# Qumulo Alerts Administrator Guide

Version 3.3.0

December 07, 2022

# Table of Contents

# Getting Started with Qumulo Alerts

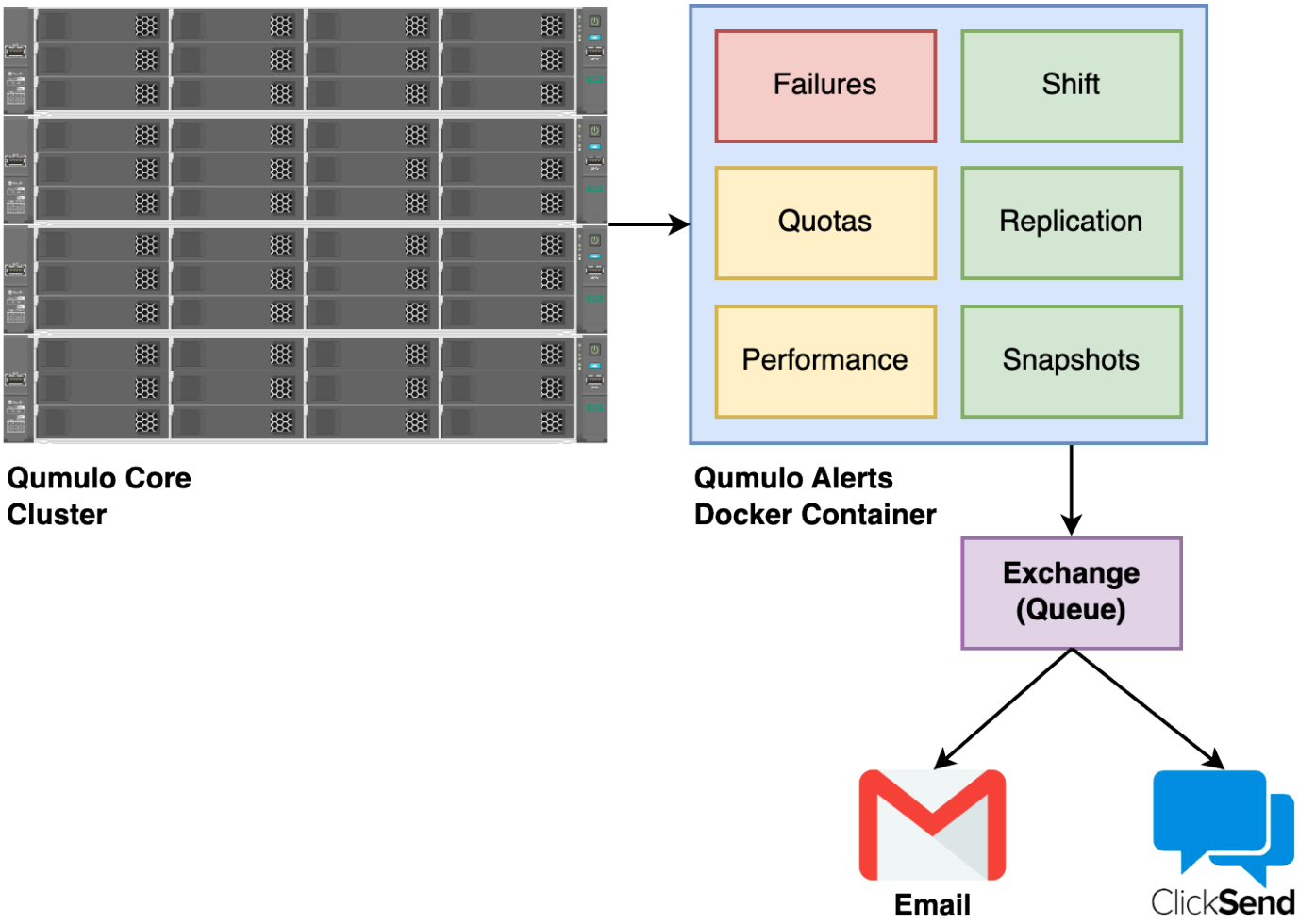
## How Qumulo Alerts Works with Qumulo Core

This section explains how Qumulo Alerts monitors alarms and alerts for a Qumulo Core Cluster.

