Qumulo Alerts Administration Guide

Version 3.3.0

December 01, 2022

Table of Contents

Getting Started with Qumulo Alerts

How Qumulo Alerts Works with Qumulo Core 2

What Alarms and Alerts Qumulo Alerts Supports 4

Installing Qumulo Alerts 6

Quick-Start Guides for Qumulo Alerts

Configuring Alarms and Alerts 8

Configuring User Notifications 9

Configuring Integration with an Email Server 10

Configuring Integration with ClickSend 11

Detailed Configuration for Qumulo Alerts

Preparing to Work with Qumulo Alerts 12

Configuring Alarms and Alerts 16

Configuring User Notifications 22

Configuring Native Language Support 26

Configuring Integration with an Email Server 28

Configuring Integration with ClickSend 32

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.

Qumulo Alerts will collect Hardware Alarms, Software Alerts, and other Information from one or more Qumulo Clusters.

This collected information is processed by AI techniques in order to generate messages to selected users or groups of users.

Architecture

Qumulo Alerts is a docker based infrastructure composed of multiple containers.

The main container, called Qumulo Alerts, uses a series of plugins to collect the information from the cluster.

That information is passed to an Exchange (queueing mechanism) and then onto multiple containers for further processing; including sending messages to selected users in their native language.

Architecture Diagram

How Producers, Consumers, their Plugins and the Exchange Work in Qumulo Alerts

Producers are docker containers that take data from various sources and passes that data to a consumer through a queuing Exchange for additional processing.

Consumers are docker containers that take data passed by the Qumulo Alerts producer through the queuing Exchange.

The Exchange is a docker container that facilitates the handling of the messages generated by Qumulo Alerts producer.

The reader should think of the exchange like you would an old fashioned telephone operator. In the past, before dial or mobile telephones, when somebody wanted to make a telephone call, you would tell the operator who you wanted to speak with. The operator would setup a connection between the two parties; the caller and “the called” (or the Producer and the Consumer).

Our exchange is a docker container that takes messages from Qumulo Alerts and passes them to other docker containers that will consume them. We haven’t talked about the consumers yet, but clearly there must be a method to notify any given group of users that something has happened on their cluster.

Both the Qumulo Alerts producers and consumers use a plugin model to gather or process alarms and alerts from a Qumulo cluster.

Plugins collect and process data for one and only one function, such as disk, node, or fan failures. This allows for finer granular control of which information is collected from the cluster and, subsequently, reported on to users or groups.

Known Limitations for Qumulo Alerts

This section lists the known limitations with the current release of Qumulo Alerts.

- **Floating IP Addresses or Network Load Balancing (NLB):** To prevent overloading any node in a 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. 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:** It might be necessary to restart Qumulo Alerts if it loses connectivity to your cluster. This scenario might occur when there is a VPN that requires occasional re-authentication between Qumulo Alerts and the cluster.

  > ⚠ Important
  > If Qumulo Alerts is disconnected from your cluster for more than XX minutes, you must start a new Qumulo Alerts session.

- **JSON-Only 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, alerts, and informational data that Qumulo Alerts collects and processes.

Alarms

The following alarms are for reported hardware changes for a Qumulo cluster.

Plugin Name	Features
CPU	Temperature Deviation
Disks	Failure, state change
Fans	Speed deviation, failure
Nodes	Addition, failure

Alerts

The following alerts are for reported software changes or 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)
Restriper	Restriper started, stopped, or percentage complete

Plugin Name	Description
Shares	SMB shares added, modified, or deleted
Users	Local users added, modified, or deleted

Informational Data

Informational data doesn't require immediate user notification. Primarily, AI techniques use informational data to determine trends or potential hardware or software failure.

Currently, Qumulo Alerts doesn't collect any informational data.

Installing Qumulo Alerts

This section explains how to install Qumulo Alerts.

Prerequisites

Before you install Qumulo Alerts, you must install Docker. For more information, see the following Docker documentation.

- Linux or macOS: Docker Engine installation overview
- Windows: Installing Docker on Windows

Prerequisites

- Docker Desktop or Enterprise
- 4 Core Processor
- 16GB Memory
- 2TB Free Disk Space

Getting Started

Before you can configure and get this software to work on your system, you will need to clone this repository on your machine.

For that, you will need to have `git` installed on your machine.

