reduxio

Integrating Reduxio TimeOS™ with
OpenStack

For more information, refer to Reduxio website at http://www.reduxio.com.
If you have comments about this documentation, submit your feedback to docs@reduxio.com.

Reduxio Systems, Inc.
111 Pine Avenue
South San Francisco, CA, 94080
United States
www.reduxio.com

Contents

Preface

Reduxio Solution Guide for OpenStack describes the Reduxio solution for available for customers who are building an OpenStack cloud using Reduxio volumes. It provides instructions for the installation, configuration and management of Reduxio HX Series storage systems in an OpenStack environment.

Intended Audience

Reduxio Solution Guide for OpenStack is intended for anyone who needs to configure Reduxio storage systems for hosting an OpenStack cloud.

This information is written for experienced system and storage administrators.

Technical Support

For additional support, refer to https://support.reduxio.com.

Overview

Introduction

OpenStack is a software suite that provides a single pane management for private and public clouds. It controls pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. OpenStack is developed by the open-source community.

The Reduxio TimeOS™ is integrated with OpenStack using a Cinder driver. This driver enables Cinder volume and snapshot management, and brings BackDating™ recovery to any second to OpenStack environments.

The Business Challenge

Customers deploying cloud environments either for internal consumption or as a hosting platform for their own customers are looking at OpenStack as a platform that provides better agility and automation than other virtualization and orchestration solutions.

There are many challenges driving this wave of change:

- **Service availability** – Private and public cloud environments are expected to be available 100% of the time and should be very reliable. Data outage or data loss will have great impact on the cloud SLAs.
- **Performance** – The mass-consolidation enabled by OpenStack can cause some major performance issues. Those issues will directly impact end-users and applications.
- **High costs** – Cost is always a key factor. Selecting a storage platform or a solution with a high total cost of ownership will waste valuable resources. Not only hardware and software efficiency is required for an effective solution, but also simplicity of storage and data management.

Reduxio Solution for OpenStack

Overview

Reduxio Storage provides a cost-effective storage infrastructure solution for OpenStack environments. The solution consists of the following components:

- **Reduxio Storage System** – The Reduxio flash storage arrays, based on Reduxio's TimeOS™ storage operating system allow you to recover application data to any second in the past, eliminates most of the complexity associated with managing storage, and provides exceptional performance and efficiency, far exceeding anything available today.
  It provides considerable higher performance and more effective capacity for OpenStack instanced than traditional SAN, and provides instance recoverability to any second in the past without upfront administration.
- **Reduxio Cinder driver** – Reduxio TimeOS is integrated with OpenStack using a Cinder block storage driver. The driver supports volume provisioning, snapshots as well as the use of BackDating.

Solution Benefits

Deploying OpenStack on Reduxio offers many compelling benefits:

- **Performance scalability** – Consolidating many instances onto a single system creates a storage bottleneck. The aggregated IOs from all compute hosts are sent to the storage concurrently, in a highly random workload. For example, a single instance may produce a thousand IOPS, but a hundred of those will already requires a storage system that can handle 100,000 IOPS.
- **Storage efficiency –** Reduxio NoDup™ in-line dedupe and compression dramatically reduces the overall capacity requirements of an OpenStack environment.

- **Instance recoverability** – In addition to standard volume snapshots, the Reduxio Cinder driver supports the recovery of Cinder volumes to any second in the past.

Solution Architecture

Overview

The Reduxio Cinder volume driver is installed and provides services to the OpenStack environment. Each OpenStack instance (i.e. virtual machine) is allocated one or more Reduxio volume. Reduxio volumes are globally "NoDuped" – deduped and compressed, expanding the total usable capacity of the system. Operating system and application binaries are stored once and kept in memory. This leaves ample room – almost the entire system capacity - available for application and user data. In addition, Reduxio volumes support the use of Cinder snapshots, as well as the use of recovery to any second using BackDating™.

The solution can be managed using:

OpenStack management tools	The OpenStack administrator can manage Reduxio volumes using the native Horizon web interface and the various command-line utilities to perform storage management functions: • Create and delete volumes. • Create and delete snapshots. • Clone volumes from "now". • Clone volumes from past timestamps. Managing Cinder volumes from the OpenStack management interfaces is the recommended choice as less steps are required to perform common tasks.
Reduxio Storage Manager and ReduxioCLI	Hosts, volumes and assignments can be created and managed using the Reduxio management interfaces. Note: Reduxio Cinder driver v1.x relies on the volume name and description fields. Do not rename volumes created by the driver, nor modify the Reduxio volume description field.

