Veritas Access 7.4.2 RESTful APIs

Table of Contents

1. Legal Notice

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043
https://www.veritas.com

1.1. Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan) CustomerCare@veritas.com
Japan Customer Care_Japan@veritas.com

1.2. Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on the cover page. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

1.3. Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

accessdocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

https://www.veritas.com

1.4. Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

2. Overview

2.1. About the Veritas Access RESTful APIs

Veritas Access RESTful APIs are an integral part of the Veritas approach towards Software Defined Storage (SDS). Veritas Access as a SDS offering provides granular control to an application or storage provisioning workflows through the use of RESTful APIs. RESTful APIs not only enable automation through customer scripts, which run storage administration commands against the Veritas Access cluster, but they also enable complex workflow deployments as part of storage provisioning for applications. REST allows an application or another proprietary or custom management interface to have the same management capabilities as the Veritas Access native management console. Veritas Access RESTful APIs allow the use of HTTP requests for applications or storage management workflows or as a portal to automate Veritas Access storage tasks such as:

- Provisioning
- Capacity planning
- Event monitoring
- Data placement on-premises or in the cloud
- Data deletion policies
- Replication and other tasks

2.2. Using the Veritas Access RESTful APIs

You can use the Veritas Access RESTful APIs using the curl command. You need to be authenticated first and then use the returned token in subsequent requests.

To use the Veritas Access RESTful APIs

1. Make sure that the Veritas Access GUI is running on the host.

```
When you execute the Veritas Access RESTful APIs, you have to provide
the host name and the port on which the Veritas Access GUI is running. By
default, the GUI runs on port 14161.
```

2. Authenticate the user.

```
curl --cookie-jar /tmp/cookies/cookies.txt -g -k -X POST -d "username=username&
password=password" https://hostname:port/api/rest/authenticate
```

```
username Username of the host on which the Veritas Access GUI is running.
password Password of the host where the Veritas Access GUI is running.
```

```
The call returns a token.
```

3. Use the returned token and cookie.
4. Call the RESTful API.

```
curl--cookie /tmp/cookies/cookies.txt hostname:port/resturl -g -k -header
"Authorization: Bearer token"
```

```
hostname Host name on which the Veritas Access GUI is running.
port Port on which the Veritas Access GUI is running.
resturl URL for the RESTful API.
token Token that is returned in step 2.
```

```
Example:
```

```
You want to call the API to get all the disks available in the cluster.
curl --cookie /tmp/cookies/cookies.txt
https://10.182.197.183:14161/api/storage/getrecords -g -k --header "Authorization:
Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJjYWZjNmFhZTIyNzEiLCJpYXQiOjE0NzAyNz
k3ODh9.P2CTswezKv-n6wNQTSmSsePC0YBAHRI_VPWF_69yI3o"
```

2.3. How to find the Veritas Access RESTful APIs

To find the Veritas Access RESTful APIs

1. Log on to the GUI.
2. Open a browser window and enter the URL for Veritas Access.

```
Append /rest to the end of the URL in the browser window.
```

```
   Example:
https://IP address of the Veritas Access cluster:14161/rest
```

3. You can also access the RESTful APIs in the GUI by navigating to Settings > Service Management.

The Veritas Access API allows users to create the pool, filesystem, share, including advanced functionality like cloud movement, replication. The Veritas Access API explorer is the easiest way to get started making API calls.

2.4. Version information

Version : 1.0

2.5. Contact information

Contact : Veritas

2.6. URI scheme

BasePath : /
Schemes : HTTPS

3. Paths

3.1. POST /api/rest/authenticate

3.1.1. Description

Authenticates users for using REST api

3.1.2. Parameters

Type	Name	Description	Schema
FormData	**password** *required*	Password	string
FormData	**username** *required*	User Name	string

3.1.3. Responses

HTTP Code	Description	Schema
200	Successfully retrieved all the policies	Auth
401	Authentication error.	No Content
403	User is not authorised or Account is locked, please try after 5 min.	No Content

3.1.4. Produces

- `application/json`

3.1.5. Tags

- Rest Authentication

3.2. POST /api/v1.0/fs

3.2.1. Description

API to create specific file system.

3.2.2. Parameters

Type	Name	Description	Schema
Body	**all** *required*	All	all

all

Name	Description	Schema
blkSize *optional*	**Example** : `"blkSize=8192"`	string
fs_name *optional*	**Example** : `"fs_1"`	string
fs_size *optional*	**Example** : `"1G"`	string
largefs *optional*	**Example** : `"no"`	string
layout *optional*	**Example** : `"simple"`	string
nColumn *optional*	**Example** : `2`	integer
nMirror *optional*	**Example** : `2`	integer
pdirEnable *optional*	**Example** : `"pdir_enable=no"`	string
pool_disks *optional*	**Example** : `"pool1"`	string
protection *optional*	**Example** : `"protection=disk"`	string
stripeUnit *optional*	**Example** : `"stripeUnit=512"`	string

3.2.3. Responses

HTTP Code	Description	Schema
200	file system is created successfully.	object
400	Input errors	inputerrors

HTTP Code	Description	Schema
500	Not able to retreive information from the server.	No Content

3.2.4. Consumes

- `application/json`

3.2.5. Produces

- `application/json`

3.2.6. Tags

- fs

3.3. GET /api/v1.0/fs

3.3.1. Description

API to get all the file systems.

3.3.2. Responses

HTTP Code	Description	Schema
200	List of file systems fetched successfully.	< File_SYSS > array
500	Not able to retreive information from the server.	No Content

3.3.3. Consumes

- `application/json`

3.3.4. Produces

- `application/json`

3.3.5. Tags

- fs

3.4. GET /api/v1.0/fs/iscsi/targets

3.4.1. Description

API for getting all the ISCSI Targets.

3.4.2. Responses

HTTP Code	Description	Schema
200	List of ISCSI Targets that have been fetched successfully..	< target > array
500	Not able to retrieve information from the server.	No Content

3.4.3. Consumes

- `application/json`

3.4.4. Produces

- `application/json`

3.4.5. Tags

- fs

3.5. GET /api/v1.0/fs/iscsi/targets/{target_name}/luns

3.5.1. Description

API for getting all the LUNS of an ISCSI Target.

3.5.2. Parameters

Type	Name	Description	Schema
Path	**target_name** *required*	Name of the ISCSI Target to fetch luns from.	string

3.5.3. Responses

HTTP Code	Description	Schema
200	List of ISCSI LUNS of a specified Target that have been fetched successfully..	< lun > array
500	Not able to retrieve information from the server.	No Content

3.5.4. Consumes

- `application/json`

3.5.5. Produces

- `application/json`

3.5.6. Tags

- fs

3.6. GET /api/v1.0/fs/replication/episodic/jobs

3.6.1. Description

API to get all the replication jobs.

3.6.2. Responses

HTTP Code	Description	Schema
200	List of replication jobs fetched successfully.	< File_Rep_Sys > array
500	Not able to retreive information from the server.	No Content

3.6.3. Consumes

- `application/json`

3.6.4. Produces

- `application/json`

3.6.5. Tags

- fs

3.7. POST /api/v1.0/fs/retention

3.7.1. Description

API to create retention on file path.

3.7.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	path and retention time.	body

body

Name	Description	Schema
file_path *required*	**Example** : `"/vx/fs1/abc.txt"`	string

Name	Description	Schema
ret_time *required*	**Example** : `"1d"`	string

3.7.3. Responses

HTTP Code	Description	Schema
200	file system retention is created successfully	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.7.4. Consumes

- `application/json`

3.7.5. Produces

- `application/json`

3.7.6. Tags

- fs

3.8. GET /api/v1.0/fs/retention

3.8.1. Description

API to get retention on file path.

3.8.2. Parameters

Type	Name	Description	Schema
Query	**file_path** *required*	path	string

3.8.3. Responses

HTTP Code	Description	Schema
200	file system retention information is retrieved successfully	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.8.4. Consumes

- `application/json`

3.8.5. Produces

- `application/json`

3.8.6. Tags

- fs

3.9. DELETE /api/v1.0/fs/retention

3.9.1. Description

API to clear retention on file path.

3.9.2. Parameters

Type	Name	Description	Schema
Body	**fs_ret_object** *required*	path.	fs_ret_object

fs_ret_object

Name	Description	Schema
file_path *required*	**Example** : `"/vx/fs1/abc.txt"`	string

3.9.3. Responses

HTTP Code	Description	Schema
200	file system retention is cleared successfully	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.9.4. Consumes

- `application/json`

3.9.5. Produces

- `application/json`

3.9.6. Tags

- fs

3.10. GET /api/v1.0/fs/tier/policies

3.10.1. Description

API to get all policies for all file systems.

3.10.2. Responses

HTTP Code	Description	Schema
200	List of policies fetched successfully.	< File_SYSS > array
500	Not able to retreive information from the server.	No Content

3.10.3. Consumes

- `application/json`

3.10.4. Produces

- `application/json`

3.10.5. Tags

- fs

3.11. GET /api/v1.0/fs/{fs_id}/tier/policies

3.11.1. Description

API to get policy for a filesystem based on id.

3.11.2. Parameters

Type	Name	Description	Schema
Path	**fs_id** *required*	FS ID	string

3.11.3. Responses

HTTP Code	Description	Schema
200	Details of a specific policy fetched successfully.	object

HTTP Code	Description	Schema
500	Not able to retreive information from the server.	No Content

3.11.4. Consumes

- `application/json`

3.11.5. Produces

- `application/json`

3.11.6. Tags

- fs

3.12. GET /api/v1.0/fs/{fs_id}/tier/policy/schedule

3.12.1. Description

API to get schedule for policy for a filesystem based on id

3.12.2. Parameters

Type	Name	Description	Schema
Path	**fs_id** *required*	FS ID	string

3.12.3. Responses

HTTP Code	Description	Schema
200	Details of a specific policy fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.12.4. Consumes

- `application/json`

3.12.5. Produces

- `application/json`

3.12.6. Tags

- fs

3.13. PUT /api/v1.0/fs/{fs_name}

3.13.1. Description

API to Update fs maxiops.

3.13.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string
Body	**All** *required*	Other	All

All

Name	Description	Schema
max_iops *optional*	**Example** : `2`	integer
op_type *optional*	**Example** : `"set"`	string

3.13.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.13.4. Consumes

- `application/json`

3.13.5. Produces

- `application/json`

3.13.6. Tags

- fs

3.14. DELETE /api/v1.0/fs/{fs_name}

3.14.1. Description

API to delete file system.

3.14.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	FS name	string

3.14.3. Responses

HTTP Code	Description	Schema
200	Details of a specific File system deleted successfully.	object
500	Not able to retreive information from the server.	No Content

3.14.4. Consumes

- `application/json`

3.14.5. Produces

- `application/json`

3.14.6. Tags

- fs

3.15. POST /api/v1.0/fs/{fs_name}/grow

3.15.1. Description

API to Perform grow operation on FS.

3.15.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string
Body	**All** *required*	Other	All

All

Name	Description	Schema
fs_size *optional*	**Example** : "1g"	string
operationOption *optional*	**Example** : "growto"	string
tier *optional*	**Example** : "primary"	string

3.15.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.15.4. Consumes

- `application/json`

3.15.5. Produces

- `application/json`

3.15.6. Tags

- fs

3.16. POST /api/v1.0/fs/{fs_name}/offline

3.16.1. Description

Offline specific file system.

3.16.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	File System name	string

3.16.3. Responses

HTTP Code	Description	Schema
200	file system is offline now.	object

HTTP Code	Description	Schema
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.16.4. Consumes

- `application/json`

3.16.5. Produces

- `application/json`

3.16.6. Tags

- fs

3.17. POST /api/v1.0/fs/{fs_name}/online

3.17.1. Description

Online specific file system.

