CloudByte™
Containerized Enterprise Storage

## CloudByte ElastiStor

# Plug-in for OpenStack Cinder

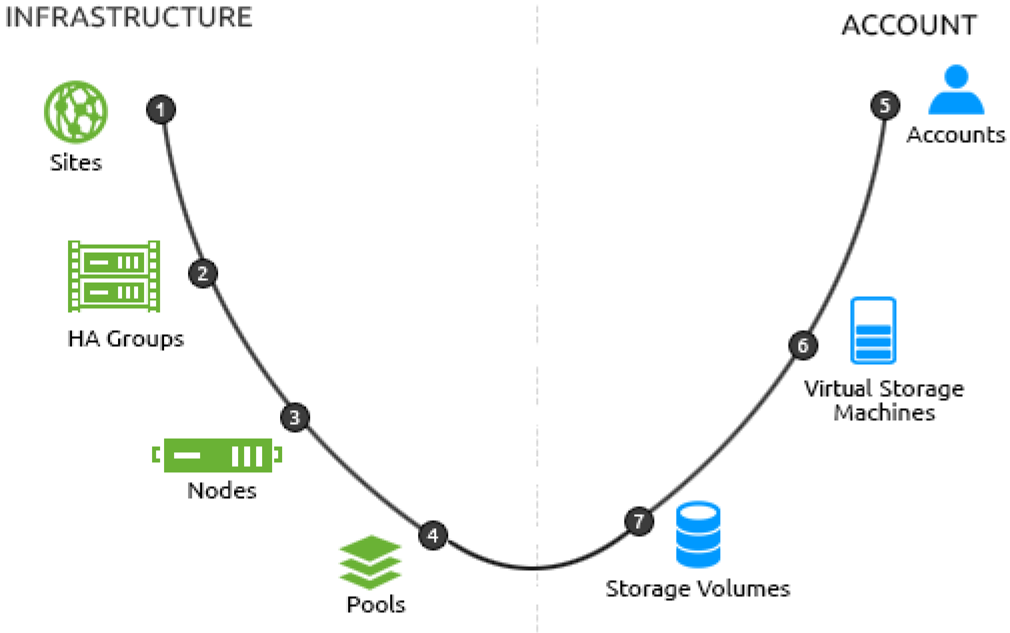
# CloudByte ElastiStor plugin for OpenStack

## Introduction

OpenStack administrators can perform the following tasks, performed using CloudByte ElastiCenter, in the OpenStack dashboard. CloudByte provides a plugin to create, configure, and manage the ElastiCenter tasks.

- Create and manage Storage Volumes
- Create and manage Storage Volume Snapshots
- Create and manage Storage Volume Clones

The following workflow illustrates the procedures involved in storage provisioning using ElastiCenter. You must use ElastiCenter to perform the first six steps that include configuring Sites, HA Groups, Nodes, Pools, Accounts, and VSMs. The key task of setting up the Storage Volumes can be performed using the OpenStack dashboard.



## Pre-requisites

- Download the iSCSI driver for OpenStack Cinder from ElastiStor Plug-ins.
- OpenStack Icehouse or higher is installed.
- A project is created in the OpenStack dashboard.
- CloudByte ElastiStor 1.4 is up and running.
- An Account is created in ElastiCenter (preferably with the same name as the project on OpenStack).

# Working with the Plug-in

**Note**: The following documentation is for CentOS. The procedures might be similar for other flavors of Linux.

## Installing the plug-in

1. Copy the folder `cloudbyte` to the following location on the machine where you have installed OpenStack: `/usr/lib/python2.6/site-packages/cinder/volume/drivers`.
2. Run the following command to edit the `cinder.conf` file in `/etc/cinder`: `vi /etc/cinder/cinder.conf`
3. Add the following lines in the `cinder.conf` file:

   **Note**: Ensure that the `[cloudbyte]` tag is present in `cinder.conf` file before you specify the driver specific properties.