Supported Configurations

The Reduxio Cinder driver was tested in the following OpenStack configurations:

Reduxio TimeOS™	Reduxio TimeOS v2.7 and higher. Note: Reduxio StorKit for OpenStack v1.x requires SSH access to the Reduxio system. A future version of the driver will support the Reduxio REST API.
OpenStack	• OpenStack DevStack Ocata running on Ubuntu 16.04. • Red Hat OpenStack. Note: For support of older OpenStack releases, contact Reduxio.
Operating System	• Ubuntu 16.04. • Red Hat Enterprise Linux 7.0.
Virtualization type	Tested with QEMU.

Reduxio Cinder Driver

Driver Features

The Reduxio Cinder Driver v1.0 provides the following capabilities:

- `cinder create` – Create and delete volumes.
- `cinder snapshot-create` – Create Reduxio bookmark.
- `cinder create --snapshot-id` – Create clone from timestamp – BackDating™ - Clone based on past timestamp.
- `cinder manage` – Import an existing volume.

This driver is supported with OpenStack Liberty and higher, and will be released together with the Newton release.

Current Limitations

- No support for 512 bytes block size (all volumes created with default 4KB block size).
- No consistency group support.
- The driver stores the Cinder volume id in the Reduxio volume description field. It is important not to update the OpenStack volumes description fields.

Configuration

The following section describes how to install the Reduxio Cinder driver and perform an initial configuration.

Installing the Cinder Driver

Note:

Starting with the Ocata release, there is no need to install the Cinder driver since it is provided with the official OpenStack release. The following instructions are provided for legacy installations.

First, download the Cinder volume and install it in the OpenStack hosts. Note that steps 1-2 are typically not required, since the Reduxio Cinder driver is supplied with the OpenStack build:

1. Download the Reduxio Cinder driver.	To download the driver from the Cinder Github repository: 1. Navigate to [Reduxio Cinder driver on Github](#). 2. Download all the files.

2. Copy the driver to each OpenStack compute host.	To copy the driver in Ubuntu-based installations:
	
	`# mkdir /opt/stack/cinder/cinder/volume/drivers/reduxio`
	`# cp * /opt/stack/cinder/cinder/volume/drivers/reduxio/`
	`# cd /opt/stack/cinder/cinder/volume/drivers`
	`# ls -laR reduxio`
	
	To copy the driver in Red Hat OpenStack installations:
	
	`# mkdir /usr/lib/python2.7/site-packages/cinder/volume/drivers/reduxio`
	`# cp * /usr/lib/python2.7/site-packages/cinder/volume/drivers/reduxio/`
	`# cd /usr/lib/python2.7/site-packages/cinder/volume/drivers`
	`# ls -laR reduxio`

Now, configure the Cinder driver:

3.	Update the Cinder configuration file.	To update /etc/cinder/cinder.conf using the openstack-config command:

3. Update the Cinder configuration file.	To update /etc/cinder/cinder.conf using the openstack-config command:

```
# openstack-config --set /etc/cinder/cinder.conf DEFAULT
default_volume_type REDUXIO_VOLUME_TYPE
# openstack-config --set /etc/cinder/cinder.conf DEFAULT #
enabled_backends REDUXIO_VOLUME_TYPE
# openstack-config --set /etc/cinder/cinder.conf
REDUXIO_VOLUME_TYPE san_ip mango
# openstack-config --set /etc/cinder/cinder.conf
REDUXIO_VOLUME_TYPE san_login rdxadmin
# openstack-config --set /etc/cinder/cinder.conf
REDUXIO_VOLUME_TYPE san_password rdxadmin
# openstack-config --set /etc/cinder/cinder.conf
REDUXIO_VOLUME_TYPE volume_driver
cinder.volume.drivers.reduxio.rdx_iscsi_driver.ReduxioISCSIDriver
# openstack-config --set /etc/cinder/cinder.conf
REDUXIO_VOLUME_TYPE volume_backend_name REDUXIO_VOLUME_TYPE
```

Note: Replace mango with the Reduxio system's management IP or host name

To manually update /etc/cinder/cinder.conf:

1. Add the following lines to /etc/cinder/cinder.conf in the [DEFAULT] section:

```
enabled_backends = REDUXIO_VOLUME_TYPE
default_volume_type = REDUXIO_VOLUME_TYPE
```