3.17.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	File System name	string

3.17.3. Responses

HTTP Code	Description	Schema
200	file system is online now.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.17.4. Consumes

- `application/json`

3.17.5. Produces

- `application/json`

3.17.6. Tags

- fs

3.18. GET /api/v1.0/fs/{fs_name}/replication/{repl_type}

3.18.1. Description

API to get replication details on fs.

3.18.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string
Path	**repl_type** *required*	Replication type	string

3.18.3. Responses

HTTP Code	Description	Schema
200	Replication Details are fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.18.4. Consumes

- `application/json`

3.18.5. Produces

- `application/json`

3.18.6. Tags

- fs

3.19. DELETE /api/v1.0/fs/{fs_name}/replication/{repl_type}

3.19.1. Description

API to unconfigure replication on fs

3.19.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string
Path	**repl_type** *required*	Replication type	string
Body	**link_name** *optional*	Link Name	link_name

link_name

Name	Description	Schema
link_name *optional*	**Example** : `"target_link"`	string

3.19.3. Responses

HTTP Code	Description	Schema
200	Replication is unconfigured successfully.	object
500	Not able to retreive information from the server.	No Content

3.19.4. Consumes

- `application/json`

3.19.5. Produces

- `application/json`

3.19.6. Tags

- fs

3.20. POST /api/v1.0/fs/{fs_name}/replication/{repl_type}/pause

3.20.1. Description

API to pause replication job.

3.20.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string
Path	**repl_type** *required*	Replication type	string
Body	**job_name** *optional*	Job Name	job_name

job_name

Name	Description	Schema
job_name *optional*	**Example** : `"job_name"`	string

3.20.3. Responses

HTTP Code	Description	Schema
200	Replication job is paused successfully.	object
500	Not able to retreive information from the server.	No Content

3.20.4. Consumes

- `application/json`

3.20.5. Produces

- `application/json`

3.20.6. Tags

- fs

3.21. POST /api/v1.0/fs/{fs_name}/replication/{repl_type}/resume

3.21.1. Description

API to resume replication job.

3.21.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string

Type	Name	Description	Schema
Path	**repl_type** *required*	Replication type	string
Body	**job_name** *optional*	Job Name	job_name

job_name

Name	Description	Schema
job_name *optional*	**Example** : `"job_name"`	string

3.21.3. Responses

HTTP Code	Description	Schema
200	Replication job is resumed successfully.	object
500	Not able to retreive information from the server.	No Content

3.21.4. Consumes

- `application/json`

3.21.5. Produces

- `application/json`

3.21.6. Tags

- fs

3.22. POST /api/v1.0/fs/{fs_name}/replication/{repl_type}/start

3.22.1. Description

API to start replication job.

3.22.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string
Path	**repl_type** *required*	Replication type	string

3.22.3. Responses

HTTP Code	Description	Schema
200	Replication job is stopped successfully.	object
500	Not able to retreive information from the server.	No Content

3.22.4. Consumes

- `application/json`

3.22.5. Produces

- `application/json`

3.22.6. Tags

- fs

3.23. POST /api/v1.0/fs/{fs_name}/replication/{repl_type}/stop

3.23.1. Description

API to stop replication job.

3.23.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	fs Name	string
Path	**repl_type** *required*	Replication type	string

3.23.3. Responses

HTTP Code	Description	Schema
200	Replication job is stopped successfully.	object
500	Not able to retreive information from the server.	No Content

3.23.4. Consumes

- `application/json`

3.23.5. Produces

- `application/json`

3.23.6. Tags

- fs

3.24. POST /api/v1.0/fs/{fs_name}/shrink

3.24.1. Description

API to Perform shrink operation on FS.

3.24.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string
Body	**All** *required*	Other	All

All

Name	Description	Schema
fs_size *optional*	**Example** : `"1g"`	string
operationOption *optional*	**Example** : `"shrinkto"`	string
tier *optional*	**Example** : `"primary"`	string

3.24.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.24.4. Consumes

- `application/json`

3.24.5. Produces

- `application/json`

3.24.6. Tags

- fs

3.25. POST /api/v1.0/fs/{fs_name}/snapshot

3.25.1. Description

Create a snapshot on the file system.

3.25.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string
Body	**snapshotObject** *required*	Snapshot name and snapshot should be removable or not.	snapshotObject

snapshotObject

Name	Description	Schema
removable *required*	**Example** : `"yes"`	string
snapshot_name *required*	**Example** : `"snap1"`	string

3.25.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.25.4. Consumes

- `application/json`

3.25.5. Produces

- `application/json`

3.25.6. Tags

- fs

3.26. GET /api/v1.0/fs/{fs_name}/snapshot

3.26.1. Description

API for getting all the snapshots on the file system.

3.26.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string

3.26.3. Responses

HTTP Code	Description	Schema
200	List of snapshots from the specific file system that have been fetched successfully..	< snapshot > array
500	Not able to retrieve information from the server.	No Content

3.26.4. Consumes

- `application/json`

3.26.5. Produces

- `application/json`

3.26.6. Tags

- fs

3.27. GET /api/v1.0/fs/{fs_name}/snapshot/{snapshot_name}

3.27.1. Description

API to get the specific snapshot from the file system..

3.27.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string
Path	**snapshot_name** *required*	Snapshot name	string

3.27.3. Responses

HTTP Code	Description	Schema
200	Details of the snapshot that was fetched successfully..	< snapshot > array
500	Not able to retrieve information from the server.	No Content

3.27.4. Consumes

- `application/json`

3.27.5. Produces

- `application/json`

3.27.6. Tags

- fs

3.28. DELETE /api/v1.0/fs/{fs_name}/snapshot/{snapshot_name}

3.28.1. Description

Destroy the snapshot for the specific file system.

3.28.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system.	string
Path	**snapshot_name** *required*	Name of the snapshot.	string

3.28.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.28.4. Consumes

- `application/json`

3.28.5. Produces

- `application/json`

3.28.6. Tags

- fs

3.29. POST /api/v1.0/fs/{fs_name}/snapshot/{snapshot_name}/offline

3.29.1. Description

Make the specific snapshot offline.

3.29.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string
Path	**snapshot_name** *required*	snapshot name	string

3.29.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.29.4. Consumes

- `application/json`

3.29.5. Produces

- `application/json`

3.29.6. Tags

- fs

3.30. POST /api/v1.0/fs/{fs_name}/snapshot/{snapshot_name}/online

3.30.1. Description

Make the specific snapshot online.

3.30.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string
Path	**snapshot_name** *required*	snapshot name	string

3.30.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.30.4. Consumes

- `application/json`

3.30.5. Produces

- `application/json`

3.30.6. Tags

- fs

3.31. POST /api/v1.0/fs/{fs_name}/snapshot/{snapshot_name}/restore

3.31.1. Description

Restore the specific snapshot.

3.31.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	Name of the file system	string
Path	**snapshot_name** *required*	snapshot name	string
Body	**force_fs_online** *required*	Make the file system online or not.	force_fs_online

force_fs_online

Name	Description	Schema
force_fs_online *required*	**Example** : `"1"`	string

3.31.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.31.4. Consumes

- `application/json`

3.31.5. Produces

- `application/json`

3.31.6. Tags

- fs

3.32. POST /api/v1.0/fs/{fs_name}/tier

3.32.1. Description

API to add tier in file system.

3.32.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string
Body	**All** *optional*	Provide the properties based on layout	All

All

Name	Description	Schema
layout *required*	Tier type **Example** : `"cloud"`	enum (cloud, ecoded, mirrored, mirrored-stripe, simple, striped, striped-mirror)
nColumn *optional*	**Example** : `2`	integer
nMirror *optional*	**Example** : `2`	integer
pool_disks *optional*	**Example** : `"pool1"`	string
protection *optional*	**Example** : `"protection=disk"`	string
provider *optional*	Cloud provider type, required for cloud layout type **Example** : `"aws"`	enum (aws, s3_compatible, azure, aws_govcloud_us, google, alibaba)
region *optional*	region based on cloud provider **Example** : `"California"`	string

Name	Description	Schema
service_name *optional*	Registered Service provider, required for cloud layout type **Example** : "aws"	string
stripeunit *optional*	**Example** : "stripeunit=512"	string
subtierType *optional*	**Example** : "Glacier"	string
tier_name *optional*	Tier name, required for cloud layout type **Example** : "AwsTier"	string
tier_size *optional*	**Example** : "1G"	string

3.32.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.32.4. Consumes

- `application/json`

3.32.5. Produces

- `application/json`

3.32.6. Tags

- fs

3.33. DELETE /api/v1.0/fs/{fs_name}/tier

3.33.1. Description

API to delete tier from the system.

3.33.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	FS name	string

3.33.3. Responses

HTTP Code	Description	Schema
200	Details of a specific File system deleted successfully.	object
500	Not able to retreive information from the server.	No Content

3.33.4. Consumes

- `application/json`

3.33.5. Produces

- `application/json`

3.33.6. Tags

- fs

3.34. POST /api/v1.0/fs/{fs_name}/tier/policy/schedule

3.34.1. Description

API to create schedule for running policy on fs tier.

3.34.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string
Body	**All** *required*	Other	All

All

Name	Description	Schema
day_of_the_month *optional*	**Example** : `"*"`	string
day_of_the_week *optional*	**Example** : `"*"`	string
frequency *optional*	**Example** : `"monthly"`	string
hour *optional*	**Example** : `5`	integer

Name	Description	Schema
islargefs *optional*	**Example** : `0`	integer
minute *optional*	**Example** : `59`	integer
month *optional*	**Example** : `"*"`	string
nodename *optional*	**Example** : `"isaA_01"`	string
operation *optional*	**Example** : `"modify"`	string

3.34.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.34.4. Consumes

- `application/json`

3.34.5. Produces

- `application/json`

3.34.6. Tags

- fs

3.35. DELETE /api/v1.0/fs/{fs_name}/tier/policy/schedule

3.35.1. Description

API to delete schedule for running policy on fs tier.

3.35.2. Parameters

Type	Name	Description	Schema
Path	**fs_name** *required*	file system name	string

Type	Name	Description	Schema
Body	**All** *required*	Other	All

All

Name	Description	Schema
day_of_the_month *optional*	**Example** : "*"	string
day_of_the_week *optional*	**Example** : "*"	string
hour *optional*	**Example** : 5	integer
islargefs *optional*	**Example** : 0	integer
minute *optional*	**Example** : 59	integer
month *optional*	**Example** : "*"	string
nodename *optional*	**Example** : "isaA_01"	string
operation *optional*	**Example** : "remove"	string

3.35.3. Responses

HTTP Code	Description	Schema
200	operation successful.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.35.4. Consumes

- `application/json`

3.35.5. Produces

- `application/json`

3.35.6. Tags

- fs

3.36. GET /api/v1.0/fs/{id}

3.36.1. Description

API to get details about a specific FS.

3.36.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	FS ID	string

3.36.3. Responses

HTTP Code	Description	Schema
200	Details of a specific file system fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.36.4. Consumes

- `application/json`

3.36.5. Produces

- `application/json`

3.36.6. Tags

- fs

3.37. GET /api/v1.0/fs/{id}/replication/episodic/job

3.37.1. Description

API to get details about a specific job.

3.37.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	FS ID	string

3.37.3. Responses

HTTP Code	Description	Schema
200	Details of a specific job fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.37.4. Consumes

- `application/json`

3.37.5. Produces

- `application/json`

3.37.6. Tags

- fs

3.38. POST /api/v1.0/fs/{job_name}/replication/{repl_type}/disable

3.38.1. Description

API to disable replication job.