Installing Qumulo Alerts

Once git is operational, then find or create a directory where you wish to store the contents of the `Qumulo Alerts` repository and clone it to your machine with the command qumulo-alerts-guide/getting-started-qumulo-alerts/installing.md

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

You will notice that the `git clone` command will create a new directory in your current location called `Qumulo Alerts`.

The contents of that directory should look like:

```
drwxr-xr-x  10 someone  somegroup       320 Aug  4 13:09 Docs
-rw-r--r--   1 someone  somegroup      1063 Aug  2 13:23 LICENSE
-rw-r--r--   1 someone  somegroup      5170 Aug  4 12:05 README.md
drwxr-xr-x   4 someone  somegroup       128 Aug  3 17:43 config
-rw-r--r--   1 someone  somegroup       994 Aug  4 13:09 docker-compose.yml
-rwxr-xr-x   1 someone  somegroup       515 Aug  4 13:09 start-docker-qumuloalerts.s
h
-rwxr-xr-x   1 someone  somegroup       301 Aug  2 13:23 stop-docker-qumuloalerts.sh
-rw-r--r--   1 someone  somegroup   7043872 Aug  4 13:09 test_email.macos-latest
-rw-r--r--   1 someone  somegroup  14714824 Aug  2 13:23 test_email.ubuntu-latest
```

Of course, the owner will not be `someone` and the group will not be `somegroup`. The owner will be you and the group will show whatever group you currently belong to. If in doubt, simply type `id -gn` to see your current group and `id -un` to see your current login id.

You will need to modify several configuration files and one shell script to get this software to run in your environment. That information will be covered in the **Configuration** section.

Additionally, git doesn't preserve execution permissions, so you will need to add the execution permissions to the files `test_email.macos-latest` and `test_email.ubuntu-latest`.

You can accomplish this by running the command

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

Quick-Start Guides for Qumulo Alerts

Quick-Start Guide to Configuring Alarms and Alerts

This section is a quick-start guide to configuring Qumulo Alerts to generate alarms and alerts.

Summary list of configuration steps

The following is a summary list of the steps necessary to successfully configure Qumulo Alerts.

Please read each individual configuration step for more detailed instructions.

Qumulo Alerts configuration

The following actions must be completed on each of the Qumulo clusters to be monitored.

1. Create a local user for Qumulo Alerts on the Qumulo cluster
2. Create a role for Qumulo Alerts on the Qumulo cluster
    - Read Required Role Privileges (page 0) for more information
3. Add the created local user as a member to the newly created Qumulo Alerts role
4. Create a long-lived token for the created local user

The next series of actions must be completed on the machine that will be running Qumulo Alerts.

1. Map out which Alarms and Alerts will be monitored
    - Check the Supported Alarms and Alerts (page 0)
2. Edit Qumulo Alerts.json
    - Add an entry for each monitored cluster
    - Add the plugin names under each **Monitor Category**
3. If monitoring **Capacity**, edit Capacity.json
4. If monitoring **Fans**, edit Fans.json
5. If monitoring **Quotas**, edit Quotas.json

Quick-Start Guide to Configuring User Notifications from Qumulo Alerts

This section is a quick-start guide to configuring user notifications from Qumulo Alerts.

Summary list of configuration steps

The following is a summary list of the steps necessary to successfully prepare and configure users to receive notifications from Qumulo Alerts.

Please read each individual configuration step for more detailed instructions.

User notification configuration

Messages are sent through Email or ClickSend to specified users in their desired language and timezone.

The next series of actions must be completed on the machine that will be running Qumulo Alerts.

1. Map out which Alarms and Alerts will be notified for a given user
    - Check the Supported Alarms and Alerts (page 0)
2. Edit QumuloUsers.json
    - Add an entry for each individual user to be notified
    - Add the plugin names under each **Notify Subcategory**

Quick-Start Guide to Configuring Qumulo Alerts Integration with an Email Server

This section is a quick-start guide to configuring Qumulo Alerts to work with an email server.

Summary list of configuration steps

The following is a summary list of the steps necessary to successfully configure the email system for Qumulo Alerts.

Please read each individual configuration step for more detailed instructions.

Email configuration

If you wish to use Email to send an alarm / alert to an individual user or a group of users, then configuring the email server is required. The following steps will assist you in making sure that your email server can successfully send Qumulo Alerts alarms and alerts.

