artifact verification using DCT, please transition to using Cosign (https://edu.chainguard.dev/open-source/sigstore/cosign/how-to-install-cosign/). See the upcoming DCT deprecation (https://forum.mattermost.com/t/upcoming-dct-deprecation/19275) Mattermost forum post for more details.
v10.0	We no longer support new installations using MySQL starting in v10. All new customers and/or deployments will only be supported with the minimum supported version of the PostgreSQL database. End of support for MySQL is targeted for Mattermost v11.
	Apps Framework is deprecated for new installs. Please extend Mattermost using webhooks, slash commands, OAuth2 apps, and plugins.
	Mattermost v10 introduces Playbooks v2 for all Enterprise licensed customers. Professional SKU customers may continue to use Playbooks v1 uninterrupted which will be maintained and supported until September 2025, followed by an appropriate grandfathering strategy. More detailed information and the discussion are available on our forums here (https://forum.mattermost.com/t/clarification-on-playbooks-in-mattermost-v10/20563).
	Renamed `Channel Moderation` to `Advanced Access Control` in the channel management section in the **System Console**.

If you're upgrading from a version earlier than...	Then...
	Renamed announcement banner feature to "system-wide notifications".
	Renamed "Collapsed Reply Threads" to "Threaded Discussions" in the System Console.
	Renamed "System Roles" to "Delegated Granular Administration" in the System Console.
	Renamed "Office 365" to "Entra ID" for SSO logins.
	Fully deprecated the `/api/v4/image` endpoint when the image proxy is disabled.
	Pre-packaged [Calls plugin v1.0.1 (https://github.com/mattermost/mattermost-plugin-calls/releases/tag/v1.0.1)](https://github.com/mattermost/mattermost-plugin-calls/releases/tag/v1.0.1). This includes breaking changes including the removal of group calls from unlicensed servers in order to focus supportability and quality on licensed servers. Unlicensed servers can continue to use Calls in direct message channels, which represent the majority of activity.
	Removed deprecated `Config.ProductSettings`, `LdapSettings.Trace`, and `AdvancedLoggingConfig` configuration fields.
	Removed deprecated `pageSize` query parameter from most API endpoints.

If you're upgrading from a version earlier than...	Then...
	Deprecated the experimental Strict CSRF token enforcement. This feature will be fully removed in Mattermost v11.
v9.11	Added support for Elasticsearch v8. Also added Beta support for OpenSearch v1.x and v2.x. A new config setting `ElasticsearchSettings.Backend` has been added to differentiate between Elasticsearch and OpenSearch. The default value is `elasticsearch` which breaks support for AWS Elasticsearch v7.10.x since the official v8 client only works from Elasticsearch v7.11+ versions. Note • For AWS customers on OpenSearch, you must modify Mattermost configuration from `elasticsearch` to `opensearch` and disable compatibility mode. See the OpenSearch documentation for details on upgrading. • After upgrading the Mattermost server, use mmctl or edit the config manually, then restart the Mattermost server. • If you are using OpenSearch, you **must** set the backend to `opensearch`. Otherwise Mattermost will not work. If you are using Elasticsearch v8, be sure to set `action.destructive_requires_name` to `false` in `elasticsearch.yml` to allow for wildcard operations to work.

If you're upgrading from a version earlier than...	Then...
v9.5	We have stopped supporting MySQL v5.7 since it's at the end of life. We urge customers to upgrade their MySQL instance at their earliest convenience.
	Added safety limit error message in compiled Team Edition and Enterprise Edition deployments when enterprise scale and access control automation features are unavailable and count of users who are registered and not deactivated exceeds 10,000. ERROR_SAFETY_LIMITS_EXCEEDED.
v9.2	Fixed data retention policies to run jobs when any custom retention policy is enabled even when the global retention policy is set to "keep-forever". Before this fix, the enabled custom data retention policies wouldn't run as long as the global data retention policy was set to "keep-forever" or was disabled. After the fix, the custom data retention policies will run automatically even when the global data retention policy is set to "keep-forever". Once the server is upgraded, posts may unintentionally be deleted. Admins should make sure to disable all custom data retention policies before upgrading, and then re-enable them again after upgrading.
v9.1	Minimum supported Desktop App version is now v5.3. OAuth/SAML flows were modified to include `desktop_login` which makes earlier versions incompatible.
v9.0	Removed the deprecated Insights feature.
	Mattermost Boards and various other plugins have transitioned to being fully community supported. See this [forum post (https://](https://)

If you're upgrading from a version earlier than...	Then...
	forum.mattermost.com/t/upcoming-product-changes-to-boards-and-various-plugins/16669) for more details.
	The `channel_viewed` websocket event was changed to `multiple_channels_viewed`, and is now only triggered for channels that actually have unread messages.

v8.1	In v8.1.2, improved performance on data retention `DeleteOrphanedRows` queries.
	
	New migration for a new table:
	
	MySQL:
	
	`CREATE TABLE IF NOT EXISTS RetentionIdsForDeletion(Id VARCHAR(26) NOT NULL, TableName VARCHAR(64), Ids json, PRIMARY KEY (Id), KEY idx_retentionidsfordeletion_tablename (TableName)) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4; ```
	
	PostgreSQL:
	
	`CREATE TABLE IF NOT EXISTS retentionidsfordeletion(id VARCHAR(26) PRIMARY KEY, tablename VARCHAR(64), ids VARCHAR(26) []); CREATE INDEX IF NOT EXISTS idx_retentionidsfordeletion_tablename ON retentionidsfordeletion(tablename); ```
	
	Hard deleting a user or a channel will now also clean up associated reactions.
	
	Removed feature flag `DataRetentionConcurrencyEnabled`. Data retention now runs without concurrency in order to avoid any performance degradation.

If you're upgrading from a version earlier than...	Then...
	Added a new configuration setting `DataRetentionSettings.RetentionIdsBatchSize`, which allows admins to configure how many batches of IDs will be fetched at a time when deleting orphaned reactions. The default value is 100.
v8.0	Insights has been deprecated for all new instances and for existing servers that upgrade to v8.0. See more details in [this forum post (https://forum.mattermost.com/t/proposal-to-revise-our-insights-feature-due-to-known-performance-issues/16212)](https://forum.mattermost.com/t/proposal-to-revise-our-insights-feature-due-to-known-performance-issues/16212) on why Insights has been deprecated.
	The Focalboard plugin is now disabled by default for all new instances and can be enabled in the **System Console > Plugin settings**.
	The Channel Export and Apps plugins are now disabled by default.
	Apps Bar is now enabled by default for on-prem servers. `ExperimentalSettings.EnableAppBar` was also renamed to `ExperimentalSettings.DisableAppBar`. See the :ref: *configuration settings <administration-guide/configure/experimental-configuration-settings:disable-apps-bar>* documentation, and [this forum article (https://forum.mattermost.com/t/channel-header-plugin-changes/13551)](https://forum.mattermost.com/t/channel-header-plugin-changes/13551) for details.
	In the main *server package*, the Go module path has changed from `github.com/mattermost/mattermost-server/server/v8` to `github.com/mattermost/mattermost/server/v8`. But with the introduction of the *public* submodule, it should no longer be necessary for third-party code to import this *server* package.

If you're upgrading from a version earlier than...	Then...
	Introduced the public (https://github.com/mattermost/mattermost/tree/master/server/public) submodule, housing the familiar *model* and *plugin* packages, but now discretely versioned from the server. It is no longer necessary to *go get* a particular commit hash, as Go programs and plugins can now opt-in to importing *github.com/mattermost/mattermost-server/server/public* and managing versions idiomatically. While this submodule has not yet shipped a v1 and will introduce breaking changes before stabilizing the API, it remains both forwards and backwards compatible with the Mattermost server itself.
	As part of the *public* submodule above, a `context.Context` is now passed to `model.Client4` methods.
	Removed support for PostgreSQL v10. The new minimum PostgreSQL version is now v11.
	The Mattermost public API for Go is now available as a distinctly versioned package. Instead of pinning a particular commit hash, use idiomatic Go to add this package as a dependency: go get *github.com/mattermost/mattermost-server/server/public*. This relocated Go API maintains backwards compatibility with Mattermost v7. Furthermore, the existing Go API previously at github.com/mattermost/mattermost-server/v6/model remains forward compatible with Mattermost v8, but may not contain newer features. Plugins do not need to be recompiled, but developers may opt in to using the new package to simplify their build process. The new public package is shipping alongside Mattermost v8 as version 0.5.0 to allow for some additional code refactoring before releasing as v1 later this year.

If you're upgrading from a version earlier than...	Then...
	Three configuration fields have been added, `LogSettings.AdvancedLoggingJSON`, `ExperimentalAuditSettings.AdvancedLoggingJSON`, and `NotificationLogSettings.AdvancedLoggingJSON` which support multi-line JSON, escaped JSON as a string, or a filename that points to a file containing JSON. The `AdvancedLoggingConfig` fields have been deprecated.
	The Go MySQL driver has changed the `maxAllowedPacket` size from 4MiB to 64MiB. This is to make it consistent with the change in the server side default value from MySQL 5.7 to MySQL 8.0. If your `max_allowed_packet` setting is not 64MiB, then please update the MySQL config DSN with an additional param of `maxAllowedPacket` to match with the server side value. Alternatively, a value of 0 can be set to to automatically fetch the server side value, on every new connection, which has a performance overhead.
	Removed `ExperimentalSettings.PatchPluginsReactDOM`. If this setting was previously enabled, confirm that: 1. All Mattermost-supported plugins are updated to the latest versions. 2. Any other plugins have been updated to support React 17. See the section for v7.7 release for more information.
	Removed deprecated `PermissionUseSlashCommands`.
	Removed deprecated `model.CommandArgs.Session`.

If you're upgrading from a version earlier than...	Then...
	For servers wanting to allow websockets to connect from origins other than the origin of the site URL, please set the `ServiceSettings.AllowCorsFrom` configuration setting. Also ensure that the `siteURL` is set correctly.
	In v8.0 release, the following repositories are merged into one: `mattermost-server`, `mattermost-webapp` and `mmctl`. Developers should read the updated [Developer Guide (https://developers.mattermost.com/contribute/developer-setup/)](https://developers.mattermost.com/contribute/developer-setup/) for details.
	Fixed an issue caused by a migration in the previous release. Query takes around 11ms on a PostgreSQL 14 DB t3.medium RDS instance. Locks on the preferences table will only be acquired if there are rows to delete, but the time taken is negligible.
	Fixed an issue where a user would still see threads in the threads view of channels they have left. Migration execution time in PostgreSQL: Execution time: 58.11 sec, DELETE 2766690. Migration execution time in MySQL: Query OK, 2766769 rows affected (4 min 47.57 sec).
	For servers wanting to allow websockets to connect from other origins, please set the `ServiceSettings.AllowCorsFrom` config setting.
	The file info stats query is now optimized by denormalizing the `channelID` column into the table itself. This will speed up the query to get the file count for a channel when selecting the right-hand pane. Migration times: On a PostgreSQL 12.14 DB with 1731 rows in FileInfo and 11M posts, it took around 0.27s

If you're upgrading from a version earlier than...	Then...
	On a MySQL 8.0.31 DB with 1405 rows in FileInfo and 11M posts, it took around 0.3s
v7.10	In v7.10.1, fixed an issue caused by a migration in the previous release. Query takes around 11ms on a PostgreSQL 14 DB t3.medium RDS instance. Locks on the preferences table will only be acquired if there are rows to delete, but the time taken is negligible.
	In v7.10.1, fixed an issue where a user would still see threads in the threads view of channels they have left. Migration execution time in MySQL: Query OK, 2766769 rows affected (4 min 47.57 sec). Migration execution time in PostgreSQL: 58.11 sec, DELETE 2766690.
	In v7.10.3, for servers wanting to allow websockets to connect from origins other than the origin of the site URL, please set the `ServiceSettings.AllowCorsFrom` configuration setting. Also ensure that the `siteURL` is set correctly.
v7.9	Added a new index on `Posts(OriginalId)`. For a database with 11.8 million posts, on a machine with a i7-11800H CPU (8 cores, 16 threads), 32GiB of RAM and SSD, the index creation takes 98.51s on MYSQL and 2.6s on PostgreSQL. • In PostgreSQL databases, the `Posts` table will be locked during index creation. To avoid locking this table, admins can create the index manually before performing the upgrade using the following non-blocking query: `CREATE INDEX CONCURRENTLY idx_posts_original_id ON Posts(OriginalId);`.

If you're upgrading from a version earlier than...	Then...
	• Admins managing PostgreSQL deployments containing fewer posts may prefer that the migration process creates the index, and accept that `Posts` table will remain locked until the migration is complete.
	In v7.9.4, fixed an issue where a user would still see threads in the threads view of channels they have left. Migration execution time in MySQL: Query OK, 2766769 rows affected (4 min 47.57 sec). Migration execution time in PostgreSQL: 58.11 sec, DELETE 2766690.
	In v7.9.4, backported a fix related Oauth 2. Query times depend on if you have rows to delete or not. With no rows to delete: • MySQL v5.7.12: 9ms • PostgreSQL v10: 21ms 4 rows: • MySQL v5.7.12: 17.2ms • PostgreSQL v10: 9.9ms Those times are based off the following table sizes: • Preferences: 2 million records • `oauthaccessdata` and sessions: 10 records You can assess the number of impacted rows by running the following: **PostgreSQL**:

If you're upgrading from a version earlier than...	Then...

	`SELECT count(o.*) FROM oauthaccessdata o WHERE NOT EXISTS ( SELECT p.* FROM preferences p WHERE o.clientid = p.name AND o.userid = p. userid AND p.category = 'oauth_app' );`
	
	MySQL:
	
	`SELECT COUNT(o.`Token`) FROM OAuthAccessData o LEFT JOIN Preferences p ON o. clientid = p.name AND o.userid = p.userid AND p.category = 'oauth_app' INNER JOIN Sessions s ON o.token = s .token WHERE p.name IS NULL;`
	
	Locks on the `oauthaccessdata` and sessions table will only be acquired if there are rows to delete.

	In v7.9.5, for servers wanting to allow websockets to connect from origins other than the origin of the site URL, please set the `ServiceSettings.AllowCorsFrom` configuration setting. Also ensure that the `siteURL` is set correctly.
v7.8	Message Priority & Acknowledgement is now enabled by default for all instances. You may disable this feature in the System Console by going to **Posts > Message Priority** or via the config `PostPriority` setting.
	In v7.8.5, fixed an issue where a user would still see threads in the threads view of channels they have left. Migration execution time in MySQL: Query OK, 2766769 rows affected (4 min 47.57 sec). Migration execution time in PostgreSQL: 58.11 sec, DELETE 2766690.

If you're upgrading from a version earlier than...	Then...

	In v7.8.5, backported a fix related Oauth 2. Query times depend on if you have rows to delete or not.
	
	With no rows to delete:
	
	• MySQL v5.7.12: 9ms
	
	• PostgreSQL v10: 21ms
	
	4 rows:
	
	• MySQL v5.7.12: 17.2ms
	
	• PostgreSQL v10: 9.9ms
	
	Those times are based off the following table sizes:
	
	• Preferences: 2 million records
	
	• `oauthaccessdata` and sessions: 10 records
	
	You can assess the number of impacted rows by running the following:
	
	PostgreSQL:
	
	`SELECT count(o.*) FROM oauthaccessdata o WHERE NOT EXISTS ( SELECT p.* FROM preferences p WHERE o.clientid = p.name AND o.userid = p. userid AND p.category = 'oauth_app' );`
	
	MySQL:
	
	`SELECT COUNT(o.`Token`) FROM OAuthAccessData o LEFT JOIN Preferences p ON o. clientid = p.name AND o.userid = p.userid AND p.category = 'oauth_app' INNER JOIN Sessions s ON o.token = s .token WHERE p.name IS NULL;`

If you're upgrading from a version earlier than...	Then...
	Locks on the `oauthaccessdata` and sessions table will only be acquired if there are rows to delete.
	In v7.8.7, for servers wanting to allow websockets to connect from origins other than the origin of the site URL, please set the `ServiceSettings.AllowCorsFrom` configuration setting. Also ensure that the `siteURL` is set correctly.

	In v7.8.11, improved performance on data retention `DeleteOrphanedRows` queries.
	
	New migration for a new table:
	
	MySQL:
	
	`CREATE TABLE IF NOT EXISTS RetentionIdsForDeletion(Id VARCHAR(26) NOT NULL, TableName VARCHAR(64), Ids json, PRIMARY KEY (Id), KEY idx_retentionidsfordeletion_tablename (TableName)) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4; ```
	
	PostgreSQL:
	
	`CREATE TABLE IF NOT EXISTS retentionidsfordeletion(id VARCHAR(26) PRIMARY KEY, tablename VARCHAR(64), ids VARCHAR(26) []); CREATE INDEX IF NOT EXISTS idx_retentionidsfordeletion_tablename ON retentionidsfordeletion(tablename); ```
	
	Hard deleting a user or a channel will now also clean up associated reactions.

If you're upgrading from a version earlier than...	Then...
	Removed feature flag `DataRetentionConcurrencyEnabled`. Data retention now runs without concurrency in order to avoid any performance degradation. Added a new configuration setting `DataRetentionSettings.RetentionIdsBatchSize`, which allows admins to configure how many batches of IDs will be fetched at a time when deleting orphaned reactions. The default value is 100.
v7.7	Plugins with a webapp component may need to be updated to work with Mattermost v7.7 release and the updated `React v17` dependency. This is to avoid plugins crashing with an error about `findDOMNode` being called on an unmounted component. While our [starter template (https://github.com/mattermost/mattermost-plugin-starter-template)](https://github.com/mattermost/mattermost-plugin-starter-template) depended on an external version of `React`, it did not do the same for `ReactDOM`. Plugins need to update their `webpack.config.js` directives to externalize `ReactDOM`. For reference, see https://github.com/mattermost/mattermost-plugin-playbooks/pull/1489. Server-side only plugins are unaffected. This change can be done for existing plugins any time prior to upgrading to Mattermost v7.7 and is backwards compatible with older versions of Mattermost. If you run into issues, you can either enable `ExperimentalSettings.PatchPluginsReactDOM` or just disable the affected plugin while it's updated.
	Denormalized `Threads` table by adding the `ThreadTeamId` column. Even though it denormalizes the schema, we gain performance by removing the unnessesary joins. Test results for schema changes are outlined below: instance: `db.r5.large` size of `Threads` table: 846313 rows

If you're upgrading from a version earlier than...	Then...
	number of posts: 12 million number of reactions: 2.5 million **MySQL:** `-- Drop any existing TeamId column from 000094_threads_teamid.up.sql SET @preparedStatement = (SELECT IF(EXISTS(`` SELECT 1 FROM INFORMATION_SCHEMA.STATISTICS WHERE table_name = 'Threads' AND table_schema = DATABASE() AND column_name = 'TeamId'), 'ALTER TABLE Threads DROP COLUMN TeamId;', 'SELECT 1;')); PREPARE removeColumnIfExists FROM @preparedStatement; EXECUTE removeColumnIfExists; DEALLOCATE PREPARE removeColumnIfExists; SET @preparedStatement = (SELECT IF(NOT EXISTS(SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS WHERE table_name = 'Threads' AND table_schema = DATABASE() AND column_name = 'ThreadTeamId'), 'ALTER TABLE Threads ADD COLUMN ThreadTeamId varchar(26) DEFAULT NULL;', 'SELECT 1;')); PREPARE addColumnIfNotExists FROM @preparedStatement; EXECUTE addColumnIfNotExists; DEALLOCATE PREPARE addColumnIfNotExists; Query OK, 0 rows affected (7.71 sec) UPDATE Threads, Channels SET Threads.ThreadTeamId = Channels.TeamId WHERE Channels.Id = Threads.ChannelId AND Threads.ThreadTeamId IS NULL; Query OK, 846313 rows affected (51.32 sec) Rows matched: 846313 Changed: 846313 Warnings: 0` **PostgreSQL:**

If you're upgrading from a version earlier than...	Then...
	`-- Drop any existing TeamId column from 000094_threads_teamid.up.sql ALTER TABLE threads DROP COLUMN IF EXISTS teamid; ALTER TABLE threads ADD COLUMN IF NOT EXISTS threadteamid VARCHAR(26); ALTER TABLE Time: 2.236 ms UPDATE threads SET threadteamid = channels. teamid FROM channels WHERE threads.threadteamid IS NULL AND channels.id = threads. channelid; UPDATE 847646 Time: 29744.608 ms (00:29.745) **Backwards-compatibility:** A previous version of Mattermost can run with the new schema changes **Table locks or impact to existing operations on the table:** Table locks - Threads table` Queries posted above can be run prior to upgrading Mattermost for both MySQL and PostgreSQL. After schema changes migration becomes instantaneous (0.78 sec).
	Starting with the Calls version shipping with v7.7, there's now a minimum version requirement when using the external RTCD service. This means that if Calls is configured to use the external service, customers need to upgrade RTCD first to at least version 0.8.0 or the plugin will fail to start.
	In v7.7.2, Message Priority & Acknowledgement is now enabled by default for all instances. You may disable this feature in the System Console by going to **Posts > Message Priority** or via the config `PostPriority` setting.
v7.5	Added a new schema migration to ensure `ParentId` column is dropped from the `Posts` table. Depending on the table size, if the column is not dropped before, a significant spike in database CPU usage is expected

If you're upgrading from a version earlier than...	Then...
	on MySQL databases. Writes to the table will be limited during the migration.
	For `PluginRegistry.registerCustomRoute`, when you register a custom route component, you must specify a CSS `grid-area` in order for it to be placed properly into the root layout (recommended: `grid-area: center`).
v7.3	Boards is moving from a channel-based to a role-based permissions system. The migration will happen automatically, but your administrator should perform a backup prior to the upgrade. We removed workspaces, so if you were a member of many boards prior to migration, they will now all appear under the same sidebar.
v7.2	Several schema changes impose additional database constraints to make the data more strict. All the commands listed below were tested on a 8 core, 16GB RAM machine. Here are the times recorded: **PostgreSQL (131869 channels, 2 teams)**: • `CREATE TYPE channel_type AS ENUM ('P', 'G', 'O', 'D');` took 14.114 milliseconds • `ALTER TABLE channels alter column type type channel_type using type::channel_type;` took 3856.790 milliseconds (3.857 seconds) • `CREATE TYPE team_type AS ENUM ('I', 'O');` took 4.191 milliseconds • `ALTER TABLE teams alter column type type team_type using type::team_type;` took 116.205 milliseconds • `CREATE TYPE upload_session_type AS ENUM ('attachment', 'import');` took 4.266 milliseconds

If you're upgrading from a version earlier than...	Then...
	• `ALTER TABLE uploadsessions alter column type type upload_session_type using type::upload_session_type;` took 37.099 milliseconds **MySQL (270959 channels, 2 teams)**: • `ALTER TABLE Channels MODIFY COLUMN Type ENUM("D", "O", "G", "P");` took 13.24 seconds • `ALTER TABLE Teams MODIFY COLUMN Type ENUM("I", "O");` took 0.04 seconds • `ALTER TABLE UploadSessions MODIFY COLUMN Type ENUM("attachment", "import");` took 0.03 seconds
v7.1	A new configuration option `MaxImageDecoderConcurrency` indicates how many images can be decoded concurrently at once. The default is -1, and the value indicates the number of CPUs present. This affects the total memory consumption of the server. The maximum memory of a single image is dictated by `MaxImageResolution * 24 bytes`. Therefore, we recommend that `MaxImageResolution * MaxImageDecoderConcurrency * 24` should be less than the allocated memory for image decoding.
	Mattermost v7.1 introduces schema changes in the form of a new column and its index. Our test results for the schema changes are included below: • MySQL 12M Posts, 2.5M Reactions - ~1min 34s (instance: PC with 8 cores, 16GB RAM) • PostgreSQL 12M Posts, 2.5M Reactions - ~1min 18s (instance: db.r5.2xlarge) You can run the following SQL queries before the upgrade to obtain a lock on `Reactions` table, so that users' reactions posted during this

If you're upgrading from a version earlier than...	Then...
	time won't be reflected in the database until the migrations are complete. This is fully backwards-compatible. MySQL: • `ALTER TABLE Reactions ADD COLUMN ChannelId varchar(26) NOT NULL DEFAULT "";` • `UPDATE Reactions SET ChannelId = COALESCE((select ChannelId from Posts where Posts.Id = Reactions.PostId), '') WHERE ChannelId="";` • `CREATE INDEX idx_reactions_channel_id ON Reactions(ChannelId) LOCK=NONE;` PostgreSQL: • `ALTER TABLE reactions ADD COLUMN IF NOT EXISTS channelid varchar(26) NOT NULL DEFAULT '';` • `UPDATE reactions SET channelid = COALESCE((select channelid from posts where posts.id = reactions.postid), '') WHERE channelid='';` • `CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_reactions_channel_id on reactions (channelid);`
v7.0	**IMPORTANT:** Session length configuration settings have changed from using a unit of *days* to *hours*. Instances using a config.json file or a database configuration for the following values should be automatically migrated to the new units, but instances using environment variables must make the following changes: 1. replace `MM_SERVICESETTINGS_SESSIONLENGTHWEBINDAYS` with `MM_SERVICESETTINGS_SESSIONLENGTHWEBINHOURS` (x24 the value). 2. replace `MM_SERVICESETTINGS_SESSIONLENGTHMOBILEINDAYS` with `MM_SERVICESETTINGS_SESSIONLENGTHMOBILEINHOURS` (x24 the value).

If you're upgrading from a version earlier than...	Then...
	3. replace `MM_SERVICESETTINGS_SESSIONLENGTHSSOINDAYS` with `MM_SERVICESETTINGS_SESSIONLENGTHSSOINHOURS` (x24 the value).
	MySQL self-hosted customers may notice the migration taking longer than usual when having a large number of rows in FileInfo table. For MySQL, it takes around 19 seconds for a table of size 700,000 rows. The time required for PostgreSQL is negligible. The testing was performed on a machine with specifications of `CPU - Intel i7 6-cores @ 2.6 GHz` and `Memory - 16 GB`.
	When a new configuration setting via **System Console > Experimental > Features > Enable App Bar** is enabled, all channel header icons registered by plugins will be moved to the new Apps Bar, even if they do not explicitly use the new registry function to render a component there. The setting for Apps Bar defaults to `false` for self-hosted deployments.
	The value of `ServiceSettings.TrustedProxyIPHeader` defaults to empty from now on. A previous bug prevented this from happening in certain conditions. Customers are requested to check for these values in their config and set them to nil if necessary. See more details here.
	Collapsed Reply Threads is now generally available and enabled by default for new Mattermost servers. For servers upgrading to v7.0 and later, please reference [this article (https://support.mattermost.com/hc/en-us/articles/6880701948564-Administrator-s-guide-to-enabling-Collapsed-Reply-Threads)](https://support.mattermost.com/hc/en-us/articles/6880701948564-Administrator-s-guide-to-enabling-Collapsed-Reply-Threads) for more information and guidance on enabling the feature.
v6.7	

If you're upgrading from a version earlier than...	Then...
	New schema changes were introduced in the form of a new index. The following summarizes the test results measuring how long it took for the database queries to run with these schema changes: MySQL 7M Posts - ~17s (Instance: db.r5.xlarge) MySQL 9M Posts - 2min 12s (Instance: db.r5.large) Postgres 7M Posts - ~9s (Instance: db.r5.xlarge) For customers wanting a zero downtime upgrade, they are encouraged to apply this index prior to doing the upgrade. This is fully backwards compatible and will not acquire any table lock or affect any existing operations on the table when run manually. Else, the queries will run during the upgrade process and will lock the table in non-MySQL environments. Run the following to apply this index: For MySQL: `CREATE INDEX idx_posts_create_at_id on Posts(CreateAt, Id) LOCK=NONE;` For Postgres: `CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_posts_create_at_id on posts(createat, id);`
	In v6.7.1, the value of `ServiceSettings.TrustedProxyIPHeader` defaults to empty from now on. A previous bug prevented this from happening in certain conditions. Customers are requested to check for these values in their config and set them to nil if necessary. See more details here.
v6.6	The Apps Framework protocol for binding/form submissions has changed, by separating the single *call* into separate *submit*, *form*, *refresh* and *lookup* calls. If any users have created their own Apps, they have to be updated to the new system.

If you're upgrading from a version earlier than...	Then...
	Channel admins can now configure certain actions to be executed automatically based on trigger conditions without writing any code. Users running an older Playbooks release need to upgrade their Playbooks instance to at least v1.26 to take advantage of the channel actions functionality.
	In v6.6.2, the value of `ServiceSettings.TrustedProxyIPHeader` defaults to empty from now on. A previous bug prevented this from happening in certain conditions. Customers are requested to check for these values in their config and set them to nil if necessary. See more details here.
v6.5	The `mattermost version` CLI command does not interact with the database anymore. Therefore the database version is not going to be printed. Also, the database migrations are not going to be applied with the version sub command. A new db migrate sub command is added to enable administrators to trigger migrations.
	In v6.5.2, the value of `ServiceSettings.TrustedProxyIPHeader` defaults to empty from now on. A previous bug prevented this from happening in certain conditions. Customers are requested to check for these values in their config and set them to nil if necessary. See more details here.
v6.4	A new schema migration system has been introduced, so we strongly recommend backing up the database before updating the server to this version. The new migration system will run through all existing migrations to record them to a new table. This will only happen for the first run in order to migrate the application to the new system. The table where the migration information is stored is called

If you're upgrading from a version earlier than...	Then...
	`db_migrations`. Additionally, a `db_lock` table is used to prevent multiple installations from running migrations in parallel. In case of an error while applying the migrations, please check this table first. Any downtime depends on how many records the database has and whether there are missing migrations in the schema. If you encounter an issue please file [an Issue (https://github.com/mattermost/mattermost-server/issues)](https://github.com/mattermost/mattermost-server/issues) by including the failing migration name, database driver/version, and the server logs. On MySQL, if you encounter an error "Failed to apply database migrations" when upgrading to v6.4.0, it means that there is a mismatch between the table collation and the default database collation. You can manually fix this by changing the database collation with `ALTER DATABASE <YOUR_DB_NAME> COLLATE = 'utf8mb4_general_ci',`. Then do the server upgrade again and the migration will be successful. It has been commonly observed on MySQL 8+ systems to have an error `Error 1267: Illegal mix of collations` when upgrading due to changing the default collation. This is caused by the database and the tables having different collations. If you get this error, please change the collations to have the same value with, for example, `ALTER DATABASE <db_name> COLLATE = '<collation>'`.
	The new migration system requires the MySQL database user to have additional *EXECUTE*, *CREATE ROUTINE*, *ALTER ROUTINE* and *REFERENCES* privileges to run schema migrations.
v6.3	In v6.3.3, the default for `ThreadAutoFollow` has been changed to `false`. This does not affect existing configurations where this value is already set to `true`.

If you're upgrading from a version earlier than...	Then...
	In v6.3.9, the value of `ServiceSettings.TrustedProxyIPHeader` defaults to empty from now on. A previous bug prevented this from happening in certain conditions. Customers are requested to check for these values in their config and set them to nil if necessary. See more details here.
v6.2	Channel results in the channel autocomplete will include private channels. Customers using Bleve or Elasticsearch for autocomplete will have to reindex their data to get the new results. Since this can take a long time, we suggest disabling autocomplete and running indexing in the background. When this is complete, re-enable autocomplete. Note Only channel members see private channel names in autocomplete results.
	In v6.2.3, the default for `ThreadAutoFollow` has been changed to `false`. This does not affect existing configurations where this value is already set to `true`.
	Mattermost Boards requires `EnableReliableWebSockets` setting to be manually set to `true` for real-time updates to appear correctly.
v6.1	Please refer to [the schema migration analysis (https://gist.github.com/streamer45/997b726a86b5d2a624ac2af435a66086)](https://gist.github.com/streamer45/997b726a86b5d2a624ac2af435a66086) when upgrading to v6.1.

If you're upgrading from a version earlier than...	Then...
	The Bleve index has been updated to use the scorch index type. This new default index type features some efficiency improvements which means that the indexes use significantly less disk space. To use this new type of index, after upgrading the server version, run a purge operation and then a reindex from the Bleve section of the System Console. Bleve is still compatible with the old indexes, so the currently indexed data will work fine if the purge and reindex is not run.
	A composite index has been added to the jobs table for better query performance. For some customers with a large jobs table, this can take a long time, so we recommend adding the index during off-hours, and then running the migration. A table with more than 1 million rows can be considered as large enough to be updated prior to the upgrade. • For PostgreSQL: `create index concurrently idx_jobs_status_type on jobs (status,type);` • For MySQL: `create index idx_jobs_status_type on Jobs (Status,Type);`
	In v6.1.3, the default for `ThreadAutoFollow` has been changed to `false`. This does not affect existing configurations where this value is already set to `true`.
	Mattermost Boards requires `EnableReliableWebSockets` setting to be manually set to `true` for real-time updates to appear correctly.
v6.0	

If you're upgrading from a version earlier than...	Then...
	Longer migration times can be expected. • See this document (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055) for the estimated upgrade times with datasets of 10+ million posts. • See this document (https://gist.github.com/streamer45/868c451164f6e8069d8b398685a31b6e) for the estimated upgrade times with datasets of 70+ million posts. The field type of Data in `model.ClusterMessage` was changed to []byte from string. A v6 server is incompatible to run along with a v5 server in a cluster. Customers upgrading from 5.x to 6.x cannot do a High Availability upgrade. While upgrading, it is required that no other v5 server runs when a v6 server is brought up. A v6 server will run significant database schema changes that can cause a large startup time depending on the dataset size. Zero downtime will not be possible, but depending on the efforts made during the migration process, it can be minimized to a large extent. 1. Low effort, long downtime - This is the usual process of starting a v6 server normally. This has two implications: during the migration process, various tables will be locked which will render those tables read-only during that period. Secondly, once the server finishes migration and starts the application, no other v5 server can run in the cluster. 2. Medium effort, medium downtime - This process will require SQL queries to be executed manually on the server. To avoid causing a table lock, a customer can choose to use the pt-online-schema-change tool for MySQL. For Postgres, the table locking is very minimal. The advantage is that since there are a lot of queries, the customer can take their own time to run individual queries during off-hours. All queries except #11 are safe to be executed this way. Then the usual method of (1) can be followed.

If you're upgrading from a version earlier than...	Then...
	3. High effort, low downtime - This process requires everything of (2), plus it tries to minimize the impact of query #11. We can do this by following step 2, and then starting v6 along with a running v5 server, and then monitor the application logs. As soon as the v6 application starts up, we need to bring down a v5 node. This minimizes the downtime to only a few seconds. It is recommended to start Mattermost directly and not through systemctl to avoid the server process getting killed during the migration. This can happen since the value of `TimeoutStartSec` in the provided systemctl service file is set to one hour. Customers using the Mattermost Kubernetes operator should be aware of the above migration information and choose the path that is most appropriate for them. If (1) is acceptable, then the normal upgrade process using the operator will suffice. For minimum downtime, follow (2) before using the operator to update Mattermost following the normal upgrade process.
	While trying to upgrade to a Mattermost version >= 6.0.x, you might encounter the following error: `Failed to alter column type. It is likely you have invalid JSON values in the column. Please fix the values manually and run the migration again.` This means that the particular column has some invalid JSON values which need to be fixed manually. A common fix that is known to work is to wipe out all `\u0000` characters. Please follow these steps: 1. Check the affected values by: `SELECT COUNT(*) FROM <table> WHERE <column> LIKE '%\u0000%';` 2. If you get a count more than 0, check those values manually, and confirm they are okay to be removed.

If you're upgrading from a version earlier than...	Then...
	3. Remove them by `UPDATE <table> SET <column> = regexp_replace(<column>, '\\u0000', '', 'g') where <column> like '%\u0000%';` Then try to start Mattermost again.
	Please see unsupported legacy releases documentation for a list of deprecations in this release.
	Focalboard plugin has been renamed to Mattermost Boards, and v0.9.1 (released with Mattermost v6.0) is now enabled by default.
	The advanced logging configuration schema changed. This is a breaking change relative to 5.x. See updated documentation.
	The existing theme names and colors, including "Mattermost", "Organization", "Mattermost Dark", and "Windows Dark" have been updated to the new "Denim", "Quartz", "Indigo", and "Onyx" theme names and colors, respectively. Anyone using the existing themes will see slightly modified theme colors after their server or workspace is upgraded. The theme variables for the existing "Mattermost", "Organization", "Mattermost Dark", and "Windows Dark" themes will still be accessible in our documentation. A custom theme can be created with these theme variables if desired. Custom themes are unaffected by this change.
	Some breaking changes to plugins are included: • Support for left-hand side-specific bot icons was dropped. • Removed a deprecated "Backend" field from the plugin manifest.

If you're upgrading from a version earlier than...	Then...
	• Converted the "Executables" field in the plugin manifest to a map.
	Mattermost Boards requires `EnableReliableWebSockets` setting to be manually set to `true` for real-time updates to appear correctly.
v5.38.0	The "config watcher" (the mechanism that automatically reloads the `config.json` file) has been removed in favor of the `mmctl config reload` command, which must be run to apply configuration changes after they are made on disk. This change improves configuration performance and robustness.
	v5.38 adds fixes for some of the incorrect mention counts and unreads around threads and channels since the introduction of Collapsed Reply Threads (Beta). This fix is done through a SQL migration, and it may take several minutes to complete for large databases. The `fixCRTChannelMembershipCounts` fix takes 1 minute and 20 seconds for a database containing approximately four million channel memberships and about 130,000 channels. The `fixCRTThreadCountsAndUnreads` fix takes about 3 minutes and 30 seconds for a database containing 56367 threads, 124587 thread memberships, and 220801 channel memberships. These are on MySQL v5.6.51.
	Focalboard v0.8.2 (released with Mattermost v5.38.0) requires Mattermost v5.37+ due to the new database connection system.
v5.37.0	The `platform` binary and "–platform" flag have been removed. If you are using the "–platform" flag or are using the `platform` binary directly to run the Mattermost server application via a systemd file or custom script, you will be required to use only the mattermost binary.

If you're upgrading from a version earlier than...	Then...
	Collapsed Reply Threads (https://mattermost.com/blog/collapsed-reply-threads-beta/) are available as Beta in Mattermost Server v5.37 and later. It's expected that you may experience bugs as we stabilize the feature. In particular, please be aware of the known issues documented here.
	v5.37 adds support for emoji standard v13.0. If you have added a custom emoji in the past that uses one of the new system names, then it is going to get overwritten by the system emoji. The workaround is to change the custom emoji name.
	Parts of Incident Collaboration are now available to all Mattermost editions. As part of this update, Incident Collaboration will require a minimum server version of v5.37. To learn more about what is available in each edition, visit our pricing page (https://mattermost.com/pricing).
	In v5.37.8, the default for `ThreadAutoFollow` has been changed to `false`. This does not affect existing configurations where this value is already set to `true`.
v5.36.0	Gossip clustering mode is now in General Availability and is no longer available as an option. All cluster traffic will always use the gossip protocol. The config setting `UseExperimentalGossip` has no effect and has only been kept for compatibility purposes. The setting to use gossip has been removed from the System Console.

If you're upgrading from a version earlier than...	Then...
	Note For High Availability upgrades, all nodes in the cluster must use a single protocol. If an existing system is not currently using gossip, one node in a cluster can't be upgraded while other nodes in the cluster use an older version. Customers must either use gossip for their High Availability upgrade, or customers must shut down all nodes, perform the upgrade, and then bring all nodes back up.
	To enable Focalboard, open the Marketplace from the sidebar menu, install the Focalboard plugin, then click on **Configure**, enable it, and save. Update your NGINX or Apache web proxy config.
v5.35.0	Due to the introduction of backend database architecture required for upcoming new features, Shared Channels and Collapsed Reply Threads, the performance of the migration process for the v5.35 release (May 16, 2021) has been noticeably affected. Depending on the size, type, and version of the database, longer than usual upgrade times should be expected. This can vary from a couple of minutes (average case) to hours (worst case, MySQL 5.x only). A moderate to significant spike in database CPU usage should also be expected during this process. More details on the performance impact of the migration and possible mitigation strategies (https://gist.github.com/streamer45/9aee4906639a49ebde68b2f3c0f924c1) are available.
	The existing password generation logic used during the bulk user import process was comparatively weak. Hence it's advised for admins to immediately reset the passwords for all the users who were

If you're upgrading from a version earlier than...	Then...
	generated during the bulk import process and whose password has not been changed even once.
	v5.35.0 introduces a new feature to search for files. Search results for files shared in the past may be incomplete until a content extraction command is executed to extract and index the content of files already in the database. Instances running Elasticsearch or Bleve search backends will also need to execute a Bulk Indexing after the content extraction is complete. Please see more details in [this blog post (https://mattermost.com/blog/file-search/)](https://mattermost.com/blog/file-search/).
v5.34.1	v5.34.1 fixes an issue where upgrading to v5.34.0 runs a migration that can cause timeouts on MySQL installations. Upgrading to v5.34.1 may also execute missing migrations that were scheduled for v5.32.0. These additions can be lengthy on very big MySQL (version 5.x) installations. • Altering of `Posts.FileIds` type (PostgreSQL only) • Added new column `ThreadMemberships.UnreadMentions` • Added new column `Channels.Shared` • Added new column `Reactions.UpdateAt` • Added new column `Reactions.DeleteAt`
v5.33.0	Deleting a reaction is now a soft delete in the `Reactions` table. A schema update is required and may take up to 15 seconds on first run with large data sets.
	WebSocket handshakes done with HTTP version lower than 1.1 will result in a warning, and the server will transparently upgrade the

If you're upgrading from a version earlier than...	Then...
	version to 1.1 to comply with the WebSocket RFC. This is done to work around incorrect Nginx (and other proxy) configs that do not set the `proxy_http_version` directive to 1.1. This facility will be removed in a future Mattermost version and it is strongly recommended to fix the proxy configuration to correctly use the WebSocket protocol.
v5.32.0	`ExperimentalChannelOrganization`, `EnableXToLeaveChannelsFromLHS`, `CloseUnusedDirectMessages`, and `ExperimentalHideTownSquareinLHS` settings are only functional if the Legacy Sidebar (`EnableLegacySidebar`) is enabled since they are not compatible with the new sidebar experience. `ExperimentalChannelSidebarOrganization` has been deprecated, since the new sidebar is now enabled for all users (https://mattermost.com/blog/custom-collapsible-channel-categories/).
	Breaking changes to the Golang client API were introduced: `GetPostThread`, `GetPostsForChannel`, `GetPostsSince`, `GetPostsAfter`, `GetPostsBefore`, and `GetPostsAroundLastUnread` now require an additional collapsedThreads parameter to be passed. Any client making use of these functions will need to update them when upgrading its dependencies.
	A breaking change was introduced when upgrading the Go version to v1.15.5 (https://go.dev/doc/go1.15) where user logins fail with AD/LDAP Sync when the certificate of the LDAP Server has no Subject Alternative Name (SAN) in it. Creating a new certificate on the AD/LDAP Server with the SAN inside fixes this.
	TLS versions 1.0 and 1.1 have been deprecated by browser vendors. Starting in Mattermost Server v5.32 (February 16), mmctl returns an error when connected to Mattermost servers deployed with these TLS versions. System admins will need to explicitly add a flag in their

If you're upgrading from a version earlier than...	Then...
	commands to continue to use them. We recommend upgrading to TLS version 1.2 or higher.
v5.31.0	For Mobile Apps v1.42.0+, the minimum server version is set to 5.31.3 as 5.31.3 fixed an issue (https://docs.mattermost.com/deployment-guide/legacy-self-hosted-changelog.html#release-v5-31-esr) where the server version was reported as v5.30.0.
v5.29.0	A new configuration setting `ThreadAutoFollow` has been added to support Collapsed Reply Threads (https://docs.google.com/presentation/d/1QSrPws3N8AMSjVyOKp15FKT7O0fGMSx8YidjSDS4Wng/edit#slide=id.g2f0aecc189_0_245) releasing in beta in Q1 2021. This setting is enabled by default and may affect server performance. It is recommended to review our documentation on hardware requirements to ensure your servers are appropriately scaled for the size of your user base.
	Disabled the xmlsec1-based SAML library in favor of the re-enabled and improved SAML library.
v5.28.0	Now when the service crashes, it will generate a coredump instead of just dumping the stack trace to the console. This allows us to preserve the full information of the crash to help with debugging it. For more information about coredumps, please see: https://man7.org/linux/man-pages/man5/core.5.html.
	In-product notices have been introduced to keep system admins and end users informed of the latest product enhancements available in

If you're upgrading from a version earlier than...	Then...
	new server and desktop versions. Learn more about in-product notices and how to disable them in our documentation.
	Disabled the xmlsec1-based SAML library in favor of the re-enabled and improved SAML library.
v5.27.0	Disabled the xmlsec1-based SAML library in favor of the re-enabled and improved SAML library.
v5.26.0	In v5.26, Elasticsearch indexes needed to be recreated. Admins should re-index Elasticsearch using the **Purge index** and then **Index now** button so that all the changes will be included in the index. Systems may be left with a limited search during the indexing, so it should be done during a time when there is little to no activity because it may take several hours.
	An `EnableExperimentalGossipEncryption` option was added under `ClusterSettings`. If set to `true`, and `UseExperimentalGossip` is also `true`, all communication through the cluster using the gossip protocol will be encrypted. The encryption uses `AES-256` by default, and it is not kept configurable by design. However, if one wishes, they can set the value in Systems table manually for the `ClusterEncryptionKey` row. A key is a byte array converted to base64. It should be either 16, 24, or 32 bytes to select AES-128, AES-192, or AES-256. To update the key, one can execute: `UPDATE Systems SET Value='<value>' WHERE Name='ClusterEncryptionKey';` in MySQL and `UPDATE systems SET value='<value>' WHERE name='ClusterEncryptionKey'` for PostgreSQL. For any change in this config setting to take effect, the whole cluster must be shut down first. Then the config change made, and then

If you're upgrading from a version earlier than...	Then...
	restarted. In a cluster, all servers either will completely use encryption or not. There cannot be any partial usage.
	SAML Setting "Use Improved SAML Library (Beta)" was forcefully disabled.
	PostgreSQL ended long-term support for version 9.4 in February 2020 (https://www.postgresql.org/support/versioning). From v5.26 Mattermost officially supports PostgreSQL version 10 as PostgreSQL 9.4 is no longer supported. New installs will require PostgreSQL 10+. Previous Mattermost versions, including our current ESR, will continue to be compatible with PostgreSQL 9.4. PostgreSQL 9.4 and all 9.x versions are now fully deprecated in our v5.30 release (December 16, 2020). Follow the Upgrading Section within the PostgreSQL documentation.
v5.25.0	Some incorrect instructions regarding SAML setup with Active Directory ADFS for setting the "Relying Party Trust Identifier" were corrected. Although the settings will continue to work, it is encouraged that you modify those settings.
	Disabled the xmlsec1-based SAML library in favor of the re-enabled and improved SAML library.

If you're upgrading from a version earlier than...	Then...
v5.24.0	A new configuration setting, `ExtendSessionLengthWithActivity` automatically extends sessions to keep users logged in if they are active in their Mattermost apps. It is recommended to enable this setting to improve user experience if compliant with your organization's policies. [Learn more here (https://mattermost.com/blog/session-expiry-experience)](https://mattermost.com/blog/session-expiry-experience).
	The `mattermost_http_request_duration_seconds` histogram metric (in Enterprise Edition) has been removed. This information was already captured by `mattermost_api_time`, which also contains the API handler name, HTTP method, and the response code. As an example, if you are using `rate(mattermost_http_request_duration_seconds_sum{server=~"$var"}[5m]) / rate(mattermost_http_request_duration_seconds_count{server=~"$var"}[5m])` to measure average call duration, it needs to be replaced with `sum(rate(mattermost_api_time_sum{server=~"$var"}[5m])) by (instance) / sum(rate(mattermost_api_time_count{server=~"$var"}[5m])) by (instance)`.
	Due to fixing performance issues related to emoji reactions, the performance of the upgrade has been affected in that the schema upgrade now takes more time in environments with lots of reactions in their database. These environments are recommended to perform the schema migration during low usage times and potentially in advance of the upgrade. Since this migration happens before the Mattermost server is fully launched, non-High Availability installs will be unreachable during this time. The migration is a single line of SQL and can be applied directly to the database through the MySQL/PSQL command line clients if you prefer to decouple this from restarting the Mattermost server. It is fully

If you're upgrading from a version earlier than...	Then...
	backwards compatible so the schema change can be applied to any previous version of Mattermost without issue. During the time the schema change is running (~30s per million rows in the Reactions table), if end users attempt to react to posts, the emoji reactions will not load for end users. MySQL: `ALTER TABLE Reactions DROP PRIMARY KEY, ADD PRIMARY KEY (PostId, UserId, EmojiName);` PostgreSQL: `ALTER TABLE reactions DROP CONSTRAINT reactions_pkey, ADD PRIMARY KEY (PostId, UserId, EmojiName);`
	On mobile apps, users will not be able to see LDAP group mentions (E20 feature) in the autocomplete dropdown. Users will still receive notifications if they are part of an LDAP group. However, the group mention keyword will not be highlighted.
	SAML Setting "Use Improved SAML Library (Beta)" was forcefully disabled. Follow instructions at https://docs.mattermost.com/administration-guide/onboard/sso-saml.html for enabling SAML using the feature-equivalent `xmlsec1` utility.
v5.22.0	Due to fixing performance issues related to emoji reactions, the performance of the upgrade has been affected in that the schema upgrade now takes more time in environments with lots of reactions in their database. These environments are recommended to perform the schema migration during low usage times and potentially in advance of the upgrade. Since this migration happens before the Mattermost server is fully launched, non-High Availability installs will be unreachable during this time. The migration is a single line of SQL and can be applied directly to the database through the MySQL/PSQL command line clients if you prefer

If you're upgrading from a version earlier than...	Then...
	to decouple this from restarting the Mattermost server. It is fully backwards compatible so the schema change can be applied to any previous version of Mattermost without issue. During the time the schema change is running (~30s per million rows in the Reactions table), if end users attempt to react to posts, the emoji reactions will not load for end users. MySQL: `ALTER TABLE Reactions DROP PRIMARY KEY, ADD PRIMARY KEY (PostId, UserId, EmojiName);` Postgres: `ALTER TABLE reactions DROP CONSTRAINT reactions_pkey, ADD PRIMARY KEY (PostId, UserId, EmojiName);`
	The Channel Moderation Settings feature is supported on mobile app versions v1.30 and later. In earlier versions of the mobile app, users who attempt to post or react to posts without proper permissions will see an error.
	Direct access to the `Props` field in the `model.Post` structure has been deprecated. The available `GetProps()` and `SetProps()` methods should now be used. Also, direct copy of the `model.Post` structure must be avoided in favor of the provided `Clone()` method.
	SAML Setting "Use Improved SAML Library (Beta)" was forcefully disabled. Follow instructions at https://docs.mattermost.com/administration-guide/onboard/sso-saml.html for enabling SAML using the feature-equivalent `xmlsec1` utility.
v5.21.0	Honour key value expiry in KVCompareAndSet, KVCompareAndDelete, and KVList. We also improved handling of plugin key value race conditions and deleted keys in Postgres.

If you're upgrading from a version earlier than...	Then...
	SAML Setting "Use Improved SAML Library (Beta)" was forcefully disabled. Follow instructions at https://docs.mattermost.com/administration-guide/onboard/sso-saml.html for enabling SAML using the feature-equivalent `xmlsec1` utility.
v5.20.0	Any [pre-packaged plugin (https://developers.mattermost.com/integrate/admin-guide/admin-plugins-beta/#pre-packaged-plugins)](https://developers.mattermost.com/integrate/admin-guide/admin-plugins-beta/#pre-packaged-plugins) that is not enabled in the `config.json` will no longer install automatically, but can continue to be installed via the [Plugin Marketplace (https://developers.mattermost.com/integrate/admin-guide/admin-plugins-beta/#plugin-marketplace)](https://developers.mattermost.com/integrate/admin-guide/admin-plugins-beta/#plugin-marketplace).
	Boolean elements from interactive dialogs are no longer serialized as strings. While we try to avoid breaking changes, this change was necessary to allow both the web and mobile apps to work with the boolean elements introduced with v5.16.
v5.19.0	`LockTeammateNameDisplay` setting was moved to Enterprise Edition E20 as it was erroneously available in Team Edition and Enterprise Edition E10.
v5.18.0	Marking a post unread from the mobile app requires v1.26 or later. If using v5.18, but mobile is on v1.25 or earlier, marking a post unread from webapp/desktop will only be reflected on mobile the next time the app launches or is brought to the foreground.
	The Go module path of `mattermost-server` was changed to comply with the Go module version specification. Developers using Go modules with `mattermost-server` as a dependency must change the module and

If you're upgrading from a version earlier than...	Then...
	import paths to `github.com/mattermost/mattermost-server/v5` when upgrade this dependency to *v5.18*. See https://go.dev/blog/v2-go-modules for further information.
	Removed `Team.InviteId` from the related Websocket event and sanitized it on all team API endpoints for users without invite permissions.
	Removed the ability to change the type of a channel using the `PUT /channels/{channel_id}` API endpoint. The new `PUT /channels/{channel_id}/privacy` endpoint should be used for that purpose.
v5.16.0	Support for Internet Explorer (IE11) is removed. See [this forum post (https://forum.mattermost.com/t/mattermost-is-dropping-support-for-internet-explorer-ie11-in-v5-16/7575)](https://forum.mattermost.com/t/mattermost-is-dropping-support-for-internet-explorer-ie11-in-v5-16/7575) to learn more.
	The [Mattermost Desktop v4.3.0 (https://github.com/mattermost/desktop/blob/master/CHANGELOG.md)](https://github.com/mattermost/desktop/blob/master/CHANGELOG.md) release includes a change to how desktop notifications are sent sent from non-secure URLs `http://`. Organizations using non-secure Mattermost Servers `http://` will need to update to Mattermost Server versions 5.16.0+, 5.15.1, 5.14.4 or 5.9.5 (ESR) to continue receiving desktop notifications when using Mattermost Desktop v4.3.0 or later.
	When enabling Guest Accounts, all users who have the ability to invite users will be able to invite guests by default. System admins will need to remove this permission on each role via **System Console > Permissions Schemes**. In Mattermost Server version 5.17, the system admin will be the only role to automatically get the invite guest

If you're upgrading from a version earlier than...	Then...
	permission, however the fix will not be applicable in 5.16 due to database migration processes.
v5.14.0	Webhooks are now only displayed if the user is the creator of the webhook or a system administrator.
	With the update from Google+ to Google People, system admins need to ensure the `GoogleSettings.Scope` config.json setting is set to `profile email` and `UserAPIEndpoint` setting should be set to `https://people.googleapis.com/v1/people/me?personFields=names,emailAddresses,nicknames,metadata` per updated documentation.
v5.12.0	If your plugin uses the `DeleteEphemeralMessage` plugin API, update it to accept a `postId string` parameter. See documentation to learn more.
	Image link and YouTube previews do not display unless **System Console > Enable Link Previews** is enabled. Please ensure that your Mattermost server is connected to the internet and has network access to the websites from which previews are expected to appear. [Learn more here (https://forum.mattermost.com/t/link-previews-managed-server-side-in-v5-12-and-later/7712)](https://forum.mattermost.com/t/link-previews-managed-server-side-in-v5-12-and-later/7712).
	`ExperimentalEnablePostMetadata` setting was removed. Post metadata, including post dimensions, is now stored in the database to correct scroll position and eliminate scroll jumps as content loads in a channel.
	Added the ability to enforce the administration of teams/channels with Group Sync. If Group Sync is enabled, all team and channel admin

If you're upgrading from a version earlier than...	Then...
	designations will be lost upon upgrade. It is highly recommended that prior to upgrading, Team and channel admins are added to admin-specific LDAP groups corresponding to their teams and channels. After upgrading, those groups will need to be role-synced to the Team or channel admin role.
v5.11.0	If your integration uses `Update.Props == nil` to clear `Props`, this will no longer work in 5.11+. Instead, use `Update.Props == {}` to clear properties. This change was made because `Update.Props == nil` unintentionally cleared all `Props`, such as the profile picture, instead of preserving them.
v5.10.0	`SupportedTimezonesPath` setting in config.json and changes to timezones in the UI based on the `timezones.json` file was removed. This was made to support storing configurations in the database.
v5.9.0	If `DisableLegacyMfa` setting in `config.json` is set to `true` and multi-factor authentication is enabled, ensure your users have upgraded to mobile app version 1.17 or later. Otherwise, users who have MFA enabled may not be able to log in successfully. If the setting is not defined in the `config.json` file, the `DisableLegacyMfa` setting is set to `false` by default to ensure no breaking changes. We recommend setting `DisableLegacyMfa` to `true` for additional security hardening.
	The public IP of the Mattermost application server is considered a reserved IP for additional security hardening in the context of untrusted

If you're upgrading from a version earlier than...	Then...
	external requests such as Open Graph metadata, webhooks, or slash commands. See documentation for additional information.
v5.8.0	The local image proxy has been added, and images displayed within the client are now affected by the `AllowUntrustedInternalConnections` setting. See documentation for more details if you have trouble loading images.
v5.6.0	Built-in WebRTC is removed. See here for more details (https://forum.mattermost.com/t/built-in-webrtc-video-and-audio-calls-removed-in-v5-6-in-favor-of-open-source-plugins/5998).
	If `EnablePublicChannelsMaterialization` setting in `config.json` is set to `false`, an offline migration prior to upgrade may be required to synchronize the materialized table for public channels to increase channel search performance in the channel switcher (CTRL/CMD+K), channel autocomplete (~), and elsewhere in the UI. Use the following steps: 1. Shut down your application servers. 2. Connect to your Mattermost database. 3. Execute the following queries: `DELETE FROM PublicChannels; INSERT INTO PublicChannels (Id, DeleteAt, TeamId, DisplayName, Name, Header, Purpose) SELECT c.Id, c.DeleteAt, c.TeamId, c.DisplayName, c.Name, c.Header, c.Purpose FROM Channels c WHERE c.Type = 'O';` The queries above rebuild the materialized `PublicChannels` table without modifying the authoritative `Channels` table.

If you're upgrading from a version earlier than...	Then...
	Note This migration is not required if the experimental `PublicChannels` feature was never disabled. This feature launched in Mattermost v5.4 with a temporary flag to disable should an issue arise, but nothing prompted doing so. If you did not modify this setting, there is no need to perform this migration.
v5.4.0	Mattermost mobile app version 1.13+ is required. File uploads will fail on earlier mobile app versions.
	In certain upgrade scenarios the new **Allow Team Administrators to edit others posts** setting under **General** then **Users and Teams** may be set to **True** while the Mattermost default in 5.1 and earlier and with new 5.4+ installations is **False**.
v5.3.0	Those servers with Elasticsearch enabled will notice that hashtag search is case-sensitive.
v5.2.0	Those servers upgrading from v4.1 - v4.4 directly to v5.2 or later and have Jira enabled will need to re-enable the Jira plugin after an upgrade.
v5.1.0	`mattermost export` CLI command is renamed to `mattermost export schedule`. Make sure to update your scripts if you use this command.

If you're upgrading from a version earlier than...	Then...
v5.0.0	All API v3 endpoints are removed. See documentation to learn how to migrate your integrations to API v4.
	`platform` binary is renamed to `mattermost` for a clearer install and upgrade experience. You should point your `systemd` service file at the new `mattermost` binary. All command line tools, including the bulk loading tool and developer tools, are also be renamed from `platform` to `mattermost`.
	A Mattermost user setting to configure desktop notification duration in **Account Settings > Notifications > Desktop Notifications** is removed.
	Slash commands configured to receive a GET request will have the payload being encoded in the query string instead of receiving it in the body of the request, consistent with standard HTTP requests. Although unlikely, this could break custom slash commands that use GET requests incorrectly.
	A new `config.json` setting to whitelist types of protocols for auto-linking will be added. If you rely on custom protocols auto-linking in Mattermost, whitelist them in `config.json` before upgrading.

If you're upgrading from a version earlier than...	Then...
	A new `config.json` setting to disable the permanent APIv4 delete team parameter (https://api.mattermost.com/#tag/teams%2Fpaths%2F~1teams~1%7Bteam_id%7D%2Fput) is added. The setting will be off by default for all new and existing installs, except those deployed on GitLab Omnibus. If you reply on the APIv4 parameter, enable the setting in `config.json` before upgrading.
	An unused `ExtraUpdateAt` field will be removed from the channel modal.

	This release includes support for post messages longer than the default of 4000 characters, but may require a manual database migration. This migration is entirely optional, and need only be done if you want to enable post messages up to 16383 characters. For many installations, no migration will be required, or the old limit remains sufficient.
	
	To check your current post limit after upgrading to 5.0.0, look for a log message on startup:
	
	`[2018/03/27 09:08:00 EDT] [INFO] Post.Message supports at most 16383 characters (65535 bytes)`
	
	As of 5.0.0, the maximum post message size is 16383 (multi-byte) characters. If your logs show a number less than this limit and you want to enable longer post messages, you will need to manually migrate your database as described below. This migration can be slow for larger `Posts` tables, so it's best to schedule this upgrade during off-peak hours.
	
	To migrate a MySQL database, connect to your database and run the following:
	
	`ALTER TABLE Posts MODIFY COLUMN Message TEXT;`

If you're upgrading from a version earlier than...	Then...
	To migrate a PostgreSQL database, connect to your database and run the following: `ALTER TABLE Posts ALTER COLUMN Message TYPE VARCHAR(65535);` Restart your Mattermost instances.
	Deployments on Enterprise E20 will need to enable `RunJobs` in the `config.json` and allow the permissions migration to complete before using Team Override Schemes.
v4.10.0	Old email invitation links will no longer work due to a bug fix where teams could be re-joined via the link. Team invite links copied from the Team Invite Link dialog, password reset links and email verification links are not affected and are still valid.
	Server logs written to **System Console > Logs** and to the `mattermost.log` file specified in **System Console > Logging > File Log Directory** now use JSON formatting. If you have built a tool that parses the server logs and sends them to an external system, make sure it supports the JSON format.
	Team icons with transparency will be filled with a white background in the Team sidebar.
	Those servers with SAML authentication enabled should upgrade during non-peak hours. SAML email addresses are migrated to lowercase to prevent login issues, which could result in longer than usual upgrade time.

If you're upgrading from a version earlier than...	Then...
	If you use PostgreSQL database and the password contains special characters (e.g. `[]`), escape them in your password, e.g., xxx[]xxx will be xxx%5B%5Dxxx.
v4.9.0	To improve the production use of Mattermost with Docker, the Docker image is now running a as non-root user and listening on port 8000. Please read the [upgrade instructions (https://github.com/mattermost/mattermost-docker#upgrading-mattermost-to-49)](https://github.com/mattermost/mattermost-docker#upgrading-mattermost-to-49) for important changes to existing installations.
	Several configuration settings have been migrated to roles in the database and changing their `config.json` values no longer takes effect. These permissions can still be modified by their respective System Console settings as before. The affected `config.json` settings are: • `RestrictPublicChannelManagement`, • `RestrictPrivateChannelManagement`, • `RestrictPublicChannelCreation`, • `RestrictPrivateChannelCreation`, • `RestrictPublicChannelDeletion`, • `RestrictPrivateChannelDeletion`, • `RestrictPrivateChannelManageMembers`, • `EnableTeamCreation`, • `EnableOnlyAdminIntegrations`, • `RestrictPostDelete`,

If you're upgrading from a version earlier than...	Then...
	• `AllowEditPost`, • `RestrictTeamInvite`, • `RestrictCustomEmojiCreation`.
	The behavior of the `config.json` setting `PostEditTimeLimit` has been updated to accommodate the migration to a roles based permission system. When post editing is permitted, set `"PostEditTimeLimit": -1` to allow editing anytime, or set `"PostEditTimeLimit"` to a positive integer to restrict editing time in seconds. If post editing is disabled, this setting does not apply.
	If using Let's Encrypt without a proxy server, the server will fail to start with an error message unless the Forward80To443 `config.json` setting is set to `true`. If forwarding port 80 to 443, the server will fail to start with an error message unless the ListenAddress `config.json` setting is set to listen on port 443.
v4.6.2	If using Let's Encrypt without a proxy server, forward port 80 through a firewall, with the Forward80To443 `config.json` setting set to `true` to complete the Let's Encrypt certification.
v4.4.0	Composite database indexes were added to the `Posts` table. This may lead to longer upgrade times for servers with more than one million messages.

If you're upgrading from a version earlier than...	Then...
	LDAP sync now depends on email. Make sure all users on your AD/LDAP server have an email address or that their account is deactivated in Mattermost.
v4.2.0	Mattermost now handles multiple content types for integrations, including plaintext content type. If your integration suddenly prints the JSON payload data instead of rendering the generated message, make sure your integration is returning the `application/json` content-type to retain previous behavior.
	By default, user-supplied URLs such as those used for Open Graph metadata, webhooks, or slash commands will no longer be allowed to connect to reserved IP addresses including loopback or link-local addresses used for internal networks. This change may cause private integrations to break in testing environments, which may point to a URL such as http://127.0.0.1:1021/my-command. If you point private integrations to such URLs, you may whitelist such domains, IP addresses, or CIDR notations via the Allowed Untrusted Internal Connections configuration setting in your local environment. Although not recommended, you may also whitelist the addresses in your production environments. See documentation to learn more. Push notification, OAuth 2.0 and WebRTC server URLs are trusted and not affected by this setting.
	Uploaded file attachments are now grouped by day and stored in `/data/<date-of-upload-as-YYYYMMDD>/teams/...` of your file storage system.

If you're upgrading from a version earlier than...	Then...
	Mattermost */platform* ` repo has been separated to `/mattermost-webapp` and `/mattermost-server`. This may affect you if you have a private fork of the `/platform` repo. [More details here (https://forum.mattermost.com/t/mattermost-separating-platform-into-two-repositories-on-september-6th/3708)](https://forum.mattermost.com/t/mattermost-separating-platform-into-two-repositories-on-september-6th/3708).
v4.0.0	(High Availability only) You must manually add new items to the `ClusterSettings` section of your existing `config.json`.
v3.9.0	Old email invitation links, password reset links, and email verification links will no longer work due to a security change. Team invite links copied from the Team Invite Link dialog are not affected and are still valid.
v3.8.0	A change is required in the proxy configuration. If you're using NGINX: 1. Open the NGINX configuration file as root. The file is usually `/etc/nginx/sites-available/mattermost` but might be different on your system. 2. Locate the line: `location /api/v3/users/websocket {` 3. Replace the line with `location ~ /api/v[0-9]+/(users/)?websocket${` If you are using a proxy other than NGINX, make the equivalent change to that proxy's configuration.

If you're upgrading from a version earlier than...	Then...
	You need to verify settings in the System Console due to a security-related change. 1. Go to the the GENERAL section of the System Console 2. Click **Logging** 3. Make sure that the **File Log Directory** field is either empty or has a directory path only. It must not have a filename as part of the path.
	Backwards compatibility with the old CLI tool was removed. If you have any scripts that rely on the old CLI, they must be revised to use the new CLI.
v3.6.0	Update the maximum number of files that can be open. On RHEL6 and Ubuntu 14.04: • Verify that the line `limit nofile 50000 50000` is included in the `/etc/init/mattermost.conf` file. On RHEL7 and Ubuntu 16.04: • Verify that the line `LimitNOFILE=49152` is included in the `/etc/systemd/system/mattermost.service` file.
	(Enterprise Only) Previous `config.json` values for restricting Public and Private channel management will be used as the default values for new settings for restricting Public and Private channel creation and deletion.

If you're upgrading from a version earlier than...	Then...
v3.4.0	If public links are enabled, existing public links will no longer be valid. This is because in earlier versions, existing public links were not invalidated when the Public Link Salt was regenerated. You must update any place where you have published these links.

Additional upgrade notes

Note

- Upgrading the Microsoft Teams Calling plugin to v2.0.0 requires users to reconnect their accounts.
-
- Keybase has stopped serving Mattermost's Ubuntu repository signing key. If you were using it, update your installation scripts to retrieve the key as mentioned in our Linux deployment documentation.
- We recommend avoiding the use of MySQL 8.0.22 as it contains an issue with JSON column types (https://bugs.mysql.com/bug.php?id=101284) changing string values to integers which is preventing Mattermost from working properly.
- When upgrading to Mattermost 7.x from a 5.x release, upgrade to 5.37.10 first before attempting the v7.x upgrade.
- SQL queries for related schema changes below are **automatically executed** during the Mattermost server upgrade process. **You don't need to run these queries manually**. These SQL queries are provided to help you understand what database changes will occur during the upgrade, and for cases where you prefer to reduce downtime by pre-applying schema changes.

On this page

- Important Upgrade Notes
  - Additional upgrade notes

Notify admin

Some Mattermost features are limited to specific plans. Users who want to access these unavailable features can request access through their system admin by sending a notification.

If access to a feature requires a plan upgrade, system admins receive notifications of these requests in order to collect data before upgrading. For example, if only one end-user requests access to AD/LDAP, it probably isn't necessary to upgrade. However, if 20 users, including team and channel admins request it, you may want to consider upgrading to support this need.

This feature is designed for informational purposes only, and no action is required unless you want to take action. Your Mattermost instance's functionality is not affected if you choose not to upgrade.

Notifications

Notifications are triggered by users. The very first time a user sends a request to upgrade or start a trial, a bot message is sent to all system admins indicating the feature or functionality the user has requested that requires an upgrade or trial. This bot message is listed in the **Direct Messages** section of the channel sidebar.

Subsequent request notifications are received by system admins at most every 14 days. When a notification is received by a system admin, a 14-day cool-off period begins. Any requests generated in the middle of the cool-off period will be held off until 14 days later, and then provided in a summarized format.

Take action

You can take action on any of the requests sent using the options provided in the message.

Dismiss notifications

You’re not obligated to upgrade or change your plan. Once you’ve read the notification, it’s marked as read and is filed in your list of direct messages. You can find the message at any time using the search bar. None of your existing Mattermost features and functions are affected negatively if you don’t upgrade.

On this page

- Notify admin
  - Notifications
  - Take action
  - Dismiss notifications

Open Source Components

The following open source components are used to provide the full benefits of Mattermost Enterprise.

Desktop

- Mattermost Desktop v5.12.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.12/NOTICE.txt).
- Mattermost Desktop v5.11.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.11/NOTICE.txt).
- Mattermost Desktop v5.10.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.10/NOTICE.txt).
- Mattermost Desktop v5.9.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.9/NOTICE.txt).
- Mattermost Desktop v5.8.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.8/NOTICE.txt).
- Mattermost Desktop v5.7.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.7/NOTICE.txt).
- Mattermost Desktop v5.6.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.6/NOTICE.txt).
- Mattermost Desktop v5.5.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.5/NOTICE.txt).
- Mattermost Desktop v5.4.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.4/NOTICE.txt).
- Mattermost Desktop v5.3.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.3/NOTICE.txt).
- Mattermost Desktop v5.2.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.2/NOTICE.txt).

- Mattermost Desktop v5.1.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.1/NOTICE.txt).
- Mattermost Desktop v5.0.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-5.0/NOTICE.txt).
- Mattermost Desktop v4.7.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.7/NOTICE.txt).
- Mattermost Desktop v4.6.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.6/NOTICE.txt).
- Mattermost Desktop v4.5.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.5/NOTICE.txt).
- Mattermost Desktop v4.4.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.4/NOTICE.txt).
- Mattermost Desktop v4.3.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.3.0/NOTICE.txt).
- Mattermost Desktop v4.2.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.2/NOTICE.txt).
- Mattermost Desktop v4.1.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.1/NOTICE.txt).
- Mattermost Desktop v4.0.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-4.0/NOTICE.txt).
- Mattermost Desktop v3.7.0 - View Open Source Components (https://github.com/mattermost/desktop/blob/release-3.7/NOTICE.txt).

Mobile

- Mattermost Mobile v2.30.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.30/NOTICE.txt).
- Mattermost Mobile v2.29.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.29/NOTICE.txt).

- Mattermost Mobile v2.28.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.28/NOTICE.txt).
- Mattermost Mobile v2.27.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.27/NOTICE.txt).
- Mattermost Mobile v2.26.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.26/NOTICE.txt).
- Mattermost Mobile v2.25.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.25/NOTICE.txt).
- Mattermost Mobile v2.24.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.24/NOTICE.txt).
- Mattermost Mobile v2.23.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.23/NOTICE.txt).
- Mattermost Mobile v2.22.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.22/NOTICE.txt).
- Mattermost Mobile v2.21.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.21/NOTICE.txt).
- Mattermost Mobile v2.20.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.20/NOTICE.txt).
- Mattermost Mobile v2.19.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.19/NOTICE.txt).
- Mattermost Mobile v2.18.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.18/NOTICE.txt).