3.38.2. Parameters

Type	Name	Description	Schema
Path	**job_name** *required*	Job Name	string
Path	**repl_type** *required*	Replication type	string

3.38.3. Responses

HTTP Code	Description	Schema
200	Replication job is disabled successfully.	object
500	Not able to retreive information from the server.	No Content

3.38.4. Consumes

- `application/json`

3.38.5. Produces

- `application/json`

3.38.6. Tags

- fs

3.39. POST /api/v1.0/fs/{job_name}/replication/{repl_type}/enable

3.39.1. Description

API to enable replication job.

3.39.2. Parameters

Type	Name	Description	Schema
Path	**job_name** *required*	Job Name	string
Path	**repl_type** *required*	Replication type	string

3.39.3. Responses

HTTP Code	Description	Schema
200	Replication job is enabled successfully.	object
500	Not able to retreive information from the server.	No Content

3.39.4. Consumes

- `application/json`

3.39.5. Produces

- `application/json`

3.39.6. Tags

- fs

3.40. GET /api/v1.0/log/events/

3.40.1. Description

API to get all the events.

3.40.2. Parameters

Type	Name	Description	Schema
Query	**limit** *optional*	limit	string
Query	**offset** *optional*	offset	string
Query	**severity** *optional*	severity of the event	string
Query	**sort_column** *optional*	Name of the column	string
Query	**sort_order** *optional*	order as ASC/DESC	string

3.40.3. Responses

HTTP Code	Description	Schema
200	List of events fetched successfully.	< EVENTDATA > array
500	Not able to retreive information from the server.	No Content

3.40.4. Consumes

- `application/json`

3.40.5. Produces

- `application/json`

3.40.6. Tags

- log

3.41. GET /api/v1.0/protocol/cifs

3.41.1. Description

API to get all the CIFS shares.

3.41.2. Responses

HTTP Code	Description	Schema
200	List of CIFS fetched successfully.	< CIFS > array
500	Not able to retreive information from the server.	No Content

3.41.3. Consumes

- `application/json`

3.41.4. Produces

- `application/json`

3.41.5. Tags

- protocols

3.42. POST /api/v1.0/protocol/cifs/share

3.42.1. Description

API to create CIFS shares.

3.42.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	share object	CIFSSHARE

3.42.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.42.4. Consumes

- `application/json`

3.42.5. Produces

- `application/json`

3.42.6. Tags

- protocols

3.43. PUT /api/v1.0/protocol/cifs/share/client/validate

3.43.1. Description

validate user and group.

3.43.2. Parameters

Type	Name	Description	Schema
Body	**user** *required*	user or group.	user

user

Name	Description	Schema
user *optional*	**Example** : `"SAMGWIN\\user1"`	string

3.43.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.43.4. Consumes

- `application/json`

3.43.5. Produces

- `application/json`

3.43.6. Tags

- protocols

3.44. PUT /api/v1.0/protocol/cifs/share/{share_name}

3.44.1. Description

modify cifs share permissions.

3.44.2. Parameters

Type	Name	Description	Schema
Path	**share_name** *required*	Name of the share	string

Type	Name	Description	Schema
Body	**users** *required*	list of permission options.	users

users

Name	Description	Schema
optionlist *optional*	**Example** : `"owner=root,dir_mask=1777"`	string

3.44.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.44.4. Consumes

- `application/json`

3.44.5. Produces

- `application/json`

3.44.6. Tags

- protocols

3.45. DELETE /api/v1.0/protocol/cifs/{share_name}

3.45.1. Description

Remove the cifs share from the cluster.

3.45.2. Parameters

Type	Name	Description	Schema
Path	**share_name** *required*	Name of the share to be removed	string

3.45.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.45.4. Consumes

- `application/json`

3.45.5. Produces

- `application/json`

3.45.6. Tags

- protocols

3.46. POST /api/v1.0/protocol/cifs/{share_name}/client/allow

3.46.1. Description

Allow the user for cifs share from the cluster.

3.46.2. Parameters

Type	Name	Description	Schema
Path	**share_name** *required*	Name of the share.	string
Body	**users** *required*	List of users and groups.	users

users

Name	Schema
userlist *optional*	< object > array

3.46.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors

HTTP Code	Description	Schema
500	Internal Server Error	No Content

3.46.4. Consumes

- `application/json`

3.46.5. Produces

- `application/json`

3.46.6. Tags

- protocols

3.47. POST /api/v1.0/protocol/cifs/{share_name}/client/deny

3.47.1. Description

Deny the user for cifs share.

3.47.2. Parameters

Type	Name	Description	Schema
Path	**share_name** *required*	Name of the share	string
Body	**users** *required*	List of users and groups.	users

users

Name	Schema
userlist *optional*	< object > array

3.47.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.47.4. Consumes

- `application/json`

3.47.5. Produces

- `application/json`

3.47.6. Tags

- protocols

3.48. GET /api/v1.0/protocol/nfs

3.48.1. Description

API to get all the NFS shares.

3.48.2. Responses

HTTP Code	Description	Schema
200	List of NFS fetched successfully.	< NFS > array
500	Not able to retreive information from the server.	No Content

3.48.3. Consumes

- `application/json`

3.48.4. Produces

- `application/json`

3.48.5. Tags

- protocols

3.49. POST /api/v1.0/protocol/nfs/share

3.49.1. Description

API to create NFS shares.

3.49.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	share object	SHARE

3.49.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.49.4. Consumes

- `application/json`

3.49.5. Produces

- `application/json`

3.49.6. Tags

- protocols

3.50. DELETE /api/v1.0/protocol/nfs/share/{path}

3.50.1. Description

API to delete NFS share.

3.50.2. Parameters

Type	Name	Description	Schema
Path	**path** *required*	Path of the share	string
Body	**clients** *required*	Name of clients in share	CLIENTINPUT

3.50.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.50.4. Consumes

- `application/json`

3.50.5. Produces

- `application/json`

3.50.6. Tags

- protocols

3.51. POST /api/v1.0/protocol/nfs/share/{path}/client

3.51.1. Description

API to add NFS client.

3.51.2. Parameters

Type	Name	Description	Schema
Path	**path** *required*	Path of the share	string
Body	**share object** *required*	Share details contains export options and name of clients	SHAREINPUT

3.51.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.51.4. Consumes

- `application/json`

3.51.5. Produces

- `application/json`

3.51.6. Tags

- protocols

3.52. DELETE /api/v1.0/protocol/nfs/share/{path}/client/{client_name}

3.52.1. Description

API to remove NFS client.

3.52.2. Parameters

Type	Name	Description	Schema
Path	**client_name** *required*	Name of the nfs client	string
Path	**path** *required*	Path of the share	string

3.52.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.52.4. Consumes

- `application/json`

3.52.5. Produces

- `application/json`

3.52.6. Tags

- protocols

3.53. POST /api/v1.0/protocol/s3/bucket

3.53.1. Description

Export a filesystem directory as S3 bucket.

3.53.2. Parameters

Type	Name	Schema
Body	**body** *required*	body

body

Name	Description	Schema
path *optional*	File system directory path **Example** : "/vx/fs1/dir1"	string
user *optional*	User name of bucket owner	string

3.53.3. Responses

HTTP Code	Description	Schema
200	FS directory exported as bucket.	Task
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.53.4. Consumes

- `application/json`

3.53.5. Produces

- `application/json`

3.53.6. Tags

- protocols

3.54. GET /api/v1.0/protocol/s3/buckets

3.54.1. Description

API returns all bucket information.

3.54.2. Responses

HTTP Code	Description	Schema
200	List of S3 buckets.	< s3_bucket > array
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.54.3. Consumes

- `application/json`

3.54.4. Produces

- `application/json`

3.54.5. Tags

- protocols

3.55. POST /api/v1.0/protocol/s3/user/keys

3.55.1. Description

API used to create access key secret keys for a user.

3.55.2. Parameters

Type	Name	Schema
Body	**body** *required*	s3_user_credentials

3.55.3. Responses

HTTP Code	Description	Schema
200	Create S3 user access and secret key.	s3_user_keys_res
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.55.4. Consumes

- `application/json`

3.55.5. Produces

- `application/json`

3.55.6. Tags

- protocols

3.56. GET /api/v1.0/protocol/s3/user/keys

3.56.1. Description

API used to get access and secret keys of an user.

3.56.2. Parameters

Type	Name	Description	Schema
Header	**password** *required*	password of S3 user	string
Header	**user_name** *required*	S3 user name	string

3.56.3. Responses

HTTP Code	Description	Schema
200	List of S3 user access and secret keys.	< s3_access_secret_key_res > array
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.56.4. Consumes

- `application/json`

3.56.5. Produces

- `application/json`

3.56.6. Tags

- protocols

3.57. PUT /api/v1.0/protocol/s3/user/keys

3.57.1. Description

API used to set access and secret keys of an user.

3.57.2. Parameters

Type	Name	Schema
Body	**body** *required*	body

body

Name	Description	Schema
access_key *optional*	Access key which needs to be set	string
password *optional*	password of S3 user	string
secret_key *optional*	Secret key which needs to be set.	string
user_name *optional*	S3 user name **Example** : `"localuser1"`	string

3.57.3. Responses

HTTP Code	Description	Schema
200	List of S3 user access and secret keys.	Response 200
400	Input errors	inputerrors
500	Internal Server Error	No Content

Response 200

Name	Description	Schema
access_key *optional*	Access key	string
message *optional*	Success message	string
success *optional*	Flag for check that operation is successfully completed or not	string

3.57.4. Consumes

- `application/json`

3.57.5. Produces

- `application/json`

3.57.6. Tags

- protocols

3.58. DELETE /api/v1.0/protocol/s3/user/keys

3.58.1. Description

API used to delete access and secret keys of an user.

3.58.2. Parameters

Type	Name	Schema
Body	**body** *required*	body

body

Name	Description	Schema
access_key *optional*	Access key of S3 user which needs to be delete	string
password *optional*	S3 user password	string
username *optional*	S3 user name	string

3.58.3. Responses

HTTP Code	Description	Schema
200	S3 server delete key.	Task
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.58.4. Consumes

- `application/json`

3.58.5. Produces

- `application/json`

3.58.6. Tags

- protocols

3.59. GET /api/v1.0/protocol/s3/users

3.59.1. Description

API returns S3 users information.

3.59.2. Responses

HTTP Code	Description	Schema
200	List of S3 user.	< s3_user > array
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.59.3. Consumes

- `application/json`

3.59.4. Produces

- `application/json`

3.59.5. Tags

- protocols

3.60. POST /api/v1.0/setting/ad/

3.60.1. Description

Configure Active Directory.

3.60.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	AD object	body

body

Name	Description	Schema
admin *required*	**Example** : `"administrator"`	string
cifs_domain *required*	**Example** : `"SAMGWIN.LOCAL"`	string
cifs_domain_controller *required*	**Example** : `"SAMGWIN.LOCAL"`	string
dns_domain *required*	**Example** : `"SAMGWIN.LOCAL"`	string
dns_nameservers *required*	**Example** : `"192.168.10.72"`	string

Name	Description	Schema
password *required*	**Example** : "2fudge"	string

3.60.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.60.4. Consumes

- `application/json`

3.60.5. Produces

- `application/json`

3.60.6. Tags

- settings

3.61. GET /api/v1.0/setting/ad/

3.61.1. Description

API to get details of active directory.

3.61.2. Responses

HTTP Code	Description	Schema
200	Active Directory details fetched successfully.	< AD > array
500	Not able to retreive information from the server.	No Content

3.61.3. Consumes

- `application/json`

3.61.4. Produces

- `application/json`

3.61.5. Tags

- settings

3.62. PUT /api/v1.0/setting/ad/

3.62.1. Description

Modify Active directory details.