2. Add the following section to /etc/cinder/cinder.conf. Replace mango with the Reduxio system's management IP or host name:

```
[REDUXIO_VOLUME_TYPE]
san_ip = mango
san_login = rdxadmin
san_password = rdxadmin
volume_driver =
cinder.volume.drivers.reduxio.rdx_iscsi_driver.ReduxioISCSIDriver
volume_backend_name = REDUXIO_VOLUME_TYPE
```

---

4. Restart the cinder services.	In Ubuntu-based installations:

```
$ sudo service openstack-cinder-api restart
$ sudo service openstack-cinder-scheduler restart
$ sudo service openstack-cinder-volume restart
```

In Red Hat OpenStack installations:

```
$ sudo systemctl restart openstack-cinder-api.service
$ sudo systemctl restart openstack-cinder-scheduler.service
$ sudo systemctl restart openstack-cinder-volume.service
$ sudo systemctl restart openstack-cinder-backup.service
```

---

5. Confirm that Cinder services are running.	`$ cinder service-list`

Configure Multipathing

The Reduxio Cinder driver supports multipathing. The driver receives the list of iSCSI port IP addresses from the Reduxio system and provides it to Nova which then invokes multiple iSCSI sessions towards the system when a Nova image is attached to a Reduxio volume.

Multipath configuration with Reduxio requires the following:

- Enable multipathing in each Nova compute host.
- Configure the iSCSI initiator, device rules and multipath configuration files according to Reduxio best practices.

To enable multipathing in Nova:

1. Enable multipathing in the Nova configuration file.	Add the following setting to the [libvirt] section to each compute host's /etc/nova/nova.conf:
	
	`[libvirt]`
	`iscsi_use_multipath = True`

2. Restart the nova-compute service on each relevant compute host.	In Ubuntu-based installations:
	
	`$ sudo service openstack-nova-compute restart`
	
	In Red Hat OpenStack installations:
	
	`$ sudo systemctl restart openstack-nova-compute.service`

3. Confirm that Nova services are running.	`$ nova service-list`

Note: To enable multipathing in an environment with existing single-path connections:

1. Enable multipathing using the procedure above.
2. Create an additional volume using Cinder.
3. Attach the new volume to an instance.
4. The new volume and all prior sessions in that host will be multipathed.

To configure the iSCSI initiator and multipathing settings, follow the instructions in the next sections. For the most updated information, refer to the Reduxio Interoperability documentation on the Reduxio Support Portal.

Red Hat / CentOS / Oracle Linux

Install the Host Attach Toolkit
Both Red Hat, CentOS and Oracle Linux operating systems provide the Open-iSCSI initiator. Typically, this initiator has to be manually installed.

To install the initiator and multipathing software:

1. Copy the iSCSI configuration files to the Linux host	Copy the following files to the host: - iscsid.conf - 99-reduxio.rules - multipath.conf
2. Install the required software	To install the Open-iSCSI initiator utilities, multipathing software, and the lsscsi command: `$ sudo yum -y install iscsi-initiator-utils device-mapper device-mapper-event-libs device-mapper-multipath lsscsi`

3.	Update the initiator configuration	Backup the existing initiator configuration file:
		
		`$ sudo cp /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.orig`
		
		Copy or update the initiator configuration file:
		
		`$ sudo cp iscsid.conf /etc/iscsi/iscsid.conf`

4.	Copy the Reduxio device rules	Copy the device rule file: `$ sudo cp 99-reduxio.rules /etc/udev/rules.d/99-reduxio.rules`

5.	Copy the multipathing configuration file	Backup the existing multipathing configuration file:
		
		`$ sudo cp /etc/multipath.conf /etc/multipath.conf.orig`
		Copy the Reduxio multipathing configuration file:
		
		`$ sudo cp multipath.conf /etc/multipath.conf`

Configure the iSCSI Initiator
To configure the initiator and multipathing software:

| 1. | Start the initiator and multipathing | ```
$ sudo chkconfig iscsi on ; service iscsi start
Starting iscsi daemon:                              [  OK  ]
$ sudo chkconfig multipathd on ; service multipathd start
Starting multipathd daemon:                         [  OK  ]
$ sudo chkconfig --list iscsi
iscsi         0:off  1:off  2:on   3:on   4:on   5:on   6:off
$ sudo chkconfig --list multipathd
multipathd 0:off  1:off  2:on   3:on   4:on   5:on   6:off
``` |
|---|---|---|
| 2. | Identify the initiator's IQN | `$ cat /etc/iscsi/initiatorname.iscsi` |
| 3. | Create a host and a volume assigned to it | Follow the instructions in the Configuration Storage chapter in the Administration Guide to create a host for this Linux Server, and a volume assigned to it. |
| 4. | Configure the Reduxio system as a new target | Use iscsiadm to discover the system using one of the four iSCSI port IPs:<br><br>`$ sudo iscsiadm --mode discovery -t st -p 172.168.1.121`<br><br>Discovery replies with the Reduxio iSCSI target name:<br>`172.168.1.121:3260,1 iqn.2013-12.com.reduxio:`<br>`af4032ff0032000e`<br><br>Perform a login to all Reduxio iSCSI ports:<br><br>```
$ sudo iscsiadm -m node -l
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260]
(multiple)
Login to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
successful.
``` |