- Mattermost Mobile v2.17.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.17/NOTICE.txt).
- Mattermost Mobile v2.16.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.16/NOTICE.txt).
- Mattermost Mobile v2.15.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.15/NOTICE.txt).
- Mattermost Mobile v2.14.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.14/NOTICE.txt).
- Mattermost Mobile v2.13.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.13/NOTICE.txt).
- Mattermost Mobile v2.12.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.12/NOTICE.txt).
- Mattermost Mobile v2.11.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.11/NOTICE.txt).
- Mattermost Mobile v2.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.10/NOTICE.txt).
- Mattermost Mobile v2.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.9/NOTICE.txt).
- Mattermost Mobile v2.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.8/NOTICE.txt).
- Mattermost Mobile v2.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.7/NOTICE.txt).
- Mattermost Mobile v2.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.6/NOTICE.txt).

- Mattermost Mobile v2.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.5/NOTICE.txt).
- Mattermost Mobile v2.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.4/NOTICE.txt).
- Mattermost Mobile v2.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.3/NOTICE.txt).
- Mattermost Mobile v2.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.2/NOTICE.txt).
- Mattermost Mobile v2.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.1/NOTICE.txt).
- Mattermost Mobile v2.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-2.0/NOTICE.txt).
- Mattermost Mobile v1.55.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.55/NOTICE.txt).
- Mattermost Mobile v1.54.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.54/NOTICE.txt).
- Mattermost Mobile v1.53.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.53/NOTICE.txt).
- Mattermost Mobile v1.52.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.52/NOTICE.txt).
- Mattermost Mobile v1.51.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.51/NOTICE.txt).
- Mattermost Mobile v1.50.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.50/NOTICE.txt).

- Mattermost Mobile v1.49.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.49/NOTICE.txt).
- Mattermost Mobile v1.48.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.48/NOTICE.txt).
- Mattermost Mobile v1.47.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.47/NOTICE.txt).
- Mattermost Mobile v1.46.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.46/NOTICE.txt).
- Mattermost Mobile v1.45.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.45/NOTICE.txt).
- Mattermost Mobile v1.44.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.44/NOTICE.txt).
- Mattermost Mobile v1.43.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.43/NOTICE.txt).
- Mattermost Mobile v1.42.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.42/NOTICE.txt).
- Mattermost Mobile v1.41.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.41/NOTICE.txt).
- Mattermost Mobile v1.40.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.40/NOTICE.txt).
- Mattermost Mobile v1.39.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.39/NOTICE.txt).

- Mattermost Mobile v1.38.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.38/NOTICE.txt).
- Mattermost Mobile v1.37.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.37/NOTICE.txt).
- Mattermost Mobile v1.36.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.36/NOTICE.txt).
- Mattermost Mobile v1.35.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.35/NOTICE.txt).
- Mattermost Mobile v1.34.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.34/NOTICE.txt).
- Mattermost Mobile v1.33.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.33/NOTICE.txt).
- Mattermost Mobile v1.32.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.32/NOTICE.txt).
- Mattermost Mobile v1.31.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.31/NOTICE.txt).
- Mattermost Mobile v1.30.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.30/NOTICE.txt).
- Mattermost Mobile v1.29.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.29/NOTICE.txt).
- Mattermost Mobile v1.28.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.28/NOTICE.txt).

- Mattermost Mobile v1.27.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.27/NOTICE.txt).
- Mattermost Mobile v1.26.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.26/NOTICE.txt).
- Mattermost Mobile v1.25.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.25/NOTICE.txt).
- Mattermost Mobile v1.24.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.24/NOTICE.txt).
- Mattermost Mobile v1.23.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.23/NOTICE.txt).
- Mattermost Mobile v1.22.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.22/NOTICE.txt).
- Mattermost Mobile v1.21.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.21/NOTICE.txt).
- Mattermost Mobile v1.20.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.20/NOTICE.txt).
- Mattermost Mobile v1.19.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.19/NOTICE.txt).
- Mattermost Mobile v1.18.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.18/NOTICE.txt).
- Mattermost Mobile v1.17.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.17/NOTICE.txt).

- Mattermost Mobile v1.16.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.16/NOTICE.txt).
- Mattermost Mobile v1.15.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.15/NOTICE.txt).
- Mattermost Mobile v1.14.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.14/NOTICE.txt).
- Mattermost Mobile v1.13.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.13/NOTICE.txt).
- Mattermost Mobile v1.12.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.12/NOTICE.txt).
- Mattermost Mobile v1.11.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.11/NOTICE.txt).
- Mattermost Mobile v1.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.10/NOTICE.txt).
- Mattermost Mobile v1.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.9/NOTICE.txt).
- Mattermost Mobile v1.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.8/NOTICE.txt).
- Mattermost Mobile v1.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.7/NOTICE.txt).
- Mattermost Mobile v1.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.6/NOTICE.txt).
- Mattermost Mobile v1.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.5/NOTICE.txt).

- Mattermost Mobile v1.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.4/NOTICE.txt).
- Mattermost Mobile v1.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.3/NOTICE.txt).
- Mattermost Mobile v1.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.2/NOTICE.txt).
- Mattermost Mobile v1.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.1/NOTICE.txt).
- Mattermost Mobile v1.0.1 - View Open Source Components (https://github.com/mattermost/mattermost-mobile/blob/release-1.0.1/NOTICE.txt).

Server

- Mattermost Enterprise Edition v10.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.10/NOTICE.txt).
- Mattermost Enterprise Edition v10.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.9/NOTICE.txt).
- Mattermost Enterprise Edition v10.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.8/NOTICE.txt).
- Mattermost Enterprise Edition v10.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.7/NOTICE.txt).
- Mattermost Enterprise Edition v10.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.6/NOTICE.txt).
- Mattermost Enterprise Edition v10.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.5/NOTICE.txt).

- Mattermost Enterprise Edition v10.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.4/NOTICE.txt).
- Mattermost Enterprise Edition v10.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.3/NOTICE.txt).
- Mattermost Enterprise Edition v10.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.2/NOTICE.txt).
- Mattermost Enterprise Edition v10.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.1/NOTICE.txt).
- Mattermost Enterprise Edition v10.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-10.0/NOTICE.txt).
- Mattermost Enterprise Edition v9.11.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.11/NOTICE.txt).
- Mattermost Enterprise Edition v9.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.10/NOTICE.txt).
- Mattermost Enterprise Edition v9.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.9/NOTICE.txt).
- Mattermost Enterprise Edition v9.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.8/NOTICE.txt).
- Mattermost Enterprise Edition v9.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.7/NOTICE.txt).
- Mattermost Enterprise Edition v9.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.6/NOTICE.txt).

- Mattermost Enterprise Edition v9.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.5/NOTICE.txt).
- Mattermost Enterprise Edition v9.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.4/NOTICE.txt).
- Mattermost Enterprise Edition v9.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.3/NOTICE.txt).
- Mattermost Enterprise Edition v9.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.2/NOTICE.txt).
- Mattermost Enterprise Edition v9.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.1/NOTICE.txt).
- Mattermost Enterprise Edition v9.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-9.0/NOTICE.txt).
- Mattermost Enterprise Edition v8.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-8.1/NOTICE.txt).
- Mattermost Enterprise Edition v8.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-8.0/NOTICE.txt).
- Mattermost Enterprise Edition v7.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.10/NOTICE.txt).
- Mattermost Enterprise Edition v7.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.9/NOTICE.txt).
- Mattermost Enterprise Edition v7.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.8/NOTICE.txt).

- Mattermost Enterprise Edition v7.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.7/NOTICE.txt).
- Mattermost Enterprise Edition v7.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.5/NOTICE.txt).
- Mattermost Enterprise Edition v7.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.4/NOTICE.txt).
- Mattermost Enterprise Edition v7.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.3/NOTICE.txt).
- Mattermost Enterprise Edition v7.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.2/NOTICE.txt).
- Mattermost Enterprise Edition v7.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.1/NOTICE.txt).
- Mattermost Enterprise Edition v7.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-7.0/NOTICE.txt).
- Mattermost Enterprise Edition v6.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.7/NOTICE.txt).
- Mattermost Enterprise Edition v6.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.6/NOTICE.txt).
- Mattermost Enterprise Edition v6.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.5/NOTICE.txt).
- Mattermost Enterprise Edition v6.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.4/NOTICE.txt).

- Mattermost Enterprise Edition v6.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.3/NOTICE.txt).
- Mattermost Enterprise Edition v6.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.2/NOTICE.txt).
- Mattermost Enterprise Edition v6.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.1/NOTICE.txt).
- Mattermost Enterprise Edition v6.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-6.0/NOTICE.txt).
- Mattermost Enterprise Edition v5.39.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.39/NOTICE.txt).
- Mattermost Enterprise Edition v5.38.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.38/NOTICE.txt).
- Mattermost Enterprise Edition v5.37.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.37/NOTICE.txt).
- Mattermost Enterprise Edition v5.36.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.36/NOTICE.txt).
- Mattermost Enterprise Edition v5.35.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.35/NOTICE.txt).
- Mattermost Enterprise Edition v5.34.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.34/NOTICE.txt).
- Mattermost Enterprise Edition v5.33.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.33/NOTICE.txt).

- Mattermost Enterprise Edition v5.32.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.32/NOTICE.txt).
- Mattermost Enterprise Edition v5.31.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.31/NOTICE.txt).
- Mattermost Enterprise Edition v5.30.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.30/NOTICE.txt).
- Mattermost Enterprise Edition v5.29.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.29/NOTICE.txt).
- Mattermost Enterprise Edition v5.28.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.28/NOTICE.txt).
- Mattermost Enterprise Edition v5.27.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.27/NOTICE.txt).
- Mattermost Enterprise Edition v5.26.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.26/NOTICE.txt).
- Mattermost Enterprise Edition v5.25.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.25/NOTICE.txt).
- Mattermost Enterprise Edition v5.24.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.24/NOTICE.txt).
- Mattermost Enterprise Edition v5.23.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.23/NOTICE.txt).
- Mattermost Enterprise Edition v5.22.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.22/NOTICE.txt).

- Mattermost Enterprise Edition v5.21.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.21/NOTICE.txt).
- Mattermost Enterprise Edition v5.20.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.20/NOTICE.txt).
- Mattermost Enterprise Edition v5.19.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.19/NOTICE.txt).
- Mattermost Enterprise Edition v5.18.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.18/NOTICE.txt).
- Mattermost Enterprise Edition v5.17.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.17/NOTICE.txt).
- Mattermost Enterprise Edition v5.16.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.16/NOTICE.txt).
- Mattermost Enterprise Edition v5.15.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.15/NOTICE.txt).
- Mattermost Enterprise Edition v5.14.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.14/NOTICE.txt).
- Mattermost Enterprise Edition v5.13.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.13/NOTICE.txt).
- Mattermost Enterprise Edition v5.12.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.12/NOTICE.txt).
- Mattermost Enterprise Edition v5.11.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.11/NOTICE.txt).

- Mattermost Enterprise Edition v5.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.10/NOTICE.txt).
- Mattermost Enterprise Edition v5.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.9/NOTICE.txt).
- Mattermost Enterprise Edition v5.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.8/NOTICE.txt).
- Mattermost Enterprise Edition v5.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.7/NOTICE.txt).
- Mattermost Enterprise Edition v5.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.6/NOTICE.txt).
- Mattermost Enterprise Edition v5.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.5/NOTICE.txt).
- Mattermost Enterprise Edition v5.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.4/NOTICE.txt).
- Mattermost Enterprise Edition v5.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.3/NOTICE.txt).
- Mattermost Enterprise Edition v5.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.2/NOTICE.txt).
- Mattermost Enterprise Edition v5.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.1/NOTICE.txt).
- Mattermost Enterprise Edition v5.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-5.0/NOTICE.txt).

- Mattermost Enterprise Edition v4.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.10/NOTICE.txt).
- Mattermost Enterprise Edition v4.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.9/NOTICE.txt).
- Mattermost Enterprise Edition v4.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.8/NOTICE.txt).
- Mattermost Enterprise Edition v4.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.7/NOTICE.txt).
- Mattermost Enterprise Edition v4.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.6/NOTICE.txt).
- Mattermost Enterprise Edition v4.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.5/NOTICE.txt).
- Mattermost Enterprise Edition v4.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.4/NOTICE.txt).
- Mattermost Enterprise Edition v4.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.3/NOTICE.txt).
- Mattermost Enterprise Edition v4.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.2/NOTICE.txt).
- Mattermost Enterprise Edition v4.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.1/NOTICE.txt).
- Mattermost Enterprise Edition v4.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-4.0/NOTICE.txt).

- Mattermost Enterprise Edition v3.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.10/NOTICE.txt).
- Mattermost Enterprise Edition v3.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.9/NOTICE.txt).
- Mattermost Enterprise Edition v3.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.8/NOTICE.txt).
- Mattermost Enterprise Edition v3.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.7/NOTICE.txt).
- Mattermost Enterprise Edition v3.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.6/NOTICE.txt).
- Mattermost Enterprise Edition v3.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.5/NOTICE.txt).
- Mattermost Enterprise Edition v3.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.4/NOTICE.txt).
- Mattermost Enterprise Edition v3.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.3/NOTICE.txt).
- Mattermost Enterprise Edition v3.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.2/NOTICE.txt).
- Mattermost Enterprise Edition v3.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-server/blob/release-3.1/NOTICE.txt).

Webapp

- Mattermost Enterprise Edition v7.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.10/NOTICE.txt).
- Mattermost Enterprise Edition v7.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.9/NOTICE.txt).
- Mattermost Enterprise Edition v7.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.8/NOTICE.txt).
- Mattermost Enterprise Edition v7.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.7/NOTICE.txt).
- Mattermost Enterprise Edition v7.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.5/NOTICE.txt).
- Mattermost Enterprise Edition v7.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.4/NOTICE.txt).
- Mattermost Enterprise Edition v7.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.3/NOTICE.txt).
- Mattermost Enterprise Edition v7.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.2/NOTICE.txt).
- Mattermost Enterprise Edition v7.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.1/NOTICE.txt).
- Mattermost Enterprise Edition v7.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-7.0/NOTICE.txt).

- Mattermost Enterprise Edition v6.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.7/NOTICE.txt).
- Mattermost Enterprise Edition v6.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.6/NOTICE.txt).
- Mattermost Enterprise Edition v6.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.5/NOTICE.txt).
- Mattermost Enterprise Edition v6.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.4/NOTICE.txt).
- Mattermost Enterprise Edition v6.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.3/NOTICE.txt).
- Mattermost Enterprise Edition v6.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.2/NOTICE.txt).
- Mattermost Enterprise Edition v6.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.1/NOTICE.txt).
- Mattermost Enterprise Edition v6.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-6.0/NOTICE.txt).
- Mattermost Enterprise Edition v5.39.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.39/NOTICE.txt).
- Mattermost Enterprise Edition v5.38.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.38/NOTICE.txt).
- Mattermost Enterprise Edition v5.37.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.37/NOTICE.txt).

- Mattermost Enterprise Edition v5.36.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.36/NOTICE.txt).
- Mattermost Enterprise Edition v5.35.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.35/NOTICE.txt).
- Mattermost Enterprise Edition v5.34.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.34/NOTICE.txt).
- Mattermost Enterprise Edition v5.33.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.33/NOTICE.txt).
- Mattermost Enterprise Edition v5.32.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.32/NOTICE.txt).
- Mattermost Enterprise Edition v5.31.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.31/NOTICE.txt).
- Mattermost Enterprise Edition v5.30.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.30/NOTICE.txt).
- Mattermost Enterprise Edition v5.29.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.29/NOTICE.txt).
- Mattermost Enterprise Edition v5.28.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.28/NOTICE.txt).
- Mattermost Enterprise Edition v5.27.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.27/NOTICE.txt).
- Mattermost Enterprise Edition v5.26.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.26/NOTICE.txt).

- Mattermost Enterprise Edition v5.25.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.25/NOTICE.txt).
- Mattermost Enterprise Edition v5.24.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.24/NOTICE.txt).
- Mattermost Enterprise Edition v5.23.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.23/NOTICE.txt).
- Mattermost Enterprise Edition v5.22.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.22/NOTICE.txt).
- Mattermost Enterprise Edition v5.21.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.21/NOTICE.txt).
- Mattermost Enterprise Edition v5.20.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.20/NOTICE.txt).
- Mattermost Enterprise Edition v5.19.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.19/NOTICE.txt).
- Mattermost Enterprise Edition v5.18.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.18/NOTICE.txt).
- Mattermost Enterprise Edition v5.17.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.17/NOTICE.txt).
- Mattermost Enterprise Edition v5.16.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.16/NOTICE.txt).
- Mattermost Enterprise Edition v5.15.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.15/NOTICE.txt).

- Mattermost Enterprise Edition v5.14.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.14/NOTICE.txt).
- Mattermost Enterprise Edition v5.13.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.13/NOTICE.txt).
- Mattermost Enterprise Edition v5.12.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.12/NOTICE.txt).
- Mattermost Enterprise Edition v5.11.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.11/NOTICE.txt).
- Mattermost Enterprise Edition v5.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.10/NOTICE.txt).
- Mattermost Enterprise Edition v5.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.9/NOTICE.txt).
- Mattermost Enterprise Edition v5.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.8/NOTICE.txt).
- Mattermost Enterprise Edition v5.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.7/NOTICE.txt).
- Mattermost Enterprise Edition v5.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.6/NOTICE.txt).
- Mattermost Enterprise Edition v5.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.5/NOTICE.txt).
- Mattermost Enterprise Edition v5.4.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.4/NOTICE.txt).

- Mattermost Enterprise Edition v5.3.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.3/NOTICE.txt).
- Mattermost Enterprise Edition v5.2.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.2/NOTICE.txt).
- Mattermost Enterprise Edition v5.1.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.1/NOTICE.txt).
- Mattermost Enterprise Edition v5.0.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-5.0/NOTICE.txt).
- Mattermost Enterprise Edition v4.10.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.10/NOTICE.txt).
- Mattermost Enterprise Edition v4.9.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.9/NOTICE.txt).
- Mattermost Enterprise Edition v4.8.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.8/NOTICE.txt).
- Mattermost Enterprise Edition v4.7.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.7/NOTICE.txt).
- Mattermost Enterprise Edition v4.6.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.6/NOTICE.txt).
- Mattermost Enterprise Edition v4.5.0 - View Open Source Components (https://github.com/mattermost/mattermost-webapp/blob/release-4.5/NOTICE.txt).

On this page

- Open Source Components
  - Desktop
  - Mobile
  - Server
  - Webapp

Prepare to upgrade Mattermost

In most cases, you can upgrade Mattermost Server in a few minutes. However, the upgrade can take longer depending on several factors, including the size and complexity of your installation, and the version that you're upgrading from. When planning an upgrade, it's worth confirming that your current database and operating system version are still supported. Details can be found on our software and hardware requirements page.

Upgrade Best Practices

Mattermost will aim to have non-locking, backwards-compatible migrations in general. This backwards compatibility guarantee extends to only the last ESR version. For example, if there are three ESR versions ESR1, ESR2, and ESR3, upgrading from ESR1 to ESR2, and then ESR2 to ESR3 will ensure backwards compatibility, but not from ESR1 to ESR3 directly.

In the case of delayed upgrades, we recommend upgrading to the closest ESR version first, and from there to the next ESR. Do not attempt to directly upgrade to the latest version as it might break backwards compatibility of the older nodes in the cluster during the upgrade.

Upgrade to Mattermost v7.1

Mattermost v7.1 introduces schema changes in the form of a new column and its index. Our test results for the schema changes include:

- PostgreSQL 12M Posts, 2.5M Reactions - ~1min 18s (instance: db.r5.2xlarge)

You can run the following SQL queries before the upgrade that obtains a lock on `Reactions` table. Users' reactions posted during this time won't be reflected in the database until the migrations are complete. This is fully backwards-compatible.

If your connection collation and table collations are different, this can result in the error *Illegal mix of collations*. To resolve this error, set the same collation

for both the connection and the table. There are different collations at different levels - connection, database, table, column, and database administrators may choose to set different collation levels for different objects.

```
ALTER TABLE reactions ADD COLUMN IF NOT EXISTS channelid varchar(26) NOT NULL DEFAULT '';
```

```
UPDATE reactions SET channelid = COALESCE((select channelid from posts where posts.id = reactions.postid), '') WHERE channelid='';
```

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_reactions_channel_id on reactions (channelid);
```

Upgrade to Mattermost v6.7

Mattermost v6.7 introduces schema changes in the form of a new index. The following notes our test results for the schema changes:

- Postgres 7M Posts - ~9s (instance: db.r5.xlarge)

If you want a zero downtime upgrade, you can apply this index prior to doing the upgrade.

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_posts_create_at_id on posts(createat, id);
```

This is fully backwards-compatible and will not acquire any table lock or affect any existing operations on the table.

Upgrade to Mattermost v6.0

A Mattermost Server v6.0 upgrade will run significant database schema changes that can cause an extended startup time depending on the dataset size. Zero downtime won't be possible for v6.0, but depending on the efforts made during the migration process, it can be minimized to a large extent.

Running queries prior to the upgrade can also save some downtime. However, some queries can still cause full table locking and require Mattermost to be in read-only mode for the duration of the query.

We strongly recommend that you:

- Set up a maintenance window outside of working hours to mitigate the migration impact.
- Keep a backup of your database to ensure you can load a previous database snapshot if necessary.
- Upgrade your instance of Mattermost to the latest Extended Support Release (ESR) first before attempting to run the Mattermost v6.0 upgrade.

Important

Support for Mattermost Server v9.11 Extended Support Release is coming to the end of its life cycle on May 15, 2025. Upgrading to Mattermost Server v10.5 Extended Support Release or later is recommended. Upgrading from a previous Extended Support Release to the latest Extended Support Release is supported. Review the Important Upgrade Notes for all intermediate versions in between to ensure you're aware of the possible migrations that could affect your upgrade.

v6.0 database schema migrations

Mattermost v6.0 introduces several database schema changes to improve both database and application performance. The upgrade will run significant database schema changes that can cause an extended startup time depending on the dataset size. We've conducted extensive tests on supported PostgreSQL database drivers, using realistic datasets of more than 10 million posts and more than 72 million posts.

The following query executed during the migration process will have a significant impact on database CPU usage and write operation restrictions for the duration of the query:

`ALTER TABLE posts ALTER COLUMN props TYPE jsonb USING props::jsonb;` (~ 11 minutes)

For a complete breakdown of PostgreSQL queries, as well as their impact and duration, see the [Mattermost v6.0 DB schema migrations analysis (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#postgresql-1)](https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#postgresql-1).

Upgrade from releases older than v5.35

Customers upgrading from a release older than Mattermost v5.35 should expect extended downtime when upgrading to v6.0 due to the introduction of backend database architecture introduced in v5.35. This upgrade path isn't recommended for large installations. We recommend upgrading to the latest Extended Support Release (ESR) first before upgrading to Mattermost v6.0. See the unsupported legacy releases documentation for additional details.

If you're upgrading from a version prior to Mattermost v5.0, you can't upgrade directly to v6.0. Instead, we strongly recommend approaching the upgrade in phases, starting with an upgrade to the latest ESR first, followed by the upgrade to v6.0. During the first phase of updates, you must also modify your service file to work with the binary changes introduced with the v5.0 release. Your execution directory should point to the Mattermost base directory (i.e. `/opt/mattermost`), and your binary should point to the `mattermost` binary (i.e. `/opt/mattermost/bin/mattermost`).

Ensure you review the Important Upgrade Notes for all intermediate release versions in between to ensure you're aware of the possible migrations that could affect your upgrade.

Note

Customers upgrading from releases older than v5.35 following our recommended upgrade process may encounter the following error during the upgrade to v6.0:

`Failed to alter column type. It is likely you have invalid JSON values in the column. Please fix the values manually and run the migration again.","caller":"sqlstore/store.go:854","error":"pq: unsupported Unicode escape sequence`

To assist with troubleshooting, you can enable `SqlSettings.Trace` to narrow down what table and column are causing issues during the upgrade. The following queries change the columns to JSONB format in PostgreSQL. Run these against your v5.39 development database to find out which table and column has Unicode issues:

```
ALTER TABLE posts ALTER COLUMN props TYPE jsonb USING props::jsonb;
ALTER TABLE channelmembers ALTER COLUMN notifyprops TYPE jsonb USING
notifyprops::jsonb; ALTER TABLE jobs ALTER COLUMN data TYPE jsonb
USING data::jsonb; ALTER TABLE linkmetadata ALTER COLUMN data TYPE
jsonb USING data::jsonb; ALTER TABLE sessions ALTER COLUMN props TYPE
jsonb USING props::jsonb; ALTER TABLE threads ALTER COLUMN
participants TYPE jsonb USING participants::jsonb; ALTER TABLE users
ALTER COLUMN props TYPE jsonb USING props::jsonb; ALTER TABLE users
ALTER COLUMN notifyprops TYPE jsonb USING notifyprops::jsonb; ALTER
TABLE users ALTER COLUMN timezone TYPE jsonb USING timezone::jsonb;
```

Once you've identified the table being affected, verify how many invalid occurrences of *u0000* you have using the following SELECT query:

```
SELECT COUNT(*) FROM TableName WHERE ColumnName LIKE '%\u0000%';
```

Then select and fix the rows accordingly. If you prefer, you can also fix all occurrences at once in a given table or column using the following UPDATE query:

```
UPDATE TableName SET ColumnName = regexp_replace(ColumnName, '\
\u0000', '', 'g') WHERE ColumnName LIKE '%\u0000%';
```

Upgrade high availability cluster-based deployments

In high availability cluster-based environments, you should expect to schedule downtime for the upgrade to v6.0. Based on your database size and setup, the migration to v6.0 can take a significant amount of time, and may even lock the tables for posts which will prevent your users from posting or receiving messages until the migration is complete.

Ensure you review the high availability cluster-based deployment upgrade guide, as well as the Important Upgrade Notes to make sure you're aware of any actions you need to take before or after upgrading from your particular version.

Important

Running two different versions of Mattermost in your cluster should not be done outside of an upgrade scenario. Due to a fundamental change to the clustering code in v6.0, nodes from different versions cannot be run, as noted in the Important Upgrade Notes product documentation.

The release of v6.0 also introduces database schema changes and longer migration times should be expected.

On this page

- Prepare to upgrade Mattermost
  - Upgrade Best Practices
  - Upgrade to Mattermost v7.1
  - Upgrade to Mattermost v6.7
  - Upgrade to Mattermost v6.0
    - v6.0 database schema migrations
  - Upgrade from releases older than v5.35
  - Upgrade high availability cluster-based deployments

Upgrade Mattermost in Kubernetes and High Availability environments

This guide provides a resilient and comprehensive strategy for upgrading Mattermost deployments managed via Kubernetes and the Mattermost Operator, including High Availability (HA) and optional Active/Active failover configurations. It outlines best practices to ensure zero downtime, minimize service risk, and provide robust fallback mechanisms.

Architecture overview

Kubernetes-based deployment

Mattermost uses Kubernetes for container orchestration, deployed and managed via Helm charts and the Mattermost Operator. This model enables scalable, highly available, and automatically managed application lifecycles.

The Mattermost Operator handles the upgrade process automatically, ensuring that pods are updated incrementally and that traffic is routed correctly throughout the upgrade. If an error occurs during the upgrade, the Operator will not apply any changes, allowing you to investigate and resolve the issue or manually roll back without impacting the live environment. See the Downgrade Mattermost Server documentation for rollback details.

Health monitoring ensures that only healthy pods are replaced, and new pods are brought online only after passing health checks. New pods are deployed with the updated version, while old pods are gracefully terminated.

High Availability

In High Availability (HA) cluster-based deployments, Mattermost runs multiple application servers in a cluster. This configuration ensures that if one server fails, others can continue to serve requests without downtime. User traffic load balancing is managed with services such as NGINX Ingress or HAProxy. PostgreSQL and file storage are deployed with replication for redundancy and failover.

Active/Active deployments

An Active/Active configuration is optional and consists of two or more Mattermost clusters running concurrently across geographically distributed regions or availability zones. Each cluster is capable of serving live user traffic and processing requests independently while remaining in sync with shared backend components such as the database and file storage.

Key benefits for enterprise customers include:

- **Resilience and uptime**: If a site becomes unavailable due to maintenance or an outage, another site can continue to serve users with minimal disruption.
- **Geographic distribution**: Users connect to the nearest cluster for lower latency and faster response times.
- **Load distribution**: Workload can be balanced between clusters to improve performance and system scalability.

These deployments require careful configuration management and coordination to ensure data consistency, upgrade safety, and seamless traffic failover. These clusters must maintain configuration/data consistency and require coordinated upgrades. See the Active/Active upgrade considerations section for recommendations.

Step 1: Prepare

This phase ensures your environment is healthy, backed up, and ready for an upgrade. Follow each step carefully to avoid disruptions and to ensure rollback readiness.

Important

- **Backup your data**: Always back up your database and file storage before starting an upgrade. This ensures you can restore to a previous state in case of issues during the upgrade process.
- **Upgrade safely**: The Mattermost Operator performs upgrade validation checks before rollout. If checks fail, the upgrade is blocked and no changes are applied. If checks pass, the Operator upgrades pods incrementally to maintain uptime.

Pre-upgrade checklist

1. **Cluster health**: Ensure all nodes are in a ready state and all pods are running without errors. This step is crucial to avoid issues during the upgrade process. Use the following commands to check cluster health:

```
kubectl get nodes kubectl get pods --all-namespaces
```

2. **Helm setup**: Verify that Helm is installed and configured correctly in your Kubernetes cluster. Ensure you have the latest version of the Mattermost Operator Helm chart. Use the following commands to check Helm setup:

```
helm repo update helm upgrade mattermost-operator mattermost/mattermost-
operator -n <OPERATOR_NAMESPACE_HERE> -f <OPTIONAL_CUSTOM_VALUES_HERE>
```

3. **Confirm Mattermost Operator version**: Make sure the image version matches the latest supported release from the Helm repository. Use the following command to check the current image version of the Mattermost Operator:

```
kubectl get deployment mattermost-operator -n mattermost
-o=jsonpath='{.spec.template.spec.containers[0].image}'
```

4. **Verify available resources**: Ensure your Kubernetes cluster has sufficient CPU and memory resources to handle the upgrade process. Use the following commands to check resource availability:

```
kubectl top nodes kubectl describe node | grep Allocatable
```

5. Back up database and file storage.

> *Use `pg_dump` or volume snapshots to create a full backup to a secure, external location (e.g., S3 or NFS). Validate the backup can be restored. See the Backup and Disaster Recovery documentation for details.*

6. **Ensure configuration consistency**: Validate that `values.yaml`, secrets, and other configuration files are version-controlled and consistent across all clusters (required for Active/Active deployments). Use tools like GitOps or

configuration management systems to ensure all clusters have the same configuration.

7. We strongly recommend performing a dry run by testing the upgrade in a staging environment that mirrors production to catch misconfigurations early.

Step 2: Perform the upgrade

This step involves updating the Mattermost Operator and Mattermost server to a new version. The Operator manages the upgrade process, ensuring that pods are updated incrementally and that traffic is routed correctly throughout the upgrade.

We recommend having a separate Mattermost custom resource. See the Deploy Mattermost on Kubernetes documentation for details.

1. In the separate resource, update the `version` field in your `mattermost-installation.yaml` file by replacing the `<new-version-tag>` with the specific version you are upgrading to:

> ```
> apiVersion: installation.mattermost.com/v1beta1 kind: Mattermost metadata:  name: <INSTALLATION_NAME_HERE> spec:  version: <new-version-tag> # Update this field
> ```
>
> *Alternatively, if you're using Helm `values.yaml` directly, update it with the desired Mattermost version:*
>
> ```
> image:  repository: mattermost/mattermost-enterprise-edition  tag: <new-version-tag>
> ```

2. Apply the updated configuration:

> *For Mattermost custom resource deployments:*
>
> ```
> kubectl apply -f mattermost-installation.yaml
> ```
>
> *For Helm values deployments:*

```
helm upgrade mattermost mattermost/mattermost-operator -f values.yaml
```

3. Monitor the rollout by tracking upgrade progress and logs with the following commands:

```
kubectl get pods -n mattermost kubectl logs -f <pod-name> -n mattermost
```

New pods are started with the updated version. Old pods are gracefully terminated after health checks pass, and traffic remains uninterrupted due to rolling upgrade behavior. NGINX, HAProxy, or Ingress configurations continue routing traffic seamlessly during the upgrade process.

Step 3: Active/Active upgrade considerations

When your deployment includes an Active/Active configuration with multi-cluster or multi-region deployments, you must ensure that all clusters are upgraded in a coordinated manner to maintain data consistency and service availability.

For coordinated site upgrades, we recommend the following steps:

- Use your global load balancer or DNS to route traffic away from the site being upgraded.
- Upgrade one site at a time while directing traffic to the other. This minimizes downtime and allows for validation of the upgrade process.
- Confirm replication health for databases and storage before proceeding to the next site.
- Monitor performance and logs before resuming traffic to the upgraded site.

To prevent split-brain scenarios during upgrades and ensure data consistency:

- Use a global load balancer or DNS to direct traffic exclusively to the active site.
- Disable writes on the site being upgraded until the upgrade is complete and validated.

- Confirm that only one site is actively writing data at any time.
- Monitor logs for signs of replication lag, file synchronization issues, or data divergence.
- Validate the health of data replication channels before and after each site upgrade.

Step 4: Validate

After the upgrade:

1. Confirm that all pods are healthy and running the expected version using the following command:

```
kubectl get pods -n mattermost
-o=jsonpath='{.items[*].spec.containers[*].image}'
```

3. Perform product smoke tests:

> - *Log in to the Mattermost web interface and navigate across teams and channels.*
> - *Test core functionalities such as posting messages and uploading files.*
> - *Verify that all integrations, webhooks, and plugins are functioning as expected.*
> - *Check for error messages in the logs.*

4. Use monitoring tools to confirm application health and performance. See the Performance monitoring with Prometheus and Grafana documentation or the Metrics plugin documentation for details on collecting and reviewing performance metrics.

Rollback strategy

In case of upgrade issues, rollback should be performed by modifying the custom resource and setting the Mattermost version back to its original value.

For Mattermost custom resource deployments, update the `version` field in your `mattermost-installation.yaml` file to the previous version:

```
apiVersion: installation.mattermost.com/v1beta1 kind: Mattermost metadata:
name: <INSTALLATION_NAME_HERE> spec:  version: <previous-version-tag> # Set
back to previous version
```

Then apply the rollback:

```
kubectl apply -f mattermost-installation.yaml
```

Alternatively, for Helm values deployments, use Helm's rollback command:

```
helm rollback mattermost <revision_number>
```

Restore your PostgreSQL database and file store backups if needed. Refer to the Backup and Disaster Recovery documentation for detailed guidance.

Frequently asked questions

How can I confirm an upgrade was successful?

A successful upgrade is indicated by:

- All pods running the expected image version
- No errors in pod logs
- Functional smoke tests passing (login, messaging, file upload, etc.)
- No anomalies in monitoring dashboards

What should I monitor post-upgrade?

We recommend monitoring the following key metrics after an upgrade:

- Pod health and restarts
- Application logs for errors or warnings

- Database replication health (if HA or Active/Active)
- Performance metrics (latency, error rate) via Prometheus/Grafana

How can I test upgrades without impacting production?

We strongly recommend deploying a separate staging environment that mirrors production, and running a full end-to-end upgrade simulation before upgrading live production clusters.

Can I upgrade the Operator and Mattermost server at the same time?

Upgrade the Operator first. Validate it's stable before upgrading the Mattermost server version.

Do I need to upgrade the database separately?

Mattermost does not upgrade the database schema by default. You must manually apply database schema updates if required by a newer Mattermost version. Review the Mattermost Server changelog for any migration steps.

What if my pods don't become ready after the upgrade?

Check the Operator logs and pod events using `kubectl` to identify and resolve issues. Look for common issues such as resource constraints, configuration errors, or network connectivity problems.

What version compatibility should I be aware of between the Mattermost Operator and the application server?

Running an older Operator version may not support newer Mattermost features or upgrade flows. Always check the [Helm chart release notes (https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-operator)](https://github.com/mattermost/mattermost-helm/tree/master/charts/mattermost-operator) for version compatibility between the Operator and the Mattermost server.

Can I automate this upgrade process using GitOps or CI/CD?

Yes. You can manage upgrades using tools like ArgoCD or FluxCD to apply Helm changes from version-controlled `values.yaml` files. Ensure changes are peer-reviewed and validated in staging before promotion.

On this page

- Upgrade Mattermost in Kubernetes and High Availability environments
  - Architecture overview
    - Kubernetes-based deployment
    - High Availability
    - Active/Active deployments
  - Step 1: Prepare
    - Pre-upgrade checklist
  - Step 2: Perform the upgrade
  - Step 3: Active/Active upgrade considerations
  - Step 4: Validate
  - Rollback strategy
  - Frequently asked questions
    - How can I confirm an upgrade was successful?
    - What should I monitor post-upgrade?
    - How can I test upgrades without impacting production?
    - Can I upgrade the Operator and Mattermost server at the same time?
    - Do I need to upgrade the database separately?
    - What if my pods don’t become ready after the upgrade?
    - What version compatibility should I be aware of between the Mattermost Operator and the application server?
    - Can I automate this upgrade process using GitOps or CI/CD?

Upgrade Mattermost Server

In most cases, you can upgrade Mattermost Server in a few minutes. However, the upgrade can take longer depending on several factors, including the size and complexity of your installation, and the version that you're upgrading from.

If this is your first Mattermost upgrade, we recommend that you read the comprehensive upgrade guide below. If you're an experienced system admin familiar with the upgrade process, see the quick start checklist.

Quick start

Experienced system admins looking for a quick overview of the Mattermost upgrade process can use the following 4-step checklist below:

1. Ensure you have a complete backup of your system before proceeding, including database and application.
2. **System requirements**: Verify that your server and PostgreSQL version meets Mattermost requirements.
3. **Preparation steps**

> - *Download the latest server version.*
> - *Extract the new files to a temporary location (* `/tmp` *)*
> - *Identify the current Mattermost directory (default:* `/opt/mattermost` *)*

4. **Upgrade process**

> - *Stop the Mattermost service:* `sudo systemctl stop mattermost`
> - *Remove old application files (preserve config, data, logs, plugins).*
> - *Copy new files to the install directory.*
> - *Change file ownership:* `sudo chown -R mattermost:mattermost mattermost`

> - *Set capabilities for low port usage if needed:* `sudo setcap cap_net_bind_service=+ep ./mattermost/bin/mattermost`
> - *Start the Mattermost service:* `sudo systemctl start mattermost`
> - *Remove temporary upgrade files.*

For detailed instructions and additional considerations, see the complete upgrade guide below.

> Tip
>
> To learn how to safely upgrade your deployment in Kubernetes for High Availability and Active/Active support, see the Upgrading Mattermost in Kubernetes and High Availability Environments documenation.

Comprehensive upgrade guide

Before you begin

Read these instructions carefully from start to finish.

Make sure that you understand how to prepare for your upgrade, familiarize yourself with all software and hardware requirements, read the important upgrade notes and that you understand each step of the upgrade process documented below before starting a Mattermost upgrade. If you have questions or concerns, you can ask on the Mattermost forum at https://forum.mattermost.com/.

Gather the following information before starting the upgrade:

- **Existing install directory - {install-path}**: If you don't know where Mattermost Server is installed, use the `whereis mattermost` command to find standard binary places and $PATH.
  - This command won't return anything if `/opt/mattermost/bin` wasn't added to the PATH.
  - Alternatively, you can use the `find / -executable -type f -iname mattermost 2> /dev/null` command to find the `mattermost` binary.

  - The output should be similar to `/opt/mattermost/bin/mattermost`.
  - The install directory is everything before the first occurrence of the string `/mattermost`. In this example, the `{install-path}` is `/opt`.
  - If that command doesn't produce any results, it's likely because your version is older; try `whereis platform` instead.
- **Location and size of your local storage directory**: The local storage directory contains all the files that users have attached to their messages.
  - If you don't know its location, open the System Console and go to **Environment > File Storage**, then read the value in **Local Storage Directory**.
  - Paths are relative to the `mattermost` directory. For example, if the local storage directory is `./data/` then the absolute path is `{install-path}/mattermost/data`.
- **Database disk space**: If you're upgrading a Mattermost deployment on the same server as your database, we recommend a minimum 2GB of free disk space to allow for extraction, copy, and cleanup, and a minimum of twice the size of your Mattermost installation available for the database.

> Tip
>
> Consider generating a migration plan using the mattermost db migrate –save-plan CLI command when upgrading to have a detailed record of the changes that will be applied to your database. This can make it easier to revert those changes if you need to downgrade later.

Upgrade Mattermost Server

1. In a terminal window on the server that hosts Mattermost, change to your home directory. Delete any files and directories that might still exist from a previous download.

```
cd /tmp
```

2. Download [the latest version of Mattermost Server (https://mattermost.com/download/)](https://mattermost.com/download/) based on your Mattermost edition.

> *Enterprise Edition*
>
> *In the following command, replace* `X.X.X` *with the version that you want to download:*
>
> ```
> wget https://releases.mattermost.com/X.X.X/mattermost-X.X.X-linux-amd64.tar.gz
> ```
>
> *Team Edition*
>
> *In the following command, replace* `X.X.X` *with the version that you want to download:*
>
> ```
> wget https://releases.mattermost.com/X.X.X/mattermost-team-X.X.X-linux-amd64.tar.gz
> ```

3. Confirm no other Mattermost zip folders exist in your `/tmp` directory. If another version's zip file does exist, delete or rename the file.

   ```
   ls -- mattermost*.gz
   ```

   If anything except the new release is returned above, rename this file or delete it completely.

4. Extract the Mattermost Server files.

   ```
   tar -xf mattermost*.gz --transform='s,^[^/]\+,\0-upgrade,'
   ```

   The `transform` option adds a suffix to the topmost extracted directory so it does not conflict with the usual install directory.

5. Stop your Mattermost server.

   ```
   sudo systemctl stop mattermost
   ```

6. Back up your data and application. Make sure you've properly backed up your database before continuing with the upgrade. In case of an unexpected failure, you should be in a position to load a previous database snapshot.

   1. Back up your database using your organization's standard procedures for backing up the database.

   2. Back up your application by copying into an archive folder (e.g. `mattermost-back-YYYY-MM-DD-HH-mm`).

      ```
      cd {install-path} sudo cp -ra mattermost/ mattermost-back-$(date
      +'%F-%H-%M')/
      ```

7. Remove all files **except** data and custom directories from within the current `mattermost` directory. We strongly recommend reading the important note below before executing the following command.

   ```
   sudo find mattermost/ mattermost/client/ -mindepth 1 -maxdepth 1 \! \(
   -type d \( -path mattermost/client -o -path mattermost/client/plugins -o
   -path mattermost/config -o -path mattermost/logs -o -path mattermost/
   plugins -o -path mattermost/data \) -prune \) | sort | sudo xargs rm -r
   ```

   Important

   By default, the following subdirectories will be preserved on upgrade: `config`, `logs`, `plugins`, `client/plugins`, and `data`. Custom directories and ANY other directories that you've added to Mattermost are NOT preserved by default. Generally, these are TLS keys or other custom information, but can also include a different directories configured for local storage other than `data`, as well as custom directories used to store attachments.

   Before continuing with your upgrade, we strongly recommend that you:

   1. Run `ls` on your Mattermost install directory to identify all default folders that exist prior to upgrading.

      A default Mattermost installation has the following files and directories:

```
$ ls /opt/mattermost ENTERPRISE-EDITION-LICENSE.txt README.md client
data i18n manifest.txt prepackaged_plugins NOTICE.txt bin config
fonts logs plugins templates
```

2. Perform a dry run of deleting the contents of the `mattermost` folder and preserving only the specified directories and their contents by running the following command. It's the same command listed above, but it omits `sudo xargs rm -r`:

```
sudo find mattermost/ mattermost/client/ -mindepth 1 -maxdepth 1 \! \
( -type d \( -path mattermost/client -o -path mattermost/client/
plugins -o -path mattermost/config -o -path mattermost/logs -o -path
mattermost/plugins -o -path mattermost/data \) -prune \) | sort
```

3. If you store TLSCert/TLSKey files or other information within your `/opt/mattermost` folder, you need to append `-o -path mattermost/yourFolderHere` to the command above to avoid having to manually copy the TLSCert/TLSKey files from the backup into the new install.

```
sudo find mattermost/ mattermost/client/ -mindepth 1 -maxdepth 1 \! \
( -type d \( -path mattermost/client -o -path mattermost/client/
plugins -o -path mattermost/config -o -path mattermost/logs -o -path
mattermost/plugins -o -path mattermost/data -o -path mattermost/
yourFolderHere \) -prune \) | sort
```

4. If you're using Bleve search, and the directory exists *within* the `mattermost` directory, the index directory path won't be preserved using the command above.

   - You can either move the bleve index directory out from the `mattermost` directory before upgrading or, following an upgrade, you can copy the contents of the bleve index directory from the `backup` directory.
   - You can then store that directory or re-index as preferred.
   - The bleve indexes can be migrated without reindexing between Mattermost versions. See our Configuration Settings documentation for details on configuring the bleve index directory.

Once you've completed all of the steps above (where applicable), you're ready to execute the full command that includes `xargs rm -r` to delete the files. Note that the following example includes `-o -path mattermost/yourFolderHere`:

```
sudo find mattermost/ mattermost/client/ -mindepth 1 -maxdepth 1 \! \(
-type d \( -path mattermost/client -o -path mattermost/client/plugins -o
-path mattermost/config -o -path mattermost/logs -o -path mattermost/
plugins -o -path mattermost/data -o -path mattermost/yourFolderHere \)
-prune \) | sort | sudo xargs rm -r
```

8. Copy the new files to your install directory.

```
sudo cp -an /tmp/mattermost-administration-guide/upgrade/. mattermost/
```

> Note
>
> The `n` (no-clobber) flag and trailing `.` on source are very important. The `n` (no-clobber) flag preserves existing configurations and logs in your installation path. The trailing `.` on source ensures all installation files are copied.

9. Change ownership of the new files after copying them. For example:

```
sudo chown -R mattermost:mattermost mattermost
```

> Note
>
> - If you didn't use `mattermost` as the owner and group of the install directory, run `sudo chown -hR {owner}:{group} {install-path}/mattermost`.
> - If you're uncertain what owner or group was defined, use the `ls -l {install-path}/mattermost/bin/mattermost` command to obtain them.

10. If you want to use port 80 or 443 to serve your server, and/or if you have TLS set up on your Mattermost server, you **must** activate the `CAP_NET_BIND_SERVICE` capability to allow the new Mattermost binary to bind to ports lower than 1024. For example:

    ```
    sudo setcap cap_net_bind_service=+ep ./mattermost/bin/mattermost
    ```

11. Start your Mattermost server.

    ```
    sudo systemctl start mattermost
    ```

12. Remove the temporary files.

    ```
    sudo rm -r /tmp/mattermost-administration-guide/upgrade/ sudo rm -i /tmp/
    mattermost*.gz
    ```

13. If you're using a high availability deployment, you need to apply the steps above on every node in your cluster. Once complete, the **Config File MD5** columns in the high availability section of the System Console should be green. If they're yellow, please ensure that all nodes have the same server version and the same configuration.

    If they continue to display as yellow, trigger a configuration propagation across the cluster by opening the System Console, changing a setting, and reverting it. This will enable the **Save** button for that page. Then, select **Save**. This will not change any configuration, but sends the existing configuration to all nodes in the cluster.

After the server is upgraded, users might need to refresh their browsers to experience any new features.

Upgrade Team Edition to Enterprise Edition

To upgrade from the Team Edition to the Enterprise Edition, follow the normal upgrade instructions provided above, making sure that you download the Enterprise Edition of Mattermost Server in Step 2.

Upload a license key

When Enterprise Edition is running, open **System Console > About > Editions and License** and upload your license key.

[Mattermost Academy Learn about upgrading Mattermost using a tarball (https://mattermost.com/pl/mattermost-academy-upgrade-tarball-training)](https://mattermost.com/pl/mattermost-academy-upgrade-tarball-training)

On this page

- Upgrade Mattermost Server
- Upgrade Mattermost Server
  - Quick start
  - Comprehensive upgrade guide
    - Before you begin
    - Upgrade Mattermost Server
  - Upgrade Team Edition to Enterprise Edition
  - Upload a license key
  - Quick start
  - Comprehensive upgrade guide
    - Before you begin
    - Upgrade Mattermost Server
  - Upgrade Team Edition to Enterprise Edition
  - Upload a license key

Welcome email to end users

To make it easy for your end users to start using Mattermost right away, we created a sample email template that you can use.

Remember to replace all the items below in bold with your information.

Email template

From: **[company name]** IT Team

To: End users

Subject: New Collaboration Platform - Mattermost

Hi all,

As some of you already know, we are moving to Mattermost as our collaboration platform. Mattermost is collaboration software you can use to talk, share files, and collaborate on projects or initiatives. Mattermost also integrates with many of the apps that you use every day, like **[add apps]**.

We are moving to Mattermost because it will host all our collaboration in one place, is instantly searchable and available from all your devices.

Some of the major benefits of using Mattermost are:

- Direct 1:1 and group messaging
- Channels for topic-based, group-based, or meeting-based chat
- Streamlined collaboration on projects
- Reduced email clutter
- Searching across messages and channels
- Sharing files

To get started:

1. Open a browser on your computer, go to **[Mattermost URL]** and log in with your **[LDAP/AD, SAML, Google, etc]** credentials. Remember to bookmark the URL so you can use it to log in next time.
2. Download the Mattermost apps for desktop and mobile. See the Use Mattermost end user documentation for details on how to get up and running quickly.
3. Start messaging!

Questions? If you have any questions, feel free to post in the **[~Mattermost channel]** or email us at **[IT email]**.

Happy collaborating!

[company name] IT Team

On this page

- Welcome email to end users
  - Email template

Upgrade Mattermost

Stay up to date with the latest features and improvements.

- Important upgrade notes - Find version-specific upgrade considerations.
- Prepare to upgrade Mattermost - Learn how to prepare for a Mattermost upgrade.
- Communicate scheduled maintenance best practices - Learn best practices for communicating scheduled server maintenance in advance of a service maintenance window.
- Upgrade Mattermost Server - Learn the basics of upgrading your Mattermost server to the latest version.
- Upgrade Mattermost in Kubernetes and High Availability environments - Learn how to upgrade Mattermost in Kubernetes and High Availability environments.
- Enterprise install and upgrade - Learn how to upgrade your Mattermost server to premium versions.
- Administrator onboarding tasks - Learn about the onboarding tasks for administrators after an upgrade.
- Enterprise roll-out-checklist - Learn about the roll-out checklist for enterprise users.
- Welcome email to end users - Learn how to send a welcome email to end users after an upgrade.
- Downgrade Mattermost Server - Find out how to roll back to older versions of Mattermost.
- Open source components - Find out about the open source components used in Mattermost.

Application architecture

Workflow overview

Users connect through various access points (web, mobile, desktop, email). Their requests are processed by the application layer (Mattermost Server), which manages API communications, authentication, notifications, and data workflows.

The backend infrastructure supports these operations by storing all data and files in well-architected storage systems.

Extendability and security layers ensure that the platform integrates seamlessly with enterprise systems while protecting sensitive data.

Core components

The technical architecture revolves around 3 main layers: Access layer, Application layer, and Backend infrastructure.

Access layer

The Access Layer includes all the ways users interact with Mattermost, ensuring secure, scalable, and reliable communication across preferred platforms. High availability measures provide uninterrupted functionality for users, even in the face of server or network failures.

- **Web Interface**: Users can access Mattermost through a web browser (Chrome, Firefox, Safari, Edge). The web client communicates with the Mattermost server over HTTPS protocols. High availability for web interfaces can be achieved through load-balanced reverse proxies (e.g., NGINX or HAProxy) that distribute user traffic across multiple Mattermost server instances. Backup proxy servers can ensure failover scenarios to keep the web interface operational during outages.
- **Desktop and Mobile Apps**: Native apps for iOS, Android, macOS, Windows, and Linux provide seamless functionality across devices. These apps rely on secure APIs to interact with the server for real-time messaging and updates. High availability can be ensured by deploying redundant server clusters to handle API requests, along with failover mechanisms that automatically redirect traffic to healthy servers. Mobile apps also benefit from retry mechanisms and fallback services for push notifications to maintain real-time responsiveness.
- **Email Interaction**: Support for email clients like Outlook, Gmail, or Thunderbird enables integration of email notifications (e.g., new message alerts, invitations) into users' typical workflows. The Access Layer ensures that users are always connected via platforms of their choice while maintaining secure, synchronized communication paths. Email services can be configured with multiple SMTP servers for redundancy, ensuring that notifications are sent without delay even if a primary mail server becomes unavailable.

The Access Layer plays a critical role in ensuring that users are always connected via the platforms of their choice while maintaining secure, synchronized communication paths. With high availability measures in place,

organizations can guarantee a seamless user experience, regardless of the scale or complexity of their deployment.

Application layer

The Mattermost Server is the heart of the platform and responsible for processing all user and system operations. It's composed of multiple modular elements as follows:

RESTful JSON Web Service: Handles all incoming API requests (from web clients, apps, and integrations) and ensures that responses are formatted in JSON. Acts as the communication bridge between the clients (Access Layer) and backend systems. To ensure high availability, this layer can be distributed across multiple servers and load-balanced, preventing service disruptions due to high traffic or server failure.

Authentication client: Manages user authentication, ensuring secure login sessions. Integrates with traditional username/password-based authentication or enterprise-grade solutions like SSO (Single Sign-On) through Active Directory/LDAP. High availability is maintained through redundant

authentication nodes and failover mechanisms, ensuring uninterrupted access even if a primary authentication service fails.

Authentication Provider: Provides pluggable authentication frameworks to support OAuth, SSO, and third-party identity services. Particularly important for enterprise environments with centralized identity management. Redundancy and failover strategies ensure reliability by distributing authentication frameworks across multiple servers and offering fallback options for seamless identity management.

Notification Service: Sends notifications through supported mediums:

- **Push Notifications**: Real-time notifications to iOS and Android devices (via a Push Notification Service). High availabilty is ensured through multiple notification servers and retry mechanisms, guaranteeing that notifications are delivered even in the event of service disruptions.
- **Email Notifications**: Delivered to users when they are offline or need event alerts. Load balancing and backup mail server configurations help ensure email delivery remains consistent and reliable.

Data Management Service: Responsible for managing message data, metadata, user profiles, and logs. Ensures the integrity of data passed between the database and the server. This layer serves as the operational core of the platform, orchestrating user activities with data handling and integration capabilities.

High availability is achieved through database replication, failover strategies, and distributed data handling mechanisms. These measures ensure uninterrupted access to data and protect against component failures or downtime. The Data Management Service serves as the operational core of the platform, orchestrating user activities with scalable and fault-tolerant data handling capabilities.

Backend infrastructure

The backend infrastructure provides the storage and data handling capabilities required for Mattermost operations. It consists of the following components:

Database Systems: Mattermost uses PostgreSQL as its primary database (supports Amazon RDS for cloud-hosted PostgreSQL) to store all persistent data, such as:

- Messages
- User accounts and credentials
- Configuration settings
- Team/channel metadata

To ensure high availability, database systems can leverage clustering, replication, and failover mechanisms. PostgreSQL supports features like synchronous and asynchronous replication to create replicas for redundancy. Cloud-hosted solutions like Amazon RDS provide automatic failover and backup capabilities, ensuring continuous operation during system failures.

File Storage: Manages all multimedia assets (e.g., file uploads, images, videos) shared across channels. Storage solutions include the following options:

- **Local Storage**: Files stored directly on the server's filesystem. For high availability, redundancy can be achieved using RAID configurations or backups to recover from disk failures.
- **Network Attached Storage (NAS)**: Common for enterprises centralizing file storage within their network. NAS setups can include fault-tolerant configurations like distributed systems or replication for uninterrupted access.
- **S3**: Offers cloud-based scalable storage for larger environments or organizations with distributed deployments. The database and file storage handle scalability, ensuring efficient support for millions of messages and files while guaranteeing data consistency. S3 inherently supports high availability by distributing data across multiple availability zones, ensuring no single point of failure.

High availability measures ensure scalable and fail-safe support for millions of messages and files while guaranteeing data consistency.

System Extensions: Mattermost is not only a collaboration tool but also a platform designed for extensibility. Key extensibility features include:

Self-Hosted Integrations: Connect Mattermost to other local or cloud-based systems like Jira, GitLab, or any custom integrations your team needs. Leverage built-in APIs and webhooks to automate workflows and trigger system-to-system communications. For high availaiblity, integrations can employ redundant communication channels and retry mechanisms to handle transient failures gracefully.

Third-Party Authentication: Bind integrations to third-party platforms (e.g., Slack-importing APIs, OAuth services). Third-party identity services ensure consistent and secure user access flows. Third-party identity services can leverage load-balancing and failover strategies to ensure consistent and secure user access flows, even under high traffic or outages.

Security and Scalability Features: Security and scalability are baked into the architecture, making Mattermost ideal for enterprise use cases:

Security

- A reverse proxy like NGINX or a hardware proxy is deployed to manage external traffic. It protects servers, enforces HTTPS, and handles load balancing.
- Configurable SSL/TLS encryption ensures data security during transmission.
- Granular user permissions and roles secure sensitive information within teams.

Scalability and High Availability: The Enterprise Edition supports deploying multiple Mattermost servers in a clustered environment to balance user requests across multiple servers for reliability and performance in large organizations. Clustering ensures automatic failover so that user traffic is shifted to functioning servers in case of outages.

Notifications and communication services: Mattermost supports asynchronous and real-time communication, enhanced by notification systems tailored for different workflows:

- **Push Notifications**: Delivered to mobile devices for message alerts or mentions. High availability is achieved with backup notification services and retry mechanisms for reliable delivery.

- **Email Integration**: Provides regular notifications when users are offline or inactive. Failover mail servers and distributed configurations ensure email notifications are sent without interruption.

These services ensure continuous engagement and communication.

Communication protocols

There are also communication protocols (HTTPS and WS) that define the type of connection the user makes with the Mattermost server. High availability measures ensure reliable and resilient connections between clients and the Mattermost server, especially in production environments.

HTTPS Connection (Secure Hypertext Transfer Protocol)

- HTTPS connections to the Mattermost server render pages and provide access to core platform functionality, but do not include real-time interactivity (which is enabled by WSS connections).
- HTTPS is a secure, encrypted protocol and is highly recommended for production. Unencrypted HTTP connections may be used in initial testing and configuration, but should never be used in a production environment. For high availability, HTTPS traffic should be handled by a reverse proxy (e.g., NGINX or HAProxy) with load balancer configurations to distribute connections across multiple Mattermost server instances. Redundant proxy servers ensure failover capabilities, providing uninterrupted service.

WSS Connection (Secure WebSocket Protocol)

Secure WebSocket (WSS) connections to the Mattermost Server enable real-time updates and notifications between clients and the server.

If a WSS connection is not available and HTTPS is substituted, the system will appear to work but real-time updates and notifications will not. In this mode of operation, updates will only appear on a page refresh. WSS has a persistent connection to the Mattermost server when a client is connected, while HTTPS has an intermittent connection and only connects to the server when a page or file is requested.

High availability for WSS connections can be achieved through clustering Mattermost servers and load balancing WebSocket connections across those cluster nodes. Proxy servers and WebSocket-specific configurations (such as

sticky sessions or connection persistence) are essential to maintain real-time interactivity during server or network failures.

By incorporating high availability strategies into communication protocols, the platform ensures secure, scalable, and reliable connections for both regular user interactions (via HTTPS) and real-time updates (via WSS). These measures are critical for mission-critical environments and distributed deployments where continuous communication is necessary.

Behind a VPN

Mattermost is intended to be installed within a private network which can offer multiple factors of authentication, including secure access to computing devices and physical locations. If outside access is required, a virtual private network client (VPN), such as OpenVPN (https://openvpn.net/), with additional authentication used to connect to Mattermost for web, desktop, and mobile experiences, is recommended.

Non-VPN setup

If Mattermost is accessible from the open internet, the following is recommended:

1. An IT admin should be assigned to set up appropriate network security, subscribe to the Mattermost security bulletin (https://mattermost.com/security-updates/#sign-up), and apply new security updates.

2. The organization enables SAML Single Sign-on or enable MFA.

If Mattermost is accessible from the open internet with no VPN or MFA set up, we recommended using it only for non-confidential, unimportant conversations where impact of a compromised system is not essential.

Mattermost services ports

The following table lists the Mattermost services ports for Mattermost Server, push proxy, and mobile app clients. System admins with clients that need to speak to the Mattermost server without a proxy can open specific firewall ports as needed.

Mattermost Server

Service Name	Config Setting	Port (default)	Protocol	Direction	Info
HTTP/ Websocket	ServiceSettings.ListenAddress	8065/80/443 (TLS)	TCP	Inbound	Exter proxy Inter proxy this port 443 runni
Cluster	ClusterSettings.GossipPort	8074	TCP/UDP	Inbound	Inter
Metrics	MetricsSettings.ListenAddress	8067	TCP	Inbound	Exter proxy Inter proxy
Database	SqlSettings.DataSource	5432 (PostgreSQL) / 3306 (MySQL)	TCP	Outbound	Usual (reco
LDAP	LdapSettings.LdapPort	389	TCP/UDP	Outbound	
S3 Storage	FileSettings.AmazonS3Endpoint	443 (TLS)	TCP	Outbound	
SMTP	EmailSettings.SMTPPort	10025	TCP/UDP	Outbound	
Push Notifications	EmailSettings.PushNotificationServer	443 (TLS)	TCP	Outbound	

Push Proxy

Service Name	Config Setting	Port (default)	Protocol	Direction	Info
Push Proxy	ListenAddress	8066	TCP	Inbound	External (no proxy) / Internal (with proxy)

Mobile Clients

To receive push notifications, your network must allow traffic on port 5223 for iOS devices (https://support.apple.com/en-us/102266) and ports 5228-5230 for Android (https://firebase.google.com/docs/cloud-messaging/concept-options#messaging-ports-and-your-firewall).

On this page

- Application architecture
  - Workflow overview
  - Core components
    - Access layer
    - Application layer
    - Backend infrastructure
    - Communication protocols
      - Mattermost services ports

Backup and disaster recovery

Options to protect your Mattermost server from different types of failures range from simple backups to sophisticated disaster recovery deployments and automation.

Backup

The state of your Mattermost server is contained in multiple data stores that need to be backed up and restored separately to fully recover your system from failure.

To back up your Mattermost server:

1. Back up your Mattermost database using standard procedures depending on your database version. PostgreSQL SQL Dump backup documentation is available online. Use the navigation at the top of the page to select your PostgreSQL version.
2. Back up your server settings stored in `config/config.json`. If you are using SAML configuration for Mattermost, your SAML certificate files will be saved in the `config` directory. Therefore, we recommend backing up the entire directory.
3. Back up files stored by your users with 1 of the following options:

> - *If you use local storage using the default `./data` directory, back up this directory.*
> - *If you use local storage using a non-default directory specified in the `Directory` setting in `config.json`, back up files in that location.*
> - *If you store your files in S3, you can typically keep the files where they are located without backup.*

> Note
>
> To make a clean backup, you must stop Mattermost during the duration of the backup, otherwise the database and files may become out of sync.

To restore a Mattermost instance from backup, restore your database, `config.json` file, and optionally the locally stored user files into the locations from which they were backed up.

Disaster recovery

An appropriate disaster recovery plan weighs the benefits of mitigating specific risks against the cost and complexity of setting up disaster recovery infrastructure and automation.

Automated backup

Automating backups for a Mattermost server provides a copy of the server's state at a particular point in time, which can be restored if a failure in the future leads to loss of data. Options include:

- Automation to periodically back up the Mattermost server, which may include all the components listed above or a subset depending on what you choose to protect.
- Automation to restore a server from backup, or deploy a new server, to reduce recovery time.
- Automation to verify a backup has been successfully produced to protect against backup automation failures.
- Storing backups off-site, to protect against physical loss of onsite systems.

Recovering from a failure using a backup is typically a manual process and will incur downtime. The alternative is to automate recovery using a high availability deployment.

High Availability deployment

Enterprise customers who use Mattermost for mission-critical operations must ensure continuous availability and operational resilience. A robust disaster recovery strategy is essential to mitigate risks associated with data center failures, ensuring that users can access Mattermost seamlessly, even in the event of unexpected outages.

This section details the steps needed to set up Mattermost in a disaster recovery mode, and how to fail over from one data center to another.

> Tip
>
> To learn how to safely upgrade your deployment in Kubernetes for High Availability and Active/Active support, see the Upgrading Mattermost in Kubernetes and High Availability Environments documenation.

Set up in one data center

As a first step, set up Mattermost in a single data center. At a very basic high level, this would be something like below:

The diagram above has a single proxy, forwarding traffic to 2 nodes. There's also a database with single writer + n readers and an S3 bucket and ES/OS using AWS OpenSearch service.

At this stage, we are ignoring other details like LDAP/SAML, SMTP etc.

> Tip
>
> The following architecture would be implemented when an entire region goes down. It does not cover the case when a single server/service goes down. For example:
>
> - If a single app node goes down, follow best practices to provision a new node.
> - If a database replica node goes down, create a new replica from AWS console. Or set a policy to do so automatically.

Replicate database

The next tasks include creating a global AWS Cluster.

1. Select the RDS instance in the AWS Console, and expand the **Actions** menu to select **Add AWS Region**.
2. Choose the secondary region and enter the other details.

> Warning
>
> Select the **Enable write forwarding** option on the secondary cluster to help forward write operations from secondary to primary. See the [AmazonRDS write forwarding (https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding-apg.html#aurora-global-database-write-forwarding-enabling-apg)](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding-apg.html#aurora-global-database-write-forwarding-enabling-apg) documentation for details.
>
> Also verify the PostgreSQL version and ensure it allows `write forwarding`. Not all PostgreSQL versions allow it. See the [Amazon RDS write forwarding region and version availability (https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding-apg.html#aurora-global-database-write-forwarding-regions-versions-apg)](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-write-forwarding-apg.html#aurora-global-database-write-forwarding-regions-versions-apg) documentation for details.

You should now have a global cluster with the primary cluster in `us-west-1`, and the secondary cluster in `us-east-1`:

DB identifier	Status	Role	Engine	Region & AZ
agnivatest-global	Available	Global database	Aurora PostgreS...	2 regions
agnivatest-db	Available	Secondary cluster	Aurora PostgreS...	us-east-1
agnivatest-db-0	Available	Reader instance	Aurora PostgreS...	us-east-1d
agnivatest-db-1	Available	Reader instance	Aurora PostgreS...	us-east-1c
agnivatest-global-cluster-1	Available	Primary cluster	Aurora PostgreS...	us-west-1
agnivatest-global-instance-1	Available	Writer instance	Aurora PostgreS...	us-west-1c
agnivatest-global-instance-1-u	Available	Reader instance	Aurora PostgreS...	us-west-1a

Replicate S3 bucket

1. Create a new S3 bucket in the secondary region.
2. Back in the original bucket, go to the **Properties** tab, and enable **Bucket versioning**.
3. Go to the **Management** tab, scroll down to **Replication Rules**, and create a new replication rule.

4. In the rule, select the source bucket, and then choose **Apply to all objects in the bucket** to replicate everything in the bucket.
5. Choose the destination bucket.
6. For the IAM role, select **Create new role**.

> Warning
>
> Select the **Replica modification sync** option for the bucket to help keep the replica and source buckets in sync with each other.

7. Select **Save**.
8. Select **Yes** when prompted to start a job to replicate any existing objects to the secondary bucket or not.
9. Perform these same steps on the secondary bucket.

Now you have bi-directional replication working between these S3 replica and source buckets.

Replicate ES/OS storage

1. To replicate ES/OS storage, set up CCR (cross-cluster replication) for AWS OpenSearch with the following requirements:

> - *Elasticsearch 7.10 or OpenSearch 2.x*
> - *Fine-grained access control enabled*
> - *Node-to-node encryption enabled*

> Tip
>
> All you need is a recent OpenSearch version with fine-grained access control enabled. Node-to-node encryption is automatically enabled once you enable fine-grained access control.

