

Qumulo Alerts Administrator Guide

Version 6.1.0



Copyright © 2026 Qumulo, Inc.

Table of Contents

Getting Started

How Qumulo Alerts Works	2
Supported Alarms and Alerts	4
Supported Language Locales	6

Installing and Configuring

8

Upgrading

Upgrading from a Previous Version	15
Upgrading from a Beta Version	16
Upgrading Grafana in Docker Container	17

Configuring Notifications

Alarm and Alert Notifications to Administrators.....	19
Default Quota Notifications.....	22
Quota Notifications to Administrators	24
Quota Notifications to Users	27

Configuring Integrations

Integration with an Email Server	29
Integration with IFTTT.....	32
Integration with SMS (ClickSend)	34

Configuring Alarm and Alert Collection.....

37

Connecting to Grafana

42

Getting Started

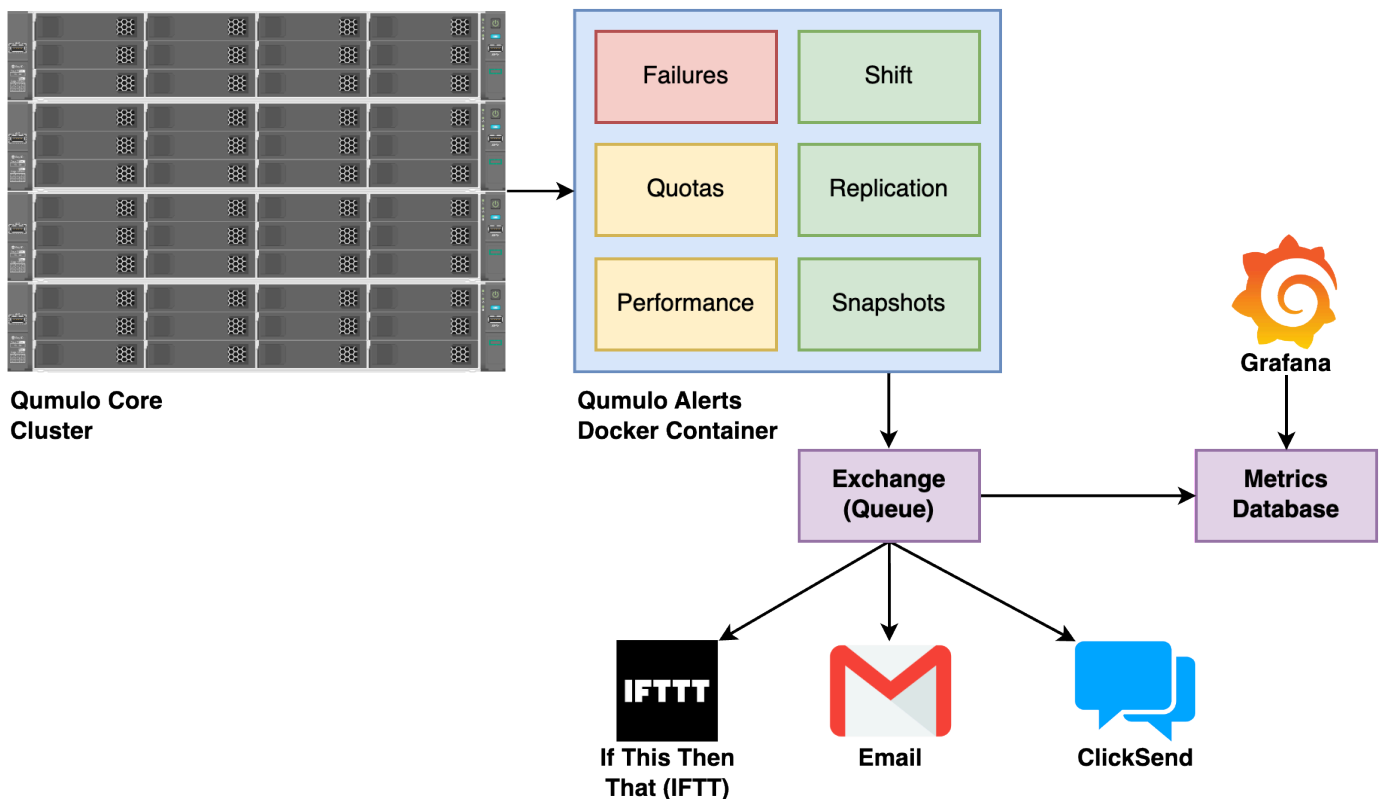
How Qumulo Alerts Works with Qumulo Core

This section explains how Qumulo Alerts monitors alarms and alerts for a Qumulo 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 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 the data to *consumers*, defined users or user groups. In addition to 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 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.

Working with the alerts CLI

The `alerts` CLI lets you configure Qumulo Alerts. For more information, use the `--help` flag.

Qumulo Alerts includes a CLI for the following operating systems:

- Ubuntu 20 and 22
- Red Hat Enterprise (RHEL) 8
- macOS
- Windows Server 2019 and Windows 10 and 11

Known Limits

This section lists the currently 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 nor 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 troubleshooting issues, [contact the Qumulo Care Team](#).

