2021

Lyve Rack R1 User Guide

P/N: 1104757-01
Revision: B
June 2021

Notices

© 2021 Seagate Cloud Systems, Inc. All rights reserved. Seagate, Seagate Technology and the Spiral logo are registered trademarks of Seagate Technology LLC in the United States and/or other countries. All other trademarks or registered trademarks are the property of their respective owners. When referring to drive capacity, one gigabyte, or GB, equals one billion bytes and one terabyte, or TB, equals one trillion bytes. Your computer's operating system may use a different standard of measurement and report a lower capacity. In addition, some of the listed capacity is used for formatting and other functions, and thus will not be available for data storage. Actual data rates may vary depending on operating environment and other factors. The export or re-export of Seagate hardware or software is regulated by the U.S. Department of Commerce, Bureau of Industry and Security (for more information, visit www.bis.doc.gov), and may be controlled for export, import and use in other countries. All coded instruction and program statements contained herein is, and remains copyrighted works and confidential proprietary information of Seagate Technology LLC or its affiliates. Any use, derivation, dissemination, reproduction, or any attempt to modify, reproduce, distribute, disclose copyrighted material of Seagate Technology LLC, for any reason, in any manner, medium, or form, in whole or in part, if not expressly authorized, is strictly prohibited. Seagate reserves the right to change, without notice, product offerings or specifications.

Table of Contents

Revision History

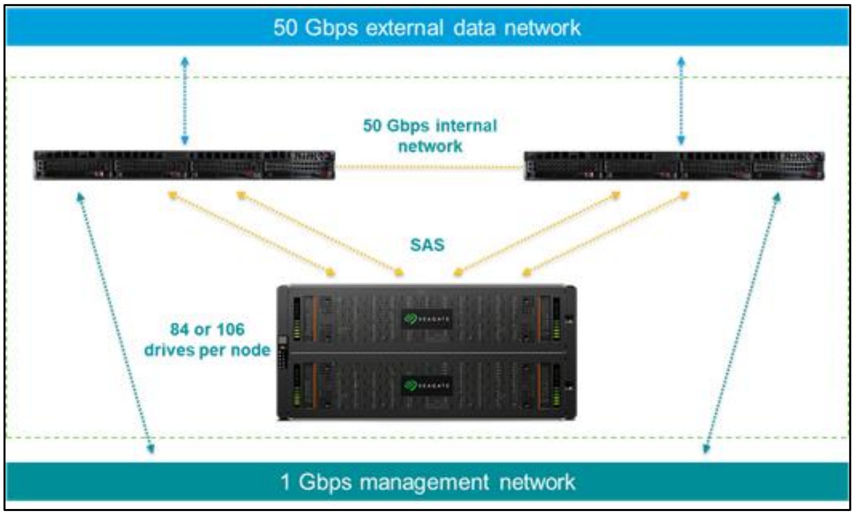
Revision History	Description	Release month and year
B	Release	June 2021

1 | Welcome to Seagate Lyve Rack!

Lyve Rack is a highly reliable mass capacity object storage solution powered by 100% open source CORTX software. Lyve Rack supports rich set of S3-compatible Storage features (such as multiple S3 Accounts, IAM users, and S3 buckets with access policies).

The data is stored using a high-performance and scalable object store. Data at-rest is protected by ADAPT – a proprietary Seagate Technology IP which dramatically reduces the time needed to re-build a failed disk and improves data durability. The storage is accessed via multipath SAS connections from a two-server cluster that receives S3 data from the users. Data in-flight is thus protected by high-availability and fast-failover provided by the dual-node cluster.

A Lyve Rack system separates management and data path. Separate IPs must be specified for management access (further named “Management IP”) and S3 data access (further named “Data IP”). In case of a component failure, active IP will transparently failover to the remaining node.


50 Gbps external data network
50 Gbps internal network
SAS
84 or 106 drives per node
1 Gbps management network

2 | Benefits

- Lyve Rack is easy to set up, maintain, and deploy.
- Industry-standard S3 object protocol with support for high capacity, dense storage.
- Built-in data protection with ADAPT Technology. Protect the most valuable business assets with Seagate Secure™ cybersecurity features and intelligent firmware. Rebuild drives faster than ever and reduce downtime with Seagate ADAPT data protection technology.
- Cost effective.

3 | Using CORTX Manager

The CORTX Manager is a graphical user interface to manage the Lyve Rack.

To start using the system, you need to complete the onboarding process to set up the system. For more information, refer Chapter 10| Onboarding Lyve Rack of the Lyve Rack R1 Installation Guide. After completing the onboarding process, you can access the CORTX Manager which is set up on your network.

3.1 | Accessing CORTX Manager

To access the CORTX Manager web interface using a web browser:

- In a web browser, enter https://<MANAGEMENT_VIP>:28100/#/login

4 | Using the CLI

This chapter introduces the command-line interface (CLI) of Lyve Rack.

Accessing the CLI

The CLI software embedded in the controller modules enables you to manage S3 operations like CRUD operations on bucket, s3accounts, and iam users.

To access the CORTX Manager CLI, you must first create your account by following onboarding steps. For more information, refer the Lyve Rack R1 Installation guide.

In the preboarding process, you can create the admin account. After creating an account, you can access the CLI by using SSH on a management host that is remotely connected through a SAS cable to a controller module's network port.

Note

When you first log in to the CLI, you will be prompted to create a user and password, which you can use to create additional users and to configure and provision the system.

CLI output formats

The CLI has two output formats:

- Console format, which is the human-to-computer interface (HCI).
  Console format enables users to interact with the CLI and obtain easily readable information. This format automatically sizes fields according to content and adjusts the content to window's size. These capabilities would present problems for a CCI in the form of scripts or another client software. In console format, some commands display confirmation prompts.
- API format, which is the computer-to-computer interface (CCI).
  API format enables any external application to interact with the storage system. Only JSON format is supported.

Scripting is not recommended using console format because labels, field sizes, and order of fields may change in future software releases.

Using CLI interactively

By default, the CLI is an interactive application. When you are logged into the CLI, the CLI waits for a command to be entered and then responds to it.

IMPORTANT

In the interactive mode, confirmation is required for commands that may cause data unavailability or data loss.

The following example shows interactively starting an SSH session, logging into the CLI, executing a command to show the alerts occurred in past 100 days, and exiting the CLI:

```
login as: <admin>
<admin>@<IP-address>'s password:
[admin@sm18-r18 ~]# cortxcli
Username:
Password:

**********************************

CORTX Interactive Shell
Type -h or --help for help.

***********************************

cortxcli$ alerts show -d100d
---------------------+----------+--------------------+--------------+--------
--+------------------+---------------+
| Alert Id     | Health |  Description                    |Severity| State|
Acknowledged | Resolved |
| 156b72a8a958 |  None  | Please contact Seagate Support. |critical| fault|
False |  False|
| 1598748bda06 |        |    | warning  | threshold_breached:high |    False
|  False   |

cortxcli$
```

Using JSON API output

The simplest mechanism to handle JSON output is by using either a JavaScript or a Python parser to interpret the data.

JSON objects can also be embedded inside of other objects and are always presented as an array as well. The JSON output always uses a hierarchical presentation of objects to identify relationships between objects.

Each object also has an object-name property that may be used in some cases to identify the object uniquely.

Command syntax

General rules for specifying commands

- Parameters enclosed in square brackets ([ ]) are optional. Do not type the bracket characters.
- Parameter values separated by '|' characters are options. Enter only one of the values. Unless specified otherwise, enumerated values are not case sensitive.
- Parameter values in angular brackets (< >) are variables. Substitute text that is appropriate for the task you want to perform. Unless specified otherwise, variable values such as names of users and volumes are case sensitive and have a maximum length in bytes. When encoded in UTF-8, a single character can occupy multiple bytes. Typically:
    - 1 byte per character for English, Dutch, French, German, Italian, and Spanish
    - 3 bytes per character for Chinese, Japanese, and Korean

User password rules

- The value is case sensitive.
- The value can have 8–32 characters.
- The value can include printable UTF-8 characters except a space or: " ' , < > \
- A value that includes only printable ASCII characters must include at least one uppercase character, one lowercase character, one numeric character, and one non-alphanumeric character. This rule does not apply if the password contains UTF-8 characters that are outside the range of printable ASCII characters.

Categorical list of commands

The following table helps you find a command within a category of functionally related commands. A command might appear in more than one category.

Table 1: Commands by category

Category	Commands
Alerts	alerts show, alerts acknowledge, alerts comment show, alerts comment add, alerts history
S3 accounts	s3accounts show, s3accounts create, s3accounts reset_password, s3accounts delete
S3 buckets	s3bucket show, s3bucket create, s3bucket delete
S3 iam users	s3iamuser show, s3iamuser create, s3iamuser delete
Users	users show, users create, users delete
System	system stop, system shutdown, system start, system status, system startup
S3 bucket policy	s3bucketpolicy show, s3bucketpolicy create, s3bucketpolicy delete
Support_bundle	support_bundle generate, support_bundle status

Each command topic includes one or more of the following sections:

- **Description** The command's purpose and notes about its usage
- **Minimum role** The minimum user role required to use the command
- **Syntax** The command's syntax
- **Parameters** Descriptions of the command's parameters

- **Output** Descriptions of fields shown in console format
- **Examples** One or more examples of the command's usage in console format
- **Basetypes** References to descriptions of basetype properties shown in API format

Help section of a command

Every command has a common parameter to display the help section of the command. Help section gives information about command syntax and its associated parameters.

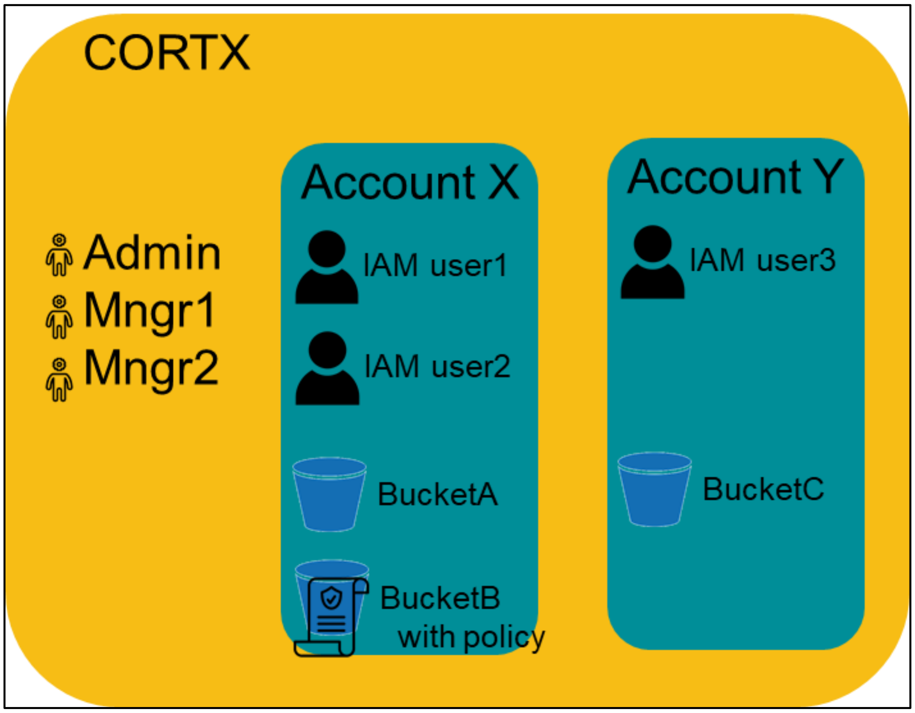
Use following syntax to display help section of the command –

```
<command name> -h
```

5 | User administration

You can manage the following types of users through CORTX Manager:

- Admin user: You can create the admin user only once during setting up the system. The admin user has all the permissions in the system. For more information, refer the Lyve Rack Installation Guide.
- Local user: The local user has two roles – Manage and Monitor.
- S3 user: The S3 user is used to access the Lyve Pilot.
- IAM user: The IAM user is used to access the Lyve Pilot.