3.62.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	AD object	body

body

Name	Description	Schema
admin *required*	**Example** : "administrator"	string
cifs_domain *required*	**Example** : "SAMGWIN.LOCAL"	string
cifs_domain_controller *required*	**Example** : "SAMGWIN.LOCAL"	string
cifs_domain_user *required*	**Example** : "administrator"	string
cifs_security *required*	**Example** : "ads"	string
cifs_work_group *required*	**Example** : "SAMGWIN"	string
dns_domain *required*	**Example** : "SAMGWIN.LOCAL"	string
dns_nameservers *required*	**Example** : "192.168.10.72"	string
password *required*	**Example** : "2fudge"	string

3.62.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.62.4. Consumes

- `application/json`

3.62.5. Produces

- `application/json`

3.62.6. Tags

- settings

3.63. POST /api/v1.0/setting/dedupe/grow

3.63.1. Description

API for grow dedupe storage.

3.63.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.63.3. Consumes

- `application/json`

3.63.4. Produces

- `application/json`

3.63.5. Tags

- settings

3.64. GET /api/v1.0/setting/dedupe/show

3.64.1. Description

API to get details of dedupe msdp server on each node.

3.64.2. Responses

HTTP Code	Description	Schema
200	List of dedup msdp server status along with node name.	< object > array
500	Not able to retreive information from the server.	No Content

3.64.3. Consumes

- `application/json`

3.64.4. Produces

- `application/json`

3.64.5. Tags

- settings

3.65. POST /api/v1.0/setting/dedupe/start

3.65.1. Description

API for start dedupe MSDP server.

3.65.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.65.3. Consumes

- `application/json`

3.65.4. Produces

- `application/json`

3.65.5. Tags

- settings

3.66. GET /api/v1.0/setting/dedupe/status

3.66.1. Description

API to get details of dedupe msdp server on each node.

3.66.2. Responses

HTTP Code	Description	Schema
200	List of dedup msdp server status along with node name.	< Response 200 > array
500	Not able to retreive information from the server.	No Content

Response 200

Name	Description	Schema
node *optional*	Node name	string
status *optional*	ISCSI server status on a node	string

3.66.3. Consumes

- `application/json`

3.66.4. Produces

- `application/json`

3.66.5. Tags

- settings

3.67. POST /api/v1.0/setting/dedupe/stop

3.67.1. Description

API for stop MSDP server.

3.67.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.67.3. Consumes

- `application/json`

3.67.4. Produces

- `application/json`

3.67.5. Tags

- settings

3.68. POST /api/v1.0/setting/dedupe/unconfig

3.68.1. Description

API for stop MSDP server.

3.68.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.68.3. Consumes

- `application/json`

3.68.4. Produces

- `application/json`

3.68.5. Tags

- settings

3.69. POST /api/v1.0/setting/protocol/cifs

3.69.1. Description

API for set CIFS homedirfs.

3.69.2. Parameters

Type	Name	Schema
Body	**body** *required*	body

body

Name	Description	Schema
full_acl *optional*	Value should be yes of full_acl else no **Example** : `"no"`	string
homedirfs *optional*	list of filesystem names	< string > array

3.69.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.69.4. Consumes

- `application/json`

3.69.5. Produces

- `application/json`

3.69.6. Tags

- settings

3.70. GET /api/v1.0/setting/protocol/cifs

3.70.1. Description

API to get details of cifs server on each node.

3.70.2. Responses

HTTP Code	Description	Schema
200	List of CIFS server status along with node name.	< Response 200 > array
500	Not able to retreive information from the server.	No Content

Response 200

Name	Description	Schema
node *optional*	Node name	string
status *optional*	CIFS server status on a node	string

3.70.3. Consumes

- `application/json`

3.70.4. Produces

- `application/json`

3.70.5. Tags

- settings

3.71. POST /api/v1.0/setting/protocol/cifs/offline

3.71.1. Description

API for stop CIFS server.

3.71.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.71.3. Consumes

- `application/json`

3.71.4. Produces

- `application/json`

3.71.5. Tags

- settings

3.72. POST /api/v1.0/setting/protocol/cifs/online

3.72.1. Description

API for start CIFS server.

3.72.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.72.3. Consumes

- `application/json`

3.72.4. Produces

- `application/json`

3.72.5. Tags

- settings

3.73. GET /api/v1.0/setting/protocol/ftp

3.73.1. Description

API to get details of ftp settings.

3.73.2. Responses

HTTP Code	Description	Schema
200	ftp settings.	< object > array
500	Not able to retreive information from the server.	No Content

3.73.3. Consumes

- `application/json`

3.73.4. Produces

- `application/json`

3.73.5. Tags

- settings

3.74. PUT /api/v1.0/setting/protocol/ftp/

3.74.1. Description

API for set ftp configuration.

3.74.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	share object	CONF

3.74.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.74.4. Consumes

- `application/json`

3.74.5. Produces

- `application/json`

3.74.6. Tags

- settings

3.75. POST /api/v1.0/setting/protocol/ftp/offline

3.75.1. Description

API for stop ftp server.

3.75.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.75.3. Consumes

- `application/json`

3.75.4. Produces

- `application/json`

3.75.5. Tags

- settings

3.76. POST /api/v1.0/setting/protocol/ftp/online

3.76.1. Description

API for start ftp server.

3.76.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.76.3. Consumes

- `application/json`

3.76.4. Produces

- `application/json`

3.76.5. Tags

- settings

3.77. GET /api/v1.0/setting/protocol/iscsi/target

3.77.1. Description

API to get details of iscsi target server on each node.

3.77.2. Responses

HTTP Code	Description	Schema
200	List of ISCSI Target server status along with node name.	< Response 200 > array
500	Not able to retreive information from the server.	No Content

Response 200

Name	Description	Schema
node *optional*	Node name	string
status *optional*	ISCSI server status on a node	string

3.77.3. Consumes

- `application/json`

3.77.4. Produces

- `application/json`

3.77.5. Tags

- settings

3.78. POST /api/v1.0/setting/protocol/nfs

3.78.1. Description

API for set server type KNFS or GNFS.

3.78.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	Object to set NFS server type	body

body

Name	Description	Schema
server_type *optional*	Type of NFS server, either KNFS of GNFS	enum (GNFS, KNFS)

3.78.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.78.4. Consumes

- `application/json`

3.78.5. Produces

- `application/json`

3.78.6. Tags

- settings

3.79. GET /api/v1.0/setting/protocol/nfs

3.79.1. Description

API to get details of nfs server on each node.

3.79.2. Responses

HTTP Code	Description	Schema
200	List of NFS server status along with node name.	< Response 200 > array
500	Not able to retreive information from the server.	No Content

Response 200

Name	Description	Schema
node *optional*	Node name	string
status *optional*	NFS server status on a node	string

3.79.3. Consumes

- `application/json`

3.79.4. Produces

- `application/json`

3.79.5. Tags

- settings

3.80. POST /api/v1.0/setting/protocol/nfs/offline

3.80.1. Description

API for stop NFS server.

3.80.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.80.3. Consumes

- `application/json`

3.80.4. Produces

- `application/json`

3.80.5. Tags

- settings

3.81. POST /api/v1.0/setting/protocol/nfs/online

3.81.1. Description

API for start NFS server.

3.81.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.81.3. Consumes

- `application/json`

3.81.4. Produces

- `application/json`

3.81.5. Tags

- settings

3.82. POST /api/v1.0/setting/protocol/s3

3.82.1. Description

API to set default configuration for S3 server like pools, FS size, FS layout and SSL enabled.

3.82.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	Configuration details	s3_default_config

3.82.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.82.4. Consumes

- `application/json`

3.82.5. Produces

- `application/json`

3.82.6. Tags

- settings

3.83. GET /api/v1.0/setting/protocol/s3

3.83.1. Description

API returns default configuration of S3 server.

3.83.2. Responses

HTTP Code	Description	Schema
200	S3 server configuration.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.83.3. Consumes

- `application/json`

3.83.4. Produces

- `application/json`

3.83.5. Tags

- settings

3.84. POST /api/v1.0/setting/protocol/s3/offline

3.84.1. Description

API to start S3 server

3.84.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.84.3. Consumes

- `application/json`

3.84.4. Produces

- `application/json`

3.84.5. Tags

- settings

3.85. POST /api/v1.0/setting/protocol/s3/online

3.85.1. Description

API to start S3 server

3.85.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.85.3. Consumes

- `application/json`

3.85.4. Produces

- `application/json`

3.85.5. Tags

- settings

3.86. GET /api/v1.0/stats/network/{duration}

3.86.1. Description

API to get network details about cluster.

3.86.2. Parameters

Type	Name	Description	Schema
Path	**duration** *required*	Duration	string

3.86.3. Responses

HTTP Code	Description	Schema
200	Network details of cluster fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.86.4. Consumes

- `application/json`

3.86.5. Produces

- `application/json`

3.86.6. Tags

- stats

3.87. GET /api/v1.0/stats/network/{node}/{duration}

3.87.1. Description

API to get details about network of specific node.

3.87.2. Parameters

Type	Name	Description	Schema
Path	**duration** *required*	Duration	string
Path	**name** *required*	Node name	string

3.87.3. Responses

HTTP Code	Description	Schema
200	Network details of node fetched successfully.	object
500	Not able to retreive information from the server.	No Content

3.87.4. Consumes

- `application/json`

3.87.5. Produces

- `application/json`

3.87.6. Tags

- stats

3.88. POST /api/v1.0/storage/cluster/scan

3.88.1. Description

Scan all SCSI devices.

3.88.2. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
500	Internal Server Error	No Content

3.88.3. Consumes

- `application/json`

3.88.4. Produces

- `application/json`

3.88.5. Tags

- storage

3.89. GET /api/v1.0/storage/disks

3.89.1. Description

API to get all the disks in the cluster.

3.89.2. Responses

HTTP Code	Description	Schema
200	List of disks fetched successfully.	< Disk > array
500	Not able to retreive information from the server.	No Content

3.89.3. Consumes

- `application/json`

3.89.4. Produces

- `application/json`

3.89.5. Tags

- storage

3.90. POST /api/v1.0/storage/disks/configure/local

3.90.1. Description

Configure the LUN as local disks.

3.90.2. Parameters

Type	Name	Description	Schema
Body	**body** *required*	disk object	localdisk

3.90.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.90.4. Consumes

- `application/json`

3.90.5. Produces

- `application/json`

3.90.6. Tags

- storage

3.91. DELETE /api/v1.0/storage/disks/{disk_name}

3.91.1. Description

Remove the disk from the cluster.

3.91.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be removed	string

3.91.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.91.4. Consumes

- `application/json`

3.91.5. Produces

- `application/json`

3.91.6. Tags

- storage

3.92. POST /api/v1.0/storage/disks/{disk_name}/format

3.92.1. Description

Format the disk.

3.92.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be moved	string
Body	**force** *required*	format the disk forcefully	force

force

Name	Description	Schema
force *required*	**Example** : `"y"`	string

3.92.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.92.4. Consumes

- `application/json`

3.92.5. Produces

- `application/json`

3.92.6. Tags

- storage

3.93. POST /api/v1.0/storage/disks/{disk_name}/grow

3.93.1. Description

Grow the disk.

3.93.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be grown	string

3.93.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.93.4. Consumes

- `application/json`

3.93.5. Produces

- `application/json`

3.93.6. Tags

- storage

3.94. POST /api/v1.0/storage/disks/{disk_name}/move

3.94.1. Description

Move the disk from source pool to destination pool .

3.94.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be moved	string
Body	**pools** *required*	source and destination pool names	pools

pools

Name	Description	Schema
dest_pool *required*	**Example** : `"pool2"`	string
src_pool *required*	**Example** : `"pool1"`	string

3.94.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.94.4. Consumes

- `application/json`

3.94.5. Produces

- `application/json`

3.94.6. Tags

- storage

3.95. GET /api/v1.0/storage/disks/{name}

3.95.1. Description

API to get details about a specific disk in the cluster.