1. Edit the QumuloEmailServer.json file
2. Enter a “from” address. This must be a valid email address.
3. Enter a “to” address. This address is ONLY used for testing the email system.
4. The following fields are optional based upon your email server. Some email servers require them and others do not. If in doubt, please talk to your email administrator.
    - Login, Password, Port, and Use
5. Enter the default language to use when sending an email. If you don’t configure anything, the default language will be “en_GB”.
6. Enter the default timezone to use when sending an email. If you don’t configure anything, the default timezone will be “UTC”.
7. Run one of the test programs (test_email.macos-latest or test_email.ubuntu-latest) to verify operation of your configured email server. You should receive a test email if everything is working.

Quick-Start Guide to Configuring Qumulo Alerts Integration with ClickSend

This section is a quick-start guide to configuring Qumulo Alerts to work with the ClickSend service.

Summary list of configuration steps

The following is a summary list of the steps necessary to successfully configure the ClickSend system for Qumulo Alerts.

Please read each individual configuration step for more detailed instructions.

ClickSend configuration

> ⚠ Important
> ClickSend is a paid service that provides delivery of SMS messages. If you wish to deliver SMS messages with Qumulo Alerts, then you are required to sign up for ClickSend before continuing with configuration.

If you wish to use ClickSend to send an alarm / alert to SMS, then configuring this service is required. The following steps will assist you in making sure that ClickSend is configured and working properly.

1. Visit http://clicksend.com and register for an account
2. On the ClickSend website, go to Developers->API Credentials and generate an API key
3. Edit QumuloClickSendServer.json
    - Add username (from the API Credentials)
    - Add token (the API token from the API Credentials)
    - If you wish to send messages within the USA or Canada, then you should sign up for a TFN (Toll Free Number). If you have a TFN, then add it to the configuration file under **senderid**.
    - Optionally, you can add **default_language** and **default_timezone**.

Detailed Configuration for Qumulo Alerts

Preparing to Work with Qumulo Alerts

This section explains how to prepare for working with Qumulo Alerts, including identifying cluster to monitor and the alarms and alerts to configure, creating a local user (required for creating a long-lived access token for API access to your cluster), creating a role with privileges for each monitored cluster, associating the local user with the role, and reating a long-lived token.

Step 1: Identifying Clusters for Qumulo Alerts to Monitor

Whether it is one or multiple clusters, Qumulo Alerts can monitor for Alarms, Alerts, and other Information.

The following information is required to successfully configure Qumulo Alerts to monitor the cluster(s).

1. Cluster address
    - You should use a fully qualified DNS name rather than an IP address.
2. Will Qumulo Alerts use a Network Load Balancer or Floating IP addresses?
3. What is the default frequency for plugins to execute?
    - Frequency can be in seconds or minutes.
4. Which Alarms, Alerts, and Information should be collected for each cluster?

Step 2: Creating a Local User for Qumulo Alerts

This section demonstrates how to create a local user for use with Qumulo Alerts.

Prior to creating a long-lived token, you must first have a local user account to generate the token.

After you login to any node of the cluster, create a user that will be used to create the long-lived token.

```
qq auth_add_user --name Qumulo Alerts --password SOME_PASSWORD
```

The password is required to create the local user and is assigned to that user. If you don’t specify the password, then the command will request and read it from the terminal.

The output of that command will resemble:

```
{
    "can_change_password": true,
    "home_directory": null,
    "id": "1009",
    "name": "Qumulo Alerts",
    "primary_group": "513",
    "sid": "S-1-5-21-2211849900-244887647-2627124441-1009",
    "uid": ""
}
```


Step 3: To Create a Role for Qumulo Alerts and Assign it to a Local User

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, do the following.

   a. Enter a name, for example `Qumulo Alerts`.

   b. Enter a description, for example `This account lets an administrator restrict the privileges of the Qumulo Alerts user.`
4. For **Privileges**, click each of the following:
   - AD_READ: Read Qumulo Active Directory Settings
   - ANALYTICS_READ: Read cluster analytics
   - 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**.
6. Click **Cluster** > **Role Management**.
7. On the **Role Management** page, in the **Qumulo Alerts** section, click **Add Member**.
8. In the **Add Member to Administrators** dialog box, for **Trustee**, enter the local username you have created earlier and then click **Yes, Add Member**.