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
PSU	Power supply 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
Quotas	Quota notification (configured percentage for specified directories)

Plugin Name	Description
Replication	Replication added, deleted, enabled, or disabled; or an error message
Restriper	Restriper started, stopped, or percentage complete
Shares	SMB shares added, modified, or deleted
Softquotas	Soft quota notification (configured percentage for specified directories)
Users	Local users added, modified, or deleted
VPN	Remote Support enabled or disabled

Informational

The following informational notifications show performance and status information for a Qumulo cluster.

Plugin Name	Description
Metrics	Performance metrics (throughput, IOPS, and latency)
OSUpgrade	Qumulo Core upgrade

What Language Locales Qumulo Alerts Supports

This section lists the language locales that Qumulo Alerts supports for notifying users through email, IFTTT, and SMS (ClickSend).

Language Locales

The consumer processes for [email \(page 29\)](#), [IFTTT \(page 32\)](#), and [SMS \(ClickSend\) \(page 34\)](#) 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 use the `--timezone` flag when you create a user by using the `alerts` CLI, Qumulo Alerts uses `UTC` time.

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

Note

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

Installing and Configuring Qumulo Alerts

This section explains how to install, start and stop, log in to, and configure Qumulo Alerts.

Prerequisites

This section lists the prerequisites for Qumulo Alerts.

Firewall Ports

Qumulo Alerts requires the following firewall ports to be open from the Qumulo Alerts instance.

Port	Target
25 , 587 , or 465	Email server
3000	Any client that queries or views Grafana dashboards
8000	Qumulo cluster

System Requirements

We recommend the following system requirements for Qumulo Alerts.

- 4-core processor
- 16 GB memory
- 500 GB disk space

Tools

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.

Configuration Details

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:** 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 minutes or seconds.)
- **Alarm and Alert Types:** Which alarms and alerts will Qumulo Alerts will collect from your Qumulo cluster?

Installing Qumulo Alerts

This section explains how to install Qumulo Alerts on your machine.

Step 1: Clone the Qumulo Alerts 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. To connect to a node in your cluster as the `root` user, use SSH and then run the `sudo -s` command.
2. To create a local user, run the `qq auth_add_user` command and specify a name and password.

```
qq auth_add_user --name QumuloAlerts \  
--password HTEj0kGI0sNOAA0
```

3. To create a role for Qumulo Alerts later, you need the user ID that appears in the command output.

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-1-5-21-1234567890-345678912-1234567890-1234",
  "uid": ""
}
```

Step 3: Create a Qumulo Core Role for Qumulo Alerts

1. Log in to the Qumulo Core 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 `QumuloAlerts`.

Important

Because Qumulo Alerts verifies that it has sufficient role permissions before starting, this name is required.

- 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:
 - `ACCESS_TOKENS_READ` : View any access tokens present in the system
 - `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
 - `FILE_READ_ACCESS` : Provides read access to all files regardless of permissions
 - `FS_ATTRIBUTES_READ` : Read file system statistics
 - `FS_DELETE_TREE_READ` : View the status of directory tree delete operations

- **FS_KEY_MANAGEMENT_READ** : Read and list public keys for various FS security features
- **FS_LOCK_READ** : View NLM and SMB locks and waiters
- **FS_SETTINGS_READ** : View file system permissions settings
- **FTP_READ** : View FTP status and settings
- **IDENTITY_MAPPING_READ** : Get AD/LDAP User Defined Mappings
- **IDENTITY_READ** : Use Qumulo's identity lookup and translation APIs
- **KERBEROS_KEYTAB_READ** : View Kerberos keytab
- **KERBEROS_SETTINGS_READ** : Read Kerberos settings
- **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_IP_ALLOCATION_READ** : View network IP address allocations
- **NETWORK_READ** : Read network status and settings
- **NFS_EXPORT_READ** : Read network status and settings
- **NFS_SETTINGS_READ** : Internal-Only: View NFS server settings
- **QUOTA_READ** : View all file system quotas
- **REBOOT_READ** : View Reboot Status
- **RECONCILER_READ** : View reconciler status and metrics
- **REPLICATION_OBJECT_READ** : View object store relationship settings and status
- **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
- **S3_UPLOADS_READ** : View all S3 uploads present in the system.
- **SAML_SETTINGS_READ** : View SAML integration settings
- **SMB_FILE_HANDLE_READ** : List open SMB file handles

- `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 Qumulo Core User

1. In the Qumulo Core 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

Run 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 On-Premises Administrator Guide.

Starting and Stopping Qumulo Alerts

- To start Qumulo Alerts, run the `./start-docker-qumulo-alerts.sh` command from the Qumulo Alerts directory.
- To stop Qumulo Alerts, run the `./stop-docker-qumulo-alerts.sh` command from the Qumulo Alerts directory.

Logging In to Qumulo Alerts

This section explains how to log in to Qumulo Alerts by using the Web UI or the `alerts` CLI.

To Log In to the Qumulo Alerts Web UI

📘 Note

The Qumulo Alerts Web UI is available in Qumulo Alerts 7.2.1 (and higher).

1. In a browser, navigate to `http://<your-host>:8080/web/login`, where `<your-host>` is the hostname or IP address of the machine running Qumulo Alerts.
2. Enter the default credentials:
 - a. For Username, enter `admin`.
 - b. For Password, enter `Admin123`.
3. Click Login.

After a successful login, Qumulo Alerts creates a session that expires after six hours.

To Log In to Qumulo Alerts by Using the alerts CLI on Linux or macOS

Qumulo Alerts includes the following binaries:

- `alerts.macos-latest`
- `alerts.redhat-8`
- `alerts.ubuntu-20.04`
- `alerts.ubuntu-latest`

1. Copy the appropriate binary to your Linux or macOS machine.
2. Link the binary for your operating system to the `alerts` CLI. For example:

```
ln -s alerts.ubuntu-20.04 alerts
```

3. Make the binary file executable. For example:

```
chmod a+x alerts.ubuntu-20.04
```

To Log In to Qumulo Alerts by Using the alerts CLI on Windows

1. Copy `alerts.windows-latest.exe` binary to your Windows machine.
2. Rename the file to `alerts.exe`

Step 2: Log In to Qumulo Alerts by Using the alerts CLI

1. To log in to Qumulo Alerts, run the `./alerts login -u admin` command on Linux or macOS or `alerts login -u admin` on Windows.
2. When prompted, enter your credentials:
 - For Login, enter `admin`.
 - For Password, enter `Admin123`.

Configuring Qumulo Alerts

1. Configure integration [with an email server \(page 29\)](#) or [with SMS \(ClickSend\) \(page 34\)](#).
2. [Configure alarm and alert notifications \(page 19\)](#).
3. [Configure Collection of Alarms and Alerts from a Qumulo Cluster \(page 37\)](#).

Upgrading

Upgrading Qumulo Alerts from a Previous Public Version

This section explains how to upgrade Qumulo Alerts from a previous public version to the latest one.

To Upgrade Qumulo Alerts to the Latest Public Version

1. To shut down Qumulo Alerts, navigate to its directory and run the `./stop-docker-qumulo-alerts.sh` command.

i Note

This process might take up to 60 seconds. The Alerts Docker container must shut down and then verify that all Qumulo Alerts Docker containers are also shut down correctly.

2. In the Qumulo Alerts directory, run the `git pull` command.
3. To remove all existing Qumulo Alerts Docker images from your machine, run the `docker system prune -a -f` command.

i Note

This release of Qumulo Alerts adds new Docker containers, making it necessary to remove all existing images.

4. To restart the Docker containers for Qumulo Alerts, pull new Docker images from the Qumulo Docker repository, and restart all Docker containers, run the `./start-docker-qumulo-alerts.sh` command.

Upgrading Qumulo Alerts from the Beta Version

This section explains how to upgrade Qumulo Alerts from the beta version.

i Note

- Whereas the beta version of Qumulo Alerts uses JSON files for configuration, the public version of Qumulo Alerts uses its API or the `aalerts` CLI to store configuration information in a database.
- It isn't possible to upgrade from the beta version of Qumulo Alerts to the public version automatically. To enable upgrades from a beta version, you must perform the following manual steps.

To Prepare for Upgrading Qumulo Alerts from the Beta Version

1. To shut down Qumulo Alerts, navigate to its directory and run the `./stop-docker-qumulo-alerts.sh` command.
2. Copy the information from the `user_token` field located in the `QumuloAlerts/config/alerts/QumuloAlerts.json` file.
3. Rename the directory of the beta version of Qumulo Alerts, for example to `QumuloAlerts.beta`.
4. [Install the latest public version of Qumulo Alerts \(page 8\)](#).
5. When you [configure alarm and alert collection from your Qumulo cluster \(page 37\)](#), use the information from the `user_token` field.

Upgrading Grafana in the Qumulo Alerts Docker Container

This section explains how to upgrade Grafana in a Qumulo Alerts Docker container from a machine running Ubuntu 22.04.

Prerequisites

To perform these instructions, ensure that you have `root` privileges and that Docker is installed.

i Note

If you experience any issues during the upgrade process, you can diagnose potential issues by running the `docker logs` command and specifying the ID of the Grafana container.

To Upgrade Grafana in the Qumulo Alerts Docker Container

1. To switch to the `root` user, run the `sudo -s` command.
2. Navigate to the `/opt/qumulo/QumuloAlerts` directory that contains the Qumulo Alerts Docker configuration files.
3. To update the Grafana configuration, ensure that the following line in the `docker-compose.yml` file specifies version `10.4.2`:

```
image: grafana/grafana-oss:${GRAFANA_VERSION:-10.4.2}
```

4. To stop all running Qumulo Alerts Docker containers, run the `stop-docker-qumulo-alerts.sh` script.
5. To remove the previous Grafana Docker image:
 - a. Run the `docker image ls` command to list all existing image IDs.
 - b. Run the `docker rmi` command and specify the ID of the image. For example:

```
docker rmi grafana/grafana:10.2.0
```

6. To restart Qumulo Alerts with the new configuration, run the `start-docker-qumulo-alerts.sh` script.
7. To verify the Qumulo Alerts Docker container's status, you can run the `docker ps -a` command or access Grafana by using its configured endpoint and port. For example:

<http://127.0.0.1:3000>

Configuring Notifications

Configuring Alarm and Alert Notifications to an Administrative Account in Qumulo Alerts

This section explains how to configure Qumulo Alerts to send alarm and alert notifications from a Qumulo cluster to an administrative account.

You must first add the account as a Qumulo Alerts user, create a notification group and configure its notifications, and then add the user to the notification group.

Step 1: Add an Administrative Account as a Qumulo Alerts User

Run the `./alerts user_add` command and specify the administrator's full name, username, password, email address, language, and time zone. For example:

```
./alerts user_add \  
  --full-name "Jane Johnson" \  
  --username jjohnson \  
  --password HTEj0kGI0sNOAA0 \  
  --email jjohnson@example.com \  
  --language en_US \  
  --timezone "America/Los_Angeles"
```

Note

- For more information about locales, see [What Language Locales Qumulo Alerts Supports](#) (page 6). The consumer processes for email (page 29), IFTTT (page 32), and SMS (ClickSend) (page 34) integrations translate messages into the recipient's native language.
- For more information about time zones, see [Converting Time Zones](#) (page 7).

The following is example output.

```
[{
  "disabled": false,
  "email": "jjohnson@example.com",
  "full_name": "Jane Johnson",
  "id": 3,
  "ifttt_event": null,
  "language": "en_US",
  "phone": null,
  "timezone": "America/Los_Angeles",
  "username": "jjohnson"
}]
```

Step 2: Create and Configure a Notification Group

Run the `./alerts notification_group_add` command and specify the notification group's name, description, and the events for which the notification group receives notifications. In the following example, the `NotifyOnHardwareChange` group receives notifications for all hardware state change events.

```
./alerts notification_group_add \
  --name NotifyOnHardwareChange \
  --description "Send a notification when any hardware changes state" \
  --event NOTIFY_FANS \
  --event NOTIFY_CPU \
  --event NOTIFY_DISKS \
  --event NOTIFY_NETWORK \
  --event NOTIFY_NODES
```

The following is example output.

```
[{
  "description": "Send a notification when any hardware changes state",
  "id": 2,
  "name": "NotifyOnHardwareChange"
}]
```

Step 3: Add a Qumulo Alerts User to a Notification Group

Run the `./alerts notification_group_add_user` command and specify the notification group name and the Qumulo Alerts user name to add to the notification group. For example:

```
./alerts notification_group_add_user \  
  --name NotifyOnHardwareChange \  
  --username jjohnson
```

The following is example output.

```
[{  
  "description": "Notify when certain hardware changes state",  
  "id": 2,  
  "name": "NotifyOnHardwareChange",  
  "users": [{  
    "can_change_password": true,  
    "disabled": false,  
    "email": "jjohnson@example.com",  
    "full_name": "Jane Johnson",  
    "id": 3,  
    "ifttt_event": null,  
    "language": "en_US",  
    "phone": null,  
    "timezone": "America/Los_Angeles",  
    "username": "jjohnson"  
  }]  
}]
```

Configuring Default Quota Notifications in Qumulo Alerts

This section explains how to configure default quota notifications in Qumulo Alerts.

Qumulo Alerts lets an administrator configure notifications that inherit a template from one of the following default quotas.

- **No-Path Quota:** This quota type has no defined file system path. It is the most common quota type and it applies thresholds to every quota defined for a Qumulo cluster.
- **Inherited-Path Quotas:** This quota type lets an administrator specify a default path for every quota defined for a Qumulo cluster. Every quota created under the default path inherits its thresholds from this quota.

You can configure quota monitoring by using *thresholds*.

- For the `--warning` flag, the threshold must be lower than the thresholds of both the `--error` and `--critical` flags.
- For the `--error` flag, the threshold must be lower than the threshold of the `--critical` flag.
- For the `--critical` flag, the threshold must be greater than the thresholds of both the `--warning` and `--error` flags.

For more information about how quotas work, see [Configuring Quota Notifications to an Administrative Account \(page 0\)](#) and [Configuring Quota Notifications to a User Account \(page 0\)](#).

To List the Predefined No-Path Quota

Qumulo Alerts comes with a predefined no-path quota. To get information about this quota, run the `./alerts default_quota_list` command.

The following is example output.

```
[{
  "items": [{
    "admin_notification": true,
    "critical": 95,
    "error": 85,
    "id": 1,
    "quota_prefix": "",
    "user_mode": "owner",
    "user_notification": false,
    "warning": 75
  }],
  "page": 1,
  "pages": 1,
  "size": 50,
  "total": 1
}]
```

To Configure an Inherited-Path Quota

Run the `./alerts default_quota_add` command and specify the default path and thresholds. For example:

```
./alerts default_quota_add \
--quota-prefix /Home \
--warning 80 \
--error 90 \
--critical 98
```

The following is example output.