2. You also need to add the `CrossClusterGet` permission on the IAM policy for the OS cluster set under the **Security Configuration** tab for your OS domain. We recommend the following as per AWS, but feel free to fine-tune as necessary:

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
"Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource":
"arn:aws:es:<region>:<acc_num>:domain/<domain_name>/*" }, { "Effect":
"Allow", "Principal": { "AWS": "*" }, "Action":
"es:ESCrossClusterGet", "Resource":
"arn:aws:es:<region>:<acc_num>:domain/<domain_name>" } ] }
```

To recap:

- Use OpenSearch 2.x.
- Enable fine-grained access control.
- Create the master user, and note the server credentials.
- Set the IAM policy as above.

> Warning
>
> After creating the master user, IP based access to the OS might not work from Mattermost application nodes. You may need to update the `ElasticSearchSettings` section in `config.json` to update the server username and password.

3. Create a new OS cluster in the secondary region. Follow the same steps again for this cluster.

> *Warning*
>
> *At this stage, ensure that you have all indices populated with data in the primary region. Run a bulk index to do that if you haven't already.*

4. Begin replication from the primary to secondary region.

> 1. *First, create a connection from secondary to primary. Note that replication in OS works in a "pull" model, so the secondary site pulls data from the primary.*
> 2. *In the Amazon OpenSearch Service console, select the secondary domain, go to the **Connections** tab, and choose **Request**.*
> 3. *For **Connection alias**, enter a name for your connection.*
> 4. *Choose **connect to a domain in another AWS account or region**, and enter the **ARN** of the primary domain.*
> 5. *Select **Request** to send a permission request to the primary domain.*
> 6. *Open the primary domain to see and accept the incoming request under the **Connections** tab.*

5. Now set up the replication rules for indices.

> 1. *SSH into an app node in the secondary region to set up an auto-follow rule for the `posts*` indices because of the daily naming scheme and monthly aggregation.*
> 2. *For the other indices, replicate each of them. You can also set up a rule with `*` to replicate everything, but that would also include the hidden and system indices which you don't want.*
> 3. *Set up the auto-follow for `posts*` indices:*

```
curl -XPOST -H 'Content-Type: application/json' -u
'<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_plugins/_replication/
_autofollow?pretty' -d ' {  "leader_alias" :
"<LEADER_ALIAS>",  "name": "autofollow-rule",  "pattern": "posts*",
"use_roles":{  "leader_cluster_role": "all_access",
"follower_cluster_role": "all_access"  } }'
```

4. Check the status of the auto-follow rule:

```
curl -H 'Content-Type: application/json' -u 'username/password'
'https://<>/_plugins/_replication/autofollow_stats?pretty' {
"num_success_start_replication" : 2,  "num_failed_start_replication" :
0,  "num_failed_leader_calls" : 0,  "failed_indices" : [ ],
"autofollow_stats" : [  {  "name" : "autofollow-rule",  "pattern" :
"posts*",  "num_success_start_replication" : 2,
"num_failed_start_replication" : 0,  "num_failed_leader_calls" : 0,
"failed_indices" : [ ],  "last_execution_time" : 1737699113927  }  ] }
```

5. Next, set up replication for the other indices:

```
curl -XPUT -H 'Content-Type: application/json' -u
'<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_plugins/_replication/
channels/_start?pretty' -d ' {  "leader_alias": "<LEADER_ALIAS>",
"leader_index": "channels",  "use_roles":{  "leader_cluster_role":
"all_access",  "follower_cluster_role": "all_access"  } }' curl -XPUT
-H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/users/_start?pretty' -d ' {
"leader_alias": "<LEADER_ALIAS>",  "leader_index": "users",
"use_roles":{  "leader_cluster_role": "all_access",
"follower_cluster_role": "all_access"  } }' curl -XPUT -H 'Content-
Type: application/json' -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/
_plugins/_replication/files/_start?pretty' -d ' {  "leader_alias":
"<LEADER_ALIAS>",  "leader_index": "files",  "use_roles":{
"leader_cluster_role": "all_access",  "follower_cluster_role":
"all_access"  } }'
```

6. Check the status of the replication rules:

```
curl -H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/channels/_status?pretty'
curl -H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/files/_status?pretty' curl
-H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/users/_status?pretty' curl
-H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/posts_<DATE>/_status?pretty'
curl -H 'Content-Type: application/json' -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/_plugins/_replication/posts_<DATE>/_status?pretty'
Sample output: {  "status" : "SYNCING",  "reason" : "User initiated",
"leader_alias" : "<LEADER_ALIAS>",  "leader_index" : "<INDEX>",
"follower_index" : "<INDEX>",  "syncing_details" : {
"leader_checkpoint" : 16,  "follower_checkpoint" : 16,  "seq_no" :
17  } }
```

7. *Check for indices. You should be able to see all the indices from the primary domain in the secondary domain:*

```
curl -s -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_cat/indices?
pretty'
```

Replicate job servers

If the job scheduler is left running in the secondary region, it will pick up jobs and start running them. Therefore, set `JobSettings.RunScheduler` to `false` on all nodes in the secondary region. When a failover happens, you need to enable it for the new primary region, and deactivate it for the new secondary region.

Test the secondary region

With the above steps complete, you have a fully functioning secondary region. You can replicate the same setup of nodes and a proxy server like the primary region. The app nodes in the secondary region won't be able to come up the first time because Mattermost will try to run some DDL statements which are not allowed with write-forwarding. So it will be stuck in a loop trying to connect. Once you fail over the region, it will start working. The primary region will still be readable, and any periodic writes will be forwarded to the secondary (now primary).

> Warning
>
> Ensure you have separate `ClusterNames` for the different clusters in two regions to use the same database across 2 clusters.

Failover RDS to secondary

To perform the failover, go to the RDS global cluster, and under **Actions**, select **Switchover or Failover global database**, and then select **switchover** to switch over without any data loss (which will take more time to complete). Alternatively, you can choose **failover** for a quicker failover at the expense of data-loss. If the entire region is unavailable anyways, then **failover** is no worse than **switchover**.

After this is done, the app nodes which were stuck trying to connect should move forward and everything should be functional. You can read/write, upload images and everything should be replicated. Everything except OpenSearch data.

Failover ES/OS to secondary

ES/OS does not allow multi-writer for a single index. You can only write to 1 index at one time. Therefore, you need to perform some manual steps to reverse the replication direction, and start replicating from secondary to primary.

For simplicity, let's say `site1` is primary, and `site2` is secondary. Therefore, OS in `site1` is the leader domain, and in `site2` is the follower. The follower pulls from the leader. To switch the direction where `site2` becomes leader, and `site1` becomes follower.

1. Remove the rule from `site1` → `site 2` in AWS Console. This will auto-pause the replication, but the indices in `site2` will still be read-only. Remove the replication rules for that.

2. Remove auto-follow rule:

```
curl -XDELETE -H 'Content-Type: application/json' -u
'<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_plugins/_replication/
_autofollow?pretty' -d ' {  "leader_alias" : "<LEADER_ALIAS>",  "name":
"autofollow-rule" }'
```

3. Check the status of the auto-follow rule as mentioned before.
4. Remove replication rules:

```
curl -XPOST -H 'Content-Type: application/json' -u
'<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_plugins/_replication/
channels/_stop?pretty' -d '{}' curl -XPOST -H 'Content-Type: application/
json' -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_plugins/
_replication/files/_stop?pretty' -d '{}' curl -XPOST -H 'Content-Type:
application/json' -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/
_plugins/_replication/users/_stop?pretty' -d '{}'
```

5. Check the status of replication rules as mentioned before.
6. Now indices will become writable
7. Add rule from `site2` → `site1` in AWS console.
8. In `site1`, make all the indices as followers. You must delete all indices first:

```
curl -XDELETE -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/posts*?
pretty' curl -XDELETE -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/
channels?pretty' curl -XDELETE -u '<USERNAME>:<PASSWORD>' 'https://
<HOSTNAME>/files?pretty' curl -XDELETE -u '<USERNAME>:<PASSWORD>'
'https://<HOSTNAME>/users?pretty'
```

9. Refresh indices:

```
curl -XPOST -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_refresh?
pretty'
```

10. Confirm that everything is deleted:

```
curl -s -u '<USERNAME>:<PASSWORD>' 'https://<HOSTNAME>/_cat/indices?
pretty'
```

11. Add the auto-follow rule add replication rules. Follow the same steps as before.
12. List the indices again to confirm that replication has started and indices are available.

S3 bucket is auto-replicated both ways

There's nothing you need to do to ensure the S3 bucket is auto-replicating both ways.

Testing end to end

Once the failover has happened, and the ES/OS replication direction has been swapped, the new site can be used normally.

This becomes the final architecture:

You can use DNS to easily switch between PRIMARY to SECONDARY during a failover.

> Tip
>
> Websockets will still point to the old data center even if you have switched DNS. You need to roll over each app node gradually to move those connections to the new data center. If all your nodes are down, no action is necessary and the clients will automatically re-connect to the new data center.

The S3 bucket is replicated bi-directionally while the database and ES/OS is replicated uni-directionally.

Failover from Single Sign-On outage

When using Single Sign-on with Mattermost Enterprise Edition an outage to your SSO provider can cause a partial outage on your Mattermost instance.

What happens during an SSO outage?

- **Most people can still log in.** By default, when a user logs in to Mattermost they receive a session token lasting 30 days (the duration can be configured in the System Console). During an SSO outage, users with valid session tokens can continue to using Mattermost uninterrupted.
- **Some people can't log in.** During an SSO outage, there are two situations under which a user cannot log in:
  - Users whose session token expires during the outage.
  - Users trying to log in to new devices.

In each case, the user cannot reach the SSO provider, and cannot log in. In this case, there are several potential mitigations:

Configure your SSO provider for High Availability

If you're using a self-hosted Single Sign-on provider, several options are available for [High Availability configurations that protect your system from unplanned outages (https://learn.microsoft.com/en-us/microsoft-identity-manager/pam/high-availability-disaster-recovery-considerations-bastion-environment)](https://learn.microsoft.com/en-us/microsoft-identity-manager/pam/high-availability-disaster-recovery-considerations-bastion-environment).

For SaaS-based authentication providers, while you still have a dependency on service uptime, you can set up redundancy in source systems from which data is being pulled. For example, with the OneLogin SaaS-based authentication service, you can set up High Availability LDAP connectivity to further reduce the chances of an outage.

Set up your own IDP to provide an automated or manual SSO failover option

Create a custom Identity Provider for SAML authentication that connects to both an active and a standby authentication option, that can be manually or automatically switched in case of an outage.

In this configuration, security should be carefully reviewed to prevent the standby SSO option from weakening your authentication protocols.

Set up a manual failover plan for SSO outages

When users are unable to reach your organization's SSO provider during an outage, an error message directing them to contact your support link (defined in your System Console settings) is displayed.

Once IT is contacted about an SSO outage issue, they can temporarily change a user's account from SSO to email-password using the System Console, and the end user can use password to claim the account, until the SSO outage is over and the account can be converted back to SSO.

When the outage is over, it's critical to switch everyone back to SSO from email-password to maintain consistency and security.

On this page

- Backup and disaster recovery
  - Backup
  - Disaster recovery
    - Automated backup
    - High Availability deployment
      - Set up in one data center
      - Replicate database
      - Replicate S3 bucket
      - Replicate ES/OS storage
      - Replicate job servers
      - Test the secondary region
      - Failover RDS to secondary
      - Failover ES/OS to secondary
      - S3 bucket is auto-replicated both ways
      - Testing end to end
    - Failover from Single Sign-On outage
    - Configure your SSO provider for High Availability
    - Set up your own IDP to provide an automated or manual SSO failover option

- Set up a manual failover plan for SSO outages

Deployment Guide

Welcome to deployment guidance for Mattermost. This guide is organized into sections based on application types and deployment scenarios to help you achieve a successful deployment of Mattermost across various platforms.

Whether you're deploying the server application, desktop application, or mobile application, or troubleshooting deployments, this guide has you covered. Use the navigation below to access detailed information about each topic.

- Quick Start Evaluation - A quick start guide to help you get started with Mattermost.
- Application architecture - An overview of Mattermost's architecture and components.
- Server deployment - Pre-deployment checks, security considerations, hardware recommendations, software prerequisites, and step-by-step guidance to deploying Mattermost server.
- Desktop deployment - Installation procedures for Mattermost's desktop applications across Windows, macOS, and Linux, and large-scale, enterprise-wide deployments.
- Mobile deployment - How to set up push notifications using Mattermost's notification service and troubleshooting tips.
- Deployment troubleshooting - Best practices for diagnosing and resolving common deployment issues.

How to use this guide

Navigate through the sections using the headings above to find the deployment instructions and troubleshooting steps pertinent to your needs. Each section is designed to provide clear, actionable information to ensure a successful deployment and operation of Mattermost.

If you are new to Mattermost, we recommend starting with the Application Architecture section to understand the prerequisites and get started, or alternatively use the Try Mattermost section to explore Mattermost before planning your production deployment.

> Note
>
> - If you encounter issues that aren't covered in this documentation:
> - Non-commercial customers: Reference the Mattermost community forums (https://forum.mattermost.com/)
> - Commercial customers: See the Mattermost Support Knowledge Base (https://support.mattermost.com/hc/en-us), or contact Mattermost Support (https://support.mattermost.com/hc/en-us/requests/new) for assistance.
> - For advanced customization or integrations, refer to the Open source components documentation for details about extending Mattermost functionality.

Enjoy deploying Mattermost with confidence!

On this page

- Deployment Guide
  - How to use this guide

Deployment troubleshooting

These guides will help you troubleshoot aspects of your Mattermost deployment.

- General deployment troubleshooting
- Docker deployment troubleshooting
- Desktop app installation troubleshooting
- Mobile applications troubleshooting
- PostgreSQL installation troubleshooting
- MySQL installation troubleshooting

Desktop App Deployment

The Mattermost desktop app is available for Windows, macOS, and Linux operating systems, and offers additional functionality beyond the web-based experience.

Learn more about desktop app software requirements, releases and server compatibility as well as the what's changed across releases.

Download

Download and install the Mattermost desktop app from the App Store (macOS) (https://www.apple.com/app-store/), Microsoft Store (Windows) (https://apps.microsoft.com/home?hl=en-US&gl=US), or by using a package manager (Linux).

In Matermost, you're notified under **Downloads** when new desktop app releases become available, where you can view the changelog and download updates.

We strongly recommend installing the desktop app on a local drive. Network shares aren't supported.

If you prefer to manage distribution of the mobile app to your users, see the additional deployment options below.

Deployment options

Learn about installation, configuration, and management options for deploying the desktop app in your environment.

- Distribute a custom desktop app
- Silent Windows desktop distribution
- MSI installer and group policy guide
- Custom dictionaries for Windows and Linux

- Managed resources for the desktop app

On this page

- Desktop App Deployment
  - Download
  - Deployment options

Desktop managed resources

The Mattermost desktop app supports managed resources. A managed resource can be any service available on the same hostname using the same protocol as the Mattermost server.

> Note
>
> Using this feature requires a *custom build of the Mattermost desktop app </deployment-guide/desktop/distribute-a-custom-desktop-app>*.

Add the path of a managed resource to your configuration file. When selected, it opens as a pop-up window in the Mattermost desktop app.

In addition to customizing the Mattermost Desktop App, the Managed Resource Paths setting on the Mattermost server must be configured.

In the below example we add the managed resource `/video`.

```
[...] managedResources: ['trusted', 'video'], [...]
```

Here are two example server URLs each with valid and invalid managed resource URLs:

- Mattermost server: `https://mattermost.my.org`
  - A valid video service: `https://mattermost.my.org/video`
  - A valid conference service: `https://mattermost.my.org/conference`
  - An invalid video service using a different protocol: `http://mattermost.my.org/video`
  - An invalid conference service having a different origin: `https://conference.my.org`

- Mattermost server: `https://my.org/mattermost`
  - A valid video service: `https://my.org/video`
  - A valid conference service: `https://my.org/conference`
  - An invalid video service using a different protocol: `http://my.org/video`
  - An invalid conference service having a different origin: `https://conference.my.org`

Desktop App custom dictionaries

From Mattermost desktop app v4.7.1, custom dictionary definitions can be served through a URL on Windows and Linux. If custom dictionaries aren't specified, default dictionary definitions are obtained automatically from Chromium's CDNs (content delivery networks). On macOS, the Mattermost desktop app uses dictionary definitions provided by Apple that can't be customized.

Prepare custom dictionaries

On Windows and Linux, the Mattermost desktop app uses [Hunspell Dictionary definitions (https://hunspell.github.io/)](https://hunspell.github.io/) by default. You can download a copy of these dictionary definitions and modify them to fit specific needs.

To access a copy of these dictionary definitions:

1. Download a copy of `hunspell_dictionaries.zip` from the [latest electron release (https://github.com/electron/electron/releases/latest)](https://github.com/electron/electron/releases/latest).
2. Extract the dictionary definitions from the ZIP file.
3. Convert the filenames to lowercase. (Chromium expects `en-us` rather than `en-US`). Once renamed, these definition files match the usage of using Chromium's CDN ones, and you can modify the definitions as needed.
4. Serve the extracted and converted files from a web server, and note the URL to the root folder of the dictionaries. Specify this URL as the **alternate dictionary URL** in the desktop app.

Configure the Desktop App

1. In the Mattermost Desktop App, go to Settings by selecting

   ⋮

   > File > Settings.
2. Under **App Options**, ensure that the **Check spelling** option is enabled.

3. Select **Use an alternative dictionary URL** to specify the URL to the root folder of your custom dictionaries, then select **Save**.

Remove custom dictionaries

Select **Revert to default** to stop using the specified dictionary definitions, and revert to default dictionary definitions.

On this page

- Desktop App custom dictionaries
  - Prepare custom dictionaries
  - Configure the Desktop App
  - Remove custom dictionaries

Desktop MSI installer and group policy guide

This page provides guidance on installing the desktop app MSI and use Group Policies in Windows for Mattermost Enterprise or Professional. The MSI installer package can be downloaded here (https://github.com/mattermost/desktop/releases/tag/v5.12.1).

> Tip
>
> Want to distribute the official Windows desktop app silently to your end users instead?

Download group policy and MSI installer files

1. Using a newly created Windows VM or dedicated Windows computer, make sure to use a Windows version that supports `Edit group policy` out of the box (i.e. Windows 10 Pro or Enterprise).

2. Navigate to the [Mattermost Desktop (https://github.com/mattermost/desktop)](https://github.com/mattermost/desktop) repository on [GitHub.com (https://github.com/)](https://github.com/).

3. Navigate to the release page for [version v5.12.1 (https://github.com/mattermost/desktop/releases/tag/v5.12.1)](https://github.com/mattermost/desktop/releases/tag/v5.12.1) and download the appropriate installer for your version of Windows (32-bit vs. 64-bit).

4. Download the [source.zip (https://github.com/mattermost/desktop/archive/v5.12.1.zip)](https://github.com/mattermost/desktop/archive/v5.12.1.zip) file as well to extract group policy files.

Install group policy files locally

The following group policies are available supporting a state option of Not Configured, Enabled, or Disabled:

Group policy	***Description***	***Mattermost release***	***Setting***
Enable Server Management	*If disabled, management of servers in the app settings are disabled.*	*v4.3 or later*	`EnableServerManagement`
Default Server List	*Define one or more default, permanent servers.*	*v4.3 or later*	`DefaultServerList`
Automatic Updates	*If disabled, automatic desktop app updates are disabled.*	*v5.1 or later*	`EnableAutoUpdates`

1. Browse to the folder the above files were downloaded to and unzip the `desktop-5.12.1.zip` file in place.

2. Navigate to the unzipped `desktop-5.12.1\resources\windows\gpo` folder and copy the contents.

3. Navigate to the `C:\Windows\PolicyDefinitions` folder and paste the files copied in the last step.

4. Verify the `mattermost.admx` file is in the `C:\Windows\PolicyDefinitions` folder.

5. Verify the `mattermost.adml` file is in the `C:\Windows\PolicyDefinitions\en-US` folder.

> Note
>
> - `\\FQDNDomain\sysvol\FQDNDomain\Policies\PolicyDefinitions` can be used instead of `C:\Windows\PolicyDefinitions` if available.
> - `\\FQDNDomain\sysvol\FQDNDomain\Policies\PolicyDefinitions\en-US` can be used instead of `C:\Windows\PolicyDefinitions\en-US` if available.

Disable automatic updates

Automatic desktop app updates can be disabled by configuring the supported group policy. Changes to group policies require you to restart Mattermost for those changes to take effect.

Configure Mattermost using group policy settings

1. Run the `Edit group policy` application by selecting **Start**, typing `gpedit` into the search field, then selecting the resulting **Edit group policy** search option.

2. In the **Edit group policy** window, navigate to `Local Computer Policy\Computer Configuration\Administrative Templates\Mattermost`. In this

example, double-click on `DefaultServerList` to set one or more default servers that will appear on app launch.

3. In the resulting window for **DefaultServerList**, select **Enabled** to turn the feature on, then select the **Show...** button in the **Options:** section of the window to add default servers.

4. In the resulting window, add desired Mattermost servers using a memorable name (i.e., Community) and the web URL of the Mattermost server (i.e., https://community.mattermost.com).

5. Select **OK** twice, then close the **Edit group policy** app.

Verify group policy settings have been applied

1. Open up the Registry Editor by selecting **Start**, typing `Registry Editor` in the search field, then selecting the **Registry Editor** option in the search results.

2. In the **Registry Editor** window, navigate to `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Mattermost\DefaultServerList` and verify the servers you added using the **Edit group policy** app are listed.

3. Once verified, close the **Registry Editor**.

Install the Mattermost Desktop App using the MSI installer

Important

- If the desktop app is running when you install via the MSI, Mattermost prompts you to close the app manually. After acknkowledging the prompt, select **Retry** to continue the MSI installation.
- Avoid selecting **Ignore**. If you do, force close the desktop app using Task Manager, ensure the `Mattermost.exe` process is stopped, and then restart the MSI installation.

1. Within the folder the MSI installer was downloaded to, double-click on the MSI installer to begin the Mattermost Desktop installation process.

2. Installation of the MSI requires admin permission, so accept the resulting request to allow the installer to make changes to your device.

3. Select **Finish** when the installation is complete.

Verify group policy settings in the installed desktop app

1. Launch the newly installed Mattermost app from the **Start** menu.
2. Verify the app loads the first server you defined in the **Edit group policy** app.

Advanced MSI options

Important

You must be a system admin to run these commands, or you must run them from an admin command prompt or PowerShell.

Silent installation

Perform a silent installation of the MSI by running the following command:

Important

Ensure the desktop app is closed before proceeding with a silent installation. Because it's a silent installation, Mattermost won't prompt you to close the desktop app.

Perform a silent installation of MSI by running the following command:

```
msiexec /i mattermost-desktop-v5.12.1-x64.msi /qn
```

> Note
>
> - You'll need to update the version details in this command as new versions of the Mattermost desktop app are released.

From version v5.9.0 of the Mattermost desktop app, the following silent MSI installation options are also available.

Install for all users

Use the `ALLUSERS` parameter to install the MSI for all users:

```
msiexec /i mattermost-desktop-v5.12.1-x64.msi ALLUSERS=1
```

> Note
>
> - Installing the MSI for all users disables automatic updates for the desktop app on Windows.
> - To disable automatic updates on a per-user basis, use the `DISABLEAUTOUPDATE` parameter: `msiexec /i mattermost-desktop-v5.11.2-x64.msi DISABLEAUTOUPDATE=1`

Specify an install directory

Use the `APPLICATIONFOLDER` parameter to specify an installation directory for the MSI installation:

```
msiexec /i mattermost-desktop-v5.12.1-x64.msi APPLICATIONFOLDER=<install directory>
```

Change this command as new versions of the Mattermost Desktop App are released.

On this page

- Desktop MSI installer and group policy guide
  - Download group policy and MSI installer files
  - Install group policy files locally
  - Configure Mattermost using group policy settings
    - Verify group policy settings have been applied
  - Install the Mattermost Desktop App using the MSI installer
    - Verify group policy settings in the installed desktop app
  - Advanced MSI options
    - Silent installation
    - Install for all users
    - Specify an install directory

Desktop app troubleshooting

Where is configuration stored locally?

The location of the Mattermost desktop app configuration file depends on the platform where you're running Mattermost (and, in the case of macOS, how you've chosen to install the app):

- Windows: `Users\<username>\AppData\Roaming\Mattermost`
- macOS installer: `/Users/<username>/Library/Application Support/Mattermost`
- macOS App Store: `/Users/<username>/Library/Containers/Mattermost.Desktop/Data/Library/Application Support/Mattermost` (via Finder: `~/Library/Application Support/Mattermost` as the extension is hidden)
- Linux: `~/.config/Mattermost`

> Note
>
> - Local configuration data is not automatically removed when uninstalling the desktop app. If you wish to remove all data, you must manually remove the files from the applicable location noted above.
> - Prior to uninstalling, you can choose to log out of any active sessions. You can terminate active sessions from another Mattermost session in **Profile > Security > View and Logout of Active Sessions**, then select **Log Out**. Desktop app sessions are labeled as **Native Desktop App**.

How do I access logs?

From Mattermost desktop v5.3, you can access logs via **Help > Show logs**, which opens the file manager window showing the location of the log file.

How do I download app diagnostics?

From Mattermost desktop v5.3, you can download a diagnostics text file via **Help > Run diagnostics**, which can be attached to a Support ticket.

Desktop App displays white screen while launching and doesn't load the page

1. Delete the local `Mattermost desktop app` configuration file. See the Where is configuration stored locally? section above for file location details.
2. Reinstall the application.

"Installation has failed" dialog

The app data might be corrupted. Remove all the files in `%LOCALAPPDATA%\mattermost`, then try reinstalling the app.

"The application "Mattermost" can't be opened" dialog

On macOS Catalina, this dialog can be triggered if the Mac Archive Utility is the default method for decompressing files. In this case using a third-party tool such as [Keka (https://www.keka.io)](https://www.keka.io) or [Unarchiver (https://macpaw.com/the-unarchiver)](https://macpaw.com/the-unarchiver) may resolve the problem.

Desktop App window is black and doesn't load the page

1. Ensure you have installed the latest desktop app version available.
2. Clear your cache and reload the app from **View > Clear Cache and Reload** or press `Ctrl Shift R` on Windows or Linux, or ⌘ ⇧ R on Mac.
3. Quit the app and restart it to see if the issue clears.

4. Disable GPU hardware acceleration.

> - *On Windows or Linux, select **File > Settings** and clear the **Use GPU hardware acceleration** option.*
> - *On macOS, select **Mattermost > Settings** and clear the **Use GPU hardware acceleration** option.*

5. If you are using a special video driver, such as Optimus, try disabling it to see if the problem is resolved.

Desktop App window is white and doesn't load the page

1. Ensure you have installed the latest desktop app version available.
2. Delete the `%userprofile%\AppData\Roaming\Mattermost` directory on your local machine.
3. Reinstall the desktop app.

Desktop App is not visible, but the Mattermost icon is in the Task Bar

This issue can occur on Windows in a multiple-monitor setup. When you disconnect the monitor that Mattermost is displayed on, Mattermost continues to display at screen coordinates that no longer exist.

To resolve this issue, you can reset the desktop app screen location by deleting the screen location file. When the file is not present, the desktop app displays on the primary monitor by default.

To reset the desktop app screen location:

1. If the desktop app is running, right-click the Mattermost icon in the task bar, then select **Close Window**.
2. Open Windows File Explorer, and go to the `%APPDATA%\Mattermost` folder.
3. Delete the file `bounds-info.json`.

Desktop App constantly refreshes the page

This issue can occur when `localStorage` has an unexpected state. To resolve the issue:

- Windows: Open Windows File Explorer, go to the `%APPDATA%\Mattermost` folder, then delete the `Local Storage` folder.
- Mac: Open Finder, go to the `~/Library/Application Support/Mattermost` folder, then delete the `Local Storage` folder.
- Linux: Open the File Manager, go to the `~/.config/Mattermost` folder, then delete the `Local Storage` folder. Linux file managers may hide folders starting with a period by default. You can delete them from the terminal using `rm -rf ~/.config/Mattermost`.

Desktop App constantly asks to log in to Mattermost server

This issue can occur after a crash or unexpected shutdown of the desktop app that causes the app data to be corrupted. To resolve the issue:

- Windows: Open Windows File Explorer, go to the `%APPDATA%\Mattermost` folder, then delete the `IndexedDB` folder and the `Cookies` and `Cookies-journal` files.
- Mac: Open Finder, go to the `~/Library/Application Support/Mattermost` folder, then delete the `IndexedDB` folder and the `Cookies` and `Cookies-journal` files.
- Linux: Open the file manager, go to the `~/.config/Mattermost` folder, then delete the `IndexedDB` folder and the `Cookies` and `Cookies-journal` files. Linux file managers may hide folders starting with a period by default. You can delete them from the terminal using `rm -rf ~/.config/Mattermost`.

"Internal error: BrowserWindow 'unresponsive' event has been emitted"

Selecting **Show Details** on the dialog provides logs. Ways to resolve the issue:

1. Clear the cache via **View > Clear Cache and Reload** or press `Ctrl Shift R` on Windows or Linux, or ⌘ ⇧ R on Mac.
2. Go to App Settings via **File > Settings** (or by pressing `Ctrl ,` on Windows or Linux, or ⌘ , on Mac) and unselect hardware acceleration.

Desktop app not responsive within Citrix Virtual Apps or Desktop Environment

Append `Mattermost.exe;` to the Registry Key `HKLM\SYSTEM\CurrentControlSet\Services\CtxUvi\UviProcessExcludes` and reboot the system.

For further assistance, review the [Troubleshooting forum (https://forum.mattermost.com/c/trouble-shoot/16)](https://forum.mattermost.com/c/trouble-shoot/16) for previously reported errors, or [join the Mattermost user community for troubleshooting help (https://mattermost.com/community/)](https://mattermost.com/community/).

Can I uninstall the desktop app I installed using snap on Linux?

Yes. Run the following command from a terminal window: `sudo snap remove mattermost-desktop`.

Report Desktop App issues

When reporting issues found in the Mattermost desktop app, it's helpful to include the contents of the Developer Tools Console along with [the information on this page (https://support.mattermost.com/hc/en-us/articles/360060662492-Opening-a-Support-Ticket-for-Self-Managed-Deployments)](https://support.mattermost.com/hc/en-us/articles/360060662492-Opening-a-Support-Ticket-for-Self-Managed-Deployments).

To access the Developer Tools Console:

1. In the menu bar, go to **View > Developer Tools > Developer Tools for Current Tab**.
2. Select the **Console** tab.
3. Right-click the log entry, then select **Save As**.
4. Save the file, then send it along with a description of your issue.
5. Close the console to disable the Developer Tools.

You can open an additional set of developer tools for each server you have added to the desktop app. The tools can be opened by pasting this command in the Developer Tools Console you opened with the steps described above:

```
document.getElementsByTagName("webview")[0].openDevTools();
```

On this page

- Desktop app troubleshooting
  - Where is configuration stored locally?
  - How do I access logs?
  - How do I download app diagnostics?
  - Desktop App displays white screen while launching and doesn't load the page
  - "Installation has failed" dialog
  - "The application "Mattermost" can't be opened" dialog
  - Desktop App window is black and doesn't load the page
  - Desktop App window is white and doesn't load the page
  - Desktop App is not visible, but the Mattermost icon is in the Task Bar
  - Desktop App constantly refreshes the page
  - Desktop App constantly asks to log in to Mattermost server
  - "Internal error: BrowserWindow 'unresponsive' event has been emitted"
  - Desktop app not responsive within Citrix Virtual Apps or Desktop Environment
  - Can I uninstall the desktop app I installed using snap on Linux?
    - Report Desktop App issues

Distribute a custom desktop app

You can customize and distribute your own Mattermost desktop application by configuring [src/common/config/buildConfig.ts (https://github.com/mattermost/desktop/blob/master/src/common/config/buildConfig.ts)](https://github.com/mattermost/desktop/blob/master/src/common/config/buildConfig.ts).

1. Configure the desktop app's `buildConfig.ts` file. You can configure the following parameters to customize the user experience, including defaultTeams, helpLink, and enableServerManagement.
2. Follow the [Mattermost Desktop Development Guide (https://developers.mattermost.com/contribute/more-info/desktop/developer-setup/)](https://developers.mattermost.com/contribute/more-info/desktop/developer-setup/) to build the application.
3. Distribute the application to your users.

`defaultTeams`

List of server URLs and their display names added to the desktop app by default, which the user cannot modify. Users can still add servers through the Server Management page unless `enableServerManagement` is set to `false`. Expects an array of key-value pairs.

Example:

```
defaultTeams: [ { name: 'example', url: 'https://example.com' }, { name:
'mattermost', url: 'https://www.mattermost.com' } ]
```

`helpLink`

The URL of the help documentation in **Help > Learn More** menu bar item. If none is specified, the menu option is hidden. Expects a string.

Example:

```
helpLink: 'https://docs.mattermost.com/messaging/managing-desktop-app-servers.html' helpLink: ''
```

`enableServerManagement`

Controls whether users can add, edit, or remove servers on the app settings page. If set to false, at least one server must be specified for `defaultTeams` or else users cannot interact with any servers. Expects a boolean, true or false.

Example:

```
enableServerManagement: true
```

Managed resources

Custom builds of the Mattermost desktop app support managed resources which are services available on the same hostname and protocol as the Mattermost server.

To configure managed resources, add their path to the `managedResources` field in your configuration file. Selecting a managed resource opens it as a pop-up window in the desktop app.

Additionally, you must configure the :ref:Managed Resource Paths <administration-guide/configure/environment-configuration-settings:managed resource paths>` server configuration setting. For example, adding the `/video` path:

```
[...] managedResources: ['trusted', 'video'], [...]
```

Below are examples of server URLs with valid and invalid managed resource URLs:

Server: `https://mattermost.my.org`

- Valid: `https://mattermost.my.org/video`
- Valid: `https://mattermost.my.org/conference`

- Invalid: `http://mattermost.my.org/video` (different protocol)
- Invalid: `https://conference.my.org` (different origin)

Server: `https://my.org/mattermost`

- Valid: `https://my.org/video`
- Valid: `https://my.org/conference`
- Invalid: `http://my.org/video` (different protocol)
- Invalid: `https://conference.my.org` (different origin)

On this page

- Distribute a custom desktop app
  - defaultTeams
  - helpLink
  - enableServerManagement
  - Managed resources

Install desktop app on Linux

This page describes how to install the Mattermost desktop app on Linux.

Ubuntu/Debian

Both a `.deb` package (Beta), and an official APT repository is available for Debian 9 and for Ubuntu releases 20.04 LTS or later. Automatic app updates are supported and enabled. When a new version of the desktop app is released, your app updates automatically.

Important

The GPG public key has changed. If you had previously set up the repository on your system, you'll need to download the new key. You can set the `UPDATE_GPG_KEY=yes` environment variable when running the setup script to configure it to overwrite the previous key on your system with the new one. The first step of installation then becomes: `curl -fsS -o- https://deb.packages.mattermost.com/setup-repo.sh | sudo UPDATE_GPG_KEY=yes bash`. Depending on your setup, additional steps may also be required, particularly for installations that don't rely on the repository setup script.

1. At the command line, set up the Mattermost repository on your system:

```
curl -fsS -o- https://deb.packages.mattermost.com/setup-repo.sh | sudo
bash
```

2. Install the Mattermost desktop app:

```
sudo apt install mattermost-desktop
```

3. Update the Mattermost desktop app:

```
sudo apt upgrade mattermost-desktop
```

Snapcraft Package

A snap is available for systems that have Snapcraft installed. Snapcraft is installed by default on Ubuntu 16.04 and later, but for most other Linux distributions you can install it manually. To install Snapcraft, see [Install snapd (https://snapcraft.io/docs/installing-snapd)](https://snapcraft.io/docs/installing-snapd) on the Snapcraft website for details.

1. At the command line, execute the following command:

```
sudo snap install mattermost-desktop --beta
```

2. Run Mattermost as a desktop app.

> Tip
>
> You can review the current version of your desktop app by selecting the **More**
>
> ⋮
>
> icon located in the top left corner of the desktop app, then selecting **Help > Version...**.

CentOS/RHEL

Beta `.rpm` packages are available for CentOS and RHEL 7 and 8. Automatic app updates aren't supported. You must update your app manually.

Install the Mattermost desktop app

1. Download the latest version of the Mattermost desktop app for 64-bit systems: [mattermost-desktop-5.12.1-linux-x86_64.rpm (https://releases.mattermost.com/desktop/5.12.1/mattermost-desktop-5.12.1-linux-x86_64.rpm)](https://releases.mattermost.com/desktop/5.12.1/mattermost-desktop-5.12.1-linux-x86_64.rpm)

2. At the command line, execute the following command:

```
sudo rpm -i mattermost-desktop-5.12.1-linux-x86_64.rpm
```

3. Run Mattermost as a desktop app.

To manually update the desktop app, run the following command:

```
sudo rpm -u mattermost-desktop-5.12.1-linux-x86_64.rpm
```

> Tip
>
> You can review the current version of your desktop app by selecting the **More**
>
> ⋮
>
> icon located in the top left corner of the desktop app, then selecting **Help > Version...**.

Generic Linux

The Desktop app is available in two formats which are usable on most Linux distributions: a compressed tarball, and an AppImage binary. Both can be downloaded from the Desktop App's Github releases page (https://github.com/mattermost/desktop/releases). Automatic app updates are supported and enabled on AppImage binary builds. When a new version of the desktop app is released, your app updates automatically.

For instructions on how to use the AppImage binary, please refer to the AppImage Quickstart documentation page.

Install the Desktop App's compressed tarball

1. Download the latest version of the Mattermost desktop app for 64-bit systems: mattermost-desktop-5.12.1-linux-x64.tar.gz (https://releases.mattermost.com/desktop/5.12.1/mattermost-desktop-5.12.1-linux-x64.tar.gz)

2. Extract the archive to a convenient location, then give `chrome-sandbox` in the extracted directory the required ownership and permissions: `sudo chown root:root chrome-sandbox && sudo chmod 4755 chrome-sandbox`
3. Execute `mattermost-desktop` located inside the extracted directory.
4. To create a Desktop launcher, open the file `README.md`, and follow the instructions in the **Desktop launcher** section.

On this page

- Install desktop app on Linux
- Install desktop app on Linux
  - Ubuntu/Debian
    - Snapcraft Package
  - CentOS/RHEL
    - Install the Mattermost desktop app
  - Generic Linux
    - Install the Desktop App's compressed tarball

Silent Windows desktop distribution

You can distribute the official Windows desktop app silently to end users, pre-configured with the server URL. Additionally, you can customize all of the desktop app settings, except the **Start app on login** option.

> Tip
>
> Want to perform a silent installation of the desktop app MSI instead?

1. Download the latest Windows installer from the Mattermost download page.
2. Move the executable file into a shared place such as a file server.
3. To create a batch file in Windows:

> - *Open a text editor of your choice, such as Notepad or Notepad++.*
> - *Copy and paste the following commands in the text file:*
>
> ```
> rem "Step 1: Install Mattermost desktop app silently into user's local disk" start \\SERVER\shared_folder\mattermost-setup-4.6.2-win.exe --silent if not exist "%APPDATA%\Mattermost" mkdir %APPDATA%\Mattermost rem "Step 2: Generate initial config.json into user's config directory" ( echo { echo "version": 2, echo "teams": [ echo { echo "name": "core", echo "url": "https://community.mattermost.com", echo "order": 0 echo } echo ], echo "showTrayIcon": true, echo "trayIconTheme": "light", echo "minimizeToTray": true, echo "notifications": { echo "flashWindow": 2, echo "bounceIcon": true, echo "bounceIconType": "informational" echo }, echo "showUnreadBadge": true, echo "useSpellChecker": true, echo "enableHardwareAcceleration": true, echo "autostart": true, echo "spellCheckerLocale": "en-US", echo "darkMode": false echo } ) > %APPDATA%\Mattermost\config.json
> ```

4. Save the text file with the extension `.bat`. For instance, `mattermost-app-install.bat`.

5. Use standard software asset management tools to distribute and deploy the batch file to each user.

Once run, the desktop app is added to the user's local directory, along with the pre-configured `config.json` file. The installer creates a shortcut for the desktop app in the user's start menu; if a zip version is used, you need to create the shortcut manually.

> Tip
>
> You can copy the executable to the folder before running it instead of using this command to install the desktop app into a shared folder. This allows the shared folder to only require read-only permissions.

Windows App: Silently removing the app

To remove the app silently from a user's computer, you can run the following command with the .exe file closed:

```
%userprofile%\AppData\local\Programs\mattermost-desktop\Uninstall
Mattermost.exe /currentuser /S
```

On this page

- Silent Windows desktop distribution
  - Windows App: Silently removing the app

Encryption options

Mattermost provides encryption-in-transit and encryption-at-rest capabilities. This page guides you through setting up appropriate encryption security.

Encryption is not required for GDPR, although it can be used as an additional safeguard against data breach.

Encryption-in-transit

Mattermost supports TLS encryption including AES-256 with 2048-bit RSA on all data transmissions between Mattermost client applications and the Mattermost server. You may either set up TLS on the Mattermost Server or install a proxy such as NGINX and set up TLS on the proxy. Refer to our configuration guide for more details.

Connections to Active Directory/LDAP can optionally be secured with TLS or stunnel.

Connections to calls are secured with a combination of:

> - *TLS: The existing WebSocket channel is used to secure the signaling path.*
> - *DTLS v1.2 (mandatory): Used for initial key exchange. Supports* `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256` *and* `TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA` *algorithms.*
> - *SRTP (mandatory): Used to encrypt all media packets (i.e. those containing voice or screen share). Supports* `AEAD_AES_128_GCM` *and* `AES128_CM_HMAC_SHA1_80` *algorithms.*

Gossip encryption (experimental)

In a High Availability mode, Mattermost supports encryption of cluster data in-transit when using the gossip protocol, which is based on principles outlined in the [SWIM protocol developed by researchers at Cornell University (https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf)](https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf). The gossip

protocol is a communication mechanism in distributed systems where nodes randomly exchange information to ensure data consistency across the network. It is decentralized, scalable, and fault-tolerant, making it ideal for systems with numerous nodes. Information is spread in a manner similar to social gossip, with nodes periodically "gossiping" updates to random peers until the network converges to a consistent state. Widely used in distributed databases, blockchain networks, and peer-to-peer systems, the protocol is simple to implement and resilient to node failures. However, it can suffer from redundancy and propagation delays in large networks.

The encryption uses AES-256 by default, and it is not configurable. However, it is possible to manually set the value in the `Systems` table for the `ClusterEncryptionKey` row. A key is a byte array converted to base64. It can be set to a length of 16, 24, or 32 bytes to select AES-128, AES-192, or AES-256 respectively.

Encryption-at-rest

Encryption-at-rest is available for messages via hardware and software disk encryption solutions applied to the Mattermost database, which resides on its own server within your infrastructure. To enable end user search and compliance reporting of message histories, Mattermost does not offer encryption within the database.

Both the database and file storage can be secured using software or hardware disk encryption solutions. Full-disk encryption methods such as LUKS (Linux), BitLocker (Windows), or database-specific solutions like Transparent Data Encryption (TDE) can be employed. Integration with external encrypted storage systems is supported.

Additionally, encryption-at-rest is available for files stored via hardware and software disk encryption solutions applied to the server used for local storage or storage via MinIO.

Database

Encryption-at-rest is available for messages via hardware and software disk encryption solutions applied to the Mattermost database, which resides on its own server within your infrastructure. See the PostgreSQL (https://

www.postgresql.org/docs/10/encryption-options.html) database documentation for details on encryption options at the disk level.

File storage

For local storage or storage via Minio, encryption-at-rest is available for files stored via hardware and software disk encryption solutions applied to the server.

For Amazon's proprietary S3 system, encryption-at-rest is available via server-side encryption with Amazon S3-managed keys in Mattermost enterprise-badge.

SAML encryption support

Mattermost supports the following encryption methods for SAML:

- aes128-gcm
- aes192-gcm
- aes256-gcm
- aes128-cbc
- aes256-cbc

On this page

- Encryption options
  - Encryption-in-transit
  - Gossip encryption (experimental)
  - Encryption-at-rest
    - Database
    - File storage
  - SAML encryption support

Manually migrate from MySQL to PostgreSQL

Migrating a MySQL database to PostgreSQL manually involves the following steps:

- install recommended tools
- familiarize yourself with system requirements and configurations
- ensure migration prerequisites are met
- review schema differences
- prepare a target database
- migrate the data
- complete post-migration wrap-up activities

> Tip
>
> See the plugin migrations section for details on migrating collaborative playbooks and Boards.

Not sure upgrading manually is the right path forward? Mattermost customers looking for tailored guidance based on their Mattermost deployment can contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/).

Tool recommendations

If you prefer to migrate to Postgres manually, we recommend the following tools for the migration process:

- pgloader
- morph

- dbcmp (optional)

This page includes instructions on how to install each of these tools, and then proceed with the database migration.

Once you've installed the necessary tools, review the system requirements and configurations documentation, and know what's required before starting your migration to prepare for your migration.

Start your migration by preparing your target database, then migrate the data, and complete all post-migration steps.

See the plugin migrations documentation for details on migrating collaborative playbooks and boards.

pgloader

Use the `pgloader` tool to migrate your data from MySQL to PostgreSQL.

Install pgloader

To install `pgloader`, see the official installation guide (https://pgloader.readthedocs.io/en/latest/install.html).

> Note
>
> If you are using MySQL v8: Due to a known bug (https://github.com/dimitri/pgloader/issues/1183) in pgloader-compiled binaries, you need to compile pgloader from the source. Please follow the steps here (https://pgloader.readthedocs.io/en/latest/install.html#build-from-sources) to build from the source.

Alternatively, you may want to use our `mattermost-pgloader` Docker image to avoid installing or building `pgloader`. See the documentation below for details.

Use pgloader

Pull the Docker image and verify pgloader

For a manual migration, run the following command to pull the `mattermost-pgloader` image and verify that pgloader is working correctly:

```
docker run -it --rm -v $(pwd):/home/migration mattermost/mattermost-pgloader:latest pgloader --version
```

This command pulls the `mattermost/mattermost-pgloader:latest` image and runs `pgloader` to check its version and ensure it works as expected.

Map local directory

Use the `-v $(pwd):/home/migration` flag to map your current working directory to the Docker container. This allows you to use your local directory for storing logs and other files.

Set network configuration

Depending on your network requirements, set the `--network` flag accordingly. For example, to access localhost, you need to set the network to `host`.

morph

The `morph` tool creates the PostgreSQL schema.

> Note
>
> Both `morph` and `dbcmp` requires Go toolchain. To install Go compiler, follow the Go documentation.

Install morph

You can install morph CLI by running the following command:

```
go install github.com/mattermost/morph/cmd/morph@v1
```

dbcmp (Optional)

The `dbcmp` tool enables you to compare the data following the migration by comparing every table and reporting whether there is a diversion between two schemas.

Install dbcmp

You can install [dbcmp (https://github.com/mattermost/dbcmp)](https://github.com/mattermost/dbcmp) by running the following command:

```
go install github.com/mattermost/dbcmp/cmd/dbcmp@latest
```

System requirements and configurations for manual migrations

Before starting a manual migration process, it's essential to ensure that your system meets the necessary requirements for a smooth and efficient migration. We strongly recommend the following system specifications and adjustments:

- Ensure you have enough system memory resources. 16GB of RAM is recommended as a default. In scenarios where system memory is insufficient, users can fine-tune pgloader settings, such as the `number of workers`, `prefetch rows`, and especially `rows per range` if `concurrency` is set above `1`. These adjustments can help optimize resource utilization based on available system resources. For further detail see pgloader documentation.
- A multi-core processor with sufficient processing power is recommended for the migration process, especially when dealing with large datasets.
- Ensure that there is enough disk space available for storing both the MySQL and PostgreSQL databases, as well as any temporary files generated during the migration process. The amount of required disk space depends on the size of the databases being migrated.

- To reduce migration time further, users may choose to manually drop indexes on the target PostgreSQL database before initiating the migration process. This approach can potentially accelerate the migration by reducing overhead with index builds during data insertion.

Before a manual migration

Important

This guide requires a schema of v7.1 ESR or later. So, if you have an earlier version and planning to migrate, please update your Mattermost Server to v7.1 at a minimum. See the extended support releases documentation for details.

- Back up your MySQL data.
- Confirm your Mattermost version. See the **About** modal for details.
- Schedule the migration window. This process requires you to stop the Mattermost Server during the migration.
- See the schema-diffs section to ensure data compatibility between schemas.
- Prepare your PostgreSQL environment by creating a database and user. See the database documentation for details.
- On newer versions (https://www.postgresql.org/docs/release/15.0/) of PostgreSQL, newly created users do not have access to `public` schema. The access should be explicitly granted by running `GRANT ALL ON SCHEMA public to mmuser`.

Schema diffs

Before the manual migration, due to differences between the two schemas, some manual steps may be required for an error-free migration.

Text to character varying

We encourage you to check if the sizes are consistent within the PostgreSQL schema limits since the Mattermost MySQL schema uses the `text` column type

in the various tables instead of `varchar` representation in the PostgreSQL schema.

You can check if there are any required deletions or updates. For example, to do so in the `Audits` table/ `Action` column; run:

```
SELECT FROM mattermost.Audits where LENGTH(Action) > 512;
```

The following table shows the deletions or updates you can proceed with that don’t incur further consequences.

Table	Column	Data type casting	Consequence on deletion
Audits	Action	text -> varchar(512)	No side effect on how the application works (The affected row needs to be deleted).
Audits	ExtraInfo	text -> varchar(1024)	No side effect on how the application works (The affected row needs to be deleted).
ClusterDiscovery	HostName	text -> varchar(512)	No side effect on how the application works (The affected row needs to be deleted).
Commands	IconURL	text -> varchar(1024)	The field can be deleted or updated with a new URL.
Commands	AutoCompleteDesc	text -> varchar(1024)	The field can be deleted or rewritten.
Commands	AutoCompleteHint	text -> varchar(1024)	The field can be deleted or rewritten.
RemoteClusters	Topics	text -> varchar(512)	The field can be removed.
Systems	Value	text -> varchar(1024)	Edge case, ideally should never happen.

The following table shows several occurrences where the schema can differ and data size constraints within the PostgreSQL schema can result in errors. Each table/row requires individual inspection hence we added the possible consequence of deletion.

> Tip
>
> Several reports have been received from our community that the `LinkMetadata` and `FileInfo` tables had some overflows, so we recommend checking these tables in particular. Please check if your data in the MySQL schema exceeds these limitations.

Table	Column	Data type casting	Consequence on deletion
Compliances	Keywords	text -> varchar(512)	The filter for compliance needs to be updated.
Compliances	Emails	text -> varchar(1024)	The filter for compliance needs to be updated.
FileInfo	Path	text -> varchar(512)	Previous links to this file won't work (The affected row needs to be deleted).
FileInfo	ThumbnailPath	text -> varchar(512)	Previous links to this file won't work (The affected row needs to be deleted).
FileInfo	PreviewPath	text -> varchar(512)	Previous links to this file won't work (The affected row needs to be deleted).
FileInfo	Name	text -> varchar(256)	Previous links to this file won't work (The affected row needs to be deleted).
FileInfo	MimeType	text -> varchar(256)	Previous links to this file won't work (The affected row needs to be deleted).
LinkMetadata	URL	text -> varchar(2048)	Previous links to this file won't work (The affected row needs to be deleted).

Table	Column	Data type casting	Consequence on deletion
RemoteClusters	SiteURL	text -> varchar(512)	Previous remote cluster will be removed (The affected row needs to be deleted).
Sessions	DeviceId	text -> varchar(512)	Users will be logged out on these devices (The affected row needs to be deleted).
UploadSessions	FileName	text -> varchar(256)	The upload session will be lost (The affected row needs to be deleted).
UploadSessions	Path	text -> varchar(512)	The upload session will be lost (The affected row needs to be deleted).

Full-text indexes

It's possible that some words in the `Posts` and `FileInfo` tables can exceed the limits of the maximum token length (https://www.postgresql.org/docs/11/textsearch-limitations.html) for full-text search indexing. In these cases, we are dropping the `idx_posts_message_txt` and `idx_fileinfo_content_txt` indexes from the PostgreSQL schema, and creating these indexes after the migration by running the following queries:

To prevent errors during the migration, we have included following queries:

```
DROP INDEX IF EXISTS {{ .source_db }}.idx_posts_message_txt; DROP INDEX IF
EXISTS {{ .source_db }}.idx_fileinfo_content_txt;
```

Unsupported unicode sequences

There is a specific unicode sequence that is [disallowed (https://www.postgresql.org/docs/16/datatype-json.html#DATATYPE-JSON)](https://www.postgresql.org/docs/16/datatype-json.html#DATATYPE-JSON) in PostgreSQL which is `\u0000`. There is a chance that this sequence may appear in several rows across a bunch of tables in your MySQL database. If it is the case, during the migration you will likely receive an error as following: `unsupported Unicode escape sequence: \u0000 cannot be converted to text.`. To prevent this from happening, we advise to sanitize your data before starting to the migration. You can use the following query to replace `\u0000` sequence with empty string.

> Note
>
> You can use this query as-is in a script, or you may need to set the delimiter to something else (e.g., `DELIMITER //`) when defining it in the MySQL console. Once you are done defining the procedure, please set the delimiter back to the original (i.e., `DELIMITER ;`).

```
CREATE PROCEDURE SanitizeUnsupportedUnicode() BEGIN DECLARE done INT DEFAULT
FALSE; DECLARE curTableName text; DECLARE curColumnName text; DECLARE
cursors CURSOR FOR SELECT table_name, column_name FROM
information_schema.COLUMNS WHERE data_type = 'json' AND table_schema =
DATABASE(); DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE; OPEN
cursors; WHILE NOT DONE DO FETCH cursors INTO curTableName, curColumnName;
SET @query_string = CONCAT('UPDATE ', curTableName, ' SET ', curColumnName,
' = REPLACE(', curColumnName, ', \'\\\\u0000\', \'\') WHERE ',
curColumnName, ' LIKE \'%\\u0000%\';'); PREPARE dynamic_query FROM
@query_string; EXECUTE dynamic_query; DEALLOCATE PREPARE dynamic_query; END
WHILE; CLOSE cursors; END; CALL SanitizeUnsupportedUnicode(); DROP PROCEDURE
IF EXISTS SanitizeUnsupportedUnicode;
```

> Note
>
> There is also a specific byte sequence value that is not allowed and will cause an `invalid byte sequence for encoding 'UTF8': 0x00"` error during the migration. To prevent this error, you can add the `remove-null-characters` clause to the text casting rules. However, since pgloader will modify the data on the fly, there may be differences between the tables (if any are affected) during the comparison phase.

Artifacts may remain from previous configurations/versions

Prior to `v6.4`, Mattermost was using [golang-migrate (https://github.com/golang-migrate/migrate)](https://github.com/golang-migrate/migrate) to handle the schema migrations. Since we don't use it anymore, we exclude the table `schema_migrations`. If you were using Mattermost before `v6.4` consider dropping this table and excluding it from comparison as well.

```
DROP TABLE mattermost.schema_migrations;
```

Some community members have reported that they had `description` and `nextsyncat` columns in their `SharedChannelRemotes` table. These columns should be removed from the table. Consider running the following DDL to drop the columns. (This migration will be added to future versions of Mattermost).

```
ALTER TABLE SharedChannelRemotes DROP COLUMN description, DROP COLUMN
nextsyncat;
```

An error has been identified in the 96th migration that was previously released. Before proceeding with the migration, it is necessary to remove a specific column. To ensure the Threads table reaches the expected state, execute the following prepared statement:

```
SET @preparedStatement = (SELECT IF(  (  SELECT COUNT(*) FROM
INFORMATION_SCHEMA.COLUMNS  WHERE table_name = 'Threads'  AND table_schema
= DATABASE()  AND column_name = 'TeamId'  ) > 0,  'ALTER TABLE Threads DROP
COLUMN TeamId;',  'SELECT 1' )); PREPARE alterIfExists FROM
@preparedStatement; EXECUTE alterIfExists; DEALLOCATE PREPARE
alterIfExists;
```

Configuration in database

If you were previously utilizing a database for handling the Mattermost configuration, those tables will not be migrated from your MySQL database with the migration script.

Two migrations are necessary:

- migrate database configuration to the file system
- migrate file system configuration back to the database

Migrate database configuration to the file system

Use the `mmctl config migrate` command to migrate your config to the file system, as follows:

```
mmctl config migrate "<DB_USER>:<DB_PASS>@tcp(<DB_HOST>:3306)/<DB_NAME>?
charset=utf8mb4,utf8&readTimeout=30s&writeTimeout=30s&multiStatements=true" /
opt/mattermost/config/config.json --local
```

Where `<DB_USER>`, `<DB_PASS>`, `<DB_HOST>`, and `<DB_NAME>` are replaced with your environment values. Ensure you use `--local` when running this command. The first parameters (`<DB_USER>`, `<DB_PASS>`) is the database the configuration is stored in, the second parameter (`<DB_HOST>`, `<DB_NAME>`) is the file we are saving the configuration to.

In the configuration file, update the `SqlSettings.DataSource` and `SqlSettings.DriverName` fields to reflect new changes. To do so, open the `json` file and change the respective fields.

Migrate file system configuration back to the database

To save configuration back to the database, Use the `mmctl config migrate` command again and reverse the parameters. Ensure you use the new database credentials moving it back to the target database.

```
SELECT * FROM Configurations WHERE Active = 't';
```

You should update the `SqlSettings.DataSource` and `SqlSettings.DriverName` fields accordingly. Also, note that the `MM_CONFIG` environment variable should point to the new DSN after the migration is completed.

Prepare target database

It is essential to create tables and indexes to ensure that the PostgreSQL database schema is properly structured according to the required specifications. Since Mattermost repository contains all of the required SQL queries to achieve that, we can leverage this by running the following steps:

- Clone the `mattermost` repository for your specific version:

```
git clone -b <your current version (eg. release-7.8)>
git@github.com:mattermost/mattermost.git --depth=1
```

- Run all schema migrations* on your PostgreSQL database using morph CLI with the following command:

```
morph apply up --driver postgres --dsn "postgres://user:pass@localhost:5432/
<target_db_mame>?sslmode=disable" --path ./mattermost/db/migrations/
postgres --number -1
```

* After `v8`, due to project re-organization, the migrations directory has been changed to `./mattermost/server/channels/db/migrations/postgres/` relative to where you cloned Mattermost repository. Please set `--path` flag accordingly.

Migrate the data

Once we set the schema to a desired state, we can start migrating the **data** by running `pgloader`

> Note
>
> In the example below, the hosts for both databases are assumed to be in the same instance. Please update addresses accordingly if they are on different machines. Also you can test the `.load` file by simply running `pgloader` with `--dry-run` flag. For instance `pgloader --dry-run migration.load` command.

Use the following configuration for the baseline of the data migration:

```
LOAD DATABASE FROM mysql://{{ .mysql_user }}:{{ .mysql_password }}
@{{ .mysql_address }}/{{ .source_db }} INTO pgsql://{{ .pg_user }}:
{{ .pg_password }}@{{ .postgres_address }}/{{ .target_db }} WITH data only,
workers = 8, concurrency = 1, multiple readers per thread, rows per range =
10000, prefetch rows = 10000, batch rows = 2500, create no tables, create no
indexes, preserve index names SET PostgreSQL PARAMETERS maintenance_work_mem
to '128MB', work_mem to '12MB' SET MySQL PARAMETERS net_read_timeout =
'120', net_write_timeout = '120' CAST column Channels.Type to "channel_type"
drop typemod, column Teams.Type to "team_type" drop typemod, column
UploadSessions.Type to "upload_session_type" drop typemod, column
ChannelBookmarks.Type to "channel_bookmark_type" drop typemod, column
Drafts.Priority to text, type int when (= precision 11) to integer drop
typemod, type bigint when (= precision 20) to bigint drop typemod, type text
to varchar drop typemod using remove-null-characters, type tinyint when (<=
precision 4) to boolean using tinyint-to-boolean, type json to jsonb drop
typemod using remove-null-characters EXCLUDING TABLE NAMES MATCHING ~<IR_>,
~<focalboard>, 'schema_migrations', 'db_migrations', 'db_lock',
'Configurations', 'ConfigurationFiles', 'db_config_migrations', 'calls'
BEFORE LOAD DO $$ ALTER SCHEMA public RENAME TO {{ .source_db }}; $$, $$
TRUNCATE TABLE {{ .source_db }}.systems; $$, $$ DROP INDEX IF EXISTS
{{ .source_db }}.idx_posts_message_txt; $$, $$ DROP INDEX IF EXISTS
{{ .source_db }}.idx_fileinfo_content_txt; $$ AFTER LOAD DO $$ UPDATE
{{ .source_db }}.db_migrations set name='add_createat_to_teamembers' where
version=92; $$, $$ ALTER SCHEMA {{ .source_db }} RENAME TO public; $$, $$
SELECT pg_catalog.set_config('search_path', '"$user", public', false); $$, $
$ ALTER USER {{ .pg_user }} SET SEARCH_PATH TO 'public'; $$;
```

Once you save this configuration file, e.g. `migration.load`, you can run the `pgloader` with the following command:

```
pgloader migration.load > migration.log
```

Feel free to contribute to and/or report your findings through your migration to us.

After the migration

Restore full-text indexes

To avoid performance regression on `Posts` and `FileInfo` table access, following queries should be executed once the migration finishes:

```
CREATE INDEX IF NOT EXISTS idx_posts_message_txt ON public.posts USING
gin(to_tsvector('english', message)); CREATE INDEX IF NOT EXISTS
idx_fileinfo_content_txt ON public.fileinfo USING
gin(to_tsvector('english', content));
```

> Note
>
> If any of the entries in your `Posts` and `FileInfo` tables exceed the limit mentioned above, index creation query will warn with the `ERROR: string is too long for tsvector` log while trying to create these indexes. This means the content that didn't fit into a `tsvector` was ignored. If you still want to index the truncated content, you can use `substring()` function on the content while creating the indexes.
>
> Start with a substring length of 1000000 and gradually decrease if the error persists:

```
-- Start with 1000000 CREATE INDEX IF NOT EXISTS idx_fileinfo_content_txt
ON public.fileinfo USING gin(to_tsvector('english',
substring(content,0,1000000))); -- If that fails, try 800000 CREATE INDEX
IF NOT EXISTS idx_fileinfo_content_txt ON public.fileinfo USING
gin(to_tsvector('english', substring(content,0,800000))); -- Continue
decreasing as needed (e.g., 600000, 500000) until the index creates
successfully
```

Compare the data

We developed an internal tool called `dbcmp` to simplify database comparison. It checks every table in two databases and reports any differences in their schemas. However, `dbcmp` does not compare individual rows. Instead, it calculates checksum values based on a specified page-size and compares them. This means it cannot generate row-level diffs.

We recommend using `dbcmp` as an additional check to verify data integrity, especially if custom casting rules (beyond the defaults or those provided by us) were used during migration. Otherwise, running this tool is not necessary.

The tool includes a few flags to run a comparison:

```
Usage:  dbcmp [flags] Flags:  --exclude strings exclude tables from
comparison, takes comma-separated values.  --include strings include only
matching tables for comparison, takes comma-separated values.  -h, --help
help for dbcmp  --source string source database dsn  --target string target
database dsn  -v, --version version for dbcmp
```

> Note
>
> `--exclude` and `--include` flags are mutually exclusive and they can't be used together.

For our case, we can simply run the following command:

```
dbcmp --source "${MYSQL_DSN}" --target "${POSTGRES_DSN} " --
include="posts,users"
```

An example command would look like: `dbcmp --source "user:password@tcp(address:3306)/db_name --target "postgres://user:password@address:5432/db_name`

> Note
>
> `POSTGRES_DSN` should start with a `postgres://` prefix. This way `dbcmp` decides which driver to use while connecting to a database.

Another exclusion we are making is in the `db_migrations` table which has a small difference (a typo in a single migration name) and creates a diff. Since we created the PostgreSQL schema with morph, and the official `mattermost` source, we can skip it safely without concerns. On the other hand, `systems` table may contain additional diffs if there were extra keys added during some of the migrations. Consider excluding the `systems` table if you run into issues, and perform a manual comparison as the data in the `systems` table is relatively smaller in size.

> Note
>
> If the `remove-null-characters` transform function is utilized during the migration and there were `0x00` byte sequences in the MySQL database, those tables will have differences during the comparison phase.

Restore the search path

If you closely examine the `pgloader` configuration file (e.g., `migration.load`), you will notice that the `search_path` of the database user is set to `public`. This is the only requirement for the Mattermost application. However, if you need to include other schemas in the search path, you should modify the `search_path` accordingly to meet your specific requirements.

Plugin migrations

On the plugin side, we are going to take a different approach from what we have done above. We are not going to use `morph` tool to create tables and indexes this time. We are going to utilize `pgloader` to create the tables on behalf of us. The reason for doing so is that collaborative playbooks and boards leverage application logic to facilitate SQL queries. But we don't want to use any level of application at this point.

Collaborative playbooks

The `pgloader` configuration provided for playbooks is based on `v1.38.1` and the plugin should be at least `v1.36.0` to perform the migration.

Once we are ready to migrate, we can start migrating the **schema** and the **data** by running `pgloader`

Use the following configuration for the baseline of the data migration:

```
LOAD DATABASE FROM mysql://{{ .mysql_user }}:{{ .mysql_password }}
@{{ .mysql_address }}/{{ .source_db }} INTO pgsql://{{ .pg_user }}:
{{ .pg_password }}@{{ .postgres_address }}/{{ .target_db }} WITH include
drop, create tables, create indexes, no foreign keys, workers = 8,
concurrency = 1, multiple readers per thread, rows per range = 50000,
preserve index names SET PostgreSQL PARAMETERS maintenance_work_mem to
'128MB', work_mem to '12MB' SET MySQL PARAMETERS net_read_timeout = '120',
net_write_timeout = '120' CAST column IR_ChannelAction.ActionType to text
drop typemod, column IR_ChannelAction.TriggerType to text drop typemod,
column IR_Incident.ChecklistsJSON to "json" drop typemod INCLUDING ONLY
TABLE NAMES MATCHING ~/IR_/ BEFORE LOAD DO $$ ALTER SCHEMA public RENAME TO
{{ .source_db }}; $$ AFTER LOAD DO $$ ALTER TABLE
{{ .source_db }}.IR_ChannelAction ALTER COLUMN ActionType TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_ChannelAction ALTER
COLUMN TriggerType TYPE varchar(65536); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN ReminderMessageTemplate TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
ReminderMessageTemplate SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN ConcatenatedInvitedUserIDs TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
ConcatenatedInvitedUserIDs SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN ConcatenatedWebhookOnCreationURLs
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER
COLUMN ConcatenatedWebhookOnCreationURLs SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Incident ALTER COLUMN ConcatenatedInvitedGroupIDs
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER
COLUMN ConcatenatedInvitedGroupIDs SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN Retrospective TYPE varchar(65536);
$$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN Retrospective
SET DEFAULT ''::text; $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER
COLUMN MessageOnJoin TYPE varchar(65536); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN MessageOnJoin SET DEFAULT
''::text; $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
ConcatenatedWebhookOnStatusUpdateURLs TYPE varchar(65536); $$, $$ ALTER
TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
ConcatenatedWebhookOnStatusUpdateURLs SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Incident ALTER COLUMN CategoryName TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
CategoryName SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Incident ALTER COLUMN ConcatenatedBroadcastChannelIds
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER
COLUMN ConcatenatedBroadcastChannelIds SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Incident ALTER COLUMN ChannelIDToRootID TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Incident ALTER COLUMN
ChannelIDToRootID SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ReminderMessageTemplate TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
```

```
ReminderMessageTemplate SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ConcatenatedInvitedUserIDs TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
ConcatenatedInvitedUserIDs SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ConcatenatedWebhookOnCreationURLs
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER
COLUMN ConcatenatedWebhookOnCreationURLs SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN ConcatenatedInvitedGroupIDs
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER
COLUMN ConcatenatedInvitedGroupIDs SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN MessageOnJoin TYPE varchar(65536);
$$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN MessageOnJoin
SET DEFAULT ''::text; $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER
COLUMN RetrospectiveTemplate TYPE varchar(65536); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN RetrospectiveTemplate SET DEFAULT
''::text; $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
ConcatenatedWebhookOnStatusUpdateURLs TYPE varchar(65536); $$, $$ ALTER
TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
ConcatenatedWebhookOnStatusUpdateURLs SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
ConcatenatedSignalAnyKeywords TYPE varchar(65536); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ConcatenatedSignalAnyKeywords SET
DEFAULT ''::text; $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER
COLUMN CategoryName TYPE varchar(65536); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN CategoryName SET DEFAULT ''::text;
$$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN ChecklistsJSON
TYPE JSON USING ChecklistsJSON::JSON; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ConcatenatedBroadcastChannelIds
TYPE varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER
COLUMN ConcatenatedBroadcastChannelIds SET DEFAULT ''::text; $$, $$ ALTER
TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN RunSummaryTemplate TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
RunSummaryTemplate SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_Playbook ALTER COLUMN ChannelNameTemplate TYPE
varchar(65536); $$, $$ ALTER TABLE {{ .source_db }}.IR_Playbook ALTER COLUMN
ChannelNameTemplate SET DEFAULT ''::text; $$, $$ ALTER TABLE
{{ .source_db }}.IR_PlaybookMember ALTER COLUMN Roles TYPE varchar(65536); $
$, $$ ALTER TABLE {{ .source_db }}.IR_Category_Item ADD CONSTRAINT
ir_category_item_categoryid FOREIGN KEY (CategoryId) REFERENCES
{{ .source_db }}.IR_Category(Id); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Metric ADD CONSTRAINT ir_metric_metricconfigid FOREIGN
KEY (MetricConfigId) REFERENCES {{ .source_db }}.IR_MetricConfig(Id); $$, $$
ALTER TABLE {{ .source_db }}.IR_Metric ADD CONSTRAINT ir_metric_incidentid
FOREIGN KEY (IncidentId) REFERENCES {{ .source_db }}.IR_Incident(Id); $$, $$
ALTER TABLE {{ .source_db }}.IR_MetricConfig ADD CONSTRAINT
ir_metricconfig_playbookid FOREIGN KEY (PlaybookId) REFERENCES
{{ .source_db }}.IR_Playbook(Id); $$, $$ ALTER TABLE
{{ .source_db }}.IR_PlaybookAutoFollow ADD CONSTRAINT
```

```
ir_playbookautofollow_playbookid FOREIGN KEY (PlaybookId) REFERENCES
{{ .source_db }}.IR_Playbook(Id); $$, $$ ALTER TABLE
{{ .source_db }}.IR_PlaybookMember ADD CONSTRAINT
ir_playbookmember_playbookid FOREIGN KEY (PlaybookId) REFERENCES
{{ .source_db }}.IR_Playbook(Id); $$, $$ ALTER TABLE
{{ .source_db }}.IR_Run_Participants ADD CONSTRAINT
ir_run_participants_incidentid FOREIGN KEY (IncidentId) REFERENCES
{{ .source_db }}.IR_Incident(Id); $$, $$ ALTER TABLE
{{ .source_db }}.IR_StatusPosts ADD CONSTRAINT ir_statusposts_incidentid
FOREIGN KEY (IncidentId) REFERENCES {{ .source_db }}.IR_Incident(Id); $$, $$
ALTER TABLE {{ .source_db }}.IR_TimelineEvent ADD CONSTRAINT
ir_timelineevent_incidentid FOREIGN KEY (IncidentId) REFERENCES
{{ .source_db }}.IR_Incident(Id); $$, $$ CREATE UNIQUE INDEX IF NOT EXISTS
ir_playbookmember_playbookid_memberid_key on
{{ .source_db }}.IR_PlaybookMember(PlaybookId,MemberId); $$, $$ CREATE INDEX
IF NOT EXISTS ir_statusposts_incidentid_postid_key on
{{ .source_db }}.IR_StatusPosts(IncidentId,PostId); $$, $$ CREATE INDEX IF
NOT EXISTS ir_playbookmember_playbookid on
{{ .source_db }}.IR_PlaybookMember(PlaybookId); $$, $$ ALTER SCHEMA
{{ .source_db }} RENAME TO public; $$, $$ SELECT
pg_catalog.set_config('search_path', '"$user", public', false); $$, $$ ALTER
USER {{ .pg_user }} SET SEARCH_PATH TO 'public'; $$;
```

```
pgloader playbooks.load > playbooks_migration.log
```

Focalboard

As of `v9.0` Boards will transition to being fully community supported as the Focalboard plugin. Hence this guide covers only version `v7.10.x` of the schema. Official announcement.

Once we are ready to migrate, we can start migrating the **schema** and the **data** by running `pgloader`

Use the following configuration for the baseline of the data migration:

```
LOAD DATABASE FROM mysql://{{ .mysql_user }}:{{ .mysql_password }}
@{{ .mysql_address }}/{{ .source_db }} INTO pgsql://{{ .pg_user }}:
{{ .pg_password }}@{{ .postgres_address }}/{{ .target_db }} WITH include
drop, create tables, create indexes, reset sequences, workers = 8,
concurrency = 1, multiple readers per thread, rows per range = 50000,
preserve index names SET PostgreSQL PARAMETERS maintenance_work_mem to
'128MB', work_mem to '12MB' SET MySQL PARAMETERS net_read_timeout = '120',
net_write_timeout = '120' CAST column focalboard_blocks.fields to "json"
drop typemod, column focalboard_blocks_history.fields to "json" drop
typemod, column focalboard_schema_migrations.name to "varchar" drop typemod,
column focalboard_sessions.props to "json" drop typemod, column
focalboard_teams.settings to "json" drop typemod, column
focalboard_users.props to "json" drop typemod, type int when (= precision
11) to int4 drop typemod, type json to jsonb drop typemod INCLUDING ONLY
TABLE NAMES MATCHING ~/focalboard/ BEFORE LOAD DO $$ ALTER SCHEMA public
RENAME TO {{ .source_db }}; $$ AFTER LOAD DO $$ UPDATE
{{ .source_db }}.focalboard_blocks SET "fields" = '{}'::json WHERE
"fields"::text = ''; $$, $$ UPDATE
{{ .source_db }}.focalboard_blocks_history SET "fields" = '{}'::json WHERE
"fields"::text = ''; $$, $$ UPDATE {{ .source_db }}.focalboard_sessions SET
"props" = '{}'::json WHERE "props"::text = ''; $$, $$ UPDATE
{{ .source_db }}.focalboard_teams SET "settings" = '{}'::json WHERE
"settings"::text = ''; $$, $$ UPDATE {{ .source_db }}.focalboard_users SET
"props" = '{}'::json WHERE "props"::text = ''; $$, $$ ALTER SCHEMA
{{ .source_db }} RENAME TO public; $$, $$ SELECT
pg_catalog.set_config('search_path', '"$user", public', false); $$, $$ ALTER
USER {{ .pg_user }} SET SEARCH_PATH TO 'public'; $$;
```

```
pgloader focalboard.load > focalboard_migration.log
```

Calls

If you are running a version of Mattermost that is greater than `v9.9` or the Calls plugin above `v0.27`, you can opt to migrate the data for the plugin. We are going to take a similar approach with Boards and Playbooks migration and let pgloader create the tables.

Once we are ready to migrate, we can start migrating the **schema** and the **data** by running `pgloader`

Use the following configuration for the baseline of the data migration:

```
LOAD DATABASE FROM mysql://{{ .mysql_user }}:{{ .mysql_password }}
@{{ .mysql_address }}/{{ .source_db }} INTO pgsql://{{ .pg_user }}:
{{ .pg_password }}@{{ .postgres_address }}/{{ .target_db }} WITH include
drop, create tables, create indexes, reset sequences, workers = 8,
concurrency = 1, multiple readers per thread, rows per range = 50000,
preserve index names SET PostgreSQL PARAMETERS maintenance_work_mem to
'128MB', work_mem to '12MB' SET MySQL PARAMETERS net_read_timeout = '120',
net_write_timeout = '120' CAST type json to jsonb drop typemod INCLUDING
ONLY TABLE NAMES MATCHING ~/calls/ BEFORE LOAD DO $$ ALTER SCHEMA public
RENAME TO {{ .source_db }}; $$ AFTER LOAD DO $$ ALTER SCHEMA
{{ .source_db }} RENAME TO public; $$, $$ SELECT
pg_catalog.set_config('search_path', '"$user", public', false); $$, $$ ALTER
USER {{ .pg_user }} SET SEARCH_PATH TO 'public'; $$;
```