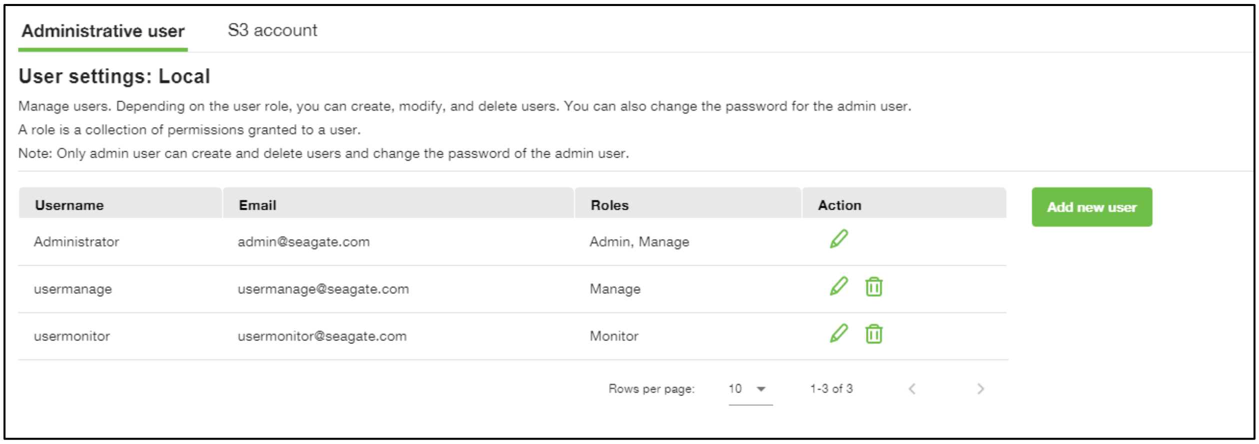


5.1 | Managing local users

Viewing users using CORTX Manager

To view users using CORTX Manager:

- Click **Manage** to view all the existing users.



User settings: Local

Manage users. Depending on the user role, you can create, modify, and delete users. You can also change the password for the admin user.
A role is a collection of permissions granted to a user.
Note: Only admin user can create and delete users and change the password of the admin user.

Username	Email	Roles	Action
Administrator	admin@seagate.com	Admin, Manage	✎
usermanage	usermanage@seagate.com	Manage	✎ 🗑
usermonitor	usermonitor@seagate.com	Monitor	✎ 🗑

Viewing users using CLI

Description

Shows list of all CORTX Manager users. Both; administrator and users can run this command.

Minimum role: monitor

Syntax

```
users show
[-d desc|asc]
[-f table|xml|json]
[-l <LIMIT>]
[-o <offset>]
[-s user_id|user_type|created_time|updated_time]
```

Parameters

`[-d desc|asc]`

Optional. Indicates “sort direction” parameter. Specifies either descending or ascending order of the users list.

Only following values are applicable to this parameter:

- desc: displays users in descending order.
- asc: displays users in ascending order. By default, the system displays users in this format.

`[-f table|xml|json]`

Optional. Specifies the output format type. Default value is table.

`[-o <offset>]`

Optional. Offset is used for defining the index from which you want to see the users.

```
[-l <LIMIT>]
```

Optional. Specifies number of users you would like to see.

```
[-s user_id|user_type|created_time|updated_time]
```

Optional. Indicates "Sort by" parameter. Specifies option by which you can sort the users list.

Only following values are applicable to this parameter:

- user_id: displays the users sorted according to their user IDs.

Note

Username and user_id is same. In the output, the username is displayed instead of user_id.

- user_type: displays the users sorted according to the user type.
- created_time: displays the users sorted according to the time of the account creation.
- updated_time: displays the users sorted according to the time of the account updated.

Output

Property name as per format	Property Type	Description
User name Table: Username xml: username json: username	String	Specifies name of the CORTX Manager user.
User Type Table: User Type xml: user_type json: user_type	String	Specifies type of the CORTX Manager user. For example: CORTX Manager
Roles Table: Roles xml: roles json: roles	String	Specifies role of the user. For example: manage or monitor.
Creation Time Table: Creation time xml: created_time json: created_time	DateTime	Specifies time of the CORTX Manager user account generation.
Last update time Table: Last update time xml: updated_time json: updated_time	DateTime	Represents the time at which the CORTX Manager user account details were updated.

Examples

To display existing users, run the following command.

```
users show
```

To display the users in descending order using “user_id” as the sorting method, run the following command.

```
users show –s user_id –d desc
```

To display the last 10 users from the chosen 20 users, run the following command.

```
users show -l 20 -s created_time -d asc -o 10
```

Creating local users from the CORTX Manager

You can assign the following roles for local users:

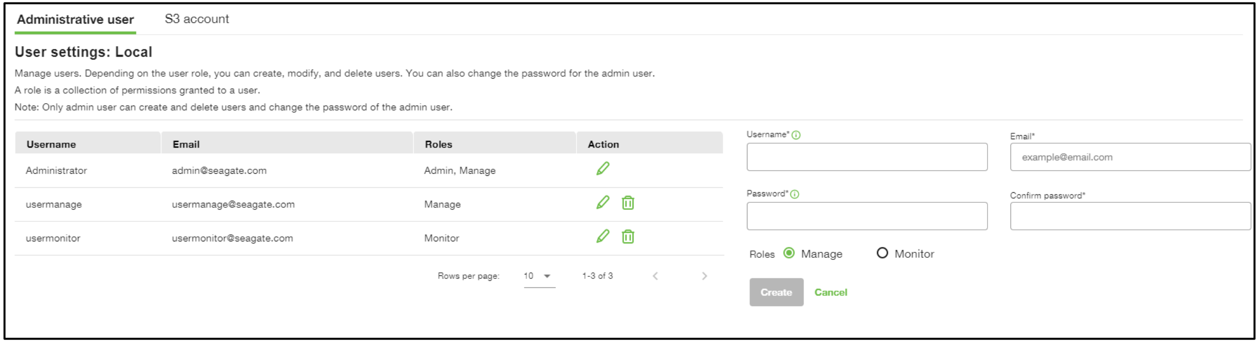
Manage: The users with Manage role can access all pages but cannot:

- Modify or delete users
- Modify or delete S3 accounts
- Access the Settings page
- Access the Lyve Pilot page
- Access System maintenance, Firmware update, and System update pages.

Monitor: The users with Monitor role can only access dashboard, Health, Audit log, About, and can only view users and S3 accounts. Users with Monitor can only modify the user with which they are logged in.

To create a local user:

1. Click **Manage → Add new user**.



User settings: Local

Manage users. Depending on the user role, you can create, modify, and delete users. You can also change the password for the admin user.

A role is a collection of permissions granted to a user.

Note: Only admin user can create and delete users and change the password of the admin user.

Username	Email	Roles	Action
Administrator	admin@seagate.com	Admin, Manage	✎
usermanage	usermanage@seagate.com	Manage	✎ 🗑
usermonitor	usermonitor@seagate.com	Monitor	✎ 🗑

Rows per page: 10 1-3 of 3

Username* ⓘ

Email* example@email.com

Password* ⓘ

Confirm password*

Roles ● Manage ○ Monitor

2. Enter unique username in the **Username** field.
3. Enter **Password** and **Confirm Password**.
   The password must contain at least 8 characters and must be a combination of one upper case, one lower case, one special character, and a numeric character.
4. Select a role for the user. You may select either Manage or Monitor.
5. Click **Create** to create the local user.

Creating users using CLI

users create

Description

Creates a CORTX Manager user. Both; administrator and users can run this command.

Minimum role: manage

Syntax

```
users create <username> <email> monitor|manage
```

Parameters

`<username>`

Required. Specifies name of the user. The length of the username must be between 4 through 64 characters and can contain alphanumeric, "-", and "_".

`<email>`

Required. Specifies an email address to the new user.

monitor|manage

Required. Specifies the role of the new user. One of the following roles can be assigned to the user.

- Monitor: users have viewing permission only.
- Manage: users have editing permission.

Output

Confirmation message is displayed.

Examples

To create a new user with name "mndr" having "mndr@acme.com" as the email address, and having editing access, run the following command.

```
users create mndr mndr@acem.com manage
```

Type "y" and then press ENTER to confirm account creation.

Modifying local user using CORTX Manager

To modify local user using CORTX Manager

1. Click **Manage**. Under the **Administrative user** tab, click ✏ to modify the user.
2. Make the required changes to the user.
3. Click **Save** to save the changes made to the user

Modify users using CLI

users reset_password

Description

Used to change password of a CORTX manager user’s account.

Minimum role manage

Syntax

```
users reset_password <username>
```

Parameters

```
<username>
```

Required. Specifies username of the CORTX manager user.

Output

Password updated.

Examples

To change the password of an existing CORTX manager user, run the following command.

```
users reset_password mndr
```

users update

Description

Used to update CORTX manager user’s information.

Minimum role: manage

Syntax

```
users update
user_id [-e] [-r]
```

Parameters

```
user_id
```

Required. Specifies username of the CORTX manager user.

`-e`

Optional. `–e` indicates email of the user. You can use this to change email address of the user.

`-r`

Optional. `–r` indicates role of the user. You can use this to change role of the user. Role can be either “manage” or “monitor”.

Output

User information updated.

Examples

To change the existing role of a CORTX manager user jadmith from monitor to manage, run the following command.

```
users update jadmith -r manage
```

Deleting local users using CORTX Manager

1. Click **Manage,** and then select the user you want to delete.
2. Click 🗑, and then click **Yes** to delete the selected user.

Deleting local users using CLI

users delete

Description

Deletes the specified CORTX Manager user. Both; administrator and users can run this command.

Minimum role: manage

Syntax

```
users delete <username>
```

Parameters

`<username>`

Required. Specifies name of the user to be deleted from the CORTX Manager system.

Output

Confirmation message is displayed.

Examples

To delete an existing mndr user, run the following command.

```
users delete mndr
```

5.2 | Managing S3 accounts

The S3 account is used to access the Lyve Pilot. Users with Admin privileges can create S3 accounts.

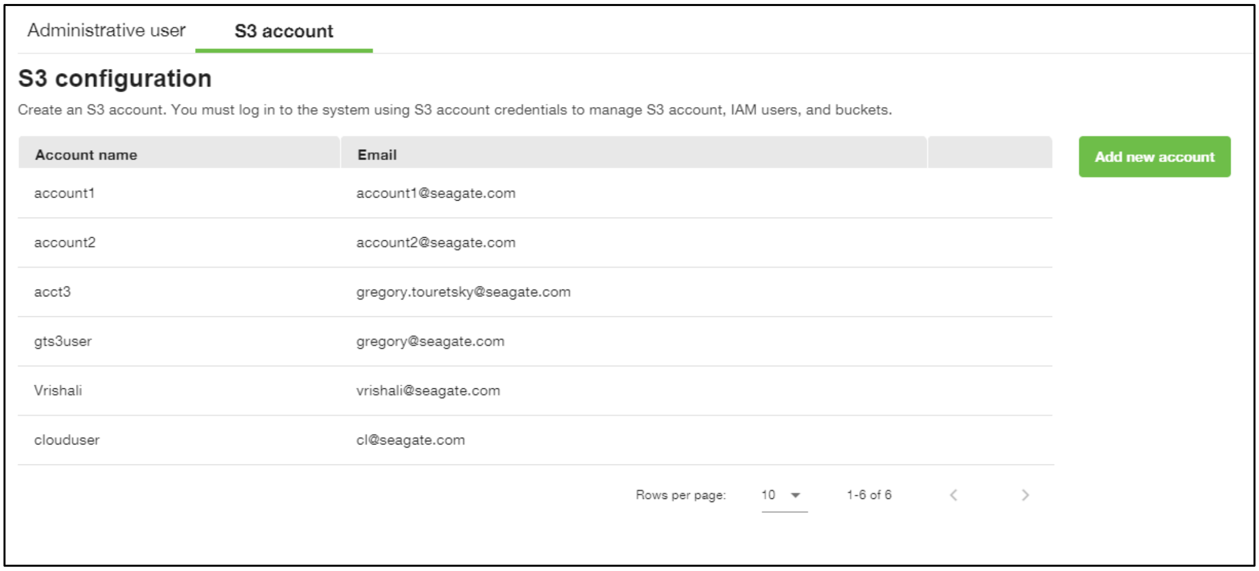
After creating the S3 account you can log in and perform the following:

- View the S3 accounts
- Edit the S3 account
- Delete the S3 account
- Copy S3 URL
    - Click [icon] to copy the S3 URL.
- Generate or delete access keys for a S3 account
    - Click on the S3 account for which you want to generate the access key
    - Click Add/Generate to generate the access key.
    - Click [icon] associated with the access key to delete the access key.
- Create IAM user
- Create bucket

Viewing S3 accounts using CORTX Manager

To view S3 accounts using CORTX Manager:

- Click **Manage,** and then click the **S3 account** tab to view the existing S3 accounts.



S3 configuration

Create an S3 account. You must log in to the system using S3 account credentials to manage S3 account, IAM users, and buckets.

Account name	Email
account1	account1@seagate.com
account2	account2@seagate.com
acct3	gregory.touretsky@seagate.com
gts3user	gregory@seagate.com
Vrishali	vrishali@seagate.com
clouduser	cl@seagate.com

Rows per page: 10 1-6 of 6

Viewing S3 account using CLI

Description

Shows account names and email addresses of all S3 accounts.

Admin user can see all S3 accounts whereas, S3 account user can see the details of only his/her account.

Minimum role: monitor

Syntax

```
s3accounts show
[-f table|xml|json]
```

Parameters

```
[-f table|xml|json]
```

Optional. Specifies the output format type. Default value is table.

Output

Property name as per format	Property Type	Description
Account Name Table: Account Name xml: account_name json: account_name	String	Specifies name of the s3 account holder.
Account Email Table: Account Email xml: account_email	String	Specifies email address of the s3 account holder.

