

# High Level Design of Cinder iSCSI Driver

March, 2016

<b>Document Name</b>	High Level Design of Cinder iSCSI Driver	<b>Customer Name</b>	iXsystems
<b>Prepared by</b>	Jitendra Pawar	<b>Version</b>	1.0
<b>Approved by</b>	Sunu Engineer	<b>Date</b>	18-03-2016



Code for all Seasons

## Contents

1. [Introduction](#)
2. [Scope](#)
3. [Intended Audience](#)
4. [High Level Design](#)
5. [Cinder Operation Implementation Approach](#)
6. [Volume statistics details](#)
7. [References](#)

## Introduction

This document provides design details for implementing iXsystems Cinder volume driver for backend storage appliance.

## Scope

The design details explained in this document are related to the OpenStack Kilo version cinder driver specifications but are also expected to be applicable to the latest OpenStack versions Icehouse and later. The iXsystems iSCSI cinder driver should work with 9.10 version of FreeNAS.

Note that this document only describes Cinder iSCSI driver related information. In order to provide shared file system service for OpenStack one needs to write Manila driver. Description about shared file system and Manila driver is out of scope of this project.

## Intended Audience

This document is mainly for developers.

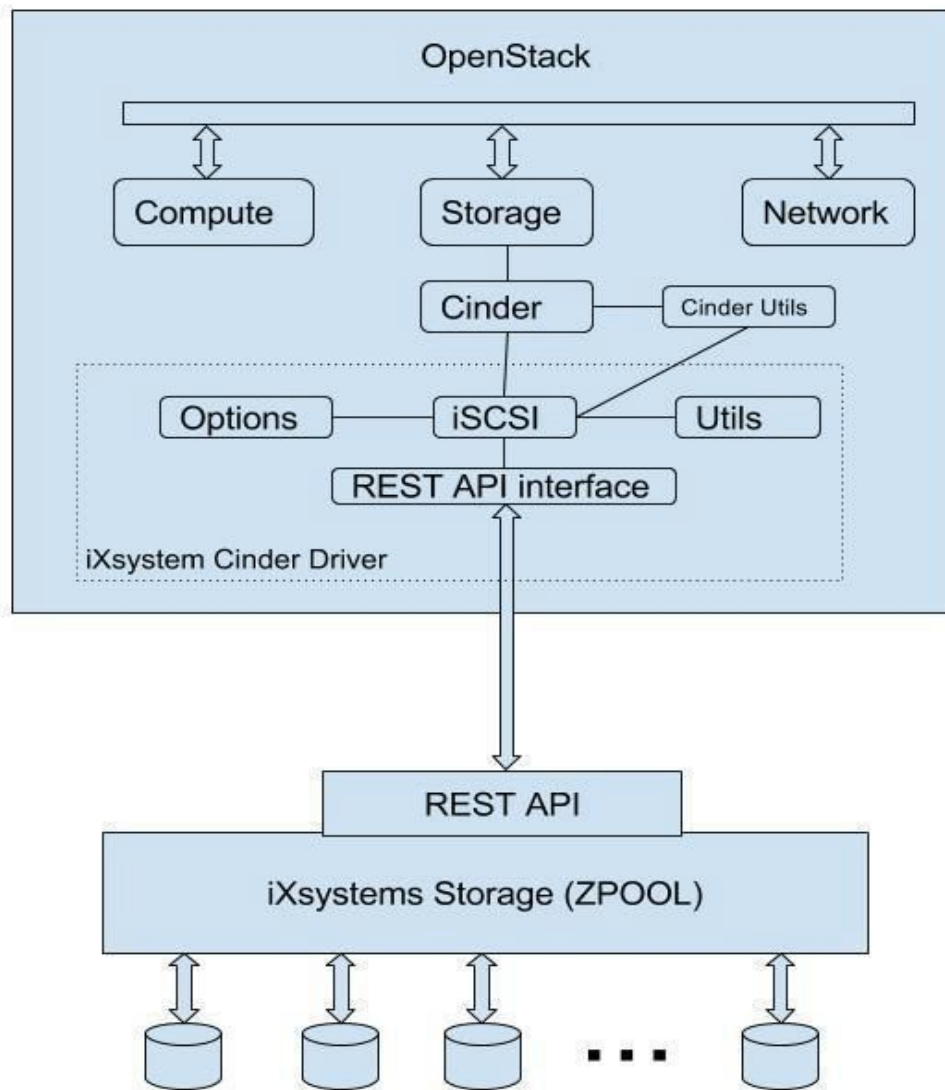
## High Level Design

The zvol on FreeNAS is volume space made available as a device instead of filesystem. The iSCSI subsystem can use these device for exporting as LUN targets. These LUN's can be accessed remotely using RESTful API.

Every Cinder iSCSI volume is represented by a single zvol in a predefined iXsystems storage namespace. For every new volume the driver creates an iSCSI target that is used to access it from compute hosts.

A pool and an enclosing namespace must be created for all iSCSI volumes to be accessed through the volume driver.

Below diagram shows high level components and architecture of iSCSI cinder driver plugin for iXsystems storage.



The main components in the driver are-

1. Cinder iSCSI driver

This component implements the Cinder operations for iXsystems storage array.