```
pgloader calls.load > calls_migration.log
```

Troubleshooting

See troubleshooting errors during migration from MySQL to PostgreSQL

On this page

- Manually migrate from MySQL to PostgreSQL
  - Tool recommendations
    - pgloader
      - Install pgloader
      - Use pgloader
        - Pull the Docker image and verify pgloader
        - Map local directory
        - Set network configuration
    - morph
      - Install morph
    - dbcmp (Optional)
      - Install dbcmp
  - System requirements and configurations for manual migrations
    - Before a manual migration

    - Schema diffs
        - Text to character varying
        - Full-text indexes
        - Unsupported unicode sequences
        - Artifacts may remain from previous configurations/versions
        - Configuration in database
            - Migrate database configuration to the file system
            - Migrate file system configuration back to the database
  - Prepare target database
  - Migrate the data
  - After the migration
    - Restore full-text indexes
    - Compare the data
    - Restore the search path
  - Plugin migrations
    - Collaborative playbooks
    - Focalboard
    - Calls
  - Troubleshooting

Mobile VPN options

To connect to your private network Mattermost instance, you need to set up a way to connect to your private network Mattermost instance, using an external proxy with encrypted transport through HTTPS and WSS network connections.

Depending on your security policies, we recommend deploying Mattermost behind a VPN and using a per-app VPN with your EMM provider, or a mobile VPN client.

Also consider deploying a mobile VPN client with multi-factor authentication (MFA) to your preferred login method, such as GitLab SSO with MFA, or run Mattermost Enterprise Edition with multi-factor authentication (MFA) enabled.

Mobile VPN options

A Virtual Private Network (VPN) allows a device outside a firewall to access content inside the firewall as if it were on the same network.

> Note
>
> Some mobile VPN options depend on the requirements of your organization and the demands and/or the needs of your users.

We recommend one of two options: per-app VPN or a device VPN to secure your deployment. Both options are compatible with most EMM providers.

We also recommend you review the following commonly-asked questions about data security on mobile devices:

- How data is handled on a device after an account is deleted?
- What post metadata is sent in mobile push notifications?
- What are my options for securing the Mobile apps?
- What are my options for securing push notifications?

Per-app VPN

A common approach is to use a per-app VPN. This provides a connection to the VPN when needed (on-demand). If using a per-app VPN with Mattermost, you can configure the following options:

- **useVPN:** Mattermost waits until the connection to the VPN server is established before making any requests (otherwise they will fail). This is only supported on iOS as Android OS cannot support waiting. It still works but the first connection attempt may fail.
- **timeoutVPN (iOS only):** How long to wait for the connection to the VPN server before trying.

Device VPN

With this option, all internet traffic routes through the VPN specified in the profile. This could cause issues for personal applications.

Connect via corporate proxy server

Review the following commonly-asked questions about connecting through a corporate proxy server:

- How do I receive mobile push notifications if my IT policy requires the use of a corporate proxy server?
- Deploy Mattermost with connection restricted post-proxy relay in DMZ or a trusted cloud environment
- Whitelist Mattermost push notification proxy to bypass your corporate proxy server
- Run App Store versions of the Mattermost Mobile apps

On this page

- Mobile VPN options
  - Mobile VPN options
    - Per-app VPN
    - Device VPN

- Connect via corporate proxy server

Deploy using an EMM provider

You can enhance mobile security by deploying the Mattermost mobile app with Enterprise Mobility Management (EMM) (https://en.wikipedia.org/wiki/Enterprise_mobility_management) and Mattermost AppConfig compatibility to secure mobile endpoints with management application configuration.

You can use an EMM to:

- Enforce users to download the Mattermost pre-built or custom apps managed by your organization.
- Set default server url address.
- Restrict users from changing servers.
- Enforce security policies.

An EMM provider pushes Mattermost Mobile apps to EMM-enrolled devices. This approach is recommended for organizations that typically use EMM solutions to deploy Mobile apps to meet security and compliance policies.

Manage app configuration using AppConfig

AppConfig is our recommended approach for app configuration and management. It was introduced by the AppConfig Community (https://www.appconfig.org/), a group of leading EMM providers and app developers who have come together to make it easier for developers and customers to drive mobility in business.

AppConfig provides an easy way to configure enterprise mobile apps with any EMM providers listed on the AppConfig website. Using AppConfig, you can manage default settings and security controls on public app stores and custom-built mobile clients. For example, you can pre-configure your Mattermost server URL and username.

Mattermost AppConfig values

The following table shows all the configuration options that can be sent from the EMM provider of your choice to the Mattermost mobile apps. You can also download an XML template of the configuration file for use with your EMM provider.

Key	Description	Type	Default Value	Valid Values	iOS Support	Android for Work Support
inAppPinCode	Require users to authenticate as the device owner before using the app. Prompts for fingerprint or passcode when the app first opens and when the app has been in the background for more than five minutes.	String	`false`	`true | false`	Yes	Yes
blurApplicationScreen	Blur the app when it's set to background to protect any confidential on-screen information. On Android, it also prevents taking	String	`false`	`true | false`	Yes	Yes

Key	Description	Type	Default Value	Valid Values	iOS Support	Android for Work Support
	screenshots of the app.					
jailbreakDetection	Disable app launch on jailbroken or rooted devices.	String	`false`	`true | false`	Yes	Yes
copyAndPasteProtection	Disable the ability to copy from or paste into any text inputs in the app.	String	`false`	`true | false`	Yes	Yes (since v1.24.0)
serverUrl	Set a default Mattermost server URL. Supports a single server only while v2.0 of the mobile app supports multiple server connections.	String		URL	Yes	Yes
serverName	Automatically populates the server	String		alphanumeric or empty	Yes	Yes

Key	Description	Type	Default Value	Valid Values	iOS Support	Android for Work Support
	display name for the URL specified in serverURL.					
allowOtherServers	Allow the user to change the above server URL. If set to `true`, users can connect to multiple servers that aren't specified in the server URL setting. If set to `false`, users can only connect to a single defined server.	String	`true`	`true | false`	Yes	Yes

Key	Description	Type	Default Value	Valid Values	iOS Support	Android for Work Support
username	Set the username or email address to use to authenticate against the Mattermost Server.	String			Yes	Yes
useVPN	Enable connection to the Mattermost Server to use a per-app VPN or VPN on-demand.	String	`false`	`true | false`	Yes	No
timeoutVPN	Set how long the request waits (in milliseconds) for an initial VPN connection to establish before timeout.	String	30000		Yes	No
vendor	Name of the EMM vendor or company	String	Mattermost		Yes	Yes

Key	Description	Type	Default Value	Valid Values	iOS Support	Android for Work Support
	deploying the app. Used in help text when prompting for passcodes so users are aware why the app is being protected.					
inAppSessionAuth	Use the app's internal browser for SSO instead of an external browser. From Mattermost v10.2 and mobile v2.2.1, deprecated in favor of the mobile external browser server configuration setting.	String	`false`	`true | false`	Yes	Yes

Other AppConfig settings

As part of AppConfig, EMM administrators can set the following additional configuration options for the Mattermost mobile apps:

1. **App Tunnel:** Leverage the "Per-app VPN" capabilities in most commercial VPN solutions.
2. **Prevent App Backup:** Prevent users from backing up app data.
3. **Enforce App Encryption:** Set security policies such as requiring encryption.
4. **Remotely Wipe App:** Use the EMM tool to distribute the app to devices as a managed application so it can be remotely wiped. If the app was previously installed, mark it so the EMM converts the app to a managed app.

Other configurations may be available depending on your EMM provider.

Important

- Mattermost only supports the AppConfig standard for securing Mattermost mobile apps via an EMM provider due to incompatibilities with app wrapping and React Native applications. Different EMM vendors refer to "wrapping" in different ways, but it ultimately comes down to unpacking the mobile client bundle, injecting additional SDKs, and re-packaging/re-signing. React Native is the technology used to develop the Mattermost mobile apps.
- Mattermost doesn't support app wrapping, and Mattermost mobile apps won't function properly when using app wrapping (e.g., Websockets for real-time messaging will break). Use app wrapping/containerization technology at your own risk.
- A Mattermost Enterprise subscription plan (or a legacy Enterprise Edition license) is required to request assistance or troubleshooting help from Mattermost Customer Support (https://mattermost.com/support/) when building and deploying custom mobile apps. Customers on other Mattermost subscription plans can develop and deploy custom mobile apps, but can't request technical support assistance through Mattermost Customer Support.

- With the release of Mattermost mobile app v2.0, mobile app v1.55 becomes the official extended support mobile release, and will be supported for an extended timeframe.

Enroll devices

When building your own custom versions or deploying the pre-built Mattermost Mobile apps, consider your organization's mobile policy:

- Can users bring their own device (BYOD) If so, what devices will be used?
- Are devices company-owned and company-issued?
- Are both options supported?
- What operating systems do you want to start testing?

Once you know what possible device configurations you'll be supporting, consider creating a sample configuration, then running validation tests against each configuration item.

Generate and assign device profiles

Generate and assign a device profile for device-wide configurations through the EMM provider.

On this page

- Deploy using an EMM provider
  - Manage app configuration using AppConfig
    - Mattermost AppConfig values
    - Other AppConfig settings
  - Enroll devices
  - Generate and assign device profiles

Distribute a custom mobile app

To control the look and feel of the Mattermost mobile app requires building your own mobile apps, hosting your own push proxy service, and managing your own app distribution.

> Note
>
> - Mattermost Enterprise customers are eligible for support guidance on distributing their own custom mobile apps.
> - With the release of Mattermost mobile app v2.0, mobile app v1.55 becomes the official extended support mobile release, and v1.55 will continue to be supported for an extended timeframe.

Key considerations

The Mattermost Mobile App is an open source project. Customizing Mattermost mobile apps requires a fork of the source code. Your team will be responsible for maintaining that fork, as well as keeping that fork updated with any changes made by Mattermost.

Building your own mobile apps will present some challenges, including:

- Installing the necessary developer tools (such as Nodejs, XCode Developer Tools, Android SDKs, as well as others).
- Obtaining and providing certificates for your custom Mattermost mobile apps*.
- Signing your custom Mattermost mobile apps*.
- Distributing your Mobile app to your users.

This means that you manage the maintenance of your custom Mattermost mobile apps, such as rebuilding and incorporating feature and/or security updates. If this isn’t done regularly, your applications won’t match the

functionality of our publicly-available applications, and could be incompatible with future versions of Mattermost Server.

This process can be complicated and can greatly increase deployment time, not only initially, but whenever the mobile apps need to be updated. We recommend having your development team review the Mattermost Mobile Apps developer documentation to ensure they understand the scale and requirements of taking this path. This documentation provides guidance on building, compiling, signing, and white-labeling Mattermost Mobile apps.

URL schema limitations

If you are building your own version of Mattermost's mobile client, you need to be aware of the following limitations:

- To allow users to simultaneously run the App Store versions of Mattermost, in addition to the custom company version, you will need to adapt the URL schemes used for the app in the build, as well as configure those schemes on the server using App Custom URL Schemes
- Be aware that the `bundleid` for the application should not include `rnbeta`.
- The same change would be required in a custom build of the Mattermost desktop app.
- The mobile and desktop custom clients would no longer be able to log into other Mattermost servers (unless they had the same custom app schema configuration change applied).

Deployment options

When you decide to build your own Mattermost mobile apps, you have multiple ways to deploy them to your organization.

Our recommend approach is to submit your app to an Enterprise App Store. Once your custom app is added to your own enterprise App Store, your users can download it from the store directly or from an EMM catalog.

Alternatively, use *an Enterprise Mobile Management (EMM) provider </deployment-guide/mobile/deploy-mobile-apps-using-emm-provider>* to push the mobile app to the user's device, and use the AppConfig standard to

enforce a selection of app-specific controls. Or, you can use another distribution method, such as a file sharing platform.

You can also submit your app to public app stores. This is the same process Mattermost uses to make Mattermost mobile apps available for everyone. However, before your app can be listed on the public app stores, you need to submit it to the public app stores for review and approval.

- As part of the submission process, you need to identify an update strategy that accounts for the release of new versions of Mattermost mobile apps that includes reviewing compatibility requirements, validating mobile app versions connecting to the server, and updating Mattermost server.
- We highly recommend you update your custom Mattermost mobile apps to incorporate any security or service releases.
- Prior to distribution, check any compatibility requirements for the mobile apps and the Mattermost server.
- Not all provided updates are compatible with all previous versions of Mattermost server. Updating only Mattermost mobile apps or updating the mobile apps before Mattermost Server can result in incompatibility issues.

Custom whitelabeling

Ensure you select a unique app name that helps users distinguish your version from others, such as "<Your Company Name> Collaboration". See our [Brand and Visual Design Guidelines (https://handbook.mattermost.com/operations/operations/company-processes/publishing/publishing-guidelines/brand-and-visual-design-guidelines#name-usage-guidelines.html)](https://handbook.mattermost.com/operations/operations/company-processes/publishing/publishing-guidelines/brand-and-visual-design-guidelines#name-usage-guidelines.html) in our company Handbook for details.

On this page

- Distribute a custom mobile app
  - Key considerations
    - URL schema limitations
  - Deployment options
  - Custom whitelabeling

Host your own push proxy service

Customers building their own custom mobile apps must host their own push proxy service using one of the following methods:

- Compile your own MPNS from the open source repository (https://github.com/mattermost/mattermost-push-proxy).
- Use the pre-compiled version of MPNS available on GitHub (https://github.com/mattermost/mattermost-push-proxy/releases).

See our developer documentation on working with the Mattermost Push Notification Service.

Enable MPNS

1. Go to **System Console > Environment > Push Notification Server**.
2. Under **Enable Push Notifications**, select **Manually enter Push Notification Service location**.
3. Enter the location of your MPNS in the **Push Notification Server** field, then select **Save**.

Deployment recommendations:

- Deploy the MPNS on your own infrastructure - whether on-premises behind your firewall or in your public cloud.
- Ensure the Mattermost server has network access to the MPNS instance. Mattermost clients don't need any inbound connectivity.
- The MPNS does not connect with Mattermost mobile apps directly; the MPNS parses and forwards push notifications from the Mattermost server to the Apple Push Notification Service (APNS) or the Firebase Cloud Messaging (FCM).
- The MPNS must be able to communicate with the Apple Push Notification Service over HTTP/2. If an outbound proxy appliance is deployed between the MPNS and APNS, ensure it supports HTTP/2.

- Configure TLS encryption between your MPNS and your Mattermost server.
- Sign the mobile applications during your build process, and obtain and configure valid push certificates for both Android and iOS. Without these certificates, the apps will be unable to send or receive notifications via your MPNS instance.
- Subscribe to Mattermost Security Bulletins (https://mattermost.com/security-updates/#sign-up) to receive and promptly apply MPNS-related security updates.

Configure ID-only push notification contents (Enterprise)

By default, push notification message content passes through Apple Push Notification Service (APNS) or Google Firebase Cloud Messaging (FCM) before it reaches a device. This potentially presents a problem for organizations with ultra-strict security and compliance requirements.

With the ID-only option, message contents can be fetched directly from the server once a push notification is delivered to a device. APNS and FCM can't read push notifications since only a unique message ID is sent in each notification payload. While message content will take slightly longer to send, this feature helps organizations meet advanced compliance requirements.

To set mobile push notification contents to ID-only:

1. Go to **System Console > Site Configuration > Notifications**.
2. Under **Push Notification Contents**, select **Full message content fetched from the server on receipt**, then select **Save**.
    - As part of the process of building the applications, you'll need to sign the applications. You must also obtain the appropriate certificate for both Android and iOS. If this isn't done, the applications won't be able to interact with your instance of the MPNS. Once this is complete, you can proceed with the deployment of your MPNS instance.
    - We strongly recommend that you subscribe to Mattermost Security Bulletins (https://mattermost.com/security-updates/#sign-up). When you're notified of security updates for the MPNS, apply them promptly.

On this page

- Host your own push proxy service
  - Enable MPNS
  - Configure ID-only push notification contents (Enterprise)

Mobile App Deployment

The Mattermost mobile app is available for iPhone and Android devices, and provides a native experience on the go, ensuring you can stay connected and productive from anywhere.

Learn more about mobile app software requirements, available releases and server compatibility, what's changed across releases, and commonly asked questions.

Download

Download and install the Mattermost mobile app from the Apple App Store (iOS) (https://www.apple.com/app-store/) or Google Play Store (Android) (https://play.google.com/store/games?hl=en). When new mobile app releases become available, your mobile app is automatically updated.

If you prefer to manage distribution of the mobile app to your users, we recommend using an EMM provider to maintain full control over the distribution process, as well as enforce or restrict specific security policies. See the deployment options below for details.

Deployment options

Learn what's required to build and deploy Mattermost mobile apps.

- Distribute custom mobile apps
- Host your own push proxy service
- Mobile VPN options
- Mobile security features
- Secure mobile file storage
- Mobile apps FAQ

On this page

- Mobile App Deployment
  - Download
  - Deployment options

Mobile apps FAQ

Can I connect to multiple Mattermost servers using the mobile apps?

Yes, using Mattermost mobile app v2.0. Mobile v1.x only supports connecting to one server at a time.

What data is stored?

The data that can be found on the device depends solely on whether or not the user is logged in to the Mattermost server, and is independent of the state of the device's connection or the state of the app. While logged in, anything that the user is normally allowed to see is eligible for storage on the device, which includes the following content:

- Messages
- Files and images that are attached to messages
- Profile pictures, usernames, and email addresses of people in the currently open channel

In addition, metadata that the app uses for keeping track of its operations is also cached. The metadata includes user IDs, channel IDs, team IDs, and message IDs.

Currently, cache cannot be reset remotely on connected mobile devices.

What about push notifications?

Push notification storage is managed by the operating system on the device. Mattermost can be configured to send limited amounts of information that does not include the message text or channel name, and it can also be configured to not send push notifications at all.

Where is the data stored and how is that data protected?

The data is stored in the app's local storage. It's protected by the security measures that a device normally provides to the apps that are installed on it.

How long is the data stored?

Data is stored until the user logs out, or until it is purged during normal cache management. Deactivating a user account forces a logout and subsequent purging of data from the device.

How is data handled after a user account is deactivated?

App data is wiped from the device when a user logs out of the app. If the user is logged in when the account is deactivated, then within one minute of deactivation the system logs the user out. Thereafter all app data is wiped from the device.

If file attachments are enabled on the server, users can download files that are attached to messages and store them on their local file system. After they are downloaded, the files are outside the control of the app and can remain on the device indefinitely.

Are messages pre-loaded?

No. Messages are sent to the device on demand. They are not pre-loaded in anticipation of users scrolling up or switching channels.

Do I need to compile the mobile apps to host my own push notification server?

Yes. To host your own push notification server, you'll need to compile the mobile apps. See documentation to learn how to compile your own mobile apps.

How do push notifications work?

Your Mattermost server sends push notifications to a hosted push proxy server, which relays them via mobile push notification services provided by Apple and Google.

To ensure push notifications are coming from a trusted source, Apple and Google only allow push notifications sent from a service using a key or signature corresponding to a secret compiled into the mobile app itself.

The full process is outlined below:

1. An action triggering a push notification is detected in the Mattermost server running in your private network.
2. Your Mattermost server sends a push notification message to a Mattermost Push Notification Service (MPNS), either self-hosted in your private network, or publicly hosted by Mattermost, Inc.
3. MPNS sends a push notification message to either Apple Push Notification Service (APNS) or to Google's Firebase Cloud Messaging (FCM) service over a TLS connection.

> - *If sent to Apple, the message has a signature corresponding to a secret compiled in the iOS app.*
> - *If sent to Google, the message uses a key corresponding to a secret compiled in the Android app.*

Regardless of whether you're using iOS or Android, the MPNS needs to have access to the appropriate secret compiled into the mobile app.

This means if you use the Mattermost apps from the [Apple App Store (https://www.apple.com/app-store/)](https://www.apple.com/app-store/) or [Google Play Store (https://play.google.com/store/games?hl=en)](https://play.google.com/store/games?hl=en), you need to use the hosted push notification service from Mattermost, Inc. If you compile the apps yourself, you must also compile and use your own MPNS with the corresponding secret.

4. Either APNS or FCM receives the push notification message from MPNS over TLS, and then relays the message to the user's iOS or Android device to be displayed.

> Note
>
> The use of push notifications with iOS and Android applications will require a moment where the contents of push notifications are visible and unencrypted by a server controlled by either Apple or Google. This is standard for any iOS or Android app. For this reason, there is an option available in Mattermost Enterprise to omit the contents of Mattermost messages from push notifications, or to configure message contents to be fetched from the server when notifications reach the device. See our Configuration Settings documentation for details.

Is TLS v1.3 supported?

Yes. Mattermost mobile app v2.0 supports both TLS v1.2 and TLS v1.3 for websocket connections.

What post metadata is sent in mobile push notifications?

The following post metadata is sent in all push notifications:

- `Team ID`
- `Channel ID`
- `Post ID`
- `User ID` (post author)
- `Username` (post author or webhook override username)
- `Root ID` (only if the post is in a thread)
- `Type` (create or clear push notification)
- `Category` (iOS only, determines if the notifications can be replied to)
- `Badge number` (what the notification badge on the app icon should be set to when the notification is received)

Additional metadata may be sent depending on the System Console setting for Push Notification Contents:

- **Generic description with sender and channel names**: `Channel name` metadata will be included.
- **Full message content sent in the notification payload**: `Post content` and `Channel name` metadata will be included.
- **Full message content fetched from the server on receipt** (available in Mattermost Enterprise): `Post content` and `Channel name` are not included in the notification payload, instead the `Post ID` is used to fetch `Post content` and `Channel name` from the server after the push notification is received on the device.

How can I use ID-Only Push Notifications to protect notification content from being exposed to third-party services?

When it comes to mobile data privacy, many organizations prioritize secure handling of messaging data, particularly when it may contain mission-critical or proprietary information. These organizations may have concerns about using mobile notifications because data must pass through third-party entities like Apple Push Notification Service (APNS) or Google Firebase Cloud Messaging (FCM) before it reaches a device.

This poses a potential risk for organizations that operate under strict compliance requirements and cannot expose message data to external entities. To solve this, we offer an option for greater protection for Mattermost push notification message data by only sending a unique message ID in the notification payload rather than the full message data (available in Mattermost Enterprise). Once the device receives the ID, it then fetches the message content directly from the server and displays the notification per usual.

External entities, such as APNS and FCM, handle only the ID and are unable to read any part of the message itself. If your organization has strict privacy or compliance needs, the ID-Only Push Notification setting offers a high level of privacy while still allowing your team members to benefit from mobile push notifications.

The following payload shows an example of the json that is transmitted to the push notification service when using the ID-Only setting:

```
{  "ack_id": "nnfbqk5bnffe5karxuzs8o5rec",  "platform": "apple_rn",
"server_id": "aoej8izzfffr9e67d6uz3g387h",  "device_id":
"32f198dbdd7427be7e6f03ba721ffdceba58c3f0bfa9c4655a6e7cc8271ba539",
"post_id": "77d9cs9aq3b1fpoepbdbmqfs4c",  "category": "CAN_REPLY",
"message": "You've received a new message.",  "badge": 3,  "channel_id":
"et3ghiycm7g7bb41ihg85pqgah",  "type": "message",  "sender_id":
"g774dzud4tgaxgphso4wm8xrxe",  "version": "v2",  "is_id_loaded": true }
```

where the following definitions are applied:

- `ack_id`: An ephemeral identifier generated per notification that determines whether the notification sent was received by the device (using same method that generates identifiers to the rest of the models in the server). This information is available in the `notifications.log` file on the Mattermost server. The `ack_id` is only used for receipt delivery from the mobile app to the Mattermost server to confirm whether the notification sent was received.

- `server_id`: A server identifier created on the server, called `DiagnosticId`. In the future, this value will be used in the mobile app (for multi-server support) to identify which server the notification belongs to.

- `device_id`: The token that APNs and FCM return when you allow the device to receive notifications. So when the user logs into Mattermost, Mattermost sends this `device_id` to attach it to the session. If the session is terminated, the `device_id` is no longer present in the server database because the session record is removed. When the user logs back in, the `device_id` is registered again with the same value because the identifier is specific to the device. This value won't be the same across apps or devices owned by the same person, but will be the same for each session the user creates from the same app on the same device.

- `version`: Tells the mobile app how data is structured so it can parse it properly. Current value is `v2`.

- `is_id_loaded`: (Mattermost Enterprise only) When true, the mobile app look for the contents of the notification on the server because those details are not part of the payload.

What are my options for securing the mobile apps?

The following options for secure mobile app deployments are available:

1. Securing network connection to mobile apps

> - *Use HTTPS and WSS network connections to encrypt transport.*
> - *Use of a mobile VPN client on mobile devices to establish secure connection to Mattermost server within private network.*

2. Use multifactor authentication options

> - *If a VPN client with multifactor authentication is not in use, it's highly recommended that MFA is required on authenticating into Mattermost, either within Mattermost itself or via your SSO provider.*

What are my options for securing push notifications?

The following options are available for securing your push notification service:

1. Protecting notification contents

> - *You can choose what type of information to include in push notifications, such as excluding the message contents if your compliance policies require it. Default server settings have message contents turned off.*

2. Disabling push notifications

> - *Push notifications can also be disabled entirely depending on security requirements. Default server settings have push notifications disabled.*

3. Encrypting connections for apps you compile yourself:

> - *When using a privately-hosted Mattermost Push Notification Service (MPNS), use encrypted TLS connections between:*
> - *MNPS and Apple Push Notification Service (APNS)*
> - *MPNS and Google's Firebase Cloud Messaging (FCM)*

> ◦ *MPNS and your Mattermost server*

4. Securing apps installed through the Apple App Store and Google Play:

> - *When using Mattermost mobile apps from the App Store and Google Play, purchase an annual subscription to Mattermost Enterprise or Professional to use Mattermost's Hosted Push Notification Service (HPNS).*

> Note
>
> For configuration details, see guides for deploying the Mattermost mobile app and deploying your own version of the apps.

Why do I sometimes see a delay in receiving a push notification?

Apple Push Notification Service (APNS) (https://developer.apple.com/documentation/usernotifications) and Google Fire Cloud Messaging (FCM) determine when your device receives a push notification from Mattermost. Thus, a delay is usually as a result of those services.

The technical flow for the device to receive a push notification is as follows:

1. User posts a message in Mattermost.
2. Mattermost server identifies if notifications need to be sent.
3. If yes, Mattermost server sends a payload containing the push notification to the push proxy.
4. The push proxy parses the notification and relays it to APNS and FCM.
5. APNS and FCM informs the relevant devices that there is a push notification for Mattermost. This usually happens almost immediately, but may be delayed by a couple of minutes.
6. Mattermost processes the notification and displays it on the user's device.

How do I deploy Mattermost with Enterprise Mobility Management (EMM) providers?

Mattermost enables customers with high privacy and custom security requirements to deploy mobile app and push notification services using keys that they alone control.

Learn more about using AppConfig for EMM providers.

How do I host the Mattermost push notification service?

First, you can use the Mattermost Hosted Push Notification Service (HPNS). Organizations can also host their own push proxy server instead. This is applicable when you want to:

1. Customize the Mattermost mobile apps;
2. Deploy your own push notification service, or
3. Repackage the mobile apps with BlueCedar or AppDome (both of which are not officially supported but have been successfully deployed by some organizations).

How do I white label the app and customize build settings?

All files in the `/assets/base` folder can be overridden as needed without conflicting with changes made to the upstream version of the app. To do this:

1. Create the folder `/assets/override`.
2. Copy any files or folders that you wish to replace from `/assets/base` into `/assets/override`.
3. Make your changes to the files in `/assets/override`.

When you compile the app or run `make dist/assets`, the contents of those two folders will be merged with files in `/assets/override`, taking precedence in the case of any conflicts. For binary files such as images, an overridden file will

completely replace the base version, while `JSON` files will be merged so that fields not set in the overridden copy use the base version.

For a more specific example of how to use this feature, see the following section.

How do I preconfigure the server URL for my users?

You can preconfigure the server URL and other settings by overriding default `config.json` settings and building the mobile apps yourself.

1. Fork the [mattermost-mobile repository (https://github.com/mattermost/mattermost-mobile)](https://github.com/mattermost/mattermost-mobile).
2. Create the file `/assets/override/config.json` in your forked mattermost-mobile repository.
3. Copy and paste all the settings from `assets/base/config.json` to the newly-created `/assets/override/config.json` file that you want to override.
4. To override the server URL, set `DefaultServerURL` to the server URL of your Mattermost server in `/assets/override/config.json`.
5. (Optional) If you want to prevent users from changing the server URL, set `AutoSelectServerUrl` to `true`.
6. (Optional) Override any other settings you like.

After the above, your `/assets/override/config.json` file would look something like this:

> ```
> { "DefaultServerURL": "my-mattermost-instance.example.com",
> "AutoSelectServerUrl": true, "ExperimentalUsernamePressIsMention":
> true }
> ```

7. Finally, [compile your own version (https://developers.mattermost.com/contribute/mobile/build-your-own/)](https://developers.mattermost.com/contribute/mobile/build-your-own/) of the Mattermost mobile app and Mattermost push proxy server.

How can I get Google SSO to work with the Mattermost mobile app?

The apps on the Apple App Store and Google Play Store cannot support Google SSO out of the box. This is because Google requires a unique Google API key that's specific to each organization.

If you need Google SSO support, you can create a custom version of the app for your own organization. Fork the [mattermost-mobile (https://github.com/mattermost/mattermost-mobile)](https://github.com/mattermost/mattermost-mobile) repository and add support for Google SSO before compiling the app yourself. If this is something you're interested in, please [file an issue in GitHub (https://github.com/mattermost/mattermost-mobile/issues)](https://github.com/mattermost/mattermost-mobile/issues) to start the discussion.

How do I configure deep linking?

The app checks for platform-specific configuration on app install. If no configuration is found, then the deep linking code sits silently and permalinks act as regular links.

Set up for iOS

1. Create an `apple-app-site-association` file in the `.well-known` directory at the root of your server. It should be accessible by navigating to `https://<your-site-name>/.well-known/apple-app-site-association`. There should not be a file extension.

2. In order to handle deep links, paste the following `JSON` into the `apple-app-site-association` file. Make sure to place your app ID in the `appID` property:

```
{ "applinks": { "apps": [], "details": [ { "appID": "<your-app-id-here>",
"paths": ["**/pl/*", "**/channels/*"] } ] } }
```

3. Add the associated domains entitlement to your app via the Apple developer portal.

4. Add an entitlement that specifies the domains your app supports via the Xcode entitlements manager.

5. Before installing the app with the new entitlement, make sure that you can view the contents of the `apple-app-site-association` file via a browser by navigating to `https://<your-site-name>/.well-known/apple-app-site-association`. The app will check for this file on install and, if found, will allow outside permalinks to open the app.

Official documentation for configuring deep linking on iOS can be found here (https://developer.apple.com/library/archive/documentation/General/Conceptual/AppSearch/UniversalLinks.html).

Set up for Android

Please refer to the the App Links Assistant in Android Studio for configuring deep linking on Android (https://developer.android.com/studio/write/app-link-indexing).

How do I connect users across internal and external networks?

By setting up global network traffic management, you can send a user to an internal or external network when connecting with a mobile app. Moreover, you can have two separate layers of restrictions on internal and external traffic, such as:

> - *In the internal network, deploy on a private network via per device VPN.*
> - *In the external network, deploy with TLS mutual auth with an NGINX proxy, and client-side certificates for desktop and iOS.*

Many services such as Microsoft Azure provide options for managing network traffic (https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview), or you can engage a services partner to assist.

How do I receive mobile push notifications if my IT policy requires the use of a corporate proxy server?

When your IT policy requires a corporate proxy to scan and audit all outbound traffic the following options are available:

Deploy Mattermost in a proxy-aware configuration with a pre-proxy relay

The Mattermost push notification service is designed to send traffic directly to the Apple Push Notification Service (APNS) (https://developer.apple.com/documentation/usernotifications) and Google Fire Cloud Messaging (FCM) services.

In a proxy-aware configuration, a pre-proxy relay accepts messages from the Mattermost Push Proxy (https://developers.mattermost.com/contribute/mobile/push-notifications/service/) and forwards them to a corporate proxy enforcing your internal IT requirements, before transmitting to their final destination.

See a sample architectural overview below:

This enables the **pre-proxy relay** to act as the APNS and to forward the request to its final destination via your corporate proxy, not requiring the APNS traffic to be proxy-aware. The APNS traffic is redirected to the pre-proxy relay via `/etc/hosts` entry. The entry uses a trusted CA that signs a certificate for the Mattermost Push Proxy to trust the pre-proxy relay. See the Apple Developer documentation on [user notifications (https://developer.apple.com/documentation/usernotifications)](https://developer.apple.com/documentation/usernotifications) for more information.

Google's [FCM traffic (https://firebase.google.com/docs/cloud-messaging)](https://firebase.google.com/docs/cloud-messaging) is proxy-aware via environment variables, so no actions are required for it.

Moreover, APNS traffic requires HTTP/2, so your corporate proxy server must support HTTP/2 requests in order to send the push notifications to Apple devices. HTTP/2 support for the pre-proxy relay is also required.

Deploy Mattermost with connection restricted post-proxy relay in DMZ or a trusted cloud environment

Some legacy corporate proxy configurations may be incompatible with the requirements of modern mobile architectures, such as the requirement of HTTP/2 requests from Apple to send push notifications to iOS devices.

In this case, a post-proxy relay can be deployed to take messages from the Mattermost server passing through your corporate IT proxy in the incompatible format, e.g. HTTP/1.1, transform it to HTTP/2 and relay it to its final destination, either to the Apple Push Notification Service (APNS) (https://developer.apple.com/documentation/usernotifications) and Google Fire Cloud Messaging (FCM) services.

Ths **post-proxy relay** can be configured using the Mattermost Push Proxy installation guide (https://developers.mattermost.com/contribute/mobile/push-notifications/service/) with connection restrictions to meet your custom security and compliance requirements.

In place of a DMZ, you can also host in a trusted cloud environment such as AWS or Azure depending on your internal approvals and policies.

Whitelist Mattermost push notification proxy to bypass your corporate proxy server

Depending on your internal IT policy and approved waivers/exceptions, you may choose to deploy the Mattermost Push Proxy (https://developers.mattermost.com/contribute/mobile/push-notifications/service/) to connect directly to Apple Push Notification Service (APNS) (https://

developer.apple.com/documentation/usernotifications) without your corporate proxy.

You will need to whitelist one subdomain and one port from Apple (https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/CommunicatingwithAPNs.html#//apple_ref/doc/uid/TP40008194-CH11-SW1) for this option:

- Development server: `api.development.push.apple.com:443`
- Production server: `api.push.apple.com:443`

Run App Store versions of the Mattermost mobile apps

You can use the mobile applications hosted by Mattermost in the Apple App Store (https://apps.apple.com/ca/app/mattermost/id1257222717) or Google Play Store (https://play.google.com/store/apps/details?id=com.mattermost.rn) and connect with the Mattermost Hosted Push Notification Service (HPNS) through your corporate proxy.

> Note
>
> The use of hosted applications by Mattermost can be deployed with Enterprise Mobility Management solutions via AppConfig but wrapping is not supported. See the product documentation for details.

How the `deviceId` behaves

The `deviceId` is a identifier provided by a push notification service, such as Apple Push Notification service (APNs) and Firebase Cloud Messaging (FCM), that identifies the relationship between device, app, and the notification service.

When the app starts, if the push notification permissions are enabled, the app will try to connect with the corresponding notification service (APNs for iOS, FCM for Android) to get the `deviceId`. If there is any change to the `deviceId`, the app will notify any connected server about this change.

Based on the Apple Developer Documentation on registering apps with APNS and the Google Documentation on FCM, the `deviceId` only will change in the following cases:

- The app is restored from a backup from a different device
- The user clears the app data
- The user reinstalled the app
- The user installed the app on a different device

If the device has a `deviceId`, when the user logs into a Mattermost server, an audit log entry `login` will store the `deviceId`, and the `deviceId` will also be added in the session data in the database. However, it's possible the `deviceId` isn't available, due to several reasons including:

- The device is not connected to the network
- The notification service is not reachable for any reason
- The app is not properly signed
- The device has not been granted the needed permissions

In these scenarios, the `login` audit log won't have the `deviceId`, and the session data won't have the `deviceId`. If the app receives the `deviceId` later, the device will send the new `deviceId` to the server, generating an `attachDeviceId` audit log, and adding the `deviceId` to the session data in the database.

Since the `deviceId` relates to the application, connections through the web browser, even on mobile, won't have a `deviceID`.

Where can I find mobile message notification logs?

Notification messages are logged to the `notifications.log` file. System admins must enable notification logs in the `config.json` file by setting `EnableFile` to `true`, and specifying an optional file location via `FileLocation`. When no location is configured, the `notifications.log` file is stored in the default

Mattermost directory. See the logging configuration settings documentation for details.

The team members / users can access their notification logs based on their device platform. Android users can view the logs using `logcat`. iOS users can acess the logs in the `console app` on their MacOS.

How do I gather mobile app logs?

See the Mattermost Support Knowledge Base article on gathering mobile app logs (https://support.mattermost.com/hc/en-us/articles/30147601934100-How-to-Gather-Mobile-App-Logs) for details.

On this page

- Mobile apps FAQ
  - Can I connect to multiple Mattermost servers using the mobile apps?
  - What data is stored?
    - What about push notifications?
  - Where is the data stored and how is that data protected?
  - How long is the data stored?
  - How is data handled after a user account is deactivated?
  - Are messages pre-loaded?
  - Do I need to compile the mobile apps to host my own push notification server?
  - How do push notifications work?
  - Is TLS v1.3 supported?
  - What post metadata is sent in mobile push notifications?
  - How can I use ID-Only Push Notifications to protect notification content from being exposed to third-party services?
  - What are my options for securing the mobile apps?
  - What are my options for securing push notifications?
  - Why do I sometimes see a delay in receiving a push notification?
  - How do I deploy Mattermost with Enterprise Mobility Management (EMM) providers?
  - How do I host the Mattermost push notification service?
  - How do I white label the app and customize build settings?

- How do I preconfigure the server URL for my users?
- How can I get Google SSO to work with the Mattermost mobile app?
- How do I configure deep linking?
- How do I connect users across internal and external networks?
- How do I receive mobile push notifications if my IT policy requires the use of a corporate proxy server?
  - Deploy Mattermost in a proxy-aware configuration with a pre-proxy relay
  - Deploy Mattermost with connection restricted post-proxy relay in DMZ or a trusted cloud environment
  - Whitelist Mattermost push notification proxy to bypass your corporate proxy server
  - Run App Store versions of the Mattermost mobile apps
  - How the deviceId behaves
- Where can I find mobile message notification logs?
- How do I gather mobile app logs?

Mobile security features

This document outlines the key security features implemented in the mobile application, explains the significance of each control, and details our continuous security practices to maintain an up-to-date security posture.

Jailbreak and root detection

Mattermost leverages built-in checks from the Expo framework to identify jailbroken (iOS) and rooted (Android) devices:

- On **Android**, the app looks for the presence of known jailbreak/root binaries, such as the `su` binary in `/system/xbin/su`, which is a common indicator that the device has been rooted to allow unauthorized elevated access.
- On **iOS**, the detection process involves checking for unusual apps (e.g., Cydia), modified system paths, and testing whether the app can alter protected system files—all signs that the device may be jailbroken.

See the jailbreak/root protection configuration setting documentation for details on enabling this feature.

> Note
>
> Client-side detection methods are not bulletproof and can be bypassed by a sufficiently motivated user with root or jailbreak access. Determined users can employ reverse engineering to disable or modify these checks to produce false positives/negatives or even modify and recompile the source code to remove them entirely, allowing app usage on rooted/jailbroken devices.

Biometric authentication

Mattermost integrates with iOS Face ID/Touch ID and Android's Biometric API. When enabled by the server administrator, biometric checks are required before accessing specific servers, and the Mattermost mobile app mandates that a device PIN or biometric lock is active.

See the biometric authentication configuration setting documentation for details on enabling this feature and the user workflows in which users must authenticate.

> Note
>
> Biometric authentication, such as Face ID or fingerprint recognition, is handled by the device's operating system. Mattermost does not provide or warrant the biometric functionality. The app will unlock if the OS determines that the biometric input is a positive match. Accuracy, security, and availability of biometric authentication depend entirely on the device and its underlying platform.

Screenshot and screen recording prevention

Preventing screenshots and screen recordings protects sensitive information from being inadvertently or maliciously shared. This control is essential in ensuring that confidential communications and data remain within the secure confines of the app. By blocking unauthorized screen captures, Mattermost significantly reduces the risk of data leakage via visual content.

- On **iOS**, the app detects attempts to capture the screen. For screen recordings, it immediately applies a blur overlay to the view to obscure any sensitive information, while screenshots will capture no content at all.
- On **Android**, the app utilizes the FLAG_SECURE flag to block screen captures and recordings.

See the prevent screen capture configuration setting documentation for details on enabling this feature.

> Note
>
> This feature prevents screen recording and screenshot capture on the mobile device where the Mattermost app is running. It cannot control or prevent recording from external devices, such as another phone or camera pointed at the screen. Mattermost provides these protections through mobile OS-level capabilities but cannot guarantee absolute prevention of visual data capture through other means.

Mobile data isolation

Mattermost mobile applications are designed to ensure that sensitive data is stored securely and isolated from other applications on the device. This isolation is achieved through a combination of OS-level security features, app sandboxing, and secure data storage practices.

Mattermost stores 3 types of data within the mobile app:

- **SQLite Database**: For offline access to posts and channels that the user has viewed.
- **Cached Files**: For temporary storage (e.g., PDFs) of viewed files. They are periodically purged.
- **Logs**: Minimal logs kept for troubleshooting purposes.

The Mattermost mobile app for Apple iOS and Android devices uses the native OS security architecture to encrypt data-at-rest on mobile devices, ensuring that stored information remains secure even if the device is lost or stolen. This encryption protects sensitive data, such as messages and files, from unauthorized access.

Data stored by the Mattermost mobile app only resides within the app's private storage container. This storage location is isolated by each platform's rigorous sandboxing model. Learn more about secure file storage for Mattermost mobile applications.

Security measures

- **OS sandboxing & app access isolation**: Mattermost's data is confined within its app-specific sandbox (see Apple's iOS sandboxing (https://support.apple.com/guide/security/security-of-runtime-process-sec15bfe098e/web) and Android's sandboxing (https://source.android.com/docs/security/app-sandbox) guidelines). This sandboxing mechanism ensures that all stored data—whether in the SQLite databases, cached files, or logs—is isolated and inaccessible to other applications. This is why TikTok, and any other application is blocked from accessing Mattermost data.
  - **iOS** employs a rigorous sandboxing model that relies on containerized file systems and strict process isolation. Each app operates within its own sandbox, with its home directory randomly assigned during installation. As a result, data stored within the Mattermost app's sandbox directory remains accessible only to the app unless explicitly shared by the user.
  - **Android** employs scoped storage by assigning each app its own Linux user ID and a dedicated private directory. This model prevents other applications from accessing an app's private directory without explicit permission. As a result, the Mattermost app's data remain inaccessible to other apps—unless the device is rooted, or the user explicitly shares the data.
- **Cache invalidation**: Cached files are periodically deleted.
- **No Google or Apple Cloud backup**: Mattermost files are not backed up during normal cloud backup procedures.
  - On **Android**, the application explicitly disables backups (using the appropriate manifest settings), ensuring that all Mattermost files remain strictly on-device.
  - On **iOS**, although the operating system automatically backs up files stored in the app's `Documents` folder, Mattermost intentionally stores its files in the `Library/Caches` directory, which is excluded from iCloud backups.

- **Controlled data export**: Data only leaves Mattermost through explicit user actions such as:
  - Manually exporting a file.
  - Takes a screenshot (if not blocked by policy or with a secondary camera).
  - Sharing content using OS features (e.g., share sheet).

> Note
>
> This feature restricts direct access to stored data through robust OS-level sandboxing and storage controls. However, it cannot prevent unauthorized extraction if attackers employ alternative methods. For instance, on jailbroken (iOS) or rooted (Android) devices, where system integrity is compromised, an attacker may bypass standard security measures.
>
> Similarly, exploiting zero-day or known OS vulnerabilities—or malicious apps that exploit permission weaknesses—could grant access to device storage. Mattermost uses mobile OS-level safeguards but cannot guarantee absolute prevention of such unauthorized data access.

Security compliance

- **Secure SDLC**: The mobile app adheres to the same secure Software Development Lifecycle (SDLC) process as all Mattermost components. This guarantees that new mobile features undergo comprehensive security reviews and that any issues are addressed before release.
- **Vulnerability management**: The Mattermost mobile app is included in Mattermost's public bug bounty program, inviting security researchers to rigorously assess the application. Additionally, established remediation SLAs ensure that any security issues are resolved promptly.
- **Continuous dependency scanning**: Mattermost uses Snyk Open Source to conduct daily scans of all dependencies. This continuous approach

ensures that any newly disclosed vulnerabilities are quickly identified and addressed within defined SLAs.

- **BESPIN compliance**: Mattermost adheres to strict security evaluations including:
  - Device Protection Profiles (NIAP Evaluation)
  - Security Dependency & Code Analysis
  - Code Complexity & Testing Standards

On this page

- Mobile security features
  - Jailbreak and root detection
  - Biometric authentication
  - Screenshot and screen recording prevention
  - Mobile data isolation
    - Security measures
  - Security compliance

Mobile deployment troubleshooting

I keep getting a message "Cannot connect to the server. Please check your server URL and internet connection."

First, confirm that your server URL has no typos and that it includes `http://` or `https://` according to the server deployment configuration.

If the server URL is correct, there could be an issue with the SSL certificate configuration.

To check your SSL certificate set up, test it by visiting a site such as SSL Labs (https://www.ssllabs.com/ssltest/index.html). If there's an error about the missing chain or certificate path, there is likely an intermediate certificate missing that needs to be included.

Please note that the apps cannot connect to servers with self-signed certificates, consider using Let's Encrypt instead.

Login with ADFS/Office365 is not working

In line with Microsoft guidance we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).

I see a "Connecting..." bar that does not go away

If your app is working properly, you should see a grey "Connecting…" bar that clears or says "Connected" after the app reconnects.

If you are seeing this message all the time, and your internet connection seems fine, ask your server administrator if the server uses NGINX or another

webserver as a reverse proxy. If so, they should check that it is configured correctly for supporting the websocket connection for APIv4 endpoints.

All my outbound connections need to go through a proxy. How can I connect to the Mattermost Hosted Push Notification Service?

You can set up an internal server to proxy the connection out of their network to the Mattermost Hosted Push Notification Service (HPNS) by following the steps below:

1. Make sure your proxy server is properly configured to support SSL. Confirm it works by checking the URL at https://www.digicert.com/help/.
2. Setup a proxy to forward requests to `https://push.mattermost.com`.
3. In Mattermost set **System Console** > **Notification Settings** > **Mobile Push** > **Enable Push Notifications** in prior versions or **System Console > Environment > Push Notification Server > Enable Push Notifications** in versions after 5.12 to "Manually enter Push Notification Service location"
4. Enter the URL of your proxy in the **Push Notification Server** field.

> Note
>
> Depending on how your proxy is configured you may need to add a port number and create a URL like `https://push.internalproxy.com:8000` mapped to `https://push.mattermost.com`

Build gets stuck at `bundleReleaseJsAndAssets`

As a workaround, you can bundle the `js` manually first with

```
react-native bundle --platform android --dev false --entry-file index.js --
bundle-output android/app/src/main/assets/index.android.bundle --assets-
dest android/app/src/main/res/
```

and then ignore the gradle task with

```
./gradlew assembleRelease -x bundleReleaseJsAndAssets
```

No image previews available in the mobile app

This can happen if the server running Mattermost has its mime types not set up correctly. A server running Linux has this file located in `/etc/mime.types`. This might vary depending on your specific OS and distribution.

Some distributions also ship without `mailcap` which can result in missing or incorrectly configured mime types.

Messages with emojis aren't being sent from the Mobile App

This can occur if the server running Mattermost is configured with an incorrect character set. To resolve this issue, in the `config.json` file under `SqlSettings`, ensure that the `DataSource` key is configured correctly, then restart the Mattermost server.

For example:

```
"SqlSettings": { "DataSource": "<user:pass>@<servername>/mattermost?
charset=utf8mb4,utf8", [...] }
```

See our Configuration Settings documentation for details on configuring the connection string to the master database.

Testing mobile push notifications

Make sure to configure push notifications for your pre-built mobile apps, or for your custom built mobile apps.

Then use the following instructions to confirm push notifications are working properly.

1. Log in to your mobile app with an account on your Mattermost Server, which we'll refer to as "Account A".
2. (iOS) When the app asks whether you wish to receive notifications, **confirm you want to receive notifications**.

3. Confirm push notifications are enabled for "Account A".

> *1. Go to the notification settings menu in the mobile app.*

2. Check that the mobile push notifications are set to send.

4. Have “Account A” put the app to background or close the app.

5. Using a browser, log in to “Account B” on the same Mattermost Server.

6. Open a direct message with “Account A”, and send a message.
7. A push notification with the message should appear on the mobile device of “Account A”.

Troubleshooting push notifications

If you did not receive a push notification when testing push notifications, use the following procedure to troubleshoot:

1. In **System Console > Environment > Logging > File Log Level**, select **DEBUG** in order to watch for push notifications in the server log.
2. Delete your mobile application, and reinstall it.
3. Log in with “Account A” and **confirm you want to receive push notifications** when prompted by the mobile app.
4. Go to **Profile** > **Security** > **View and Logout of Active Sessions** to confirm that there is a session for the native mobile app matching your login time.
5. Retest push notifications.
6. If no push notification displays, go to **System Console** > **Server Logs**, then select **Reload**. Look at the bottom of the logs for a message similar to:

```
[2016/04/21 03:16:44 UTC] [DEBG] Sending push notification to 608xyz0... wi msg of '@accountb: Hello'
```

> - *If the log message displays, it means a message was sent to the HPNS server and was not received by your mobile app. Please [create a support ticket (https://support.mattermost.com/hc/en-us/requests/new)](https://support.mattermost.com/hc/en-us/requests/new) with the subject “HPNS issue” for help from Mattermost’s Support team.*
> - *If the log message does not display, it means no mobile push notification was sent to “Account A”. Please repeat the process starting at step 2 and double-check each step.*

Important

To conserve disk space, once your push notification issue is resolved, go to **System Console > Environment > Logging > File Log Level**, then select **ERROR** to switch your logging detail level from **DEBUG** to **Errors Only**.

If push notifications are not being delivered on the mobile device, confirm that you're logged in to the **Native** mobile app session through **Profile > Security > View and Log Out of Active Sessions**. Otherwise, the *DeviceId* won't get registered in the *Sessions* table and notifications won't be delivered.

On this page

- Mobile deployment troubleshooting
  - I keep getting a message "Cannot connect to the server. Please check your server URL and internet connection."
  - Login with ADFS/Office365 is not working
  - I see a "Connecting…" bar that does not go away
  - All my outbound connections need to go through a proxy. How can I connect to the Mattermost Hosted Push Notification Service?
  - Build gets stuck at bundleReleaseJsAndAssets
  - No image previews available in the mobile app
  - Messages with emojis aren't being sent from the Mobile App
  - Testing mobile push notifications
  - Troubleshooting push notifications

Secure file storage

This document outlines the security measures governing file storage in the Mattermost mobile app for iOS and Android. It describes how files are stored, accessed, and protected within the application container, addressing concerns related to sensitive data on mobile devices. The objective is to ensure that only authorized personnel can view, access, or share such data while preventing exposure to unauthorized parties and ensuring isolation from third-party applications.

Mattermost leverages robust sandboxing mechanisms on both iOS and Android to securely store files in its cache folder within the application container, ensuring isolation from unauthorized third-party apps. User-initiated actions (download, share, copy link) are controlled by server-side permissions, allowing administrators to restrict access to sensitive data, like HIPAA-regulated information, to authorized personnel only.

File storage and app sandbox security

Files uploaded or accessed via the Mattermost mobile app are stored in the app's cache folder, which resides within the app's private storage container. This storage location is isolated by each platform's sandboxing model:

iOS – app sandbox

- **Sandboxing & storage:**

  iOS employs a rigorous sandboxing model that relies on containerized file systems and strict process isolation. Each app operates within its own sandbox, with its home directory randomly assigned during installation. As a result, files stored within the Mattermost app's cache folder remain accessible only to the app unless explicitly shared by the user.

- **Secure file viewing:**

  - Non-image and non-video files are rendered using secure frameworks such as QLPreviewController. This controller displays file previews within

the app's sandbox, ensuring that raw file data isn't exposed to external processes.

- Supported image and video files are previewed directly within the app, with the files downloaded to the app's cache folder located within its secure sandbox. For unsupported image and video formats, the Mattermost mobile app uses the QLPreviewController framework, just as it does for other file types.
- If the file format is also unsupported by QLPreviewController and mobile downloads are enabled on the server, users can download the file to a location of their choosing. However, if mobile downloads are disabled, these files become unavailable to the user.

- **Official references:**
  - Apple App Sandbox Design Guide (https://developer.apple.com/library/archive/documentation/Security/Conceptual/AppSandboxDesignGuide/AboutAppSandbox/AboutAppSandbox.html)
  - Security of runtime process in iOS, iPadOS and visionOS (https://support.apple.com/en-gb/guide/security/sec15bfe098e/web)
  - Apple Security Guide (https://help.apple.com/pdf/security/en_GB/apple-platform-security-guide-b.pdf)

Android – scoped storage

- **Sandboxing & storage:**

  Android employs scoped storage by assigning each app its own Linux user ID and a dedicated private directory. This model prevents other applications from accessing an app's cache without explicit permission. As a result, the Mattermost app's files remain inaccessible to other apps—unless the device is rooted, or the user explicitly shares the data.

- **Secure file viewing:**
  - When users attempt to view non-image/video files, Mattermost uses an `Intent.ACTION_VIEW` to open the file. This intent delegates rendering to an external app only if the user explicitly triggers the action, while the file remains securely stored within Mattermost's cache folder.

  - Viewing non-image/video files is available only if mobile downloads are enabled on the server.
  - Image and video files with supported formats are previewed directly within the app, with the files downloaded to the app's cache folder located within its secure sandbox. For unsupported image and video formats, the Mattermost mobile app uses `Intent.ACTION_VIEW` to open the file with an external application, just as it does for other file types.

- **Official reference:**
  - [Scoped Storage (https://developer.android.com/about/versions/10/privacy/changes#scoped-storage)](https://developer.android.com/about/versions/10/privacy/changes#scoped-storage)

Additionally, both platforms may clear cached data automatically (e.g., under low storage conditions or upon user logout), thereby reducing the risk of sensitive information remaining on the device.

Differentiating file handling to external applications

- **Previewing files:**

  File previewing follows the secure viewing practices described above in the "Secure File Viewing" sections for iOS and Android. All files prior to being previewed are stored in the cache folder of the Mattermost app sandbox. Images and videos with supported formats are previewed directly within the Mattermost mobile app. Non-image and non-video files are also previewed in-app in iOS but are handed off to an external application in Android while the raw data remains securely stored in the app's cache. Previewing non-image/non-video files is possible only if mobile downloads are enabled on the server side.

- **Downloading files:**
  - When a user initiates a file download, the file is first stored in Mattermost's private cache directory. The user is then immediately prompted to save it to another destination via platform-specific sharing methods.

- On **iOS** devices:
  - The built-in sandboxing ensures that the downloaded files remain inaccessible to other apps unless the user explicitly shares or saves them through the system's file sharing mechanisms. Once downloaded, the file appears in the gallery.
  - If the server has the download setting enabled, the user has two options:
    1. Download the file: Opens the **Files** app, allowing the user to save the file to any permitted location, including iCloud.
    2. Share the file: Launches the native share sheet, enabling the user to share the file with other compatible applications.
- On **Android** devices, the app employs the system's file picker. This allows the end-user to select a preferred storage location. Once saved, the file becomes accessible to other apps based on the chosen location and the device's file sharing permissions.

File accessibility options

Depending on server configuration, users may download files, share them, or copy a public link directly from the Mattermost mobile app. These actions are always user-initiated, ensuring that sensitive data is only exposed with explicit consent.

Additionally, administrators have the option to restrict mobile file uploading and downloading to ensure that sensitive content remains within approved channels. When mobile file downloads are disabled, the app allows only image and video file previews. However, if general file sharing is enabled on the Mattermost server, file operations via mobile browsers will still be possible.

Layered defense strategies for sensitive eata

Mattermost is designed to protect sensitive data—such as sensitive information and HIPAA-regulated information—by ensuring that only authorized personnel can access or share it. The platform combines secure mobile app features with powerful server-side controls, providing a layered defense that minimizes the risk of data exposure.

Core defense pillars

- **Robust authentication:** Mattermost requires user authentication through SSO (e.g., SAML, LDAP, OpenID Connect) or traditional username/password logins. This authentication is managed by server-side identity controls, ensuring that only verified users can access the app and its data. For more details, see the Mattermost Security Overview (https://docs.mattermost.com/about/security.html#authentication-safeguards).
- **Server-side access controls:** Administrators can enforce policies through the System Console to restrict file downloads, sharing, and public link generation. Currently, policies are applied at the server level. For more details, see Configuration Settings - File Sharing and Downloads.
- **Sandbox isolation:** As discussed earlier, Mattermost's mobile apps store files in a sandboxed environment. This isolation ensures that even if a device is shared or compromised, other apps cannot access the cached files without explicit user action.
- **Controlled file previews:** File previews (via QLPreviewController on **iOS** or open intent on **Android**) occur within the application's controlled environment or delegate to OS-approved apps without duplicating sensitive data outside the sandbox.
- **OS-Level encryption:** While files in the cache folder are stored unencrypted by default, they are protected at rest by OS-level encryption (iOS File Data Protection and Android Full Disk Encryption).

By combining strict authentication, server-side controls, sandbox isolation, and secure file handling, Mattermost ensures that sensitive data remains confined to authorized users, with access and sharing tightly controlled by server policies, platform security features, and explicit user consent.

Testing procedure to validate file isolation

The playbook below aims to test and confirm that files stored in the Mattermost app's private cache folder—where sensitive data may reside—are not accessible by unauthorized third-party applications. The tests are designed to show that while public directories, such as Downloads/ can be accessed, the

private Mattermost cache remains isolated unless a file is explicitly shared by an authorized user.

1. **Preparation**
    - On **Android**:
        - Install Adobe Acrobat Reader for PDF viewing.
        - Install Files by Google as a file manager.
        - Install Google Chrome as a system browser.
    - On **iOS**:
        - Install Adobe Acrobat Reader (from the App Store).
        - Use the built-in Files app.
        - Use Safari as the default browser.
    - Any PDF viewer, file manager, or browser can be used for these steps.
2. **Control test: Verify public directory accessibility**
    - **Android:**
        - Open Files by Google.
        - Navigate to a known public directory (e.g., Downloads/).
        - Confirm that files in this location are visible and can be opened.
    - **iOS:**
        - Open the Files app.
        - Browse to an accessible directory (e.g., iCloud Drive or "On My iPhone" public folders).
        - Confirm these files are accessible.

    This step demonstrates that allowed directories can be accessed, setting a baseline for comparison.

3. **Attempt unauthorized access to Mattermost's private cache**
   - **Android:**
     - In Files by Google, try navigating through the directory structure to identify files of the Mattermost application.
     - Verify that you cannot access private directories such as Mattermost app's private cache folder.
   - **iOS:**
     - Open the Files app.
     - Verify that the Mattermost app's private cache folder does not appear under "On My iPhone/iPad."
4. **Attempt unauthorized access to a specific file in Mattermost's private cache**
   - **Android:**
     - Use the search functionality in Files by Google to search for a known file name from the Mattermost cache.
     - Verify that the file does not appear in search results.
   - **iOS:**
     - Use the Files app's search capability to look for a specific file name from the Mattermost cache.
     - Confirm that the file does not appear in search results.
5. **Attempt authorized access to downloaded file**
   - **Android:**
     - Confirm that mobile downloads are enabled on the server.
     - Download a file using the Mattermost mobile app.
     - Open the Files by Google app and verify that the downloaded file is visible and opens correctly.

- **iOS:**
    - Confirm that mobile downloads are enabled on the server.
    - Download a file using the Mattermost mobile app.
    - Open the iOS Files app and verify that the downloaded file appears and can be opened successfully.

6. **Repeat across multiple apps and devices**

   The steps above use the Files app of Google and iOS. Repeat these tests using each suggested app on different devices and OS versions (both Android and iOS) to ensure that the private cache folder remains inaccessible under all conditions.

Test execution & findings

Mattermost executed the playbook above, including scanning for hidden files, both during normal operation and after forcing the app to stop or crash, and observed the following results:

- **Devices Tested**: iPhone 12, iOS 18.4, Android 11
- **Apps Used**: Adobe Reader, Google Files app, Android's native Files app, iOS Files app, Files Media Manager, Hidden Files Finder, Hidden Files Finder & Recover
- **Results**: Third-party file browsers and recovery tools consistently failed to access the Mattermost app's cache folder or its contents. Access was only possible when files were explicitly downloaded or shared via Mattermost's controlled functionality by an authorized user.
- **Outcome**: Files remain isolated within the app container, adhering to platform security guidelines and protecting against unauthorized access.

The documented testing procedures confirm that the app's file storage remains inaccessible to unauthorized third-party applications, thereby maintaining a strong security posture. Coupled with strong authentication and native OS encryption, the Mattermost mobile app provides a robust framework that keeps sensitive data on mobile devices confined to authorized users while addressing key access control and risk mitigation requirements.

On this page

- Secure file storage
  - File storage and app sandbox security
    - iOS – app sandbox
    - Android – scoped storage
  - Differentiating file handling to external applications
  - File accessibility options
  - Layered defense strategies for sensitive eata
    - Core defense pillars
  - Testing procedure to validate file isolation
  - Test execution & findings

Automated PostgreSQL migration

Migrating databases can be a daunting task, and it can be easy to overlook or misinterpret some of the required steps if you haven't performed a migration before. Our `migration-assist` tool provides an efficient, error-free migration experience that automates the tasks to be executed, even in air-gapped deployment environments.

Not sure this tool is right for your Mattermost deployment? Mattermost customers looking for tailored guidance based on their Mattermost deployment can contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/).

Install

Download the Mattermost `migration-assist` tool from the GitHub repository [releases page (https://github.com/mattermost/migration-assist/releases)](https://github.com/mattermost/migration-assist/releases).

While you can run the `migration-assist` tool on the same server as your Mattermost deployment, we recommend running the tool in a virtual machine on the same network as your Mattermost server instead. The tool itself is lightweight and does not require a large server. A server with 2 CPU cores and 16 GB of RAM should be sufficient. If preferred, you can download and compile the `migration-assist` tool yourself.

You'll also need to install the `pgloader` tool to migrate your data from MySQL to PostgreSQL. We recommend running `pgloader` in a virtual machine on the same network as your Mattermost server. You can use our official Mattermost Docker image for pgloader (`mattermost/pgloader:latest`); please note that it does **not** currently support MySQL's `caching_sha2_password` authentication plugin. If you require `caching_sha2_password` support, you'll need to build your own image and include the [qitab/qmynd (https://github.com/qitab/qmynd)](https://github.com/qitab/qmynd) library. See the pgloader installation documentation for details.

Usage

Important

- If you encounter heap exhaustion errors in `pgloader`, edit your generated `migration.load` and under the `WITH` block set: `prefetch rows = 1000` and consider reducing it if the issue persists.
- Please make sure you have the necessary environment to perform the migration. Ensure that the MySQL and PostgreSQL databases are running and accessible. To set up a PostgreSQL instance, see the prepare your Mattermost database documentation for details.

Step 1 - Check the MySQL database schema

Run the following command to check the MySQL database schema:

```
migration-assist mysql "<MYSQL_DSN>" # example DSN:
"user:password@tcp(address:3306)/db_name"
```

This command outputs the readiness status and prints required fixes for common issues. The flags for fixes are as follows (you can combine all of them):

```
--fix-artifacts Removes the artifacts from older versions of Mattermost --
fix-unicode Strips unsupported Unicode characters from MySQL tables --fix-
varchar Removes rows exceeding column lengths
```

Step 2 - Create the PostgreSQL database schema

Before running migrations, ensure the `public` schema is owned by your migration user:

```
sudo -u postgres psql -d mattermost -c
"ALTER SCHEMA public OWNER TO mmuser; GRANT ALL ON SCHEMA public TO
mmuser;"
```

Then run:

```
migration-assist postgres "<POSTGRES_DSN>" \  --run-migrations \  --
mattermost-version="<MATTERMOST_VERSION>"
```

- `<POSTGRES_DSN>` example: `postgres://user:password@address:5432/db_name`
- `<MATTERMOST_VERSION>` example: `10.5.4`

By default, two pre-checks run before migration:

- `--check-schema-owner=true`
- `--check-tables-empty=true`

To disable them:

```
--check-schema-owner=false \ --check-tables-empty=false
```

Step 3 - Generate a pgloader configuration

Run the following command to emit a pgloader configuration file:

```
migration-assist pgloader \  --mysql="<MYSQL_DSN>" \  --
postgres="<POSTGRES_DSN>" \  --remove-null-chars \  > migration.load
```

- By default, null characters in text columns are stripped. Disable with `--remove-null-chars=false`.
- If you run out of heap, edit `migration.load` and under the `WITH` block set:

  ```
  prefetch rows = 1000
  ```

Step 4 - Run pgloader

Run pgloader with the generated configuration file:

```
pgloader migration.load > migration.log
```

Carefully review *migration.log* for errors (e.g., duplicate-key or missing-table warnings). Use the `mattermost/pgloader:latest` Docker image to avoid build/auth issues.

Step 5 - Restore full-text indexes & create all indexes

Run:

```
migration-assist postgres post-migrate --create-indexes "<POSTGRES_DSN>"
```

- The `--create-indexes` flag rebuilds full-text indexes on `Posts` and `FileInfo`, plus all other Mattermost indexes.
- Omitting that flag only restores full-text indexes for `Posts` and `FileInfo`.

See the Restore full-text indexes documentation for details.

Step 6 - Complete plugin migrations

Generate pgloader configs for Playbooks, Boards, and Calls:

```
migration-assist pgloader boards \  --mysql="<MYSQL_DSN>" \  --
postgres="<POSTGRES_DSN>" > boards.load migration-assist pgloader playbooks
\  --mysql="<MYSQL_DSN>" \  --postgres="<POSTGRES_DSN>" > playbooks.load
migration-assist pgloader calls \  --mysql="<MYSQL_DSN>" \  --
postgres="<POSTGRES_DSN>" > calls.load
```

Run each:

```
pgloader boards.load > boards_migration.log pgloader playbooks.load >
playbooks_migration.log pgloader calls.load > calls_migration.log
```

Skip any plugin you don't use; check logs for JSON or missing-table errors. See the Plugin migrations guide for more.

Step 7 - Configure Mattermost to use PostgreSQL

In your `config.json` or via environment variables, update:

```
"SqlSettings": {  "DriverName": "postgres",  "DataSource": "postgres://
mmuser:pass@db:5432/mattermost?sslmode=disable" }
```

If your config was stored in the database, update `MM_CONFIG` accordingly. See the environment configuration settings documentation for details.

> Note
>
> If your Mattermost deployment was initially configured with MySQL, there's a good chance your systemd service file has a `BindsTo=mysql.service` directive in it. This will cause the Mattermost server to be shut down if you deactivate your MySQL service. To fix this, update all references to `mysql.service` in your service file to use `postgresql.service` instead. This is only an issue if your Database and Mattermost are running on the same system.

Air-gapped environments

Follow these steps to migrate within an air-gapped environment:

1. **Verify** that the *migration-assist* binary is the latest version available to benefit from improvements and fixes.
2. **Transfer** the latest *migration-assist* binary into the air-gapped environment (e.g., via secure media).
3. **Generate** the MySQL schema+data output using fix flags. This produces `mysql.output`:

   ```
   migration-assist mysql "user:pass@tcp(localhost:3306)/mattermost" \  --
   fix-artifacts --fix-unicode --fix-varchar > mysql.output
   ```

4. **Apply** migrations using the generated output:

   ```
   migration-assist postgres \  "postgres://mmuser:pass@localhost:5432/
   imported?sslmode=disable" \  --run-migrations --applied-migrations="./
   mysql.output"
   ```

5. **Continue** from **Step 3** through **Step 7** above to complete data transfer and configuration.

Tool commands

The `migration-assist` tool offers 3 core commands:

1. `migration-assist mysql` — Checks MySQL schema readiness and offers fixes.
2. `migration-assist postgres` — Builds the PostgreSQL schema and applies migrations.
3. `migration-assist pgloader` — Generates a pgloader config for data transfer.

Compile the migration-assist tool

Requires Go ≥ v1.22. Install with:

```
go install github.com/mattermost/migration-assist/cmd/migration-assist@latest
```

Troubleshooting

See troubleshooting errors during migration from MySQL to PostgreSQL.

On this page

- Automated PostgreSQL migration
  - Install
  - Usage
    - Step 1 - Check the MySQL database schema
    - Step 2 - Create the PostgreSQL database schema
    - Step 3 - Generate a pgloader configuration
    - Step 4 - Run pgloader
    - Step 5 - Restore full-text indexes & create all indexes
    - Step 6 - Complete plugin migrations
    - Step 7 - Configure Mattermost to use PostgreSQL
  - Air-gapped environments

- Tool commands
- Compile the migration-assist tool
- Troubleshooting

Migration guidelines from MySQL to PostgreSQL

From Mattermost v8.0, PostgreSQL is our database of choice for Mattermost to enhance the platform's performance and capabilities. Recognizing the importance of supporting the community members who are interested in migrating from a MySQL database, we have taken proactive measures to provide guidance and best practices.

- Automated migration from MySQL to PostgreSQL - A comprehensive set of guidelines and a `migration-assist` tool to streamline the migration process, alleviate potential challenges, and faciliate a smooth transition.
- Manually migrate from MySQL to PostgreSQL - A good option if your organization has database administrators to own the migration process, or if you want to learn what the `migration-assist` tool automates for you.

Frequently asked questions

Why is Mattermost dropping support for MySQL?

Mattermost has decided to drop support for MySQL databases to streamline development and focus on delivering more efficient and optimized features. By supporting only PostgreSQL, which is generally considered more advanced and better-suited for enterprise environments, Mattermost can reduce complexity, improve performance, and better allocate resources to enhance the product. This change helps ensure more consistent, robust, and scalable database interactions for all Mattermost deployments.

Is migrating to PostgreSQL before upgrading Mattermost recommended?

Yes. We also recommend upgrading to Mattermost server v8.x or later before migrating to PostgreSQL.

Can the migration-assist be run on the Mattermost server?

Technically, yes. The `migration-assist` tool can be run on the Mattermost server. However, it is recommended to run the tool on a separate server to avoid any performance issues. We advise running the migration against a copy of the MySQL database to ensure that the migration process does not impact the production environment.

How large should the PostgreSQL server be?

The size of the PostgreSQL server should match that of the MySQL server initially. We recommend monitoring the performance of the PostgreSQL server and adjusting the resources as needed.

How large should the server running `migration-assist` server be?

The tool itself is lightweight and does not require a large server. A server with 2 CPU cores and 16 GB of RAM should be sufficient for default configurations. However, you may need to adjust the resources based on the size of the MySQL database, your downtime limitations, and the configuration of `pgloader`.

Do we/will Mattermost bundle pgloader or is that a separate install?

Mattermost doesn't bundle pgloader with the Mattermost server. You will need to install pgloader separately. For more information, see the install pgloader documentation.

Are there any other migrations available for plugins, or just Boards, Playbooks, and Calls?

We're developing migrations for other plugins, such as NPS-plugin. Please stay tuned for updates.

Do these processes support AWS RDS databases?

Yes, the processes support AWS RDS databases. However, you may need to adjust the security group settings to allow the migration process to access the databases.

Does Mattermost support MariaDB? If not, why not?

Mattermost does not support MariaDB. The primary reason for this decision is to streamline development and maintenance by focusing on a single database technology. By standardizing on PostgreSQL, Mattermost can deliver optimized features, improved performance, and a more focused engineering effort. PostgreSQL is well-regarded for its robustness, advanced features, and suitability for enterprise use, making it the chosen database for Mattermost.

MariaDB, while compatible with MySQL to a large extent, introduces additional complexity and potential inconsistency, which the Mattermost development team aims to avoid by limiting database support.

Because of these reasons we also don't support migrations directly from MariaDB to PostgreSQL. Oracle provides documentation on [how to migrate from MariaDB to MySQL](https://blogs.oracle.com/mysql/post/how-to-migrate-from-mariadb-to-mysql-80) which can be used as a starting point for the migration to PostgreSQL.

Troubleshooting

Unsupported authentication for MySQL

If you are facing an error due to authentication with MySQL v8, it may be related to a [known issue (https://github.com/dimitri/pgloader/issues/782)](https://github.com/dimitri/pgloader/issues/782) with pgloader. The fix is to set the default authentication method to `mysql_native_password` in your MySQL configuration. To do so, add the `default-authentication-plugin=mysql_native_password` value to your `mysql.cnf` file. Also, do not forget to update your user to use this authentication method.