Property name as per format	Property Type	Description
json: account_email		

Examples

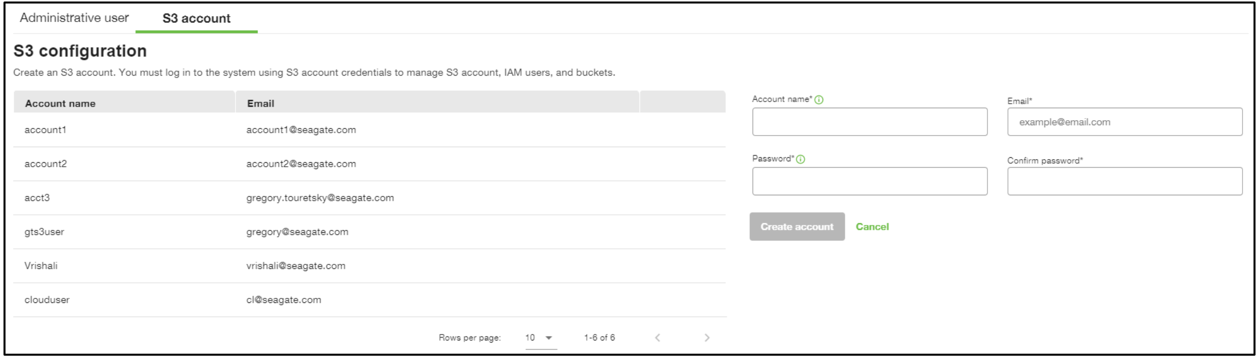
To display all active s3 accounts in tabular format, run the following command.

```
s3accounts show
```

Creating S3 account using CORTX Manager

To create S3 account using CORTX Manager:

1. Click **Manage**, and then click the **S3 account** tab.



S3 configuration

Create an S3 account. You must log in to the system using S3 account credentials to manage S3 account, IAM users, and buckets.

Account name	Email
account1	account1@seagate.com
account2	account2@seagate.com
acct3	gregory.touretsky@seagate.com
gts3user	gregory@seagate.com
Vrishali	vrishali@seagate.com
clouduser	cl@seagate.com

Rows per page: 10 1-6 of 6

Account name*

Email* example@email.com

Password*

Confirm password*

2. Click **Add new account**.
3. Enter unique username in the **Username** field.
4. Enter the email address in the **Email** field.
5. Enter **Password** and **Confirm Password**.
   The password must contain at least 8 characters and must be a combination of one upper case, one lower case, one special character, and a numeric character.
6. Click **Create account** to create the S3 account.
7. Download the account information for later use.

Creating S3 accounts using CLI

s3accounts create

Description

Creates a new S3 account. Only admin user can create S3 accounts.

Minimum role: manage

Syntax

```
s3accounts create <account_name> <account_email>
```

Parameters

`<account_name>`

Required. Specifies a name to the new S3 account that is being created.

`<account_email>`

Required. Specifies an email address to the new S3 account.

Output

Property name as per format	Property Type	Description
Account Name Table: Account Name xml: account_name json: account_name	String	Specifies name of the s3 account holder.
Account Email Table: Account Email xml: account_email json: account_email	String	Specifies email address of the s3 account holder.
Permanent Access Key Table: Permanent Access Key xml: permanent_access_key json: permanent_access_key	String	Specifies unique key used to communicate with Amazon S3.
Permanent Secret Key Table: Permanent Secret Key xml: permanent_secret_key json: permanent_secret_key	String	Specifies unique key used to communicate with Amazon S3 using APIs.

Examples

To create an s3 account with name "vahgar" having email address as "vahgar1808@ihsoj.com", run the following command.

```
s3accounts create vahgar vahgar1808@ihsoj.com
```

Create a password for the new S3 account using the following rules.

The password must be minimum 8 characters with at least 1 lowercase, 1 uppercase, 1 numeric, and 1 special character.

Type "y" and then press ENTER to confirm account creation.

Modifying S3 accounts using CORTX Manager

1. Log in to CORTX Manager using the S3 account credentials.
2. Under the **S3 account** tab, click ✎ to modify the S3 account.

3. Click **Save** to save the changes made to the user.

Modifying S3 accounts using CLI

s3accounts reset_password

Description

Resets password of the specified S3 account. Only S3 account users can reset the password of their own S3 account.

Minimum role: manage

Syntax

```
s3accounts reset_password <account_name>
```

Parameters

```
<account_name>
```

Required. It is used to indicate the name of the S3 account of which the password has to be reset.

Output

Property name as per format	Property Type	Description
Account Name Table: Account Name xml: account_name json: account_name	string	Specifies name of the S3 account holder whose password needs to be reset.

Examples

To reset the password of an S3 account (shweni), run the following command.

```
s3accounts reset_password shweni
```

Type a new password. Confirm the password.

Type “y” and then press ENTER to confirm password change.

Deleting S3 accounts using CORTX Manager

1. Log in to CORTX Manager using the S3 account credentials.

2. Click 🗑 associated with the user to delete the S3 account.

Deleting S3 accounts using CLI

s3accounts delete

Description

Deletes the specified S3 account. Only S3 account users can delete their own S3 accounts.

All S3 buckets and S3 IAM users present in an S3 account must be deleted in order to delete the S3 account.

Minimum role: manage

Syntax

```
s3accounts delete <account_name>
```

Parameters

`<account_name>`

Required. Specifies name of the S3 account to be deleted.

Output

Confirmation message is displayed.

Examples

To delete an S3 account (shweni), run the following command.

```
s3accounts delete shweni
```

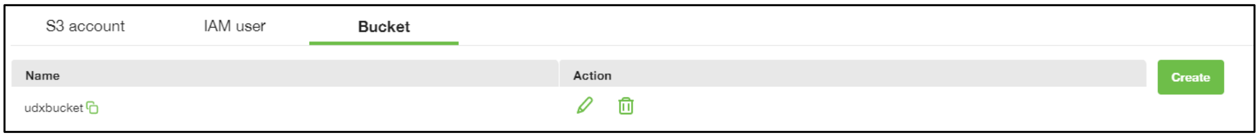
Type “y” and then press ENTER to confirm account deletion.

5.3 | Managing Buckets

Viewing buckets using CORTX Manager

1. You must log in to CORTX Manager using the S3 account credentials.

2. Click the **bucket** tab to view the buckets.



Viewing buckets using CLI

s3buckets show

Description

Shows all available S3 buckets. Only S3 account users can run this command.

Minimum role: monitor

Syntax

```
s3buckets show
[-f table|xml|json]
```

Parameters

```
[-f table|xml|json]
```

Optional. Specifies the output format type. Default value is table.

Output

Property name as per format	Property Type	Description
Bucket Name Table: Bucket Name xml: bucket_name json: bucket_name	String	Specifies name of the S3 bucket.

Examples

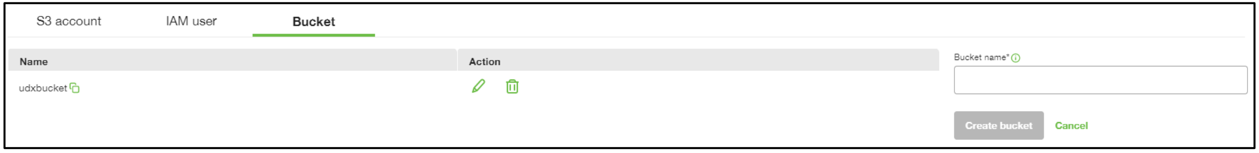
To show all available S3 buckets in XML format, run the following command.

```
s3bucket show –f xml
```

Creating a bucket using CORTX Manager

1. You must log in to CORTX Manager using the S3 account credentials.

2. On the **bucket** tab, click **Create**.



Name	Action
udxbucket	✎ 🗑

3. Enter a name for the bucket, and then click **Create** bucket.
4. Copy the bucket URL, and then click **Ok**.

Creating a bucket using CLI

s3buckets create

Description

Creates new S3 bucket. Only S3 account user can run this command.

Minimum role: manage

Syntax

```
s3buckets create <bucket_name>
```

Parameters

```
<bucket_name>
```

Required. Used to provide a new S3 bucket name.

Output

Confirmation message is displayed.

Examples

To create a new S3 bucket having (jadmith) as its name, run the following command.

```
s3buckets create jadmith
```

Editing a bucket policy using CORTX Manager

1. Click ✎ associated with the bucket you want to modify.
2. Enter a new JSON policy in the text box.
3. Click **Update** to save the changes.

Deleting a bucket using CORTX Manager

- Click 🗑 associated with the user to delete the bucket.

Deleting a bucket using CLI

s3buckets delete

Description

Deletes the specified S3 bucket. Only S3 account user can run this command.

Minimum role: manage

Syntax

```
s3buckets delete <bucket_name>
```

Parameters

```
<bucket_name>
```

Required. Specifies the name of the S3 bucket to be deleted.

Output

Confirmation message is displayed.

Examples

To delete s3 bucket having (jadmith) as its name, run the following command.

```
s3buckets delete jadmith
```

Viewing S3 bucket policy using CLI

s3bucketpolicy show

Description

Shows policy of the specified bucket.

Minimum role: S3 account user

Syntax

```
s3bucketpolicy show <bucket_name>
[-f xml|json]
```

Parameters

```
<bucket_name>
```

Required. Specifies name of the bucket of which you would like to see the policy.

```
[-f xml|json]
```

Optional. Specifies the output format. Default value is JSON.

Output

CLI console displays bucket policy details. If the bucket has no policy then console shows an error message.

Examples

To show policy of S3 bucket having a name as (jadmith), run the following command.

```
s3bucketpolicy show jadmith
```

Creating S3 bucket policy using CLI

s3bucketpolicy create

Description

Creates a new or replaces an existing policy of the specified bucket. Only an S3 account user can run this command.

Minimum role: S3 account user

Syntax

```
s3bucketpolicy create <bucket_name> <id> <statement> <version>
```

Parameters

```
<bucket_name>
```

Required. Specifies name of the bucket of which you would like to create a new or replace an existing policy.

```
<id>
```

Required. Specifies id of the new policy which you want to create or replace an old one.

```
<statement>
```

Required. Specifies path to the file.

```
<version>
```

Required. Specifies policy version. Default value is “10/17/2012”

Output

Confirmation message is displayed.

Examples

To create a policy for s3 bucket named jadmith, and having following values,

- Id: policyID1
- Statement: /admin/policy.json
- Version: 2012-10-18

run the following command.

```
s3bucketpolicy create jadmith policyID1 /admin/policy.json 2012-10-18
```

Type "y" and then press ENTER to confirm S3 bucket policy creation.

Deleting S3 bucket policy using CLI

s3bucketpolicy delete

Description

Deletes an existing policy of the specified bucket.

Minimum role: S3 account user

Syntax

```
s3bucketpolicy delete <bucket_name>
```

Parameters

```
<bucket_name>
```

Required. Specifies name of the bucket of which you would like to delete the policy.

Output

Confirmation message is displayed.

Examples

To delete an existing bucket policy of s3 bucket named jadmith, run the following command.

```
s3bucketpolicy delete jadmith
```

5.4 | Bucket policy examples

Following are some bucket policy examples which you can use.