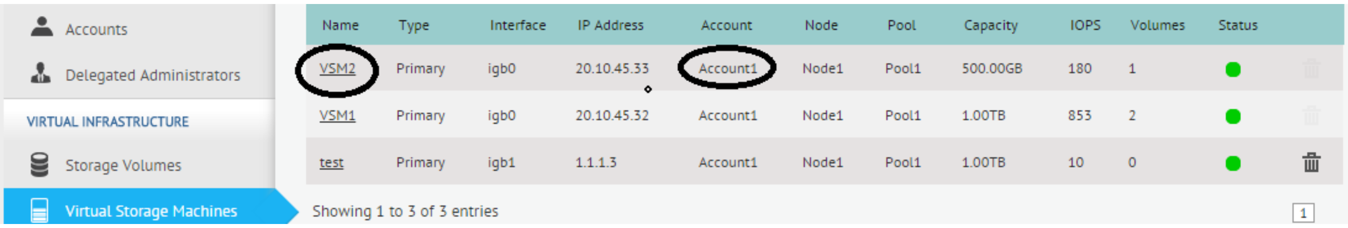
```
[cloudbyte]
volume_driver = cinder.volume.drivers.cloudbyte.cloudbyte.CloudByteISCSIDriver
volume_backend_name = cloudbyte
san_ip = 20.10.22.50
san_password = na
tsm_name = TSM1
cb_account_name = Account1
cb_apikey = IaOU6Qagh-HqIVpNBh69DrT8hpej5dD_g914MHjq-rxfFUPkGOjcVvWPjovlaTT24fn_NL-
d1KTK8D4tof7tUA
confirm_vol_sleep_counter = 10
confirm_vol_sleep_interval = 5
add_qos-
group=latency:15,graceallowed:true,iopscontrol:true,memlimit:0,tpcontrol:false,throughput:0,iops:50,
networkspeed:0
create_volume-
e=compression:off,deduplication:off,blocklength:512B,sync:always,protocoltype:ISCSI,recordsize:16k

[DEFAULT]

enabled_backends=cloudbyte

iscsi_helper=tgtadm
```

4. Modify the highlighted section in the configuration properties by specifying the following details:
   - **san_ip** : The IP address of ElastiCenter
   - **tsm_name** : The VSM on which Storage Volume is created in ElastiCenter
   - **cb_account_name** : The Account on which VSM is created in ElastiCenter



| Name | Type | Interface | IP Address | Account | Node | Pool | Capacity | IOPS | Volumes | Status |
|---|---|---|---|---|---|---|---|---|---|---|
| VSM2 | Primary | igb0 | 20.10.45.33 | Account1 | Node1 | Pool1 | 500.00GB | 180 | 1 | ● |
| VSM1 | Primary | igb0 | 20.10.45.32 | Account1 | Node1 | Pool1 | 1.00TB | 853 | 2 | ● |
| test | Primary | igb1 | 1.1.1.3 | Account1 | Node1 | Pool1 | 1.00TB | 10 | 0 | ● |

Showing 1 to 3 of 3 entries

   - **cb_apikey** : The REST API key that is generated. For details, see CloudByte ElastiStor API reference
   - **enabled_backends** : Specify **cloudbyte** as the value.
   - **iscsi_helper** : Set **tgtadm** as the value.

5. Restart the cinder services using the following commands:
    - `service openstack-cinder-api restart`
    - `service openstack-cinder-scheduler restart`
    - `service openstack-cinder-volume restart`

# Provisioning the Storage

## Simple provisioning

You can use the OpenStack dashboard or the Command Line Interface for basic storage provisioning on ElastiStor. All the storage provisioning commands are same as the standard Openstack cinder commands.

## Advanced provisioning

To set the specified set of QoS parameters, IOPS in particular, perform the following procedures:

1. Fetch the VSM details such as the Account ID and dataset ID by specifying the following URL: https://*ElastiCenter_IP_address*/client/apicommand=listTsm&apiKey=*REST_API_key*&response=json. For more details, see the section *Forming the API request* in "CloudByte ElastiStor API reference".
2. Run the following provisioning command in the Openstack console:

```
 cinder create --metadata accountid=Account_ID tsmid=VSM_ID datasetid=Dataset_ID
apiKey=REST_API_Key iops=Volume_IOPS --name Volume_Name Volume_Size_in_GB
```