```
[{
  "admin_notification": true,
  "critical": 98,
  "error": 90,
  "id": 2,
  "quota_prefix": "/Home/",
  "user_mode": "owner",
  "user_notification": false,
  "warning": 80
}]
```

Configuring Quota Notifications to an Administrative Account in Qumulo Alerts

This section explains how to configure Qumulo Alerts to send quota notifications from a Qumulo cluster to an administrative account.

You can configure quota monitoring by using *thresholds*.

- For the `--warning` flag, the threshold must be lower than the thresholds of both the `--error` and `--critical` flags.
- For the `--error` flag, the threshold must be lower than the threshold of the `--critical` flag.
- For the `--critical` flag, the threshold must be greater than the thresholds of both the `--warning` and `--error` flags.

You can configure unattached quotas or attach them to a Qumulo cluster.

To Configure Quota Notifications with Two Thresholds

Run the `./alerts quota_add` command and specify the quota path to monitor. The following example specifies the warning threshold and the error threshold and doesn't attach the quota to a Qumulo cluster.

```
./alerts quota_add \  
  --quotapath /Reports/Sales \  
  --warning 80 \  
  --error 85
```

The following is example output.

```
[{  
  "admin_notification": true,  
  "critical": 95,  
  "error": 85,  
  "id": 2,  
  "quota_path": "/Reports/Sales/",  
  "user_email": "",  
  "user_mode": "direct",  
  "user_notification": false,  
  "warning": 80  
}]
```

To Configure Quota Notifications with a Single Threshold

Run the `./alerts quota_add` command and specify the quota path. The following example specifies the error threshold and attaches the quota to the fully qualified domain name (FQDN) of a Qumulo cluster.

```
./alerts quota_add \  
  --quotapath /Reports/Marketing \  
  --error 90 \  
  --cluster-include cluster.example.com
```

i Note

When you add a quota and attach it to a Qumulo cluster, the alerts CLI doesn't list the cluster.

The following is example output.

```
[{  
  "admin_notification": true,  
  "critical": 95,  
  "error": 90,  
  "id": 3,  
  "quota_path": "/Movies/Dutch/",  
  "user_email": "",  
  "user_mode": "direct",  
  "user_notification": false,  
  "warning": 75  
}]
```

To List All Defined Quotas and Attached Clusters

Run the `./alerts quota_list` command.

The following is example output. In this example, the second quota is attached to the fully qualified domain name (FQDN) of a Qumulo cluster.

```
[{
  "items": [{
    "admin_notification": true,
    "clusters": [],
    "critical": 95,
    "error": 85,
    "id": 2,
    "quota_path": "/Reports/Sales/",
    "user_email": "",
    "user_mode": "direct",
    "user_notification": false,
    "warning": 80
  },{
    "admin_notification": true,
    "clusters": [{
      "frequency": 1,
      "name": "cluster.example.com",
      "nlb": false,
      "port": 8000
    }],
    "critical": 95,
    "error": 90,
    "id": 3,
    "quota_path": "/Reports/Marketing/",
    "user_email": "",
    "user_mode": "direct",
    "user_notification": false,
    "warning": 75
  }],
  "page": 1,
  "pages": 1,
  "size": 50,
  "total": 2
}]
```

Configuring Quota Notifications to One or More User Accounts in Qumulo Alerts

This section explains how to configure Qumulo Alerts to send quota notifications from a Qumulo cluster to one or more user accounts.

To Configure Quota Notifications by Using the Qumulo Alerts Web UI

1. On the sidebar, under **Quotas**, click **Quotas**.
2. Click **+ Add Quota** or click **Edit** next to an existing rule.
3. In the **Clusters** section, select one or more Qumulo clusters to add to this quota and then enter the quota rules:
 - For **Quota Path**, enter the quota directory path on your Qumulo cluster (for example, `/Reports/Marketing`).
 - For **Warning (%)**, enter the quota usage percentage that triggers a *warning* notification (`70` by default).
 - For **Error (%)**, enter the quota usage percentage that triggers an *error* notification (`80` by default).
 - For **Critical (%)**, enter the quota usage percentage that triggers a *critical* notification (`90` by default).
4. Click **Notify Users entered below** and enter an email address.
5. (Optional) To add more addresses, click **+**.
6. Click **Save**.

To Configure Quota Notifications by Using the alerts CLI

Run the `./alerts quota_add` command and specify the quota path, the email address to notify, and the fully qualified domain name (FQDN) of your Qumulo cluster. For example:

```
./alerts quota_add \  
  --quotapath /Reports/Marketing \  
  --user-notification \  
  --user-mode direct \  
  --user-email jjohnson@example.com \  
  --cluster-include cluster.example.com
```

Note

For the `--user-email` flag, you can specify a comma-delimited list of email addresses to notify, if you also specify `--user-notification --user-mode direct`.

The following is example output.