3.95.2. Parameters

Type	Name	Description	Schema
Path	**name** *required*	Disk Name	string

3.95.3. Responses

HTTP Code	Description	Schema
200	Details of a specific disk fetched successfully.	< Disk > array
500	Not able to retreive information from the server.	No Content

3.95.4. Consumes

- `application/json`

3.95.5. Produces

- `application/json`

3.95.6. Tags

- storage

3.96. GET /api/v1.0/storage/nodes

3.96.1. Description

API to get all the nodes in the cluster.

3.96.2. Responses

HTTP Code	Description	Schema
200	List of nodes fetched successfully.	< nodes > array
500	Not able to retreive information from the server.	No Content

3.96.3. Consumes

- `application/json`

3.96.4. Produces

- `application/json`

3.96.5. Tags

- storage

3.97. POST /api/v1.0/storage/nodes/add

3.97.1. Description

add node to the cluster .

3.97.2. Parameters

Type	Name	Description	Schema
Body	**node_ip** *required*	IP of new node	node_ip

node_ip

Name	Description	Schema
node_ip *required*	**Example** : `"10.20.30.40"`	string

3.97.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.97.4. Consumes

- `application/json`

3.97.5. Produces

- `application/json`

3.97.6. Tags

- storage

3.98. GET /api/v1.0/storage/nodes/{node_id}

3.98.1. Description

API to get details about a specific node in the cluster.

3.98.2. Parameters

Type	Name	Description	Schema
Path	**node_id** *required*	node id	string

3.98.3. Responses

HTTP Code	Description	Schema
200	Details of a specific node fetched successfully.	< nodes > array
500	Not able to retreive information from the server.	No Content

3.98.4. Consumes

- `application/json`

3.98.5. Produces

- `application/json`

3.98.6. Tags

- storage

3.99. DELETE /api/v1.0/storage/nodes/{node_name}

3.99.1. Description

Remove the node from the cluster.

3.99.2. Parameters

Type	Name	Description	Schema
Path	**node_name** *required*	Name of the node to be removed	string

3.99.3. Responses

HTTP Code	Description	Schema
200	Task created successfully.	object
400	Input errors	inputerrors
500	Internal Server Error	No Content

3.99.4. Consumes

- `application/json`

3.99.5. Produces

- `application/json`

3.99.6. Tags

- storage

3.100. GET /api/v1.0/storage/pools

3.100.1. Description

API to get all the pools in the cluster.

3.100.2. Responses

HTTP Code	Description	Schema
200	List of pools fetched successfully.	< Pool > array
500	Not able to retreive information from the server.	No Content

3.100.3. Consumes

- `application/json`

3.100.4. Produces

- `application/json`

3.100.5. Tags

- storage

3.101. GET /api/v1.0/storage/pools/{pool_id}

3.101.1. Description

API to get details about a specific pool in the cluster.

3.101.2. Parameters

Type	Name	Description	Schema
Path	**pool_id** *required*	Pool ID	integer

3.101.3. Responses

HTTP Code	Description	Schema
200	Details of a specific pool fetched successfully.	< Pool > array
500	Not able to retreive information from the server.	No Content

3.101.4. Consumes

- `application/json`

3.101.5. Produces

- `application/json`

3.101.6. Tags

- storage

3.102. POST /api/v1.0/storage/pools/{pool_name}

3.102.1. Description

API to create specific pool in the cluster.

3.102.2. Parameters

Type	Name	Description	Schema
Path	**pool_name** *required*	pool name	string
Body	**disk_isolated** *required*	disk and isolated for pool creation	disk_isolated

disk_isolated

Name	Description	Schema
disks *required*		< string > array
isolated *optional*	**Example** : `"no"`	integer

3.102.3. Responses

HTTP Code	Description	Schema
200	Pool is created successfully.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.102.4. Consumes

- `application/json`

3.102.5. Produces

- `application/json`

3.102.6. Tags

- storage

3.103. DELETE /api/v1.0/storage/pools/{pool_name}

3.103.1. Description

API to remove a specific pool from the cluster.

3.103.2. Parameters

Type	Name	Description	Schema
Path	**pool_name** *required*	Pool name	string

3.103.3. Responses

HTTP Code	Description	Schema
200	Pool is deleted successfully.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.103.4. Consumes

- `application/json`

3.103.5. Produces

- `application/json`

3.103.6. Tags

- storage

3.104. PUT /api/v1.0/storage/pools/{pool_name}/disk/{disk_name}

3.104.1. Description

API to add a specific disk in the pool.

3.104.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be added	< string > array
Path	**pool_name** *required*	Pool name	string

3.104.3. Responses

HTTP Code	Description	Schema
200	Disk is added successfully.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.104.4. Consumes

- `application/json`

3.104.5. Produces

- `application/json`

3.104.6. Tags

- storage

3.105. DELETE /api/v1.0/storage/pools/{pool_name}/disk/{disk_name}

3.105.1. Description

API to remove a specific disk from the pool.

3.105.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be removed	< string > array
Path	**pool_name** *required*	Pool name	string

3.105.3. Responses

HTTP Code	Description	Schema
200	Disk is removed successfully.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.105.4. Consumes

- `application/json`

3.105.5. Produces

- `application/json`

3.105.6. Tags

- storage

3.106. PUT /api/v1.0/storage/pools/{pool_name}/disk/{disk_name}/spare

3.106.1. Description

API to add a specific disk as spare disk.

3.106.2. Parameters

Type	Name	Description	Schema
Path	**disk_name** *required*	Name of the disk to be added	< string > array
Path	**pool_name** *required*	Pool name	string

3.106.3. Responses

HTTP Code	Description	Schema
200	Disk is added as spare disk successfully.	object
400	Input errors	inputerrors
500	Not able to retreive information from the server.	No Content

3.106.4. Consumes

- `application/json`

3.106.5. Produces

- `application/json`

3.106.6. Tags

- storage

4. Definitions

4.1. AD

Name	Description	Schema
cifs_domain *required*	**Example** : "SAMGWIN.LOCAL"	string
cifs_domain_controller *required*	**Example** : "SAMGWIN.LOCAL"	string
cifs_domain_user *required*	**Example** : "administrator"	string
cifs_security *required*	**Example** : "ads"	string
cifs_workgroup *required*	**Example** : "SAMGWIN"	string
dns_domain *required*	**Example** : "SAMGWIN.LOCAL"	string
nameservers *required*	**Example** : "192.168.10.72"	string

4.2. Auth

Name	Schema
role *optional*	string
token *optional*	string

4.3. CIFS

Name	Description	Schema
fs_name *required*	**Example** : "fs-2"	string
fs_status *required*	**Example** : "online"	string

Name	Description	Schema
mount_point_type *required*	**Example** : "CIFS"	string
parent_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs-2"	string
parent_type *required*	**Example** : "CIFS"	string
path *required*	**Example** : "/vx/fs-2"	string
share_name *required*	**Example** : "/vx/fs-2"	string
status *required*	**Example** : "online"	string
subsys_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string

4.4. CIFSSHARE

Name	Description	Schema
path *required*	**Example** : "/vx/fs1"	string
shareDetails *required*		< SHAREDETAILS > array

4.5. CLIENT

Name	Description	Schema
client *optional*	**Example** : "*"	string

4.6. CLIENTINPUT

Name	Schema
client_details *required*	< CLIENT > array

4.7. CONF

Name	Description	Schema
allow_delete *optional*	**Example** : "yes"	enum (yes, no)

Name	Description	Schema
allow_non_ssl *optional*	**Example** : `"yes"`	enum (yes, no)
anonymous_login_dir *optional*	**Example** : `"/dir"`	string
anonymous_logon *optional*	**Example** : `"yes"`	enum (yes, no)
anonymous_write *optional*	**Example** : `"yes"`	enum (yes, no)
chroot_users *optional*	**Example** : `"yes"`	enum (yes, no)
create_homedirs *optional*	**Example** : `"yes"`	enum (yes, no)
homedir_path *optional*	**Example** : `"/path"`	string
idle_timeout *optional*	**Example** : `15.0`	number
listen_ipv6 *optional*	**Example** : `"yes"`	enum (yes, no)
listen_port *optional*	**Example** : `21.0`	number
max_conn_per_client *optional*	maximum connections per client **Example** : `2000.0`	number
max_connections *optional*	**Example** : `8192.0`	number
passive_port_range *optional*	**Example** : `"3000:4000"`	string
security *optional*	**Example** : `"nis_ldap"`	string
umask *optional*	**Example** : `77.0`	number
user_logon *optional*	**Example** : `"yes"`	enum (yes, no)

4.8. Disk

Name	Description	Schema
NODE_NAMES *required*	**Example** : "isaB_01,isaB_02"	string
available_size *required*	**Example** : 2199023255550	integer
device_group_name *required*	**Example** : "-"	string
enclosure *required*	**Example** : "emc0"	string
fs_id *required*	**Example** : "f3223sdf232"	string
fs_name *required*	**Example** : "fs1"	string
health *required*	**Example** : "online"	string
id *required*	**Example** : "EMC%5FSYMMETRIX%5F005678417279%5F1000278000"	string
name *required*	**Example** : "emc0_0278"	string
path *required*	**Example** : "/vx/fs1"	string
serial *required*	**Example** : "1000278000"	string
size *required*	**Example** : 2199023255550	integer
spare *required*	**Example** : "spare"	string
status *required*	**Example** : "enabled"	string
vendor *required*	**Example** : "EMC"	string
version *required*	**Example** : "-"	string

4.9. EVENT

Name	Description	Schema
CATEGORY_ID *required*	**Example** : "1"	string
CATEGORY_ID:1 *required*	**Example** : "1"	string

Name	Description	Schema
CREATED_TIME *required*	**Example** : "2018-03-05 18:03:16"	string
CREATE_TIME *required*	**Example** : "1520301796"	string
EVENT_KEY *required*	**Example** : "event.info.cifs_config.update"	string
EVENT_MSG *required*	**Example** : "CIFS configuration updated on the cluster isaA"	string
INSERTION_STRINGS *required*	**Example** : "none"	string
NAME *required*	**Example** : "Information"	string
OBJECTID *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string
OBJECTNAME *required*	**Example** : "isaA"	string
OBJECTTYPE *required*	**Example** : "vrts_cluster"	string
SEARCHSTRING *required*	**Example** : "vrts_clusterisaA"	string
USER *required*	**Example** : "master"	string

4.10. EVENTDATA

Name	Description	Schema
eventdata *required*		< EVENT > array
total *required*	**Example** : "30"	string

4.11. File_Rep_Sys

Name	Description	Schema
block_size *optional*	**Example** : 8192	integer
cifs_shared *optional*	**Example** : "no"	string

Name	Description	Schema
cloud_provider *optional*	**Example** : "null"	string
defrag_status *optional*	**Example** : "Not Running"	string
encrypted *optional*	**Example** : "null"	string
end_point *optional*	**Example** : "null"	string
file_storage_capacity *optional*	**Example** : 1073741824	integer
file_storage_used *optional*	**Example** : "5.16947968E7"	integer
fs_id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs1"	string
fs_type *optional*	**Example** : "Normal"	string
ftp_shared *optional*	**Example** : "no"	string
fullfsck_status *optional*	**Example** : "Not Running"	string
has_s3_bucket *optional*	**Example** : 0	integer
high_watermark *optional*	**Example** : "null"	integer
id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs1"	string
iops_read *optional*	**Example** : 0	integer
iops_total *optional*	**Example** : 0	integer
iops_write *optional*	**Example** : 0	integer
is_encrypted *optional*	**Example** : false	boolean
is_quota_enabled *optional*	**Example** : "null"	integer
islargefs *optional*	**Example** : 0	integer