### How Qumulo Alerts Works

Qumulo Alerts is a Docker-based system that comprises multiple containers. The main container uses a series of *plugins* to collect *hardware alarms* and *software alerts* from Qumulo Core clusters.

In Qumulo Alerts, *producers* are Docker containers that take data from various sources, pass it through *the Exchange*, a processing queue, and finally give it to *consumers*, defined users or user groups. Aside from processing data, the Exchange facilitates the transfers between the producers and consumers.



Both producers and consumers use plugins that help process alarms and alerts from a Qumulo Core cluster. A *plugin* is a mechanism that processes a single function, such as fan failure, disk failure, or node failure. Plugins help with granular control over the information that Qumulo Alerts collects and processes.

## Known Limitations of Qumulo Alerts

This section lists the currenly known limitations for Qumulo Alerts.

- **Floating IP Addresses or Network Load Balancing (NLB):** To prevent overloading any node in a Qumulo cluster, Qumulo Alerts plugins connect to all nodes in the cluster by using floating IP addresses or an NLB.

  > ⚠ Important
  > Qumulo Alerts can't function if neither IP addresses or NLBs are configured.

- **Error Logging:** Qumulo Alerts generates a large number of error messages that can help you debug issues. However, currently, all logging remains within the Docker container and is therefore not accessible easily. For help with debugging issues, contact Qumulo Care .
- **Connectivity to the Cluster:** If Qumulo Alerts loses connectivity to your Qumulo cluster for more than 10 minutes, you can start a new Qumulo Alerts session or contact Qumulo Care for help with debugging your running session. This scenario might occur when there is a VPN that requires re-authentication.

  > ⚠ Important
  > When you terminate your Qumulo Alerts session and start a new one, all logging data from the session running in the Docker container is discarded.

- **JSON-Based Configuration:** Currently, to configure Qumulo Alerts, an administrator must edit multiple JSON files.

# What Alarms and Alerts Qumulo Alerts Supports

This section lists the alarms and alerts that Qumulo Alerts collects and processes.

## Alarms

The following alarms report hardware changes in a Qumulo cluster.

| Plugin Name | Description |
|---|---|
| CPU | Temperature deviation |
| Disks | Failure, state change |
| Fans | Speed deviation, failure |
| Network | Link failure |
| Nodes | Addition, failure |

## Alerts

The following alerts report software changes and changes in environmental conditions for a Qumulo cluster.

| Plugin Name | Description |
|---|---|
| AD | Joining or leaving an Active Directory domain |
| Audit | Auditing enabled or disabled |
| Capacity | Change in cluster capacity (configured percentage of the entire cluster) |
| Exports | NFS exports created, modified, or deleted |
| FTP | FTP enabled or disabled |
| Groups | Local groups added, modified, or deleted |
| Monitoring | Cloud-based monitoring enabled, disabled, or unreachable |
| OSUpgrade | Qumulo Core upgrade |
| Quotas | Quota notification (configured percentage for specified directories) |

| Plugin Name | Description |
| --- | --- |
| Restriper | Restriper started, stopped, or percentage complete |
| Shares | SMB shares added, modified, or deleted |
| Users | Local users added, modified, or deleted |

# What Language Locales Qumulo Alerts Supports

This section lists the language locales that Qumulo Alerts supports for notifying users through email and SMS.

## Language Locales

The consumer processes for email and ClickSend integrations translate messages into the recipient's native language.

| Code | Description |
|---|---|
| `de_AT` | German (Austria) |
| `de_CH` | German (Switzerland) |
| `de_DE` | German (Germany) |
| `en_GB` | English (Great Britain) |
| `en_US` | English (USA) |
| `es_ES` | Spanish (Spain) |
| `fr_BE` | French (Belgium) |
| `fr_CA` | French (Canada) |
| `fr_CH` | French (Switzerland) |
| `fr_FR` | French (France) |
| `hu_HU` | Hungarian (Hungary) |
| `it_CH` | Italian (Switzerland) |
| `it_IT` | Italian (Italy) |
| `ja_JP` | Japanese (Japan) |
| `ko_KR` | Korean (Korea) |
| `pl_PL` | Polish (Poland) |
| `sk_SK` | Slovak (Slovakia) |

| Code | Description |
| --- | --- |
| `tr_TR` | Turkish (Turkey) |
| `zh_TW` | Traditional Chinese (Taiwan) |

## Converting Time Zones

Each message that the Exchange processes contains a timestamp encoded in UTC time by default. This timestamp must match the recipient's time zone. If you don't define the `default_timezone` in your `*Server.json` files, Qumulo Alerts uses `UTC`.