Policy to allow an account access to List objects

```
[
    {
      "Sid": "Stmt1462526862401",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::raghav.bucket",
      "Principal": {
"CanonicalUser":"21ecf5c623762f3d59d4abc8d92bca25d021a3c31594eb387c8dfee82661
cba8" }
    }
]
```

Policy to allow an account access to List objects, but only those which start with prefix projects

```
[
    {
      "Sid": "Stmt1462526862401",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::raghav.bucket",
        "Condition": {
        "StringEquals": {
          "s3:prefix": "projects"
        }
      },
      "Principal": {
"CanonicalUser":"21ecf5c623762f3d59d4abc8d92bca25d021a3c31594eb387c8dfee82661
cba8" }
    }
]
```

Policy to allow put object by any cross account IAM user with the help of ARN

```
[
      {
         "Effect": "Allow",
         "Principal": {"AWS":"arn:aws:iam::account-number-without-
hyphens:user/username"},
         "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::raghav.bucket/*"
      }
]
```

Policy with multiple ARNs for principal

```
[
      {
         "Effect": "Allow",
         "Principal": {
                "AWS":["arn:aws:iam::account-number-without-
hyphens:user/username","arn:aws:iam::account-number-without-hyphens:root"]
                },
         "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::raghav.bucket/*"
      }
]
```

Policy to allow a put object to IAM user but deny delete object inside the bucket

```
[
      {
         "Effect": "Allow",
         "Principal": {"AWS":"arn:aws:iam::account-number-without-
hyphens:user/username"},
         "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::raghav.bucket"
      },
      {
         "Effect": "Deny",
         "Principal": {"AWS":"arn:aws:iam::account-number-without-
hyphens:user/username"},
         "Action": "s3:DeleteObject",
         "Resource": "arn:aws:s3:::raghav.bucket/*"
      }
]
```

Allow PutObject to a user only if the user provides a bucket-owner-read on the object

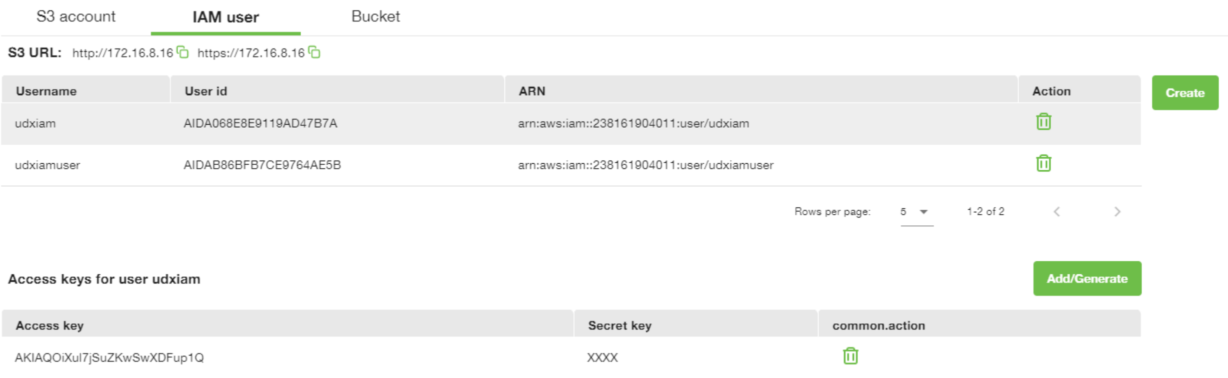
```
[
    {
      "Sid": "Stmt1462526862401",
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::raghav.bucket/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-read"
        }
      },
      "Principal":{"AWS":"arn:aws:iam::account-number-without-
hyphens:user/username"}
    }
]
```

5.5 | Managing IAM users

You must log in to CORTX Manager using the S3 account credentials.

You can:

- View IAM users
- Create IAM user
- Modify IAM user
- Delete IAM user
- Copy S3 URL
  - Click ⧉ to copy the S3 URL.
- Add or generate access keys for the IAM user
  - Click on the IAM user for which you want to generate the access key.
  - Click **Add/Generate** to generate the access key.
- Delete access keys for the IAM user
  - Click 🗑 associated with the access key to delete the access key



S3 URL: http://172.16.8.16 ⧉ https://172.16.8.16 ⧉

Username	User id	ARN	Action
udxiam	AIDA068E8E9119AD47B7A	arn:aws:iam::238161904011:user/udxiam	🗑
udxiamuser	AIDAB86BFB7CE9764AE5B	arn:aws:iam::238161904011:user/udxiamuser	🗑

Rows per page: 5 ▾ 1-2 of 2 ‹ ›

Access keys for user udxiam

Access key	Secret key	common.action
AKIAQOiXul7jSuZKwSwXDFup1Q	XXXX	🗑

Viewing IAM users using CLI

s3iamusers show

Description

Shows all S3 iam users available in the logged-in S3 account. Only S3 account user can run this command.

Minimum role: **monitor**

Syntax

```
s3iamusers show
[-f table|xml|json]
```

Parameters

```
[-f table|xml|json]
```

Optional. Specifies the output format type. Default value is table.

Output

Property name as per format	Property Type	Description
User Name Table: User Name xml: user_name json: user_name	String	Specifies name of the S3 iam user.
User ID Table: User ID xml: user_id json: user_id	String	Specifies user ID of the S3 iam user.
ARN Table: ARN xml: arn json: arn	String	Specifies Amazon Resource Number. It is a resource name of the resource present in the S3 protocol such as S3 iam user.

Examples

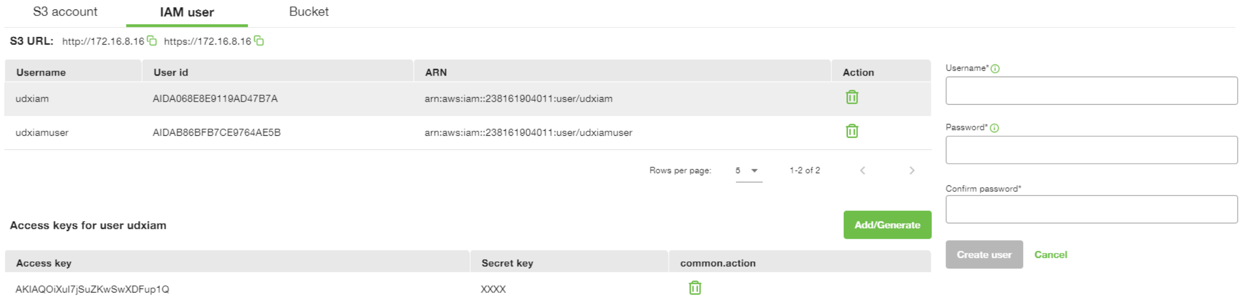
To see the S3 iam users of an S3 account (shweni) in JSON format,

Shweni must log into the CORTX Manager using his S3 account login credentials and then must run the following command.

```
s3iamusers show –f [json]
```

Creating IAM users using CORTX Manager

1. On the **IAM user** tab, click **Create**.



S3 URL: http://172.16.8.16 https://172.16.8.16

Username	User id	ARN	Action
udxiam	AIDA068E8E9119AD47B7A	arn:aws:iam::238161904011:user/udxiam	🗑
udxiamuser	AIDAB86BFB7CE9764AE5B	arn:aws:iam::238161904011:user/udxiamuser	🗑

Access keys for user udxiam

Access key	Secret key	common.action
AKIAQOiXul7jSuZKwSwXDFup1Q	XXXX	🗑

2. Enter username in the **Username** field.

3. Enter **Password** and **Confirm Password**.
   The password must contain at least 8 characters and must be a combination of one upper case, one lower case, one special character, and a numeric character.
4. Click **Create user** to create the user.
5. Click **Download and close** to download the account information like access key and secret key for later use.

Creating IAM user using CLI

s3iamusers create

Description

Creates an S3 iam user account. Only S3 account user can run this command.

Minimum role: **manage**

Syntax

```
s3iamusers create <user_name>
```

Parameters

```
<user_name>
```

Required. Specifies name of the S3 iam user.

Output

Property name as per format	Property Type	Description
User Name Table: User Name xml: username json: username	String	Specifies name of the S3 iam user.
User ID Table: User ID xml: user_id json: user_id	String	Specifies user ID of the S3 iam user.
ARN Table: ARN xml: arn json: arn	String	Specifies Amazon Resource Number. It is a resource name of the resource present in the S3 protocol such as S3 iam user.

Examples

To create an S3 iam user with (amarta) name, run the following command.

```
s3iamusers create amarta
```

Create a password for the new S3 iam account using the following rules.

The password must be minimum 8 characters with at least 1 lowercase, 1 uppercase, 1 numeric, and 1 special character.

Type "y" and then press ENTER to confirm account creation.

Deleting IAM users using CORTX Manager

- Click 🗑 associated with the user to delete the IAM user.

Deleting IAM users using CLI

s3iamusers delete

Description

Deletes the specified S3 iam user account. Only S3 account user can run this command.

Minimum role: **manage**

Syntax

```
s3iamusers delete <user_name>
```

Parameters

```
<user_name>
```

Required. Specifies account name of the S3 iam user to be deleted.

Output

Confirmation message is displayed.

Examples

To delete S3 iam user (amarta), run the following command.

```
s3iamusers delete amarta
```

6 | Lyve Pilot registration

Lyve Pilot is Seagate's data management software that gives users the ability to securely move data from endpoint to edge to core.

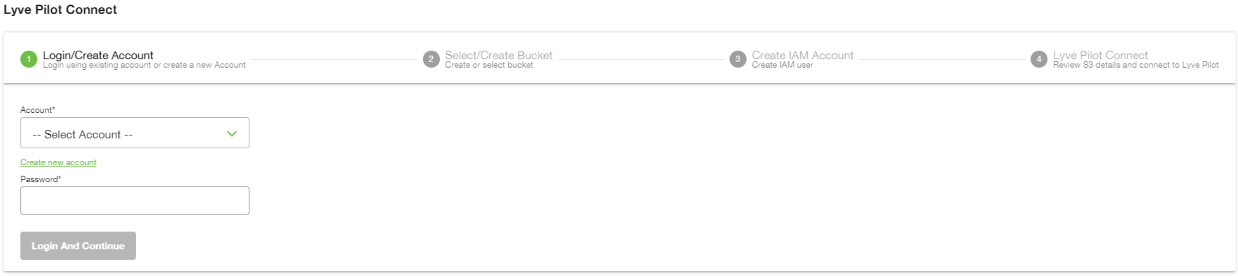
Lyve Pilot brings Seagate technologies together into an ecosystem for data management from endpoint to edge to core, enabling customer use cases for IT 4.0. IoT devices generate much more data than current and near-term Internet capability can transport. Computing resources and storage are moving to the edge to process this data. As data is processed at the edge and needs to move to private or public clouds, portable shuttles might be required to efficiently move this processed data.

With Lyve Pilot, you can import data from endpoint devices. During import, the system generates metadata, which includes fingerprint and provenance information to help route the data. Lyve Pilot uses the metadata to coordinate data movement and storage in the customer’s private cloud. Using Seagate Secure devices as the foundation, data is secured at rest and, using TLS connections, the data is secured in flight.

Lyve Pilot pulls together Seagate technologies into a common user experience and full technology portfolio for data management.

To register Lyve Pilot:

1. Click **Lyve Pilot** to open the **Lyve Pilot Registration** page.



Lyve Pilot Connect

1 Login/Create Account
Login using existing account or create a new Account

2 Select/Create Bucket
Create or select bucket

3 Create IAM Account
Create IAM user

4 Lyve Pilot Connect
Review S3 details and connect to Lyve Pilot

Account*
-- Select Account --
Create new account
Password*
Login And Continue

2. Select the S3 account from the dropdown list and enter the appropriate password for the selected S3 account.
   If an existing S3 account is not available, click **Create new account** to create a new S3 account. For more information on how to create a new S3 account, see Creating S3 account using CORTX Manager.
3. In the **Select Bucket** list, select a bucket.
   If there are no buckets, then click **Create new Bucket** to create a new bucket. For more information on how to create a new bucket, see Creating a bucket using CORTX Manager.
4. After selecting or creating a bucket, you must create an IAM account.
5. To create an IAM account:
   a. Enter username in the **Username** field.
   b. Enter **Password** and **Confirm Password**.
      The password must contain at least 8 characters and must be a combination of one upper case, one lower case, one special character, and a numeric character.
   c. Click **Create** to create the IAM account.
6. On the **Registration** page, copy the registration token. Use the registration token to add a device on the Lyve Pilot portal.
7. In the **URL** field, enter the URL provided by the Lyve Pilot portal.
8. In the **PIN** field, enter the PIN provided by the Lyve Pilot portal

9. Select the check boxes, and then click **Register** to complete your Lyve Pilot registration.

7 | Creating and uploading SSL certificate

An SSL certificate is used on a https connection to encrypt the communication from a S3 Client or your web browser to CORTX Manager. By default, CORTX Manager uses a CORTX Manager provided self-signed certificate. Alternatively, you can upload a user-provided self-signed certificate or a user provided certificate authority (CA) signed certificate. This step can be done during onboarding or afterwards.

To create SSL certificate:

I. Verify openssl is installed on the system. Run the following command to verify openssl.

```
$ openssl version
```

II. Generate self-signed certificate and private key for your server

```
$ openssl req -newkey rsa:2048 -keyout domain_srv.key -nodes -x509 -days 365
-out domain_srv.crt
```

Please, carefully fill in requested fields after executing command:

```
Country Name - e.g. US
State or Province Name - e.g. Colorado
Locality Name - e.g. Denver
Organization Name - e.g. Seagate
Organizational Unit Name – e.g. Support
Common Name - Domain or group of sub-domains for which certificate is issued.
Email Address - e.g. test@mail.com
```

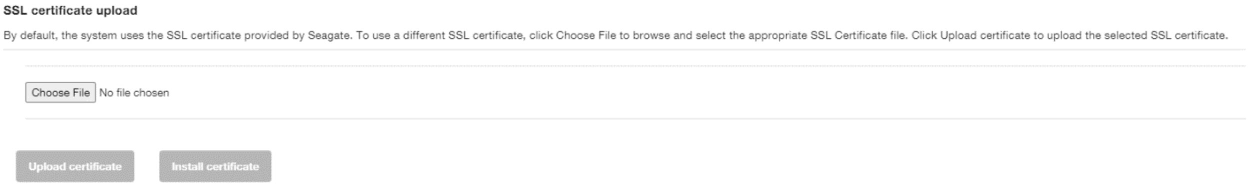
1. Run the following command.

   ```
   $ cat domain_srv.crt domain_srv.key > certificate.pem
   ```

A“certificate.pem” file is generated which can be uploaded by the admin user.