```
ALTER USER '<mysql_user>'@'%' IDENTIFIED WITH mysql_native_password BY
'<mysql_password>';
```

Errors during the pgloader command execution

If you encounter errors during the execution of the `pgloader` command, ensure that both of the databases are accessible and that the users have the necessary permissions to access the database. Do not continue with the migration if there are errors during the execution of the `pgloader` command.

> Note
>
> For experienced users, it is recoverable to run the `pgloader` without requiring a restart of the migration from scratch. In this case, you will need to manually fix the issues with the table, and run the `pgloader` command with a tailored configuration specifically for those tables. Also ensure that the schema name is reverted back to `public`, and the `search_path` is restored (or remove necessary clauses from the configuration).

The following sections detail how to resolve some common errors you may encounter during the execution of the `pgloader` command:

Invalid input syntax for type JSON

If you receive an error message similar to the following:

```
ERROR Database error 22P02: invalid input syntax for type json
```

The data in the MySQL database is not in a valid JSON format. You can fix this issue by updating the data in the MySQL database to be in a valid JSON format. To find out which row is causing the issue, run the following query (where `<table_name>` and `<column_name>` should be replaced with the actual table and column names indicated in the `pgloader` output):

```
SELECT * FROM <table_name> WHERE JSON_VALID(<column_name>) = 0;
```

You can find and update the data in the MySQL database to be in a valid JSON format with the query above. After updating the data, you can run the `pgloader` command again.

Failed to find column or table

If you receive an error message similar to the following:

```
pgloader failed to find column
```

The column or table is missing in the PostgreSQL database. You can fix this issue by checking whether you have created the correct version of Postgres schema. After re-creating the schema, you can run the `pgloader` command again.

Fell through ECASE expression

If you receive an error message similar to the following:

```
ERROR mysql: 76 fell through ECASE expression.
```

It is a [known issue (https://github.com/dimitri/pgloader/issues/1183)](https://github.com/dimitri/pgloader/issues/1183) with pgloader. You can fix this issue by either compiling `pgloader` from source or simply avoid this by running `pgloader` with our docker image. See: install pgloader for more information.

> Note
>
> Also, there may be cases where pgloader continues to migrate remaining tables and skip one or more tables during migration. In such cases, we recommend identifying issues with the table and fixing them before running the `pgloader` command again with a clean database. It is possible to run the `pgloader` command with the `--debug` flag to get more information about the errors.

Mattermost can’t connect to the PostgreSQL database

If you are facing an issue where Mattermost can’t connect to the PostgreSQL database, ensure that the PostgreSQL server is running and that the database is accessible. If there were errors during the execution of the `pgloader` command, it can fail to revert the schema name back to `public`, or potentially

fail to restore the `search_path`. You can manually revert the schema name back to `public` and restore the `search_path` by running the following commands:

1. Connect to PostgreSQL using `sudo -u postgres psql`.
2. Select the `mattermost` database using `\c mattermost`. Verify you are using the right database by running `SELECT current_database();`. The command should output `mattermost`.
3. Revert the schema name change (optional)

```
ALTER SCHEMA <schema_name> RENAME TO public;
```

Also ensure that the database user has the necessary settings to have default access to the `public` schema. You can do this by running the following commands:

4. Set the search_path for the mmuser:

   ```
   ALTER USER mmuser SET search_path TO "$user", public;
   ```

5. Terminate the connection and connect again to your psql server.
6. Verify the search_path is set correctly:

   ```
   SHOW search_path;
   ```

   This should return `"$user", public`.

7. If the issue persists, also run:

   ```
   SELECT pg_catalog.set_config('search_path', '"$user", public', false);
   ```

Permission issues when accessing the schema in PostgreSQL

If you run into a permission issue during step 2 where the command reports a permission issue similar to the following:

```
An Error Occurred: could not check schema owner: the user “mmuser” is not
owner of the “public” schema
```

Ensure that the `mmuser` user in PostgreSQL is the owner of the schema.

1. Connect to PostgreSQL using `sudo -u postgres psql`.
2. Select the `mattermost` database using `\c mattermost`. Verify you are using the right database by running `SELECT current_database();`. The command should output `mattermost`.
3. Run the following commands:

   ```
   ALTER SCHEMA public OWNER TO mmuser; GRANT USAGE, CREATE ON SCHEMA
   public TO mmuser;
   ```

Then, re-run the command from step 2.

Incoming webhook channel names become case-sensitive after migration

After migrating from MySQL to PostgreSQL, incoming webhooks that previously worked with mixed-case channel names (e.g., `"Tech-Talk"`) will fail with the following error:

```
GetChannelByName: Channel does not exist.
```

This occurs because Mattermost doesn’t allow the creation of channels with uppercase characters. All new channels names must only contain lowercase characters. Additionally, MySQL is case-insensitive when matching channel names, while PostgreSQL is case-sensitive. This means that webhook payloads that worked with MySQL’s forgiving case-insensitive matching will fail with PostgreSQL’s strict case-sensitive matching. In rare cases, older channels may still exist with mixed-case names in your database. These channels are considered invalid and may cause issues when performing operations like updating channel headers.

Solution:

Update all webhook payloads to use only lowercase channel names.

Prevention:

Before migrating to PostgreSQL:

1. Audit your incoming webhooks and update any that use mixed-case channel names in their payloads.
2. Check for any legacy channels with mixed-case names that may have been created in earlier versions.
3. Consider renaming any mixed-case channels to lowercase to prevent issues.

This is expected behavior with PostgreSQL and reflects Mattermost's channel naming requirements.

Contact Support

Mattermost customers looking for guidance tailored to their Mattermost deployment can contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) for guidance.

On this page

- Migration guidelines from MySQL to PostgreSQL
  - Frequently asked questions
    - Why is Mattermost dropping support for MySQL?
    - Is migrating to PostgreSQL before upgrading Mattermost recommended?
    - Can the migration-assist be run on the Mattermost server?
    - How large should the PostgreSQL server be?
    - How large should the server running migration-assist server be?
    - Do we/will Mattermost bundle pgloader or is that a separate install?
    - Are there any other migrations available for plugins, or just Boards, Playbooks, and Calls?
    - Do these processes support AWS RDS databases?
    - Does Mattermost support MariaDB? If not, why not?

- Troubleshooting
  - Unsupported authentication for MySQL
  - Errors during the pgloader command execution
    - Invalid input syntax for type JSON
    - Failed to find column or table
    - Fell through ECASE expression
  - Mattermost can’t connect to the PostgreSQL database
  - Permission issues when accessing the schema in PostgreSQL
  - Incoming webhook channel names become case-sensitive after migration
- Contact Support

Quick Start Evaluation

This guide provides instructions for quickly trying out Mattermost using either Docker or Azure Marketplace. These options are ideal for testing and evaluation purposes as they allow you to quickly get a Mattermost instance up and running for exploration and testing. However, these quick start options are not recommended for production use. They are configured for demonstration purposes only.

Deployment options

Azure Marketplace

Mattermost is available as a pre-configured virtual machine image in the Azure Marketplace. This option is preferred for customers already using Azure, as it integrates seamlessly within their existing Azure infrastructure.

Azure Prerequisites

- An Azure subscription
- Basic familiarity with Azure Portal

The Solution Template sets up the following components:

1. **Virtual Machine (Ubuntu):** Hosts the Mattermost application and its associated database, providing the computing environment required to run the system.
2. **Public IP Address:** Allows access to the Mattermost workspace over the public internet, enabling users to connect to the platform securely.
3. **Network Interface:** Facilitates communication between the Virtual Machine and other Azure resources through a defined network layer.
4. **Network Security Group:** Acts as a virtual firewall, controlling inbound and outbound traffic to protect the deployed resources from unauthorized access.

5. **Disk:** Provides persistent storage mounted to the Virtual Machine for storing files generated by Mattermost, including uploaded documents and system configurations.
6. **Virtual Network:** Creates a secure, isolated network environment for the deployed resources, enabling private and controlled connectivity between infrastructure components.

Step 1: Navigate to the Azure Marketplace Offer

Go to the **Mattermost VM Complete** offering in the Azure Marketplace by clicking [this link (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-all-in-one)](https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-all-in-one).

Step 2: Select an Azure Plan and Start Deployment

Select an available Mattermost plan from the dropdown menu, then click **Create** to begin the configuration process.

Step 3: Configure Azure Basics

This step includes several key configurations, all managed on a single screen in Azure:

1. **Subscription:** Choose the Azure subscription you'd like to use.
2. **Resource Group:** Either select an existing resource group or create a new one.
3. **Region:** Select the deployment region for your Virtual Machine (e.g., `East US`, `West Europe`).
4. **VM Name:** Provide a unique name for your Virtual Machine.
5. **First Administrator:** Input the desired *username* and *email* for the first admin of the Mattermost instance.
6. **Authentication Method:** Use **password** authentication for simplicity.

We recommend giving your VM a unique, meaningful name (e.g., yourorganization's name).

Step 4: Choose VM Size and Networking Configuration

1. **Virtual Machine Size:** For trials, we recommend the **D2s v3** size. For Production use, we recommend a more scalable deployment option, such as [Mattermost for AKS (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator)](https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator).

2. **DNS Prefix:** Enter a unique DNS name for the instance. This DNS will allow public access to your workspace. It is important that you do not already have a matching DNS name within your Azure subscription, or your deployment may fail later in the process.

3. **Networking:** Select or create a new Virtual Network and Subnet. Creating new values ensures smooth setup and prevents deployment conflicts.

Step 5: Submit and Deploy

Once the configuration is complete, finalize by clicking **Create**. Azure will take 5-10 minutes to provision the infrastructure and create your new Mattermost instance. During this time, the necessary components (e.g., Virtual Machine, Public IP, Network) are set up.

Once the deployment is complete:

- Navigate to the **Public IP Address** section of your resources.
- Locate a resource named `<your VM name>-ip` and click on it.
- Your DNS name will be listed under "DNS Name" in the "Essentials" pane.
- The DNS name is formatted as follows:

`<dns name you chose>.<region>.cloudapp.azure.com` (e.g., `myorg.eastus.cloudapp.azure.com`).

Step 6: Log In to Your Mattermost Instance

Access your Mattermost deployment via the DNS name from your Public IP resource. Use the admin email and password you set in Step 3 to log in.

Congratulations! You’ve successfully deployed Mattermost via the Azure Marketplace Solution Template. You’re now ready to start managing your team collaboration using Mattermost.

Docker Preview Container

The fastest way to try Mattermost is using the official Docker preview container. This method requires minimal setup and provides a fully functional Mattermost instance.

Docker Preview Prerequisites

- Docker installed on your system
- At least 1GB of available RAM
- At least 1GB of available disk space

Run Mattermost using Docker

1. Pull and run the Mattermost preview container:

```
docker run --name mattermost-preview -d --publish 8065:8065 mattermost/
mattermost-preview
```

2. Access Mattermost at `http://localhost:8065`

> *3. Create your first admin account when prompted.*

Next steps

After setting up your Mattermost instance using either method:

- Create your first team and channels
- Invite users to join your workspace
- Explore Mattermost features and integrations
- Review the Application architecture to understand the system better

- Consider Server deployment for a production deployment

For additional help or questions, visit the [Mattermost community forums (https://forum.mattermost.com/)](https://forum.mattermost.com/) or refer to the Deployment troubleshooting guide.

On this page

- Quick Start Evaluation
- Quick Start Evaluation
  - Deployment options
    - Azure Marketplace
      - Azure Prerequisites
      - Step 1: Navigate to the Azure Marketplace Offer
      - Step 2: Select an Azure Plan and Start Deployment
      - Step 3: Configure Azure Basics
      - Step 4: Choose VM Size and Networking Configuration
      - Step 5: Submit and Deploy
      - Step 6: Log In to Your Mattermost Instance
    - Docker Preview Container
      - Docker Preview Prerequisites
      - Run Mattermost using Docker
  - Next steps
  - Deployment options
  - Next steps

Deploy Mattermost in Air-Gapped Environments

This guide outlines how to deploy Mattermost in a self-hosted air-gapped environment, focusing on setting up the supporting infrastructure required for a successful deployment. An air-gapped environment is one that is isolated from the public internet, requiring all necessary components to be available locally.

Deploying Mattermost in an air-gapped environment involves detailed preparation to ensure uninterrupted availability of critical components within the network. These efforts mitigate risks associated with the lack of internet connectivity by ensuring access to secure, localized resources. Key considerations include:

- Container images required when deploying in Kubernetes or Docker.
- Mattermost server packages and dependencies required when deploying directly on Linux servers.
- Access to a PostgreSQL database.
- Object storage or a shared filesystem service for reliability accessing files from multiple Mattermost servers.
- Other optional supporting services such as:

  > - *SMTP for email functionality.*
  > - *LDAP or SAML for authentication.*
  > - *Elasticsearch for performant post searching.*
  > - *Prometheus or other metric scraping services for enhanced observability.*

Refer to Mattermost deployment documentation for setting up Mattermost as needed. This guide focuses on the supporting infrastructure needed for an air-gapped deployment which is required before deploying Mattermost in these environments.

Set up a self-hosted private container registry

A private container registry securely stores Docker images for air-gapped deployments, ensuring compliance with data isolation requirements. Use it to enable local deployments in Kubernetes or Docker.

The process for uploading images to the registry will need to be tailored to the security policies of the air-gapped network. For example, perhaps the registry is allowed to pull images from the public internet. If not, images will have to be uploaded to the registry from a machine that has access to the public internet.

Ideally the air-gapped network already has a private registry available. If not, you can set up a private registry using Docker Registry or Harbor.

Docker registry

1. **Install Docker Registry**:

   ```
   docker run -d -p 5000:5000 --restart=always --name registry registry:2
   ```

2. **Configure persistent storage**:

   ```
   docker run -d -p 5000:5000 --restart=always --name registry \ -v /mnt/
   registry:/var/lib/registry \ registry:2
   ```

3. **Add TLS security** (recommended):

   1. Generate self-signed certificates:

      ```
      mkdir -p certs openssl req -newkey rsa:4096 -nodes -sha256 -keyout
      certs/domain.key \ -x509 -days 365 -out certs/domain.crt
      ```

   2. Run the registry with TLS:

      ```
      docker run -d -p 5000:5000 --restart=always --name registry \ -v /
      mnt/registry:/var/lib/registry \ -v $(pwd)/certs:/certs \ -e
      REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \ -e
      REGISTRY_HTTP_TLS_KEY=/certs/domain.key \ registry:2
      ```

Harbor registry

For more advanced features, consider using [Harbor (https://github.com/goharbor/harbor)](https://github.com/goharbor/harbor).

1. **Download Harbor offline installer** before going air-gapped:

   Download from https://github.com/goharbor/harbor/releases

2. **Extract and configure**:

   ```
   tar xzvf harbor-offline-installer-v2.x.x.tgz cd harbor cp harbor.yml.tmpl harbor.yml # Edit harbor.yml to configure settings
   ```

3. **Install Harbor**:

   ```
   ./install.sh --with-trivy
   ```

4. **Access Harbor** at `https://harbor-hostname` (based on your configuration)

Populate your private registry

If the private registry cannot access the public internet, retrieve images on an external machine and securely transfer them using approved data transfer methods.

```
# Pull the required Mattermost images docker pull mattermost/mattermost-enterprise-edition:latest # Tag the images for your private registry docker tag mattermost/mattermost-enterprise-edition:latest registry.example.com:5000/mattermost/mattermost-enterprise-edition:latest # Push to your private registry docker push registry.example.com:5000/mattermost/mattermost-enterprise-edition:latest
```

Configure Kubernetes to use private image registries

When using Kubernetes in an air-gapped environment, you need to configure it to use your private registry.

Create registry credentials

1. **Create a kubernetes secret for registry authentication**:

```
kubectl create secret docker-registry regcred \  --docker-server=registry.example.com:5000 \  --docker-username=your_username \  --docker-password=your_password \  --docker-email=your_email@example.com
```

2. **Reference the secret in pod specifications**:

```
apiVersion: v1 kind: Pod metadata:  name: mattermost-pod spec: containers:  - name: mattermost  image: registry.example.com:5000/mattermost/mattermost-enterprise-edition:latest  imagePullSecrets:  - name: regcred
```

3. **For Helm deployments**, specify the registry in `values.yaml`:

```
image:  repository: registry.example.com:5000/mattermost/mattermost-enterprise-edition  tag: latest  pullPolicy: IfNotPresent imagePullSecrets:  - name: regcred
```

Configure Docker to use private image registries

Configure Docker on all hosts to trust and use your private registry.

Docker daemon configuration

1. **Add your registry to Docker's trusted registries**:

   Edit or create `/etc/docker/daemon.json`:

```
{  "insecure-registries": ["registry.example.com:5000"] }
```

   For registries using self-signed certificates:

```
mkdir -p /etc/docker/certs.d/registry.example.com:5000 cp domain.crt /etc/docker/certs.d/registry.example.com:5000/ca.crt
```

2. **Restart Docker daemon**:

```
systemctl restart docker
```

3. **Test the configuration**:

```
docker pull registry.example.com:5000/mattermost/mattermost-enterprise-
edition:latest
```

Set up a private Debian package mirror

A local Debian mirror allows you to maintain packages for system updates and dependencies. In this case, the mirror will be used to provide packages for Mattermost server to debian-based hosts.

Ideally the air-gapped network already has a local mirror available. If not, you can set up a local mirror such as Aptly or debmirror.

Use Aptly

1. **Install Aptly** (on an internet-connected machine):

```
apt-get update apt-get install aptly gnupg
```

2. **Create GPG key for signing packages**:

```
gpg --gen-key
```

3. **Create a mirror configuration**:

```
aptly mirror create -architectures=amd64 debian-bullseye http://
deb.debian.org/debian bullseye main contrib non-free
```

4. **Update the mirror to download packages**:

```
aptly mirror update debian-bullseye
```

5. **Create and publish a snapshot**:

```
aptly snapshot create debian-bullseye-$(date +%Y%m%d) from mirror debian-
bullseye aptly publish snapshot debian-bullseye-$(date +%Y%m%d)
```

6. **Serve the repository**:

```
aptly serve
```

Use debmirror

For a simpler approach:

1. **Install debmirror**:

```
apt-get install debmirror
```

2. **Create a mirror script**:

```
#!/bin/bash debmirror --host=deb.debian.org \  --root=/debian \  --
method=http \  --dist=bullseye \  --section=main,contrib,non-free \  --
arch=amd64 \  --nosource \  --progress \  --ignore-release-gpg \  /path/
to/mirror/debian
```

3. **Set up a web server** (like nginx) to serve the mirror:

```
apt-get install nginx # Create nginx configuration cat > /etc/nginx/
sites-available/debian-mirror << EOF server {  listen 80;  server_name
mirror.example.com;  root /path/to/mirror;  autoindex on; } EOF ln -s /
etc/nginx/sites-available/debian-mirror /etc/nginx/sites-enabled/
systemctl restart nginx
```

Client configuration

On air-gapped systems, configure apt to use your local mirror:

```
cat > /etc/apt/sources.list << EOF deb http://mirror.example.com/debian
bullseye main contrib non-free EOF
```

Set up a private RHEL package mirror

For Red Hat Enterprise Linux environments, you'll need a local repository mirror.

Ideally the air-gapped network already has a local mirror available. If not, you can set up a local mirror such as reposync.

Use reposync

1. **Install required tools** (on an internet-connected RHEL system):

```
yum install yum-utils createrepo
```

2. **Download packages**:

```
mkdir -p /var/www/html/repos/rhel8 reposync -p /var/www/html/repos/rhel8
--download-metadata --repo=rhel-8-for-x86_64-baseos-rpms reposync -p /
var/www/html/repos/rhel8 --download-metadata --repo=rhel-8-for-x86_64-
appstream-rpms
```

3. **Create repository metadata**:

```
createrepo /var/www/html/repos/rhel8/rhel-8-for-x86_64-baseos-rpms
createrepo /var/www/html/repos/rhel8/rhel-8-for-x86_64-appstream-rpms
```

4. **Set up a web server**:

```
yum install httpd systemctl enable httpd systemctl start httpd
```

Client configuration

On air-gapped RHEL systems:

1. **Disable existing repositories**:

```
cd /etc/yum.repos.d/ mkdir backup mv *.repo backup/
```

2. **Create new repository files**:

```
cat > /etc/yum.repos.d/local-baseos.repo << EOF [local-baseos] name=Red Hat Enterprise Linux 8 BaseOS baseurl=http://mirror.example.com/repos/rhel8/rhel-8-for-x86_64-baseos-rpms enabled=1 gpgcheck=0 EOF cat > /etc/yum.repos.d/local-appstream.repo << EOF [local-appstream] name=Red Hat Enterprise Linux 8 AppStream baseurl=http://mirror.example.com/repos/rhel8/rhel-8-for-x86_64-appstream-rpms enabled=1 gpgcheck=0 EOF
```

3. **Clear cache and test**:

```
yum clean all yum repolist
```

Mattermost server configuration for air-gapped deployments

When deploying Mattermost in an air-gapped environment, there are configuration options available to accommodate the lack of internet access. The following covers these configuration options and offers recommendations for settings.

Mobile push notifications

Mattermost can use mobile push notifications to notify users of new messages and activity. These notifications require a server component to be deployed to send the notifications to the mobile devices. By default, Mattermost will use the public push notification service which is not available in an air-gapped environment. We recommend disabling push notifications in **System Console > Environment > Push Notification Server**.

Website link previews

Website link previews require a connection to the internet to fetch the content of the links. We recommend disabling website link previews in **System Console > Site Configuration > Posts**.

Additional considerations

Mattermost server plugins may require external connectivity to function. These will need to be reviewed on a case-by-case basis to determine if they provide useful functionality with no external dependencies.

Remember that air-gapped environments require ongoing maintenance to stay secure and up-to-date. Regular updates to the Mattermost server and other components are required to ensure the environment remains secure and up-to-date.

Network security

In air-gapped environments, network security is critical:

1. **Implement strict firewall rules** to control traffic between network segments.
2. **Use network segmentation** to isolate critical infrastructure components.
3. **Regularly audit network access** to ensure the environment remains properly isolated.

Transfer data to air-gapped networks

For initial setup and updates:

1. **Use approved data diodes** or one-way transfer devices.
2. **Implement strict media control** for any physical media entering the air-gapped environment.
3. **Scan all incoming data** for malware before allowing it into the environment.

Keep systems updated

Develop a process for regular updates:

1. **Schedule periodic updates** to your local mirrors.
2. **Maintain a consistent testing process** for all updates before deployment.

3. **Keep comprehensive documentation** of all packages and versions in use.

Monitoring and logging

Ensure robust monitoring within the air-gapped environment:

1. **Deploy local monitoring solutions** that don't require internet access.
2. **Establish baselines** for normal system behavior.
3. **Implement centralized logging** for security analysis and troubleshooting.

On this page

- Deploy Mattermost in Air-Gapped Environments
- Deploy Mattermost in Air-Gapped Environments
  - Set up a self-hosted private container registry
    - Docker registry
    - Harbor registry
    - Populate your private registry
  - Configure Kubernetes to use private image registries
    - Create registry credentials
  - Configure Docker to use private image registries
    - Docker daemon configuration
  - Set up a private Debian package mirror
    - Use Aptly
    - Use debmirror
    - Client configuration
  - Set up a private RHEL package mirror
    - Use reposync
    - Client configuration
  - Mattermost server configuration for air-gapped deployments
    - Mobile push notifications
    - Website link previews

- Additional considerations
  - Network security
  - Transfer data to air-gapped networks
  - Keep systems updated
  - Monitoring and logging
- Set up a self-hosted private container registry
  - Populate your private registry
- Configure Kubernetes to use private image registries
  - Create registry credentials
- Configure Docker to use private image registries
  - Docker daemon configuration
- Set up a private Debian package mirror
  - Client configuration
- Set up a private RHEL package mirror
  - Use reposync
  - Client configuration
- Mattermost server configuration for air-gapped deployments
  - Mobile push notifications
  - Website link previews
- Additional considerations
  - Network security
  - Transfer data to air-gapped networks
  - Keep systems updated
  - Monitoring and logging

Enable Email (Recommended)

You can deploy Mattermost server in **Preview Mode** on AWS Elastic Beanstalk using a Docker container. This is a great option for exploring functionality, testing, and development purposes, as it allows you to quickly set up a Mattermost instance without needing to manage the underlying infrastructure. This deployment method shouldn't be used in production environments.

The Elastic Beanstalk application creation process below combines Application and Environment creation in a single workflow:

1. From your AWS console (https://console.aws.amazon.com/console/home) select **Elastic Beanstalk**.
2. Select the **Create Application** button on the Elastic Beanstalk home page.
3. Enter an **Application name** for the Elastic Beanstalk application.
4. (Optional) Specify **Application tags**.
5. Select **Docker** for the Platform.
6. Choose **Docker running on 64bit Amazon Linux 2** for the **Platform** branch. Note that multi-container Docker is now deprecated.
7. Leave the platform version preselected as **3.4.8 (Recommended)**.
8. For **Application code**, select **Upload your code**.
9. Download the `Dockerrun.aws.json` file from https://raw.githubusercontent.com/mattermost/mattermost-docker-preview/master/Dockerrun.aws.json.
10. Set a unique **Version Label**, choose **Local File**, then select **Choose file** button to browse for the downloaded file in the previous step. You should see a **File successfully uploaded** message.
11. Select **Create Application**. It may take a few minutes for Beanstalk to launch your environment. If the launch is successful, you will see a see a large green checkmark and the **Health status** displayed in green.

12. Test your environment by selecting the domain link next to your application name at the top of the dashboard. Alternatively, enter the domain into your browser in the form of `http://<your-ebs-application-url>.elasticbeanstalk.com`. You can also map your own domain if you prefer. When everything is working correctly, the domain navigates you to the Mattermost Login page. Enjoy exploring Mattermost!

The default Docker instance for Mattermost is designed for product evaluation, and sets `SendEmailNotifications=false` so the product can function without enabling email. To see the product's full functionality, we recommend enabling SMTP email.

See Configuration Settings documentation for more configuration and customization options for your deployment.

Step 1: Install Docker

This guide provides step-by-step instructions for deploying Mattermost using Docker containers.

> Warning
>
> - Mattermost server deployment using Docker is officially supported on Linux operating systems only.
> - macOS and Windows Docker deployments are supported for testing and development purposes only.
> - Docker is not ideal for High Availability (HA) because it lacks key features including automatic failover, shared storage, and load balancing. Docker also has challenges managing multiple nodes and recovering from failures. We recommend deploying on Kubernetes for HA for these features to ensure reliability.

If you don't have Docker installed, follow the instructions below based on your operating system. You'll need Docker Engine (https://docs.docker.com/engine/install/) and Docker Compose (https://docs.docker.com/compose/install/) (release 1.28 or later).

Step 2: Deploy Mattermost on Docker

This section provides a quick start guide for deploying Mattermost on Docker by leveraging Docker Compose (https://docs.docker.com/compose/install/).

> Note
>
> - The deployment configuration results in two separate containers: a container for the database and a container for the application. An optional third container results when using NGINX for reverse proxy.
> - Encountering issues with your Docker deployment? See the Docker deployment troubleshooting documentation for details.

1. In a terminal window, clone the repository and enter the directory.

```
git clone https://github.com/mattermost/docker cd docker
```

2. Create your `.env` file by copying and adjusting the `env.example` file.

```
cp env.example .env
```

   Important

   - At a minimum, you must edit the `DOMAIN` value in the `.env` file to correspond to the domain for your Mattermost server.
   - We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) via `MM_SUPPORTSETTINGS_SUPPORTEMAIL`. This is the email address your users will contact when they need help.

3. Create the required directories and set their permissions.

```
mkdir -p ./volumes/app/mattermost/{config,data,logs,plugins,client/
plugins,bleve-indexes} sudo chown -R 2000:2000 ./volumes/app/mattermost
```

4. *(Optional)* Configure TLS for NGINX. If you're not using the included NGINX reverse proxy, you can skip this step.

Create a new certificate and key

```
bash scripts/issue-certificate.sh -d <YOUR_MM_DOMAIN> -o ${PWD}/certs
```

To include the certificate and key, uncomment the following lines in your `.env` file and ensure they point to the appropriate files.

```
#CERT_PATH=./certs/etc/letsencrypt/live/${DOMAIN}/fullchain.pem
#KEY_PATH=./certs/etc/letsencrypt/live/${DOMAIN}/privkey.pem
```

Use a pre-existing certificate and key

```
mkdir -p ./volumes/web/cert cp <PATH-TO-PRE-EXISTING-CERT>.pem ./volumes/
web/cert/cert.pem cp <PATH-TO-PRE-EXISTING-KEY>.pem ./volumes/web/cert/key-
no-password.pem
```

To include the certificate and key, ensure the following lines in your `.env` file points to the appropriate files.

```
CERT_PATH=./volumes/web/cert/cert.pem KEY_PATH=./volumes/web/cert/key-no-
password.pem
```

5. Deploy Mattermost.

Deploy without using the included NGINX

```
docker compose -f docker-compose.yml -f docker-compose.without-nginx.yml up
-d
```

To access your new Mattermost deployment, navigate to `http://<YOUR_MM_DOMAIN>:8065/` in your browser.

To shut down your deployment:

```
docker compose -f docker-compose.yml -f docker-compose.without-nginx.yml
down
```

Deploy using the included NGINX

```
docker compose -f docker-compose.yml -f docker-compose.nginx.yml up -d
```

To access your new Mattermost deployment via HTTPS, navigate to `https://<YOUR_MM_DOMAIN>/` in your browser.

To shut down your deployment:

```
docker compose -f docker-compose.yml -f docker-compose.nginx.yml down
```

6. Create your first Mattermost system admin user, invite more users, and explore the Mattermost platform.

Configure SSO With GitLab (Optional)

To use SSO with GitLab with a self-signed certificate, you have to add the PKI chain for your authority. This is required to avoid the `Token request failed: certificate signed by unknown authority` error.

To add the PKI chain, uncomment the following line in your `.env` file, and ensure it points to your `pki_chain.pem` file:

```
#GITLAB_PKI_CHAIN_PATH=<path_to_your_gitlab_pki>/pki_chain.pem
```

Then uncomment the following line in your `docker-compose.yml` file, and ensure it points to the same `pki_chain.pem` file:

```
# - ${GITLAB_PKI_CHAIN_PATH}:/etc/ssl/certs/pki_chain.pem:ro
```

Upgrade from mattermost-docker

The [mattermost-docker (https://github.com/mattermost/mattermost-docker)](https://github.com/mattermost/mattermost-docker) GitHub repository is deprecated. Visit the [mattermost/docker (https://github.com/mattermost/docker)](https://github.com/mattermost/docker) GitHub repository to access the official Docker deployment solution for Mattermost.

To migrate from an existing `mattermost/mattermost-prod-app` image, we recommend migrating to either `mattermost/mattermost-enterprise-edition` or `mattermost/mattermost-team-edition` images, which are the official images supported by Mattermost. These images support PostgreSQL v11+ databases,

which we know has been a long-running challenge for the community, and you will not lose any features or functionality by moving to these new images.

For additional help or questions, please refer to [this issue (https://github.com/mattermost/mattermost-docker/issues/489)](https://github.com/mattermost/mattermost-docker/issues/489).

Install a different version of Mattermost

1. Shut down your deployment.
2. Run `git pull` to fetch any recent changes to the repository, paying attention to any potential `env.example` changes.
3. Adjust the `MATTERMOST_IMAGE_TAG` in the `.env` file to point your desired [enterprise (https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated) or [team (https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated) image version.

   Important

   For production environments, we recommend using specific version tags such as `MATTERMOST_IMAGE_TAG=release-10.5` rather than generic tags like `MATTERMOST_IMAGE_TAG=release-10`. Generic `release-x` tags are intended for development use only and do not automatically receive new patch releases within that major version. Using specific version tags ensures a more reproducible and deterministic environment for your production deployment.

4. Redeploy Mattermost.

Troubleshooting

Troubleshooting your Docker deployment

If deploying on an M1 Mac and encountering permission issues in the Docker container, redo the third step and skip this command:

```
sudo chown -R 2000:2000 ./volumes/app/mattermost
```

If having issues deploying on Docker generally, ensure the docker daemon is enabled and running:

```
sudo systemctl enable --now docker
```

To remove all data and settings for your Mattermost deployment:

```
sudo rm -rf ./volumes
```

Troubleshooting PostgreSQL

For quick start deployments, you can change the Postgres username and/or password (recommended) in the `.env` file. If your database is managed externally, you'll need to change the password in your database management tool. Then, update the `.env` file with the new credentials.

Troubleshooting TLS & NGINX

For an in-depth guide to configuring the TLS certificate and key for Nginx, please refer to this document in the repository (https://github.com/mattermost/docker/blob/main/docs/issuing-letsencrypt-certificate.md).

Trial Mattermost using Docker Preview

Looking for a way to evaluate Mattermost on a single local machine using Docker? We recommend using the Mattermost Docker Preview Image (https://github.com/mattermost/mattermost-docker-preview) to install Mattermost in Preview Mode.

Important

- This local image is self-contained (i.e., it has an internal database and works out of the box). Dropping a container using this image removes data and configuration as expected. You can see the configuration settings documentation to learn more about customizing your trial deployment.
- **Preview Mode** shouldn't be used in a production environment, as it uses a known password string, contains other non-production configuration settings, has email disabled, keeps no persistent data (all data lives inside the container), and doesn't support upgrades.
- If you are planning to use the calling functionality in **Preview Mode** on a non-local environment, you should ensure that the server is running on a secure (HTTPs) connection and that the network requirements to run calls are met.

1. Install Docker (https://www.docker.com/get-started/).
2. Once you have Docker, run the following command in a terminal window:

```
docker run --name mattermost-preview -d --publish 8065:8065 --publish
8443:8443 mattermost/mattermost-preview
```

3. When Docker is done fetching the image, navigate to `http://localhost:8065/` in your browser to preview Mattermost.

4. Select **Don't have an account** in the top right corner of the screen to create an account for your preview instance. If you don't see this option, ensure that the Enable open server configuration setting is enabled. This setting is disabled for self-hosted Mattermost deployments by default.

5. Log in to your preview instance with your user credentials.

Troubleshooting your preview deployment

The **Preview Mode** Docker instance for Mattermost is designed for product evaluation, and sets `SendEmailNotifications=false` so the product can function without enabling email. See the Configuration Settings documentation to customize your deployment.

To update your Mattermost preview image and container, you must first stop and delete your existing **mattermost-preview** container by running the following commands:

```
docker pull mattermost/mattermost-preview docker stop mattermost-preview
docker rm mattermost-preview
```

Once the new image is pulled and the container is stopped and deleted you need to run the `docker run` command from above.

Important

On Linux, include `sudo` in front of all `docker` commands.

To access a shell inside the container, run the following command:

```
docker exec -ti mattermost-preview /bin/bash
```

> Note
>
> See the deployment troubleshooting documentation for resolutions to common deployment issues.

Deploy Mattermost using Containers

You can deploy Mattermost Server using container technologies for exploring functionality, testing, and development purposes, as it allows you to quickly set up a Mattermost instance without needing to manage the underlying infrastructure. This deployment method shouldn't be used in production environments as it doesn't support clustered deployments or High Availability (HA) configurations out-of-the-box.

Choose your preferred container platform below for specific deployment instructions:

Docker

This guide provides step-by-step instructions for deploying Mattermost using Docker containers.

> Warning
>
> - Mattermost server deployment using Docker is officially supported on Linux operating systems only.
> - macOS and Windows Docker deployments are supported for testing and development purposes only.
> - Docker is not ideal for High Availability (HA) because it lacks key features including automatic failover, shared storage, and load balancing. Docker also has challenges managing multiple nodes and recovering from failures. We recommend deploying on Kubernetes for HA for these features to ensure reliability.

Step 1: Install Docker

If you don't have Docker installed, follow the instructions below based on your operating system. You'll need Docker Engine (https://docs.docker.com/engine/install/) and Docker Compose (https://docs.docker.com/compose/install/) (release 1.28 or later).

Step 2: Deploy Mattermost on Docker

This section provides a quick start guide for deploying Mattermost on Docker by leveraging Docker Compose (https://docs.docker.com/compose/install/).

> Note
>
> - The deployment configuration results in two separate containers: a container for the database and a container for the application. An optional third container results when using NGINX for reverse proxy.
> - Encountering issues with your Docker deployment? See the Docker deployment troubleshooting documentation for details.

1. In a terminal window, clone the repository and enter the directory.

   ```
   git clone https://github.com/mattermost/docker cd docker
   ```

2. Create your `.env` file by copying and adjusting the `env.example` file.

   ```
   cp env.example .env
   ```

   Important

   - At a minimum, you must edit the `DOMAIN` value in the `.env` file to correspond to the domain for your Mattermost server.
   - We recommend configuring the Support Email (https://docs.mattermost.com/administration/config-settings.html#support-

email) via `MM_SUPPORTSETTINGS_SUPPORTEMAIL`. This is the email address your users will contact when they need help.

3. Create the required directories and set their permissions.

```
mkdir -p ./volumes/app/mattermost/{config,data,logs,plugins,client/
plugins,bleve-indexes} sudo chown -R 2000:2000 ./volumes/app/mattermost
```

4. *(Optional)* Configure TLS for NGINX. If you're not using the included NGINX reverse proxy, you can skip this step.

Create a new certificate and key

```
bash scripts/issue-certificate.sh -d <YOUR_MM_DOMAIN> -o ${PWD}/certs
```

To include the certificate and key, uncomment the following lines in your `.env` file and ensure they point to the appropriate files.

```
#CERT_PATH=./certs/etc/letsencrypt/live/${DOMAIN}/fullchain.pem
#KEY_PATH=./certs/etc/letsencrypt/live/${DOMAIN}/privkey.pem
```

Use a pre-existing certificate and key

```
mkdir -p ./volumes/web/cert cp <PATH-TO-PRE-EXISTING-CERT>.pem ./volumes/
web/cert/cert.pem cp <PATH-TO-PRE-EXISTING-KEY>.pem ./volumes/web/cert/key-
no-password.pem
```

To include the certificate and key, ensure the following lines in your `.env` file points to the appropriate files.

```
CERT_PATH=./volumes/web/cert/cert.pem KEY_PATH=./volumes/web/cert/key-no-
password.pem
```

5. Deploy Mattermost.

Deploy without using the included NGINX

```
docker compose -f docker-compose.yml -f docker-compose.without-nginx.yml up
-d
```

To access your new Mattermost deployment, navigate to `http://<YOUR_MM_DOMAIN>:8065/` in your browser.

To shut down your deployment:

> ```
> docker compose -f docker-compose.yml -f docker-compose.without-nginx.yml
> down
> ```

Deploy using the included NGINX

> ```
> docker compose -f docker-compose.yml -f docker-compose.nginx.yml up -d
> ```

To access your new Mattermost deployment via HTTPS, navigate to `https://<YOUR_MM_DOMAIN>/` in your browser.

To shut down your deployment:

> ```
> docker compose -f docker-compose.yml -f docker-compose.nginx.yml down
> ```

6. Create your first Mattermost system admin user, invite more users, and explore the Mattermost platform.

Configure SSO With GitLab (Optional)

To use SSO with GitLab with a self-signed certificate, you have to add the PKI chain for your authority. This is required to avoid the `Token request failed: certificate signed by unknown authority` error.

To add the PKI chain, uncomment the following line in your `.env` file, and ensure it points to your `pki_chain.pem` file:

```
#GITLAB_PKI_CHAIN_PATH=<path_to_your_gitlab_pki>/pki_chain.pem
```

Then uncomment the following line in your `docker-compose.yml` file, and ensure it points to the same `pki_chain.pem` file:

```
# - ${GITLAB_PKI_CHAIN_PATH}:/etc/ssl/certs/pki_chain.pem:ro
```

Upgrade from mattermost-docker

The [mattermost-docker (https://github.com/mattermost/mattermost-docker)](https://github.com/mattermost/mattermost-docker) GitHub repository is deprecated. Visit the [mattermost/docker (https://github.com/mattermost/docker)](https://github.com/mattermost/docker) GitHub repository to access the official Docker deployment solution for Mattermost.

To migrate from an existing `mattermost/mattermost-prod-app` image, we recommend migrating to either `mattermost/mattermost-enterprise-edition` or `mattermost/mattermost-team-edition` images, which are the official images supported by Mattermost. These images support PostgreSQL v11+ databases, which we know has been a long-running challenge for the community, and you will not lose any features or functionality by moving to these new images.

For additional help or questions, please refer to [this issue (https://github.com/mattermost/mattermost-docker/issues/489)](https://github.com/mattermost/mattermost-docker/issues/489).

Install a different version of Mattermost

1. Shut down your deployment.
2. Run `git pull` to fetch any recent changes to the repository, paying attention to any potential `env.example` changes.
3. Adjust the `MATTERMOST_IMAGE_TAG` in the `.env` file to point your desired [enterprise (https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated) or [team (https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated) image version.

Important

For production environments, we recommend using specific version tags such as `MATTERMOST_IMAGE_TAG=release-10.5` rather than generic tags like `MATTERMOST_IMAGE_TAG=release-10`. Generic `release-x` tags are intended for development use only and do not automatically receive new patch releases within that major version. Using specific version tags ensures a more reproducible and deterministic environment for your production deployment.

4. Redeploy Mattermost.

Troubleshooting

Troubleshooting your Docker deployment

If deploying on an M1 Mac and encountering permission issues in the Docker container, redo the third step and skip this command:

```
sudo chown -R 2000:2000 ./volumes/app/mattermost
```

If having issues deploying on Docker generally, ensure the docker daemon is enabled and running:

```
sudo systemctl enable --now docker
```

To remove all data and settings for your Mattermost deployment:

```
sudo rm -rf ./volumes
```

Troubleshooting PostgreSQL

For quick start deployments, you can change the Postgres username and/or password (recommended) in the `.env` file. If your database is managed externally, you'll need to change the password in your database management tool. Then, update the `.env` file with the new credentials.

Troubleshooting TLS & NGINX

For an in-depth guide to configuring the TLS certificate and key for Nginx, please refer to this document in the repository (https://github.com/mattermost/docker/blob/main/docs/issuing-letsencrypt-certificate.md).

Trial Mattermost using Docker Preview

Looking for a way to evaluate Mattermost on a single local machine using Docker? We recommend using the Mattermost Docker Preview Image (https://github.com/mattermost/mattermost-docker-preview) to install Mattermost in Preview Mode.

Important

- This local image is self-contained (i.e., it has an internal database and works out of the box). Dropping a container using this image removes data and configuration as expected. You can see the configuration settings documentation to learn more about customizing your trial deployment.
- **Preview Mode** shouldn't be used in a production environment, as it uses a known password string, contains other non-production configuration settings, has email disabled, keeps no persistent data (all data lives inside the container), and doesn't support upgrades.
- If you are planning to use the calling functionality in **Preview Mode** on a non-local environment, you should ensure that the server is running on a secure (HTTPs) connection and that the network requirements to run calls are met.

1. Install Docker (https://www.docker.com/get-started/).
2. Once you have Docker, run the following command in a terminal window:

```
docker run --name mattermost-preview -d --publish 8065:8065 --publish
8443:8443 mattermost/mattermost-preview
```

3. When Docker is done fetching the image, navigate to `http://localhost:8065/` in your browser to preview Mattermost.

4. Select **Don't have an account** in the top right corner of the screen to create an account for your preview instance. If you don't see this option, ensure that the Enable open server configuration setting is enabled. This setting is disabled for self-hosted Mattermost deployments by default.

5. Log in to your preview instance with your user credentials.

Troubleshooting your preview deployment

The **Preview Mode** Docker instance for Mattermost is designed for product evaluation, and sets `SendEmailNotifications=false` so the product can function without enabling email. See the Configuration Settings documentation to customize your deployment.

To update your Mattermost preview image and container, you must first stop and delete your existing **mattermost-preview** container by running the following commands:

```
docker pull mattermost/mattermost-preview docker stop mattermost-preview
docker rm mattermost-preview
```

Once the new image is pulled and the container is stopped and deleted you need to run the `docker run` command from above.

Important

On Linux, include `sudo` in front of all `docker` commands.

To access a shell inside the container, run the following command:

```
docker exec -ti mattermost-preview /bin/bash
```

> Note
>
> See the deployment troubleshooting documentation for resolutions to common deployment issues.

AWS Elastic Beanstalk

You can deploy Mattermost server in **Preview Mode** on AWS Elastic Beanstalk using a Docker container. This is a great option for exploring functionality, testing, and development purposes, as it allows you to quickly set up a Mattermost instance without needing to manage the underlying infrastructure. This deployment method shouldn't be used in production environments.

The Elastic Beanstalk application creation process below combines Application and Environment creation in a single workflow:

1. From your [AWS console (https://console.aws.amazon.com/console/home)](https://console.aws.amazon.com/console/home) select **Elastic Beanstalk**.
2. Select the **Create Application** button on the Elastic Beanstalk home page.
3. Enter an **Application name** for the Elastic Beanstalk application.
4. (Optional) Specify **Application tags**.
5. Select **Docker** for the Platform.
6. Choose **Docker running on 64bit Amazon Linux 2** for the **Platform** branch. Note that multi-container Docker is now deprecated.
7. Leave the platform version preselected as **3.4.8 (Recommended)**.
8. For **Application code**, select **Upload your code**.
9. Download the `Dockerrun.aws.json` file from https://raw.githubusercontent.com/mattermost/mattermost-docker-preview/master/Dockerrun.aws.json.
10. Set a unique **Version Label**, choose **Local File**, then select **Choose file** button to browse for the downloaded file in the previous step. You should see a **File successfully uploaded** message.
11. Select **Create Application**. It may take a few minutes for Beanstalk to launch your environment. If the launch is successful, you will see a see a large green checkmark and the **Health status** displayed in green.
12. Test your environment by selecting the domain link next to your application name at the top of the dashboard. Alternatively, enter the domain into your browser in the form of `http://<your-ebs-application-url>.elasticbeanstalk.com`. You can also map your own domain if you prefer.

When everything is working correctly, the domain navigates you to the Mattermost Login page. Enjoy exploring Mattermost!

Enable Email (Recommended)

The default Docker instance for Mattermost is designed for product evaluation, and sets `SendEmailNotifications=false` so the product can function without enabling email. To see the product's full functionality, we recommend enabling SMTP email.

See Configuration Settings documentation for more configuration and customization options for your deployment.

Secure your Mattermost deployment

Deploying Mattermost using Docker containers can be made secure with proper configurations for HTTPS and reverse proxying. This guide outlines the steps to set up TLS and an NGINX reverse proxy for your Mattermost deployment, ensuring secure communication between users and your server.

1. Set Up an NGINX Container to serve as the reverse proxy. You can use NGINX either as a separate container or installed on the host machine.

2. Bind Volumes for NGINX Configuration and TLS Certificates:

> - *Bind Docker volumes for NGINX configuration files and TLS certificates to ensure persistent and secure storage of these assets.*
> - *Use permission restrictions on host directories where sensitive files such as TLS keys are stored.*

3. Create the NGINX Configuration File by designing a robust `nginx.conf` file to configure reverse proxying and HTTPS. Here's a basic example:

```
server {  listen 443 ssl;  server_name your-domain.com;
ssl_certificate /etc/nginx/certs/fullchain.pem;  ssl_certificate_key /
etc/nginx/certs/privkey.pem;  ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers HIGH:!aNULL:!MD5;  location / {  proxy_pass http://
mattermost:8065;  proxy_set_header Host $host;  proxy_set_header X-Real-
IP $remote_addr;  proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;  proxy_set_header X-Forwarded-Proto
https;  } }
```

Verify the configuration with `nginx -t` before applying.

4. Obtain TLS Certificates:

> - *Use Let's Encrypt for free, automated certificates. Tools like Certbot can help automate the process.*
> - *Alternatively, purchase certificates from a trusted certificate authority (CA) and ensure proper setup of intermediate and root certificate chains.*
>
> *Keep private keys secure and avoid storing them directly inside Docker images.*

5. Connect Containers Using Docker Networking:

> - *Use Docker's networking features to isolate and link containers.*
> - *Create a custom Docker bridge network to ensure secure communication. For example:*
>
> ```
> docker network create mattermost-network
> ```
>
> - *Launch the Mattermost and NGINX containers on the same network:*
>
> ```
> docker network connect mattermost-network mattermost docker
> network connect mattermost-network nginx
> ```

6. Point your domain to the server IP address:

   Ensure your domain (e.g., your-domain.com) points to the public IP address of your server. If your IP is dynamic, consider setting up Dynamic DNS (DDNS) for seamless connectivity.

7. After placing the certificates and updating the configuration, restart the NGINX container:
8. Use logs (docker logs nginx) to troubleshoot and validate the container's operation.
9. Verify HTTPS Access by visiting `https://your-domain.com` in a web browser to confirm Mattermost is running securely over HTTPS.
10. Use tools such as SSL Labs : https://www.ssllabs.com/ssltest/ to validate the quality of your TLS setup.
11. Enable HTTP Strict Transport Security (HSTS) in your NGINX configuration to prevent downgrade attacks.
12. Use NGINX rate-limiting features to restrict abusive traffic, such as excessive requests:

Additionally, consider:

- Use Docker's security features such as Seccomp profiles and AppArmor to secure your container runtime.
- Avoid running containers with elevated privileges `--privileged` and utilize user namespaces.
- Always use trusted images (e.g., official NGINX and Mattermost images) to prevent exposure to vulnerabilities in third-party images.
- Update Mattermost, NGINX, and Docker to their latest versions regularly to ensure patches for known vulnerabilities are applied.
- Set up proper firewall rules to restrict unauthorized access and monitor traffic using tools like Fail2Ban or Wazuh.

By following these steps, your Mattermost deployment using Docker containers will be accessible securely over HTTPS with efficient proxying through NGINX. Implementing the additional security recommendations will further protect your environment against evolving threats.

On this page

- Deploy Mattermost using Containers
- Deploy Mattermost using Containers

- Docker
  - Step 1: Install Docker
  - Step 2: Deploy Mattermost on Docker
  - Create a new certificate and key
  - Use a pre-existing certificate and key
  - Deploy without using the included NGINX
  - Deploy using the included NGINX
  - Configure SSO With GitLab (Optional)
  - Upgrade from mattermost-docker
  - Install a different version of Mattermost
  - Troubleshooting
    - Troubleshooting your Docker deployment
    - Troubleshooting PostgreSQL
  - Trial Mattermost using Docker Preview
    - Troubleshooting your preview deployment
- AWS Elastic Beanstalk
  - Enable Email (Recommended)
- Secure your Mattermost deployment
- Secure your Mattermost deployment

Deploy Mattermost on Kubernetes

Mattermost server can be deployed on various Kubernetes platforms, providing a scalable and robust infrastructure for your team communication needs. This guide covers deployment options for major cloud providers and general Kubernetes installations.

> Tip
>
> To learn how to safely upgrade your deployment in Kubernetes for High Availability with Active/Active support, see the Upgrading Mattermost in Kubernetes and High Availability Environments documenation.

Platform

Choose your preferred platform below for specific deployment instructions:

Mattermost Operator

You can use the Mattermost Kubernetes Operator to deploy Mattermost on Kubernetes using S3-compatible storage and a managed database service. While the operator supports a range of configurations, we strongly recommend using a cloud-native approach for production environments.

Prerequisites

Before you begin, ensure you have the following:

- A functioning Kubernetes cluster (see the Kubernetes setup guide (https://kubernetes.io/docs/setup/)). Your cluster should be running a supported Kubernetes version (https://kubernetes.io/releases/).
- The *kubectl* command-line tool installed on your local machine (see the kubectl installation guide (https://kubernetes.io/docs/reference/kubectl/)).

- A fundamental understanding of Kubernetes concepts, such as deployments, pods, and applying manifests.
- Sufficient Kubernetes resources allocated based on your expected user load. Consult the scaling for Enterprise documentation for resource requirements at different scales.

Installation steps

The installation process involves setting up necessary operators and then deploying Mattermost itself.

Step 1: Install the NGINX Ingress Controller

Follow the instructions in the Kubernetes deployment documentation to install the NGINX ingress controller on your Kubernetes cluster. Mattermost recommends installing the Nginx Operator via helm, regardless of platform you are installing to.

Step 2: Install the Mattermost Operator

The Mattermost Kubernetes Operator can be installed using Helm.

1. Install Helm (version 3.13.0 or later). See the Helm quickstart documentation for installation instructions.
2. Add the Mattermost Helm repository:

```
helm repo add mattermost https://helm.mattermost.com
```

3. Create a file named `config.yaml` and populate it with the contents of the [Mattermost operator values file (https://github.com/mattermost/mattermost-helm/blob/master/charts/mattermost-operator/values.yaml)](https://github.com/mattermost/mattermost-helm/blob/master/charts/mattermost-operator/values.yaml). This file allows for customization of the operator.
4. Create a namespace for the Mattermost Operator:

```
kubectl create ns mattermost-operator
```

5. Install the Mattermost Operator. If you don't specify a version, the latest version of the Mattermost Operator will be installed.

```
helm install <your-release-name> mattermost/mattermost-operator -n
<namespace_name>
```

For example:

```
helm install mattermost-operator mattermost/mattermost-operator -n
mattermost-operator
```

To use your custom `config.yaml` *file:*

```
helm install mattermost-operator mattermost/mattermost-operator -n
mattermost-operator -f config.yaml
```

Step 3: Deploy Mattermost

Note

- A Mattermost Enterprise license is required for multi-server deployments.
- For single-server deployments without an Enterprise license, add `Replicas: 1` to the `spec` section in step 2 below. See the high availability documentation for more on highly-available deployments.

1. **(Mattermost Enterprise only)** Create a Mattermost license secret. Create a file named `mattermost-license-secret.yaml` with the following content, replacing `[LICENSE_FILE_CONTENTS]` with your actual license:

```
apiVersion: v1 kind: Secret metadata:  name: my-mattermost-license type:
Opaque stringData:  license: <LICENSE_FILE_CONTENTS>
```

2. Create a Mattermost installation manifest file named `mattermost-installation.yaml`. File names in this guide are suggestions; you can use different names. Use the following template, adjusting the values as needed:

```
apiVersion: installation.mattermost.com/v1beta1 kind: Mattermost
metadata:  name: <INSTALLATION_NAME_HERE> # Example: mm-example-full
spec:  size: <SIZE_VALUE_HERE> # Example: 5000users  ingress:  enabled:
true  host: <FULL_DOMAIN_NAME_HERE> # Example: example.mattermost-
example.com  annotations:  kubernetes.io/ingress.class: nginx version:
<VERSION_HERE> # Example: 9.3.0 licenseSecret: "" # If you created a
license secret, put the name here
```

Key fields in the manifest include:

- `metadata.name`*: The name of your Mattermost deployment in Kubernetes.*
- `spec.size`*: The size of your installation (e.g., "100users", "1000users", etc.).*
- `spec.ingress.host`*: The DNS name for your Mattermost installation.*
- `spec.version`*: The Mattermost version. See the server version archive for available versions.*
- `spec.licenseSecret`*: The name of the Kubernetes secret containing your license (required for Enterprise).*

For a full list of configurable fields, see the [example manifest (https://github.com/mattermost/mattermost-operator/blob/master/docs/examples/mattermost_full.yaml)](https://github.com/mattermost/mattermost-operator/blob/master/docs/examples/mattermost_full.yaml) and the [Custom Resource Definition (https://github.com/mattermost/mattermost-operator/blob/master/config/crd/bases/installation.mattermost.com_mattermosts.yaml)](https://github.com/mattermost/mattermost-operator/blob/master/config/crd/bases/installation.mattermost.com_mattermosts.yaml).

3. Create a file named `mattermost-database-secret.yaml` for database credentials. This secret must be in the same namespace as the Mattermost installation.

```
apiVersion: v1 data:  DB_CONNECTION_CHECK_URL:
<DB_CONNECTION_CHECK_URL>  DB_CONNECTION_STRING: <DB_CONNECTION_STRING>
MM_SQLSETTINGS_DATASOURCEREPLICAS: <MM_SQLSETTINGS_DATASOURCEREPLICAS>
kind: Secret metadata:  name: my-postgres-connection type: Opaque
```

Example for AWS Aurora with PostgreSQL:


```
apiVersion: v1 data:  DB_CONNECTION_CHECK_URL:
cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
GVyLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD9jb2
5uZWN0X3RpbWVvdXQ9MTAK  DB_CONNECTION_STRING:
cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
GVyLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD9jb2
5uZWN0X3RpbWVvdXQ9MTAK  MM_SQLSETTINGS_DATASOURCEREPLICAS:
cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
GVyLXJvLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD
9jb25uZWN0X3RpbWVvdXQ9MTAK kind: Secret metadata:  name: my-postgres-
connection type: Opaque
```


Step 4: Create the Filestore Secret

Create a file named `mattermost-filestore-secret.yaml` to store the credentials for your object storage service (e.g., AWS S3, MinIO). This secret must be created in the same namespace where you intend to install Mattermost. The file should contain the following YAML structure:

```
apiVersion: v1 kind: Secret metadata:  name: <secret-name> # Choose a
descriptive name (e.g., my-s3-credentials) type: Opaque data:  accesskey:
<base64-encoded-access-key>  secretkey: <base64-encoded-secret-key>
```

Key	Description	Required
accesskey	Base64-encoded access key for your storage service.	Yes
secretkey	Base64-encoded secret key for your storage service.	Yes
metadata.name	The name of the Kubernetes secret.	Yes

Important

The `accesskey` and `secretkey` values must be **base64-encoded**. Do not enter the raw keys directly. Use a command-line tool or online encoder to generate the base64 strings.

Example (AWS S3):

```
apiVersion: v1 kind: Secret metadata:  name: my-s3-credentials type: Opaque
data:  accesskey: QUNDRVNTX0tFWQo= # Example: Replace with your actual
encoded key  secretkey: U1VQRVJfU0VDUkVUX0tFWQo= # Example: Replace with
your actual encoded key
```

Step 5: Configure the Mattermost Installation Manifest

1. Modify the `mattermost-installation.yaml` file (created in step 2) to connect Mattermost to your external database and object storage. Refer to the supported fields for guidance on where to add these configurations within the YAML structure.

2. Connect to the database:

> *1. Add the following to the `spec` section of your manifest:*

```
spec:  database:  external:  secret: <database-secret-name> # The name
of the database secret (e.g., my-postgres-connection)
```

3. Connect to Object Storage:

> *1. Add the following to the `spec` section of your manifest:*
>
> ```
> spec:  fileStore:  external:  url: <storage-service-url> # The URL of
> your storage service (e.g., s3.amazonaws.com)  bucket: <bucket-name> #
> The name of your storage bucket  secret: <filestore-secret-name> # The
> name of the filestore secret (e.g., my-s3-credentials)
> ```

4. If you are using Amazon S3, it's recommended to enable server-side encryption (SSE) and SSL. Add the following environment variables to the `mattermostEnv` section:

```
spec:  mattermostEnv:  MM_FILESETTINGS_AMAZONS3SSL: true
MM_FILESETTINGS_AMAZONS3SSE: true
```

Review Mattermost Resource Status

After a Mattermost installation has been created with the Operator, you can review its status with the following:

```
kubectl -n [namespace] get mattermost
```

The `kubectl describe` command can be used to obtain more information about the Mattermost server pods:

```
kubectl -n [namespace] describe pod
```

Follow logs

The following command can be used to follow logs on any kubernetes pod:

```
kubectl -n [namespace] logs -f [pod name]
```

If the `-n [namespace]` is omitted, then the default namespace of the current context is used. We recommend specifying the namespace based on your deployment.

This command can be used to review the Mattermost Operator or Mattermost server logs as needed.

> Note
>
> - If you're new to Kubernetes or prefer a managed solution, consider using a service like Amazon EKS (https://aws.amazon.com/eks/), Azure Kubernetes Service (https://azure.microsoft.com/en-ca/products/kubernetes-service/), Google Kubernetes Engine (https://cloud.google.com/kubernetes-engine/), or DigitalOcean Kubernetes (https://www.digitalocean.com/products/kubernetes/).- While this guidance focuses on using external, managed services for your database and file storage, the Mattermost Operator *does* offer the flexibility to use other solutions. For example, you could choose to deploy a PostgreSQL database within your Kubernetes cluster using the CloudNative PG operator (or externally however you wish), or use a self-hosted MinIO instance for object storage.
> - While using managed cloud services is generally simpler to maintain and our recommended approach for production deployments, using self-managed services like MinIO for storage and CloudNative PG for PostgreSQL are also valid options if you have the expertise to manage them.
> - If you choose to use self-managed components, you'll need to adapt the instructions accordingly, pointing to your internal services instead.
> - To customize your production deployment, refer to the configuration settings documentation.
> - If you encounter issues during deployment, consult the deployment troubleshooting guide.

Azure

You can use a supported Azure Marketplace Container Offer (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator) to install Mattermost on your existing Azure infrastructure.

Before you begin

Before deploying, make sure you have the following:

- **An AKS cluster**: with the Application Gateway Ingress Controller (AGIC) add-on (https://learn.microsoft.com/en-us/azure/application-gateway/tutorial-ingress-controller-add-on-new) enabled or another Ingress controller deployed.
- **PostgreSQL v13.0+ database**: Azure Database for PostgreSQL - Flexible Server with Private Access (https://learn.microsoft.com/en-us/azure/postgresql/) is recommended. Deploy one by following this Microsoft quick start guide (https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/quickstart-create-server-portal).
- **Private Network Connectivity**: Verify that there is network connectivity between your AKS cluster and the PostgreSQL database.
- **Valid DNS name and TLS certificate**: You must have access to a DNS zone and provide a valid TLS key and certificate for the Ingress Controller.
- **Node Capacity**: At least 2 AKS nodes for high availability when deploying for 100 users or more.
- **License Key**: Trial or Enterprise license to test high availability and other Enterprise features.

Installation steps

The installation process includes deploying Mattermost and updating the server.

Step 1: Deploy Mattermost

1. Deploy Mattermost from the [Azure Marketplace Container Offer (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator)](https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator) and select **Get it now**.

> - *Alternatively, you can go to the* `Extensions + Applications` *section of your AKS cluster and install the Mattermost offering from there. Visit the Microsoft cluster extensions documentation to learn more.*

2. Choose the **Resource Group** and the **Region** of your installed AKS and PostgreSQL database.

3. Choose your AKS cluster.

4. Fill in the details for your PostgreSQL database. Ensure the user specified has full access.

5. Specify Deployment Details including Deployment Name and Deployment Size. Click the checkbox to Deploy Minio, a required utility for this installation that will provide filestore functionality for your Mattermost instance.

6. Configure Mattermost installation hostname and Ingress details. The AGIC add-on is used in the following example to show the ingress annotations required.

> 1. *You can use any pre-installed Ingress Controller in your cluster as long as it supports Kubernetes Ingress and TLS termination.*
>
> ```
> kubernetes.io/ingress.class: azure/application-gateway
> appgw.ingress.kubernetes.io/ssl-redirect: "true"
> ```

7. Additionally, we recommend considering:

> 1. *Enforcing a minimum TLS version (e.g., TLS 1.2).*
> 2. *Deploying a Web Application Firewall (WAF) for additional protection, if supported by your ingress controller.*
> 3. *Limiting access using Kubernetes Network Policies.*

8. Ensure that everything is running. You should be able to check the installed plugin from the **AKS Extensions + Applications** page under the **Settings** menu.

> 1. *When the deployment is complete, obtain the hostname or IP address of your Mattermost deployment using the following command:*
>
> ```
> kubectl -n mattermost-operator get ingress
> ```

9. Use your IP address from the `ADDRESS` column, and create a DNS record in your domain registration service.

10. Access your working Mattermost installation at the URL you've determined in your DNS record.

Learn more about administrating your Mattermost server by visiting the Administration Guide.

Step 2: Upgrade Mattermost via your AKS cluster

1. Visit the `Extensions + Applications` section of your AKS cluster where your Mattermost installation is deployed.
2. You can enable minor version auto upgrades since these are not updating Mattermost version
3. Expand the `Configuration Settings` table and add the below configuration and the version you want to install as a value.

> ```
> global.azure.mattermost.version .. image:: /_static/images/
> global-azure-mattermost-version.png :alt: An example of using custom
> Mattermost version.
> ```

4. Select **Save** and wait for the upgrade.

Important

You are responsible for Azure costs associated with any infrastructure you spin up to host a Mattermost server, and Azure credits cannot be applied towards the purchase of a Mattermost license.

Oracle

You can use the supported Oracle Cloud Marketplace listing to install Mattermost Enterprise Edition on Oracle Cloud Infrastructure (OCI) using Oracle Kubernetes Engine (OKE).

Before you begin

Before deploying, make sure you have the following:

- **Oracle Cloud Account** with appropriate permissions
- **Permissions** to create/manage OKE, Compute, Networking, Database, Resource Manager, and Secrets
- **Compartment** for deployment
- **Domain Name and TLS Certificate** for secure access
- **Mattermost License Key** (Trial or Enterprise)

- **Node Capacity**: At least 2 OKE nodes for high availability when deploying for 100 users or more

Installation steps

The installation process includes deploying Mattermost and configuring the necessary components.

Step 1: Start from Oracle Cloud Marketplace

Go to the Mattermost listing and select **Launch Stack**.

Step 2: Stack Information

On the **Create stack** page, review the information, and then set the name, compartment, and Terraform version.

Step 3: Configure Variables

Set all the details for your Mattermost deployment. Each section is important for a successful and secure installation.

OKE Cluster Configuration

- **Create new OKE Cluster:**
  - Check this if you want to create a new Kubernetes cluster.
  - If you already have a cluster, you can uncheck and select your existing one.
- **Kubernetes Version:**
  - Choose the latest stable version unless you have a specific requirement.
- **Node Pool Shape (Flex/Fixed):**
  - Select a shape that fits your workload. For production, use at least 2 OCPUs and 16GB RAM per node.
- **Number of Nodes:**
  - Minimum 2 for high availability. For testing, 1 is enough. For production environments, always use at least 2 nodes and enable high availability.

- **Operating System:**
  - Oracle Linux 8 is recommended for best compatibility.

OKE Network Configuration

- **Worker Node Visibility:**
  - Private is more secure for production. Public is easier for testing. For production environments, use private nodes and restrict access to the API endpoint.
- **API Endpoint Visibility:**
  - Public allows you to manage the cluster from anywhere. Private is more secure but requires VPN or bastion.
- **Create new Virtual Cloud Network (VCN):**
  - Check this to create a new network, or uncheck to use an existing one.
- **VCN CIDR Block:**
  - Set a unique network range (e.g., `10.20.0.0/16`). Avoid overlap with other networks.

OKE Worker Nodes

- **Enable Cluster Autoscaler:**
  - Allows the cluster to automatically add or remove nodes based on usage.
- **Initial/Min Number of Worker Nodes:**
  - Set the minimum number of nodes. For high availability, use at least 2. Autoscaling helps manage costs and performance automatically.
- **Node Shape:**
  - Choose a shape (e.g., `VM.Standard.E4.Flex`) and set OCPUs and memory.

- **Auto Generate SSH Key:**
  - Enable this if you do not have your own SSH key for node access.
- **Image OS and Version:**
  - Oracle Linux 8 is recommended.

PostgreSQL Configuration

- **Admin Username:**
  - The main user for your PostgreSQL database (e.g., `admin1`).
- **Password Type:**
  - `PLAIN_TEXT` for testing, `SECRET` for production (uses Oracle Vault). Always use Oracle Vault for production passwords.
- **Password/Secret Name:**
  - Enter a strong password or the name of a secret in Oracle Vault.
- **Database Password:**
  - Required if not using a secret.

General Configuration

- **Cluster Name Prefix:**
  - Used to identify all resources (e.g., `mm-oke`).
- **Show Advanced Options:**
  - Enable for more control (encryption keys, SSH keys, etc.). Use advanced options if you need custom encryption or want to manage your own SSH keys.
- **PostgreSQL Deployment Strategy:**
  - Use "Database For PostgreSQL" for managed service.

- **Object Storage for File Storage:**
  - Enable to use OCI Object Storage for Mattermost files.
- **Mattermost Version:**
  - Use the latest stable version.
- **Namespace:**
  - Default is `mattermost`.
- **License Key:**
  - Upload or paste your Mattermost license.
- **Helm Repository:**
  - Default is `https://helm.mattermost.com`.

Step 4: Review and Apply

Check all your settings and select **Create** to start the deployment. Monitor the Resource Manager job and logs.

Step 5: After Deployment

When the job is finished, your OKE cluster, PostgreSQL database, and Mattermost will be ready. To find the Mattermost web address, run:

```
kubectl -n mattermost-operator get ingress
```

Copy the address and create a DNS record for your domain. Open your browser and go to your Mattermost URL.

Step 6: Upgrade Mattermost

To upgrade your Mattermost installation:

1. Access your OKE cluster through the Oracle Cloud Console
2. Navigate to the Mattermost operator deployment
3. Update the Mattermost version in the configuration
4. Apply the changes and wait for the upgrade to complete

> Tip
>
> **Tips for Success**
>
> - Make sure you have all the permissions you need before you start.
> - Use Oracle Vault to store passwords and sensitive data.
> - Use private nodes and secure your network for production.
> - Always monitor logs from the Resource Manager and pods using `kubectl logs` for more specific error messages.
> - For more details, see the official OCI Database with PostgreSQL documentation and OKE documentation.

Common Errors and How to Avoid Them

- **Error: Kubernetes API not reachable**
  - *Cause:* API endpoint is private and you're not connected to the VCN via VPN or Bastion.

  - *Solution:* Ensure you have access to the network or make the endpoint public for testing.

- **Error: Stack creation fails with missing permissions**
  - *Cause:* IAM policies are not set properly for the user or group.
  - *Solution:* Ensure you have permissions for Resource Manager, OKE, Networking, and Secrets.
- **Error: No ingress returned by kubectl**
  - *Cause:* Mattermost Ingress might not be ready or was misconfigured.
  - *Solution:* Check with `kubectl describe ingress` and validate DNS, TLS, and Helm values.
- **Error: PostgreSQL password rejected**
  - *Cause:* Password not set or mismatched with Oracle Vault.
  - *Solution:* Re-check the password value or Vault secret used during setup.

Important

You are responsible for Oracle Cloud Infrastructure costs for the resources you create. Oracle Cloud credits cannot be used to buy a Mattermost license.

Learn more about managing your Mattermost server by visiting the Administration Guide.

Frequently Asked Questions

Why are my pods failing with a `CrashLoopBackOff` error after adding a custom CA certificate to my Docker image?

You may see a `CrashLoopBackOff` error after adding a custom CA certificate to your Docker image's `/etc/ssl/certs` directory and deploying it to your Kubernetes environment via the Mattermost Enterprise Edition Helm Chart. This issue typically arises because the custom CA certificate is not being recognized by the system's certificate trust store, leading to TLS handshake

failures when the application attempts to connect to services that require the custom CA.

While core functionality may remain operational, you may notice the following symptoms:

- Pods stuck in a crashloop with the error message: backoff - restarting failed container in pod.
- Debugging commands like `kubectl describe` and `kubectl logs` provide little to no valuable information.
- Integrations may be blocked.

Can I resolve this issue without rebuilding the Docker image?

Yes. We recommend using Kubernetes-native solutions to manage custom CA certificates to simplify deployment processes and minimize disruptions caused by image rebuilds. You can inject the certificate directly into the pod using Kubernetes resources instead of modifying the Docker image to manage custom CA certificates dynamically without needing to rebuild and redeploy your Docker image every time the certificate changes.

Use a Kubernetes secret to store your custom CA certificate and then mount it into the pod:

1. Create a Kubernetes secret with your custom CA certificate.
2. Mount the certificate into the pod using the Helm chart's configuration options. This method simplifies management and avoids the need to rebuild your Docker image for future certificate updates.

Alternatively, to dynamically inject certificates without modifying the Docker image, use an `initContainer` to copy the certificate into the pod's filesystem and update the certificate trust store before the main container starts.

How to troubleshoot the root cause of the `CrashLoopBackOff` error?

Use `kubectl describe pods` to check detailed event logs.

Consider logging tools like Grafana to aggregate and analyze logs for additional insights.

Where is data stored in a self-hosted Kubernetes deployment?

Where data is stored depends on the backend database (such as RDS, Azure Postgres, self-hosted PostgreSQL), and the backend filestore (such as AWS S3, Minio, Mounted volume) configured during deployment.

We recommend the following S3 Options:

- AWS S3
- MinIO (Self-Hosted)

For Volume Mounts, the following options use Kubernetes Volume Mounting to provide filestores as a "local" directory to the Mattermost server:

- NFS
- AzureBlob

Not all types of [Kubernetes persistent volumes (https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes)](https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes) have been tested with Mattermost, and some may have limitations or specific configurations that may require additional setup to ensure proper permissions and access. We recommend system admins review documentation for their preferred persistent volume types and test to ensure compatibility with Mattermost.

On this page

- Deploy Mattermost on Kubernetes
- Deploy Mattermost on Kubernetes
  - Platform
    - Mattermost Operator
      - Prerequisites
      - Installation steps
        - Step 1: Install the NGINX Ingress Controller
        - Step 2: Install the Mattermost Operator
        - Step 3: Deploy Mattermost

- Step 4: Create the Filestore Secret
- Step 5: Configure the Mattermost Installation Manifest
- Review Mattermost Resource Status
- Azure
  - Before you begin
  - Installation steps
    - Step 1: Deploy Mattermost
    - Step 2: Upgrade Mattermost via your AKS cluster
- Oracle
  - Before you begin
  - Installation steps
    - Step 1: Start from Oracle Cloud Marketplace
    - Step 2: Stack Information
    - Step 3: Configure Variables
      - OKE Cluster Configuration
      - OKE Network Configuration
      - OKE Worker Nodes
      - PostgreSQL Configuration
      - General Configuration
    - Step 4: Review and Apply
    - Step 5: After Deployment
    - Step 6: Upgrade Mattermost
  - Common Errors and How to Avoid Them
- Frequently Asked Questions
  - Why are my pods failing with a CrashLoopBackOff error after adding a custom CA certificate to my Docker image?
  - Can I resolve this issue without rebuilding the Docker image?
  - How to troubleshoot the root cause of the CrashLoopBackOff error?
  - Where is data stored in a self-hosted Kubernetes deployment?
- Platform
- Frequently Asked Questions
  - Why are my pods failing with a CrashLoopBackOff error after adding a custom CA certificate to my Docker image?

- Can I resolve this issue without rebuilding the Docker image?
- How to troubleshoot the root cause of the CrashLoopBackOff error?
- Where is data stored in a self-hosted Kubernetes deployment?

Deploy Mattermost on Linux

Mattermost Server can be deployed on various Linux distributions, providing a flexible and robust platform for smaller teams and non-commercial customers. We don't recommend deploying Mattermost Server and database on a single system for production use, but it is a good option for development and testing purposes.

This page covers deployment options for major Linux distributions and installation methods. Choose your preferred platform below for specific deployment instructions:

Manual Install

Minimum system requirements:

- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)
- Database: PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can install the Mattermost Server on any 64-bit Linux system using the tarball. This is the most flexible installation method, but it comes with the highest effort, typically favored by advanced system administrators.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, download the Mattermost server, install the server, and set up the server.

Step 1: Install PostgreSQL database or get database connection credentials

- Install PostgreSQL locally on the same server by following the PostgreSQL installation (https://www.postgresql.org/download/) documentation.

- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Download

In a terminal window, ssh onto the system that will host the Mattermost Server. Using `wget`, download the Mattermost Server release you want to install using one of the following commands. Replace `amd64` with the appropriate architecture (e.g., `arm64` for ARM-based systems) in the link as needed.

Latest release

```
wget https://releases.mattermost.com/10.10.1/mattermost-10.10.1-linux-amd64.tar.gz
```

Current ESR

```
wget https://releases.mattermost.com/10.5.9/mattermost-10.5.9-linux-amd64.tar.gz
```

Older releases

If you are looking for an older release, Enterprise and Team Edition releases can be found in our version archive documentation.

Step 4: Install Mattermost server

Install the Mattermost Server by extracting the tarball, creating users and groups, and setting file/folder permissions.

1. First extract the tarball:

```
tar -xvzf mattermost*.gz
```

2. Move the entire folder to the `/opt` directory (or whatever path you require):

```
sudo mv mattermost /opt
```

3. Create the default storage folder. By default the Mattermost Server uses `/opt/mattermost/data` as the folder for files. This can be changed in the System Console during setup (even using alternative storage such as S3):

```
sudo mkdir /opt/mattermost/data
```

> Note
>
> If you choose a custom path, ensure this alternate path is used in all steps that follow.

4. Set up a user and group called `mattermost`:

```
sudo useradd --system --user-group mattermost
```

> Note
>
> If you choose a custom user and group name, ensure it is used in all the steps that follow.

5. Set the file and folder permissions for your installation:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

6. Give the `mattermost` group write permissions to the application folder:

```
sudo chmod -R g+w /opt/mattermost
```

You will now have the latest Mattermost Server version installed on your system. Starting and stopping the Mattermost Server is done using `systemd`.

7. Create the systemd unit file:

```
sudo touch /lib/systemd/system/mattermost.service
```

8. As root, edit the systemd unit file at `/lib/systemd/system/mattermost.service` to add the following lines:

```
[Unit] Description=Mattermost After=network.target [Service] Type=notify
ExecStart=/opt/mattermost/bin/mattermost TimeoutStartSec=3600
KillMode=mixed Restart=always RestartSec=10 WorkingDirectory=/opt/
mattermost User=mattermost Group=mattermost LimitNOFILE=49152 [Install]
WantedBy=multi-user.target
```

> *Note*
>
> *If you are installing the Mattermost server on the same system as your database, you may want to add both* `After=postgresql.service` *and* `BindsTo=postgresql.service` *to the* `[Unit]` *section of the systemd unit file.*

9. Save the file and reload systemd using `sudo systemctl daemon-reload`. Mattermost Server is now installed and is ready for setup.

Step 5: Set up the server

Before you start the Mattermost Server, you need to edit the configuration file. A default configuration file is located at `/opt/mattermost/config/config.json`. We recommend taking a backup of this default config ahead of making changes:

```
sudo cp /opt/mattermost/config/config.json /opt/mattermost/config/
config.defaults.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.
- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: `curl http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run sudo `systemctl enable mattermost.service` so that Mattermost will start on system boot.