```
[{
  "admin_notification": true,
  "critical": 95,
  "error": 85,
  "id": 1,
  "quota_path": "/Reports/Marketing/",
  "user_email": "jjohnson@example.com",
  "user_mode": "direct",
  "user_notification": true,
  "warning": 75
}]
```

Configuring Integrations

Configuring Qumulo Alerts Integration with an Email Server

This section explains how to integrate an email server with Qumulo Alerts and test the integration.

i Note

After May 2022, only organizations with access to the Google Admin Console can use SMTP relay to Route outgoing SMTP relay messages through Google.

To Integrate an Email Server with Qumulo Alerts and Test the Integration by Using the Qumulo Alerts Web UI

i Note

- Depending on your SMTP server configuration, the Username, Password, and Security fields might be optional.
- For more information about locales, see [What Language Locales Qumulo Alerts Supports](#) (page 6). The consumer processes for email (page 29), IFTTT (page 32), and SMS (ClickSend) (page 34) integrations translate messages into the recipient's native language.
- For more information about time zones, see [Converting Time Zones](#) (page 7).

1. On the sidebar, under Servers, click **Email Server**.
2. Enter the SMTP configuration details:
 - a. For **SMTP Server**, enter the hostname or IP address of your SMTP server (for example, `mail.example.com`).
 - b. For **Port**, select the SMTP server port.
 - c. For **From Address**, select the email address to appear in the From field of outgoing notifications (for example, `alerts@example.com`).
 - d. For **Security**, select the connection security type.
 - e. (Optional) For **Username**, enter the username for SMTP authentication.
 - f. (Optional) For **Password**, enter the password for SMTP authentication.

✓ Tip

To keep the current password, leave **Password** empty.

- g. For **To Address** (used only for testing the connection), enter the email address to receive the test message when you click **Test Connection**.
 - h. For **Default Language**, enter a [supported language locale \(page 6\)](#) to use for notification email templates when a user-level language isn't configured.
 - i. For **Default Timezone**, enter the the time zone for formatting timestamps in notification emails when a user-level timezone isn't configured.
3. Click **Save Configuration**.
 4. Click **Test Connection**.
 5. Qumulo Alerts sends a test message to the configured email address.
- If the test succeeds, a confirmation message appears.
 - If the test fails, check the configuration details.

To Integrate an Email Server with Qumulo Alerts and Test the Integration by Using the alerts CLI

i Note

- Depending on the type of SMTP email server that you use, the `--login`, `--password`, and `--security` flags might be optional.
- For more information about locales, see [What Language Locales Qumulo Alerts Supports \(page 6\)](#). The consumer processes for email (page 29), IFTTT (page 32), and SMS (ClickSend) (page 34) integrations translate messages into the recipient's native language.
- For more information about time zones, see [Converting Time Zones \(page 7\)](#).

1. Run the `./alerts_email_server_add` command and specify the sender's email address, recipient's email address, email server hostname and port, language, and time zone. For example:

```
./alerts email_server_add \  
  --from-addr alerts@example.com \  
  --to-addr name@example.com \  
  --server mail.example.com \  
  --port 25  
  --language en_US  
  --timezone "America/Los_Angeles"
```

The following is example output.

```
[{  
  "from_address": "alerts@example.com",  
  "language": "en_US",  
  "login": null,  
  "password": null,  
  "port": 25,  
  "security": null,  
  "server": "mail.example.com",  
  "timezone": "America/Los_Angeles",  
  "to_address": "name@example.com"  
}]
```

2. Run the `./alerts email_server_test` command.

A successful response returns the `[{ "ok": true }]` JSON output.

Configuring Qumulo Alerts Integration with IFTTT

This section explains how to integrate IFTTT with Qumulo Alerts and test the integration.

[IFTTT \(If This Then That\)](#) is a paid, third-party service that provides delivery of messages by using [Webhooks integrations](#) and events. For more information, see the [IFTTT documentation](#).

To Integrate IFTTT with Qumulo Alerts and Test the Integration by Using the Qumulo Alerts Web UI

1. Create a Webhooks applet in IFTTT and obtain the Webhooks key from the [IFTTT Maker Webhooks](#) page.
 2. Log in to Qumulo Alerts.
 3. On the sidebar, under **Servers**, click IFTTT.
 4. For the **Webhook Key** field, enter your IFTTT Webhooks key.
 5. Click **Save Configuration**.
 6. Click **Test Connection**.
 7. Qumulo Alerts sends a test event to your IFTTT Webhooks.
- If the test succeeds, a confirmation message appears.
 - If the test fails, check your Webhooks key and that your IFTTT applet is active.

To Integrate IFTTT with Qumulo Alerts and Test the Integration by Using the alerts CLI

1. Run the `./alerts ifttt_server_add` command and specify the IFTTT server token, language, and time zone. For example:

```
./alerts ifttt_server_add \  
  --token abcABde12f3g4567CDE89 \  
  --language en_US \  
  --timezone "America/Phoenix"
```

Note

- For more information about locales, see [What Language Locales Qumulo Alerts Supports](#) (page 6). The consumer processes for email (page 29), IFTTT (page 32), and SMS (ClickSend) (page 34) integrations translate messages into the recipient's native language.
- For more information about time zones, see [Converting Time Zones](#) (page 7).

The following is example output.