To upload SSL certificate:

1. On the CORTX manager, log in with Admin credentials, and then click **Settings → SSL Certificate**



SSL certificate upload

By default, the system uses the SSL certificate provided by Seagate. To use a different SSL certificate, click Choose File to browse and select the appropriate SSL Certificate file. Click Upload certificate to upload the selected SSL certificate.

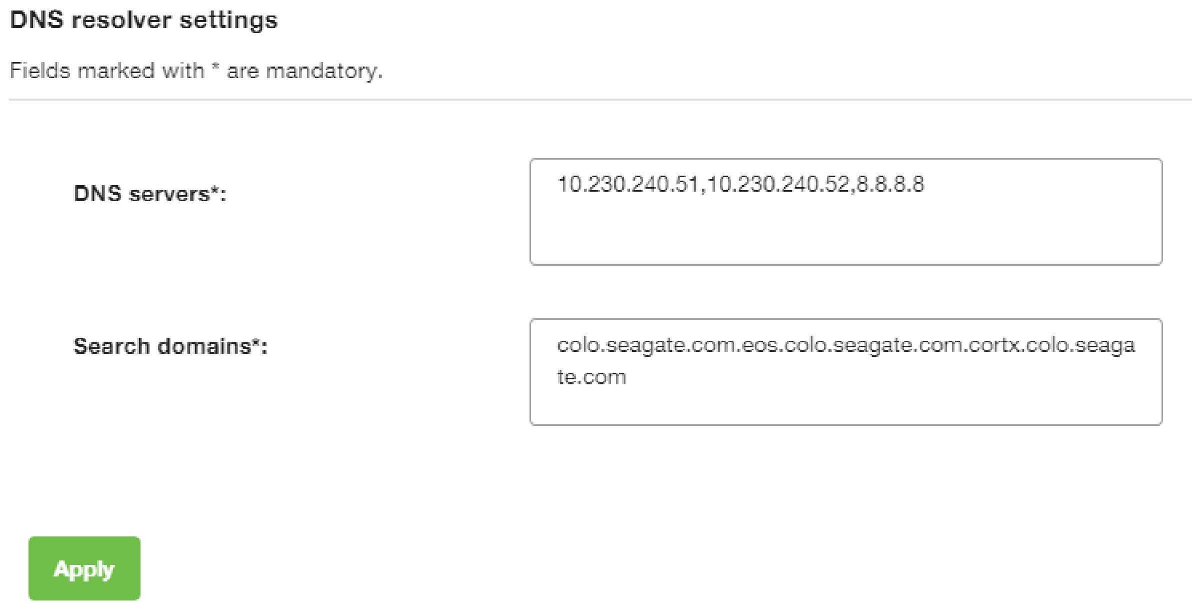
2. Click **Choose File** to browse and select the appropriate .pem SSL certificate file, and then click **Upload certificate**.

3. Click **Yes** to install the SSL certificate.
   After installing the SSL certificate, you must log out and log in as Admin user.
4. Click **Settings** → **SSL certificate** to view the installed SSL certificate.

8 | Configuring DNS resolver settings

To configure DNS resolver settings:

1. On the **DNS resolver settings** page, enter values for **DNS Server** and **Search Domain**.



DNS resolver settings

Fields marked with * are mandatory.

DNS servers*: 10.230.240.51,10.230.240.52,8.8.8.8

Search domains*: colo.seagate.com.eos.colo.seagate.com.cortx.colo.seagate.com

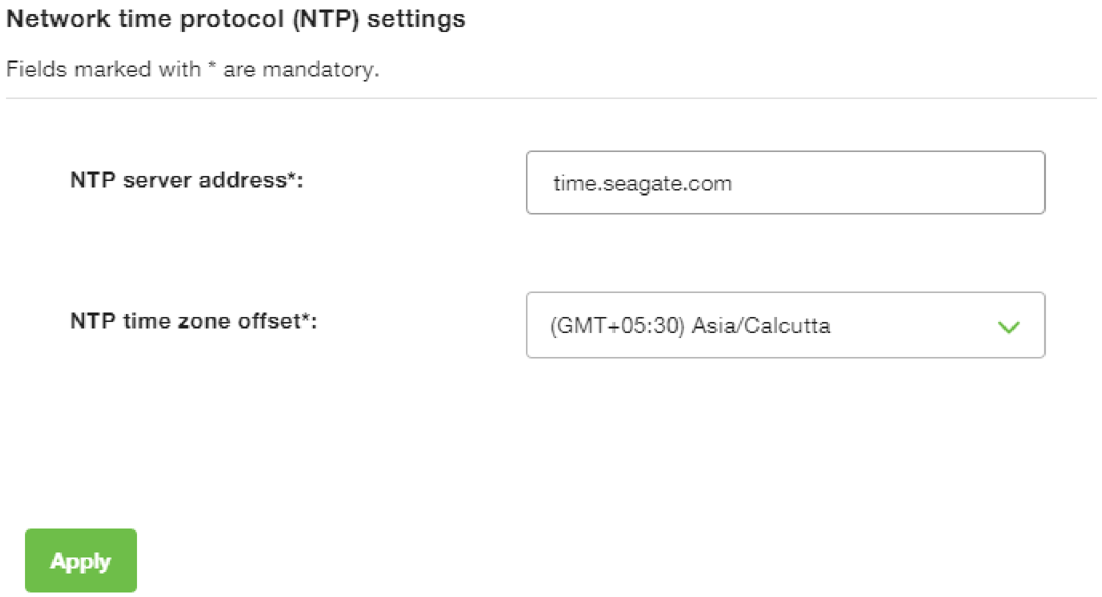
2. Click **Apply** to save the changes.

9 | Configuring network time protocol

Lyve Rack and any S3 Clients must be time synchronized via an NTP server. CORTX Manager allows the setting of the NTP server address and a time zone. The time zone on CORTX Manager does not have to match the S3 Client(s). Once the CORTX Manager setting is applied, the setting is then configured on both servers in Lyve Rack.

To configure network time protocol:

1. On the **Network time protocol (NTP)** page, enter NTP server address and select the time zone. The selected time zone is used by the system.



Network time protocol (NTP) settings

Fields marked with * are mandatory.

NTP server address*: time.seagate.com

NTP time zone offset*: (GMT+05:30) Asia/Calcutta

2. Click **Apply** to save the changes.

10 | Configuring notifications

The system offers you to configure notifications. You can configure the system to receive notification via email using the Simple Network Management Protocol (SNMP). Once configured, you can receive notifications about any system updates or alerts. You have an option to skip configuring the notifications, but it is not recommended. It is recommended to configure at least one email to receive system notification.

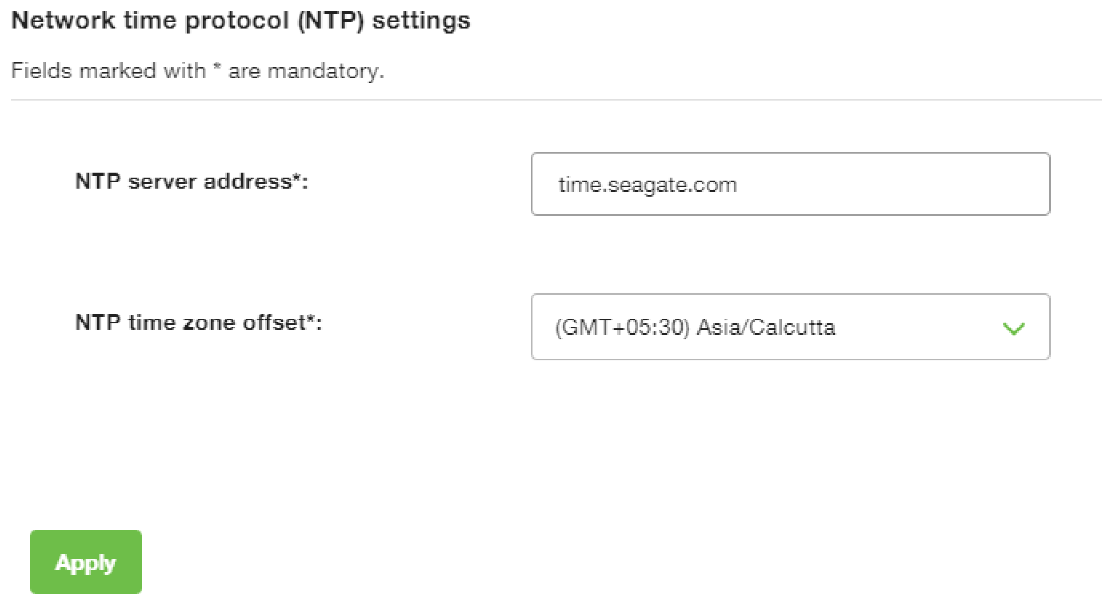
Table 2: Supported and unsupported email configurations lists the supported and unsupported email configurations.

Table 2: Supported and unsupported email configurations

Type	Supported/Unsupported
By encryption:	
No encryption	Supported
SSL/TLS	Supported
STARTTLS	Supported
By authentication:	
SMTP servers which support/require authentication	Supported
SMTP servers which do not support authentication	Not supported

To configure notifications:

1. On the **Notifications** page, select the **Email** check box, and then click **Continue**.



Network time protocol (NTP) settings

Fields marked with * are mandatory.

NTP server address*: time.seagate.com

NTP time zone offset*: (GMT+05:30) Asia/Calcutta

2. Enter values for **SMTP server**, **Sender email**, **Protocol**, **SMTP port**, **Sender password**, and **Confirm password**.
3. In the **Receiver email addresses**, you can enter multiple email addresses separated by comma (,).
4. Click **Send test email** to verify the email configuration. If you do not receive test email on the configured email addresses, then check the email configuration.
5. Click **Apply** to save the changes.

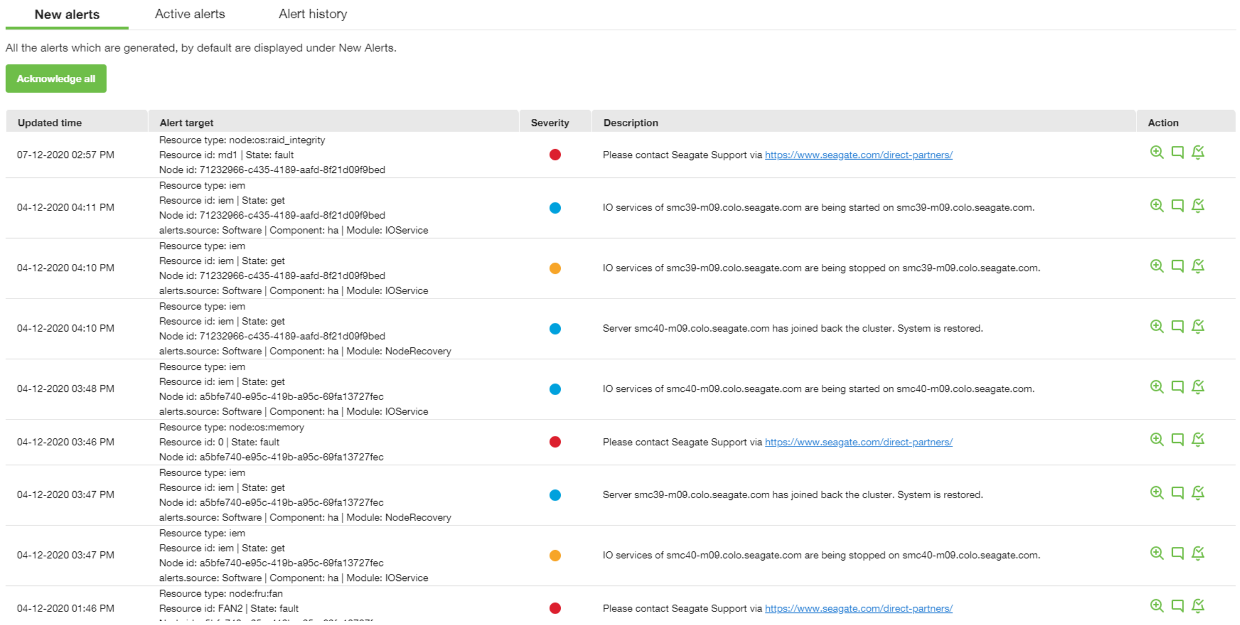
11 | Alerts

An alert is triggered when any of the system component malfunctions. For example, if a hard drive stops to function, an alert is triggered.

Viewing alerts using CORTX Manager

- On the **Dashboard**, under **New Alerts** click ⊕ to open the alerts page.

On the **New alerts** tab, you can view all the new alerts. When an alert is generated, it is by default in the New Alert category.



All the alerts which are generated, by default are displayed under New Alerts.