Step 4: To 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:1009
```

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.

Next Steps

After you finish preparing for working with Qumulo Alerts, you can begin to configuring it for your environment.

ℹ Note
Although configuring Qumulo Alerts to work with an email server or the ClickSend service is optional, you must configure at least one of these means for your users to receive an alarm or alert.

1. To monitor each of your clusters, configure Qumulo Alerts (page 0).
2. To send messages to users when an alarm or alert triggers, configure user notifications (page 0).
3. (Optional) To send email notifications, configure Qumulo Alerts to work with an email server (page 0).
4. (Optional) To send SMS notifications, configure Qumulo Alerts to work with the ClickSend service (page 0).

Configuring Alarms and Alerts

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

Directory Structure

You should be in the `config` directory and looking at the following subdirectories:

```
drwxr-xr-x  12 someone  somegroup  384 Aug  4 13:09 consumer
drwxr-xr-x   7 someone  somegroup  224 Aug  4 13:09 alerts
```

Let's start by editing the files in the `alerts` subdirectory.

`cd alerts`

You should see:

```
-rw-r--r--  1 someone  somegroup   909 Nov  3 20:10 Qumulo Alerts.json
-rw-r--r--  1 someone  somegroup   209 Oct 14 09:41 QumuloExchange.json
drwxr-xr-x  4 someone  somegroup   128 Nov  2 11:27 schema
```

Qumulo Alerts json file

If this file is not configured or configured improperly, then the collection of alarms and alerts from a cluster will fail.

Once you have it open in your favorite text based editor, you should see the following:

```
[
    {
        "cluster_name": "cluster_1.qumulo.com",
        "cluster_port": 8000,
        "user_token": "TOKEN",
        "nlb": false,
        "frequency":
        {
            "seconds": 60
        },
        "monitor":
        [
            {
                "category": "Alarms",
                "subcategory": ["Disk", "Node"],
                "enabled": true
            },
            {
                "category": "Alerts",
                "subcategory": ["Quotas"],
                "enabled": true,
                "frequency":
                {
                    "minutes": 5
                }
            },
            {
                "category": "Alerts",
                "subcategory": ["Capacity"],
                "enabled": true
            },
            {
                "category": "Informational",
                "subcategory": ["*"],
                "enabled": false
            }
        ]
    },
    {
        "cluster_name": "cluster_2.qumulo.com",
        "cluster_port": 8000,
        "user_token": "TOKEN",
        "nlb": false,
        "frequency":
        {
            "minutes": 2
```

```
        },
        "monitor":
        [
            {
                "category": "Alarms",
                "subcategory": ["*"],
                "enabled": false
            }
        ]
    }
]
```

- `cluster_name` - Valid DNS name of a cluster used to generate alarms / alerts
- `cluster_port` - The port needed to communicate with the API. Default is 8000.
- `user_token` - The long-lived token needed to the communicate with the API.
- `nlb` - Network Load Balancer
- `frequency` - Default frequency to execute each plugin
- `monitor` - An array of Alarms, Alerts, and Informational plugins to execute

A single cluster example

The following is an example configuration to monitor only alarms.

```
[
    {
        "cluster_name": "test_cluster.corp.garagecorp.com",
        "cluster_port": 8000,
        "user_token": "access-v1:D3d4qFEPA5eXV0jYdDFvmRKwKYu/BhKHVpzShY5nEjwBAAAA6AM
AAAAAAACUyeFNvYTHnyIlYGMAAAAAq9jvHw==",
        "nlb": false,
        "frequency":
        {
            "seconds": 60
        },
        "monitor":
        [
            {
                "category": "Alarms",
                "subcategory": ["Disks", "Nodes"],
                "enabled": true
            }
        ]
    }
]
```

Using the above example, let us highlight a few of the configuration features.

1. **cluster_name** - Fully qualified DNS name of test_cluster.corp.garagecorp.com
2. **user_token** - Token created after executing the instructions in the Create Long-Lived Token (page 0) section.
3. **nlb** - Set to **false**. This means that Floating IP's (page 21) are required.
4. **frequency** - This is the default time period used to execute each plugin. The example is 60 seconds.
5. **monitor** - An array of **Alarms**, **Alerts**, and/or **Informational** that specify the plugins to execute at each frequency interval.

Each entry in the **monitor** array specifies a category of **Alarms**, **Alerts**, or **Informational**.

Monitor example

Using the above example, let us highlight several of the features needed to configure plugins.

1. **category** - This must be one of **Alarms**, **Alerts**, or **Informational**. Our example has specified **Alarms**."
2. **subcategory** - An array of plugin names separated by a comma. In our example, the plugins for **Disks** and **Nodes** are configured.

3. **enabled** - The plugins are **enabled** by specifying **true**

Another single cluster example

The following is another example configuration to monitor for alarms and alerts.