		Login to [iface: default, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260] successful. Login to [iface: default, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260] successful. Login to [iface: default, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260] successful.
5.	Setup automatic login at boot	Run the following command: `$ sudo iscsiadm -m node -T iqn.2013-12.com.reduxio:ff4032ff0032000e -p 172.168.1.121 -o update -n node.startup -v automatic`

6.	Review the connectivity and multipathing status	To list the multipathd effective settings:
		
		`$ sudo multipathd show config`
		
		To list the active sessions:
		
		`$ sudo iscsiadm -m session`
		`tcp: [1] 172.168.1.121:3260,1 iqn.2013-`
		`12.com.reduxio:af4032ff0032000e (non-flash)`
		`tcp: [2] 172.168.1.122:3260,1 iqn.2013-`
		`12.com.reduxio:af4032ff0032000e (non-flash)`
		`tcp: [3] 172.168.1.123:3260,1 iqn.2013-`
		`12.com.reduxio:af4032ff0032000e (non-flash)`
		`tcp: [4] 172.168.1.124:3260,1 iqn.2013-`
		`12.com.reduxio:af4032ff0032000e (non-flash)`
		
		To review the multipathing status:
		
		`$ sudo multipath -ll`
		`mpatha (36f4032f0003500000000000000008ab) dm-3 REDUXIO,TCAS`
		`size=500G features='1 queue_if_no_path' hwhandler='1 alua'`
		`wp=rw`
		`|-+- policy='round-robin 0' prio=130 status=active`
		`| `- 10:0:0:120 sde 8:64 active ready running`
		`` `-+- policy='round-robin 0' prio=10 status=enabled``
		`  |- 4:0:0:120  sdd 8:48 active ready running`
		`  `- 3:0:0:120 sdc 8:32 active ready running`
		
		Any Reduxio volume assigned to this host should be detected as a new block device.
		
		Identify the new devices in /var/log/messages, fdisk -l or lsscsi commands.

To list the open sessions:

`$ sudo iscsiadm -m session --show`

To rescan an existing session:

```
$ sudo iscsiadm -m session --rescan
Rescanning session [sid: 1, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal:
172.168.1.121,3260]
Rescanning session [sid: 1577, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal:
172.168.1.122,3260]
Rescanning session [sid: 2, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal:
172.168.1.123,3260]
Rescanning session [sid: 8, target: iqn.2013-12.com.reduxio:af4032ff0032000e, portal:
```

```
172.168.1.124,3260]
$ multipathd reconfigure
```

To manage the Red Hat Enterprise Linux 6 iSCSI initiator, refer to [Red Hat Storage Management Guide](#).

Ubuntu Server

Install the Host Attach Toolkit

The Ubuntu Server Linux operating systems provide the Open-iSCSI initiator. Typically, this initiator has to be manually installed.

To install the initiator and multipathing software:

1.	Copy the iSCSI configuration files to the Linux host	Copy the following files to the host: • iscsid.conf • 99-reduxio.rules • multipath.conf
2.	Install the required software	To install the Open-iSCSI initiator utilities, multipathing software, and the lsscsi command: `$ sudo apt-get -y install open-iscsi open-iscsi-utils multipath-tools lsscsi`

3.	Update the initiator configuration	Backup the existing initiator configuration file:
		
		 `$ sudo cp /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.orig`
		
		 Copy or update the initiator configuration file:
		
		 `$ sudo cp iscsid.conf /etc/iscsi/iscsid.conf`

4.	Copy the Reduxio device rules	Copy the device rule file: `$ sudo cp 99-reduxio.rules /etc/udev/rules.d/99-reduxio.rules`

5.	Copy the multipathing configuration file	Backup the existing multipathing configuration file:
		
		 `$ sudo cp /etc/multipath.conf /etc/multipath.conf.orig`
		 Copy the Reduxio multipathing configuration file:
		