Updated time	Alert target	Severity	Description	Action
07-12-2020 02:57 PM	Resource type: node:os:raid_integrity Resource id: md1 | State: fault Node id: 71232966-c435-4189-aafd-8f21d09f9bed	🔴	Please contact Seagate Support via https://www.seagate.com/direct-partners/	⊕ □ 🔕
04-12-2020 04:11 PM	Resource type: iem Resource id: iem | State: get Node id: 71232966-c435-4189-aafd-8f21d09f9bed alerts.source: Software | Component: ha | Module: IOService	🔵	IO services of smc39-m09.colo.seagate.com are being started on smc39-m09.colo.seagate.com.	⊕ □ 🔕
04-12-2020 04:10 PM	Resource type: iem Resource id: iem | State: get Node id: 71232966-c435-4189-aafd-8f21d09f9bed alerts.source: Software | Component: ha | Module: IOService	🟠	IO services of smc39-m09.colo.seagate.com are being stopped on smc39-m09.colo.seagate.com.	⊕ □ 🔕
04-12-2020 04:10 PM	Resource type: iem Resource id: iem | State: get Node id: 71232966-c435-4189-aafd-8f21d09f9bed alerts.source: Software | Component: ha | Module: NodeRecovery	🔵	Server smc40-m09.colo.seagate.com has joined back the cluster. System is restored.	⊕ □ 🔕
04-12-2020 03:48 PM	Resource type: iem Resource id: iem | State: get Node id: a5bfe740-e95c-419b-a95c-69fa13727fec alerts.source: Software | Component: ha | Module: IOService	🔵	IO services of smc40-m09.colo.seagate.com are being started on smc40-m09.colo.seagate.com.	⊕ □ 🔕
04-12-2020 03:46 PM	Resource type: node:os:memory Resource id: 0 | State: fault Node id: a5bfe740-e95c-419b-a95c-69fa13727fec	🔴	Please contact Seagate Support via https://www.seagate.com/direct-partners/	⊕ □ 🔕
04-12-2020 03:47 PM	Resource type: iem Resource id: iem | State: get Node id: a5bfe740-e95c-419b-a95c-69fa13727fec alerts.source: Software | Component: ha | Module: NodeRecovery	🔵	Server smc39-m09.colo.seagate.com has joined back the cluster. System is restored.	⊕ □ 🔕
04-12-2020 03:47 PM	Resource type: iem Resource id: iem | State: get Node id: a5bfe740-e95c-419b-a95c-69fa13727fec alerts.source: Software | Component: ha | Module: IOService	🟠	IO services of smc40-m09.colo.seagate.com are being stopped on smc40-m09.colo.seagate.com.	⊕ □ 🔕
04-12-2020 01:46 PM	Resource type: node:fru:fan Resource id: FAN2 | State: fault	🔴	Please contact Seagate Support via https://www.seagate.com/direct-partners/	⊕ □ 🔕

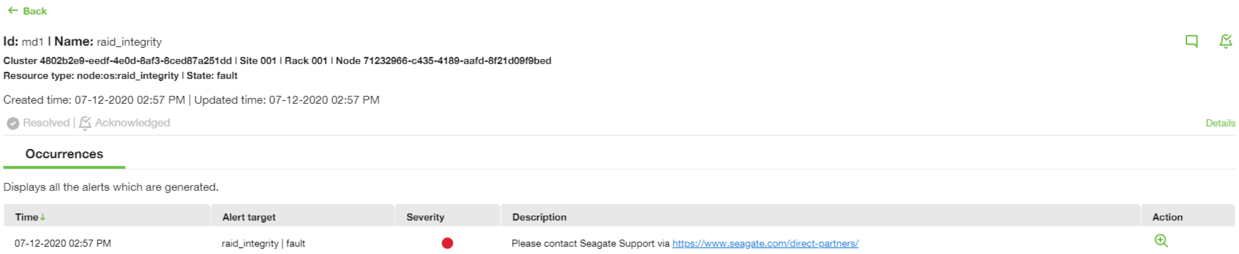
On the **Active alerts** tab, you view all the active alerts. Active alerts are alerts which are either acknowledged or resolved.

Column Name	Description
Updated time	Displays the time when the alert was updated.
Alert target	Displays details of the alert such as following. • Resource type – specifies type of the resource such as enclosure, or node, or node OS, and so on. • Resource id – specifies unique identification of the resource. State - Specifies state of the alert. It can have any of the following values:

Column Name	Description
	• Fault • Missing • Fault_resolved • Insertion • Threshold_breached:low • Threshold_breached:up • Node id
Severity	Displays the severity of the alert.
Description	Provides more information about the alert.
Action	You can take an action on a specific alert. You can: • View alert details • Add comments • Acknowledge an alert

Viewing alert details using CORTX Manager

Click the ⊕ associated with an alert to view the details of each alert.



← Back

Id: md1 I **Name:** raid_integrity

Cluster 4802b2e9-eedf-4e0d-8af3-8ced87a251dd I Site 001 I Rack 001 I Node 71232966-c435-4189-aafd-8f21d09f9bed

Resource type: node:os:raid_integrity I State: fault

Created time: 07-12-2020 02:57 PM | Updated time: 07-12-2020 02:57 PM

Resolved | Acknowledged

Displays all the alerts which are generated.

Time↓	Alert target	Severity	Description	Action
07-12-2020 02:57 PM	raid_integrity | fault	●	Please contact Seagate Support via https://www.seagate.com/direct-partners/	⊕

Viewing alerts using CLI

alerts show

Description

Shows information about the alerts on the storage system. Alerts are generated by SSPL. A system which monitors disks, fans, controllers, and so on.

Alerts which are Acknowledged and Resolved are visible in "alerts history" output. To fetch alert id of "un-acknowledged alerts" you must run "alerts show" command and not "alerts history".

Minimum role: monitor

Syntax

```
alerts show
[-a]
[-d <x>s|<y>m|<z>h|<q>d]
[-f table|xml|json]
[-l <LIMIT>]
[-s]
```

Parameters

`[-a]`

Optional. Displays all active alerts.

`-d <x>s|<y>m|<z>h|<q>d`

Optional. `-d` indicates duration. Displays information about the alerts generated in the specified duration. Default duration is of 60 seconds.

- `<x>s`: To note the duration in seconds.
- `<y>m`: To note the duration in minutes.
- `<z>h`: To note the duration in hours.
- `<q>d`: To note the duration in days.

Where `<x>`, `<y>`, `<z>`, `<q>` is amounts of seconds, minutes, hours, days respectively.

`[-f table|xml|json]`

Optional. Specifies format of the output.

`[-l <LIMIT>]`

Optional. `l` indicates limit. Displays the specified number of alerts. Default value is 1000.

`[-s]`

Optional. Displays all alerts – Active and Deactive.

Output

Property name as per format	Property type	Description
Alert ID Table: Alert Id xml: alert_uuid json: alert_uuid	String	Specifies alert's unique Id (UUID – Universally Unique Identifier).
Health Table: Health xml: health	String	Specifies health of the alert. Values: • Degraded

Property name as per format	Property type	Description
json: health		- None - Fault
Description Table: Description xml: description json: description	String	Displays description of the alert.
Severity Table: Severity xml: severity json: severity	String	Specifies severity of the alert. Values: - Warning - Error - Critical - Informational - Alert - Notice - Configuration - Detail - Debug
State Table: State xml: state json: state	String	Specifies state of the alert. Values: - Fault - Missing - Fault_resolved - Insertion - Threshold_breached:low - Threshold_breached:up
Acknowledged Table: Acknowledged xml: acknowledged json: acknowledged	Boolean	Specifies if the alert has been acknowledged by a user. Values: - True – if the alert has been acknowledged - False - if the alert has not been acknowledged
Resolved Table: Resolved xml: resolved json: resolved	Boolean	Specifies if the alert has been resolved by the CORTX Manager. Values: - True – if the alert has been resolved by the CORTX Manager - False - if the alert has not been resolved
Comments Table: N/A xml: comments json: comments	String	Shows comment made on the alert.
Component Table: N/A xml: component json: component	String	Specifies the CORTX component that has generated IEM (Interesting Event Message). For example: S3, SSPL, and so on.
Created time Table: N/A xml: created_time json: created_time	DateTime	Specifies time of the alert generation.
Disk Slot Table: N/A xml: disk_slot json: disk_slot	Number	Specifies slot number of the disk. For example: 23, 82, and so on.
Durable ID	String	Specifies resource id of the component.

Property name as per format	Property type	Description
Table: N/A xml: durable_id json: durable_id		For example: psu_0.0
Enclosure ID Table: N/A xml: enclosure_id json: enclosure_id	Int	Specifies id of the storage enclosure.
Event Details Table: N/A xml: event_details json: event_details	String	This field contains details of the fault occurred in the component. This field is used to display all details present on the Alert's detail page of CORTX Manager.
Extended Information Table: N/A xml: extended_info json: extended_info	String	Contains a specific information about a component.
Health Recommendation Table: N/A xml: health_recommendation json: health_recommendation	String	Displays recommendations to resolve the issue occurred.
Host ID Table: N/A xml: host_id json: host_id	String	Specifies name of the system host.
Location Table: N/A xml: location json: location	String	Specifies component location.
Module Table: N/A xml: module json: module	String	Specifies sub module of the component that has generated the IEM.
Module Name Table: N/A xml: module_name json: module_name	String	Specifies name of the module. For example: enclosure:fru:disk
Module Type Table: N/A xml: module_type json: module_type	String	Specifies type of the module. For example: disk
Name Table: N/A xml: name json: name	String	Name of the component. For example: Sideplane will have "Right Sideplane" as Name
Sensor Information Table: N/A xml: sensor_info json: sensor_info	String	This field is used to determine duplicate alerts, resolving the bad alerts. For example: 11_2_10_2_disk_00.85_enclosure:fru:disk
Serial Number Table: N/A xml: serial_number json: serial_number	String	Specifies serial number of the hardware component For example: disk, controller and so on.
Source Table: N/A xml: source json: source	String	Indicates type of component such as - Hardware, or Software.
State	String	Represents the current state of an alert

Property name as per format	Property type	Description
Table: N/A xml: state json: state		• Values • Fault • Fault_resolved • Missing • Insertion
Updated Time Table: N/A xml: updated_time json: updated_time	Number	Represents the time at which the alert was updated. It is indicated in Unix time (also known as Epoch time). For example: 1587980409
Version Table: N/A xml: version json: version	String	Represents version of the operating system. This field is visible when "node:os:system" alert has occurred. For example: "version": "3.10.0-862.el7.x86_64"
Volume Group Table: N/A xml: volume_group json: volume_group	String	In case of logical volume alerts volume group represents the group the volume belongs to. For example: "volume-group": "UNGROUPEDVOLUMES"
Volume Size Table: N/A xml: volume_size json: volume_size	String	Specifies remaining storage capacity of a volume. For example: If 2000GB storage has been utilized in a volume and total size of that volume is 8000GB then 6000GB will be the volume size.
Volume Total Size Table: N/A xml: volume_total_size json: volume_total_size	String	Represents total storage capacity of the volume. For example: 8000GB

Examples

To display all active alerts for last 10 days, run the following command.

```
alerts show -d 10d -a
```

To display 25 alerts generated during last 24 hours, run the following command.

```
alerts show -d 24h -l 25
```

Viewing alert comments using CLI

alerts comment show

Description

Displays comment associated with the specified alert.

Minimum role: monitor

Syntax

```
alerts comment show
<alert_uuid>
[-f table|xml|json]
```

Parameters

```
<alert_uuid>
```

Required. UUID stands for "Universally Unique Identifier". Specifies a unique ID of the alert of which you would like to see the comment.

```
[-f table|xml|json]
```

Optional. Specifies the output format type. Default value is *table*.

Output

Property name as per format	Property Type	Description
Comment ID Table: Comment ID xml: comment_id json: comment_id	String	Specifies comment ID of the component.
Comment Table: Comment xml: comment json: comment	String	Specifies the comment associated with the alert.
Created by Table: Created by xml: created_by json: created_by	String	Displays time at which the comment has been added.

Examples

To display comment of a specific alert (alert UUID - mndr18r08m2016j0sh1), run the following command.

```
Alerts comment show mndr18r08m2016j0sh1
```

To display comment of a specific alert (alert UUID - mndr18r08m2016j0sh1) in XML format, run the following command.

```
Alerts comment show mndr18r08m2016j0sh1 –f xml
```

Adding comments using CORTX Manager

To add comments using CORTX Manager:

- Click the 🗨 to open the **Comments** screen. Add your comment in the **comment** field, and then click **Save**.

> **Note**
>
> You can also view the previous comments on the Comments screen.

Adding comments using CLI

alerts comment add

Description

Adds comment to the specified alert.

Minimum role: manage

Syntax

```
alerts comment add <alerts_uuid> <comment_text>
```

Parameters

```
<alerts_uuid>
```

Required. Specifies ID of the alert for which you would like to add a comment.

```
<comment_text>
```

Required. Comment message that needs to get displayed along with the alert. You can use "_" to separate the words in a sentence. Spaces are not allowed in a sentence.

Type "y" and then press ENTER to confirm addition of comment text.

Output

A confirmation message is displayed.