Name	Description	Schema
latency_read *optional*	**Example** : `0`	integer
latency_total *optional*	**Example** : `0`	integer
latency_write *optional*	**Example** : `0`	integer
layout *optional*	**Example** : `"simple"`	string
low_watermark *optional*	**Example** : `"null"`	integer
maxiops *optional*	**Example** : `0`	integer
name *optional*	**Example** : `"fs1"`	string
ncols *optional*	**Example** : `1`	integer
nfs_shared *optional*	**Example** : `"no"`	string
nmirrors *optional*	**Example** : `1`	integer
offline_nodes *optional*	**Example** : `""`	string
online_nodes *optional*	**Example** : `"isaA_01"`	string
pool_names *optional*	**Example** : `"pool1"`	string
project_id *optional*	**Example** : `"null"`	string
region *optional*	**Example** : `"null"`	string
resync_status *optional*	**Example** : `"None"`	string
rollsync_status *optional*	**Example** : `"Not Running"`	string
s3_shared *optional*	**Example** : `"No"`	string
secondary_tier *optional*	**Example** : `"no"`	string
service_name *optional*	**Example** : `"Service name"`	string

Name	Description	Schema
share *optional*	**Example** : 0	integer
status *optional*	**Example** : "online"	string
storage_account *optional*	**Example** : "null"	string
stripe_unit *optional*	**Example** : "0.00 K"	string
sub_type *optional*	**Example** : "null"	string
subsys_id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string
throughput_read *optional*	**Example** : 0	integer
throughput_total *optional*	**Example** : 0	integer
throughput_write *optional*	**Example** : 0	integer
tier_name *optional*	**Example** : "null"	string
tier_size *optional*	**Example** : "null"	integer
type *optional*	**Example** : "null"	string
version *optional*	**Example** : "Version 13"	string
worm_status *optional*	**Example** : "No"	string

4.12. File_SYSS

Name	Description	Schema
block_size *optional*	**Example** : 8192	integer
cifs_shared *optional*	**Example** : "no"	string
cloud_provider *optional*	**Example** : "null"	string

Name	Description	Schema
defrag_status *optional*	**Example** : "Not Running"	string
encrypted *optional*	**Example** : "null"	string
end_point *optional*	**Example** : "null"	string
file_storage_capacity *optional*	**Example** : 1073741824	integer
file_storage_used *optional*	**Example** : "5.16947968E7"	integer
fs_id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs1"	string
fs_type *optional*	**Example** : "Normal"	string
ftp_shared *optional*	**Example** : "no"	string
fullfsck_status *optional*	**Example** : "Not Running"	string
has_s3_bucket *optional*	**Example** : 0	integer
high_watermark *optional*	**Example** : "null"	integer
id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs1"	string
iops_read *optional*	**Example** : 0	integer
iops_total *optional*	**Example** : 0	integer
iops_write *optional*	**Example** : 0	integer
is_encrypted *optional*	**Example** : false	boolean
is_quota_enabled *optional*	**Example** : "null"	integer
islargefs *optional*	**Example** : 0	integer
latency_read *optional*	**Example** : 0	integer

Name	Description	Schema
latency_total *optional*	**Example** : `0`	integer
latency_write *optional*	**Example** : `0`	integer
layout *optional*	**Example** : `"simple"`	string
low_watermark *optional*	**Example** : `"null"`	integer
maxiops *optional*	**Example** : `0`	integer
name *optional*	**Example** : `"fs1"`	string
ncols *optional*	**Example** : `1`	integer
nfs_shared *optional*	**Example** : `"no"`	string
nmirrors *optional*	**Example** : `1`	integer
offline_nodes *optional*	**Example** : `""`	string
online_nodes *optional*	**Example** : `"isaA_01"`	string
pool_names *optional*	**Example** : `"pool1"`	string
project_id *optional*	**Example** : `"null"`	string
region *optional*	**Example** : `"null"`	string
resync_status *optional*	**Example** : `"None"`	string
rollsync_status *optional*	**Example** : `"Not Running"`	string
s3_shared *optional*	**Example** : `"No"`	string
secondary_tier *optional*	**Example** : `"no"`	string
service_name *optional*	**Example** : `"Service name"`	string
share *optional*	**Example** : `0`	integer

Name	Description	Schema
status *optional*	**Example** : "online"	string
storage_account *optional*	**Example** : "null"	string
stripe_unit *optional*	**Example** : "0.00 K"	string
sub_type *optional*	**Example** : "null"	string
subsys_id *optional*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string
throughput_read *optional*	**Example** : 0	integer
throughput_total *optional*	**Example** : 0	integer
throughput_write *optional*	**Example** : 0	integer
tier_name *optional*	**Example** : "null"	string
tier_size *optional*	**Example** : "null"	integer
type *optional*	**Example** : "null"	string
version *optional*	**Example** : "Version 13"	string
worm_status *optional*	**Example** : "No"	string

4.13. NFS

Name	Description	Schema
fs_name *required*	**Example** : "fs-2"	string
fs_status *required*	**Example** : "online"	string
mount_point_type *required*	**Example** : "NFS"	string
parent_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs-2"	string

Name	Description	Schema
parent_type *required*	**Example** : "NFS"	string
path *required*	**Example** : "/vx/fs-2"	string
share_name *required*	**Example** : "/vx/fs-2"	string
status *required*	**Example** : "online"	string
subsys_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string

4.14. Pool

Name	Description	Schema
available_size *required*	**Example** : 1128096731074920	integer
id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_pool1"	string
name *required*	**Example** : "pool1"	string
ndevices *required*	**Example** : 2	integer
state *required*	**Example** : 0	integer
subsys_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}"	string
total_size *required*	**Example** : 1128098930098176	integer
used_size *required*	**Example** : 2199023256	integer

4.15. RETENTION

Name	Description	Schema
message *optional*	**Example** : "ACCESS Retention SUCCESS V-493-10-2830 Successfully cleared retention on /vx/default_access_fs/hi.txt"	string
status *optional*	**Example** : "SUCCESS"	string

4.16. SHARE

Name	Description	Schema
path *required*	**Example** : "/vx/fs1"	string
share_details *required*		< SHARE_DETAILS > array

4.17. SHAREDETAILS

Name	Description	Schema
shareName *required*	**Example** : "sharename"	string
shareOptions *required*	**Example** : "ro,noguest,no_full_acl,fs_mode=1777,dir_mask=775,create_mask=775"	string

4.18. SHAREINPUT

Name	Schema
share_details *required*	< SHARE_DETAILS > array

4.19. SHARE_DETAILS

Name	Description	Schema
client *required*	**Example** : "*"	string
export_options *required*	**Example** : "ro,root_squash"	string

4.20. STATUS

Name	Description	Schema
nodename *optional*	**Example** : "isaA_01"	string
status *optional*	**Example** : "ONLINE"	enum (ONLINE, OFFLINE)

4.21. Task

Name	Schema
message *optional*	string
taskId *optional*	string

4.22. ftp_settings_response

Name	Schema
conf *optional*	< CONF > array
status *optional*	< STATUS > array

4.23. inputerrors

Name	Schema
error_summary *optional*	string
message *optional*	string

4.24. localdisk

Name	Description	Schema
node_name *required*	**Example** : `"isaB_01"`	string
product_id *required*	**Example** : `"/vx/fs1"`	string
serial_num *optional*	**Example** : `"1000278000"`	string
vendor_id *required*	**Example** : `"EMC%5FSYMMETRIX%5F005678417279%5F1000278000"`	string

4.25. lun

Name	Description	Schema
lun_name *required*	**Example** : `"lun1"`	string
size *required*	**Example** : `"1G"`	string

Name	Description	Schema
target_name *required*	**Example** : "iqn.2017-12.com.veritas:target2"	string

4.26. nodes

Name	Description	Schema
architecture *required*	**Example** : "x86_64"	string
build_version *required*	**Example** : "7.3.1.000"	string
cifs_server_status *required*	**Example** : "online"	string
cpu_used *required*	**Example** : "11.09"	integer
cvm_state *required*	**Example** : "Slave"	string
discovery *required*	**Example** : 1	integer
family *required*	**Example** : "Red Hat Enterprise Linux Server release 7.3 (Maipo)"	string
id *required*	**Example** : "{00010050-56b9-78c6-0000-000000000000}"	string
isa_version *required*	**Example** : "7.3.1.000"	string
memory_used *required*	**Example** : "36.3279378275"	integer
name *required*	**Example** : "access_01"	string
nfs_server_status *required*	**Example** : "online"	string
os_version *required*	**Example** : "3.10.0-514.el7.x86_64"	string
platform *required*	**Example** : "Linux"	string
s3_server_status *required*	**Example** : "online"	string
state *required*	**Example** : "RUNNING"	string
subsys_id *required*	**Example** : "{50a8a9be-9237-a985-1598-d17d7098ad0d}"	string

4.27. s3_access_secret_key_res

Name	Description	Schema
access_key *optional*	Access key of user. **Example** : `"ZDllMTgyMGJmYzEzNDV"`	string
secret_key *optional*	Secret key of user. **Example** : `"NThhY2Y0NTUwNDRiZTgyY2ZhZGM2OTkzN2ZlMGI"`	string
status *optional*	Status of S3 key. **Example** : `"Active"`	string
user_name *optional*	Name of S3 user. **Example** : `"localuser1"`	string

4.28. s3_bucket

Name	Description	Schema
bucket *optional*	Bucket name **Example** : `"bucket1"`	string
fsname *optional*	Associated file system name **Example** : `"fs1"`	string
owner *optional*	Name of bucket owner **Example** : `"localuser1"`	string
pool *optional*	Pool associated with bucket **Example** : `"pool1"`	string

4.29. s3_config_fs_type

Name	Description	Schema
blkSize *required*	**Example** : `"blksize=8192"`	string
fs_encrypt *optional*	KMS must be configured to created encrypted filesystems, Applicable only for CFS filesystems **Example** : `"no"`	enum (yes, no)
largefs *required*	**Example** : `"yes"`	enum (yes, no)
layout *required*	**Example** : `"mirrored-stripe"`	string
nColumn *optional*	**Example** : `2.0`	number
nMirror *optional*	**Example** : `2.0`	number
nParity *optional*	Applicable only for erasure coded layout **Example** : `2.0`	number

Name	Description	Schema
partitionDir *optional*	Applicable for non CFS filesystem **Example** : "no"	enum (yes, no)
rotating_parit y *optional*	Value can be disk or node, Applicable only for erasure coded layout **Example** : "yes"	enum (yes, no)
stripeUnit *optional*	**Example** : 512.0	number
stripe_aligned *optional*	Value can be yes or no, Applicable only for erasure coded layout **Example** : "yes"	enum (yes, no)
stripetag *optional*	Value can be disk or node, Applicable only for erasure coded layout **Example** : "disk"	enum (disk, node)

4.30. s3_default_config

Name	Description	Schema
fs_size *optional*	Set the default size for bucket **Example** : "10G"	string
fs_type *optional*	Set the default fs layout for bucket	s3_config_fs_type
set_pools *optional*	Comma separated list of pools **Example** : "pool2,pool1"	string
ssl_enabled *optional*	Set SSL setting for S3 server **Example** : "no"	string
unset_pools *optional*	Comma separated list of pools **Example** : "pool1"	string

4.31. s3_default_config_response

Name	Description	Schema
admin_url *optional*	objectaccess server admin URL	string
enabled *optional*	1 for S3 server is enabled and 0 for not **Example** : 1.0	number
fs_blksize *optional*	Block size **Example** : 8192.0	number
fs_encrypt *optional*	1 for encyption enable and 0 for disable **Example** : 0.0	number
fs_ncolumns *optional*	Number of column **Example** : 2.0	number