		 `$ sudo cp multipath.conf /etc/multipath.conf`

Configure the iSCSI Initiator

To configure the initiator and multipathing software:

1.	Restart the initiator and multipathing	`$ sudo service open-iscsi restart`
		 ` * Starting iSCSI initiator service iscsid          [ OK ]`
		 ` * Setting up iSCSI targets`
		 `iscsiadm: No records found                         [ OK ]`
		 ` * Mounting network filesystems                    [ OK ]`
		 `$ service multipath-tools restart`
		 `* Stopping multipath daemon multipathd             [ OK ]`
		 `* Starting multipath daemon multipathd             [ OK ]`

2.	Identify the initiator's IQN	`$ cat /etc/iscsi/initiatorname.iscsi`
3.	Create a host and a volume assigned to it	Follow the instructions in the Configuration Storage chapter in the Administration Guide to create a host for this Linux Server, and a volume assigned to it.

4.	Configure the Reduxio system as a new target	Use iscsiadm to discover the system using one of the four iSCSI port IPs:
		
		``
		$ sudo iscsiadm --mode discovery -t st -p 172.168.1.121
		``
		
		Discovery replies with the Reduxio iSCSI target name:
		``
		172.168.1.121:3260,1 iqn.2013-12.com.reduxio:
		af4032ff0032000e
		``
		
		Perform a login to all Reduxio iSCSI ports:
		
		``
		$ sudo iscsiadm -m node -l
		Logging in to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
		(multiple)
		Logging in to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260]
		(multiple)
		Logging in to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260]
		(multiple)
		Logging in to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260]
		(multiple)
		Login to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
		successful.
		Login to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260]
		successful.
		Login to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260]
		successful.
		Login to [iface: default, target: iqn.2013-
		12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260]
		successful.
		``

5.	Setup automatic login at boot	Run the following command:
		
		``
		$ sudo iscsiadm -m node -T iqn.2013-
		12.com.reduxio:ff4032ff0032000e -p 172.168.1.121 -o update -n
		node.startup -v automatic
		``

6.	Review the connectivity and multipathing status	To list the multipathd effective settings: ``` $ sudo multipathd show config ``` To list the active sessions: ``` $ sudo iscsiadm -m session tcp: [1] 10.46.199.22:3260,1 iqn.2013- 12.com.reduxio:ff4032ff0035000e (non-flash) tcp: [2] 10.46.199.11:3260,1 iqn.2013- 12.com.reduxio:ff4032ff0035000e (non-flash) tcp: [3] 10.64.199.11:3260,1 iqn.2013- 12.com.reduxio:ff4032ff0035000e (non-flash) tcp: [4] 10.64.199.22:3260,1 iqn.2013- 12.com.reduxio:ff4032ff0035000e (non-flash) ``` To review the multipathing status: ``` $ multipath -ll mpatha (36f4032f000350000000000000000008ab) dm-3 REDUXIO,TCAS size=500G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw |-+- policy='round-robin 0' prio=130 status=active | `- 10:0:0:120 sde 8:64 active ready running `-+- policy='round-robin 0' prio=10 status=enabled ```

```
|- 4:0:0:120  sdd 8:48 active ready running
`- 3:0:0:120  sdc 8:32 active ready running
```

Any Reduxio volume assigned to this host should be detected as a new block device.

Identify the new devices in `/var/log/messages`, `fdisk -l` or `lsscsi` commands.

To manage the Ubuntu Server iSCSI initiator, refer to https://help.ubuntu.com/lts/serverguide/iscsi-initiator.html

To manage the Ubuntu Server DM-Multipath, refer to https://help.ubuntu.com/lts/serverguide/dm-multipath-chapter.html

Debian

Install the Host Attach Tool

The Ubuntu Server Linux operating systems provide the Open-iSCSI initiator. Typically, this initiator has to be manually installed.

To install the initiator and multipathing software:

1. Copy the iSCSI configuration files to the Linux host	Copy the following files to the host: • iscsid.conf • 99-reduxio.rules • multipath.conf
2. Install the required software	To install the Open-iSCSI initiator utilities, multipathing software, and the lsscsi command: `$ sudo apt-get -y install open-iscsi open-iscsi-utils multipath-tools lsscsi`

3. Update the initiator configuration	Backup the existing initiator configuration file:
	`$ sudo cp /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.orig`
	
	Copy or update the initiator configuration file:
	`$ sudo cp iscsid.conf /etc/iscsi/iscsid.conf`

4. Copy the Reduxio device rules	Copy the device rule file: `$ sudo cp 99-reduxio.rules /etc/udev/rules.d/99-reduxio.rules`