Examples

To add a comment (Renew SSL certificate) to an alert (alert id - 883a3b682629e4da785f86), run the following command.

```
alerts comment add mndr18r08m2016j0sh1 Renew_SSL_certificate
```

Acknowledging an alert using CORTX Manager

To acknowledge an alert using CORTX Manager:

- Click the 🔔 associated with an alert to acknowledge the alert.

Acknowledging all alerts using CORTX Manager

To acknowledge all alerts using CORTX Manager:

- On the **New alerts** tab, click **Acknowledge all** to acknowledge all the new alerts.

Acknowledging alerts using CLI

alerts acknowledge

Description

Acknowledges all alerts.

Minimum role: manage

Syntax

```
alerts acknowledge
<alerts_id> [-ack]
```

Parameters

```
<alerts_id>
```

Required. A variable used to specify the alert ID.

```
[-ack]
```

Optional. Marks the specified alert as “Acknowledged”. If this parameter is not used, then the alert is considered as “Unacknowledged”. Default value is false.

Output

A confirmation message is displayed.

Examples

To acknowledge an alert (alert ID-18082016), run the following command.

```
alerts acknowledge 18082016 -ack
```

Type "y" and then press ENTER to confirm the acknowledgement.

Viewing alert history using CORTX Manager

On the **Alerts history** tab, you can view all the alert history. Displays alerts which are acknowledged as well as resolved.

Viewing alert history using CLI

alerts history

Description

Shows history of the specified alert.

Alerts which are Acknowledged and Resolved are visible in "alerts history" output. To fetch alert id of "un-acknowledged alerts" you must run "alerts show" command and not "alerts history".

Minimum role: monitor

Syntax

```
alerts history
-d <x>s|<y>m|<z>h|<q>d
[-e <%y>-<%m>-<%d>]
[-f table|xml|json]
[-i <sensor_info>]
[-l <LIMIT>]
[-s <%y>-<%m>-<%d>]
```

Parameters

```
-d <x>s|<y>m|<z>h|<q>d
```

Optional. "-d" indicates duration. Displays information about the alerts generated in the specified duration. Default duration is of 60 seconds. The duration must be written in "<x>s" or "<y>m" or "<z>h" or "<q>d" format where,

- <x>s: specifies the duration in seconds.
- <y>m: specifies the duration in minutes.
- <z>h: specifies the duration in hours.
- <q>d: specifies the duration in days.

Where x, y, z, q are values of seconds, minutes, hours, days respectively.

`[-e <%y>-<%m>-<%d>]`

Optional. Specifies the end date till which you would like to see the alert history.

The date must be in '*%Y-%m-%d*' format.

- *%y:* specifies year
- *%m:* specifies month
- *%d:* specifies day

`[-f table|xml|json]`

Optional. Specifies the output format type. Default value is table.

`[-i <sensor_info>]`

Optional. Specifies sensor information of the resource. It helps in identifying node, site, rack, & cluster of the resource.

`[-l <LIMIT>]`

Optional. "l" indicates limit. Displays the specified number of alerts. Default value is 1000.

`[-s <%y>-<%m>-<%d>]`

Optional. Specifies the start date from which you would like to see the alert history.

The date must be in '*%Y-%m-%d*' format.

- *%y:* specifies year
- *%m:* specifies month
- *%d:* specifies day

Output

Property name as per format	Property Type	Description
Alert ID Table: Alert Id xml: alert_uuid json: alert_uuid	String	Specifies alert's unique Id (UUID – Universally Unique Identifier).
Health Table: Health xml: health json: health	String	Specifies health of the alert. Values: • Degraded • None • Null
Description Table: Description xml: description json: description	String	Displays description of the alert.
Severity Table: Severity xml: severity json: severity	String	Specifies severity of the alert. Values: • Warning • Error

Property name as per format	Property Type	Description
		• Critical • Informational • Alert • Notice • Configuration • Detail • Debug
State Table: State xml: state json: state	String	Specifies state of the alert. Values: • Fault • Missing • Fault_resolved • Insertion • Threshold_breached:low • Threshold_breached:up
Acknowledged Table: Acknowledged xml: acknowledged json: acknowledged	Boolean	Specifies if the alert has been acknowledged by a user. Values: • True – if the alert has been acknowledged • False - if the alert has not been acknowledged
Resolved Table: Resolved xml: resolved json: resolved	Boolean	Specifies if the alert has been resolved by the CORTX Manager. Values: • True – if the alert has been resolved by the CORTX Manager • False - if the alert has not been resolved
Sensor Information Table: N/A xml: sensor_info json: sensor_info	String	This field is used to determine duplicate alerts, resolving the bad alerts. For example: 11_2_10_2_disk_00.85_enclosure:fru:disk
Comments Table: N/A xml: comments json: comments	String	Shows comment made on the alert.
Component Table: N/A xml: component json: component	String	Specifies the CORTX component that has generated IEM (Interesting Event Message). For example: S3, SSPL, and so on.
Created Time Table: N/A xml: created_time json: created_time	DateTime	Specifies time of the generation of the alert.
Disk Slot Table: N/A xml: disk_slot json: disk_slot	Number	Specifies slot number of the disk. For example: 23, 82, and so on.
Durable ID Table: N/A xml: durable_id json: durable_id	String	Specifies the resource id of the component. For example: psu_0.0
Enclosure ID Table: N/A xml: enclosure_id json: enclosure_id	Int	Specifies id of the storage enclosure.
Events Details	String	This field contains details of the component fault.

Property name as per format	**Property Type**	**Description**
Table: N/A xml: event_details json: event_details		This field is used to display all details present on the Alert's detail page of CORTX Manager.
Extended Information Table: N/A xml: extended_info json: extended_info	String	Contains a specific information about a component.
Health Recommendation Table: N/A xml: health_recommendation json: health_recommendation	String	Displays recommendations to resolve the issue occurred.
Host ID Table: N/A xml: host_id json: host_id	String	Specifies name of the system host.
Location Table: N/A xml: location json: location	String	Specifies component location.
Module Table: N/A xml: module json: module	String	Sub module of the component that has generated the IEM.
Module Name Table: N/A xml: module_name json: module_name	String	Specifies name of the module. For example: enclosure:fru:disk
Module Type Table: N/A xml: module_type json: module_type	String	Specifies type of the module. For example: disk
Name Table: N/A xml: name json: name	String	Name of the component. For example: Sideplane will have "Right Sideplane" as Name
Sensor Information Table: N/A xml: sensor_info json: sensor_info	String	This field is used to determine duplicate alerts, resolving the bad alerts. For example: 11_2_10_2_disk_00.85_enclosure:fru:disk
Serial Number Table: N/A xml: serial_number json: serial_number	String	Specifies serial number of the HW component For example: disk, controller and so on.
Source Table: N/A xml: source json: source	String	Indicates type of component such as - Hardware, or Software.
Updated Time Table: N/A xml: updated_time json: updated_time	Number	Represents the time at which the alert was updated. It is indicated in Unix time (also known as Epoch time) format. For example: 1587980409

Property name as per format	Property Type	Description
Version Table: N/A xml: version json: version	String	Represents version of the operating system. This field is visible when "node:os:system" alert has occurred. For example: "version": "3.10.0-862.el7.x86_64"
Volume Group Table: N/A xml: volume_group json: volume_group	String	In case of logical volume alerts volume group represents the group to which the volume belongs to. For example: "volume-group": "UNGROUPEDVOLUMES"
Volume Size Table: N/A xml: volume_size json: volume_size	String	Specifies remaining storage capacity of a volume. For example: If 2000GB storage has been utilized in a volume and total size of that volume is 8000GB then 6000GB will be the volume size.
Volume Total Size Table: N/A xml: volume_total_size json: volume_total_size	String	Represents total storage capacity of the volume. For example: 8000GB

Examples

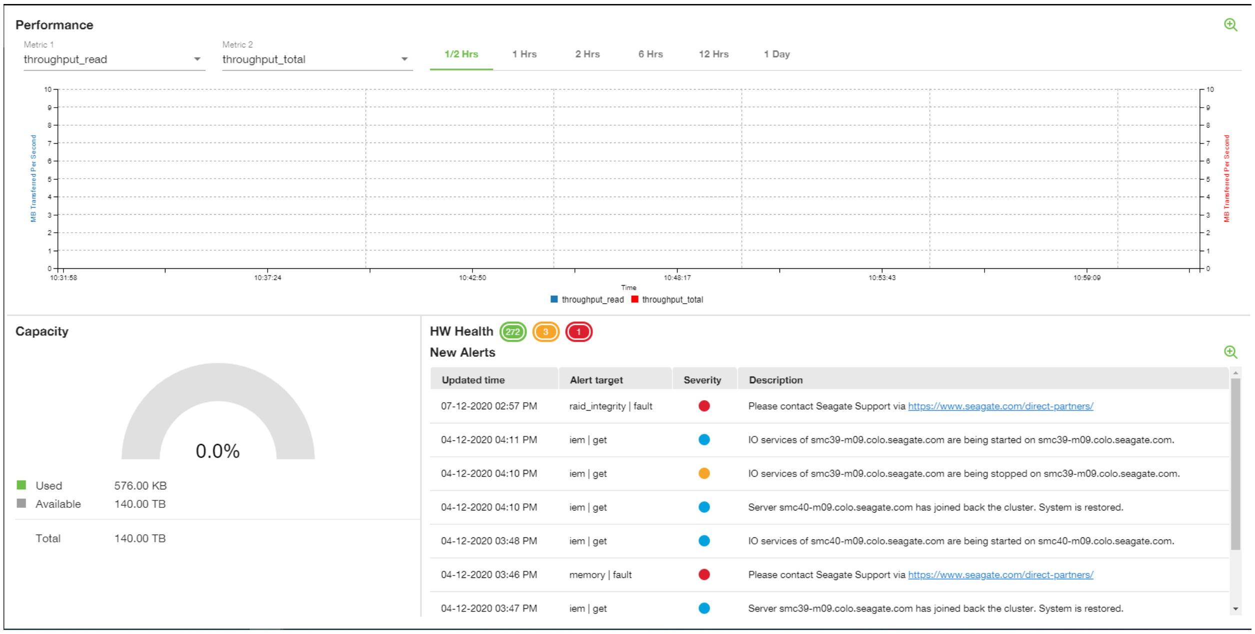
To display alert history from 18th August 2016, run the following command.

```
alerts history -s 2016-08-18
```

To display alert history from 20th May 2016 to 18th August 2016 in JSON format, run the following command.

```
alerts history -s 2016-05-20 -e 2016-08-18 -f json
```

12 | Dashboard



Metric 1: throughput_read

Metric 2: throughput_total

🟩	Used	576.00 KB
▨	Available	140.00 TB
	Total	140.00 TB

Updated time	Alert target	Severity	Description
07-12-2020 02:57 PM	raid_integrity | fault	●	Please contact Seagate Support via https://www.seagate.com/direct-partners/
04-12-2020 04:11 PM	iem | get	●	IO services of smc39-m09.colo.seagate.com are being started on smc39-m09.colo.seagate.com.
04-12-2020 04:10 PM	iem | get	●	IO services of smc39-m09.colo.seagate.com are being stopped on smc39-m09.colo.seagate.com.
04-12-2020 04:10 PM	iem | get	●	Server smc40-m09.colo.seagate.com has joined back the cluster. System is restored.
04-12-2020 03:48 PM	iem | get	●	IO services of smc40-m09.colo.seagate.com are being started on smc40-m09.colo.seagate.com.
04-12-2020 03:46 PM	memory | fault	●	Please contact Seagate Support via https://www.seagate.com/direct-partners/
04-12-2020 03:47 PM	iem | get	●	Server smc39-m09.colo.seagate.com has joined back the cluster. System is restored.

Dashboard helps you to analyze the performance of the system. Dashboard displays important information at one place. It helps you to analyze performance of the system as well as monitor the health of the system.

Performance: Displays the system performance for selected parameters and selected timeline.

Adding graphs: You can add graphs as per your requirements. Click ⊕ in the upper right corner and then click **Add graph** to add a graph. You can add maximum 4 graphs.

Capacity: It displays the storage capacity of your system.

HW Health: It displays the health of your system. The system health is categorized into different severity levels and a severity level is indicated by a different color.

Depending on the severity level of the alert, it is classified as good health 🟩, Warning 🟧, or Critical 🟥.

The ● is an informational alert.

New Alerts: All the alerts which are generated recently and are not resolved are displayed under New Alerts. You can click the (icon?) to see the details of new alerts.

Column Name	Description
Updated time	Displays the time when the alert was updated? Or generated?

Column Name	Description
Alert target	Displays the location of the alert.
Severity	Displays the severity of the alert.
Description	Provides more information about the alert.

Checking system health: It displays the health of your storage enclosure and servers. The system health is categorized into different severity levels and a severity level is indicated by a different color.

13 | System CLI commands

system status

Description

Shows status information of CORTX running on all existing nodes.