Name	Description	Schema
fs_nmirrors *optional*	Number of default mirrors **Example** : `2.0`	number
fs_nparity *optional*	Number of nparity for erasure coded layout **Example** : `1.0`	number
fs_pdirenable *optional*	1 for partition directory enabled else 0 **Example** : `1.0`	number
fs_protection *optional*		string
fs_rotating_parity *optional*	1 for rotation parity yes and 0 for no. **Example** : `1.0`	number
fs_stripe_aligned *optional*	1 for striped aligned and 0 for not aligned **Example** : `0.0`	number
fs_stripetag *optional*	**Example** : `"node"`	enum (node, disk)
fs_stripeunit *optional*	**Example** : `512.0`	number
fs_type *optional*	Default FS layout for S3 bucket **Example** : `"largefs-ecoded"`	string
id *optional*		string
pools *optional*	Comma separated pool list of default pools **Example** : `"pool1,pool2"`	string
s3_url *optional*	objectaccess server S3 URL	string
ssl_enabled *optional*	1 for ssl enabled else 0 **Example** : `0.0`	number

4.32. s3_user

Name	Description	Schema
access_key *optional*	S3 sccess key associated with user **Example** : `"ZTI5NjBiYzEyNzJkMTd"`	string
user_name *optional*	S3 user name **Example** : `"localuser1"`	string

4.33. s3_user_credentials

Name	Description	Schema
password *optional*	S3 user password for which S3 keys needs to be created **Example** : `"password"`	string

Name	Description	Schema
username *optional*	S3 user name for which S3 keys needs to be created **Example** : "localuser1"	string

4.34. s3_user_keys_res

Name	Description	Schema
access_key *optional*	access key for user **Example** : "YzZjMGM4ZmM4ZWQ1NmY"	string
secret_key *optional*	secret key for user **Example** : "MWViZTUyYTE2MjU0OWNlM2JhYWQ5YjdiODUwMmY"	string

4.35. snapshot

Name	Description	Schema
ctime *required*	**Example** : "2018.Jan.30.18:40:38"	string
fs_id *required*	**Example** : "{4a6c7f1e-2471-11e7-8645-08abd7770b58}_fs_1"	string
fs_name *required*	**Example** : "fs_1"	string
fs_status *required*	**Example** : "online"	string
mount_point_type *optional*	**Example** : "/vx/fs1"	string
mtime *required*	**Example** : "2018.Jan.30.18:41:16"	string
name *required*	**Example** : "snap1"	string
preserved *optional*	**Example** : "no"	string
removable *required*	**Example** : "yes"	string
size *required*	**Example** : "1"	string
status *required*	**Example** : "online"	string

4.36. target

Name	Description	Schema
initiator_name *optional*	**Example** : `"iqn.2017-12.com.veritas:initiator"`	string
portal_ip *required*	**Example** : `"192.168.10.11"`	string
status *required*	**Example** : `"online"`	string
store *required*	**Example** : `"fs1"`	string
target_auth *required*	**Example** : `"root"`	string
target_name *required*	**Example** : `"iqn.2017-12.com.veritas:target2"`	string

5. Security

5.1. AdminSecurity

Type : apiKey
Name : Authorization
In : HEADER

6. Object Access APIs

6.1. About the Object Access Server

Veritas Object Access provides an implementation of the simple object access service. This section explains the details of the RESTful APIs that are supported for the object access service. The supported APIs are compatible with Amazon S3 unless specified explicitly.

Amazon S3 supports a wide range of services and APIs. The Veritas Access object access server does not support all of Amazon's APIs. Any API that is not documented in this guide should be treated as a non-supported API. There can be exceptions or differences in API behavior. Any exceptions or differences is described in this section.

- User management
- Simple storage service protocol implementation for Amazon S3. The S3 service mandates signing of every request by using access and secret keys as specified by Amazon AWS.

You can use RESTful APIs or the user helper script to create the initial access and secret keys by using the object access user management APIs. After creating the initial AWS access and secret keys, you can use these keys for S3 service object access.

6.2. About the object access user management APIs

User authentication on the Veritas Access server is done using authentication services such as NIS, LDAP, and Active Directory(AD). For authentication to work correctly, the Veritas Access cluster must be configured with the correct authentication service and user password authentication needs to work. The user and identity management APIs depend on the correct working of the background authentication services.

All of the APIs must be called in HTTP POST requests. APIs are called by providing specific URI headers. The requests and responses are encoded in XML format.

6.2.1. Common Error Response

In case of errors, the following XML response is returned to the REST client.

- Response content type is "application/xml"
- RequestID is a unique ID generated per request
- Response content type is XML with following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>AccessDenied</Code>
    <Message>Access Denied.</Message>
  </Error>
  <RequestId>8c94eebd-52c8-4a69-b96e-926af9f791a5</RequestId>
</ErrorResponse>
```

6.2.2. Create access and secret keys for a given user

Use this API to create access and secret keys for a given user. Users can create at a maximum two keys. The secret key created in this API never gets displayed again, so users need to note down the access and secret keys in a secure and accessible location.

Request Parameters

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute, "CreateAccessKey". Parameter 'UserName' Valid username for authentication
- HTTP Header 'VRTSPassword'- Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=CreateAccessKey
&UserName=user1
&Version=2010-05-08

VRTSPassword: urlencode(password)
```

Example Response


```
<?xml version="1.0" encoding="UTF-8"?>
<CreateAccessKeyResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <CreateAccessKeyResult>
    <AccessKey>
      <UserName>user1</UserName>
      <AccessKeyId>YTEzYjdhZGZkMzcwMmE</AccessKeyId>
      <Status>Active</Status>
      <SecretAccessKey>ZGJjYWJjNTZkOWRjYTkwODU4OWMyM2Y5YjI1ODE</SecretAccessKey>
    </AccessKey>
  </CreateAccessKeyResult>
  <ResponseMetadata>
    <RequestId>e101ac91-8a0b-4251-afa2-a6ee0411c59b</RequestId>
  </ResponseMetadata>
</CreateAccessKeyResponse>
```


Possible Error Response

- **Success HTTP** status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exits. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.
- **LimitExceeded** The request was rejected because it attempted to create/delete resources beyond the current account limits. HTTP status code 409.

6.2.3. Delete access and secret key for User

Delete access key. Users must provide a valid access key, username, and password. Use this API to delete the access key. Users must provide a valid access key, username, and password.

Request Parameters

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute, "DeleteAccessKey" Parameter 'UserName' Valid username for authentication Parameter 'AccessKeyId' Access key ID to delete.
- HTTP Header 'VRTSPassword'- Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=DeleteAccessKey
&UserName=user1
&Version=2010-05-08
&AccessKeyId=ZjhmZjM4ODEwN2ZhZGQ

VRTSPassword: urlencode(password)
```

Example Response

```
<DeleteAccessKeyResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <ResponseMetadata>
    <RequestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</RequestId>
  </ResponseMetadata>
</DeleteAccessKeyResponse>
```

Possible Error Response

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exits. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

6.2.4. List access keys for user

Use this API to list access keys created by the user. Users must provide a valid username and password.

Request Parameters

- HTTP URI Parameters - Parameter 'VRTSAction' Service request to execute "ListAccessKeys" Parameter 'UserName' Valid username for authentication
- HTTP Header parameter 'VRTSPassword': Password for user. Password should be url percent encoded.

Example Request

```
http://admin.accessclus1:4567/?VRTSAction=ListAccessKeys
&UserName=user1
&Version=2010-05-08

VRTSPassword: urlencode(password)
```

Example Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAccessKeysResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <ListAccessKeysResult>
    <UserName>support</UserName>
    <AccessKeyMetadata>
      <member>
        <UserName>support</UserName>
    <AccessKeyId>YTEzYjdhZGZkMzcwMmE</AccessKeyId>
    <Status>Active</Status>
      </member>
      <member>
        <UserName>support</UserName>
    <AccessKeyId>YWU4NGRiYzRjNjQ5NGZ</AccessKeyId>
    <Status>Active</Status>
      </member>
    </AccessKeyMetadata>
      <IsTruncated>false</IsTruncated>
  </ListAccessKeysResult>
  <ResponseMetadata>
    <RequestId>c9639b1c-eeda-11e6-8a40-0026b97fab39</RequestId>
  </ResponseMetadata>
</ListAccessKeysResponse>
```

Possible Error response

- **Success** HTTP status code 200.
- **AccessDenied** Request was rejected because user authentication failed. HTTP status code 403.
- **NoSuchEntity** Request was rejected because referenced entity does not exits. In this case there are no access keys created for given user. HTTP status code 404.
- **InternalError** Request failed because of an internal server error. HTTP status code 500.

6.3. Object Access simple storage service (S3) APIs

6.3.1. Common Error Response

In case of error following xml response is returned to REST client.