| Parameter | Description | Example |
|---|---|---|
| **Account_ID** | ID of the Account that manages the VSM. | `fb7ebb49-ec5e-47d8-85f7-a0b130988bc9` |
| **VSM_ID** | ID of the VSM on which Storage Volume is created on ElastiCenter. | `a53f558d-70cc-3b09-a32a-3af89310ad80` |
| **Dataset_ID** | ID of the dataset associated with the VSM. | `c90a6b7a-eb96-3ed4-9287-0698c49baa4d` |
| **REST_API_Key** | The REST API key of the Account that manages the VSM. | `gY_GYJJc-qN9Hl326tIXlauEnOIKiRO5eodKaYDkOtDRZUk-tgY5XH8kVPZxWfg4iPitK0s7TIb2jSMQLpMowcw` |
| **Volume_IOPS** | IOPS required for the Volume. | 50 |
| **Volume_Name** | A unique name for the Volume. | `Vol1ProjA` |
| **Volume_Size_in_GB** | Size of the Volume in GB. | `5` |

## Multiple backend provisioning

The following sample cinder.conf configuration files detail the multiple backend configuration:

```
[cloudbyte-gold]
volume_driver = cinder.volume.drivers.cloudbyte.cloudbyte.CloudByteISCSIDriver
volume_backend_name = cloudbyte-gold
san_ip = 172.16.51.5
san_password = na
```

```
tsm_name = VSM1
cb_account_name = Acc1
cb_apikey = edTzOA0RUsXfPTCJcdt1PGa9HaqLMW0gtkhT7iIZ348T6iwhsQc4p7E81bqXwNOyubmUeW84nS-
ICi2Xg_qR9w
confirm_vol_sleep_counter = 10
confirm_vol_sleep_interval = 5
add_qosgroup=latency:15,graceallowed:true,iopscontrol:true,memlimit:0,tpcontrol:false,
throughput:0,iops:100,networkspeed:0
create_volume=compression:off,deduplication:off,blocklength:512B,sync:always,
protocoltype:ISCSI,recordsize:32k
[cloudbyte-silver]
volume_driver = cinder.volume.drivers.cloudbyte.cloudbyte.CloudByteISCSIDriver
volume_backend_name = cloudbyte-silver
san_ip = 172.16.51.5
san_password = na
tsm_name = VSM2
cb_account_name = Acc2
cb_apikey = edTzOA0RUsXfPTCJcdt1PGa9HaqLMW0gtkhT7iIZ348T6iwhsQc4p7E81bqXwNOyubmUeW84nS-
ICi2Xg_qR9w
confirm_vol_sleep_counter = 10
confirm_vol_sleep_interval = 5
add_qosgroup=latency:15,graceallowed:true,iopscontrol:true,memlimit:0,tpcontrol:false,
throughput:0,iops:50,networkspeed:0
create_volume=compression:off,deduplication:off,blocklength:512B,sync:always,
protocoltype:ISCSI,recordsize:32k
...
...
...
enabled_backends = cloudbyte-silver, cloudbyte-gold
```

The following sample procedures (performed on OpenStack CLI) illustrate how to connect the volume type with OpenStack:

1. Run the following command to create a Volume type:`cinder --os-username admin --os-tenant-name admin type-create cb_gold`
2. Run the following command to connect the Volume type with back-end:`cinder --os-username admin --os-tenant-name admin type-key cb_gold set volume_backend_name=cloudbyte-gold`
3. Run the following command to verify the successful connection of the Volume type:`cinder --os-username admin --os-tenant-name admin extra-specs-list`
4. Restart the following services:
    - `cinder-api`
    - `cinder-scheduler`
    - `cinder-volume services`

## Plug-in tips

### How will I know if the plug-in is successfully installed?

1. Go to `/var/log/cinder/scheduler.log.`
2. At command prompt, enter the following command: `vi scheduler.log`
3. In the log, search for the following line: `Received volume service`

If the search output has details about `enabled_backends` that you specified while installing the plug-in, it implies that the plug-in has been successfully installed.

### What should I do if I encounter an error while verifying if the installation is successful?

Check the following Cinder logs to find the cause:

- api.log
- volume.log
- scheduler.log

## Limitations

The current release supports only iSCSI protocol. Future releases will support multiple protocols.