```
[{  
  "language": "en_US",  
  "timezone": "America/Phoenix",  
  "token": "abcABde12f3g4567CDE89"  
}]
```

2. Run the `./alerts ifttt_server_test` command.

A successful response returns the `[{ "ok": true }]` JSON output.

Configuring Qumulo Alerts Integration with ClickSend (SMS)

This section explains how to integrate ClickSend with Qumulo Alerts and test the integration.

[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.

Important

To be able to send SMS in the U.S. and Canada, you must sign up for a dedicated toll-free number (TFN).

To Integrate ClickSend with Qumulo Alerts and Test the Integration by Using the Qumulo Alerts Web UI

1. On the sidebar, under Servers, click **ClickSend SMS**.
2. Enter the ClickSend configuration details:
 - a. For **Username**, enter your ClickSend account username (typically, your email address).
 - b. For **API Key**, enter the API key from your ClickSend account dashboard.

Tip

To keep the current key, leave **API Key** empty when you edit an existing ClickSend configuration.

- c. For **From Number**, enter the phone number or sender ID to appear as the sender of outgoing SMS messages, formatted according to the [E.164 standard](#) (for example, **+15555550100**).
 - d. For **Test Number**, enter the phone number to receive the test SMS when you click **Test Connection**.
3. Click **Save Configuration**.

Note

You must configure recipient phone numbers separately for each user on the [Alert Recipients](#) page. For more information, see [Configuring Alarm and Alert Notifications to an Administrative Account](#) (page 19).

4. Click **Test Connection**.
 5. Qumulo Alerts sends a test SMS to the configured test number.
- If the test succeeds, a confirmation message appears.
 - If the test fails, check the configuration details.

To Integrate ClickSend with Qumulo Alerts and Test the Integration by Using the alerts CLI

1. Run the `./alerts clicksend_server_add` command and specify the username, token, sender ID, and recipient's phone number.

```
./alerts clicksend_server_add \  
  --username name@example.com \  
  --token 12345678-ABCDEFGH-12345678-ABCDEFGH \  
  --senderid "+15551234567" \  
  --to-address "+15550987654"
```

Note

- For the `--username` and `--token` flags, see [API Credentials](#) in the ClickSend documentation.
- The `--senderid` flag is mandatory for the U.S. and Canada. For more information, see [How to Register a Toll-Free Number \(TFN\) with ClickSend](#) in the ClickSend documentation.
- For more information about locales, see [What Language Locales Qumulo Alerts Supports](#) (page 6). The consumer processes for email (page 29), IFTTT (page 32), and SMS (ClickSend) (page 34) integrations translate messages into the recipient's native language.
- For more information about time zones, see [Converting Time Zones](#) (page 7).

The following is example output.

```
[{
  "language": "en_GB",
  "senderid": "+15551234567",
  "timezone": "UTC",
  "to_address": "+15550987654",
  "username": "name@example.com"
}]
```

2. Run the `./alerts clicksend_server_test` command.

Note

For integration testing to complete successfully, the `--to-address` flag must be configured already.

A successful response returns the `[{ "ok": true }]` JSON output. In addition, the recipient's phone number receives a test message.

Configuring Alarm and Alert Collection from a Qumulo Cluster

This section explains how to collect alarms and alerts from a Qumulo Cluster by using the Qumulo Alerts Web UI and the `alerts` CLI.

To Configure Alarm and Alert Collection by Using the Qumulo Alerts Web UI

This section explains how to configure alarm and alert collection by adding your Qumulo cluster to Qumulo Alerts and then configuring alert plugins for your cluster in the Qumulo Alerts Web UI.

⚠ Important

- When you add a Qumulo cluster to Qumulo Alerts, all available alarm and alert plugins are enabled for that cluster by default.
- If you use a floating IP address, you must click **Network Load Balancer** to ensure that Qumulo Alerts connects to the node that currently uses the floating IP address.
- To avoid spreading a plugin's API request load across the nodes of a Qumulo cluster, all alarm and alert plugins communicate with your cluster by using either a network load balancer or floating IP addresses. You can configure *one*—but not both—of these communication methods.

Step 1: Add a Qumulo Cluster to Qumulo Alerts

1. Log in to the Qumulo Alerts Web UI.
2. On the sidebar, under **Monitoring**, click **Clusters**.
3. Click **+ Add Cluster** and enter the following details:
 - a. For **Cluster Name/IP**, specify the hostname or IP address for your Qumulo cluster.

⚠ Important

Qumulo Alerts uses this value as the display name. It isn't possible to change it after saving the cluster configuration.

- b. For **Access Token**, specify the long-lived access token that you created while [Installing and Configuring Qumulo Alerts \(page 12\)](#) guide.
- c. For **Port**, specify the REST API port (`8000` by default).

- d. For **Polling Frequency (minutes)**, specify how frequently Qumulo Alerts should poll your Qumulo cluster.
- e. (Optional) If your Qumulo cluster is accessible through a floating IP address behind a network load balancer, click **Network Load Balancer**.

4. Click **Save**.

Step 2: Configure Alarm or Alert Plugins for a Qumulo Cluster

✓ Tip

To view all available plugin names and categories before configuring a cluster, click **Monitoring > Alert Types** on the sidebar

1. Log in to the Qumulo Alerts Web UI.
2. On the sidebar, under **Monitoring**, click **Clusters** and then click **Edit** next to the cluster to configure.
3. In the **Plugins** section, click individual alarm and alert plugins to enable or disable them for your Qumulo cluster.
4. Click **Save**.

To Configure Alarm and Alert Collection by Using the alerts CLI

This section explains how to collect information and specific alarms; all alarms; or all alarms, alerts, and informational messages by using the **alerts** CLI.

Collecting Information about Specific Alarms

Run the **./alerts cluster_add** command and specify the fully qualified domain name (FQDN) of your Qumulo cluster, your long-lived access token for the Qumulo REST API, and the plugins or plugin categories to include or exclude from monitoring.

In the following example, we include the plugins **Disks** and **Nodes**.