- Response content type will be "application/xml"
- RequestId is unique id generated per request
- Response content will be xml of following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorResponse xmlns="http://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>AccessDenied</Code>
    <Resource>/bucket1/file.txt</Resource>
    <Message>Access Denied.</Message>
  </Error>
  <RequestId>8c94eebd-52c8-4a69-b96e-926af9f791a5</RequestId>
</ErrorResponse>
```

6.3.2. Operations on Service

GET service

Description - The GET service returns list of all buckets owned by authenticated sender. The implementation of this API is fully complaint with Amazon's GET service API.

6.3.3. Operations on Buckets

The following operations can be performed on the buckets:

Delete Bucket

Description - The DELETE Bucket API deletes the bucket. All objects in the bucket must be deleted before the bucket itself can be deleted.

Security Considerations - Only the bucket owner is allowed to perform DELETE operation.

GET Bucket (List Objects) Version 1

Description - The GET bucket API returns a list of all the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

Differences with respect to Amazon's API - This API is not fully complaint with Amazon's List Objects API. The following is the list of differences with respect to the API response.

- *Request Parameters, delimiter* : Veritas Access supports only "/" string as delimiter. For any other String Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
- *Request Parameters, marker* : Veritas Access expects marker which was returned by server in last request. Using object key as marker is not supported. The value of "NextMarker" of response should be used in request as marker.
- *Response*: The Keys in response will not be in lexicographical order. Application should not assume lexicographical order of response keys.

Security Considerations - Caller must have read permissions on the bucket for the GET Bucket API.

GET Bucket (List Objects) Version 2

Description - The GET bucket API returns a list of all the objects in a bucket. You can use the request parameters as a selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

Differences with respect to Amazon's API - This API is not fully complaint with Amazon's List Objects API. The following is the list of differences with respect to the API response.

- *Request Parameters, delimiter* : Veritas Access supports only "/" string as delimiter. For any other string, the Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.
- *Request Parameters, fetch-owner* : Veritas Access does not support Fetch-owner request paameter.
- *Request Parameters, start-after* : Veritas Access does not support start-after request paameter.
- *Response*: The Keys in response will not be in lexicographical order. Application should not assume lexicographical order of response keys.

Security Considerations - Caller must have read permissions on the bucket for the GET Bucket API.

GET Bucket ACL

Description - The Get Bucket ACL API returns the access control list of the bucket.

Request Headers - This API uses only request headers which are common to APIs.

Security Considerations - For the Get Bucket ACL to succeed, the caller should have READ ACP permission on the bucket. For more information on ACL, refer to ACL.

GET Bucket Location

Description - The GET Bucket Location API returns the bucket's region using LocationConstraint of that object.

Differences with respect to Amazon's API - This API is not fully complaint with Amazon's Get Bucket Location API. The following is the list of differences with respect to the API response.

- *Response*: Veritas Object Access server returns an empty string as the location constraint. This empty string represents US East as the location as per the AWS guide.

Security Considerations - Caller must have read permissions on the bucket for the Get Bucket Location API.

Head Bucket

Description - The Head Bucket API is useful to determine if a bucket exists or not. The operation returns a 200 OK if the bucket exists and the user have permissions to access it. Otherwise, the operation may return responses such as “404 Not Found” and “403 Forbidden”.

Security Considerations - Caller must have read permissions on the bucket for the Head Bucket API.

List Multipart Uploads

Description - The List Multipart Uploads API lists in-progress multipart uploads. An in-progress multipart upload is a multipart upload that has been initiated using the Initiate Multipart Upload request, but has not yet been completed or aborted. This operation returns at most 1,000 multipart uploads in the response and 1,000 multipart uploads is the maximum number of uploads a response can include, which is also the default value.

- *Request Parameters, delimiter* : Veritas Access supports only "/" string as delimiter. For any other string, the Object Access server will return S3_InvalidArgument with HTTP_BAD_REQUEST error code.

Security Considerations - Caller should have read permission on bucket for List Multipart Uploads API.

Put Bucket

Description - The Put Bucket API creates a new bucket. You are not permitted to create buckets using anonymous requests. You become the bucket owner when you create a bucket. The bucket name should comply with DNS-naming conventions.

Supported Request Headers

- x-amz-acl
- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-control

You can set the Canned ACL, grant list, or the detailed ACL using this request header. For more information on ACL, refer to ACL.

Differences with respect to Amazon’s API - This API is not fully complaint with Amazon’s Put Bucket API. The following request elements are ignored by the server.

- CreateBucketConfiguration
- LocationConstraint

Security Considerations - Only a caller with a valid access and secret key can create a bucket on the server.

Put Bucket ACL

Description - Put Bucket ACL sets permission on the existing bucket by using an access control list. You can use one of two ways to set the bucket permission:

1. Specify the ACL in the request body.
2. Specify the permission using request headers. If you specify ACL in the request headers, you must use either a canned ACL or the list of x-amz-grant-<read|write|read-acp|write-acp|full-control>. You cannot use both in the same request. For more information on ACL, refer to ACL.

Supported Request Headers -

- x-amz-acl
- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-control

Non-Supported Request Headers/features - Veritas Access server does not support an email address in the Grantee field.

Differences with respect to Amazon's API -

- While specifying 'CanonicalUser', the user name of grantee and UID of user id needs to specified.
- The 'DisplayName' should be the username of the grantee which should resolve to user id in server's authentication method (passwd/NIS/LDAP/AD).

Security Considerations - Caller should have write-acp permission for the Put Bucket ACL API.

6.3.4. Operations on Objects

Delete Object

Description - The Delete Object API deletes the specified object in the bucket.

Differences with respect to Amazon's API - Veritas Access do not support Object versioning. So, version based object deletion is not supported and will be ignored.

- x-amz-mfa is ignored by the Veritas Delete Object API.

Security Considerations - Caller should have write permissions on the bucket to delete an object.

Get Object

Description - The Get Object API retrieves objects from an Access S3 bucket.

Differences with respect to Amazon's API -

- Range is supported by Veritas Get Object API
- The following parameters are not supported by Veritas Get Object API:
  - If-Modified-Since
  - If-Unmodified-Since
  - If-Match
  - If-None-Match
  - Server-Side encryption

Security Considerations - Caller should have read permissions on the object for the Get object API.

Head Object

Description - The Head Object API retrieves metadata from an object without returning the object itself. This operation is used when you are only interested in an object's metadata.

Differences with respect to Amazon's API - The Veritas Head Object API is not fully compliant with Amazon's Head Object API. The following request parameters are not supported:

- Range
- If-Modified-Since
- If-Unmodified-Since
- If-Match
- If-None-Match
- Server-Side encryption

Security Considerations - Caller should have read permission on object for head object API.

List Parts

Description - The List Parts API lists the parts that have been uploaded for a specific multipart upload. This request returns a maximum of 1,000 uploaded parts. The default number of parts returned is 1,000 parts.

Security Considerations - Caller should have read permission on object for list parts API. By default, the bucket owner has permission to perform list parts API.

Put Object

Description - The Put Object API adds an object to a bucket.

Supported Request Headers - Server will take appropriate action for the following request headers.

- Content-Length
- Content-MD5
- Content-Type
- Expect
- x-amz-meta
- x-amz-acl
- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-control For details on ACL please see ACL

Non-Supported Request Headers - Following request headers are ignored by server.

- Cache-Control
- Content-Disposition
- Content-Encoding
- Expires
- x-amz-website-redirect-location
- Server side encryption headers

Differences with respect to Amazon's API - The object created via this method will be visible to all clients immediately after the completion for the PUT operation. The server works on the following consistency guarantees: -

- For (any number of) an ongoing download, if the PUT on the same object completes successfully, the ongoing request will get data plus the metadata of old object only.
- Any new GET object request after a successful PUT operation will get the latest object.
- Requests made from different nodes of cluster will not change the above semantics.

Put Object Copy

Description - The Put Object Copy API creates a copy of an object that is already stored in the Veritas Access S3 server.

Non-Supported Request Headers - The Following headers are not supported by Put Object Copy API.

- x-amz-tagging-directive
- x-amz-website-redirect-location

- Server side encryption headers

Differences with respect to Amazon's API - The object created by this method is visible to all clients immediately after the completion for the PUT operation. The server works on the following consistency guarantees: -

- For (any number of) an ongoing download, if the PUT on the same object completes successfully, the ongoing request gets data and metadata of old object only.
- Any new GET object request after a successful PUT operation will get the latest object.
- Requests made from different nodes of cluster will not change the above semantics.

Security Considerations - User should have read permission on the source object to perform the Put Copy opertion. Public user is not allowed to perform put object copy operation.

Initiate Multipart upload

Description - The Initiate Multipart Upload API initiates a multipart upload and returns an upload ID. This upload ID is used to associate all of the parts in the specific multipart upload.

Non-Supported Request Headers - Following request headers are not supported for initiate multipart.

- Expires
- x-amz-storage-class
- x-amz-website-redirect-location
- Server side encryption headers

Complete Multipart upload

Description - The Complete Multipart Upload API completes a multipart upload by assembling previously uploaded parts. This API is compliant with the Amazon's API with respect to request parameters.

Differences with respect to Amazon's API - We differ from Amazon in part numbers. The part numbers have to be compulsorily in the increments of 1. For example, the sequence (1, 3, 5, 7) will not work with the Veritas Object Access complete multipart API.

Abort Multipart Upload

Description - The Abort Multipart Upload API aborts aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID. The storage consumed by any previously uploaded parts will be freed.

Security Considerations - By default, the bucket owner and the initiator of the multipart upload are allowed to perform this action.

Upload part

Description - The Upload Part API uploads a part in a multipart upload. There is no minimum allowable part size for a multipart upload.

Non-Supported Request Headers - The following request headers are not supported for upload part.

- Expect
- Server side encryption headers

Upload part copy

Description - The Upload part copy API uploads a part by copying data from an existing object as data source. You specify the data source by adding the request header x-amz-copy-source in your request and a byte range by adding the request header x-amz-copy-source-range in your request. There is no minimum allowable part size for a multipart upload.

Non-Supported Request Headers - Following request headers are not supported for upload part.

- Expect
- Server side encryption headers
- Versioning

Security Considerations - User should have read permission on the source object to perform upload part copy opertion.

6.3.5. Access Control Lists (ACL) in Veritas Object Access server

The access control list enables users to manage and share buckets or objects in the Veritas Object Access server. Every bucket and Object has an associated ACL. This ACL defines which user/group will get what type of access for given resources.

What resource I can grant permissions

- Bucket
- Objects in buckets

What type of Permissions I can grant

- **Read**: This permission allows user to read content of resource.
- **Write**: This permission allows user to write new/delete existing content of resource.
- **Read ACP**: This permission allows user to read contents of ACL set on resource.
- **Write ACP**: This permission allows user to edit contents of ACL set on resource.
- **Full Control**: This permission allows users to perform all above operations.

To whom I can grant permissions

- Users (By using Canonical user ID)
- Predefined Groups (By using group URI), Following predefined groups are supported:

    - Authenticated Users group - This group is represented by URI, http://acs.amazonaws.com/groups/global/AuthenticatedUsers. This group represents all users who are requesting a resource or action by using a valid access and secret key. The request must be authenticated or signed.
    - All Users group - This group is represented by URI, http://acs.amazonaws.com/groups/global/AllUsers All non-signed/authenticated requests are considered as an anonymous request. To get access for an anonymous request, the resource must have required permissions for AllUsers group. **Veritas Object Access server maps 'all user group' to user id 'nobody'**.
    - Log Delivery group - This group is represented by URI, http://acs.amazonaws.com/groups/s3/LogDelivery Permission set of this group will be ignored for now, Since Veritas Object Access server do not support bucket logging.

How permissions work on bucket

- **Read**: This permission allows caller to list objects in bucket and Head bucket.
- **Write**: This permission allows caller create/delete objects in bucket.
- **Read-ACP**: This permission allows caller read bucket ACL by using Get Bucket ACL API.
- **Write-ACP**: This permission allows caller set new set of ACLs on bucket.
- **Full Control**: This permission allows caller to list object, list ACL, create/delete objects and set new ACLs on bucket.

How permissions work on object

- **Read**: This permission allows caller to download (GET + HEAD) object.
- **Write**: This permission is ignored for object.
- **Read-ACP**: This permission allows caller read object ACL by using Get Object ACL API.
- **Write-ACP**: This permission allows caller set new set of ACLs on object.
- **Full Control**: This permission allows caller to GET/HEAD object, list ACL and set new ACLs on object.

How to set permissions on resources

There are 3 ways to set permissions on resource. You can use any of either three ways to set permission on new resource or existing resources. The resource can be object or it can be bucket.

- Setting ACL using request header x-amz-acl i.e. "Canned ACL"
- Setting ACL using Grant header
- Setting ACL using complete details in request body.

The section below explains each approach in more detail.

Setting ACL using request header x-amz-acl i.e. "Canned ACL"

A user can set an ACL on a resource by using the request header x-amz-acl. This request header covers a set of permissions. The following are the list of the Canned ACLs supported by the Veritas

Object Access Server. You can use this header in the create bucket/object call.

- private (Applies to:Bucket and Object) With this ACL, only the owner has full control. If there is no ACL set on the object or if the object is created by some other protocol, this default ACL will be set by the server.
- public-read (*Applies to: Bucket and Object*) The resource owner has full control and S3 All users group has read access.
- public-read-write (*Applies to: Bucket and Object*) The resource owner has full control and S3 All users group has read-write access. Setting this ACL on a bucket is not recommended.
- aws-exec-read: This ACL is not supported by the Veritas Object Access Server and will be ignored.
- authenticated-read (*Applies to: Bucket and Object*): The resource owner has full control and S3 AuthenticatedUsers group has read access.
- bucket-owner-read (*Applies to: Object*): The resource owner has full control and the bucket owner has read access.
- bucket-owner-full-control (*Applies to: Object*) The resource owner has full control and the bucket owner has full control.
- log-delivery-write: This ACL is not supported by the Veritas Object Access Server and will be ignored.

Setting ACL using Grant headers

User can set ACL in PUT request by using the following grant headers. These headers are supported with PUT Object, create bucket API.

- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-control

Following are examples of grant header, x-amz-grant-write: uri="url",id="123"

The valid input for uri is, "http://acs.amazonaws.com/groups/global/AuthenticatedUsers", "http://acs.amazonaws.com/groups/global/AllUsers", "http://acs.amazonaws.com/groups/s3/LogDelivery"

The valid input for id is number which represents UID of grantee. The users email address is not supported with Grant headers.

Setting ACL using complete details in request body.

The ACL can be set by sending data in request body itself. User can send correct xml in PutObjectACL and PutBucketACL reqeust.

Differences with respect to Amazon's API - The ID field of Grantee should be the UID of the user.

The DisplayName name should be the user’s login name. The email address in DisplayName is not supported.