Minimum role: admin

Syntax

```
system status
```

Parameters

This command has no parameters associated with it.

Output

Property name as per format	Property Type	Description
Resource name	String	Unique ID of the node
Online/Offline	Boolean	Status of the system • True=Online • False=Offline
Stand-By Status	Boolean	Status of CORTX • True= CORTX is stopped • False= CORTX is working

Examples

To know status of the running system, run the following command.

```
system status
```

system stop

Description

Stops specified node/cluster which is in operation.

Minimum role: admin

Syntax

```
system stop <resource_name>
```

Parameters

`<resource_name>`

Required. Specifies node id of the system controller or cluster which you want to stop.

Output

The user gets logged out of the CLI shell.

Examples

To stop a system by stopping a node named (cub-win312.ad.acme.com), run the following command.

```
system stop cub-win312.ad.acme.com
```

system shutdown

Description

Shuts down running node/cluster.

Minimum role: admin

Syntax

```
system shutdown <resource_name>
```

Parameters

`<resource_name>`

Required. Specifies ID of node/cluster which you want to shut down.

Output

The user gets logged out of the CLI shell.

Examples

To shut down a system using its (cub-win312.ad.acme.com) node, run the following command.

```
system shutdown cub-win312.ad.acme.com
```

system start

Description

Starts the specified node/cluster.

Minimum role: admin

Syntax

```
system start <resource_name>
```

Parameters

```
<resource_name>
```

Required. Specifies node id of the system controller or cluster which you want to start.

Output

N/A

Examples

To start a system having a node named (cub-win312.ad.acme.com), run the following command.

```
system start cub-win312.ad.acme.com
```

system startup

Description

Starts the CORTX cluster and it is used when both nodes are powered ON and yet CORTX doesn’t work.

Minimum role: No login/permissions required.

Syntax

```
system startup
```

Parameters

N/A

Output

N/A

Examples

To start a system having a node named (cub-win312.ad.acme.com), run the following command.

```
system startup
```

Support bundle

Support bundle commands will not be executed within CORTXCLI shell.

support_bundle generate

Description

Generates support bundles for all components. Support Bundle generation requires "admin" privileges.

Minimum role: No login/permissions required.

Syntax

```
support_bundle generate <comment>
-c <component name>
```

Parameters

`<comment>`

Specifies the reason for generating support bundle.

`-c <component name>`

Optional. Specifies component name of which you would like to create support bundle. If it is not specified, it will create support bundle for all the components.

Output

Following is an example of output of this command.

Please use the below ID for checking the status of Support Bundle.

```
SBnyddt3gb
```

File location: `/tmp/support_bundle/`

Examples

To generate a support bundle for motr with "1st support bundle" as the comment, run the following command.

```
support_bundle generate "1st support bundle" -c motr
```

support_bundle status

Description

Shows status of all support bundles.

Minimum role: No login/permissions required.

Syntax

```
support_bundle status <bundle_id>
[-f table|xml|json]
```

Parameters

```
<bundle_id>
```

Required. Specifies unique id of the bundle of which you would like to see the status.

```
[-f table|xml|json]
```

Optional. Specifies the output format type. Default value - table.

Output

Property name as per format	Property Type	Description
Bundle Id Table: Bundle ID xml: bundle_id json: bundle_id	String	Unique ID of the bundle. For example: SBmyde6gb
Comment Table: Comment xml: comment json: comment	String	Specifies comment related to the support bundle.
Node Name Table: Node Name xml: node_name json: node_name	String	Specifies name of the node. For example: server node 1
Message Table: Message xml: message json: message	String	Provides an additional message about this bundle.
Result Table: Result xml: result json: result	String	Displays result of the bundle.

Examples

To display status of a bundle having <bundle_id> as its id, run the following command.

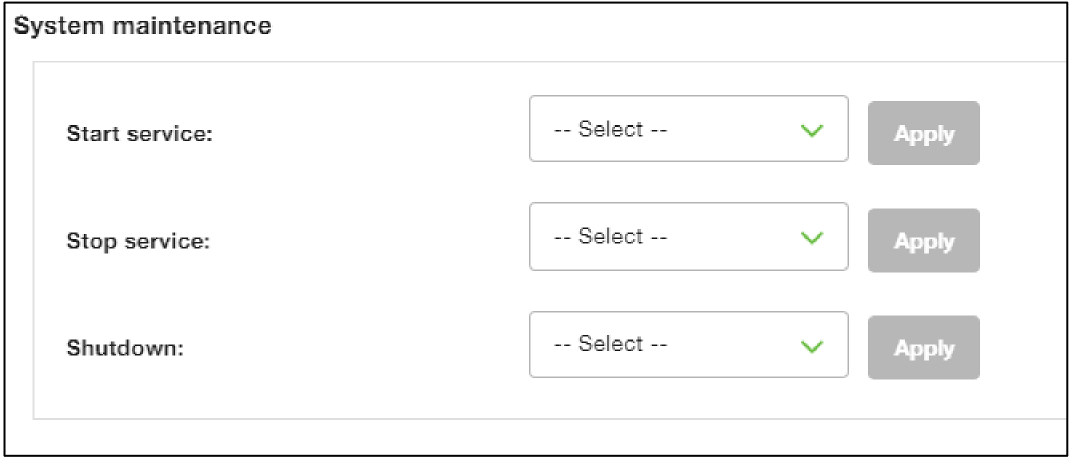
```
support_bundle status bundleID1
```

14 | Maintaining the system

Maintaining system

On the System maintenance page, you can control the services you want to start, stop, or shutdown.

1. Click **Maintenance → System maintenance** to open the **System maintenance** page.



System maintenance

Start service: -- Select --

Stop service: -- Select --

Shutdown: -- Select --

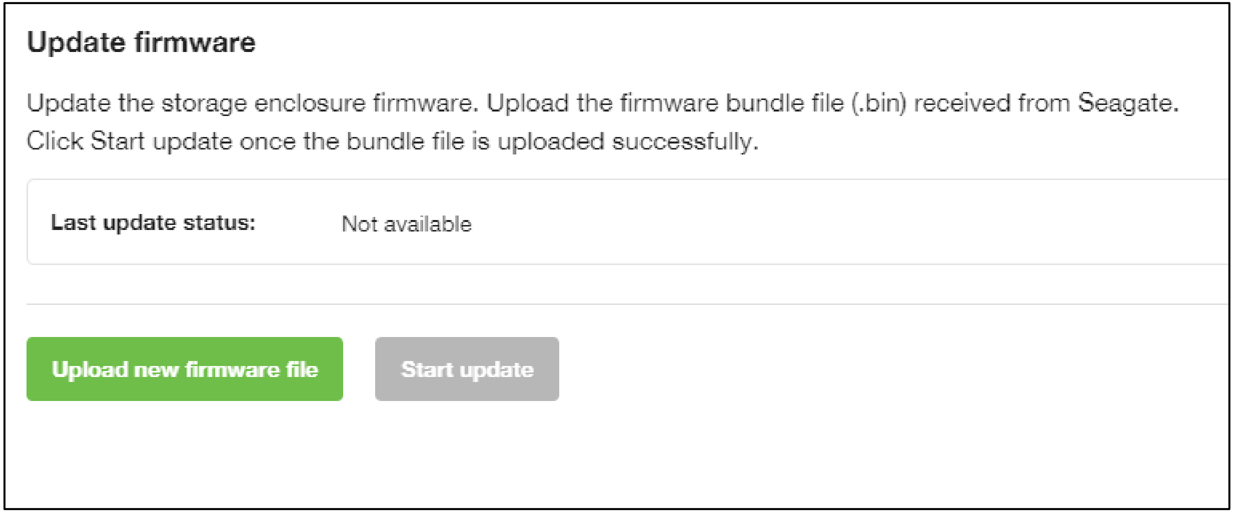
2. Select the appropriate values in the **Start service**, **Stop service**, and **Shutdown** dropdown list, and then click **Apply** for the selected option.

Updating firmware

You can update the firmware of the storage enclosure.

To update the storage enclosure firmware:

1. Click **Maintenance → Firmware update** to open the **Update firmware** page.



Update firmware

Update the storage enclosure firmware. Upload the firmware bundle file (.bin) received from Seagate. Click Start update once the bundle file is uploaded successfully.

Last update status: Not available

2. Click **Upload new firmware file** to browse and select the appropriate .bin firmware file.

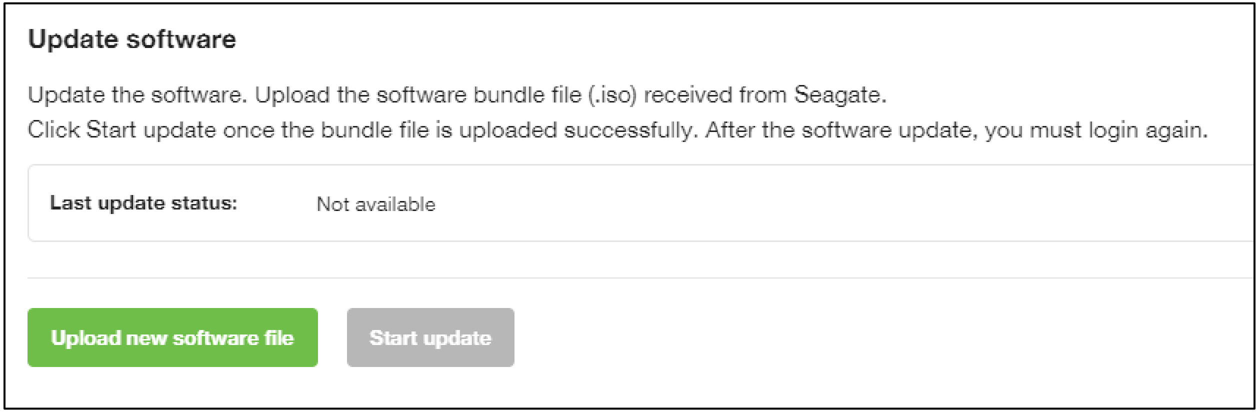
3. Click **Start update** to update the storage enclosure firmware.

Updating software

You can update the software by using the .iso file received from Seagate.

To update the software:

1. Click **Maintenance → Software update** to open the **Update software** page.



Update software

Update the software. Upload the software bundle file (.iso) received from Seagate.
Click Start update once the bundle file is uploaded successfully. After the software update, you must login again.

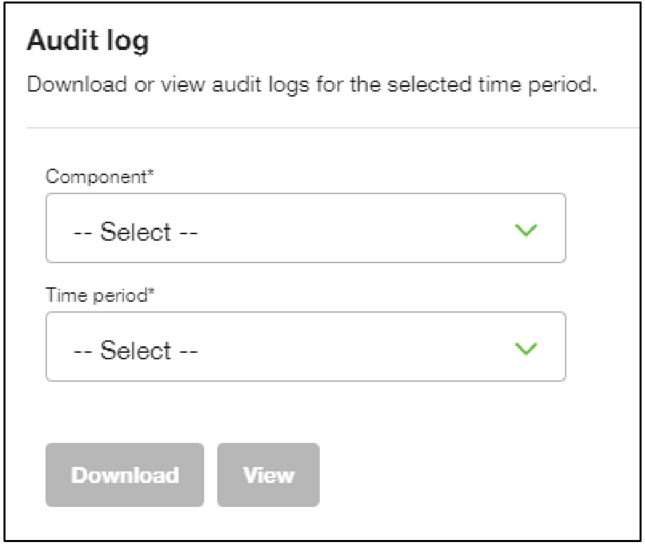
Last update status: Not available

2. Click **Upload new software file** to browse and select the appropriate .iso software update file.
3. Click **Start update** to update the software.

Auditing log

You can view or download the audit logs for the selected time period.

1. Click **Maintenance → Audit log** to open the Audit log page.



Audit log

Download or view audit logs for the selected time period.

Component*: -- Select --

Time period*: -- Select --

2. In the **Component** dropdown list, select the component for which you want to see the audit logs.

3. In the **Time period** dropdown list, select the time period.
4. Click **Download** to download the audit logs
5. Click **View** to view the audit logs.

15 | Switching Lyve Rack ON/OFF

Startup procedure:

1. Flip both power switches on the enclosure.
2. Wait for the enclosure to fully power on (~3 min).
3. Press power buttons on both servers.
4. Wait for the both servers to boot (~5-7 min).
5. Open a browser and connect to CORTX Manager.

At this point the system is ready for use.

Shutdown procedure:

1. Stop the I/O from the S3 clients.
2. Log in to CORTX Manager.
3. Navigate to **Maintenance**.
4. Under **System Maintenance**, click **Manage**.
5. Under **Shutdown**, select a node you want to shut down from the dropdown list, and then click **Apply**.

Note

The selected node will be powered off.

6. Repeat Step 5 for the other node.
7. Wait for both nodes to shutdown (~5 min).
8. Wait for approximately 5 min to allow the enclosure to spin down the drives.
9. Flip both power switches on the enclosure to power off.

This completes the shutdown procedure.