Each translated message that a user receives includes a time zone in the `Continent/City` format (for example, `America/Phoenix`). For more information, see List of TZ Database Time Zones.

> **Note**
> `UTC` doesn't follow the `Continent/City` format.

# Installing and Configuring Qumulo Alerts

## Installing Qumulo Alerts and Connecting it to a Qumulo Cluster

This section explains how to install Qumulo Alerts and connect it to a Qumulo cluster.

### Prerequisites

We recommend the following system requirements for Qumulo Alerts.

- 4-core processor
- 16 GB memory
- 2 TB disk space

Before you install Qumulo Alerts, make sure you have the following tools:

- Git (You can also browse the QumuloAlerts GitHub repository.)
- Docker
- Docker Compose Plugin

> ⚠ Important
> Qumulo Alerts requires the Docker Compose Plugin to operate correctly.

Before you connect Qumulo Alerts to a Qumulo cluster, collect the information that can help you configure Qumulo Alerts to monitor your cluster.

- **Cluster Address:** What is your cluster's address? Use a fully qualified domain name (FQDN) rather than an IP address.
- **Traffic Distribution:** Will your Qumulo Alerts installation use a network load balancer or a floating IP address?
- **Default Plugin Frequency**: What should be the default frequency for plugin execution? (You can specify the frequency in seconds or minutes.)
- **Alarm and Alert Types:** Decide which alarms and alerts Qumulo Alerts will collect for your cluster.

### Step 1: Clone the QumuloAlerts Repository

Navigate to the directory where you want Git to download files and run the following command.

```
git clone https://github.com/Qumulo/QumuloAlerts.git
```

Git creates a directory called `QumuloAlerts` and places the necessary files in it.

## Step 2: Create a Local User for Qumulo Alerts

To be able to generate access tokens, you must create a local user for Qumulo Alerts.

1. Use SSH to log in to any node in your cluster.
2. To create a local user, use the `auth_add_user` command and specify a name and password.

   ```
   qq auth_add_user --name QumuloAlerts --password <password>
   ```

3. You will need the user ID that appears in the command output to create a role for Qumulo Alerts.

   In the following example, the user ID is `1234`.

   ```
   {
     "can_change_password": true,
     "home_directory": null,
     "id": "1234",
     "name": "QumuloAlerts",
     "primary_group": "999",
     "sid": "S-1-5-21-1234567890-098765432-1234567890-1234",
     "uid": ""
   }
   ```

## Step 3: Create a Role for Qumulo Alerts

1. Log in to the Qumulo Web UI and then click **Cluster** > **Role Management**.
2. On the **Role Management** page, click **Create Role**.
3. On the **Create Role** page:

   a. Enter a name, for example `QumuloAlerts`.

   b. Enter a description, for example `This account lets an administrator restrict the privileges of the QumuloAlerts user.`
4. For **Privileges**, click all of the following:

   - AD_READ: Read Qumulo Active Directory Settings

- ANALYTICS_READ: Read cluster analytics
- AUDIT_READ: Read audit settings
- CHECKSUMMING_READ: View the status of checksumming
- CLUSTER_READ: View nodes, disks, protection status, and SSL certificate
- DNS_READ: Read DNS setting
- ENCRYPTION_READ: View the status of at rest encryption
- FTP_READ: View FTP status and settings
- IDENTITY_MAPPING_READ: Get AD/LDAP User Defined Mappings
- LDAP_READ: View LDAP settings
- LOCAL_GROUP_READ: View local groups and members
- LOCAL_USER_READ: Get information about local users
- METRICS_READ: Get all metrics
- NETWORK_READ: Read network status and settings
- NFS_EXPORT_READ: Read network status and settings
- QUOTA_READ: View all file system quotas
- REBOOT_READ: View Reboot Status
- RECONCILER_READ: View reconciler status and metrics
- REPLICATION_SOURCE_READ: View source relationship settings and status
- REPLICATION_TARGET_READ: View target relationship settings and status
- ROLE_READ: View roles and assignments
- S3_BUCKETS_READ: View all S3 buckets present in the system
- S3_CREDENTIALS_READ: View any S3 access key present in the system
- S3_SETTINGS_READ: View S3 server settings
- SAML_SETTINGS_READ: View SAML integration settings
- SMB_SESSION_READ: List logged on SMB sessions
- SMB_SHARE_READ: View configuration of SMB shares and SMB server settings
- SNAPSHOT_CALCULATE_USED_CAPACITY_READ: Recalculate capacity usage of snapshots
- SNAPSHOT_DIFFERENCE_READ: View the changes between snapshots

- SNAPSHOT_POLICY_READ: View snapshot policies and status
- SNAPSHOT_READ: List snapshots and view their status and cached capacity.
- SUPPORT_READ: View support configuration and status
- TENANT_READ: View any tenant information
- TIME_READ: View time and time settings
- UNCONFIGURED_NODE_READ: List unconfigured Qumulo nodes
- UPGRADE_READ: View upgrade configuration and status

5. Click **Save**.

## Step 4: Assign the Qumulo Alerts Role to Your Local User

1. In the Web UI, click **Cluster** > **Role Management**.
2. On the **Role Management** page, in the **QumuloAlerts** section, click **Add Member**.
3. In the **Add Member to Administrators** dialog box, for **Trustee**, enter the local username you have created earlier (for example, `QumuloAlerts`) and then click **Yes, Add Member**.

## Step 5: Create a Long-Lived Access Token

Use the `auth_create_access_token` command and specify the ID of the local user. For example:

```
qq auth_create_access_token auth_id:1234
```

The `auth_create_access_token` command returns a JSON response that contains the bearer token body and the access token ID, which you can use to manage the access token.

```
{
  "bearer_token": "access-v1:abAcde...==",
  "id": "12345678901234567890123"
}
```

> ⚠ Important
> As soon as you receive your bearer token, record it in a safe place. If you misplace the bearer token, you can't retrieve it at a later time. You must create a new access token.

For more information, see Using Qumulo Core Access Tokens in the Qumulo Administrator Guide.

## Step 6: Configure Qumulo Alerts

1. Configure alarms and alerts (page 13).
2. Configure user notifications (page 16).
3. Configure integration with an email server (page 18) or integration with the ClickSend service (page 21).

## Step 7: Start Qumulo Alerts

To start Qumulo Alerts, run the following command from the directory to which you cloned the `QumuloAlerts` repository.

```
./start-docker-qumulo-alerts.sh
```

To reapply changes to the Qumulo Alerts configuration, run the following command.

```
./stop-docker-qumulo-alerts.sh && ./start-docker-qumulo-alerts.sh
```

# Configuring Alarms and Alerts

This section explains how to configure Qumulo Alerts to generate alarms and alerts.

## Configuring Monitoring for a Qumulo Cluster

To configure Qumulo Alerts to connect to, and collect data from a Qumulo cluster, you must edit `QumuloAlerts.json`, located in the `config/alerts` directory, in the directory that you cloned from GitHub (page 0).

The following is an explanation of the JSON keys that configure Qumulo cluster monitoring.