Step 6: Update the server

Updating your Mattermost Server installation when using the tarball requires several manual steps. See the upgrade Mattermost Server documentation for details.

Remove Mattermost

To remove the Mattermost Server for any reason, you must stop the Mattermost Server, back up all important files, and then run this command:

```
sudo rm - rf /opt/mattermost
```

> Note
>
> Depending on your configuration, there are several important folders in `/opt/mattermost` to backup. These are `config`, `logs`, `plugins`, `client/plugins`, and `data`. We strongly recommend you back up these locations before running the `rm` command.

You may also remove the Mattermost systemd unit file and the user/group created for running the application.

Ubuntu

Minimum system requirements:

- Operating System: 20.04 LTS, 22.04 LTS, 24.04 LTS
- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)
- Database: [PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)](https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can deploy Mattermost server using our `.deb` signed packages using the Mattermost PPA (Personal Package Archive). This is the quickest way to install a Mattermost Server that provides automatic updates. This install method is used for both single and clustered installations, as you can tools like Packer for a clustered deployment.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, add the PPA repository, install Mattermost server, configure the server, and update the server.

Step 1: Install PostgreSQL database or get database connection credentials

Mattermost requires a PostgreSQL database. You can either:

- Install PostgreSQL locally on the same server by following the [PostgreSQL installation (https://www.postgresql.org/download/)](https://www.postgresql.org/download/) documentation.
- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Add the Mattermost Server PPA repository

Important

The GPG public key has changed. You can [import the new public key (https://deb.packages.mattermost.com/pubkey.gpg)](https://deb.packages.mattermost.com/pubkey.gpg) or run the automatic Mattermost PPA repository setup script provided below. Depending on your setup, additional steps may also be required, particularly for installations that didn't rely on the repository setup script. We recommend deleting the old key from `/etc/apt/trusted.gpg.d` before adding the apt repository.

- For Ubuntu Jammy - 22.04 LTS and Ubuntu Noble - 24.04 LTS:

  ```
  sudo rm /usr/share/keyrings/mattermost-archive-keyring.gpg
  ```

  ```
  curl -sL -o- https://deb.packages.mattermost.com/pubkey.gpg | gpg --dearmor | sudo tee /usr/share/keyrings/mattermost-archive-keyring.gpg > /dev/null
  ```

In a terminal window, run the following repository setup command to add the Mattermost Server repositories:

```
curl -o- https://deb.packages.mattermost.com/repo-setup.sh | sudo bash -s
mattermost
```

This command configures the repositories needed for a PostgreSQL database, configures an NGINX web server to act as a proxy, configures certbot to issue and renew the SSL certificate, and configures the Mattermost repository so that you can run the install command.

Step 4: Install Mattermost server

Ahead of installing the Mattermost Server, it's good practice to update all your repositories and, where required, update existing packages by running the following command:

```
sudo apt update
```

After any updates and system reboots are complete, you can install the Mattermost Server by running:

```
sudo apt install mattermost -y
```

You now have the latest Mattermost Server version installed on your system.

The installation path is `/opt/mattermost`. The package will have added a user and group named `mattermost`. The required systemd unit file has also been created but will not be set to active.

> Note
>
> Since the signed package from the Mattermost repository is used for mulitple installation types, we don't add any dependencies in the systemd unit file. If you are installing the Mattermost server on the same system as your database, you may want to add both `After=postgresql.service` and `BindsTo=postgresql.service` to the `[Unit]` section of the systemd unit file.

Step 5: Configure the server

Before you start the Mattermost Server, you need to edit the configuration file. A sample configuration file is located at `/opt/mattermost/config/config.defaults.json`.

Rename this configuration file with correct permissions:

```
sudo install -C -m 600 -o mattermost -g mattermost /opt/mattermost/config/
config.defaults.json /opt/mattermost/config/config.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.
- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost Server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run `sudo systemctl enable mattermost.service` so that Mattermost will start on system boot.

> Note
>
> The value of the `sslmode` property in the `DataSource` configuration is entirely dependent on your native environment. Please consult the native environment setup documentation for guidance on its value. The available options for `sslmode` are `disable` or `require`. For example, if you are using Amazon Lightsail as your data source, you must set `sslmode` to `require` to successfully connect to the database.

Step 6: Update the server

When a new Mattermost version is released, run: `sudo apt update && sudo apt upgrade` to download and update your Mattermost instance.

> Note
>
> When you run the `sudo apt upgrade` command, `mattermost-server` will be updated along with any other packages. We strongly recommend you stop the Mattermost Server before running the `apt` command using `sudo systemctl stop mattermost`.

Remove Mattermost

Run the following command to remove the Mattermost Server:

```
sudo apt remove --purge mattermost
```

RHEL/CentOS

Minimum system requirements:

- Operating System: Enterprise Linux 7+, Oracle Linux 6+, Oracle Linux 7+
- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)

- Database: [PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)](https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can deploy Mattermost Server using our `rpm` signed packages available through the Mattermost Yum repository.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, download the Mattermost server, install the server, set up the server, and update the server.

Step 1: Install PostgreSQL database or get database connection credentials

- Install PostgreSQL locally on the same server by following the [PostgreSQL installation (https://www.postgresql.org/download/)](https://www.postgresql.org/download/) documentation.
- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Download the latest Mattermost Server tarball

In a terminal window, ssh onto the system that will host the Mattermost Server. Using `wget`, download the Mattermost Server release you want to install using one of the following commands. Replace `amd64` with the appropriate architecture (e.g., `arm64` for ARM-based systems) in the link as needed.

Latest release

```
wget https://releases.mattermost.com/10.10.1/mattermost-10.10.1-linux-amd64.tar.gz
```

Current ESR

```
wget https://releases.mattermost.com/10.5.9/mattermost-10.5.9-linux-amd64.tar.gz
```

Older releases

If you are looking for an older release, Enterprise and Team Edition releases can be found in our version archive documentation.

Step 4: Install Mattermost server

1. Ahead of installing the Mattermost Server, we recommend updating all your repositories and, where required, update existing packages by running the following commands:

> ```
> sudo dnf update sudo dnf upgrade
> ```

2. After any updates, and any system reboots, are complete, install the Mattermost Server by extracting the tarball, creating users and groups, and setting file/folder permissions.

> *1. First extract the tarball:*
>
> > ```
> > tar -xvzf mattermost*.gz
> > ```
>
> *2. Now move the entire folder to the `/opt` directory (or whatever path you require):*
>
> > ```
> > sudo mv mattermost /opt
> > ```
>
> *3. Create the default storage folder. By default the Mattermost Server uses `/opt/mattermost/data` as the folder for files. This can be changed in the System Console during setup (even using alternative storage such as S3):*

```
sudo mkdir /opt/mattermost/data
```

> Note
>
> If you choose a custom path, ensure this alternate path is used in all steps that follow.`

3. Set up a user and group called `mattermost`:

```
sudo useradd --system --user-group mattermost
```

> Note
>
> If you choose a custom user and group name, ensure it is used in all the steps that follow.

4. Set the file and folder permissions for your installation:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

5. Give the `mattermost` group write permissions to the application folder:

```
sudo chmod -R g+w /opt/mattermost
```

> *You will now have the latest Mattermost Server version installed on your system. Starting and stopping the Mattermost Server is done using `systemd`.*

6. Create the systemd unit file:

```
sudo touch /lib/systemd/system/mattermost.service
```

7. As root, edit the systemd unit file at `/lib/systemd/system/mattermost.service` to add the following lines:

```
[Unit] Description=Mattermost After=network.target [Service] Type=notify
ExecStart=/opt/mattermost/bin/mattermost TimeoutStartSec=3600
KillMode=mixed Restart=always RestartSec=10 WorkingDirectory=/opt/
mattermost User=mattermost Group=mattermost LimitNOFILE=49152 [Install]
WantedBy=multi-user.target
```

8. Save the file and reload systemd using `sudo systemctl daemon-reload`. Mattermost Server is now installed and is ready for setup.

Step 5: Set up the server

Before you start the Mattermost Server, you need to edit the configuration file. A default configuration file is located at `/opt/mattermost/config/config.json`. We recommend taking a backup of this default config ahead of making changes:

```
sudo cp /opt/mattermost/config/config.json /opt/mattermost/config/
config.defaults.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.
- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

> Note
>
> We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run sudo `systemctl enable mattermost.service` so that Mattermost will start on system boot. If you don't receive an error when starting Mattermost after the previous step, you are good to go. If you did receive an error, continue on.

Important

Modify SELinux settings: When deploying Mattermost from RHEL9, which has SELinux running with enforceing mode enabled by default, additional configuration is required.

- SELinux is a security module that provides access control security policies. It's enabled by default on RHEL and CentOS systems. SELinux can block access to files, directories, and ports, which can cause issues when starting Mattermost. To resolve these issues, you'll need to set the appropriate SELinux contexts for the Mattermost binaries and directories, and allow Mattermost to bind to ports.
- Ensure that SELinux is enabled and in enforcing mode by running the `sestatus` command. If it's `enforcing`, you'll need to configure it properly.
- Set bin contexts for `/opt/mattermost/bin`: SELinux enforces security contexts for binaries. To label the Mattermost binaries as safe, you'll need to set them to the below SELinux context.

```
sudo semanage fcontext -a -t bin_t "/opt/mattermost/bin(/.*)?" sudo
restorecon -RF /opt/mattermost/bin
```

Now, try starting Mattermost again with

```
sudo systemctl start mattermost
```

If you don't receive an error, verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server. You're all set!

If on starting Mattermost you receive an error, before moving on, check for the existence of a file in `/opt/mattermost/logs` - if `mattermost.log` exists in that directory, it's more likely you're dealing with a configuration issue in `config.json`. Double check the previous steps before continuing

Try different contexts for `/opt/mattermost`: SELinux enforces security contexts for files and directories. To label your Mattermost directory as safe, you'll need to set an appropriate SELinux context.

> 1. *Check current context by running `ls -Z /opt/mattermost`. When you see something like `drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 mattermost` returned, the `default_t` indicates that SELinux doesn't know what this directory is for.*
>
> 2. *Set a safe context by assigning a SELinux type that's compatible with web services or applications by running `sudo semanage fcontext -a -t httpd_sys_content_t "/opt/mattermost(/.*)?"`. A common one is `httpd_sys_content_t`, used for serving files. Ensure you match the directory and its contents recursively. Run the `sudo restorecon -R /opt/mattermost` to apply the changes.*

Allow Mattermost to bind to ports: When Mattermost needs specific ports (e.g., 8065), ensure that SELinux allows it by allowing Mattermost to bind to ports. Run the `sudo semanage port -l | grep 8065` command, and if the port's not listed, you'll need to add it by running `sudo semanage port -a -t http_port_t -p tcp 8065`, replacing the `8065` with the required port.

Handle custom policies: If Mattermost requires actions that SELinux blocks, you'll need to generate a custom policy.

> 1. *Check for SELinux denials first in the logs by running* `sudo ausearch -m avc -ts recent`*, or by checking the audit log:* `sudo cat /var/log/audit/audit.log | grep denied`*.*
> 2. *If needed, generate a policy module by installing* `audit2allow` *to generate policies automatically.*
>
> ```
> sudo yum install -y policycoreutils-python-utils sudo grep
> mattermost /var/log/audit/audit.log | audit2allow -M
> mattermost_policy sudo semodule -i mattermost_policy.pp
> ```

Test the configuration: Restart Mattermost to confirm the configuation works as expected by running `sudo systemctl restart mattermost`. In the case of failures, revisit the logs to identify other SELinux-related issues.

Need Mattermost working quickly for testing purposes?

- You can change SELinux to permissive mode by running the `sudo setenforce 0`. command where policies aren't enforced, only logged.
- This command changes the SELinux mode to "permissive". While in permissive mode, policies aren't enforced, and violations are logged instead of being blocked. This can be helpful for debugging and troubleshooting issues related to SELinux policies.
- Ensure you re-enable enforcing mode once context is working as needed by running the `sudo setenforce 1` command.

See the following SELinux resources for additional details:

> - *SELinux User's and Administrator's Guide (https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/index)*
> - *SELinux Project Wiki (https://github.com/SELinuxProject/selinux)*
> - *Introduction to SELinux (https://github.blog/developer-skills/programming-languages-and-frameworks/introduction-to-selinux/)*

> - *A Sysadmin's Guide to SELinux: 42 Answers to the Big Questions (https://opensource.com/article/18/7/sysadmin-guide-selinux)*
> - *Mastering SELinux: A Comprehensive Guide to Linux Security (https://srivastavayushmaan1347.medium.com/mastering-selinux-a-comprehensive-guide-to-linux-security-8bed9976da88)*

Step 6: Update the server

Updating your Mattermost Server installation when using the tarball requires several manual steps. See the upgrade Mattermost Server documentation for details.

Remove Mattermost

To remove the Mattermost Server, you must stop the Mattermost Server, back up all important files, and then run this command:

```
sudo rm /opt/mattermost
```

> Note
>
> Depending on your configuration, there are several important folders in `/opt/mattermost` to backup. These are `config`, `logs`, `plugins`, `client/plugins`, and `data`. We strongly recommend you back up these locations before running the `rm` command.

You may also remove the Mattermost systemd unit file and the user/group created for running the application.

Secure your Mattermost deployment

Configuring TLS and setting up an NGINX proxy ensures secure communication between clients and your Mattermost server. This setup allows

you to serve HTTPS traffic while proxying requests to Mattermost. You don't need TLS enabled within Mattermost itself as NGINX will handle HTTPS traffic.

1. Install NGINX on the host server. See the set up NGINX proxy documentation for details.
2. Obtain a TLS certificate from a trusted certificate authority (CA) or use a self-signed certificate for testing purposes.
3. Configure NGINX with TLS certificates to serve HTTPS traffic. NGINX serves as a proxy, forwarding requests to the Mattermost application running locally or on a separate server.

> Note
>
> See the deployment troubleshooting documentation for resolutions to common deployment issues.

On this page

- Deploy Mattermost on Linux
- Deploy Mattermost on Linux
  - Manual Install
    - Latest release
    - Current ESR
    - Older releases
    - Remove Mattermost
  - Ubuntu
    - Remove Mattermost
  - RHEL/CentOS
    - Latest release
    - Current ESR
    - Older releases
    - Remove Mattermost
  - Secure your Mattermost deployment
  - Secure your Mattermost deployment

Deploy Mattermost as an Out-of-Band (OOB) Communication System

When a critical incident disrupts an organization's primary infrastructure—due to a cyberattack, cloud outage, or internal failure—a secure, independent communication channel becomes essential. Mattermost can serve as that channel by operating in an Out-of-Band (OOB) environment: isolated, pre-provisioned, and ready to support incident response and business continuity teams.

This document outlines architectural guidance for deploying Mattermost in an OOB scenario across generic, AWS, and Azure environments.

Generic OOB architecture

This platform-agnostic design establishes a fallback communication system, decoupled from primary networks and identity providers.

Architecture components

- **Users (Cybersecurity, IR Teams)**: Pre-authorized personnel trained to access and use the OOB platform during emergencies. Credentials and access methods should be provisioned in advance.
- **Secure Access Layer**: A firewall or access gateway protecting entry into the OOB environment. May include network policies, IP allowlists, or WAFs.

- **Secondary IdP / Emergency Auth**: A dedicated identity provider not dependent on the organization's primary SSO or IdP (e.g., a minimal Azure AD tenant, Okta org, or locally stored credentials with MFA).
- **VPN / Bastion Host**: A hardened jumpbox or independent VPN granting access to internal OOB components and Kubernetes control plane.
- **Failover DNS Manager**: Services like Cloudflare Load Balancer, Azure Traffic Manager, or Route 53 to monitor primary system health and reroute DNS to the OOB ingress.
- **Ingress Controller**: e.g., ingress-nginx to handle HTTP(S) routing within the cluster, directing traffic to Mattermost Web, API, and WebSocket endpoints.
- **Mattermost Operator & App**: Automates lifecycle of Mattermost instances in Kubernetes; ensures updates, scaling, and configuration consistency.
- **PostgreSQL Database**: Persistent storage for channels, messages, users, and configurations; can be self-hosted or managed externally with hardened access.
- **Backups**:
  - **Velero**: Backs up Kubernetes resources to external object store (S3, Blob).
  - **pg_dump / Cloud-native backup**: Regular logical exports or managed backups.
  - **Object Storage**: Durable, versioned storage (e.g., S3, Azure Blob) for snapshots and dumps.

AWS-based OOB deployment

Deploying Mattermost in AWS leverages managed services and strong isolation capabilities.

AWS components

- **Amazon EKS**: Managed Kubernetes for Mattermost Operator and application, with HA and auto-scaling.
- **Route 53**: DNS service for health-checked failover to the OOB environment.
- **Amazon Aurora PostgreSQL**: High-performance managed PostgreSQL with automated failovers.
- **Amazon S3**: Object storage for Velero snapshots, configurations, and database backups (with versioning and cross-region replication).
- **IAM Roles**: Scoped roles for EKS nodes and Velero using IRSA, enforcing least-privilege.

> Tip
>
> - Use separate AWS accounts or VPCs for full isolation.
> - Configure Route 53 health checks against public or synthetic endpoints.
> - Encrypt secrets and enforce least-privilege IAM policies.

Azure-Based OOB deployment

For Azure customers, AKS and related services provide a robust OOB platform for Mattermost.

Azure components

- **Azure Kubernetes Service (AKS)**: Runs Mattermost workloads with native autoscaling and Azure AD integration.
- **Azure DNS + Traffic Manager**: Global traffic routing and DNS failover based on endpoint health.
- **Azure Database for PostgreSQL**: Managed DB service with automated backups and patching.
- **Azure Blob Storage**: Destination for Velero snapshots and DB dumps; supports lifecycle policies and secure access.
- **Azure AD (Secondary Tenant)**: Acts as the IdP for OOB; deployed in a separate directory with scoped permissions and MFA.

> Tip
>
> - Deploy in a dedicated subscription for security and billing separation.
> - Use Azure Monitor for health checks and Traffic Manager probes.
> - Harden AKS node pools and isolate workloads with namespaces and network policies.

Operational guidelines

Domain	Recommendation
Access Control	Isolate authentication from primary infrastructure. Use MFA, secure vaults, and emergency tokens. Preload user accounts in the secondary IdP.
DNS Failover	Employ TTL-aware health checks via Route 53 or Traffic Manager. Monitor critical HTTP(S) or TCP endpoints.
Backup Strategy	Schedule frequent Velero and database backups. Store in versioned, encrypted object storage.
Disaster Recovery Drills	Conduct quarterly failover exercises: DNS switching, Mattermost access, and data restores.
Security	Treat OOB as Tier 0. Use hardened OS images, audit logging, strict RBAC, and centralized monitoring. Rotate secrets regularly.

Mattermost can be an effective OOB communication platform when deployed with isolation, automation, and operational readiness. Whether on AWS, Azure, or a generic environment, prioritize independence, simplicity, and resilience. Pair these designs with standard IR playbooks and routine failover testing to build a confident, secure response capability.

On this page

- Deploy Mattermost as an Out-of-Band (OOB) Communication System
  - Generic OOB architecture
    - Architecture components
  - AWS-based OOB deployment
    - AWS components

- Azure-Based OOB deployment
  - Azure components
- Operational guidelines

Docker deployment troubleshooting

Permission issues on M1 Mac

If you're deploying the Mattermost server using Docker on an M1 Mac and encountering permission issues in the Docker container, re-create the required directories and set their permissions, then skip the following command because it causes the deploy to stop working.

```
sudo chown -R 2000:2000 ./volumes/app/mattermost
```

If you're experiencing issues deploying on Docker generally, ensure the docker daemon is enabled and running:

```
sudo systemctl enable --now docker
```

To remove all data and settings for your Mattermost deployment:

```
sudo rm -rf ./volumes
```

TLS and NGINX issues

For an in-depth guide to configuring the TLS certificate and key for NGINX, please refer to [this document in the repository (https://github.com/mattermost/docker/blob/main/docs/issuing-letsencrypt-certificate.md)](https://github.com/mattermost/docker/blob/main/docs/issuing-letsencrypt-certificate.md).

Install a different version of Mattermost

1. Shut down your deployment.
2. Run `git pull` to fetch any recent changes to the repository, paying attention to any potential `env.example` changes.

3. Adjust the `MATTERMOST_IMAGE_TAG` in the `.env` file to point your desired [enterprise (https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-enterprise-edition/tags?page=1&ordering=last_updated) or [team (https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated)](https://hub.docker.com/r/mattermost/mattermost-team-edition/tags?page=1&ordering=last_updated) image version.

   Important

   For production environments, we recommend using specific version tags such as `MATTERMOST_IMAGE_TAG=release-10.5` rather than generic tags like `MATTERMOST_IMAGE_TAG=release-10`. Generic `release-x` tags are intended for development use only and do not automatically receive new patch releases within that major version. Using specific version tags ensures a more reproducible and deterministic environment for your production deployment.

4. Redeploy Mattermost.

Unintentional version downgrades

If you experience an unintentional downgrade when using generic `MATTERMOST_IMAGE_TAG=release-x` tags, this is because these tags are designed for development use and may not point to the latest patch release within that major version.

Solution: Use a more specific version tag for your Docker image, such as `MATTERMOST_IMAGE_TAG=release-10.5`, to avoid unexpected version changes and ensure consistent deployments.

> Note
>
> A [pipeline improvement (https://github.com/mattermost/mattermost/issues/30656)](https://github.com/mattermost/mattermost/issues/30656) is in progress to ensure that generic `release-x` tags are updated to the latest version from the corresponding release branch. Once this improvement is implemented, the behavior of these tags will be more predictable.

Upgrading from `mattermost-docker`

For an in-depth guide to upgrading from the deprecated mattermost-docker repository (https://github.com/mattermost/mattermost-docker), please refer to this document (https://github.com/mattermost/docker/blob/main/scripts/UPGRADE.md). For additional help, please refer to this issue (https://github.com/mattermost/mattermost-docker/issues/489).

On this page

- Docker deployment troubleshooting
  - Permission issues on M1 Mac
  - TLS and NGINX issues
  - Install a different version of Mattermost
  - Unintentional version downgrades
  - Upgrading from mattermost-docker

(Optional) Use an Image proxy

When enabled, the image proxy needs to be publicly accessible to both the Mattermost client and server.

Mattermost clients will use the image proxy to load all external images. The Mattermost server will use the image proxy when possible, but will not use it when requesting content that may not be an image, such as for image previews of plaintext URLs (https://github.com/mattermost/mattermost/issues/11857).

Configure an image proxy by going to **System Console > Environment > Image Proxy**.

Local image proxy

The local image proxy is available as part of the Mattermost server deployment. When using the local image proxy, images are served to clients through the server which helps anonymize users. If SSL is enabled on the server, it provides a secure connection. This method does not offer any caching behavior.

> Note
>
> With the local image proxy enabled, requests for images hosted on the local network are now affected by the `AllowUntrustedInternalConnections` setting. See documentation for more information or if you are seeing unintentionally blocked images.

atmos/camo image proxy

The atmos/camo (https://github.com/atmos/camo) image proxy is a standalone image proxy that can be deployed separately from the Mattermost server. It provides additional configuration options over the built-in image

proxy, and it can also be used if isolation between the Mattermost server and image proxy is desired.

Once you've deployed an `atmos/camo` (https://github.com/atmos/camo) instance, you must specify the **Remote Image Proxy URL** and **Remote Image Proxy Options** settings. The **Remote Image Proxy Options** should be set to the image proxy's shared key which is specified with the `CAMO_KEY` environment variable used when setting up the image proxy.

For example, if the image proxy is located at `https://image-proxy.mattermost.com`, it would be configured as follows:

> - ***Image Proxy Type***: `atmos/camo`
> - ***Remote Image Proxy URL***: `https://image-proxy.mattermost.com`
> - ***Remote Image Proxy Options***: `CAMO_KEY`, *which is the secret string used for the sample* `atmos/camo` *deployment.*

On this page

- (Optional) Use an Image proxy
  - Local image proxy
  - atmos/camo image proxy

Before you begin

You can use a supported Azure Marketplace Container Offer (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator) to install Mattermost on your existing Azure infrastructure.

Before deploying, make sure you have the following:

- **An AKS cluster**: with the Application Gateway Ingress Controller (AGIC) add-on (https://learn.microsoft.com/en-us/azure/application-gateway/tutorial-ingress-controller-add-on-new) enabled or another Ingress controller deployed.
- **PostgreSQL v13.0+ database**: Azure Database for PostgreSQL - Flexible Server with Private Access (https://learn.microsoft.com/en-us/azure/postgresql/) is recommended. Deploy one by following this Microsoft quick start guide (https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/quickstart-create-server-portal).
- **Private Network Connectivity**: Verify that there is network connectivity between your AKS cluster and the PostgreSQL database.
- **Valid DNS name and TLS certificate**: You must have access to a DNS zone and provide a valid TLS key and certificate for the Ingress Controller.
- **Node Capacity**: At least 2 AKS nodes for high availability when deploying for 100 users or more.
- **License Key**: Trial or Enterprise license to test high availability and other Enterprise features.

Installation steps

The installation process includes deploying Mattermost and updating the server.

Step 1: Deploy Mattermost

1. Deploy Mattermost from the [Azure Marketplace Container Offer (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator)](https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mattermost.mattermost-operator) and select **Get it now**.

> - *Alternatively, you can go to the* `Extensions + Applications` *section of your AKS cluster and install the Mattermost offering from there. Visit the Microsoft cluster extensions documentation to learn more.*

2. Choose the **Resource Group** and the **Region** of your installed AKS and PostgreSQL database.

3. Choose your AKS cluster.

4. Fill in the details for your PostgreSQL database. Ensure the user specified has full access.

5. Specify Deployment Details including Deployment Name and Deployment Size. Click the checkbox to Deploy Minio, a required utility for this installation that will provide filestore functionality for your Mattermost instance.

6. Configure Mattermost installation hostname and Ingress details. The AGIC add-on is used in the following example to show the ingress annotations required.

> 1. *You can use any pre-installed Ingress Controller in your cluster as long as it supports Kubernetes Ingress and TLS termination.*
>
> ```
> kubernetes.io/ingress.class: azure/application-gateway
> appgw.ingress.kubernetes.io/ssl-redirect: "true"
> ```

7. Additionally, we recommend considering:

> 1. *Enforcing a minimum TLS version (e.g., TLS 1.2).*
> 2. *Deploying a Web Application Firewall (WAF) for additional protection, if supported by your ingress controller.*
> 3. *Limiting access using Kubernetes Network Policies.*

8. Ensure that everything is running. You should be able to check the installed plugin from the **AKS Extensions + Applications** page under the **Settings** menu.

> 1. *When the deployment is complete, obtain the hostname or IP address of your Mattermost deployment using the following command:*
>
> ```
> kubectl -n mattermost-operator get ingress
> ```

9. Use your IP address from the `ADDRESS` column, and create a DNS record in your domain registration service.

10. Access your working Mattermost installation at the URL you've determined in your DNS record.

Learn more about administrating your Mattermost server by visiting the Administration Guide.

Step 2: Upgrade Mattermost via your AKS cluster

1. Visit the `Extensions + Applications` section of your AKS cluster where your Mattermost installation is deployed.
2. You can enable minor version auto upgrades since these are not updating Mattermost version
3. Expand the `Configuration Settings` table and add the below configuration and the version you want to install as a value.

> ```
> global.azure.mattermost.version .. image:: /_static/images/
> global-azure-mattermost-version.png :alt: An example of using custom
> Mattermost version.
> ```

4. Select **Save** and wait for the upgrade.

Important

You are responsible for Azure costs associated with any infrastructure you spin up to host a Mattermost server, and Azure credits cannot be applied towards the purchase of a Mattermost license.

Before you begin

You can use the supported Oracle Cloud Marketplace listing to install Mattermost Enterprise Edition on Oracle Cloud Infrastructure (OCI) using Oracle Kubernetes Engine (OKE).

Before deploying, make sure you have the following:

- **Oracle Cloud Account** with appropriate permissions
- **Permissions** to create/manage OKE, Compute, Networking, Database, Resource Manager, and Secrets
- **Compartment** for deployment
- **Domain Name and TLS Certificate** for secure access
- **Mattermost License Key** (Trial or Enterprise)
- **Node Capacity**: At least 2 OKE nodes for high availability when deploying for 100 users or more

Installation steps

The installation process includes deploying Mattermost and configuring the necessary components.

Step 1: Start from Oracle Cloud Marketplace

Go to the Mattermost listing and select **Launch Stack**.

Step 2: Stack Information

On the **Create stack** page, review the information, and then set the name, compartment, and Terraform version.

Step 3: Configure Variables

Set all the details for your Mattermost deployment. Each section is important for a successful and secure installation.

OKE Cluster Configuration

- **Create new OKE Cluster:**
  - Check this if you want to create a new Kubernetes cluster.

  - If you already have a cluster, you can uncheck and select your existing one.
- **Kubernetes Version:**
  - Choose the latest stable version unless you have a specific requirement.
- **Node Pool Shape (Flex/Fixed):**
  - Select a shape that fits your workload. For production, use at least 2 OCPUs and 16GB RAM per node.
- **Number of Nodes:**
  - Minimum 2 for high availability. For testing, 1 is enough. For production environments, always use at least 2 nodes and enable high availability.
- **Operating System:**
  - Oracle Linux 8 is recommended for best compatibility.

OKE Network Configuration

- **Worker Node Visibility:**
  - Private is more secure for production. Public is easier for testing. For production environments, use private nodes and restrict access to the API endpoint.
- **API Endpoint Visibility:**
  - Public allows you to manage the cluster from anywhere. Private is more secure but requires VPN or bastion.
- **Create new Virtual Cloud Network (VCN):**
  - Check this to create a new network, or uncheck to use an existing one.
- **VCN CIDR Block:**
  - Set a unique network range (e.g., `10.20.0.0/16`). Avoid overlap with other networks.

OKE Worker Nodes

- **Enable Cluster Autoscaler:**
  - Allows the cluster to automatically add or remove nodes based on usage.
- **Initial/Min Number of Worker Nodes:**
  - Set the minimum number of nodes. For high availability, use at least 2. Autoscaling helps manage costs and performance automatically.
- **Node Shape:**
  - Choose a shape (e.g., `VM.Standard.E4.Flex`) and set OCPUs and memory.
- **Auto Generate SSH Key:**
  - Enable this if you do not have your own SSH key for node access.
- **Image OS and Version:**
  - Oracle Linux 8 is recommended.

PostgreSQL Configuration

- **Admin Username:**
  - The main user for your PostgreSQL database (e.g., `admin1`).
- **Password Type:**
  - `PLAIN_TEXT` for testing, `SECRET` for production (uses Oracle Vault). Always use Oracle Vault for production passwords.
- **Password/Secret Name:**
  - Enter a strong password or the name of a secret in Oracle Vault.
- **Database Password:**
  - Required if not using a secret.

General Configuration

- **Cluster Name Prefix:**
  - Used to identify all resources (e.g., `mm-oke`).
- **Show Advanced Options:**
  - Enable for more control (encryption keys, SSH keys, etc.). Use advanced options if you need custom encryption or want to manage your own SSH keys.
- **PostgreSQL Deployment Strategy:**
  - Use "Database For PostgreSQL" for managed service.
- **Object Storage for File Storage:**
  - Enable to use OCI Object Storage for Mattermost files.
- **Mattermost Version:**
  - Use the latest stable version.
- **Namespace:**
  - Default is `mattermost`.
- **License Key:**
  - Upload or paste your Mattermost license.
- **Helm Repository:**
  - Default is `https://helm.mattermost.com`.

Step 4: Review and Apply

Check all your settings and select **Create** to start the deployment. Monitor the Resource Manager job and logs.

Step 5: After Deployment

When the job is finished, your OKE cluster, PostgreSQL database, and Mattermost will be ready. To find the Mattermost web address, run:

```
kubectl -n mattermost-operator get ingress
```

Copy the address and create a DNS record for your domain. Open your browser and go to your Mattermost URL.

Step 6: Upgrade Mattermost

To upgrade your Mattermost installation:

1. Access your OKE cluster through the Oracle Cloud Console
2. Navigate to the Mattermost operator deployment
3. Update the Mattermost version in the configuration
4. Apply the changes and wait for the upgrade to complete

> Tip
>
> **Tips for Success**
>
> - Make sure you have all the permissions you need before you start.
> - Use Oracle Vault to store passwords and sensitive data.
> - Use private nodes and secure your network for production.
> - Always monitor logs from the Resource Manager and pods using `kubectl logs` for more specific error messages.
> - For more details, see the official OCI Database with PostgreSQL documentation and OKE documentation.

Common Errors and How to Avoid Them

- **Error: Kubernetes API not reachable**
  - *Cause:* API endpoint is private and you're not connected to the VCN via VPN or Bastion.
  - *Solution:* Ensure you have access to the network or make the endpoint public for testing.
- **Error: Stack creation fails with missing permissions**
  - *Cause:* IAM policies are not set properly for the user or group.
  - *Solution:* Ensure you have permissions for Resource Manager, OKE, Networking, and Secrets.
- **Error: No ingress returned by kubectl**
  - *Cause:* Mattermost Ingress might not be ready or was misconfigured.

  - *Solution:* Check with `kubectl describe ingress` and validate DNS, TLS, and Helm values.

- **Error: PostgreSQL password rejected**
  - *Cause:* Password not set or mismatched with Oracle Vault.
  - *Solution:* Re-check the password value or Vault secret used during setup.

Important

You are responsible for Oracle Cloud Infrastructure costs for the resources you create. Oracle Cloud credits cannot be used to buy a Mattermost license.

Learn more about managing your Mattermost server by visiting the Administration Guide.

Prerequisites

You can use the Mattermost Kubernetes Operator to deploy Mattermost on Kubernetes using S3-compatible storage and a managed database service. While the operator supports a range of configurations, we strongly recommend using a cloud-native approach for production environments.

Before you begin, ensure you have the following:

- A functioning Kubernetes cluster (see the Kubernetes setup guide (https://kubernetes.io/docs/setup/)). Your cluster should be running a supported Kubernetes version (https://kubernetes.io/releases/).
- The *kubectl* command-line tool installed on your local machine (see the kubectl installation guide (https://kubernetes.io/docs/reference/kubectl/)).
- A fundamental understanding of Kubernetes concepts, such as deployments, pods, and applying manifests.
- Sufficient Kubernetes resources allocated based on your expected user load. Consult the scaling for Enterprise documentation for resource requirements at different scales.

Installation steps

The installation process involves setting up necessary operators and then deploying Mattermost itself.

Step 1: Install the NGINX Ingress Controller

Follow the instructions in the Kubernetes deployment documentation to install the NGINX ingress controller on your Kubernetes cluster. Mattermost recommends installing the Nginx Operator via helm, regardless of platform you are installing to.

Step 2: Install the Mattermost Operator

The Mattermost Kubernetes Operator can be installed using Helm.

1. Install Helm (version 3.13.0 or later). See the Helm quickstart documentation for installation instructions.
2. Add the Mattermost Helm repository:

```
helm repo add mattermost https://helm.mattermost.com
```

3. Create a file named `config.yaml` and populate it with the contents of the [Mattermost operator values file (https://github.com/mattermost/mattermost-helm/blob/master/charts/mattermost-operator/values.yaml)](https://github.com/mattermost/mattermost-helm/blob/master/charts/mattermost-operator/values.yaml). This file allows for customization of the operator.
4. Create a namespace for the Mattermost Operator:

```
kubectl create ns mattermost-operator
```

5. Install the Mattermost Operator. If you don't specify a version, the latest version of the Mattermost Operator will be installed.

```
helm install <your-release-name> mattermost/mattermost-operator -n
<namespace_name>
```

For example:

```
helm install mattermost-operator mattermost/mattermost-operator -n
mattermost-operator
```

To use your custom `config.yaml` file:

```
helm install mattermost-operator mattermost/mattermost-operator -n
mattermost-operator -f config.yaml
```

Step 3: Deploy Mattermost

> Note
>
> - A Mattermost Enterprise license is required for multi-server deployments.
> - For single-server deployments without an Enterprise license, add `Replicas: 1` to the `spec` section in step 2 below. See the high availability documentation for more on highly-available deployments.

1. **(Mattermost Enterprise only)** Create a Mattermost license secret. Create a file named `mattermost-license-secret.yaml` with the following content, replacing `[LICENSE_FILE_CONTENTS]` with your actual license:

```
apiVersion: v1 kind: Secret metadata:  name: my-mattermost-license type:
Opaque stringData:  license: <LICENSE_FILE_CONTENTS>
```

2. Create a Mattermost installation manifest file named `mattermost-installation.yaml`. File names in this guide are suggestions; you can use different names. Use the following template, adjusting the values as needed:

```
apiVersion: installation.mattermost.com/v1beta1 kind: Mattermost
metadata:  name: <INSTALLATION_NAME_HERE> # Example: mm-example-full
spec:  size: <SIZE_VALUE_HERE> # Example: 5000users  ingress:  enabled:
true  host: <FULL_DOMAIN_NAME_HERE> # Example: example.mattermost-
example.com  annotations:  kubernetes.io/ingress.class: nginx version:
<VERSION_HERE> # Example: 9.3.0 licenseSecret: "" # If you created a
license secret, put the name here
```

> *Key fields in the manifest include:*
>
> - `metadata.name`*: The name of your Mattermost deployment in Kubernetes.*
> - `spec.size`*: The size of your installation (e.g., "100users", "1000users", etc.).*
> - `spec.ingress.host`*: The DNS name for your Mattermost installation.*

> - `spec.version`: *The Mattermost version. See the server version archive for available versions.*
> - `spec.licenseSecret`: *The name of the Kubernetes secret containing your license (required for Enterprise).*
>
> *For a full list of configurable fields, see the [example manifest (https://github.com/mattermost/mattermost-operator/blob/master/docs/examples/mattermost_full.yaml)](https://github.com/mattermost/mattermost-operator/blob/master/docs/examples/mattermost_full.yaml) and the [Custom Resource Definition (https://github.com/mattermost/mattermost-operator/blob/master/config/crd/bases/installation.mattermost.com_mattermosts.yaml)](https://github.com/mattermost/mattermost-operator/blob/master/config/crd/bases/installation.mattermost.com_mattermosts.yaml).*

3. Create a file named `mattermost-database-secret.yaml` for database credentials. This secret must be in the same namespace as the Mattermost installation.

> ```
> apiVersion: v1 data:  DB_CONNECTION_CHECK_URL:
> <DB_CONNECTION_CHECK_URL>  DB_CONNECTION_STRING: <DB_CONNECTION_STRING>
> MM_SQLSETTINGS_DATASOURCEREPLICAS: <MM_SQLSETTINGS_DATASOURCEREPLICAS>
> kind: Secret metadata:  name: my-postgres-connection type: Opaque
> ```
>
> *Example for AWS Aurora with PostgreSQL:*
>
> ```
> apiVersion: v1 data:  DB_CONNECTION_CHECK_URL:
> cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
> GVyLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD9jb2
> 5uZWN0X3RpbWVvdXQ9MTAK  DB_CONNECTION_STRING:
> cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
> GVyLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD9jb2
> 5uZWN0X3RpbWVvdXQ9MTAK  MM_SQLSETTINGS_DATASOURCEREPLICAS:
> cG9zdGdyZXM6Ly91c2VyOnN1cGVyX3NlY3JldF9wYXNzd29yZEBteS1kYXRhYmFzZS5jbHVzd
> GVyLXJvLWFiY2QudXMtZWFzdC0xLnJkcy5hbWF6b25hd3MuY29tOjU0MzIvbWF0dGVybW9zdD
> 9jb25uZWN0X3RpbWVvdXQ9MTAK kind: Secret metadata:  name: my-postgres-
> connection type: Opaque
> ```

Step 4: Create the Filestore Secret

Create a file named `mattermost-filestore-secret.yaml` to store the credentials for your object storage service (e.g., AWS S3, MinIO). This secret must be created in the same namespace where you intend to install Mattermost. The file should contain the following YAML structure:

```
apiVersion: v1 kind: Secret metadata:  name: <secret-name> # Choose a
descriptive name (e.g., my-s3-credentials) type: Opaque data:  accesskey:
<base64-encoded-access-key>  secretkey: <base64-encoded-secret-key>
```

Key	Description	Required
accesskey	Base64-encoded access key for your storage service.	Yes
secretkey	Base64-encoded secret key for your storage service.	Yes
metadata.name	The name of the Kubernetes secret.	Yes

Important

The `accesskey` and `secretkey` values must be **base64-encoded**. Do not enter the raw keys directly. Use a command-line tool or online encoder to generate the base64 strings.

Example (AWS S3):

```
apiVersion: v1 kind: Secret metadata:  name: my-s3-credentials type: Opaque
data:  accesskey: QUNDRVNTX0tFWQo= # Example: Replace with your actual
encoded key  secretkey: U1VQRVJfU0VDUkVUX0tFWQo= # Example: Replace with
your actual encoded key
```

Step 5: Configure the Mattermost Installation Manifest

1. Modify the `mattermost-installation.yaml` file (created in step 2) to connect Mattermost to your external database and object storage. Refer to the supported fields for guidance on where to add these configurations within the YAML structure.

2. Connect to the database:

> *1. Add the following to the `spec` section of your manifest:*
>
> ```
> spec: database: external: secret: <database-secret-name> # The name of the database secret (e.g., my-postgres-connection)
> ```

3. Connect to Object Storage:

> *1. Add the following to the `spec` section of your manifest:*
>
> ```
> spec: fileStore: external: url: <storage-service-url> # The URL of your storage service (e.g., s3.amazonaws.com) bucket: <bucket-name> # The name of your storage bucket secret: <filestore-secret-name> # The name of the filestore secret (e.g., my-s3-credentials)
> ```

4. If you are using Amazon S3, it's recommended to enable server-side encryption (SSE) and SSL. Add the following environment variables to the `mattermostEnv` section:

> ```
> spec: mattermostEnv: MM_FILESETTINGS_AMAZONS3SSL: true
> MM_FILESETTINGS_AMAZONS3SSE: true
> ```

Review Mattermost Resource Status

After a Mattermost installation has been created with the Operator, you can review its status with the following:

```
kubectl -n [namespace] get mattermost
```

The `kubectl describe` command can be used to obtain more information about the Mattermost server pods:

```
kubectl -n [namespace] describe pod
```

Follow logs

The following command can be used to follow logs on any kubernetes pod:

```
kubectl -n [namespace] logs -f [pod name]
```

If the `-n [namespace]` is omitted, then the default namespace of the current context is used. We recommend specifying the namespace based on your deployment.

This command can be used to review the Mattermost Operator or Mattermost server logs as needed.

Note

- If you're new to Kubernetes or prefer a managed solution, consider using a service like Amazon EKS (https://aws.amazon.com/eks/), Azure Kubernetes Service (https://azure.microsoft.com/en-ca/products/kubernetes-service/), Google Kubernetes Engine (https://cloud.google.com/kubernetes-engine/), or DigitalOcean Kubernetes (https://www.digitalocean.com/products/kubernetes/).- While this guidance focuses on using external, managed services for your database and file storage, the Mattermost Operator *does* offer the flexibility to use other solutions. For example, you could choose to deploy a PostgreSQL database within your Kubernetes cluster using the CloudNative PG operator (or externally however you wish), or use a self-hosted MinIO instance for object storage.
- While using managed cloud services is generally simpler to maintain and our recommended approach for production deployments, using self-managed services like MinIO for storage and CloudNative PG for PostgreSQL are also valid options if you have the expertise to manage them.
- If you choose to use self-managed components, you'll need to adapt the instructions accordingly, pointing to your internal services instead.
- To customize your production deployment, refer to the configuration settings documentation.
- If you encounter issues during deployment, consult the deployment troubleshooting guide.

Remove Mattermost

Minimum system requirements:

- Operating System: Enterprise Linux 7+, Oracle Linux 6+, Oracle Linux 7+
- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)
- Database: [PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)](https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can deploy Mattermost Server using our `rpm` signed packages available through the Mattermost Yum repository.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, download the Mattermost server, install the server, set up the server, and update the server.

Step 1: Install PostgreSQL database or get database connection credentials

- Install PostgreSQL locally on the same server by following the [PostgreSQL installation (https://www.postgresql.org/download/)](https://www.postgresql.org/download/) documentation.
- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Download the latest Mattermost Server tarball

In a terminal window, ssh onto the system that will host the Mattermost Server. Using `wget`, download the Mattermost Server release you want to install using one of the following commands. Replace `amd64` with the appropriate architecture (e.g., `arm64` for ARM-based systems) in the link as needed.

Latest release

```
wget https://releases.mattermost.com/10.10.1/mattermost-10.10.1-linux-amd64.tar.gz
```

Current ESR

```
wget https://releases.mattermost.com/10.5.9/mattermost-10.5.9-linux-amd64.tar.gz
```

Older releases

If you are looking for an older release, Enterprise and Team Edition releases can be found in our version archive documentation.

Step 4: Install Mattermost server

1. Ahead of installing the Mattermost Server, we recommend updating all your repositories and, where required, update existing packages by running the following commands:

> ```
> sudo dnf update sudo dnf upgrade
> ```

2. After any updates, and any system reboots, are complete, install the Mattermost Server by extracting the tarball, creating users and groups, and setting file/folder permissions.

> *1. First extract the tarball:*
>
> > ```
> > tar -xvzf mattermost*.gz
> > ```
>
> *2. Now move the entire folder to the `/opt` directory (or whatever path you require):*

```
sudo mv mattermost /opt
```

3. *Create the default storage folder. By default the Mattermost Server uses `/opt/mattermost/data` as the folder for files. This can be changed in the System Console during setup (even using alternative storage such as S3):*

```
sudo mkdir /opt/mattermost/data
```

> Note
>
> If you choose a custom path, ensure this alternate path is used in all steps that follow.`

3. Set up a user and group called `mattermost`:

```
sudo useradd --system --user-group mattermost
```

> Note
>
> If you choose a custom user and group name, ensure it is used in all the steps that follow.

4. Set the file and folder permissions for your installation:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

5. Give the `mattermost` group write permissions to the application folder:

```
sudo chmod -R g+w /opt/mattermost
```

> *You will now have the latest Mattermost Server version installed on your system. Starting and stopping the Mattermost Server is done using* `systemd`.

6. Create the systemd unit file:

```
sudo touch /lib/systemd/system/mattermost.service
```

7. As root, edit the systemd unit file at `/lib/systemd/system/mattermost.service` to add the following lines:

```
[Unit] Description=Mattermost After=network.target [Service] Type=notify
ExecStart=/opt/mattermost/bin/mattermost TimeoutStartSec=3600
KillMode=mixed Restart=always RestartSec=10 WorkingDirectory=/opt/
mattermost User=mattermost Group=mattermost LimitNOFILE=49152 [Install]
WantedBy=multi-user.target
```

8. Save the file and reload systemd using `sudo systemctl daemon-reload`. Mattermost Server is now installed and is ready for setup.

Step 5: Set up the server

Before you start the Mattermost Server, you need to edit the configuration file. A default configuration file is located at `/opt/mattermost/config/config.json`. We recommend taking a backup of this default config ahead of making changes:

```
sudo cp /opt/mattermost/config/config.json /opt/mattermost/config/
config.defaults.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.
- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

> Note
>
> We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run sudo `systemctl enable mattermost.service` so that Mattermost will start on system boot. If you don't receive an error when starting Mattermost after the previous step, you are good to go. If you did receive an error, continue on.

Important

Modify SELinux settings: When deploying Mattermost from RHEL9, which has SELinux running with enforceing mode enabled by default, additional configuration is required.

- SELinux is a security module that provides access control security policies. It's enabled by default on RHEL and CentOS systems. SELinux can block access to files, directories, and ports, which can cause issues when starting Mattermost. To resolve these issues, you'll need to set the appropriate SELinux contexts for the Mattermost binaries and directories, and allow Mattermost to bind to ports.
- Ensure that SELinux is enabled and in enforcing mode by running the `sestatus` command. If it's `enforcing`, you'll need to configure it properly.
- Set bin contexts for `/opt/mattermost/bin`: SELinux enforces security contexts for binaries. To label the Mattermost binaries as safe, you'll need to set them to the below SELinux context.

```
sudo semanage fcontext -a -t bin_t "/opt/mattermost/bin(/.*)?" sudo
restorecon -RF /opt/mattermost/bin
```

Now, try starting Mattermost again with

```
sudo systemctl start mattermost
```

If you don't receive an error, verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server. You're all set!

If on starting Mattermost you receive an error, before moving on, check for the existence of a file in `/opt/mattermost/logs` - if `mattermost.log` exists in that directory, it's more likely you're dealing with a configuration issue in `config.json`. Double check the previous steps before continuing

Try different contexts for `/opt/mattermost`: SELinux enforces security contexts for files and directories. To label your Mattermost directory as safe, you'll need to set an appropriate SELinux context.

> 1. *Check current context by running `ls -Z /opt/mattermost`. When you see something like `drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 mattermost` returned, the `default_t` indicates that SELinux doesn't know what this directory is for.*
>
> 2. *Set a safe context by assigning a SELinux type that's compatible with web services or applications by running `sudo semanage fcontext -a -t httpd_sys_content_t "/opt/mattermost(/.*)?"`. A common one is `httpd_sys_content_t`, used for serving files. Ensure you match the directory and its contents recursively. Run the `sudo restorecon -R /opt/mattermost` to apply the changes.*

Allow Mattermost to bind to ports: When Mattermost needs specific ports (e.g., 8065), ensure that SELinux allows it by allowing Mattermost to bind to ports. Run the `sudo semanage port -l | grep 8065` command, and if the port's not listed, you'll need to add it by running `sudo semanage port -a -t http_port_t -p tcp 8065`, replacing the `8065` with the required port.

Handle custom policies: If Mattermost requires actions that SELinux blocks, you'll need to generate a custom policy.

> 1. *Check for SELinux denials first in the logs by running* `sudo ausearch -m avc -ts recent`*, or by checking the audit log:* `sudo cat /var/log/audit/audit.log | grep denied`*.*
> 2. *If needed, generate a policy module by installing* `audit2allow` *to generate policies automatically.*
>
> ```
> sudo yum install -y policycoreutils-python-utils sudo grep
> mattermost /var/log/audit/audit.log | audit2allow -M
> mattermost_policy sudo semodule -i mattermost_policy.pp
> ```

Test the configuration: Restart Mattermost to confirm the configuation works as expected by running `sudo systemctl restart mattermost`. In the case of failures, revisit the logs to identify other SELinux-related issues.

Need Mattermost working quickly for testing purposes?

- You can change SELinux to permissive mode by running the `sudo setenforce 0`. command where policies aren't enforced, only logged.
- This command changes the SELinux mode to "permissive". While in permissive mode, policies aren't enforced, and violations are logged instead of being blocked. This can be helpful for debugging and troubleshooting issues related to SELinux policies.
- Ensure you re-enable enforcing mode once context is working as needed by running the `sudo setenforce 1` command.

See the following SELinux resources for additional details:

> - *SELinux User's and Administrator's Guide (https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/index)*
> - *SELinux Project Wiki (https://github.com/SELinuxProject/selinux)*
> - *Introduction to SELinux (https://github.blog/developer-skills/programming-languages-and-frameworks/introduction-to-selinux/)*

> - *A Sysadmin's Guide to SELinux: 42 Answers to the Big Questions (https://opensource.com/article/18/7/sysadmin-guide-selinux)*
> - *Mastering SELinux: A Comprehensive Guide to Linux Security (https://srivastavayushmaan1347.medium.com/mastering-selinux-a-comprehensive-guide-to-linux-security-8bed9976da88)*

Step 6: Update the server

Updating your Mattermost Server installation when using the tarball requires several manual steps. See the upgrade Mattermost Server documentation for details.

To remove the Mattermost Server, you must stop the Mattermost Server, back up all important files, and then run this command:

```
sudo rm /opt/mattermost
```

> Note
>
> Depending on your configuration, there are several important folders in `/opt/mattermost` to backup. These are `config`, `logs`, `plugins`, `client/plugins`, and `data`. We strongly recommend you back up these locations before running the `rm` command.

You may also remove the Mattermost systemd unit file and the user/group created for running the application.

On this page

- Remove Mattermost
-
  - Latest release
  - Current ESR
  - Older releases
  - Remove Mattermost

Remove Mattermost

Minimum system requirements:

- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)
- Database: PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can install the Mattermost Server on any 64-bit Linux system using the tarball. This is the most flexible installation method, but it comes with the highest effort, typically favored by advanced system administrators.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, download the Mattermost server, install the server, and set up the server.

Step 1: Install PostgreSQL database or get database connection credentials

- Install PostgreSQL locally on the same server by following the PostgreSQL installation (https://www.postgresql.org/download/) documentation.
- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Download

In a terminal window, ssh onto the system that will host the Mattermost Server. Using `wget`, download the Mattermost Server release you want to install using one of the following commands. Replace `amd64` with the appropriate architecture (e.g., `arm64` for ARM-based systems) in the link as needed.

Latest release

```
wget https://releases.mattermost.com/10.10.1/mattermost-10.10.1-linux-amd64.tar.gz
```

Current ESR

```
wget https://releases.mattermost.com/10.5.9/mattermost-10.5.9-linux-amd64.tar.gz
```

Older releases

If you are looking for an older release, Enterprise and Team Edition releases can be found in our version archive documentation.

Step 4: Install Mattermost server

Install the Mattermost Server by extracting the tarball, creating users and groups, and setting file/folder permissions.

1. First extract the tarball:

```
tar -xvzf mattermost*.gz
```

2. Move the entire folder to the `/opt` directory (or whatever path you require):

```
sudo mv mattermost /opt
```

3. Create the default storage folder. By default the Mattermost Server uses `/opt/mattermost/data` as the folder for files. This can be changed in the System Console during setup (even using alternative storage such as S3):

```
sudo mkdir /opt/mattermost/data
```

> Note
>
> If you choose a custom path, ensure this alternate path is used in all steps that follow.

4. Set up a user and group called `mattermost`:

```
sudo useradd --system --user-group mattermost
```

> Note
>
> If you choose a custom user and group name, ensure it is used in all the steps that follow.

5. Set the file and folder permissions for your installation:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

6. Give the `mattermost` group write permissions to the application folder:

```
sudo chmod -R g+w /opt/mattermost
```

> *You will now have the latest Mattermost Server version installed on your system. Starting and stopping the Mattermost Server is done using `systemd`.*

7. Create the systemd unit file:

```
sudo touch /lib/systemd/system/mattermost.service
```

8. As root, edit the systemd unit file at `/lib/systemd/system/mattermost.service` to add the following lines:

```
[Unit] Description=Mattermost After=network.target [Service] Type=notify
ExecStart=/opt/mattermost/bin/mattermost TimeoutStartSec=3600
KillMode=mixed Restart=always RestartSec=10 WorkingDirectory=/opt/
mattermost User=mattermost Group=mattermost LimitNOFILE=49152 [Install]
WantedBy=multi-user.target
```

> *Note*
>
> *If you are installing the Mattermost server on the same system as your database, you may want to add both* `After=postgresql.service` *and* `BindsTo=postgresql.service` *to the* `[Unit]` *section of the systemd unit file.*

9. Save the file and reload systemd using `sudo systemctl daemon-reload`. Mattermost Server is now installed and is ready for setup.

Step 5: Set up the server

Before you start the Mattermost Server, you need to edit the configuration file. A default configuration file is located at `/opt/mattermost/config/config.json`. We recommend taking a backup of this default config ahead of making changes:

```
sudo cp /opt/mattermost/config/config.json /opt/mattermost/config/
config.defaults.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?`

`sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.

- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: `curl http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run sudo `systemctl enable mattermost.service` so that Mattermost will start on system boot.

Step 6: Update the server

Updating your Mattermost Server installation when using the tarball requires several manual steps. See the upgrade Mattermost Server documentation for details.

To remove the Mattermost Server for any reason, you must stop the Mattermost Server, back up all important files, and then run this command:

```
sudo rm - rf /opt/mattermost
```

> Note
>
> Depending on your configuration, there are several important folders in `/opt/mattermost` to backup. These are `config`, `logs`, `plugins`, `client/plugins`, and `data`. We strongly recommend you back up these locations before running the `rm` command.

You may also remove the Mattermost systemd unit file and the user/group created for running the application.

On this page

- Remove Mattermost
-
  - Latest release
  - Current ESR
  - Older releases
  - Remove Mattermost

Remove Mattermost

Minimum system requirements:

- Operating System: 20.04 LTS, 22.04 LTS, 24.04 LTS
- Hardware: 1 vCPU/core with 2GB RAM (support for up to 1,000 users)
- Database: PostgreSQL v13+ (https://docs.mattermost.com/deployment-guide/postgres-migration.html)
- Network:
  - Application 80/443, TLS, TCP Inbound
  - Administrator Console 8065, TLS, TCP Inbound
  - SMTP port 10025, TCP/UDP Outbound

You can deploy Mattermost server using our `.deb` signed packages using the Mattermost PPA (Personal Package Archive). This is the quickest way to install a Mattermost Server that provides automatic updates. This install method is used for both single and clustered installations, as you can tools like Packer for a clustered deployment.

This Mattermost deployment includes the following steps: install PostgreSQL database, prepare the database, add the PPA repository, install Mattermost server, configure the server, and update the server.

Step 1: Install PostgreSQL database or get database connection credentials

Mattermost requires a PostgreSQL database. You can either:

- Install PostgreSQL locally on the same server by following the PostgreSQL installation (https://www.postgresql.org/download/) documentation.
- Use an external PostgreSQL database server. Ensure you have connection credentials, including hostname, port, database name, username, and password available.
- Use a managed database service.

Step 2: Prepare the database

Follow the database preparation documentation to set up your PostgreSQL database for Mattermost.

Step 3: Add the Mattermost Server PPA repository

Important

The GPG public key has changed. You can [import the new public key (https://deb.packages.mattermost.com/pubkey.gpg)](https://deb.packages.mattermost.com/pubkey.gpg) or run the automatic Mattermost PPA repository setup script provided below. Depending on your setup, additional steps may also be required, particularly for installations that didn't rely on the repository setup script. We recommend deleting the old key from `/etc/apt/trusted.gpg.d` before adding the apt repository.

- For Ubuntu Jammy - 22.04 LTS and Ubuntu Noble - 24.04 LTS:

  `sudo rm /usr/share/keyrings/mattermost-archive-keyring.gpg`

  `curl -sL -o- https://deb.packages.mattermost.com/pubkey.gpg | gpg --dearmor | sudo tee /usr/share/keyrings/mattermost-archive-keyring.gpg > /dev/null`

In a terminal window, run the following repository setup command to add the Mattermost Server repositories:

```
curl -o- https://deb.packages.mattermost.com/repo-setup.sh | sudo bash -s
mattermost
```

This command configures the repositories needed for a PostgreSQL database, configures an NGINX web server to act as a proxy, configures certbot to issue and renew the SSL certificate, and configures the Mattermost repository so that you can run the install command.

Step 4: Install Mattermost server

Ahead of installing the Mattermost Server, it's good practice to update all your repositories and, where required, update existing packages by running the following command:

```
sudo apt update
```

After any updates and system reboots are complete, you can install the Mattermost Server by running:

```
sudo apt install mattermost -y
```

You now have the latest Mattermost Server version installed on your system.

The installation path is `/opt/mattermost`. The package will have added a user and group named `mattermost`. The required systemd unit file has also been created but will not be set to active.

> Note
>
> Since the signed package from the Mattermost repository is used for mulitple installation types, we don't add any dependencies in the systemd unit file. If you are installing the Mattermost server on the same system as your database, you may want to add both `After=postgresql.service` and `BindsTo=postgresql.service` to the `[Unit]` section of the systemd unit file.

Step 5: Configure the server

Before you start the Mattermost Server, you need to edit the configuration file. A sample configuration file is located at `/opt/mattermost/config/config.defaults.json`.

Rename this configuration file with correct permissions:

```
sudo install -C -m 600 -o mattermost -g mattermost /opt/mattermost/config/
config.defaults.json /opt/mattermost/config/config.json
```

Configure the following properties in this file:

- Under `SqlSettings`, set `DriverName` to `"postgres"`. This is the default and recommended database for all Mattermost installations.
- Under `SqlSettings`, set `DataSource` to `"postgres://mmuser:<mmuser-password>@<host-name-or-IP>:5432/mattermost?sslmode=disable&connect_timeout=10"` replacing `mmuser`, `<mmuser-password>`, `<host-name-or-IP>` and `mattermost` with your database name.

- Under `ServiceSettings`, set `"SiteURL"`: The domain name for the Mattermost application (e.g. `https://mattermost.example.com`).

We recommend configuring the [Support Email (https://docs.mattermost.com/administration/config-settings.html#support-email)](https://docs.mattermost.com/administration/config-settings.html#support-email) under `SupportSettings`, set `"SupportEmail"`. This is the email address your users will contact when they need help.

After modifying the `config.json` configuration file, you can now start the Mattermost Server:

```
sudo systemctl start mattermost
```

Verify that Mattermost is running: curl `http://localhost:8065`. You should see the HTML that's returned by the Mattermost Server.

The final step, depending on your requirements, is to run `sudo systemctl enable mattermost.service` so that Mattermost will start on system boot.

> Note
>
> The value of the `sslmode` property in the `DataSource` configuration is entirely dependent on your native environment. Please consult the native environment setup documentation for guidance on its value. The available options for `sslmode` are `disable` or `require`. For example, if you are using Amazon Lightsail as your data source, you must set `sslmode` to `require` to successfully connect to the database.

Step 6: Update the server

When a new Mattermost version is released, run: `sudo apt update && sudo apt upgrade` to download and update your Mattermost instance.

> Note
>
> When you run the `sudo apt upgrade` command, `mattermost-server` will be updated along with any other packages. We strongly recommend you stop the Mattermost Server before running the `apt` command using `sudo systemctl stop mattermost`.

Run the following command to remove the Mattermost Server:

```
sudo apt remove --purge mattermost
```

Deployment Solution Programs

Mattermost's **Deployment Solutions Programs** help IT administrators understand how Mattermost is being offered in third-party deployment solutions, including other open source projects as well as in commercial solutions.

This is an optional program for third-party developers to increase awareness about their work and to enable Mattermost to refer its communities to different solutions.

Individuals or companies interested in exploring deployment solutions with Mattermost are encouraged to reach out to us via the Mattermost Partner Program Form (https://support.mattermost.com/hc/en-us/requests/new?ticket_form_id=24278352368532).

Deployment solutions are recognized by Mattermost at three-levels:

- **Community Deployment Solutions** - Basic orchestration solutions for Mattermost, typically from the open source community or hosting companies who let us know about their work. The work of partners is included in blog posts and on social media as a benefit to the user and customer communities.

  > *Examples:*
  >
  > - *Puppet deployment solution for Mattermost (https://forge.puppet.com/liger1978/mattermost) by Richard Grainger*
  > - *Heroku deployment solution for Mattermost (https://chrisdecairos.ca/deploying-mattermost-to-heroku/) by Christopher De Cairos*

- **Registered Deployment Solutions** - Orchestration solutions that follow detailed Mattermost guidelines on keeping up-to-date with the latest Mattermost version and security updates (typically a one-line change), linking to official documentation, supporting branding guidelines, and maintaining a changelog. This level of engagement allows Mattermost to more prominently promote the work, knowing that it's committed to meeting explicit standards.

- **Certified Deployment Solutions** - These solutions meet all the requirements of Registered Deployment Solutions, with the addition that they'll automate the version upgrade and security update processes (typically a one-line change). There is the added benefit that when Mattermost announces new versions and security updates, we can also announce the availability of updates to Certified Deployment Solutions.

To summarize the commitment level of different solutions:

Deployment Solution Requirement	Community	Registered	Certified
Installation	✓	✓	✓
Minimum Documentation	✓	✓	✓
Security Updates		✓	✓
Branding		✓	✓
Upgrade			✓

Requirement details are outlined below.

Deployment solution program requirements

Installation

1. **Installation is designed for officially supported operating systems and platforms**. EXCEPTION: RHEL equivalents (CentOS, Amazon Linux, Oracle Linux, Scientific Linux) are acceptable as long as the exception is noted in README or equivalent with `This deployment uses [OPERATING_SYSTEM]`

`as an equivalent to the officially supported version of Red Hat Enterprise Linux.`

2. **Automated installation passes basic testing**. After installation run the following manual tests:

> *1. Create a new user account and use that account to create a new team and post to Town Square channel.*
>
> *2. Create a second user account and join the newly created team and reply to first user's post in Town Square.*
>
> *3. Go back to first user account and post reply with an image attached.*
>
> *4. Confirm there are no errors and no red alert bar at the top of the screen with "Mattermost unreachable error" (which would indicate a websocket configuration error).*

Documentation

1. **Include link to official Mattermost documentation**. README or equivalent contains statement `Please see https://docs.mattermost.com for official documentation.`

> *Include the following information in text, markdown, HTML or other format. Square brackets `[]` indicate optional statements depending on the configuration of your solution:*
>
> > *This deployment solution installs [and upgrades] a Mattermost server to provide secure, private cloud messaging for teams and enterprises. More information is available at: https://mattermost.com*
> >
> > *Following automated deployment, the following steps are required to make your system production-ready:*
> >
> > - *[Configure SSL for Mattermost](https://docs.mattermost.com/deployment-guide/server/setup-nginx-proxy.html#configure-nginx-with-ssl-and-http-2)*

> - *[Configure SMTP email for Mattermost](https://docs.mattermost.com/administration-guide/configure/smtp-email.html)*

2. **Unofficial deployment options should be documented**. Unofficial deployment configurations, such as use of Linux operating systems that are not officially supported, should be documented in the README.