```
[
    {
        "cluster_name": "test_cluster.corp.garagecorp.com",
        "cluster_port": 8000,
        "user_token": "access-v1:D3d4qFEPA5eXV0jYdDFvmRKwKYu/BhKHVpzShY5nEjwBAAAA6AM
AAAAAAACUyeFNvYTHnyIlYGMAAAAAq9jvHw==",
        "nlb": false,
        "frequency":
        {
            "seconds": 60
        },
        "monitor":
        [
            {
                "category": "Alarms",
                "subcategory": ["Disks", "Nodes"],
                "enabled": true
            },
            {
                "category": "Alerts",
                "subcategory": ["*"],
                "frequency":
                {
                    "minutes": 5
                },
                "enabled": true
            }
        ]
    }
]
```

The above is a duplicate of the previous example with the exception of adding a new monitor item for **Alerts**.

Additional Monitor Example

Using the above example, let us highlight several of the features needed to configure plugins for **Alarms**, **Alerts**, or **Informational**.

1. **category** - Our example for the second item in the Monitor array has specified **Alerts**."

2. **subcategory** - A “*” is specified; which represents every plugin available for this category.
3. **frequency** - A specific interval of 5 minutes for all of the **Alerts** plugins.
4. **enabled** - The plugins are **enabled** by specifying **true**

Load Balancing

Each plugin configured under **Alarms**, **Alerts**, or **Informational** communicates with the cluster using one of two load balancing techniques. This prevents spreading the load of the plugin API requests across all of the nodes of a Qumulo cluster.

The two techniques used to achieve load balancing are

1. Network Load Balancer
2. Floating IP’s

These two techniques are mutually exclusive and cannot be combined.

Network Load Balancer

A Network Load Balancer is an external device that will spread the load across multiple nodes within a Qumulo cluster.

The **cluster_name** is the DNS address used to communicate with the network load balancer.

Network Load Balancing is enabled by setting the **nlb** flag to **true** in the cluster confiuration.

> ⚠ Important
> The configuration of an external load balancer is beyond the scope of this manual.

Floating IP’s

In lieu of an external Network Load Balancer, Qumulo Alerts will use the Floating IP’s configured for the cluster being monitored.

If no floating IP’s are configured, then Qumulo Alerts will consider this an error and terminate.

Frequency

Frequency is the time period between the invocation of each plugin.

There are two possible frequencies. The **default** frequency is specified and required when configuring the cluster. The second and optional frequency is configured for each **category** of **Alarms**, **Alerts**, or **Informational**.

The **default** frequency applies to all plugins without a specific frequency in any given **category**.

The frequency is specified in either seconds or minutes. The default is 60 seconds.

Configuring User Notifications

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

Directory Structure

You should be in the `config` directory and looking at the following subdirectories:

```
drwxr-xr-x  12 someone  somegroup  384 Aug  4 13:09 consumer
drwxr-xr-x   7 someone  somegroup  224 Aug  4 13:09 alerts
```

Let's start by editing the files in the `consumer` subdirectory.

`cd consumer`

You should see:

```
-rw-r--r--  1 someone  somegroup   208 Aug  3 17:06 QumuloEmailServer.json
-rw-r--r--  1 someone  somegroup   226 Aug 10 11:13 QumuloExchangeEMail.json
-rw-r--r--  1 someone  somegroup   118 Aug  3 17:06 QumuloClickSend.json
-rw-r--r--  1 someone  somegroup  1050 Aug 10 11:13 QumuloUsers.json
```

Configuring Users for Alarms and Alerts

Using your favorite text based editor, open the **QumuloUsers.json** file.

That is the main file used to configure users and what alarms/alerts will generate messages.

Let's look at the contents of the file.