| Name | Description |
|---|---|
| `cluster_name` | The fully qualified domain name (FQDN) for your Qumulo cluster |
| `cluster_port` | The port for communicating with the Qumulo REST API, `8000` by default |
| `frequency` | The time period between each plugin's execution, including the unit (seconds or minutes) and value, that applies to all plugins in a category without a specific frequency, 60 seconds by default<br><br>You can configure a distinct frequency for each category of alarm or alert. |
| `monitor` | An array of notifications to which a user subscribes<br><br>• `category` can be `Alarms` or `Alerts`<br>• `subcategory` is an array of plugin names<br><br>For a list of available alarm and alert plugin names, see What Alarms and Alerts Qumulo Alerts Supports (page 4). |

| Name | Description |
| --- | --- |
| `nlb` | Network load balancer: when set to `false`, floating IP address configuration is required<br><br>To prevent spreading the load of a plugin's API requests across all nodes in a Qumulo cluster, each alarm or alert plugin that you configure communicates with your cluster by using either a network load balancer or floating IPs.<br><br>**⚠ Important**<br>You can configure *one*—but not both—of these communication methods. |
| `user_token` | The long-lived access token for communicating with the Qumulo REST API |

## Example: Monitoring Alarms for Disks and Nodes on a Single Cluster

```
[{
  "cluster_name": "test-cluster.corp.example.com",
  "cluster_port": 8000,
  "user_token": "access-v1:abcdefgh1234567...",
  "nlb": false,
  "frequency": {"seconds": 60},
  "monitor": [{
    "category": "Alarms",
    "subcategory": ["Disks", "Nodes"],
    "enabled": true
  }]
}]
```

## Example: Monitoring Alarms for Disks and Nodes and All Alerts on a Single Cluster

In the following example, the wildcard `*` specifies *every plugin available in this category*.

```
[{
  "cluster_name": "test_cluster.corp.example.com",
  "cluster_port": 8000,
  "user_token": "access-v1:abcdefgh1234567...",
  "nlb": false,
  "frequency": {"seconds": 60},
  "monitor": [{
    "category": "Alarms",
    "subcategory": ["Disks", "Nodes"],
    "enabled": true
  },{
    "category": "Alerts",
    "subcategory": ["*"],
    "frequency": {"minutes": 5},
    "enabled": true
  }]
}]
```

# Configuring User Notifications

This section explains how to configure user notifications from Qumulo Alerts.

## Configuring User Notifications about a Qumulo Cluster

To configure Qumulo Alerts to notify users about alarms and alerts from a Qumulo cluster, you must edit `QumuloUsers.json`, located in the `config/consumer` directory, in the directory that you cloned from GitHub (page 0).

The following is an explanation of the JSON keys that configure user notifications.

> **Note**
> Although both `email_address` and `phone_number` are optional keys, you must specify at least one of them for the user to receive notifications.

| Name | Description |
|---|---|
| `email_address` | (Optional) The recipient's email address |
| `full_name` | The recipient's full name |
| `language` | The recipient's language locale<br><br>The consumer processes for email and ClickSend integrations translate messages into the recipient's native language. For more information, see What Language Locales Qumulo Alerts Supports (page 6). |
| `notify` | An array of notifications to which a user subscribes<br><br>• `category` can be `Alarms` or `Alerts`<br>• `subcategory` is an array of plugin names<br><br>For a list of available alarm and alert plugin names, see What Alarms and Alerts Qumulo Alerts Supports (page 4). |
| `phone_number` | (Optional) The recipient's phone number, starting with a plus ( `+` ) followed by the international calling code |
| `short_name` | The shortened form of the recipient's name |

| Name | Description |
| --- | --- |
| `timezone` | The recipient's timezone<br><br>For more information, see Converting Time Zones (page 7). |

## Example: Notifying Users in Different Countries about Different Alarms and Alerts

In the following example, the wildcard `*` specifies *all plugins in this category generate notifications*.

```
[{
  "full_name": "Linda Johnson",
  "short_name": "Linda",
  "email_address": "ljohnson@example.com",
  "phone_number": "+15555555555",
  "language": "en_US",
  "timezone": "America/Phoenix",
  "notify": [{
    "category": "Alarms",
    "subcategory": ["Disk", "Node"],
    "enabled": true
  }]
},{
  "full_name": "Yennefer Martinez",
  "short_name": "Yen",
  "email_address": "yennefer@example.com",
  "phone_number": "+905555555555"
  "language": "tr_TR",
  "timezone": "Europe/Istanbul",
  "notify": [{
    "category": "Alarms",
    "subcategory": ["*"],
    "enabled": true
  },{
    "category": "Alerts",
    "subcategory": ["Quotas"],
    "enabled": true
  }]
}]
```

# Configuring Qumulo Alerts Integration with an Email Server

This section explains how to configure Qumulo Alerts to work with an email server.