Security updates

1. **Document commitment to providing security updates.** README or equivalent states `We highly recommend users subscribe to the Mattermost security updates email list. When notified of a security update, the maintainers of this deployment solution will make an effort to update to the secure version within 10 days.`

2. **Commit to making an effort for your deployment to provide the latest security update within 10 days of announcement**. This is typically a one-line change.

Branding

Support Mattermost branding and naming guidelines. To ensure naming is clear across deployment solutions, README or equivalent should contain `The name for this deployment solution in the context of [Mattermost branding guidelines](https://handbook.mattermost.com/operations/operations/publishing/publishing-guidelines/brand-and-visual-design-guidelines) is `[NAME] for Mattermost by [CREATOR]`.` For example, `Multi-node Docker deployment solution for Mattermost by John Doe`. This is the name that will be used to refer to your work in Mattermost community materials.

Upgrade

Support upgrade of Mattermost. Enable user interface or command line upgrade of a Mattermost deployment to latest version based on [upgrade procedure when Mattermost is embedded (https://developers.mattermost.com/integrate/faq/#how-should-i-automate-the-install-and-upgrade-of-mattermost-when-included-in-another-application)](https://developers.mattermost.com/integrate/faq/#how-should-i-automate-the-install-and-upgrade-of-mattermost-when-included-in-another-application).

On this page

- Deployment Solution Programs
  - Deployment solution program requirements
    - Installation
    - Documentation
    - Security updates
    - Branding
    - Upgrade

Prepare your Mattermost Server environment

This guide outlines the key preparation steps required before installing the Mattermost Server, focusing on setting up the database and file storage systems.

Before installing Mattermost Server, review the following preparation requirements:

- Review software and hardware requirements - Ensure your system meets the minimum requirements for Mattermost deployment.
- Set up an NGINX proxy - Configure NGINX as a reverse proxy for enhanced security and performance.
- Configure Mattermost Calls - Set up real-time communication capabilities for voice and video calls.
- Set up TLS - Enable secure communication with SSL/TLS encryption.
- Use an image proxy - Configure image proxy for enhanced privacy and security.

Database preparation

PostgreSQL v13+ is required for Mattermost server installations. MySQL database support is being deprecated starting with Mattermost v11. See the PostgreSQL migration documentation for guidance on migrating from MySQL to PostgreSQL.

1. Create an PostgreSQL server instance. See the PostgreSQL documentation for details. When the installation is complete, the PostgreSQL server is running, and a Linux user account called postgres has been created.
2. Create the Mattermost database and user:
    1. Access PostgreSQL by running:

```
sudo -u postgres psql
```

2. Create the database:

```
CREATE DATABASE mattermost WITH ENCODING 'UTF8'
LC_COLLATE='en_US.UTF-8' LC_CTYPE='en_US.UTF-8' TEMPLATE=template0;
```

If this steps fails with an error message like `invalid LC_COLLATE locale name: "en_US.UTF-8"`, you need to generate the locale first using `locale-gen en_US.UTF-8`.

3. Create the Mattermost user with a secure password:

```
CREATE USER mmuser WITH PASSWORD 'mmuser-password';
```

4. Grant database access to the user:

```
GRANT ALL PRIVILEGES ON DATABASE mattermost to mmuser;
```

5. If using PostgreSQL v15.x or later, additional grants are required:

```
ALTER DATABASE mattermost OWNER TO mmuser; ALTER SCHEMA public OWNER
TO mmuser; GRANT USAGE, CREATE ON SCHEMA public TO mmuser;
```

3. Configure PostgreSQL for remote connections (if database is on a separate server):

1. Edit `postgresql.conf` to allow remote connections:

   Ubuntu/Debian

   Edit `/etc/postgresql/{version}/main/postgresql.conf`:

```
listen_addresses = '*'
```

   RHEL/CentOS

   Edit `/var/lib/pgsql/{version}/data/postgresql.conf`:

```
listen_addresses = '*'
```

2. Configure client authentication by editing `pg_hba.conf`:

   Add the following line, replacing `{mattermost-server-IP}`:

   ```
   host all all {mattermost-server-IP}/32 md5
   ```

4. Restart the PostgreSQL service to apply the configuration changes:

   Ubuntu/Debian

   ```
   sudo systemctl restart postgresql
   ```

   RHEL/CentOS

   ```
   sudo systemctl restart postgresql
   ```

Important

If you are upgrading a major version of Postgres, ensure that `ANALYZE VERBOSE` is run on the database post upgrade. This is required to re-populate the `pg_statistics` table used to generate optimal query plans. Database performance may suffer if this step is skipped.

Once you've completed the database preparation, return to the Linux deployment documentation to continue with your Mattermost server installation.

File storage preparation

Mattermost requires a file storage system for storing user files, images, and attachments. You have several options, including:

- S3-compatibile object storage (recommended)
- local file storage

S3-compatible object storage (Recommended)

For production environments, we recommend using S3-compatible object storage such as:

- Amazon S3
- MinIO
- Digital Ocean Spaces
- Other S3-compatible services

When using S3 storage, you'll need:

1. A bucket created specifically for Mattermost
2. Access credentials (Access Key and Secret Key)
3. Appropriate bucket policies configured
4. The following information for configuration:
    - Bucket name
    - Region (if applicable)
    - Access Key
    - Secret Key
    - Endpoint URL (for non-AWS S3 services)

Local file storage

For simple deployments, you can use local file storage. However, we don't recommend this for production environments or multi-node deployments.

1. Create a directory for file storage:

    ```
    sudo mkdir -p /opt/mattermost/data
    ```

2. Set appropriate permissions:

```
sudo chown -R mattermost:mattermost /opt/mattermost/data
```

(Optional) Use an image proxy

Using an image proxy means that all requests for images made by Mattermost clients will go through the proxy instead of contacting third-party servers directly. This helps protect user privacy by preventing third-party servers from tracking who views an image. This also prevents the use of tracking pixels (invisible images that do the same thing without the user even seeing an image).

Certain proxy servers also provide a layer of caching which can make loading images faster and more reliable. This caching also helps preserve posts by protecting them from dead images.

Network preparation

The following table outlines the network ports and protocols required for Mattermost server:

Service Name	Config Setting	Port (default)	Protocol	Direction	Info
HTTP/ Websocket	ServiceSettings.ListenAddress	8065/80/443 (TLS)	TCP	Inbound	Exter proxy Inter proxy this port 443 runni
Cluster	ClusterSettings.GossipPort	8074	TCP/UDP	Inbound	Inter
Metrics	MetricsSettings.ListenAddress	8067	TCP	Inbound	Exter proxy Inter proxy
Database	SqlSettings.DataSource	5432 (PostgreSQL) / 3306 (MySQL)	TCP	Outbound	Usual (reco
LDAP	LdapSettings.LdapPort	389	TCP/UDP	Outbound	
S3 Storage	FileSettings.AmazonS3Endpoint	443 (TLS)	TCP	Outbound	
SMTP	EmailSettings.SMTPPort	10025	TCP/UDP	Outbound	
Push Notifications	EmailSettings.PushNotificationServer	443 (TLS)	TCP	Outbound	

> Note
>
> - All outbound ports may vary based on your specific configuration
> - Mattermost can be configured to use an outbound proxy for any HTTP/HTTPS traffic (see below)
> - Calls service may require additional ports

Outbound proxy configuration

If your deployment requires using an outbound proxy, you can configure Mattermost using environment variables:

1. Configure the proxy settings in your service configuration:

```
Environment=HTTP_PROXY=http://proxy.example.com:3128
Environment=HTTPS_PROXY=https://proxy.example.com:3128
Environment=NO_PROXY=localhost,127.0.0.1,.internal.example.com
```

2. For authenticated proxies, include credentials in the URL:

```
Environment=HTTP_PROXY=http://username:password@proxy.example.com:3128
Environment=HTTPS_PROXY=https://username:password@proxy.example.com:3128
```

3. The `NO_PROXY` variable can include: - IP addresses (e.g., `1.2.3.4`) - CIDR ranges (e.g., `1.2.3.4/8`) - Domain names (e.g., `example.com`) - Subdomains (e.g., `.example.com`)

Note

When using an HTTPS proxy, ensure your Mattermost server has the proxy's root certificate configured to avoid connection issues.

Example systemd Service Configuration

```
[Unit] Description=Mattermost After=network.target
After=postgresql.service BindsTo=postgresql.service [Service]
Type=notify ExecStart=/opt/mattermost/bin/mattermost
TimeoutStartSec=3600 KillMode=mixed Restart=always RestartSec=10
WorkingDirectory=/opt/mattermost User=mattermost Group=mattermost
LimitNOFILE=49152 # Configure proxy settings if needed
#Environment=HTTP_PROXY=http://proxy.example.com:3128
#Environment=HTTPS_PROXY=https://proxy.example.com:3128
#Environment=NO_PROXY=localhost,127.0.0.1,.internal.example.com #
Recommended security options ProtectSystem=full PrivateTmp=true
NoNewPrivileges=true [Install] WantedBy=postgresql.service
```

System requirements

Ensure your system meets these minimum requirements:

- Operating System: 64-bit Linux distribution
- Hardware: 1 vCPU/core with 2GB RAM (supports up to 1,000 users)
- Storage: Minimum 10GB available space
- Database: PostgreSQL v13+
- Network: Reliable internet connection with sufficient bandwidth

See the software and hardware requirements documentation for additional requirements.

Next steps

Once you've completed these preparation steps, you can proceed with installing the Mattermost server. Choose your preferred installation method:

- Deploy with Kubernetes
- Deploy on Linux
- Deploy with Containers

On this page

- Prepare your Mattermost Server environment
- Prepare your Mattermost Server environment
  - Database preparation
  - File storage preparation
    - S3-compatible object storage (Recommended)
    - Local file storage
    - (Optional) Use an image proxy
  - Network preparation
    - Outbound proxy configuration
  - System requirements
  - Next steps
  - Database preparation
  - File storage preparation
    - S3-compatible object storage (Recommended)
    - Local file storage
    - (Optional) Use an image proxy
  - Network preparation
    - Outbound proxy configuration
  - System requirements
  - Next steps

Prepare your Mattermost MySQL database

Important

- PostgreSQL is our preferred database of choice. See the database software documentation for details on database version support, and see the Migrate from MySQL to PostgreSQL documentation for details on migrating from MySQL to PostgreSQL.
- MySQL 8.0.22 contains an [issue with JSON column types (https://bugs.mysql.com/bug.php?id=101284)](https://bugs.mysql.com/bug.php?id=101284) changing string values to integers which is preventing Mattermost from working properly. Users are advised to avoid this database version.

Set up the Mattermost MySQL database

To set up a MySQL database for use by the Mattermost server:

1. Log into the server that will host the database, ands install MySQL.
2. Log in to MySQL as *root* by running `sudo mysql`.
3. Create the Mattermost user *mmuser* by running `mysql> create user 'mmuser'@'%' identified by 'mmuser-password';`.

> - *Use a password that is more secure than* `mmuser-password`*.*
> - *The* `%` *means that* `mmuser` *can connect from any machine on the network. However, it's more secure to use the IP address of the machine that hosts Mattermost. For example, if you install Mattermost on the machine with IP address* `10.10.10.2`*, then use the following command:* `mysql> create user 'mmuser'@'10.10.10.2' identified by 'mmuser-password';`

4. Create the Mattermost database by running `mysql> create database mattermost;`.

5. Grant access privileges to the user `mmuser` by running `mysql> grant all privileges on mattermost.* to 'mmuser'@'%';`.

> *Note*
>
> *This query grants the MySQL user we just created all privileges on the database for convenience. If you need more security, use the following query to grant the user only the privileges necessary to run Mattermost:* `mysql> GRANT ALTER, CREATE, DELETE, DROP, INDEX, INSERT, SELECT, UPDATE, REFERENCES ON mattermost.* TO 'mmuser'@'%';`

6. Log out of MySQL by running `mysql> exit`. Once the database is installed and the initial setup is complete, you can install the Mattermost server.

> Note
>
> If you have installed MySQL on its own server, edit the `/etc/mysql/mysql.conf.d/mysqld.cnf` file and comment out the `bind-address = 127.0.0.1` using the `#` symbol, then restart your database server.

Back up the database

Back up your Mattermost database using standard procedures depending on your database version. MySQL backup documentation <https://dev.mysql.com/doc/refman/8.4/en/backup-types.html>`_ is available online. Use the selector on the page to choose your MySQL version.

Upgrade Mattermost

Upgrade to v7.1

Mattermost v7.1 introduces schema changes in the form of a new column and its index. Our test results for the schema changes include: 12M Posts, 2.5M Reactions - ~1min 34s (instance: PC with 8 cores, 16GB RAM).

You can run the following SQL queries before the upgrade that obtains a lock on `Reactions` table.

```
ALTER TABLE Reactions ADD COLUMN ChannelId varchar(26) NOT NULL DEFAULT "";
```

```
UPDATE Reactions SET ChannelId = COALESCE((select ChannelId from Posts where Posts.Id = Reactions.PostId), '') WHERE ChannelId="";
```

```
CREATE INDEX idx_reactions_channel_id ON Reactions(ChannelId) LOCK=NONE;
```

Users' reactions posted during this time won't be reflected in the database until the migrations are complete. This is fully backwards-compatible.

If your connection collation and table collations are different, this can result in the error *Illegal mix of collations*. To resolve this error, set the same collation for both the connection and the table. There are different collations at different levels - connection, database, table, column, and database administrators may choose to set different collation levels for different objects.

Upgrade to v7.0

Self-hosted Mattermost customers using MySQL databases may notice the migration to release v7.0 taking longer than usual when there are a large number of rows in the `FileInfo` table. See the important upgrade notes documentation for details.

Upgrade to v6.7

Mattermost v6.7 introduces schema changes in the form of a new index. The following notes our test results for the schema changes:

- 7M Posts - ~17s (instance: db.r5.xlarge)
- 9M Posts - 2min 12s (instance: db.r5.large)

If you want a zero downtime upgrade, you can apply this index prior to doing the upgrade:

```
CREATE INDEX idx_posts_create_at_id on Posts(CreateAt, Id) LOCK=NONE;
```

This is fully backwards-compatible and will not acquire any table lock or affect any existing operations on the table.

v6.0 database schema migrations

The release of v6.0 introduces database schema changes and longer migration times should be expected, especially on MySQL installations.

Mattermost v6.0 introduces several database schema changes to improve both database and application performance. The upgrade will run significant database schema changes that can cause an extended startup time depending on the dataset size. We've conducted extensive tests on supported MySQL database drivers, using realistic datasets of more than 10 million posts and more than 72 million posts.

A migration to v6.0 of 10+ million posts will take approximately 1 hour and 22 minutes to complete for a MySQL database.

A large migration from v5.39 to v6.0 of 72+ million posts will take approximately 3 hours and 40 minutes to complete for a MySQL database. See the [Migration results analysis (https://gist.github.com/streamer45/868c451164f6e8069d8b398685a31b6e)](https://gist.github.com/streamer45/868c451164f6e8069d8b398685a31b6e) documentation for test specifications, data sizes, and test results.

The following queries, executed during the migration process on an environment with 10+ million posts, will have a significant impact on database CPU usage and write operation restrictions for the duration of the query:

`ALTER TABLE Posts MODIFY Props JSON;` (~26 minutes)

`ALTER TABLE Posts DROP COLUMN ParentId;` (~26 minutes)

`ALTER TABLE Posts MODIFY COLUMN FileIds text;` (~26 minutes)

See the [Mattermost v6.0 DB schema migrations analysis (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055)](https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055) documentation for test specifications, data sizes, test results, and a complete breakdown of MySQL queries, as well as their impact and duration.

MySQL Mitigation Strategies

Run combined queries prior to the upgrade. The previous queries can be combined when run prior to the upgrade as follows:

```
ALTER TABLE Posts MODIFY COLUMN FileIds text, MODIFY COLUMN Props JSON;
```

This limits the time taken to that of a single query of that type.

Online migration: An online migration that avoids locking can be attempted on MySQL installations, especially for particularly heavy queries or very big datasets (tens of millions of posts or more). This can be done through an external tool like [pt-online-schema-change (https://docs.percona.com/percona-toolkit/pt-online-schema-change.html)](https://docs.percona.com/percona-toolkit/pt-online-schema-change.html). However, the online migration process can cause a significant spike in CPU usage on the database instance it runs.

See the [Mattermost v6.0 DB schema migrations analysis (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#online-migration-mysql)](https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#online-migration-mysql) documentation for a sample execution and additional caveats.

High availabiilty configuration setting recommendations

For MySQL, we recommend the following configuration options for high performance:

- `innodb_buffer_pool_size`: Set to ~70% of your total RAM.
- `innodb_log_file_size`: Set to 256 MB. Increasing this helps in write intensive operations. Recovery times will be longer.
- `innodb_flush_log_at_trx_commit`: 2. This can potentially cause up to one second of loss of transaction data.
- `max_heap_table_size`: 64 MB.
- `tmp_table_size`: 64 MB.

Encryption-at-rest

Encryption-at-rest is available for messages via hardware and software disk encryption solutions applied to the Mattermost database, which resides on its own server within your infrastructure. See the [MySQL (https://www.percona.com/blog/mysql-data-at-rest-encryption/)](https://www.percona.com/blog/mysql-data-at-rest-encryption/) database documentation for details on encryption options at the disk level.

Use sockets for the database

```
mysql -u root -p
```

```
CREATE DATABASE mattermostdb; CREATE USER mmuser IDENTIFIED BY
'mmuser_password'; GRANT ALL ON mattermostdb.* TO mmuser;
```

Mattermost is configured in `/etc/webapps/mattermost/config.json`, and strings need to be quoted.

- set `DriverName` to `mysql`.
- set `DataSource` to `mmuser:mmuser_password@unix(/run/mysqld/mysqld.sock)/mattermostdb?charset=utf8mb4,utf8`.

Mattermost configuration in the database

You can use the database as the single source of truth for the active configuration of your Mattermost installation. This changes the Mattermost binary from reading the default `config.json` file to reading the configuration settings stored within a configuration table in the database.

```
mysql://mmuser:really_secure_password@tcp(127.0.0.1:3306)/mattermost?
charset=utf8mb4,utf8&writeTimeout=30s
```

Create an environment file

> Note
>
> If you're running Mattermost in a High Availability cluster-based deployment, this step must be done on all servers in the cluster.

Create the file `/opt/mattermost/config/mattermost.environment` to set the `MM_CONFIG` environment variable to the database connection string. For example:

```
MM_CONFIG='mysql://mmuser:mostest@tcp(127.0.0.1:3306)/mattermost?
charset=utf8mb4,utf8&writeTimeout=30s'
```

> Note
>
> Be sure to escape any single quotes in the database connection string by placing a `\` in front of them like this `\'`. For example: `MM_CONFIG='mysql://mmuser:it\'s-a-password!@tcp(127.0.0.1:3306)/mattermost?charset=utf8mb4,utf8&writeTimeout=30s'`

```
MM_CONFIG='mysql://mmuser:it\'s-a-password!@tcp(127.0.0.1:3306)/mattermost?
charset=utf8mb4,utf8&writeTimeout=30s'
```

Finally, run this command to verify the permissions on your Mattermost directory:

```
sudo chown -R mattermost:mattermost /opt/mattermost
```

Modify the Mattermost `systemd` file

First, find the `mattermost.service` file using:

```
sudo systemctl status mattermost.service
```

The second line of output will have the location of the running `mattermost.service`.

```
Loaded: loaded (/lib/systemd/system/mattermost.service; enabled; vendor
preset: enabled)
```

Edit this file as *root* to add the below text just above the line that begins with `ExecStart`:

```
EnvironmentFile=/opt/mattermost/config/mattermost.environment
```

Here's a complete `mattermost.service` file with the `EnvironmentFile` line added:

```
[Unit] Description=Mattermost After=network.target After=mysql.service
Requires=mysql.service [Service] Type=notify EnvironmentFile=/opt/
mattermost/config/mattermost.environment ExecStart=/opt/mattermost/bin/
mattermost TimeoutStartSec=3600 KillMode=mixed Restart=always RestartSec=10
WorkingDirectory=/opt/mattermost User=mattermost Group=mattermost
LimitNOFILE=49152 [Install] WantedBy=mysql.service
```

Technical notes about searching

By default, Mattermost uses full text search support included in MySQL. Select the **product menu**

then select **About Mattermost** to see which database you're using.

- Stop words are filtered out of search results. See [MySQL (https://dev.mysql.com/doc/refman/5.7/en/fulltext-stopwords.html)](https://dev.mysql.com/doc/refman/5.7/en/fulltext-stopwords.html) database documentation for a full list of applicable stop words.
- Hashtags or recent mentions of usernames containing a dot don't return results.
- Avoid using underline `_` symbol to perform a wildcard search. Use the asterisk `*` symbol instead.
- Stop words that are excluded from search in MySQL include: `"a", "about", "an", "are", "as", "at", "be", "by", "com", "de", "en", "for", "from", "how", "i", "in", "is", "it", "la", "of", "on", "or", "that", "the", "this", "to", "was", "what", "when", "where", "who", "will", "with", "und", "the", "www"`.

Perform searches in Chinese, Korean, and Japanese

The best experience for searching in Chinese, Korean, and Japanese is to use MySQL 5.7.6 or later with special configuration. See the Chinese, Japanese and Korean Search documentation for details.

You can perform searches without this configuration by adding wildcards `*` to the end of search terms.

Migrate from Bitnami to a self-hosted Mattermost deployment

If you're planning a migration from Bitnami to a self-hosted Mattermost installation with a MySQL database, read these notes in our migration guide: Migrating from Bitnami.

Downgrade Mattermost v6.0 to v5.38

```
INSERT INTO Systems (Name,Value) VALUES ('Version','5.38.0') ON DUPLICATE
KEY UPDATE Value = '5.38.0'; CREATE INDEX idx_status_status ON Status
(Status); DROP INDEX idx_status_status_dndendtime ON Status; CREATE INDEX
idx_channelmembers_user_id ON ChannelMembers (UserId); DROP INDEX
idx_channelmembers_channel_id_scheme_guest_user_id ON ChannelMembers; DROP
INDEX idx_channelmembers_user_id_channel_id_last_viewed_at ON
ChannelMembers; CREATE INDEX idx_threads_channel_id ON Threads (ChannelId);
DROP INDEX idx_threads_channel_id_last_reply_at ON Threads; CREATE INDEX
idx_channels_team_id ON Channels (TeamId); DROP INDEX
idx_channels_team_id_type ON Channels; DROP INDEX
idx_channels_team_id_display_name ON Channels; CREATE INDEX
idx_posts_root_id ON Posts (RootId); DROP INDEX idx_posts_root_id_delete_at
ON Posts; ALTER TABLE CommandWebhooks ADD COLUMN ParentId varchar(26);
UPDATE CommandWebhooks SET ParentId = ''; ALTER TABLE Posts ADD COLUMN
ParentId varchar(26); UPDATE Posts SET ParentId = ''; ALTER TABLE Users
MODIFY Timezone text; ALTER TABLE Users MODIFY NotifyProps text; ALTER
TABLE Users MODIFY Props text; ALTER TABLE Threads MODIFY Participants
longtext; ALTER TABLE Sessions MODIFY Props text; ALTER TABLE Posts MODIFY
Props text; ALTER TABLE Jobs MODIFY Data text; ALTER TABLE LinkMetadata
MODIFY Data text; ALTER TABLE ChannelMembers MODIFY NotifyProps text;
```

> Note
>
> The inverse of [the final v6.0 upgrade query (https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#mysql-1)](https://gist.github.com/streamer45/59b3582118913d4fc5e8ff81ea78b055#mysql-1) is intentionally omitted from these downgrade queries because its result is backwards compatible, and running the query would unnecessarily delay the downgrade process.

On this page

- Prepare your Mattermost MySQL database
- Prepare your Mattermost MySQL database
  - Set up the Mattermost MySQL database
  - Back up the database
  - Upgrade Mattermost
    - Upgrade to v7.1
    - Upgrade to v7.0
    - Upgrade to v6.7
    - v6.0 database schema migrations
      - MySQL Mitigation Strategies
  - High availabiilty configuration setting recommendations
  - Encryption-at-rest
  - Use sockets for the database
  - Mattermost configuration in the database
    - Create an environment file
    - Modify the Mattermost systemd file
  - Technical notes about searching
    - Perform searches in Chinese, Korean, and Japanese
  - Migrate from Bitnami to a self-hosted Mattermost deployment
  - Downgrade Mattermost v6.0 to v5.38
  - Set up the Mattermost MySQL database
  - Back up the database
  - Upgrade Mattermost
  - High availabiilty configuration setting recommendations
  - Encryption-at-rest
  - Use sockets for the database
  - Mattermost configuration in the database
    - Create an environment file
    - Modify the Mattermost systemd file
  - Technical notes about searching
    - Perform searches in Chinese, Korean, and Japanese
  - Migrate from Bitnami to a self-hosted Mattermost deployment

- Downgrade Mattermost v6.0 to v5.38

Reference Architecture for Mattermost Server

The following diagrams detail suggested architecture configurations of high availability Mattermost deployments at different scales. Hardware and infrastructure requirements will vary significantly based on usage and policies. See the scaling for enterprise documentation for reference architecture guidance at scale, including hardware and infrastructure requirements.

High availability in Mattermost consists of running redundant Mattermost application servers, redundant database servers, and redundant load balancers so that failure of any one of these components does not interrupt operation of the system. Upon failure of one component, the remaining application servers, database servers, and load balancers must be sized and configured to carry the full load of the system. If this requirement is not met, an outage of one component can result in an overload of the remaining components, causing a complete system outage.

Important

Mattermost does not support high availability deployments spanning multiple datacenters. All nodes in a high availability cluster must reside within the same datacenter to ensure proper functionality and performance.

You can apply most configuration changes and dot release security updates without interrupting service, provided that you update the system components in the correct sequence. Changes to configuration settings that require a server restart, and server version upgrades that involve a change to the database schema, require a short period of downtime. Downtime for a server restart is around 5 seconds. For a database schema update, downtime can be up to 30 seconds.

Designed for scale

Mattermost is designed to be able to handle a large number of concurrent users, and the architecture can be scaled up or down as needed. The architecture is also designed to be flexible, allowing for the addition of new

components or services as needed. The following diagrams show the recommended architecture for Mattermost deployments at 5,000, 10,000, 25,000, and 50,000 users. The diagrams are organized by user count and include a general diagram and AWS and Azure versions of each diagram. See the scaling for enterprise documentation for more information on scaling Mattermost deployments.

- Each generalized diagram represents a full High Availability deployment across all critical components. The proxy, database, file storage, and Elasticsearch layers can be replaced by cloud services.
- Each AWS diagram represents a full High Availability deployment on Amazon Web Services making full use of the available services.
- Each Azure diagram represents a full High Availability deployment on Microsoft Azure making full use of the available services.
- Push proxy can be replaced by the Mattermost hosted push notification service.

AWS

5,000 users

10,000 users

25,000 users

50,000 users

Azure

5,000 users

10,000 users

25,000 users

50,000 users

Oracle

5,000 users

On this page

- Reference Architecture for Mattermost Server
- Reference Architecture for Mattermost Server
  - Designed for scale
    - AWS
      - 5,000 users
      - 10,000 users
      - 25,000 users
      - 50,000 users
    - Azure
      - 5,000 users
      - 10,000 users
      - 25,000 users
      - 50,000 users
    - Oracle
      - 5,000 users
  - Designed for scale

Server deployment planning

Core technology stack

Mattermost's architecture is built on modern, reliable technologies:

- **Backend**: Written in Go, providing high performance and concurrent processing
- **Frontend**: React-based web application and mobile apps
- **Database**: PostgreSQL for primary data storage
- **Search**: Elasticsearch (optional) for advanced search capabilities
- **File Storage**: Local filesystem or cloud storage (S3, MinIO) for media and attachments
- **Caching**: Built-in support for Redis for enhanced performance

Deployment options

Mattermost offers several deployment options to suit your organization's needs:

1. Kubernetes (Recommended)

   Our recommended approach for production deployments offers:

   - Scalability and high availability
   - Automated updates and rollbacks
   - Infrastructure as code
   - Built-in monitoring and logging
   - Easy integration with existing DevOps workflows

2. Linux Server Installation

A direct installation on Linux servers offers:

- Simple, straightforward setup
- Full control over the installation
- For situations where containers aren't preferred

3. Container-Based Deployment

   Docker containers are suitable for smaller deployments only as it offers:

   - Simplified installation and updates
   - Consistent environments
   - Easy dependency management
   - No support for high availability

Prerequisites

Before deploying Mattermost, ensure you have reviewed the software and hardware requirements, and have:

- A supported Linux distribution
- Database server (PostgreSQL 13+)
- Reverse proxy (NGINX recommended)
- SSL/TLS certificates for secure communication
- Adequate storage for files and database
- Network access and firewall configurations
- System requirements met based on expected user load

Plan your deployment

When planning your Mattermost deployment, consider the following when choosing the deployment method that best aligns with your organization's requirements, technical expertise, and infrastructure capabilities:

- Expected user count and growth
- High availability requirements
- Backup and disaster recovery needs
- Integration with existing systems
- Security and compliance requirements
- Monitoring and maintenance strategy

The following server, desktop, and mobile application sections provide detailed instructions for each deployment approach.

Minimum database version policy

To make planning easier and ensure your Mattermost deployment remains fast and secure, we are introducing a policy for updating the minimum supported version of PostgreSQL. The oldest supported PostgreSQL version Mattermost supports will match the oldest version supported by the PostgreSQL community. This ensures you benefit from the latest features and security updates.

This policy change takes effect from Mattermost v10.6, where the minimum PostgreSQL version required will be PostgreSQL 13. This aligns with the PostgreSQL community's support policy, which provides 5 years of support for each major version.

> Note
>
> Mattermost v10.6 is not an Extended Support Release (ESR). Going forward, this database version support policy will only apply to ESR releases.

When a PostgreSQL version reaches its end of life (EOL), Mattermost will require a newer version starting with the next scheduled ESR release. This means the following future PostgreSQL minimum version increases as follows:

Mattermost Version	Release Date	Minimum PostgreSQL Version
v9.11 ESR	2024-8-15	11.x
v10.5 ESR	2025-2-15	11.x
v10.6	2025-3-15	13.x
v10.11 ESR	2025-8-15	13.x
v11.5 ESR *	2026-2-15	14.x (EOL 2026-11-12)

* Forcasted release version and date.

Customers will have 9 months to plan, test, and upgrade their PostgreSQL version before the new requirement takes effect. This policy aims to provide clarity and transparency so you can align database upgrades with the Mattermost release schedule. Contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/). to discuss your options.

On this page

- Server deployment planning
  - Core technology stack
  - Deployment options
  - Prerequisites
  - Plan your deployment
    - Minimum database version policy

(Recommended) Set up an NGINX proxy

A proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. Mattermost recommends using a proxy in front of Mattermost to increase security, performance and the ability to monitor and shape traffic connecting to Mattermost:

- **Security:** A proxy server can manage Secure Socket Layer (TLS/SSL) encryption and set policy on how network traffic will be routed to the Mattermost server.
- **Performance:** In a High Availability configuration, the proxy server balances the network load across multiple Mattermost servers for optimized performance. A hardware proxy with dedicated devices for processing SSL encryption and decryption can also be used to increase performance.
- **Monitoring**: A proxy server can monitor connection traffic and record traffic in standard audit logs that common monitoring tools like Kibana and Splunk can consume and report on. Some of the events that can be captured include file uploads and downloads, which are not tracked by the Mattermost server logging process.

Mattermost supports the [NGINX proxy (https://www.f5.com/go/product/welcome-to-nginx)](https://www.f5.com/go/product/welcome-to-nginx)

Install NGINX server

NGINX is a popular web server and is responsible for hosting some of the largest and highest-traffic sites on the internet. It's more resource-friendly than Apache in most cases, and can be used as a web server or reverse proxy.

In a production setting, we recommend using a proxy server for greater security and performance of Mattermost:

- SSL termination
- HTTP to HTTPS redirect
- Port mapping `:80` to `:8065`
- Standard request logs

Install NGINX on Ubuntu Server

1. Log in to the server that will host the proxy and open a terminal window.

2. Install NGINX.

> *Because NGINX is available in Ubuntu's default repositories, it's possible to install it from these repositories using the* `apt` *packaging system. First, update your local* `apt` *package index for access to the most recent package listings. Then, install* `nginx`*:*
>
> ```
> sudo apt update sudo apt install nginx
> ```
>
> *After accepting the procedure,* `apt` *will install NGINX and any required dependencies to your server.*

3. After installing it, you already have everything you need. You can point your browser to your server IP address. You should see the default NGINX landing page:

>
>
>
> *If you see this page, you've successfully installed NGINX on your web server. This page is included with NGINX to show you that the server is running correctly.*
>
> *Or you can also verify it by running* `curl http://localhost`*.*
>
> *If NGINX is running, you see the following output:*
>
> ```
> <!DOCTYPE html> <html> <head> <title>Welcome to nginx!</title> . . .
> Thank you for using nginx. </body> </html>
> ```

Manage the NGINX process

Now that you have your web server up and running, let's review some basic management commands. These are all run in the command line interface.

To stop your web server, use: `sudo systemctl stop nginx`

To start the web server when it's stopped, use: `sudo systemctl start nginx`

To stop and then start the service again, use: `sudo systemctl restart nginx`

If you're simply making configuration changes, NGINX can often reload without dropping connections. To do this, use: `sudo systemctl reload nginx`

By default, NGINX is configured to start automatically when the server boots. If this isn't what you want, you can disable this behavior using: `sudo systemctl disable nginx`

To re-enable the service to start up at boot, use: `sudo systemctl enable nginx`

What to do next

1. Map a fully qualified domain name (FQDN) such as `mattermost.example.com` on your DNS server/service, to point to the NGINX server.
2. Configure NGINX to proxy connections from the internet to the Mattermost server.

Configure NGINX as a proxy for Mattermost server

NGINX is configured using a file in the `/etc/nginx/sites-available` directory. You need to create the file and then enable it. When creating the file, you need the IP address of your Mattermost server and the fully qualified domain name (FQDN) of your Mattermost website.

1. Log in to the server that hosts NGINX and open a terminal window.
2. Create a configuration file for Mattermost by running the following command:

> `sudo touch /etc/nginx/sites-available/mattermost` *on Ubuntu*

> `sudo touch /etc/nginx/conf.d/mattermost` *on RHEL 8*

3. Open the file `/etc/nginx/sites-available/mattermost` (Ubuntu) or `/etc/nginx/conf.d/mattermost` (RHEL 8) as *root* user in a text editor and replace its contents, if any, with the following lines. Make sure that you use your own values for the Mattermost server IP address and FQDN for *server_name*.

SSL and HTTP/2 are enabled in the provided configuration example.

Note

- If you're going to use Let's Encrypt to manage your SSL certificate, stop at step 3 and see the NGINX HTTP/2 and SSL product documentation for details.
- You'll need valid SSL certificates in order for NGINX to pin the certificates properly. Additionally, your browser must have permissions to accept the certificate as a valid CA-signed certificate.
- Note that the IP address included in the examples in this documentation may not match your network configuration.
- If you're running NGINX on the same machine as Mattermost, and NGINX resolves `localhost` to more than one IP address (IPv4 or IPv6), we recommend using `127.0.0.1` instead of `localhost`.

```
upstream backend { server 10.10.10.2:8065; keepalive 32; } server
{ listen 80 default_server; server_name mattermost.example.com; return
301 https://$server_name$request_uri; } server { listen 443 ssl http2;
listen [::]:443 ssl http2; server_name mattermost.example.com;
ssl_certificate /etc/letsencrypt/live/{domain-name}/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/{domain-name}/privkey.pem;
ssl_session_timeout 1d; # Enable TLS versions (TLSv1.3 is required
upcoming HTTP/3 QUIC). ssl_protocols TLSv1.2 TLSv1.3; # Enable
TLSv1.3's 0-RTT. Use $ssl_early_data when reverse proxying to # prevent
replay attacks. # # @see: https://nginx.org/en/docs/http/
ngx_http_ssl_module.html#ssl_early_data ssl_early_data on; ssl_ciphers
'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA384'; ssl_prefer_server_ciphers on;
ssl_session_cache shared:SSL:50m; # HSTS (ngx_http_headers_module is
required) (15768000 seconds = six months) add_header Strict-Transport-
Security max-age=15768000; # OCSP Stapling --- # fetch OCSP records
from URL in ssl_certificate and cache them ssl_stapling on;
ssl_stapling_verify on; add_header X-Early-Data $tls1_3_early_data;
location ~ /api/v[0-9]+/(users/)?websocket$ { proxy_set_header Upgrade
$http_upgrade; proxy_set_header Connection "upgrade";
client_max_body_size 50M; proxy_set_header Host $host; proxy_set_header
X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; client_body_timeout 60s; send_timeout 300s;
lingering_timeout 5s; proxy_connect_timeout 90s; proxy_send_timeout
300s; proxy_read_timeout 90s; proxy_http_version 1.1; proxy_pass
http://backend; } location / { client_max_body_size 100M;
proxy_set_header Connection ""; proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-
For $proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto
$scheme; proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256
16k; proxy_buffer_size 16k; proxy_read_timeout 600s; proxy_http_version
1.1; proxy_pass http://backend; } } # This block is useful for
debugging TLS v1.3. Please feel free to remove this # and use the
`$ssl_early_data` variable exposed by NGINX directly should you # wish
to do so. map $ssl_early_data $tls1_3_early_data { "~."
$ssl_early_data; default ""; }
```

4. Remove the existing default sites-enabled file by running `sudo rm /etc/nginx/sites-enabled/default` (Ubuntu) or `sudo rm /etc/nginx/conf.d/default` (RHEL 8)

5. Enable the mattermost configuration by running `sudo ln -s /etc/nginx/sites-available/mattermost /etc/nginx/sites-enabled/mattermost` (Ubuntu) or `sudo ln -s /etc/nginx/conf.d/mattermost /etc/nginx/conf.d/default.conf` (RHEL 8)
6. Restart NGINX by running `sudo systemctl restart nginx`.
7. Verify that you can see Mattermost through the proxy by running `curl https://localhost`. If everything is working, you will see the HTML for the Mattermost signup page.
8. Restrict access to port 8065.

> *By default, the Mattermost server accepts connections on port 8065 from every machine on the network. Use your firewall to deny connections on port 8065 to all machines except the machine that hosts NGINX and the machine that you use to administer the Mattermost server. If you're installing on Amazon Web Services, you can use Security Groups to restrict access.*

Now that NGINX is installed and running, you can configure it to use SSL, which allows you to use HTTPS connections and the HTTP/2 protocol.

Configure NGINX with SSL and HTTP/2

NGINX is configured using a file in the `/etc/nginx/sites-available` directory. You need to create the file and then enable it. When creating the file, you need the IP address of your Mattermost server and the fully qualified domain name (FQDN) of your Mattermost website.

Using SSL gives greater security by ensuring that communications between Mattermost clients and the Mattermost server are encrypted. It also allows you to configure NGINX to use the HTTP/2 protocol.

Although you can configure HTTP/2 without SSL, both Firefox and Chrome browsers support HTTP/2 on secure connections only.

You can use any certificate that you want, but these instructions show you how to download and install certificates from [Let's Encrypt (https://letsencrypt.org/)](https://letsencrypt.org/), a free certificate authority.

> Note
>
> If Let's Encrypt is enabled, forward port 80 through a firewall, with Forward80To443 `config.json` setting set to `true` to complete the Let's Encrypt certification. See the Let's Encrypt/Certbot documentation for additional assistance.

1. Log in to the server that hosts NGINX and open a terminal window.
2. Open the your Mattermost `nginx.conf` file as *root* in a text editor, then update the `{ip}` address in the `upstream backend` to point towards Mattermost (such as `127.0.0.1:8065`), and update the `server_name` to be your domain for Mattermost.

> *Note*
>
> - *On Ubuntu this file is located at `/etc/nginx/sites-available/`. If you don't have this file, run `sudo touch /etc/nginx/sites-available/mattermost`.*
> - *On CentOS/RHEL this file is located at `/etc/nginx/conf.d/`. If you don't have this file, run `sudo touch /etc/nginx/conf.d/mattermost`.*
> - *The IP address included in the examples in this documentation may not match your network configuration.*
> - *If you're running NGINX on the same machine as Mattermost, and NGINX resolves `localhost` to more than one IP address (IPv4 or IPv6), we recommend using `127.0.0.1` instead of `localhost`.*

```
upstream backend { server {ip}:8065; keepalive 32; } server { listen 80
default_server; server_name mattermost.example.com; location ~ /api/
v[0-9]+/(users/)?websocket$ { proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade"; client_max_body_size 50M;
proxy_set_header Host $http_host; proxy_set_header X-Real-IP
$remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; client_body_timeout 60s; send_timeout 300s;
lingering_timeout 5s; proxy_connect_timeout 90s; proxy_send_timeout 300s;
proxy_read_timeout 90s; proxy_pass http://backend; } location /
{ client_max_body_size 50M; proxy_set_header Connection "";
proxy_set_header Host $http_host; proxy_set_header X-Real-IP
$remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; proxy_read_timeout 600s; proxy_http_version 1.1;
proxy_pass http://backend; } }
```

3. Remove the existing default sites-enabled file by running `sudo rm /etc/nginx/sites-enabled/default` (Ubuntu) or `sudo rm /etc/nginx/conf.d/default` (RHEL 8).

4. Enable the Mattermost configuration by running `sudo ln -s /etc/nginx/sites-available/mattermost /etc/nginx/sites-enabled/mattermost` (Ubuntu) or `sudo ln -s /etc/nginx/conf.d/mattermost /etc/nginx/conf.d/default.conf` (RHEL 8).

5. Run `sudo nginx -t` to ensure your configuration is done properly. If you get an error, look into the NGINX config and make the needed changes to the file under `/etc/nginx/sites-available/mattermost`.

6. Restart NGINX by running `sudo systemctl start nginx`.

7. Verify that you can see Mattermost through the proxy by running `curl http://localhost`.

> *If everything is working, you will see the HTML for the Mattermost signup page. You will see invalid certificate when accessing through the IP or*

> *localhost. Use the full FQDN domain to verify if the SSL certificate has pinned properly and is valid.*

8. Install and update Snap by running `sudo snap install core; sudo snap refresh core`.
9. Install the Certbot package by running `sudo snap install --classic certbot`.
10. Add a symbolic link to ensure Certbot can run by running `sudo ln -s /snap/bin/certbot /usr/bin/certbot`.
11. Run the Let's Encrypt installer dry-run to ensure your DNS is configured properly by running `sudo certbot certonly --dry-run`.

> *This will prompt you to enter your email, accept the TOS, share your email, and select the domain you're activating certbot for. This will validate that your DNS points to this server properly and you are able to successfully generate a certificate. If this finishes successfully, proceed to step 12.*

12. Run the Let's Encrypt installer by running `sudo certbot`. This will run certbot and will automatically edit your NGINX config file for the site(s) selected.
13. Ensure your SSL is configured properly by running `curl https://{your domain here}`
14. Finally, we suggest editing your config file again to increase your SSL security settings above the default Let's Encrypt. This is the same file from Step 2 above. Edit it to look like the below:

```
upstream backend { server {ip}:8065; keepalive 32; } proxy_cache_path /
var/cache/nginx levels=1:2 keys_zone=mattermost_cache:10m max_size=3g
inactive=120m use_temp_path=off; server { server_name
mattermost.example.com; location ~ /api/v[0-9]+/(users/)?websocket$
{ proxy_set_header Upgrade $http_upgrade; proxy_set_header Connection
"upgrade"; client_max_body_size 50M; proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; client_body_timeout 60s; send_timeout 300s;
lingering_timeout 5s; proxy_connect_timeout 90s; proxy_send_timeout 300s;
proxy_read_timeout 90s; proxy_http_version 1.1; proxy_pass http://
backend; } location / { client_max_body_size 50M; proxy_set_header
Connection ""; proxy_set_header Host $http_host; proxy_set_header X-Real-
IP $remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; proxy_read_timeout 600s; proxy_http_version 1.1;
proxy_pass http://backend; } listen 443 ssl http2; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/mattermost.example.com/
fullchain.pem; # managed by Certbot ssl_certificate_key /etc/letsencrypt/
live/mattermost.example.com/privkey.pem; # managed by Certbot # include /
etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
ssl_session_timeout 1d; # Enable TLS versions (TLSv1.3 is required
upcoming HTTP/3 QUIC). ssl_protocols TLSv1.2 TLSv1.3; # Enable TLSv1.3's
0-RTT. Use $ssl_early_data when reverse proxying to # prevent replay
attacks. # # @see: https://nginx.org/en/docs/http/
ngx_http_ssl_module.html#ssl_early_data ssl_early_data on; ssl_ciphers
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES256-SHA;
ssl_prefer_server_ciphers on; ssl_session_cache shared:SSL:50m; # HSTS
(ngx_http_headers_module is required) (15768000 seconds = six months)
add_header Strict-Transport-Security max-age=15768000; # OCSP Stapling
--- # fetch OCSP records from URL in ssl_certificate and cache them
ssl_stapling on; ssl_stapling_verify on; } server { if ($host =
mattermost.example.com) { return 301 https://$host$request_uri; } #
managed by Certbot listen 80 default_server; server_name
mattermost.example.com; return 404; # managed by Certbot }
```

15. Check that your SSL certificate is set up correctly.

> - *Test the SSL certificate by visiting a site such as https://www.ssllabs.com/ssltest/index.html.*

> - *If there's an error about the missing chain or certificate path, there is likely an intermediate certificate missing that needs to be included.*

High-performance scaling configuration

For high-scale deployments with multiple Mattermost servers and heavy traffic loads, additional NGINX optimizations are recommended. This configuration is based on performance testing with large-scale Mattermost deployments.

> Note
>
> These settings are designed for high-performance environments. For standard deployments, the basic configuration above should be sufficient.

NGINX main configuration optimizations

Update your main NGINX configuration file (`/etc/nginx/nginx.conf`) with the following performance optimizations:

```
user www-data; worker_processes auto; worker_rlimit_nofile 100000; pid /run/
nginx.pid; include /etc/nginx/modules-enabled/*.conf; events
{ worker_connections 20000; use epoll; } http { map $status $loggable
{ ~^[23] 0; default 1; } sendfile on; tcp_nopush on; tcp_nodelay off;
keepalive_timeout 75s; keepalive_requests 16384; types_hash_max_size 2048;
include /etc/nginx/mime.types; default_type application/octet-stream;
ssl_prefer_server_ciphers on; access_log /var/log/nginx/access.log combined
if=$loggable; error_log /var/log/nginx/error.log; gzip on; include /etc/
nginx/sites-enabled/*; }
```

Key optimizations in this configuration:

- **worker_processes auto**: Automatically sets worker processes based on CPU cores
- **worker_rlimit_nofile 100000**: Increases file descriptor limit for high-concurrency
- **worker_connections 20000**: Allows each worker to handle more concurrent connections

- **use epoll**: Enables efficient connection handling on Linux
- **keepalive_timeout 75s** and **keepalive_requests 16384**: Optimizes connection reuse
- **Conditional logging**: Reduces log volume by only logging errors and non-2xx/3xx responses

Multi-node backend configuration

For deployments with multiple Mattermost application servers, configure your site file (`/etc/nginx/sites-available/mattermost`) with load balancing:

```
upstream backend { server 172.27.205.186:8065 max_fails=0; server
172.27.213.167:8065 max_fails=0; # Add additional Mattermost servers as
needed keepalive 256; } proxy_cache_path /var/cache/nginx levels=1:2
keys_zone=mattermost_cache:10m max_size=3g inactive=60m use_temp_path=off;
server { listen 80 reuseport; server_name _; location ~ /api/v[0-9]+/
(users/)?websocket$ { proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade"; include /etc/nginx/snippets/
proxy.conf; } location ~ /api/v[0-9]+/users/[a-z0-9]+/image$
{ proxy_set_header Connection ""; include /etc/nginx/snippets/proxy.conf;
include /etc/nginx/snippets/cache.conf; proxy_ignore_headers Cache-Control
Expires; proxy_cache_valid 200 24h; } location / { proxy_set_header
Connection ""; include /etc/nginx/snippets/proxy.conf; include /etc/nginx/
snippets/cache.conf; } }
```

NGINX proxy configuration snippets

Create optimized proxy configuration snippets for reuse across different locations.

Create `/etc/nginx/snippets/proxy.conf` :

```
client_max_body_size 50M; proxy_set_header Host $http_host; proxy_set_header
X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Frame-Options SAMEORIGIN; proxy_buffers 256 16k;
proxy_buffer_size 16k; client_body_timeout 60s; send_timeout 300s;
lingering_timeout 5s; proxy_connect_timeout 30s; proxy_send_timeout 90s;
proxy_read_timeout 90s; proxy_http_version 1.1; proxy_pass http://backend;
```

Create `/etc/nginx/snippets/cache.conf` :

```
proxy_cache mattermost_cache; proxy_cache_revalidate on;
proxy_cache_min_uses 2; proxy_cache_use_stale timeout; proxy_cache_lock on;
```

Key performance optimizations

The high-performance configuration includes several critical optimizations:

Load Balancing:

- **max_fails=0**: Prevents servers from being marked as unavailable
- **keepalive 256**: Maintains persistent connections to backend servers
- **reuseport**: Allows multiple worker processes to bind to the same port

Caching Strategy:

- **User images cached for 24 hours**: Reduces load on application servers for static content
- **Cache revalidation**: Ensures fresh content while maintaining performance
- **Cache locking**: Prevents cache stampede scenarios

Buffer and Timeout Optimization:

- **proxy_buffers 256 16k**: Handles high-throughput data transfer efficiently
- **Optimized timeouts**: Balances responsiveness with resource usage
- **HTTP/1.1 keepalive**: Reduces connection overhead

Resource Limits:

- **50M client_max_body_size**: Accommodates large file uploads
- **Increased file descriptor limits**: Supports high-concurrency scenarios

Implementation notes

1. **Replace IP addresses**: Update the backend server IPs (172.27.205.186, 172.27.213.167) with your actual Mattermost server addresses.

2. **Create cache directory**: Ensure the cache directory exists and has proper permissions:

```
sudo mkdir -p /var/cache/nginx sudo chown -R www-data:www-data /var/
cache/nginx
```

3. **Test configuration**: Always test your configuration before applying:

```
sudo nginx -t
```

4. **Monitor performance**: Use NGINX access logs and monitoring tools to verify the performance improvements.

5. **Scale incrementally**: Apply these optimizations gradually and monitor their impact on your specific deployment.

NGINX configuration FAQ

Why am I seeing the error "Too many redirects?"

You may see this error if you're setting Mattermost in a sub-path. To resolve this error, add the following block to the HEAD request:

```
location ~* ^/sub-path { client_max_body_size 250M; proxy_set_header
Connection ""; if ($request_method = HEAD) { return 200; } }
```

Why are Websocket connections returning a 403 error?

This is likely due to a failing cross-origin check. A check is applied for WebSocket code to see if the `Origin` header is the same as the host header. If it's not, a 403 error is returned. Open the file `/etc/nginx/sites-available/mattermost` as *root* in a text editor and make sure that the host header being set in the proxy is dynamic:

```
location ~ /api/v[0-9]+/(users/)?websocket$ { proxy_pass http://backend;
(...)  proxy_set_header Host $host;  proxy_set_header X-Forwarded-For
$remote_addr; }
```

Then in `config.json` set the `AllowCorsFrom` setting to match the domain being used by clients. You may need to add variations of the host name that clients may send. Your NGINX log will be helpful in diagnosing the problem.

```
"EnableUserAccessTokens": false,
"AllowCorsFrom": "domain.com domain.com:443 im.domain.com",
"SessionLengthWebInDays": 30,
```

How do I setup an NGINX proxy with the Mattermost Docker installation?

1. Find the name of the Mattermost network and connect it to the NGINX proxy.

```
docker network ls # Grep the name of your Mattermost network like
"mymattermost_default". docker network connect mymattermost_default nginx-
proxy
```

2. Restart the Mattermost Docker containers.

```
docker-compose stop app docker-compose start app
```

> Tip
>
> You don't need to run the 'web' container, since NGINX proxy accepts incoming requests.

3. Update your `docker-compose.yml` file to include a new environment variable `VIRTUAL_HOST` and an `expose` directive.

```
environment: # set same as db credentials and dbname - MM_USERNAME=mmuser -
MM_PASSWORD=mmuser-password - MM_DBNAME=mattermost -
VIRTUAL_HOST=mymattermost.tld expose: - "80" - "443"
```

Why does NGINX fail when installing GitLab CE with Mattermost on Azure?

You may need to update the Callback URLs for the Application entry of Mattermost inside your GitLab instance.

1. Log in to your GitLab instance as the admin.
2. Go to **Admin > Applications**.
3. Select **Edit** on GitLab-Mattermost.
4. Update the callback URLs to your new domain/URL.
5. Save the changes.
6. Update the external URL for GitLab and Mattermost in the `/etc/gitlab/gitlab.rb` configuration file.

Why does Certbot fail the http-01 challenge?

```
Requesting a certificate for yourdomain.com Performing the following
challenges: http-01 challenge for yourdomain.com Waiting for verification...
Challenge failed for domain yourdomain.com http-01 challenge for
yourdomain.com Cleaning up challenges Some challenges have failed.
```

If you see the above errors this is typically because Certbot wasn't able to access port 80. This can be due to a firewall or other DNS configuration. Make sure that your A/AAAA records are pointing to this server and your `server_name` within the NGINX config doesn't have a redirect.

> Note
>
> If you're using Cloudflare you'll need to disable `force traffic to https`.

Certbot rate limiting

If you're running certbot as stand-alone you'll see this error:

```
Error: Could not issue a Let's Encrypt SSL/TLS certificate for example.com.
One of the Let's Encrypt rate limits has been exceeded for example.com. See
the related Knowledge Base article for details. Details Invalid response
from https://acme-v02.api.letsencrypt.org/acme/new-order. Details: Type:
urn:ietf:params:acme:error:rateLimited Status: 429 Detail: Error creating
new order :: too many failed authorizations recently: see https://
letsencrypt.org/docs/rate-limits/
```

If you're running Let's Encrypt within Mattermost you'll see this error:

```
{"level":"error","ts":1609092001.752515,"caller":"http/
server.go:3088","msg":"http: TLS handshake error from ip:port: 429
urn:ietf:params:acme:error:rateLimited: Error creating new order :: too many
failed authorizations recently: see https://letsencrypt.org/docs/rate-
limits/","source":"httpserver"}
```

This means that you've attempted to generate a cert too many times. You can find more information here (https://letsencrypt.org/docs/rate-limits).

On this page

- (Recommended) Set up an NGINX proxy
  - Install NGINX server
    - Install NGINX on Ubuntu Server
    - Manage the NGINX process
    - What to do next
  - Configure NGINX as a proxy for Mattermost server
  - Configure NGINX with SSL and HTTP/2
  - High-performance scaling configuration
    - NGINX main configuration optimizations
    - Multi-node backend configuration
    - NGINX proxy configuration snippets
    - Key performance optimizations
    - Implementation notes
  - NGINX configuration FAQ
    - Why am I seeing the error "Too many redirects?"
    - Why are Websocket connections returning a 403 error?

- How do I setup an NGINX proxy with the Mattermost Docker installation?
- Why does NGINX fail when installing GitLab CE with Mattermost on Azure?
- Why does Certbot fail the http-01 challenge?
  - Certbot rate limiting

(Optional) Set up TLS

You have two options if you want users to connect with HTTPS:

Install a proxy such as NGINX and then set up TLS on the proxy. This is our recommended option if you have a large number of users (more than 200), or if you want to use a reverse proxy for other reasons, such as load balancing or caching. A proxy server delivers better performance and provides standard HTTP request logs.

Alternatively, if you have fewer than 200 users, you can set up TLS on Mattermost server. This is the easiest option when you don't need to use a reverse proxy.

> - *You can use [Let's Encrypt (https://letsencrypt.org/)](https://letsencrypt.org/) to automatically install and set up the certificate.*
> - *You can also specify your own certificate.*
> - *You can use a self-signed certificate, but this is not recommended for production environments.*

Configure TLS on the Mattermost server

1. In **System Console > Environment > Web Server**:

> 1. *Change the **Listen Address** setting to `:443`.*
> 2. *Change the **Connection Security** setting to `TLS`.*
> 3. *Change the **Forward port 80 to 443** setting to `true`.*

2. Activate the `CAP_NET_BIND_SERVICE` capability to allow Mattermost to bind to low ports:

> ```
> sudo setcap cap_net_bind_service=+ep /opt/mattermost/bin/mattermost
> ```

3. Install the security certificate. Use Let's Encrypt to automatically install and setup the certificate, or specify your own certificate.

Use a Let's Encrypt certificate

The certificate is retrieved the first time that a client tries to connect to the Mattermost server. Certificates are retrieved for any hostname a client tries to reach the server at.

1. Change the **Use Let's Encrypt** setting to `true`.
2. Restart the Mattermost server for these changes to take effect.

> Note
>
> - If Let's Encrypt is enabled, forward port 80 through a firewall, with Forward80To443 `config.json` setting set to `true` to complete the Let's Encrypt certification.
> - Your Mattermost server must be accessible from the Let's Encrypt CA in order to verify your domain name and issue the certificate. Be sure to open your firewall and configure any reverse proxies to forward traffic to ports 80 and 443. More information can be found at Let's Encrypt (https://letsencrypt.org/how-it-works/).

Use your own certificate

1. Change the **Use Let's Encrypt** setting to `false`.
2. Change the **TLS Certificate File** setting to the location of the certificate file.
3. Change the **TLS Key File** setting to the location of the private key file.
4. Restart the Mattermost server for these changes to take effect.

> Note
>
> Password-protected certificates aren't supported.

Use TLS on NGINX (as a proxy)

> Warning
>
> Do not set up TLS on Mattermost before doing so for NGINX. It breaks the connection as the TLS prevents it from successfully communicating with the Mattermost server.

- NGINX will act as a forward proxy to encrypt the traffic between the client and Mattermost server. After installing the SSL certificate, the incoming traffic will be handled via NGINX on port 443 exposed to the internet, proxy to the Mattermost server running on port 80.
- (Optional) Upstream encryption between NGINX to Mattermost server is allowed.
- Follow [NGINX's guide on setting up SSL Termination for TCP Upstream Servers (https://docs.nginx.com/nginx/admin-guide/security-controls/terminating-ssl-tcp/)](https://docs.nginx.com/nginx/admin-guide/security-controls/terminating-ssl-tcp/).

More helpful resources:

- [NGINX's SSL blog (https://www.f5.com/company/blog/nginx/nginx-ssl/)](https://www.f5.com/company/blog/nginx/nginx-ssl/)
- [NGINX's SSL guide (https://docs.nginx.com/nginx/admin-guide/security-controls/terminating-ssl-http/)](https://docs.nginx.com/nginx/admin-guide/security-controls/terminating-ssl-http/)

On this page

- (Optional) Set up TLS
  - Configure TLS on the Mattermost server
  - Use a Let's Encrypt certificate
  - Use your own certificate
  - Use TLS on NGINX (as a proxy)

PostgreSQL installation troubleshooting

From Mattermost v8.0, PostgreSQL is our database of choice for Mattermost to enhance the platform's performance and capabilities.

PostgreSQL v15 introduces changes that may affect compatibility with previous releases. If you're deploying a fresh installation of PostgreSQL v15, run this command: `GRANT CREATE ON SCHEMA public TO PUBLIC` to ensure that you can use Mattermost.

PostgreSQL full-text search fails to use indexes with non-English `default_text_search_config`

Mattermost uses `default_text_search_config` for full-text search in PostgreSQL databases, as opposed to a hardcoded text search config. However, indexes are still created with a hardcoded text search config (english) and as a result, full-text search may never use the indexes.

Some of the tables in Mattermost, like `Posts` or `Users`, contain GIN indexes to improve the database full-text search feature in PostgreSQL.

These indexes need to be built against a specific language, and when they're created they're hard-coded to English. Full-text search queries are always performed using the `default_text_search_config` database setting. In order for the full-text search feature to leverage the indexes, the language specified in the query needs to match the language specified in the index.

If the `default_text_search_config` is not set to `english`, the GIN indexes will not be used. Database administrators can work around this by dropping the specific GIN index they're interested in and rebuilding it with the value of `default_text_search_config`.

For example, if the default language of your server is Spanish:

`sql # Create the new index with a new name before dropping the old one CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_posts_message_txt_spanish ON posts USING

`gin(to_tsvector('spanish', message)); # Check that the new index does work. If it does, drop the old one and rename the new one: DROP INDEX CONCURRENTLY IF EXISTS idx_posts_message_txt; ALTER INDEX idx_posts_message_txt_spanish RENAME TO idx_posts_message_txt; ```

Change the PostgreSQL username and password

We recommend changing the PostgreSQL username and password in the `.env` file.

On this page

- PostgreSQL installation troubleshooting
  - PostgreSQL full-text search fails to use indexes with non-English default_text_search_config
  - Change the PostgreSQL username and password

MySQL installation troubleshooting

Before you can run the Mattermost server, you must first install and configure a database. You can start Mattermost by navigating to the `/opt/mattermost` directory and entering the command `sudo -u mattermost bin/mattermost`. If the Mattermost server can't connect to the database, it will fail to start. This section deals with MySQL database issues that you may encounter when you start up Mattermost for the first time.

> Note
>
> - Additional database tuning guidance is available for specific Mattermost releases. See the important upgrade notes documentation for more details.
> - See the database configuration settings documentation for details on configuration options specific to MySQL databases.

How you install MySQL varies depending upon which Linux distribution you use. However, once MySQL is installed, the configuration instructions are the same. For all distributions you must create a `mattermost` database and a `mattermost` database user. Failure to create these database objects or improperly referencing them from the Mattermost configuration file, `/opt/mattermost/config/config.json`, causes Mattermost to fail. The troubleshooting tips given here deal with these specific issues.

Before proceeding, confirm that your MySQL server is running. You can do this by issuing the command `mysqladmin -u root -p status`. When prompted, enter your password. If MySQL is running you should see output like the following:

```
Uptime: 877134 Threads: 1 Questions: 9902 Slow queries: 0 Opens: 522
Flush tables: 1 Open tables: 371 Queries per second avg: 0.011
```

If MySQL is not running, review the instructions for installation on your distribution.

> Warning
>
> Some of the commands used in this section alter the database. **Use these commands only if your Mattermost installation has failed.** Do not directly manipulate the MySQL database for a working Mattermost installation.

The `mattermost` database

The database created during installation is named `mattermost`. If you fail to create this database or you misname it, you will see an error such as the following when you attempt to start the Mattermost server:

```
[2017/09/20 17:11:37 EDT] [INFO] Pinging SQL master database [2017/09/20
17:11:37 EDT] [ERROR] Failed to ping DB retrying in 10 seconds err-Error
1044: Access denied for user 'mmuser'@'%' to database 'mattermost'
```

Note that MySQL is specifically denying access to the `mattermost` database. This may mean that you have failed to create a database named `mattermost` or you may have incorrectly referenced this database from the `/opt/mattermost/config/config.json` file.

Checking that the Database Exists

To confirm that the `mattermost` database exists, open MySQL as *root* by executing `mysql -u root -p`. When prompted, enter your password and then issue the command `show databases;`. This command displays all the databases. You should see something similar to the following:

```
+--------------------+ | Database | +--------------------+ |
information_schema | | mattermost | | mysql | | performance_schema | |
sys | +--------------------+ 5 rows in set (0.03 sec)
```

No mattermost Database

If the `mattermost` database doesn't exist, create a database named `mattermost` by opening MySQL as root and issuing the command: `create database mattermost;`.

If you accidentally created a database with the wrong name, you can remove it by issuing the command: `drop database misnamed;`.

After creating the database, attempt to restart the Mattermost server by navigating to the `/opt/mattermost` directory and entering the command `sudo -u mattermost bin/mattermost`.

The mattermost Database Exists

If the `mattermost` database does exist, confirm that you have defined the database driver correctly in the `/opt/mattermost/config/config.json` file. Open this file in a text editor, and review the value of `"DataSource"`. It should be:

> ```
> "mmuser:*mmuser-password*@tcp(*host-name-or-IP*:3306)/mattermost?charset=utf8mb4,utf8&writeTimeout=30s"
> ```

You should also confirm that `DriverName` element (found immediately above the `DataSource` element) is set to `mysql`.

If you correct an error, restart the Mattermost server by navigating to the `/opt/mattermost` directory and entering the command `sudo -u mattermost bin/mattermost`.

The Database User

During the installation you should create a MySQL database user from the *mysql* prompt by issuing the command

samp:

> *create user 'mmuser'@'%' identified by '{mmuser-password}';*.

The `mmuser-password` value is a placeholder for the password you chose. You may also have specified an IP address rather than the wild card `%`.

> Note
>
> A MySQL user is fully defined by their username and the host that they access MySQL from. These elements are separated by the `@` sign. The `%` character is a wild card indicating that the user can access MySQL from any IP address. If the user you created accesses MySQL from a specific IP address such as `10.10.10.2`, please adjust your actions accordingly.

If the user and host combination that you created does not exist, you will see an error such as:

```
[2017/09/20 17:06:18 EDT] [INFO] Pinging SQL master database [2017/09/20
17:06:18 EDT] [ERROR] Failed to ping DB retrying in 10 seconds err-Error
1045: Access denied for user 'mmuser'@'localhost' (using password: YES)
```

Checking that mmuser Exists

To check that this user exists, log in to MySQL as *root*: `mysql -u root -p`.

When prompted, enter the root password that you created when installing MySQL. From the `mysql` prompt enter the command `select User, Host from mysql.user;`. You should see something like the following:

```
+------------------+-----------+ | User | Host | +------------------
+-----------+ | mmuser | % | | debian-sys-maint | localhost | |
mysql.session | localhost | | mysql.sys | localhost | | root | localhost
| +------------------+-----------+ 5 rows in set (0.00 sec)
```

User Doesn't Exist

If `'mmuser'@'%'` does not exist, create this user by logging into MySQL as *root* and issuing the command:

samp:

> *create user 'mmuser'@'%' identified by '{mmuser-password}';*.

After creating a user, ensure that this user has rights to the `mattermost` database.

User Exists

If the user `mmuser` exists, the DataSource element of the `/opt/mattermost/config/config.json` file may be incorrect. Open this file and search for `DataSource`. Its value should be:

> ```
> "mmuser:*mmuser-password*@tcp(*host-name-or-IP*:3306)/mattermost?charset=utf8mb4,utf8&writeTimeout=30s"
> ```

If you correct an error, restart the Mattermost server by navigating to the `/opt/mattermost` directory and issuing the command: `sudo -u mattermost bin/mattermost`.

The user password

Mattermost will fail if you use an incorrect password for `mmuser`. An incorrect password displays an error message such as the following:

> ```
> [2017/09/20 17:09:10 EDT] [INFO] Pinging SQL master database [2017/09/20 17:09:10 EDT] [ERROR] Failed to ping DB retrying in 10 seconds err-Error 1045: Access denied for user 'mmuser'@'localhost' (using password: YES)
> ```

The password in ``config.json``

The DataSource element of the `/opt/mattermost/config/config.json` file references the `mmuser` password. Open this file and search for `DataSource`. It's value should be:

> ```
> "mmuser:*mmuser-password*@tcp(*host-name-or-IP*:3306)/mattermost?charset=utf8mb4,utf8&writeTimeout=30s"
> ```

Check that the password is correct. If you correct an error, restart the Mattermost server by navigating to `/opt/mattermost` and issuing the command: `sudo -u mattermost bin/mattermost`.

Unsure of Password

If you are not sure that the `mmuser` password is correct, attempt to log in to MySQL as `mmuser` by issuing the command `mysql -u mmuser -p`. You will be

prompted for your password. If your login fails, you are not using the correct password.

With a new database installation, the easiest solution for an unknown password is to remove the existing `mmuser` and then recreating that user. You can do this by logging in to MySQL as `root` and issuing the following commands:

- `drop user mmuser;`
- `flush privileges;`
- samp:

  create user 'mmuser'@'%' identified by '{mmuser-password}';

If you recreate `mmuser`, ensure that this user has rights to the `mattermost` database by following the instructions in the granting privileges to mmuser section below.

Insufficient user privileges

If the database exists and the username and password are correct, the `mmuser` may not have sufficient rights to access the `mattermost` database. If this is the case, you may see an error message such as:

```
[2017/09/20 17:20:53 EDT] [INFO] Pinging SQL master database [2017/09/20
17:20:53 EDT] [ERROR] Failed to ping DB retrying in 10 seconds err-Error
1044: Access denied for user 'mmuser'@'%' to database 'mattermost
```

> Note
>
> Examine the error message closely. The user name displayed in the error message is the user identified in the `DataSource` element of the `/opt/mattermost/config/config.json` file. For example, if the error message reads `Access denied for user 'muser'@'%' ...`, you will know that you have misidentified the user as `muser` in the `config.json` file.

You can check if the user `mmuser` has access to the `mattermost` database by logging in to MySQL as `mmuser` and issuing the command: `show databases;`. If this user does not have rights to view the `mattermost` database, you will not see it in the output.

```
+--------------------+ | Database | +--------------------+ |
information_schema | +--------------------+ 1 rows in set (0.00 sec)
```

Granting privileges to mmuser

If the `mattermost` database exists and `mmuser` cannot view it, exit from MySQL and then log in again as root. Issue the command `grant all privileges on mattermost.* to 'mmuser'@'%';` to grant all rights on `mattermost` to `mmuser`.

Restart the Mattermost server by navigating to the `/opt/mattermost` directory and entering the command `sudo -u mattermost bin/mattermost`.

Server is set to SYSTEM timezone

Mattermost customers using v7.7 or earlier may see an errors occur on servers using MySQL in cases where the server is set to SYSTEM timezone and doesn't support named timezones. These errors can be fixed by populating the `timezone` tables on the server. Refer to the following docs for more information on loading the timezone table:

- Linux (https://dev.mysql.com/doc/refman/5.7/en/mysql-tzinfo-to-sql.html)
- Windows (https://dev.mysql.com/downloads/timezones.html)

> Note
>
> This issue has been addressed from Mattermost v7.8.

Maximum allowed packet

The Go MySQL driver has changed the `maxAllowedPacket` size from 4MiB to 64MiB. This is to make it consistent with the change in the server-side default value from MySQL 5.7 to MySQL 8.0.

If your `max_allowed_packet` setting is not 64MiB, update the MySQL configuration DSN with an additional parameter of `maxAllowedPacket` to match with the server-side value. Alternatively, set a value of `0` to fetch the server-side value automatically on every new connection which has a performance overhead.

On this page

- MySQL installation troubleshooting
  - The mattermost database
  - The Database User
  - The user password
  - Insufficient user privileges
  - Server is set to SYSTEM timezone
  - Maximum allowed packet

General deployment troubleshooting

This document summarizes common deployment troubleshooting issues and resolutions. Some of these suggestions can be done directly, and others may need consultation from your network administrator.

Start Mattermost at system boot

To have the Mattermost Server start at system boot, the systemd unit file needs to be enabled. Run the following command:

```
sudo systemctl enable mattermost.service
```

If your database is on the same system as your Mattermost Server, we recommend editing the default `/lib/systemd/system/mattermost.service` systemd unit file to add `After=postgresql.service` and `BindsTo=postgresql.service` to the `[Unit]` section.

Run Mattermost without a proxy

Mattermost binds to 443 instead of 8065. The Mattermost binary requires the correct permissions to do that binding. You must activate the `CAP_NET_BIND_SERVICE` capability to allow the new Mattermost binary to bind to ports lower than 1024 by running the following command:

```
sudo setcap cap_net_bind_service=+ep ./mattermost/bin/mattermost
```

> Note
>
> We highly recommend using a proxy in front of Mattermost server for up to 200 concurrent users. If you have fewer than 200 concurrent users, you can set up TLS. If you're exceeding 200 concurrent users, you'll need a proxy, such as NGINX, in front of Mattermost to manage the traffic.

Review Mattermost logs

You can access logs for Mattermost and use them for troubleshooting. These steps assume that you have appropriate system admin permissions to do so.

Mattermost Server logs

- Ensure that log files are being created: Navigate to **System Console > Environment > Logging**, confirm that **Output logs to file** is set to **true**.
- You can obtain the path for the log files in **System Console > Environment > Logging > File Log Directory**.