```
[
    {
        "full_name": "Joe Schmo",
        "short_name": "Joe",
        "email_address": "joe@anywhere.com",
        "phone_number": "+14801234567",
        "language": "en_US",
        "timezone": "America/Phoenix",
        "notify":
        [
            {
                "category": "Alarms",
                "subcategory": ["Disk", "Node"],
                "enabled": true
            }
        ]
    },
    {
        "full_name": "Amy Iknow",
        "short_name": "amy",
        "email_address": "amy@anywho.com",
        "phone_number": "+9041396742005"
        "language": "tr_TR",
        "timezone": "Europe/Istanbul",
        "notify":
        [
            {
                "category": "Alarms",
                "subcategory": ["*"],
                "enabled": true
            },
            {
                "category": "Alerts",
                "subcategory": ["Quotas"],
                "enabled": true
            },
            {
                "category": "Informational",
                "subcategory": ["*"],
                "enabled": false
            }
        ]
    }
]
```

The configuration file is an array of User entries. In the example above, there are two entries. Let's go through them in a little more detail.

1. `full_name` - Joe Schmo (Of course, this would be the name of whomever should receive the message)
2. `short_name` - Joe
3. `email_address` - joe@anywhere.com (This is an optional field. If it doesn't exist, then an email will not be sent for this user)
4. `phone_number` - +14801234567 (This is an optional field. If it doesn't exist, then an SMS will not be sent for this user)
5. `language` - en_US (English - USA)
6. `timezone` - America/Phoenix
7. `notify` - This is an array of notifications that Joe or Amy want to subscribe to. If the notification is not in the array, then any notification sent from Qumulo Alerts will not be sent to this user.

> ⚠ Important
> Phone numbers must always start with the international dialing standard of +COUNTRY_CODE. I.E.: +1 for the USA, +41 for Switzerland, +48 for Poland, etc.

Language Definitions

The Email or ClickSend consumer processes translate messages into your native language.

For information about currently supported Native Languages, see Natural Language Support (page 0).

Timezones

Timezones are used when messages are translated and output to the user.

For more information about Timezone support, see Timezones (page 0).

Notify and Plugins

The notify section is composed of one or plugins for actions on which the user would like to receive messages .

Let's look at the one item for Joe Schmo.

1. `category` - This must be one of **Alarms**, **Alerts**, or **Informational**
2. `subcategory` - This is an array of items that correspond to the plugins used in Qumulo Alerts. In the example above, Joe Schmo wants to subscribe to **Disks** and **Nodes**. This corresponds to Alarms of Disk and Node failures (or state changes).
3. `enabled` - In this example, it is **true**. If the entry is **false**, then Joe Schmo will not receive

any Alarms for Disks or Nodes.

> **Note**
>
> - using a "*" within the subcategory specifies that **ALL** plugins for this category will generate messages.

> **Note**
>
> - Do not forget the comma between any two users definied in the file. Notice between Joe Schmo and Amy Iknow that there is a comma between their entries.
> - The last user defined in the file has no comma.
> - The same is true of the **notify** array for Amy Iknow. In that array, there are three entries: **Alarms**, **Alerts**, and **informational**. There is a comma between **Alarms** and **Alerts** and between **Alerts** and **Informational**. There is no comma after the **Informational** entry.

Configuring Native Language Support

This section explains how to configure native language support for email and SMS messages.

Native Language Support

Qumulo Alerts supports formatting all user notification messages into native language.

Language Definitions

The currently supported languages and their codes are:

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

Language Code	Description
sk_SK	Slovakian Language - Slovakia
tr_TR	Turkish Language - Turkey
zh_TW	Traditional Chinese Language - Taiwan

Timezones

Timezones are used when messages are translated and output to the user.

Each message received from the alerts processed through the exchange has a timestamp on it. That timestamp is encoded with UTC time and will need to be changed to match the timezone of the recipient.

The software utilizes the standard TZ database format of **Continent/City**. Example: **America/Phoenix**.

For a detailed listing, look at the following article:

Database of TZ timezones

If you do not define a timezone for the **default_timezone** in one of the consumer configuration files, then it is assumed that **UTC** will be used. UTC, by the way, is the only timezone that does not follow the format of **Continent/City**.

Configuring Qumulo Alerts Integration with an Email Server

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

> **Note**
>
> - Email is one of three processes known as **consumers**.
> - Consumers accept messages from Qumulo Alerts through the Exchange.

Directory Structure

You should be in the `config` directory and looking at the following subdirectories:

```
drwxr-xr-x  12 someone  somegroup  384 Aug  4 13:09 consumer
drwxr-xr-x   7 someone  somegroup  224 Aug  4 13:09 alerts
```

Let's start by editing the files in the `consumer` subdirectory.

`cd consumer`

You should see:

```
-rw-r--r--  1 someone  somegroup   208 Aug  3 17:06 QumuloEmailServer.json
-rw-r--r--  1 someone  somegroup   226 Aug 10 11:13 QumuloExchangeEMail.json
-rw-r--r--  1 someone  somegroup   118 Aug  3 17:06 QumuloClickSend.json
-rw-r--r--  1 someone  somegroup  1050 Aug 10 11:13 QumuloUsers.json
```

QumuloEmailServer json file

If this file is not configured or configured improperly, then the email docker container will halt. It is entirely optional to configure email. If your organization wants to only use ClickSend (SMS), then you can skip this section.

If, however, you wish to send emails, then you need to configure the following entry properly. And, more importantly, you need to verify the successful operation of the email server through the email relay.

If you wish to configure email, let's get started by editing the file `QumuloEmailServer.json` using your favorite text based editor.

Once you have it open in the editor, you should see the following:

```
{
    "from_addr": "",
    "to_addr": "",
    "login": "",
    "password": "",
    "server": "smtp.gmail.com",
    "port": 465,
    "use": "ssl",
    "default_language": "en_GB",
    "default_timezone": "UTC"
}
```

- `from_addr` - FROM email address
- `to_addr` - TO email address **(used ONLY to test the email server)**
- `login` - Username needed to login to the email server
- `password` - Password needed to login to the email server
- `server` - FQDN or IP address of the email server
- `port` - TCP port needed to communicate with the email server
- `use` - Either `ssl` or `tls`
- `default_language` - Language to be used for translation (see below)
- `detault_timezone` - Timezone to be used for date time messages

Language Definitions

The Email or ClickSend consumer processes will translate messages into your native language.

For information about currently supported Native Languages, see Natural Language Support (page 0).

Timezones

Timezones are used when messages are translated and output to the user.

For more information about Timezone support, see Timezones (page 0).

Correctly edited QumuloEmailServer.json file

```
{
    "from_addr": "admin@xyzcorp.com",
    "to_addr": "joe@xyzcorp.com",
    "login": "admin@xyzcorp.com",
    "password": "ThisisMyPassword$!@",
    "server": "smtp.xyzcorp.com",
    "port": 587,
    "use": "ssl",
    "default_language": "en_GB",
    "default_timezone": "UTC"
}
```

1. The **from_addr** will be used in every message from the email server. It signifies to the email recipient the address that sent the email. This should be something that will readily identify why the email recipient is receiving this email.
2. The **to_addr** is **ONLY** used to test the email server. The actual **to_addr** for a real Qumulo Alerts messages comes from the **QumuloUsers.json** file.
3. The **login** is required by many email relays to verify that the somebody is not trying to utilize the email services of your organization to send bulk email.
4. The **password** is required, along with the login, by many email relays.
5. The **server** is the actual address of the STMP email relay.
6. The **port** is the port required by SMTP email relay. In most cases, it will be 587 for SSL and/or TLS. In some rare cases, some organizations still use 25.
7. The **use** field specifies whether your SMTP email relay requires either SSL or TLS encryption.
8. The **default_language** specifies the language that should be used to translate each of the messages. This can be overriden by individual users; specified in the **QumuloUsers.json** file that is described later.
9. The **default_timezone** specifies the timezone used to output the messages. Each event that occurs within Alerts is encoded with a timestamp based upon the UTC timezone. But, humans like to see time messages based upon their location. This field, like **default_language** can be overriden by individual users in the **QumuloUsers.json** file that is described later.

Optional Email fields

The fields **login**, **password**, and **use** are entirely optional based upon your email relay service. If your organization utilizes a simple SMTP relay behind a VPN, it is entirely possible that you do not have to login. So, it is safe (in those cases) to remove the login, password, and use fields.

A note about using gmail as a relay

As of the end of May 2022, only organizations with access to the Google Admin Console can utilize the SMTP relay. If your organization has that access, then you need to configure it using these instructions:

Using Google as an SMTP relay

Testing Email

Before you continue, you should test to see if your configuration entry for email will work with your email system. We have made this easy for you by including an email test program that will read your configuration file and send a test email.

In the Qumulo Alerts directory, you will find two versions of the test_email program. These are compiled versions of the same program for the architectures: `macos-latest` and `ubuntu-latest`.

Before you run any of these programs, you should verify that you have execute permissions. This needs to be done because the process of executing "git clone" to download this software does not guarantee that permissions are preserved.

Set the execute permissions for either Linux or MacOS with the command:

```
chmod a+x test_email.macos-latest (or test_email.ubuntu-latest)
```

Now, find the program based upon your architecture and run it by typing `./test_email.ARCHITECTURE --config ./config/consumer/` in a terminal window. If there are no errors, a test email should be sent to the **to_addr** in your config file.

Configuring Qumulo Alerts Integration with ClickSend

This section is a quick-start guide to configuring Qumulo Alerts to work with the ClickSend service.

> **Important**
> ClickSend is a paid service that provides delivery of SMS messages. If you wish to deliver SMS messages with Qumulo Alerts, then you are required to sign up for a ClickSend account before continuing with configuration.

What is ClickSend

ClickSend is a global leader in business communication solutions.

From bulk marketing to transaction based SMS, ClickSend is built on technical expertise, quality support and service reliability.

ClickSend has offices around the world and has been operating since 2013. Their head office is located in Perth, Australia.

> **Important**
> If you wish to send SMS messages in the USA and Canada, there are significant limitations due to current laws. In order to get around those limitations, you should sign up for a dedicated TFN (Toll Free Number) on the ClickSend website.

Getting started with ClickSend

Sign up for ClickSend

Configure Qumulo Alerts to use ClickSend

This section demonstrates how to configure the email system for delivery of alarms and alerts.

> **Note**
> - ClickSend is one of three processes known as **consumers**.
> - Consumers accept messages from Qumulo Alerts through the Exchange.

Directory Structure

You should be in the `config` directory and looking at the following subdirectories:

```
drwxr-xr-x  12 someone  somegroup  384 Aug  4 13:09 consumer
drwxr-xr-x   7 someone  somegroup  224 Aug  4 13:09 alerts
```

Let's start by editing the files in the `consumer` subdirectory.

`cd consumer`

You should see:

```
-rw-r--r--  1 someone  somegroup   208 Aug  3 17:06 QumuloEmailServer.json
-rw-r--r--  1 someone  somegroup   226 Aug 10 11:13 QumuloExchangeEMail.json
-rw-r--r--  1 someone  somegroup   118 Aug  3 17:06 QumuloClickSend.json
-rw-r--r--  1 someone  somegroup  1050 Aug 10 11:13 QumuloUsers.json
```

QumuloClickSend json file

If this file is not configured or configured improperly, then the ClickSend docker container will halt. It is entirely optional to configure email. If your organization wants to only use Email, then you can skip this section.

If, however, you wish to send SMS messages, then you need to configure the following entry properly.

If you wish to configure ClickSend, let's get started by editing the file `QumuloClickSend.json` using your favorite text based editor.

Once you have it open in the editor, you should see the following:

```
{
    "username": "",
    "token": "",
    "senderid": "",
    "default_language": "en_GB",
    "default_timezone": "UTC"
}
```

- `username` - name used to login to the ClickSend service.
- `token` - API Token created on the **Developers->API Credentials** page at ClickSend.com
- `senderid` - The TFN number created at ClickSend.com. This field is optional **except** in the USA and Canada
- `default_language` - Language to be used for translation (see below)

- `detault_timezone` - Timezone to be used for date time messages

Language Definitions

The Email or ClickSend consumer processes will translate messages into your native language.

For information about currently supported Native Languages, see Natural Language Support (page 0).

Timezones

Timezones are used when messages are translated and output to the user.

For more information about Timezone support, see Timezones (page 0).

Correctly edited QumuloClickSend.json file

```
{
    "username": "joe@garagecorp.com",
    "token": "B88DE584-E3DF-7173-E2345-78BDE4567",
    "senderid": "+18441234567",
    "default_language": "en_GB",
    "default_timezone": "UTC"
}
```

1. The **username** is the login name for the ClickSend service.
2. The **token** is API Token for the ClickSend service.
3. The **senderid** is the TFN number created at ClickSend.com. This field is entirely optional **except** in the USA or Canada.
4. The **default_language** specifies the language that should be used to translate each of the messages. This can be overriden by individual users as specified in the **QumuloUsers.json** file.
5. The **default_timezone** specifies the timezone used to output the messages. Each event that occurs within Alerts is encoded with a timestamp based upon the UTC timezone. But, humans like to see time messages based upon their location. This field, like **default_language** can be overriden by individual users in the **QumuloUsers.json** file.

> **Note**
>
> For detailed information about the **QumuloUsers.json** file, see User Notification (page 0).