## Configuring Qumulo Alerts Integration with an Email Server

To configure Qumulo Alerts to integrate with an email server, you must edit `QumuloEmailServer.json`, located in the `config/consumers` directory, in the directory that you cloned from GitHub (page 0).

The following is an explanation of the JSON keys that configure integration with an email server.

> **Note**
> The `login`, `password`, and `use` keys are optional depending on the type of SMTP email server that you use.

| Name | Description |
|---|---|
| `default_language` | The recipient's language locale<br><br>The consumer processes for email and ClickSend integrations translate messages into the recipient's native language. For more information, see What Language Locales Qumulo Alerts Supports (page 6). |
| `default_timezone` | The recipient's timezone<br><br>For more information, see Converting Time Zones (page 7). |
| `from_addr` | The sender's email address |
| `login` | (Optional) The username for the SMTP email server |
| `password` | (Optional) The password for the SMTP email server |
| `port` | The port for the SMTP email server, commonly `587` for SSL or TLS |
| `server` | The fully qualified domain name (FQDN) or IP address of the SMTP email server |

| Name | Description |
| --- | --- |
| `to_addr` | The recipient's email address<br><br>⚠ Important<br>This field is only for testing the email server. Normally, Qumulo Alerts uses the values from `QumuloUsers.json`. |
| `use` | (Optional) The security protocol to use: `ssl` or `tls` |

### Using Gmail as an SMTP Relay

After May 2022, only organizations with access to the Google Admin Console can use SMTP relay. If your organization has this access, see Route outgoing SMTP relay messages through Google .

## Example: Configured Email Server Integration

```
{
  "from_addr": "robert@example.com",
  "to_addr": "test@example.com",
  "login": "robert@xyzcorp.com",
  "password": "<password>",
  "server": "smtp.xyzcorp.com",
  "port": 587,
  "use": "tls",
  "default_language": "en_GB",
  "default_timezone": "UTC"
}
```

## To Test Integration with Your Email Server

1. To add execution permissions to the `test_email.macos-latest` and `test_email.ubuntu-latest` files, run the following command.

   ```
   chmod a+x test_email.*
   ```

2. To test the integration, run the executable for your operating system. For example:

```
./test_email.ubuntu-latest --config ./config/consumer/
```

The email server sends a test message to the email address specified in the `to_addr` key.

# Configuring Qumulo Alerts Integration with ClickSend

This section explains how to configure Qumulo Alerts to work with the ClickSend service.

ClickSend is a paid, third-party service that provides delivery of messages as SMS (and other formats). For more information, see How to get started with ClickSend in the ClickSend documentation.

> **Note**
> To be able to send SMS in the U.S. and Canada, you must sign up for a dedicated TFN.

## Configuring Qumulo Alerts Integration with the ClickSend Service

To configure Qumulo Alerts to integrate with the ClickSend Service, you must edit `QumuloClickSendServer.json`, located in the `config/consumers` directory, in the directory that you cloned from GitHub (page 0).

The following is an explanation of the JSON keys that configure integration with an email server.

| Name | Description |
|---|---|
| `default_language` | The recipient's language locale<br><br>The consumer processes for email and ClickSend integrations translate messages into the recipient's native language. For more information, see What Language Locales Qumulo Alerts Supports (page 6). |
| `default_timezone` | The recipient's timezone<br><br>For more information, see Converting Time Zones (page 7). |
| `senderid` | Your ClickSend toll-free number (TFN)<br><br>**Important**<br>This field is mandatory for the U.S. and Canada.<br><br>For more information, see TFN Application approval process in the ClickSend documentation. |

| Name | Description |
| --- | --- |
| `token` | Your ClickSend API key<br><br>For more information, see API credentials in the ClickSend documentation. |

## Example: Configured ClickSend Service Integration

```
{
    "username": "mary@example.com",
    "token": "A12BC345-D6EF-7890-G1234-56HIJ7890",
    "senderid": "+15555555555",
    "default_language": "en_GB",
    "default_timezone": "UTC"
}
```