The resulting server log file is called `mattermost.log` and can be opened with a standard text editor or shared directly.

> Note
>
> For a more complete log open **System Console > Environment > Logging**, set **File Log Level** to **DEBUG**, then replicate the issue to log it again. Make sure to revert the file log level to **INFO** after troubleshooting to save disk space.

If filesystem access is not possible, navigate to **System Console > Reporting > Server Logs** to locate the current system logs which can be copied to a file.

You can find more on logging settings here.

Mattermost Desktop App logs

Access desktop app logs by going to **Help > Show logs** from the menu bar.

Alternatively, you can access desktop app log files in the following directory:

- **Windows:** `%userprofile%\AppData\Roaming\Mattermost\logs`
- **Linux:** `~/.local/share/Mattermost/logs` OR `~/.config/Mattermost/logs`
- **MacOS:** `~/Library/Logs/Mattermost` (DMG installation) OR `~Library/Containers/Mattermost.Desktop/Data/Library/Logs/Mattermost` (Appstore installation only)

Mattermost web logs

The browser-based app does not produce additional log files. If the app has to be debugged, use the development tools integrated in your browser for action history.

Mattermost Push Notification Service logs

Logging for the Mattermost Push Notification Service is handled via system log with logger and is appended to `/var/log/syslog`.

Review Mattermost environment

Put together a timeline to eliminate events prior to the error/problem occurring. For example, if you recently reconfigured your firewall and are now having connection issues it might be worth reviewing the settings or rolling back to see whether that resolves the problem.

- If the problem occurred subsequent to some period of normal operation, did anything change in the environment?

  > - *Was the client, host, or server upgraded?*
  > - *Was an operating system update applied?*
  > - *Did the network environment change? For example, was a server moved or a domain migrated?*

> - *Did the system (client or server) recently fail or abnormally terminate?*

- How many users are impacted?

> - *Is this problem affecting one, some, or all users?*
> - *Is the problem occurring only for a user who was recently added to the environment, such as a new employee?*
> - *Do differences exist between the users who are affected and the users who are not affected?*

You can also search the error messages online. Existing solutions from our forum (https://forum.mattermost.com/t/how-to-use-the-troubleshooting-forum/150) can often be found and applied.

Connect to another server

1. Create an account at https://community.mattermost.com.
2. Erase your mobile application and reinstall it.
3. In your mobile app, enter the server URL https://community.mattermost.com and then your login credentials to test whether the connection is working.

Connect with another device

- If you have another mobile device available, try connecting with that to see if your issue still reproduces.
- If you don't have another device available, check with other teammates to see if they are having the same issue.

Opening a support ticket for self-hosted deployments

If you have a paid subscription to a Mattermost offering, such as Mattermost Professional or Mattermost Enterprise, you're entitled to open support tickets

via our online support portal (https://support.mattermost.com/hc/en-us/requests/new).

When opening a Support ticket as part of your paid subscription, it's important that you provide us with as much information as you can in a timely manner. Knowing what information is relevant can be confusing. We use the anagram C.L.U.E.S. to remember what we need:

- Configurations
- Logs
- Users affected
- Environment
- Steps to reproduce

C.L.U.E.S. represents all of the information that can clarify your issue. With these details, we can begin searching for a cause, whether it's a simple configuration change or a product bug. It also helps us when we need to escalate the issue to our developers so they can spend as much time as possible improving our product.

General guidelines for information

Follow these guidelines when providing diagnostic data to us:

- Make sure the files you provide are as complete as possible, rather than providing a few lines. Entire log files and configurations provide us with important context.
- Provide configuration and log files in plaintext format if possible, as these are far easier for us to search than screenshots.
- Be sure to sanitize configuration and log files to remove usernames, passwords, and LDAP groups. Replace these details with example strings that contain the same special characters if possible, as special characters are common causes of configuration errors.
- Provide screenshots or screen recordings of unexpected product behavior so that we know exactly what your users are seeing.

Configuration

Why we need your configuration data

On Linux systems, settings are generally stored in configuration files. Many issues can be resolved by enabling or disabling a configuration setting. In order to find a resolution, we need to have as complete a picture of your system setup as possible. This also helps us to reproduce bugs so our developers can fix them.

What configuration data includes

Configuration includes (but is not limited to):

- The Mattermost `config.json` file.
- The configuration for the reverse proxy, e.g. NGINX, HAProxy, AWS.
- The database configuration.
- SAML configuration when the issue is regarding SAML authentication. The configuration for the Mattermost service is in the SAML IdP.
- Any other systems that Mattermost connects to or systems that exist between the user and the Mattermost server.

How to access your configuration data

Mattermost configuration

The Mattermost configuration is usually stored at `/opt/mattermost/config/config.json`. If you've migrated the Mattermost configuration to the database, you can get the configuration using `mmctl` or by running this database query:

```
SELECT Value FROM Configurations WHERE Active = 1;
```

Reverse Proxy configuration

NGINX usually splits its configuration into two parts: the main server configuration at `/etc/nginx/nginx.conf`, and a virtual server configuration. On Ubuntu, this is stored in `/etc/nginx/sites-available`. Providing both of these configuration files is helpful, but providing the latter is more important.

SAML configuration

If the issue you're seeing is with SAML login, we will need to see the full configuration for the Mattermost service in the SAML provider. The configuration for the Mattermost service is in the SAML IdP. Providing screenshots similar to the ones in the setup documentation is sufficient because most SAML providers are configured using a web interface.

LDAP configuration

The LDAP administrator should confirm the correct values for the following Mattermost LDAP settings:

- LDAP server hostname.
- LDAP connection port, security, and certificates.
- BaseDN, bind username, and bind password.
- User, Group, Guest, and Admin filters.
- Display attributes.

These can be provided as a text file or as screenshots from the LDAP server.

Other configurations

If you're experiencing an issue on mobile, and you're using an MDM or VPN to connect to the server, those configurations will be necessary to diagnose the problem. A system admin for the external system should be able to provide you with the configuration.

Logs

Why we need them

Nearly all computer systems have logs of errors and application behavior that can show us what's happening when an application is running. Error logs are invaluable when diagnosing a problem, but only if they're as complete as possible.

What logs are available

Mattermost

Mattermost has two log files, one for general messages and the other for notification-related messages. These are found at:

- `/opt/mattermost/logs/mattermost.log`
- `/opt/mattermost/logs/notification.log`

Proxy

The location of these depend on your proxy configuration, but a good place to start looking is in `/var/log`. Your proxy administrator should be able to help you find the logs.

Database

PostgreSQL and MySQL have different logs, and their location varies based on your configuration. If the issue is related to database connectivity, check the database documentation to locate the logs.

SAML, LDAP, and other systems

Your organization's system admin should be able to find these for you.

How to access logs

Mattermost

Make sure debug logging is enabled so that we can get the most information from the logs. To do this, go to **System Console > Environment > Logging**, then set both **Console File Level** and **File Log Level** to **DEBUG**. Remember to save your changes.

If the behavior started at a known time or date, use `journalctl` to get the logs like this:

```
sudo journalctl -u mattermost --since "2020-08-23 17:15:00" >
mattermost_journalctl.log
```

Replace 2020-08-23 17:15:00 with the date and time (relative to the server) when the behavior started. To get the server time, use the `date` command. If the log files generated are too large to send, compress them with this command:

```
tar -czf /tmp/mattermost.log.tgz
```

The compressed logs will be located on the server at `/tmp/mattermost.log.tgz`.

If the compressed file is still too big, use these commands to split the compressed file into two or more 20MB files:

```
mkdir -p /tmp/mattermost-logs cd /tmp/mattermost-logs tar czf - /opt/
mattermost/logs/mattermost.log | split -b 20m - mattermost.log.tgz.
```

The compressed files will be located on the server at `/tmp/mattermost-logs` and be named `mattermost.log.tgz.aa`, `mattermost.log.tgz.ab`, and so on. Use a file transfer client that supports SSH/SFTP, such as Cyberduck, to copy these files from the server.

If you are experiencing issues with Elasticsearch, LDAP, or the database, you can enable trace logging in `config.json` by setting `Trace` to `true` under their respective settings. Combining this with `DEBUG` level file log output will result in huge log files, so only leave trace logging on long enough to replicate the behavior. The resulting logs will also contain a lot more sensitive data, including user data, so be sure to sanitize it completely before sharing it with us.

System logs

The location of log files for other systems varies, but a good way to get the logs for all processes on the Mattermost server is to use `journalctl` like this:

```
sudo journalctl --since "2020-08-23 17:15:00" > mattermost_journalctl.log
```

Replace 2020-08-23 17:15:00` with the date and time (relative to the server) when the error occurred. You can use `--until` with the same timestamp format to get the logs between two times:

```
sudo journalctl --since "2020-08-23 17:15:00" --until "2020-08-23 16:30:00"
> mattermost_journalctl.log
```

Users affected

Why we need it

Mattermost servers are chaotic places. Thousands of posts, websocket actions, and webhook calls happen every second while users can be in dozens of channels across multiple teams. Knowing which users are affected by a problem can help us sift through all this information to find the root cause.

What information to include

This should be a detailed explanation of anything the end users who are reporting the unexpected behavior have in common. This includes (but is not limited to):

- Team and Channel memberships, including Direct and Group Messages.
- Authentication methods.
- Client operating system and app versions.
- How users connect to the Mattermost server.
- Any other things these users have in common such as when they joined, whether their login information recently changed, or if they are being synchronized via LDAP.

Note for Agents: This information is also required:

- Customer name
- Customer contacts
- Customer license, e.g. Enterprise/Professional
- Customer tier

Environment

Where the Mattermost server sits in your architecture has a lot of impact on potential issues. For example, a misconfigured proxy server can prevent users from connecting even if there’s nothing wrong with Mattermost.

What information to include

Because of this, having a complete picture of the servers and network that the Mattermost server operates in is key to solving problems. This includes (but is not limited to):

- Mattermost version (e.g. 7.3.0, 7.8.3)
- Server OS and version (e.g. RHEL7, Ubuntu 20.04)
- Any orchestration/automation used like Docker or Kubernetes
- Reverse proxy and version (e.g. NGINX 1.16)
- Database type and version (e.g. PostgreSQL 13)
- SAML provider (e.g. Windows Server 2012 Active Directory, Okta, KeyCloak)
- LDAP provider (e.g. Windows Server 2016 Active Directory, Okta, OpenLDAP)
- The type and version of any proxies or VPNs on the network that the Mattermost server is connecting through

Be as specific as possible when describing the environment. If you are seeing errors like **Connection Refused** be sure to include any firewalls or filtering proxies that may be on your network, either inbound or outbound.

Examples

Mattermost server

> - *External hostname: mattermost.example.com*
> - *Internal hostname: mattermost.lan*
> - *Mattermost v7.3.0*
> - *Zoom plugin v1.4.1*
> - *NGINX v1.18.0*

Database server

> - *Internal hostname: postgresql.lan*

> - *PostgreSQL v13*
> - *LDAP Provider - 192.168.1.102*
> - *Internal hostname: ldap.lan*
> - *OpenLDAP 2.4.54 (Docker container)*

Mattermost servers

> - *Hostnames: mm1.local.lan, mm2.local.lan, mm3.local.lan, mm4.local.lan*

Mattermost server versions

> - *mm1-3: 5.25.4*
> - *mm4: 5.21.0*

Proxy server

> - *External hostname: mattermost.example.com*
> - *Internal hostname: proxy.local.lan*
> - *NGINX v1.16.0*

Database servers

> - *Hostnames: db1.local.lan, db2.local.lan, db3.local.lan*
> - *Primary: db1.local.lan*
> - *Read-Only: db2.local.lan, db3.local.lan*
> - *PostgreSQL v13*

Elasticsearch server

> - *Hostname: elastic.local.lan*
> - *Elasticsearch v7.9 with these plugins*
> - *analysis-icu*

Steps to reproduce

What it is

If the behavior only happens when the user performs a specific action, providing detailed steps to reproduce it will help us make sure we find and fix the right bug. These details should be as descriptive as possible, but nothing is better than a screenshot or a screen recording of the behavior.

A short summary of the steps to reproduce is also helpful. If you want some examples, look at the bug tickets on some Mattermost Jira tickets.

How to provide these details

macOS

Press `⌘ ⇧ 5` to open the screen recording tool and select the region of the screen you want to record. To take a screenshot, press `⌘ ⇧ 4` and select the region to take a screenshot. The screenshot files are placed on the desktop by default.

Windows

Press `Ctrl Shift S` to open the snipping tool to take a screenshot. If you want to take a screen recording you'll need to install third-party software such as OBS (https://obsproject.com/).

iOS

Take a screenshot or screen recording on iPhone (https://support.apple.com/guide/iphone/take-a-screenshot-iphc872c0115/ios).

Android

Take a screenshot or record your screen on your Android device (https://support.google.com/android/answer/9075928?hl=en).

Appendix

A note on mobile issues

Because the mobile app doesn't have a debug mode, diagnosing issues stemming from user data requires a proxy like Charles or mitmproxy. These will intercept and record traffic from the client which can then be replayed to reproduce issues. Talk to a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/) for help setting these up.

SAML login issues

If the issue is with SAML login, one important piece of context is the SAML login flow. This contains headers and authentication information that can reveal issues that are easy to fix. Follow these instructions to view the SAML login flow if you are experiencing SAML authentication.

Checking keys and certificates

Key and certificate files should never be shared, but if the error indicates a problem with the format of a key or certificate, then you should verify the format of the keys and certificates by running this command:

```
cat -A /path/to/key-or.cert
```

The output must meet these criteria exactly to be valid:

- Start with `-----BEGIN CERTIFICATE-----$`.
- All lines must end with `$`. If they end with `^M$` then convert them to UNIX line endings with `dos2unix`.
- End with `-----END CERTIFICATE-----$`.

On this page

- General deployment troubleshooting
  - Start Mattermost at system boot
  - Run Mattermost without a proxy
  - Review Mattermost logs
    - Mattermost Server logs
    - Mattermost Desktop App logs
    - Mattermost web logs
    - Mattermost Push Notification Service logs

- Review Mattermost environment
- Connect to another server
- Connect with another device
- Opening a support ticket for self-hosted deployments
  - General guidelines for information
  - Configuration
    - Why we need your configuration data
    - What configuration data includes
    - How to access your configuration data
  - Logs
    - Why we need them
    - What logs are available
    - How to access logs
  - Users affected
    - Why we need it
    - What information to include
  - Environment
    - What information to include
  - Steps to reproduce
    - What it is
    - How to provide these details
- Appendix
  - Checking keys and certificates

Software and hardware requirements

This guide outlines minimum software and hardware requirements for deploying Mattermost. Requirements may vary based on utilization and observing performance of pilot projects is recommended prior to scale out.

Deployment overview

Please see the Application architecture documentation for a summary of software systems and components whose requirements are described in this document.

Software requirements

Client software

Desktop apps

Operating System	Self-Hosted Technical Requirement	Cloud Technical Requirement
Windows	Windows 10+	Windows 10+
Mac	macOS 12+	macOS 12+
Linux	Ubuntu LTS releases 22.04 or later	Ubuntu LTS releases 22.04 or later

Though not officially supported, the Linux desktop app also runs on RHEL/CentOS 7+.

> Note
>
> - * Integrated Windows Authentication is not supported by Mattermost desktop apps. If you use ADFS we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).
> - The minimum content size is 769 x 600 pixels. When the width is 768 pixels or less, the desktop app switches to mobile view. When the height is below 600 pixels, screen elements may become misplaced.

PC web

Browser	Self-Hosted Technical Requirement	Cloud Technical Requirement
Chrome	v134+	v134+
Firefox	v128+	v128+
Safari	v17.4+	v17.4+
Edge	v134+	v134+

* Internet Explorer (IE11) is no longer supported. We recommend using the Mattermost desktop app (https://mattermost.com/apps/) or another supported browser. See this forum post (https://forum.mattermost.com/t/mattermost-is-dropping-support-for-internet-explorer-ie11-in-v5-16/7575) to learn more.

Mobile apps

Operating System	Technical Requirement
iOS	iPhone 6s devices and later with iOS 15.1+
Android	Android devices with Android 7+

> Note
>
> - * Integrated Windows Authentication is not supported by Mattermost mobile apps. If you use ADFS we recommend configuring intranet forms-based authentication for devices that do not support WIA (https://learn.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-intranet-forms-based-authentication-for-devices-that-do-not-support-wia).
> - The minimum and target content size is 320 x 460 pixels, matching the available space when the mobile app is opened in Safari on the minimum supported iOS device.

Mobile web

Browser	Technical Requirement
iOS	iOS 15.1+ with Safari 17.4+ or Chrome 134+
Android	Android 7+ with Chrome 134+

Email client

- *Desktop clients:* Outlook 2010+, Apple Mail version 7+, Thunderbird 38.2+

- *Web based clients:* Entra ID, Outlook, Gmail, Yahoo, AOL
- *Mobile clients:* iOS Mail App (iOS 7+), Gmail Mobile App (Android, iOS)

Server software

Mattermost server operating system

- Ubuntu, Debian Buster, CentOS 6+, CentOS 7+, RedHat Enterprise Linux 7+, Oracle Linux 6+, Oracle Linux 7+.
- Using the Mattermost Docker deployment (https://github.com/mattermost/docker) on a Docker-compatible operating system (Linux-based OS) is still recommended.

While community support exists for Fedora, FreeBSD, and Arch Linux, Mattermost does not currently include production support for these platforms.

Database software

- PostgreSQL 13.0+

Amazon Aurora equivalents of PostgreSQL is also supported.

Important

- MariaDB v10+ no longer functions as a MySQL drop-in replacement, and it's not supported for Mattermost due to the requirement of MySQL 5.7.12. Prior versions of MariaDB were not officially supported but may have functioned in older Mattermost releases. If you are running MariaDB now, migrating from MariaDB to the MySQL equivalent is recommended.
- Deployments requiring searching in Chinese, Japanese, and Korean languages require MySQL 5.7.6+ and the configuration of ngram Full-Text parser (https://dev.mysql.com/doc/refman/5.7/en/fulltext-search-ngram.html). For searching two characters, you will also need to set `ft_min_word_len` and `innodb_ft_min_token_size` to `2` and restart MySQL. See CJK discussion (https://github.com/mattermost/mattermost/issues/2033#issuecomment-183872616) for details.

Minimum PostgreSQL database support policy

To make planning easier and ensure your Mattermost deployment remains fast and secure, we are introducing a policy for updating the minimum supported version of PostgreSQL. The oldest supported PostgreSQL version Mattermost supports will match the oldest version supported by the PostgreSQL community. This ensures you benefit from the latest features and security updates.

This policy change takes effect from Mattermost v10.6, where the minimum PostgreSQL version required will be PostgreSQL 13. This aligns with the PostgreSQL community's support policy, which provides 5 years of support for each major version.

> Note
>
> Mattermost v10.6 is not an Extended Support Release (ESR). Going forward, this database version support policy will only apply to ESR releases.

When a PostgreSQL version reaches its end of life (EOL), Mattermost will require a newer version starting with the next scheduled ESR release. This means the following future PostgreSQL minimum version increases as follows:

Mattermost Version	Release Date	Minimum PostgreSQL Version
v9.11 ESR	2024-8-15	11.x
v10.5 ESR	2025-2-15	11.x
v10.6	2025-3-15	13.x
v10.11 ESR	2025-8-15	13.x
v11.5 ESR *	2026-2-15	14.x (EOL 2026-11-12)

* Forcasted release version and date.

Customers will have 9 months to plan, test, and upgrade their PostgreSQL version before the new requirement takes effect. This policy aims to provide clarity and transparency so you can align database upgrades with the Mattermost release schedule. Contact a [Mattermost Expert (https://mattermost.com/contact-sales/)](https://mattermost.com/contact-sales/). to discuss your options.

Database Search limitations

Common limitations:

- Only the initial **1 MB** of the file content is available for search, even though much bigger files can be uploaded.

Search limitations on PostgreSQL:

- Email addresses do not return results.
- Hashtags or recent mentions of usernames containing a dash do not return search results.
- Terms containing a dash return incorrect results as dashes are ignored in the search query.

- Limitations set by [PostgreSQL itself (https://www.postgresql.org/docs/current/textsearch-limitations.html)](https://www.postgresql.org/docs/current/textsearch-limitations.html):
  - One of them is: `The length of a tsvector (lexemes + positions) must be less than 1 megabyte`, which means that, based on the file content, even files with content less than 1 MB won't be searchable if they hit the `tsvector` limit of 1 MB.
- If any of the above is an issue, you can set up and enable enterprise search.

MySQL Support

MySQL database support is being deprecated starting with Mattermost v11. See the PostgreSQL migration documentation for guidance on migrating from MySQL to PostgreSQL.

- Search limitations on MySQL: Hashtags or recent mentions of usernames containing a dot do not return search results.
- The migration system requires the MySQL database user to have additional *EXECUTE*, *CREATE ROUTINE*, *ALTER ROUTINE* and *REFERENCES* privileges to run schema migrations.
- MariaDB v10+ no longer functions as a MySQL drop-in replacement, and it's not supported for Mattermost due to the requirement of MySQL 5.7.12. Prior versions of MariaDB were not officially supported but may have functioned in older Mattermost releases. If you are running MariaDB now, migrating from MariaDB to the MySQL equivalent is recommended.
- Deployments requiring searching in Chinese, Japanese, and Korean languages require MySQL 5.7.6+ and the configuration of [ngram Full-Text parser (https://dev.mysql.com/doc/refman/5.7/en/fulltext-search-ngram.html)](https://dev.mysql.com/doc/refman/5.7/en/fulltext-search-ngram.html). For searching two characters, you will also need to set `ft_min_word_len` and `innodb_ft_min_token_size` to `2` and restart MySQL. See [CJK discussion (https://github.com/mattermost/mattermost/issues/2033#issuecomment-183872616)](https://github.com/mattermost/mattermost/issues/2033#issuecomment-183872616) for details.

Important

MySQL 8.0.22 contains an [issue with JSON column types (https://bugs.mysql.com/bug.php?id=101284)](https://bugs.mysql.com/bug.php?id=101284) changing string values to integers

which is preventing Mattermost from working properly. Users are advised to avoid this database version.

In MySQL 8.0.4, the default authentication plugin was changed from `mysql_native_password` to `caching_sha2_password`. Therefore, you will need to enable `mysql_native_password` by adding the following entry in your MySQL configuration file:

> ```
> [mysqld] default-authentication-plugin=mysql_native_password
> ```

In MySQL 8, the default collation changed to `utf8mb4_0900_ai_ci` (https://dev.mysql.com/doc/mysqld-version-reference/en/optvar-changes-8-0.html). Therefore, if you update your MySQL installation to version 8, you'll need to convert your database tables to use the new default collation:

```
ALTER TABLE table_name CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci;
```

If this change isn't made, tables in the database may end up having different collations which will cause errors when executing queries.

In MySQL versions 8.0.0 - 8.0.11 `ADMIN` is a [reserved keyword (https://dev.mysql.com/doc/refman/8.0/en/keywords.html)](https://dev.mysql.com/doc/refman/8.0/en/keywords.html), which is why our requirement for MySQL is version 8.0.12.

MySQL 8.0.22 contains an [issue with JSON column types (https://bugs.mysql.com/bug.php?id=101284)](https://bugs.mysql.com/bug.php?id=101284) changing string values to integers which is preventing Mattermost from working properly. Users are advised to avoid this database version.

Hardware requirements

Usage of CPU, RAM, and storage space can vary significantly based on user behavior. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are.

Moreover, memory requirements can be driven by peak file sharing activity. Recommendation is based on default 50 MB maximum file size, which can be

adjusted from the System Console. Changing this number may change memory requirements.

For deployments larger than 2,000 users, it is recommended to use the Mattermost open source load testing framework to simulate usage of your system at full scale: https://github.com/mattermost/mattermost-load-test-ng.

Mattermost supports any 64-bit x86 processor architecture.

Hardware requirements for team deployments

Most small to medium Mattermost team deployments can be supported on a single server with the following specifications based on registered users:

- 1 - 1,000 users - 1 vCPU/cores, 2 GB RAM
- 1,000 - 2,000 users - 2 vCPUs/cores, 4 GB RAM

Hardware requirements for enterprise deployments (multi-server)

Scale requirements

For Enterprise Edition deployments with a multi-server setup, see our scaling guide.

It is highly recommended that pilots are run before enterprise-wide deployments in order to estimate full scale usage based on your specific organizational needs. You can use the Mattermost open source load testing framework to simulate usage of your system: https://github.com/mattermost/mattermost-load-test-ng.

Mattermost's performance monitoring tools can be used for detailed performance measurements and to inspect the running system to ensure sizing and installation is correct.

System requirements

For Enterprise Edition deployments with a multi-server setup, we highly recommend the following systems to support your Mattermost deployment:

- Prometheus to track system health of your Mattermost deployment, through performance monitoring feature available in Mattermost Enterprise.
- Grafana to visualize the system health metrics collected by Prometheus with the performance monitoring feature. Grafana 5.0.0 and later is recommended.
- Elasticsearch to support highly efficient database searches in a cluster environment. Elasticsearch v7.17+ is supported, and Elasticsearch v8.x or AWS OpenSearch is recommended from Mattermost v9.11. Learn more.
- MinIO or AWS S3. Mattermost is compatible with object storage systems which implement the S3 API. Other S3-compatible systems may work, but are not officially supported. Learn more about file storage configuration options in our documentation.

On this page

- Software and hardware requirements
  - Deployment overview
  - Software requirements
    - Client software
      - Desktop apps
      - PC web
      - Mobile apps
      - Mobile web
      - Email client
    - Server software
      - Mattermost server operating system
      - Database software
        - Minimum PostgreSQL database support policy
        - Database Search limitations
        - MySQL Support

- Hardware requirements
  - Hardware requirements for team deployments
  - Hardware requirements for enterprise deployments (multi-server)
    - Scale requirements
    - System requirements

Configuring transport encryption

The components of the Mattermost setup are shown in the following diagram, including the transport encryption used. Aside from the encryption between the nodes of the Mattermost cluster, all transports rely on TLS encryption.

> Note
>
> The transport between the Application servers is not used by default and requires additional setup steps. Enhancing the core product to include automatic encryption between cluster nodes is in progress and planned for a later release.

Configuring proxy to Mattermost transport encryption

Mattermost is able to encrypt the traffic between the proxy and the application server using TLS.

Prerequisites

- Operational Mattermost server or cluster.
- Authentication credentials for Mattermost user on application server.

Example environment

In this scenario there is one Mattermost application server and one NGINX server, both running Ubuntu 20.04, with the following IPs:

- **transport-encryption-mattermost1:** 10.10.250.146
- **transport-encryption-nginx:** 10.10.250.107

Configuring NGINX

On the NGINX server, connect to both servers with a sudo or root user. Open the Mattermost proxy configuration and search for the following line twice:

```
proxy_pass http://backend;
```

Change the protocol from `http` to `https`:

```
proxy_pass https://backend;
```

Afterwards do not reload the NGINX server yet to minimize the downtime of the service.

Configuring Mattermost

On the Mattermost server, change to the config directory of Mattermost and generate a self-signed certificate that will be used to encrypt the traffic between the proxy server and the application server.

Note: Alternatively you can sign a certificate from your company's CA.

```
cd /opt/mattermost/config openssl req -x509 -newkey rsa:4096 -keyout
key.pem -out cert.pem -days 365 -nodes chown root:mattermost *.pem chmod
640 *.pem
```

Once complete, open the file `config.json` and modify the values `ConnectionSecurity`, `TLSCertFile` and `TLSKeyFile` in the `ServiceSettings` section.

Before

```
{  "ServiceSettings": {  "SiteURL": "https://transport-
encryption.dev.example.com",  "WebsocketURL": "",  "LicenseFileLocation":
"",  "ListenAddress": ":8065",  "ConnectionSecurity": "",  "TLSCertFile":
"",  "TLSKeyFile": "",  "...":"..."  },  "...":"..." }
```

After

```
{  "ServiceSettings": {  "SiteURL": "https://transport-
encryption.dev.example.com",  "WebsocketURL": "",  "LicenseFileLocation":
"",  "ListenAddress": ":8065",  "ConnectionSecurity": "TLS",
"TLSCertFile": "/opt/mattermost/config/cert.pem",  "TLSKeyFile": "/opt/
mattermost/config/key.pem",  "...":"..."  },  "...":"..." }
```

Restart the Mattermost server and ensure it’s up and running:

```
sudo systemctl restart mattermost systemctl status mattermost
```

```
● mattermost.service - Mattermost Loaded: loaded (/lib/systemd/system/
mattermost.service; static; vendor preset: enabled) Active: active (running)
since Mon 2019-10-28 16:45:29 UTC; 1h 15min ago [...]
```

Finally, on the **NGINX server**, reload the configuration to ensure that requests are sent on HTTPS:

```
sudo systemctl reload nginx
```

Configuring database transport encryption

Mattermost is able to encrypt the traffic between the database and the application using TLS. This guide describes the setup steps for a single, separate MySQL server.

Prerequisites

- Operational Mattermost server or cluster.
- Operational MySQL server.

- Confirmed connectivity between Mattermost and MySQL server.
- Authentication credentials for Mattermost user on MySQL server.

Example environment

In this scenario there is one Mattermost application server and one MySQL server, both running Ubuntu 20.04, with the following IPs:

- **transport-encryption-mattermost1:** 10.10.250.146
- **transport-encryption-mysql1:** 10.10.250.148

Configuring MySQL

As a first step, connect to both servers with a sudo or root user.

Execute the following command to prepare the server for SSL connections:

```
sudo mysql_ssl_rsa_setup --uid=mysql
```

This generates self-signed certificates in `/var/lib/mysql/` that the MySQL server uses to encrypt the connection. If you would like to use certificates from your company CA, please follow the MySQL documentation for configuration steps.

Note: Optionally, it can be enforced that all connections must be made via a local socket connection or TLS. To do this, open `/etc/mysql/mysql.conf.d/mysqld.cnf` and append the following line to the file:

```
require_secure_transport = ON
```

Last but not least, restart the server and confirm it is up and running:

> ```
>
> ```
>
> *systemctl restart mysql systemctl status mysql*

```
● mysql.service - MySQL Community Server Loaded: loaded (/lib/systemd/
system/mysql.service; enabled; vendor preset: enabled) Active: active
(running) since Fri 2019-10-18 16:41:25 UTC; 2s ago Process: 8380
ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
(code=exited, status=0/SUCCESS) Process: 8360 ExecStartPre=/usr/share/mysql/
mysql-systemd-start pre (code=exited, status=0/SUCCESS) Main PID: 8382
(mysqld) Tasks: 27 (limit: 2361) CGroup: /system.slice/mysql.service
└─8382 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid Oct 18
16:41:25 transport-encryption-mysql1 systemd[1]: Stopped MySQL Community
Server. Oct 18 16:41:25 transport-encryption-mysql1 systemd[1]: Starting
MySQL Community Server... Oct 18 16:41:25 transport-encryption-mysql1
systemd[1]: Started MySQL Community Server.
```

Configuring Mattermost

On the Mattermost server, open the file `config.json` and look for the `DataSource` value in the `SqlSettings` section. It should look similar to this:

```
"DataSource": "mmuser:sad09zusaopdhsad123@tcp(10.10.250.148:3306)/
mattermost?charset=utf8mb4,utf8\u0026writeTimeout=30s",
```

At the end of the line, we can configure that TLS must be turned on with the `tls` flag which supports the following values:

- true (Require TLS + a trusted certificate)
- false
- skip-verify (Require TLS + accept self-signed)
- preferred (Try TLS, fallback to unencrypted)

In our case we need to use `skip-verify` since we use a self-signed certificate. The configuration setting will now look like this:

```
"DataSource": "mmuser:sad09zusaopdhsad123@tcp(10.10.250.148:3306)/
mattermost?charset=utf8mb4,utf8\u0026writeTimeout=30s&tls=skip-verify",
```

If you're running Mattermost in a cluster, be sure to update the value on each node of the cluster. If you are using configuration in the database, be sure to update the `systemd` unit file and enable TLS for the configuration store.

Once complete, restart the Mattermost server and ensure the system is operational:

```
sudo systemctl restart mattermost systemctl status mattermost
```

```
● mattermost.service - Mattermost Loaded: loaded (/lib/systemd/system/
mattermost.service; static; vendor preset: enabled) Active: active (running)
since Fri 2019-10-18 16:47:08 UTC; 3s ago Process: 3424 ExecStartPre=/opt/
mattermost/bin/pre_start.sh (code=exited, status=0/SUCCESS) Main PID: 3443
(mattermost) Tasks: 20 (limit: 2361) CGroup: /system.slice/
mattermost.service ├─3443 /opt/mattermost/bin/mattermost --config=mysql://
mmuser:sad09zusaopdhsad123@tcp(10.10.250.148:3306)/mattermost?
charset=utf8mb4,utf8&writeTimeout=30s&tls=skip-verify └─3459 plugins/
com.mattermost.nps/server/dist/plugin-linux-amd64 Oct 18 16:47:08 transport-
encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.8637397,"caller":"scheduler/
worker.go:36","msg":"Worker started","worker":"Plugins"} Oct 18 16:47:08
transport-encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.8639545,"caller":"jobs/
jobs_watcher.go:38","msg":"Watcher Started"} Oct 18 16:47:08 transport-
encryption-mattermost1 mattermost[3443]:
{"level":"info","ts":1571417228.8641603,"caller":"jobs/
schedulers.go:72","msg":"Starting schedulers."} Oct 18 16:47:08 transport-
encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.8645394,"caller":"app/
web_hub.go:436","msg":"Hub for index 0 is starting with goroutine 3923"} Oct
18 16:47:08 transport-encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.8648505,"caller":"app/
web_hub.go:436","msg":"Hub for index 1 is starting with goroutine 3924"} Oct
18 16:47:08 transport-encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.8656101,"caller":"web/
static.go:31","msg":"Using client directory at /opt/mattermost/client"} Oct
18 16:47:08 transport-encryption-mattermost1 mattermost[3443]:
{"level":"info","ts":1571417228.8681324,"caller":"commands/
server.go:105","msg":"Sending systemd READY notification."} Oct 18 16:47:08
transport-encryption-mattermost1 systemd[1]: Started Mattermost. Oct 18
16:47:08 transport-encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.9003174,"caller":"jobs/
schedulers.go:166","msg":"Next run time for
scheduler","scheduler_name":"MigrationsSched Oct 18 16:47:08 transport-
encryption-mattermost1 mattermost[3443]:
{"level":"debug","ts":1571417228.9025588,"caller":"jobs/
schedulers.go:166","msg":"Next run time for
scheduler","scheduler_name":"PluginsSchedule
```

Configuring cluster transport encryption

Mattermost is able to encrypt the messages sent within the cluster of a deployment using SSH tunneling. The guide walks through the deployment of this solution on Ubuntu 20.04, but it can be adapted for any Linux operating system.

While this document only describes the configuration of a three-node cluster, it is by no means limited to that number.

Prerequisites

- SSH port whitelisted between each node of the deployment.
- Active ufw/iptables on each node.
- Access to the root/sudo user of each node for configuration.
- A configured Mattermost cluster.
- Mattermost running with a dedicated service user.
- Mattermost service is stopped on each cluster node.

> Note
>
> Support on the application level is currently in development and, when available, will deprecate this document.

Example environment

In this scenario there are three application nodes in our environment with the following hostname/IP mapping:

- **transport-encryption-mattermost1:** 10.10.250.146
- **transport-encryption-mattermost2:** 10.10.250.231
- **transport-encryption-mattermost3:** 10.10.250.165

Preparations

- Connect to each Mattermost server with a sudo or root user.
- Make a note of the IP from each cluster member used for the internal communication.
- Ensure `AllowTcpForwarding` is enabled in `/etc/ssh/sshd_config` of each cluster node.

SSH authentication

On each node, generate a SSH key-pair for the service account. In our scenario this is called `mattermost`:

```
sudo -u mattermost ssh-keygen -t rsa
```

```
Generating public/private rsa key pair. Enter file in which to save the key
(/home/mattermost/.ssh/id_rsa): Enter passphrase (empty for no passphrase):
Enter same passphrase again: Your identification has been saved in /home/
mattermost/.ssh/id_rsa. Your public key has been saved in /home/
mattermost/.ssh/id_rsa.pub. The key fingerprint is: SHA256:redacted
mattermost@transport-encryption-mattermost1
```

The location of the SSH key itself is irrelevant if company policies require the usage of another storage location.

Next, ensure that the SSH public key of each node is added to the `authorized_keys` file of the other nodes of the cluster. To do so, copy the contents of `/home/mattermost/.ssh/id_rsa.pub` of nodes 2 and 3, and add it to `/home/mattermost/.ssh/authorized_keys` of node 1.

Repeat this step for each node of the cluster. As a result, each node should be able to establish an SSH connection to the other nodes of the cluster.

> Note
>
> This service account can be separate from the service account already used for the Mattermost `systemd` service itself. It's important that this service account is allowed to create a SSH tunnel with port forwarding, but it doesn't require any additional permissions.

ufw configuration

As a next step, allow SSH access from each of the other member nodes, e.g.:

- mattermost1 allows from mattermost2 and mattermost3
- mattermost2 allows from mattermost1 and mattermost3
- mattermost3 allows from mattermost1 and mattermost2

To do so, we add an exception in the firewall. The commands for `mattermost1` look as follows:

```
sudo ufw allow from 10.10.250.231/32 to any port ssh sudo ufw allow from
10.10.250.165/32 to any port ssh sudo ufw status
```

```
Rule added Rule added Status: active To Action From -- ------ ---- 22/tcp
ALLOW 10.10.250.10 8065/tcp ALLOW Anywhere 22/tcp ALLOW 10.10.250.231 22/tcp
ALLOW 10.10.250.165
```

Repeat the same steps on the other nodes, replacing the IPs with the ones from the other member nodes. Do so for each member node, excluding the node itself.

Next, open `/etc/ufw/after.rules` and add the following block to the bottom of the file:

```
*nat :POSTROUTING ACCEPT [0:0] :PREROUTING ACCEPT [0:0] -A OUTPUT -p tcp -d
10.10.250.231 --dport 8075 -j DNAT --to-destination 127.0.0.1:18075 -A
OUTPUT -p tcp -d 10.10.250.231 --dport 8074 -j DNAT --to-destination
127.0.0.1:18074 -A OUTPUT -p tcp -d 10.10.250.165 --dport 8075 -j DNAT --to-
destination 127.0.0.1:28075 -A OUTPUT -p tcp -d 10.10.250.165 --dport 8074
-j DNAT --to-destination 127.0.0.1:28074 COMMIT
```

Two lines always belong to a single node, so in a deployment with four nodes:

```
-A OUTPUT -p tcp -d ip_node_2 --dport 8075 -j DNAT --to-destination
127.0.0.1:18075 -A OUTPUT -p tcp -d ip_node_2 --dport 8074 -j DNAT --to-
destination 127.0.0.1:18074 -A OUTPUT -p tcp -d ip_node_3 --dport 8075 -j
DNAT --to-destination 127.0.0.1:28075 -A OUTPUT -p tcp -d ip_node_3 --dport
8074 -j DNAT --to-destination 127.0.0.1:28074 -A OUTPUT -p tcp -d ip_node_4
--dport 8075 -j DNAT --to-destination 127.0.0.1:38075 -A OUTPUT -p tcp -d
ip_node_4 --dport 8074 -j DNAT --to-destination 127.0.0.1:38074
```

Please be aware that the ports on the right side must be unique, so if you have a cluster of six nodes, use 8075 and 8074 with 1 to 5 in front of it. If the cluster is of bigger size, additional ports must be used.

Ensure that your operating system has IP forwarding enabled using the following command:

```
sysctl -w net.ipv4.ip_forward=1
```

After that, reload the ufw rules and confirm that the iptable rules were successfully created:

```
iptables -t nat -L
```

```
Chain PREROUTING (policy ACCEPT) target prot opt source destination Chain
INPUT (policy ACCEPT) target prot opt source destination Chain OUTPUT
(policy ACCEPT) target prot opt source destination DNAT tcp -- anywhere
10.10.250.231 tcp dpt:8075 to:127.0.0.1:18075 DNAT tcp -- anywhere
10.10.250.231 tcp dpt:8074 to:127.0.0.1:18074 DNAT tcp -- anywhere
10.10.250.165 tcp dpt:8075 to:127.0.0.1:28075 DNAT tcp -- anywhere
10.10.250.165 tcp dpt:8074 to:127.0.0.1:28074
```

Repeat those steps for every node on the cluster. At the end of this section the following should be configured:

- SSH access enabled in firewall from each cluster node to another.
- Per node 2 iptables rules for port 8074 and 8075.
- IP forwarding enabled.

SSH configuration

As a next step, ensure that the SSH tunnels are created as part of the Mattermost service start. To do so, create a file called `pre_start.sh` in `/opt/mattermost/bin` on `mattermost1`:

```
#!/bin/bash ssh -N -f -o ServerAliveInterval=60 -o ExitOnForwardFailure=yes
-L 18075:10.10.250.231:8075 10.10.250.231 || true ssh -N -f -o
ServerAliveInterval=60 -o ExitOnForwardFailure=yes -L
18074:10.10.250.231:8074 10.10.250.231 || true ssh -N -f -o
ServerAliveInterval=60 -o ExitOnForwardFailure=yes -L
28075:10.10.250.165:8075 10.10.250.165 || true ssh -N -f -o
ServerAliveInterval=60 -o ExitOnForwardFailure=yes -L
28074:10.10.250.165:8074 10.10.250.165 || true
```

> Note
>
> - We’re ignoring the error from the SSH connection itself in case a tunnel is already active. Otherwise the Mattermost server would fail to start.
> - Please make sure to back up this script in case of a version upgrade.

Afterwards, set the executable bit on the shell script:

```
chmod +x /opt/mattermost/bin/pre_start.sh
```

Open the systemd unit file of Mattermost and search for `Type=Notify`. After this, enter a `ExecStartPre` script that will be executed before Mattermost itself is started:

```
[Service] Type=notify ExecStartPre=/opt/mattermost/bin/pre_start.sh
```

Reload the systemd daemon afterwards:

```
systemctl daemon-reload
```

Repeat the same steps on each of the member nodes and adapt the node IPs and amount of entries for your environment.

Cluster start

Once each node is configured, restart the service on each cluster and confirm that it's running using the command below:

```
systemctl start mattermost systemctl status mattermost.service
```

```
● mattermost.service - Mattermost Loaded: loaded (/lib/systemd/system/
mattermost.service; static; vendor preset: enabled) Active: active (running)
since Fri 2019-10-04 19:44:20 UTC; 5min ago Process: 16734 ExecStartPre=/
opt/mattermost/bin/pre_start.sh (code=exited, status=0/SUCCESS)
```

Next, open the Mattermost System Console and confirm that each node is reporting successfully in the High Availability section.

On this page

- Configuring transport encryption
  - Configuring proxy to Mattermost transport encryption
    - Prerequisites
    - Example environment
    - Configuring NGINX
    - Configuring Mattermost
  - Configuring database transport encryption
    - Prerequisites
    - Example environment
    - Configuring MySQL
    - Configuring Mattermost

- Configuring cluster transport encryption
  - Prerequisites
  - Example environment
  - Preparations
  - SSH authentication
  - ufw configuration
  - SSH configuration
  - Cluster start

Secure Mattermost

Mattermost ships with several security features that can help organizations safeguard their data. Mattermost Enterprise Edition includes features designed to offer extra layers of security protection.

- Encryption options - Setup encryption for data in transit and at rest.
- Transport encryption - Secure data in transit between Mattermost and other services.
- Multi-factor authentication - Require users to provide a secure one-time code in addition to their username and password to log in to Mattermost.
- ID-only push notifications - Enable fully private mobile notifications to protect against iOS and Android notification infrastructure breaches.
- Enterprise mobility management - Secure mobile endpoints with management application configuration.
- Delegated granular administration - Grant user access to specific areas of the Mattermost System Console.
- Custom terms of service - Increase clarity on legal Mattermost expectations for internal employees and guests.
- Manage session length - Control how long user sessions remain active.
- User and group provisioning via AD/LDAP - Provision and synchronize users and groups to pre-defined roles.
- SAML-based single sign-on (SSO) - Enable login using a single user ID and password managed through a SAML 2.0 Service Provider.
- SAML SSO technical documentation - Technical details on SAML SSO.
- Certificate-based authentication - Identify a user or a device before granting access to Mattermost.
- Manage file sharing and downloads - Control file sharing and downloads in Mattermost.

Dependency Vulnerability Analysis

This document provides context on why certain third-party dependencies in Mattermost, although flagged as vulnerable by security scanners, do not pose a risk in Mattermost deployments.

This analysis is regularly updated as new vulnerability reports are received and evaluated.

Overview

Mattermost regularly scans its dependencies for known vulnerabilities. Some dependencies may be flagged as vulnerable by security scanners, but these vulnerabilities might not be applicable to Mattermost due to:

- How the dependency is used in Mattermost
- The specific version or configuration implemented
- Mitigations already in place
- False positives in the scanning process

Dependency Analysis Table

Below is a list of dependencies flagged as vulnerable by security scanners, along with the justification for why each issue is not relevant to Mattermost deployments:

Dependency / Version	Vulnerability	False Positive Justification
github.com/mholt/archiver/v3 v3.5.1	GHSA-rhh4-rh7c-7r5v CVE-2024-0406	Mattermost doesn't use the vulnerable `Unarchive()` function.
github.com/mholt/archiver/v3 v3.5.1	GHSA-7vpp-9cxj-q8g CVE-2025-3445	Mattermost doesn't use the vulnerable `Unarchive()` function.
github.com/redis/go-redis/v9 v9.7.0	GHSA-92cp-5422-2mw7 CVE-2025-29923	Mattermost doesn't use this transitive dependency.
github.com/blevesearch/bleve/v2 v2.4.4-0.20250115090822-cbafdca08538	GHSA-9w9f-6mg8-jp7w CVE-2022-31022	Mattermost doesn't use the vulnerable `bleve/http` package.
golang stdlib v1.23.7	CVE-2025-22871	Mattermost doesn't use the vulnerable functions `chunkedReader.Read` and `readChunkLine`.

On this page

- Dependency Vulnerability Analysis
  - Overview
  - Dependency Analysis Table

Mobile Security

Continuous compliance through secure SDLC practices and proactive vulnerability management further reinforces the platform's resilience.

Mobile Device Management (MDM)

Mattermost supports the ability for an EMM provider to push Mattermost Mobile apps to EMM-enrolled devices. This approach is recommended for organizations that typically use EMM solutions to deploy Mobile apps to meet security and compliance policies. Learn more about deploying Mattermost mobile using an EMM provider.

Remote wipe capability

Administrators can remotely wipe Mattermost data from mobile devices in case of loss or theft. This capability prevents unauthorized access to sensitive information by ensuring that data is erased from compromised devices.

Compliance policies

Mattermost can be integrated with mobile device management solutions to enforce compliance policies. These policies ensure that mobile devices accessing the application adhere to security standards, such as encryption, password complexity, and device integrity. Learm more about compliance with Mattermost.

Mobile access platforms

Mattermost mobile applications can be operated under the protection of mobile access platforms like Hypori (https://www.hypori.com/). These platforms provide an additional layer of security by creating a virtualized environment for mobile applications, ensuring that sensitive data is isolated from the device's operating system. This approach enhances data protection and minimizes the risk of data leakage or unauthorized access.

Jailbreak and root detection

Jailbreaking or rooting a device disables many built-in security measures, making the device prone to malware and unauthorized access. This protection enables Mattermost administrators to deny access from any mobile device that appears to have been tampered with or is running a non-standard version of the operating system. Additionally, a rooted device may stop checking for software updates and security patches, and it might not be able to install them because the kernel is no longer properly signed.

By detecting such devices, Mattermost ensures that only secure, uncompromised devices can access sensitive data. When enabled by the system admin, this proactive measure ensures the underlying mobile platform operates reliably and performs the expected kernel and operating system security protections before sending any customer data to the Mattermost application, and minimizes risk in environments where personal device security cannot be guaranteed. Learn more about Mattermost mobile jailbreak and root detection.

Biometric authentication

Native biometric authentication ensures only the authorized device owner can access the Mattermost application. By utilizing hardware-level security, biometrics significantly enhance data protection, especially in cases of lost or stolen devices. This advanced security measure is far more robust and user-friendly compared to traditional passwords, adding a resilient layer of protection against unauthorized access.

Administrators can mandate biometric authentication each time users attempt to open the Mattermost application, further safeguarding customer data and mitigating risks. Learn more about Mattermost mobile biometric authentication, and the user workflows in which users must authenticate, when biometric authentication is enabled.

Screenshot and screen recording prevention

Preventing screenshots and screen recordings protects sensitive information from being inadvertently or maliciously shared. This control is essential in ensuring that confidential communications and data remain within the secure

confines of the app. By blocking unauthorized screen captures, Mattermost significantly reduces the risk of data leakage via visual content by preventing users from taking screenshots or recording their screens on the mobile device. This feature ensures that users can not intentionally or inadvertently capture an image of any information shared in Mattermost. Learn more about Mattermost mobile screenshot and screen recording prevention.

App sandboxing and secure data storage

Sandboxing is a critical defense that isolates the application's data from other apps—even if malicious software is present on the device. This isolation helps maintain user privacy and data integrity by ensuring that only Mattermost has access to its stored data. Learn more about Mattermost mobile app sandboxing and secure data storage.

Learn more about how Mattermost leverages robust sandboxing mechanisms on both iOS and Android to securely store files in its cache folder within the application container, ensuring isolation from unauthorized third-party apps.

Push notification message visibility

Push notifications are a convenient way to stay updated, but they can also pose security risks if sensitive information is displayed. Mattermost provides options to control the visibility of message content in push notifications, ensuring that sensitive information is not inadvertently exposed through locked mobile screens and via relay servers from Apple and Google when sending notifications to iOS or Android mobile apps.

Disable downloads

Environments with strict data loss prevention (DLP) policies or where sensitive information must not be stored on mobile devices can benefit from disabling file uploads and downloads on mobile devices.

Disabling file uploads adds an additional layer of security by reducing the risk of malware or malicious files being introduced into the system, ensuring tighter control over sensitive corporate data, and preventing accidental leaks from unsecure mobile networks.

Similarly, by disabling downloads, Mattermost ensures that files cannot be saved locally on the device, reducing the risk of unauthorized access or data leakage. Learn more about disabling mobile uploads and disabling mobile downloads in the Mattermost mobile app.

On this page

- Mobile Security
  - Mobile Device Management (MDM)
    - Remote wipe capability
    - Compliance policies
    - Mobile access platforms
  - Jailbreak and root detection
  - Biometric authentication
  - Screenshot and screen recording prevention
  - App sandboxing and secure data storage
  - Push notification message visibility
  - Disable downloads

Zero Trust with Mattermost

Mattermost helps organizations adopt and implement Zero Trust principles to safeguard their mission-critical communications and collaboration.

For security teams, Mattermost's zero-trust-first approach ensures consistent compliance with organizational risk policies by automating key governance processes like incident response or data lifecycle management.

Unlike traditional security approaches, Zero Trust assumes every user and system may be a potential threat. Mattermost implements this paradigm by offering customizable, secure solutions that protect sensitive communication workflows from both internal and external risks.

This document outlines how Mattermost supports the core tenets of Zero Trust, for organizations of different sizes, including Identity and access management, Continuous monitoring, Deployment and host control, Encryption, Micro-segmentation, Multi-factor authentication, Data management, and Incident response. Links to detailed documentation resources are provided below.

Identity and access management

Mattermost integrates seamlessly with enterprise identity providers (IdPs), enabling strong identity verification and strict access control.

By using one of the secure identity mechanisms listed below and enforcing least-privilege access via roles and groups, Mattermost ensures that only verified individuals gain access to the platform and its resources:

- SAML (https://docs.mattermost.com/administration-guide/onboard/sso-saml.html): Enables seamless Single Sign-On, ensuring centralized authentication to continuously enforce user verification.
- LDAP (https://docs.mattermost.com/administration-guide/onboard/ad-ldap.html): Facilitates integration with enterprise directories to tightly control user access, adhering to granular identity verification.

- OpenID Connect (https://docs.mattermost.com/administration-guide/configure/authentication-configuration-settings.html#openid-connect): Provides secure, standards-based user authentication to verify identities and enforce secure access.

- Session Management (https://docs.mattermost.com/administration-guide/configure/environment-configuration-settings.html#session-lengths): Strengthens continuous authentication by controlling session lengths and automatically revoking sessions based on inactivity or policy violations, ensuring constant identity verification. By limiting session lifetimes and enforcing strict session policies, Mattermost mitigates the risk of stolen session tokens or extended unauthorized access.

Authorized users can seamlessly be added and removed from channels utilizing the native AD/LDAP integration based on group memberships:

- LDAP Synchronized User Groups (https://docs.mattermost.com/administration-guide/onboard/ad-ldap-groups-synchronization.html): Automates user management and access control by dynamically syncing with organizational directories to minimize risks and enforce policies.

Continuous monitoring

Mattermost offers tools for monitoring activity, identifying suspicious behavior, session management, and real-time incident response. Audit trails and performance monitoring ensure the proactive detection of potential issues or breaches, delivering visibility into the activity across the platform.

- Audit Logging (https://docs.mattermost.com/administration-guide/manage/logging.html): Tracks detailed activity logs for monitoring and identifying real-time anomaly-detection use cases, such as detecting anomalous behavior from compromised accounts or insider threats, or responding to unusual file-sharing activity within sensitive channels.

- SIEM Integrations (https://developers.mattermost.com/integrate/webhooks/): Streamlines monitoring within existing security systems to detect and respond to lateral movement threats or policy violations consistently.

- Performance Monitoring (https://docs.mattermost.com/administration-guide/scale/deploy-prometheus-grafana-for-performance-monitoring.html):

Protects against potential threats by analyzing system and user behaviors via proactive monitoring.

Deployment and host control

Flexibility and control to host Mattermost securely to minimize the risk of vulnerabilities, downtime, and unauthorized modifications by ensuring secure, efficient, and reliable deployment of applications while maintaining strict control over the hosting environment.

Mattermost's self-hosting enables tailored configurations for on-premises systems with specialized security needs, while cloud IP filtering ensures scalable control for remote or hybrid teams operating across distributed environments:

- Self-hosting Mattermost: Enforces stricter data sovereignty requirements, and complete control over deployment environments, enabling organizations to implement custom Zero Trust security measures.
- Cloud IP Filtering: Prevents untrusted entities from gaining initial access, restricting platform access to trusted network ranges, enforcing an evaluation of every connection.

Encryption

Encryption protects both data at rest and data in transit, ensuring end-to-end security for sensitive communications. Encryption mitigates the risk of data theft in both storage and transfer, while granular permissions limit access to sensitive files and data to only authorized users.

- Database Encryption (https://docs.mattermost.com/deployment-guide/encryption-options.html#database): Protects user and organizational data at rest, safeguarding sensitive information from unauthorized access.
- Transport Layer Security (TLS) Encryption (https://docs.mattermost.com/deployment-guide/encryption-options.html#encryption-in-transit): Secures data in transit by encrypting communications.
- Policy Enforcement (https://docs.mattermost.com/deployment-guide/encryption-options.html#file-storage): Ensures strict compliance through automated enforcement, protecting data integrity.

Micro-segmentation

Segmenting and isolating sensitive resources is vital in minimizing lateral movement during an attack. Mattermost supports micro-segmentation through its organizational and role-based capabilities. Micro-segmentation enables organizations to restrict access to sensitive conversations, ensuring secure communication channels tailored to individual teams or missions.

To achieve comprehensive micro-segmentation, the following areas of Mattermost functionality play a critical role.

Access control and permissions

Ensure precise and role-based access to sensitive resources in order to minimize the risk of unauthorized access and potential data breaches:

- Advanced Access Controls (https://docs.mattermost.com/administration-guide/manage/team-channel-members.html#advanced-access-controls): Enforces specific permissions configurations to restrict access based on roles.
- Playbook-Specific Permissions (https://docs.mattermost.com/end-user-guide/workflow-automation/share-and-collaborate.html): Controls access to sensitive workflows, ensuring resources are available only to authorized team members.

Organizational design and user management

Establish a structured and scalable framework for managing user identities, roles, and access workflows to ensure accountability, facilitate collaboration, and enforce security policies:

- Teams (https://docs.mattermost.com/end-user-guide/collaborate/organize-using-teams.html): Enables segmentation of access and collaboration, fostering compartmentalization and limiting exposure to unauthorized users.
- Private Channels (https://docs.mattermost.com/end-user-guide/collaborate/channel-types.html#private-channels): Restricts conversations to authorized participants, protecting sensitive data and adhering to need-to-know principles.

- Guest Accounts (https://docs.mattermost.com/administration-guide/onboard/guest-accounts.html): Enables secure, scoped access for external parties, ensuring least privilege principles are maintained.
- Custom User Groups (https://docs.mattermost.com/end-user-guide/collaborate/organize-using-custom-user-groups.html): Allows precise administrative control of access and permissions for specific user sets, enhancing access segmentation.

Administrative controls

Enforce logical segmentation through team-level and group-level management, enhancing productivity and security by aligning user access with their specific roles:

- Delegated Granular Administration (https://docs.mattermost.com/administration-guide/onboard/delegated-granular-administration.html): Ensures operational security by enabling controlled management access based on responsibilities.
- Custom Terms of Service (https://docs.mattermost.com/administration-guide/comply/custom-terms-of-service.html): Requires users to acknowledge organization-specific Terms of Service before access ensures alignment with security policies and strengthens compliance, particularly in regulated industries where custom terms may reflect specific mandates.
- Granular Permissions (https://docs.mattermost.com/administration-guide/onboard/delegated-granular-administration.html): Facilitates precise control over user and system permissions, adhering to the principle of least privilege.
- Read-Only Permissions for Files (https://docs.mattermost.com/administration-guide/configure/site-configuration-settings.html#file-sharing-and-downloads): Limits file-sharing capabilities to safeguard sensitive information from unauthorized alterations.

Security policies and tokens

Enhance security with tailored authentication tools to protect systems and data from unauthorized API usage and credential misuse by establishing and enforcing secure, consistent, and scalable authentication mechanisms:

- Personal Access Tokens (https://developers.mattermost.com/integrate/reference/personal-access-token/): Enables secure API access with identity verification aligned to least privilege.

Multi-factor authentication (MFA)

Mattermost supports MFA to strengthen authentication practices by adding an extra layer of protection for high-risk workflows beyond passwords:

- MFA (https://docs.mattermost.com/administration-guide/onboard/multi-factor-authentication.html): Enhances user identity verification by requiring multiple factors for authentication. MFA ensures that unauthorized users are denied access even if passwords are compromised, reducing the risk of account breaches.

Alternatively, often enforced through the identity provider (IDP).

Data management

Data management directly addresses how sensitive information is managed, controlled, and safeguarded at every stage of the data lifecycle. Proper data retention practices ensure that data is not only securely stored but also that it is not retained longer than necessary, thereby reducing risks.

By retaining data only for the duration that it is needed and then securely disposing of it, the exposure to malicious activity or unauthorized access is significantly reduced. Even if attackers gain access, their exposure is minimized. The less data stored, the smaller the “footprint” for potential exploitation:

- Data Retention Policies (https://docs.mattermost.com/administration-guide/comply/data-retention-policy.html): Enforces strict retention controls to reduce data exposure and help comply with governance standards.

- Compliance Export (https://docs.mattermost.com/administration-guide/comply/compliance-export.html): Ensures data portability for audit and compliance purposes in a secure and controlled manner.
- Compliance Monitoring (https://docs.mattermost.com/administration-guide/comply/compliance-monitoring.html): Offers visibility into adherence to security and compliance policies, supporting compliance mandates.
- E-Discovery (https://docs.mattermost.com/administration-guide/comply/electronic-discovery.html): Boosts organizational oversight by ensuring discoverability of stored data for legal and compliance audits under secure protocols. E-Discovery capabilities help organizations meet compliance expectations for legal audits under frameworks like GDPR or HIPAA without sacrificing secure collaboration workflows.
- Archiving Inactive Teams or Channels (https://docs.mattermost.com/administration-guide/manage/team-channel-members.html#archive-a-team) & Unarchive Channels (https://docs.mattermost.com/end-user-guide/collaborate/archive-unarchive-channels.html): Reduces the potential attack surface by securely deactivating and storing inactive resources, minimizing both live data exposure and the likelihood of exploitation. This approach ensures adherence to security best practices while maintaining the ability to securely restore resources if needed.

Incident response

Incident response ensures that organizations can effectively detect, investigate, and respond to security threats within a framework that assumes no entity, whether inside or outside the network, should be trusted by default. Incident response is the operational arm that ensures that organizations are vigilant, prepared, and capable of protecting themselves in a dynamic and evolving threat landscape.

Mattermost Playbooks reduce the time to respond to threats and ensure compliancy-aligned documentation through automated incident notifications by empowering organizations to predefine and automate incident response workflows, ensuring that responses are consistent, documented, and transparent:

- Incident-Specific Channels for Secure Collaboration (https://docs.mattermost.com/end-user-guide/workflow-automation/work-with-

playbooks.html#actions): Maintains secure collaboration workflows across broader incident response workflows involving external tools, enforcing a centralized control model for operational continuity during incidents. Incident-specific channels reduce the time to assemble expert response teams, ensuring faster mitigation of active threats like phishing or ransomware attacks.

- Automated Incident Notifications (https://docs.mattermost.com/end-user-guide/workflow-automation/notifications-and-updates.html): Streamlines response workflows with authenticated alerts.

Enhance learning from incidents, ensure historical accountability, reduce future attack surfaces, and meet compliance expectations by securely centralizing documentation to improve future response processes:

- Post-Incident Documentation: Enables secure storage and access for learnings, ensuring compliance with attack surface minimization principles.

By embedding Zero Trust principles across access, monitoring, data management, and incident response, Mattermost equips organizations with the tools needed to safeguard collaboration workflows in today's evolving threat landscape.

Discover how Mattermost can transform your Zero Trust strategy today. Book a live demo with a Mattermost Zero Trust expert (https://mattermost.com/contact-sales/) to explore tailored solutions for your organization's secure collaboration needs.

On this page

- Zero Trust with Mattermost
  - Identity and access management
  - Continuous monitoring
  - Deployment and host control
  - Encryption
  - Micro-segmentation
    - Access control and permissions
    - Organizational design and user management
    - Administrative controls
    - Security policies and tokens

- Multi-factor authentication (MFA)
- Data management
- Incident response

Security Guide

Mattermost offers a comprehensive set of security features to protect information in its mobile applications. These features, ranging from encryption and authentication to device management and user education, provide a secure environment for team collaboration and communication. By prioritizing security, Mattermost ensures that customers can trust the mobile application to safeguard their data and privacy.

Data-in-transit encryption

Mattermost can be configured to use Transport Layer Security (TLS) to encrypt data transmitted over the network. TLS provides a secure channel for data exchange, protecting it from eavesdropping and tampering during transmission. This encryption ensures that only the intended recipients can access the content, preventing unauthorized parties from intercepting or reading the information.

Mattermost administrators who wish to configure access settings for private and public networks can do so with 3rd party infrastructure using a reverse proxy. This ensures that devices connected to secure networks can safely access the application while restricting access from untrusted networks. Learn more about Mattermost data-in-transit encryption.

Data-at-rest encryption

Encryption-at-rest ensures that messages, files, and other data stored in the Mattermost database and file storage are protected from unauthorized access by safeguarding data on physical storage media (e.g., disks) by encrypting it, making it inaccessible without the appropriate encryption keys. Learn more about Mattermost data-at-rest encryption.

Encryption-at-rest also available for files stored in Amazon’s proprietary S3 system using server-side encryption with Amazon S3-managed keys (Mattermost Enterprise) when users choose not to use open source options.

We strongly recommend regularly rotating and securely storing encryption keys using tools, enabling logging and monitoring for access to encrypted data, and ensuring that backup data is encrypted.

Authentication and access control

Single Sign-On (SSO)

The mobile application integrates with Single Sign-On providers, allowing users to authenticate using their existing credentials from other trusted systems. This reduces the risk of password-related security breaches and streamlines the login process. Learn more about Mattermost SSO.

Multi-Factor Authentication (MFA)

An additional layer of security beyond username and password. Customers can enable and enforce MFA to protect accounts from unauthorized access, even if login credentials are compromised.

User password requirements

System administrators can configure user password settings to help safeguard the platform against a range of common attack vectors while maintaining usability and compliance with enterprise security policies:

- Enforcing longer passwords ensures a baseline level of strength for every user's credentials. Learn more about configuring a minimum password length.
- Enforcing character complexity protects against attackers exploiting weak or overly simple passwords by enforcing passwords that resist dictionary attacks and common password vulnerabilities. Learn more about configuring password requirements.
- Limiting the number of failed authentication attempts before locking the account temporarily or permanently mitigates brute-force, where attackers attempt to guess passwords by repeatedly entering potential combinations. Learn more about configuring the maximum number of login attempts.

- Enabling the forgot password flow adds a layer of convenience by ensuring users can reset their password when needed while preventing users from being locked out due to legitimate loss of credentials. Learn more about enabling a password reset workflow.

Session management

System administrators can configure session management settings, including session length, session cache, and idle timeout to ensure user sessions are managed effectively and securely. Session fixation attacks are mitigated as Mattermost sets a new session cookie with each login. Learn more about session management configuration settings.

Protection against brute force attacks

System administrators can rate limit Mattermost APIs based on query frequency, memory store size, remote address, and headers.

Remote session revocation & password reset

System administrators can remotely revoke user sessions across web, mobile devices, and desktop apps. User passwords can be remotely reset to enhance security.

Admins can also enforce re-login after a specified period of time by defining session lengths and by revoking user sessions to force users to log back into the system immediately.

Role-Based Access Control (ABAC)

Administrators can set granular permissions to control access to sensitive information within the application. This feature ensures that users only have access to the data necessary for their roles, minimizing the risk of accidental or intentional data exposure. Learn more about Mattermost role-based access control.

Cross-origin requests control

Choose whether to restrict or enable cross-origin requests for enhanced control.

Public link management

Public links for account creation, file, and image shares can be invalidated by regenerating salts to ensure security.

Public links can also be disabled by setting the public link salt to an empty string. This prevents the creation of new public links and invalidates existing ones.

LLM context management

Mattermost Agents are designed to ensure that only necessary information is sent to the Large Language Model (LLM) for generating accurate responses in Mattermost. Learn how Mattermost Agents manages LLM context and how to ensure data privacy.

Audit logs and monitoring

Mattermost writes logs to both the console and to a log file in a machine-readable JSON format. Commercial customers can additionally log directly to syslog and TCP socket destination targets. Learn more about Mattermost logging.

Activity monitoring

Mattermost stores a complete history of messages, including edits and deletes, along with all files uploaded. User interface actions for deleting messages and channels only remove the data from the user interface; the data is retained within your database. If your compliance guidelines require it, you can disable users' ability to edit and delete their messages after they are posted. Learn more about Mattermost permissions.

The Mattermost mobile app generates audit logs that record user activities and system events. These logs enable administrators to monitor access and identify potential security threats, ensuring timely detection and response to suspicious behavior.

Regular security updates

Patch management

Mattermost regularly releases security updates to address vulnerabilities and enhance the application's security posture. Users are encouraged to keep their mobile applications up to date to benefit from the latest security improvements. Learn more about Mattermost releases and the release life cycle.

Community and expert contributions

Being an open-core platform, Mattermost benefits from contributions from the security community and experts. These contributions help identify and mitigate potential security risks, ensuring that the mobile application remains robust and secure. Learn more about contributing to Mattermost (https://mattermost.com/contribute/).

HIPAA compliance*

Deploying Mattermost as part of a HIPAA-compliant IT infrastructure requires a deployment team trained on HIPAA-compliance requirements and standards (https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/).

HIPAA-compliant deployments commonly consider the following:

- Omitting the contents of messages from mobile push and email notifications:
  - If your Push Notifications Contents option is set to `Send full message snippet` there is a chance Personal Health Information (PHI) contained in messages could be displayed on a user's locked phone as a notification. To avoid this, set the option to `Send generic description with user and channel names` or `Send generic description with only sender name`.
  - Similarly, setting Email Notifications Contents to `Send generic description with only sender name` will only send the team name and name of the person who sent the message, with no information about channel name or message contents included in email notifications.

- Beyond Technical Safeguards, HIPAA compliance deployments also require:
  - Administrative Safeguards
  - Physical Safeguards
  - Organizational requirements and other standards.

To learn more, please review HIPAA requirements from the US Department of Health and Human Services (https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/).

FINRA compliance*

Mattermost Enterprise features help users to meet the cybersecurity requirements of the United States Financial Industry Regulatory Authority (FINRA) (https://www.finra.org/rules-guidance/key-topics/cybersecurity) as part of a customer's existing operational systems, including technology governance, system change management, risk assessments, technical controls, incident response, vendor management, data loss prevention, and staff training.

FINRA reviews a firm's ability to protect the confidentiality, integrity, and availability of sensitive customer information. This includes reviewing each firm's compliance with SEC regulations, including:

- Regulation S-P (17 CFR §248.30) (https://www.ecfr.gov/current/title-17/chapter-II/part-248/subpart-A/subject-group-ECFR83262a0bce5ffaa/section-248.30), which requires firms to adopt written policies and procedures to protect customer information against cyber-attacks and other forms of unauthorized access.
- Regulation S-ID (17 CFR §248.201-202) (https://www.ecfr.gov/current/title-17/chapter-II/part-248), which outlines a firm's duties regarding the detection, prevention, and mitigation of identity theft.
- The Securities Exchange Act of 1934 (17 CFR §240.17a-4(f)) (https://www.ecfr.gov/current/title-17/chapter-II/part-240), which requires firms to preserve electronically stored records in a non-rewriteable, non-erasable format.

Mattermost supports FINRA compliance as part of a customer's integrated operations in the following ways:

- **Continuous archiving:** Configuration as a non-rewriteable, non-erasable system of record for all messages and files entered into the system. Moreover, automated compliance exports and integration support for Smarsh/Actiance and Global Relay provide third-party eDiscovery options.
- **Secure deployment:** Deployment within private, public, and on-premises networks with existing FINRA-compliant safeguards and infrastructure to protect customer information from cyber attack.
- **Support for intrusion detection:** Ability to support multi-layered intrusion detection from authentication systems to application servers to database access, including configuration of proxy, application, and database logging to deeply audit system interactions.
- **Multi-layered disaster recovery:** High Availability configuration, automated data back up, and enterprise information archiving integration to prevent data loss and recover from disaster.

***DISCLAIMER:** MATTERMOST DOES NOT POSITION ITS PRODUCTS AS "GUARANTEED COMPLIANCE SOLUTIONS". WE MAKE NO GUARANTEE THAT YOU WILL ACHIEVE REGULATORY COMPLIANCE USING MATTERMOST PRODUCTS. YOUR LEVEL OF SUCCESS IN ACHIEVING REGULATORY COMPLIANCE DEPENDS ON YOUR INTERPRETATION OF THE APPLICABLE REGULATION, AND THE ACTIONS YOU TAKE TO COMPLY WITH THEIR REQUIREMENTS. SINCE THESE FACTORS DIFFER ACCORDING TO INDIVIDUALS AND BUSINESSES, WE CANNOT GUARANTEE YOUR SUCCESS, NOR ARE WE RESPONSIBLE FOR ANY OF YOUR ACTIONS. NO GUARANTEES ARE MADE THAT YOU WILL ACHIEVE ANY SPECIFIC COMPLIANCE RESULTS FROM THE USE OF MATTERMOST OR FROM ANY RECOMMENDATIONS CONTAINED ON OUR WEBSITES, AND AS SUCH, THIS SHOULD NOT BE A SUBSTITUTE TO CONSULTING WITH YOUR OWN LEGAL AND COMPLIANCE REPRESENTATIVES ON THESE MATTERS.

On this page

- Security Guide
  - Data-in-transit encryption
  - Data-at-rest encryption

- Authentication and access control
  - Single Sign-On (SSO)
  - Multi-Factor Authentication (MFA)
  - User password requirements
  - Session management
  - Protection against brute force attacks
  - Remote session revocation & password reset
  - Role-Based Access Control (ABAC)
  - Cross-origin requests control
- Public link management
- LLM context management
- Audit logs and monitoring
  - Activity monitoring
- Regular security updates
  - Patch management
  - Community and expert contributions
- HIPAA compliance*
- FINRA compliance*