```
./alerts cluster_add \  
  --name cluster.example.com \  
  --token 12345678901234567890 \  
  -pi Disks \  
  -pi Nodes
```

The following is example output.

```
[{
  "frequency": 1,
  "id": 1,
  "name": "cluster.example.com",
  "nlb": false,
  "plugins": [{
    "category": "Alarms",
    "description": "Get Disk State Information",
    "frequency": null,
    "name": "Disks"
  },{
    "category": "Alarms",
    "description": "Get Cluster Node Failures",
    "frequency": null,
    "name": "Nodes"
  }],
  "port": 8000
}]
```

i Note

- For the `--nlb` flag, the `false` setting requires floating IP address configuration.
- To avoid spreading a plugin's API request load across the nodes of a Qumulo cluster, all alarm and alert plugins communicate with your cluster by using either a network load balancer or floating IP addresses. You can configure *one*—but not both—of these communication methods.

Collecting Information about All Alarms

Run the `./alerts cluster_add` command and specify the fully qualified domain name (FQDN) of your Qumulo cluster, your long-lived access token for the Qumulo REST API, and the plugins or plugin categories to include or exclude from monitoring.

In the following example, we include the `Alarms` category.

```
./alerts cluster_add \
  --name cluster.example.com \
  --token 12345678901234567890 \
  -pc Alarms
```

The following is example output.

```
[{
  "frequency": 1,
  "id": 1,
  "name": "cluster.example.com",
  "nlb": false,
  "plugins": [{
    "category": "Alarms",
    "description": "Get Disk State Information",
    "frequency": null,
    "name": "Disks"
  },{
    "category": "Alarms",
    "description": "Get Cluster Node Failures",
    "frequency": null,
    "name": "Nodes"
  },{
    "category": "Alarms",
    "description": "Get Fan Failures",
    "frequency": null,
    "name": "Fans"
  },{
    "category": "Alarms",
    "description": "Get CPU Overtemp",
    "frequency": null,
    "name": "CPU"
  },
  ...
],
"port": 8000
}]
```

Collecting Information about All Alarms, Alerts, and Informational Messages

Run the `./alerts cluster_add` command and specify the fully qualified domain name (FQDN) of your Qumulo cluster, your long-lived access token for the Qumulo REST API, and the plugins or plugin categories to include or exclude from monitoring.

In the following example, we include the `Alarms`, `Alerts`, and `Informational` categories.

```
./alerts cluster_add \
--name cluster.example.com \
--token 12345678901234567890 \
-pc Alarms \
-pc Alerts \
-pc Informational
```

The following is example output.

```
[{
  "frequency": 1,
  "id": 1,
  "name": "cluster.example.com",
  "nlb": false,
  "plugins": [{
    "category": "Alarms",
    "description": "Get Disk State Information",
    "frequency": null,
    "name": "Disks"
  },{
    "category": "Alarms",
    "description": "Get Cluster Node Failures",
    "frequency": null,
    "name": "Nodes"
  },{
    "category": "Alerts",
    "description": "Get Active Directory State",
    "frequency": null,
    "name": "AD"
  },{
    "category": "Alerts",
    "description": "Get Audit Status",
    "frequency": null,
    "name": "Audit"
  },{
    "category": "Alerts",
    "description": "Get Cluster Volume Capacity",
    "frequency": null,
    "name": "Capacity"
  },
  ...
],
  "port": 8000
}]
```

Connecting to Grafana to View Visualizations of Qumulo Alerts Data

This section explains how to connect to the Qumulo Alerts instance of [Grafana](#) to view visualizations and information about your Qumulo cluster from prebuilt dashboards.

To Connect to the Grafana Endpoint

1. In a browser, navigate to the hostname of your running Grafana instance on port 3000.
For example:

```
http://203.0.113.0:3000
```

✓ Tip

Running the `./start-docker-qumulo-alerts.sh` script starts Grafana.

2. When prompted, enter the default credentials:
 - a. For Login, enter `qumulo`.
 - b. For Password, enter `Admin123`.

Grafana displays visualizations and information about your cluster.

3. [Change the default Grafana password.](#)