2. REST API interface

This component is responsible for communicating with to backend iXsystems storage array using RESTful calls.

3. Configuration (Options)

This component is responsible for holding configurations required for iSCSI cinder driver for example- Communication options, transport options, authentication options, iXstems storage options, provisioning options etc.

#### 4. Utilities

This is set of utilities required for the iSCSI driver. It mostly responsible for URL parsing and string manipulation related functionalities.

## Cinder Operation Implementation Approach

The below chart shows the implementation approach for OpenStack Kilo release with available API's provided by FreeNAS or Cinder utilities. Please note that the API's can differ across all versions of OpenStack. Like the API's set supported for OpenStack Kilo version may not same for other OpenStack versions like Havana, Icehouse, Juno, Liberty etc. as listed [here](#). The 12 Cinder operations are applicable from OpenStack release Icehouse onwards.

Sr. No.	Operation Name	Description	Solution
1	Volume Create	Creates a volume. A Volume is represented by a zvol	Use FreeNAS REST API- POST /api/v1.0/storage/volume/tank/zvols/ HTTP/1.1
2	Volume Delete	Deletes a volume	Use FreeNAS REST API- DELETE /api/v1.0/storage/volume/tank/zvols/myzvol/ HTTP/1.1
3	Volume Attach	Attaches a volume to an openstack instance.	In order to attach a volume to a nova instance, 1) create a connector using bricks utility which is defined in <b>cinder/brick/initiator/connector.py</b> 2) call connect_volume method defined bricks utility <b>connect_volume(self, connection_properties)</b> The connection_properties describes the information needed by the specific protocol to use to make the connection
4	Volume Detach	Detach a volume from an openstack instance.	In order to detach a volume to a nova instance, 1) create a connector using bricks utility which is defined in <b>cinder/brick/initiator/connector.py</b> 2) call disconnect_volume method defined bricks utility <b>disconnect_volume(self, connection_properties, device_info):</b>

			The connection_properties are the same as from connect_volume. The device_info is returned from connect volume
5	Snapshot Create	Creates a volume snapshot, which is a point-in-time, complete copy of a volume.	Use FreeNAS REST API- POST /api/v1.0/storage/snapshot/ HTTP/1.1
6	Snapshot Delete	Deletes a volume snapshot	Use FreeNAS REST API- DELETE /api/v1.0/storage/snapshot/(string: id)
7	Create Volume from Snapshot	Create a new Volume from existing snapshot	To create a new volume from an existing snapshot perform following steps- 1) Create a new volume with a specified parameters 2) Copy snapshot to a newly created volume using copy_volume utility defined in cinder/volume/utils.py
8	Copy Image to Volume	Create bootable volumes directly from glance images.  <a href="https://wiki.openstack.org/wiki/CreateVolumeFromImage">https://wiki.openstack.org/wiki/CreateVolumeFromImage</a>	Steps to copy an image to a volume- 1) Attach the volume 2) Fetch the image from image_service and write it to the volume using fetch_to_raw cinder utility in cinder/image/image_utils.py 3) Detach the volume
9	Copy Volume to Image	Create an image of a volume.	Steps to copy volume to an image- 1) Attach the volume 2) Copy the volume to the specified image using upload_volume util in cinder/image/image_utils.py 3) Detach the volume
10	Clone Volume	Makes a new volume from an existing volume	Steps to clone a volume- 1) create a temporary snapshot of a source volume 2) create a destination volume using source volume parameters 3) copy volume using cinder copy_volume util 4) delete temporary snapshot 5) export newly cloned volume
11	Extend Volume	Increase the size of existing zvol	There is no REST API available for this operation. An issue is present for the same- <a href="https://bugs.freenas.org/issues/14136">https://bugs.freenas.org/issues/14136</a>
12	Get Volume Stats	Show Volume section from <a href="#">this</a> link	Use FreeNAS REST API-

		describes about Volume statistics-	GET /api/v1.0/storage/volume/tank/status/ HTTP/1.1
--	--	---------------------------------------	---

## Volume statistics details

The below table describes the suggested default for different parameters required for providing volume statistics. iXsystem can provide alternate values if the values mentioned in below table are not as per expectations.

Sr. No.	Volume Statistics	Details / Availability	Comments
1	Driver Version	1.0	This is suggested version, Please confirm or provide alternate version
2	Free Capacity in GB	From GET STATUS API response	
3	Reserved Percentage	% of backend capacity reserved to avoid over provisioning	Not available.
4	Storage Protocol	iSCSI	This will be hard-coded
5	Total Capacity in GB	From GET STATUS API response	
6	Vendor Name	iXsystems Storage Array	This is suggested name, Please confirm or provide alternate name
7	Volume Backend Name	From GET STATUS API response	

## References

1. Cinder Driver Features  
<https://github.com/openstack/cinder/blob/master/doc/source/devref/drivers.rst>
2. FreeNAS User Guide 9.10  
[http://doc.freenas.org/9.3/freenas\\_storage.html](http://doc.freenas.org/9.3/freenas_storage.html)
3. OpenStack Wiki  
[https://wiki.openstack.org/wiki/Main\\_Page](https://wiki.openstack.org/wiki/Main_Page)



Code for all Seasons