5. Copy the multipathing configuration file	Backup the existing multipathing configuration file:
	`$ sudo cp /etc/multipath.conf /etc/multipath.conf.orig`
	Copy the Reduxio multipathing configuration file:
	`$ sudo cp multipath.conf /etc/multipath.conf`

Configure the iSCSI Initiator

To configure the initiator and multipathing software:

| 1. Restart the initiator and multipathing | ```
$ sudo service open-iscsi restart
 * Starting iSCSI initiator service iscsid          [ OK ]
 * Setting up iSCSI targets
iscsiadm: No records found                          [ OK ]
 * Mounting network filesystems                      [ OK ]
``` |
|---|---|

| | | ```
$ sudo service multipath-tools restart
* Stopping multipath daemon multipathd        [ OK ]
* Starting multipath daemon multipathd        [ OK ]
``` |
| 2. | Identify the initiator's IQN | ```
$ cat /etc/iscsi/initiatorname.iscsi
``` |
| 3. | Create a host and a volume assigned to it | Follow the instructions in the Configuration Storage chapter in the Administration Guide to create a host for this Linux Server, and a volume assigned to it. |
| 4. | Configure the Reduxio system as a new target | Use iscsiadm to discover the system using one of the four iSCSI port IPs:<br><br>```
$ sudo iscsiadm --mode discovery -t st -p 172.168.1.121
```<br><br>Discovery replies with the Reduxio iSCSI target name:<br>```
172.168.1.121:3260,1 iqn.2013-12.com.reduxio:
af4032ff0032000e
```<br><br>Perform a login to all Reduxio iSCSI ports:<br><br>```
$ sudo iscsiadm -m node -l
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260]
(multiple)
Logging in to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260]
(multiple)
Login to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.121,3260]
successful.
Login to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.122,3260]
successful.
Login to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.123,3260]
successful.
Login to [iface: default, target: iqn.2013-
12.com.reduxio:af4032ff0032000e, portal: 172.168.1.124,3260]
successful.
``` |
| 5. | Setup automatic login at boot | Run the following command:<br><br>```
$ sudo iscsiadm -m node -T iqn.2013-
12.com.reduxio:ff4032ff0032000e -p 172.168.1.121 -o update -n
node.startup -v automatic
``` |
| 6. | Review the connectivity and multipathing status | To list the multipathd effective settings:<br><br>```
$ multipathd show config
```<br><br>To list the active sessions:<br><br>```
$ iscsiadm -m session
tcp: [1] 10.46.199.22:3260,1 iqn.2013-
12.com.reduxio:ff4032ff0035000e (non-flash)
tcp: [2] 10.46.199.11:3260,1 iqn.2013-
12.com.reduxio:ff4032ff0035000e (non-flash)
tcp: [3] 10.64.199.11:3260,1 iqn.2013-
12.com.reduxio:ff4032ff0035000e (non-flash)
``` |

```
tcp: [4] 10.64.199.22:3260,1 iqn.2013-
12.com.reduxio:ff4032ff0035000e (non-flash)
```

To review the multipathing status:

```
$ multipath -ll
mpatha (36f4032f000350000000000000008ab) dm-3 REDUXIO,TCAS
size=500G features='1 queue_if_no_path' hwhandler='1 alua'
wp=rw
|-+- policy='round-robin 0' prio=130 status=active
| `- 10:0:0:120 sde 8:64 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 4:0:0:120  sdd 8:48 active ready running
  `- 3:0:0:120  sdc 8:32 active ready running
```

Any Reduxio volume assigned to this host should be detected as a new block device.

Identify the new devices in **/var/log/messages**, **fdisk -l** or **lsscsi** commands.

Update Cinder Quotas

It is important to update Cinder quotas to values that correspond to Reduxio maximum number of volumes (1,000 volumes in v2.6 and higher), and the maximum volume size (100,240 gigabytes in v2.6 and higher).

To list the existing quotas:

```
$ openstack quota show demo
+-------------------------------+-------------------------------+
| Field                         | Value                         |
+-------------------------------+-------------------------------+
| backup_gigabytes              | 1000                          |
| backups                       | 10                            |
| cores                         | 20                            |
| fixed-ips                     | -1                            |
| floating-ips                  | 50                            |
| gigabytes                     | 1000                          |
| gigabytes_REDUXIO_VOLUME_TYPE | -1                            |
| injected-file-size            | 10240                         |
| injected-files                | 5                             |
| injected-path-size            | 255                           |
| instances                     | 10                            |
| key-pairs                     | 100                           |
| network                       | 10                            |
| per_volume_gigabytes          | -1                            |
| port                          | 50                            |
| project                       | 07374454338540cd857623075d086f9f |
| properties                    | 128                           |
| ram                           | 51200                         |
| rbac_policy                   | 10                            |
| router                        | 10                            |
| secgroup-rules                | 100                           |
| secgroups                     | 10                            |
| server_group_members          | 10                            |
| server_groups                 | 10                            |
| snapshots                     | 10                            |
| snapshots_REDUXIO_VOLUME_TYPE | -1                            |
| subnet                        | 10                            |
| subnetpool                    | -1                            |
| volumes                       | 10                            |
| volumes_REDUXIO_VOLUME_TYPE   | -1                            |
+-------------------------------+-------------------------------+
```

To set the recommended quotas:

```
$ openstack quota set --per-volume-gigabytes --gigabytes 102400 --volumes 1000 --snapshots
--instances 1000 demo
```

Storage Provisioning

To create a new volume, run the following from the OpenStack controller:

```
$ cinder create --volume-type REDUXIO_VOLUME_TYPE --display-name vol1 100
```

To list the volumes:

```
$ cinder list
```

Network Configuration

Each host must have connectivity to the Reduxio iSCSI ports. To configure eth1 as a bridge that provides connectivity to this network, run the following on the compute host:

```
$ sudo ovs-vsctl add-br br-ex-iscsi
$ sudo ovs-vsctl add-port br-ex-iscsi eth5
$ neutron net-create ext-net-iscsi --shared --router:external=True
$ neutron subnet-create ext-net-iscsi --name ext-subnet --allocation-pool
start=10.17.56.1,end=10.17.56.10 --disable-dhcp 10.0.0.0/8
$ neutron router-create iscsi-router
$ neutron router-gateway-set iscsi-router ext-net-iscsi
$ sudo ifconfig br-ex-iscsi 10.17.56.100 netmask 255.0.0.0 mtu 9000 up
```

To review the setup:

```
$ neutron net-list
+--------------------------------------+-------------+---------------------------------
----------------------+
| id                                   | name        | subnets
|
+--------------------------------------+-------------+---------------------------------
----------------------+
| 6c917a40-3c07-4c4d-a563-999059dfcdc2 | public      | 302de9a1-165d-4ebe-9e64-
7cf9ede96594 2001:db8::/64        |
|                                      |             | d921aa37-b2a1-4ac0-8120-
f8bd71b64c2e 172.24.4.0/24         |
| fa919239-94cc-49f7-b861-67cd5853fadf | ext-net-iscsi | fabb456f-ef5d-4681-a84a-
4698ff3a1e56 10.0.0.0/8            |
| 2d702e87-bc3c-4089-b557-c7e6c27f89e7 | private     | 328554df-d5c3-48f1-8a4c-
2d89d3d1cddb 10.0.0.0/24           |
|                                      |             | ce3a02e9-7fc9-400d-8096-
7aef62276522 fde6:3ad8:da9c::/64 |
+--------------------------------------+-------------+---------------------------------
----------------------+

$ sudo ovs-vsctl show
```

Deploy an Instance on a Reduxio volume

The Nova services enables the user to create a new instance by assigning it a Cinder volume. The volume must be larger than the flavor dictates.

To view the existing instance flavors:

```
stack@devstack:~$ nova flavor-list
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1  | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
```

```
| 42 | m1.nano   | 64    | 0   | 0 |          | 1 | 1.0 | True |
| 5  | m1.xlarge | 16384 | 160 | 0 |          | 8 | 1.0 | True |
| 84 | m1.micro  | 128   | 0   | 0 |          | 1 | 1.0 | True |
+----+----------+----------+------+----------+------+-------+------------+----------+
```

To view the available Glance images:

```
stack@devstack:~$ nova image-list
+--------------------------------------+------------------------------+--------+--------+
| ID                                   | Name                         | Status | Server |
+--------------------------------------+------------------------------+--------+--------+
| 8d3f497c-4cb0-448d-9edc-5aca04cf5270 | cirros-0.3.4-x86_64-uec      | ACTIVE |        |
| ed729b92-af7c-4dd1-8712-8f0c58c3535f | cirros-0.3.4-x86_64-uec-kernel | ACTIVE |      |
| 0afd82a5-55f3-4b55-a08a-a30564415d73 | cirros-0.3.4-x86_64-uec-ramdisk| ACTIVE |      |
+--------------------------------------+------------------------------+--------+--------+
```

To create a new instance named instance1 configured with flavor m1.large, based on the cirros image, and deployed on the vol1 volume:

```
stack@devstack:~$ nova boot --nic net-id=7878fcf7-a961-4050-8a8c-9484f0a36615 --flavor 1 --
image 2637004d-90be-4a19-bb89-9d8fe46a24fd --block-device source=volume,id=f747e1c8-4b60-
4679-92c5-edca3eec044d,dest=volume,shutdown=preserve instance1
```

Manage an existing Reduxio volume

Volumes created directly by Reduxio Storage Manager and ReduxioCLI can be imported into OpenStack. An imported volume will be assigned a volume id which is stored in the Reduxio volume's description field.

BackDating™

Overview

Backdating enables the administrator to revert or clone volumes to any point in time back in the history. Unlike legacy storage designs, Reduxio operates as a "data recorder" and keeps track of both **location** – *where is the data stored (LUN and offset)*, and **time** – *when was the data written (timestamp)*.

BackDating is the evolution of snapshots. Snapshots only provide a set of pre-defined point in time (PITs) copies per volume, at the cost of complex planning, scheduling and application-aware backup software. Backdating replaces all of that with a simple to use timeline. It is possible to select any point on this timeline and clone a volume on its basis.

Backdating™ improves on the standard Cinder snapshot mechanism, and lets the OpenStack user to recover instances from any point in time in the past.

Concepts

History Timeline

A history timeline can be thought of as a continuous timeline of seconds from the creation of a volume to the current time, as seen in Figure 1 - History Timeline.

In such a time   volume, such that it is recoverable to every second in the past – from the current time back to the volume's creation time. However, typically as data ages, its history becomes less relevant. The volume' timeline was therefore designed with multiple granularity levels of history defined by a policy. The most recent part of history still maintains the continuous timeline of seconds; however, moving towards the past, there are portions when only minutes, hours, days and weeks are kept. Each volume is assigned to a single history policy that dictates its history.

Bookmarks

Bookmarks are user-initiated labels for specific time points in a volume that have significance to the user. For example, a bookmark can be created for a certain application volume before a major change such as a service pack update, or a large data processing batch job. These bookmarks would enable the administrator to identify these times more easily, and recover from failures by cloning or reverting volumes to the bookmarked time.

History is deleted according to the history policies. However, bookmarks can be used to guarantee that certain timestamps will not be deleted. For example, use a bookmark to maintain a baseline version of a database. Database clones can be created using the bookmark, without the risk of the baseline version being deleted from the system. All the data blocks required for its timestamp remain until the bookmark itself is deleted.

There are two types of bookmarks:

Automatic	The bookmark is automatically deleted based on the history policy assigned to the volume.

	For example, a volume is configured for storing a database. The default policy is configured. An automatic bookmark is then created for a database backup.
	This bookmark will be automatically deleted after the maximum retention level is reached for this volume. For example, when using the Default-Apps policy, automatic bookmarks will be deleted after one year.
Manual	The bookmark is kept by the system until it is manually deleted. This is especially useful for keeping important time points from being deleted accidentally. However, to avoid a full system state, manual bookmarks should eventually be deleted.

Cinder snapshots are stored in Reduxio as manual bookmarks.

History Policies
The system maintains a certain amount of history for each volume, referred to as the retention period. The amount is defined by setting a history policy, which dictates the deletion of past blocks. Only blocks that have no future reference are deleted.

Clones
Clones provide an advanced functionality that is highly beneficial to various recovery, test and development use cases.

Clones are writeable and independent	A cloned volume is 100% equivalent to the volume it based on, at the chosen timestamp. However, the clone and the source volume are entirely independent from each other. The source volume can be deleted without affecting its clone/s. Clones are standard read/write volumes. This enables the administrator to clone a certain application, and keep running from the clone.
One-to-many	A single volume can be cloned to many volumes. Note that the clones are similar to other volumes and are accounted together towards the maximum number of volumes.
	For example, if the maximum number of volumes is 1,000, there can be a single volume and 999 clones.
Multi-level Cascading	Cloned volumes can be recursively cloned for versioning purposes. For example, vol1 can be cloned to vol2, which can then be cloned to vol3.
Automatic Consistency	Data is consistent across volumes in every point in the timeline. This provides an inherent support for consistency groups. As long as a set of related volumes maintain the same timestamps (by configuring the same or a similar history policy), reverting or cloning volumes to the same timestamps across the volumes will provide a consistent view.
	For example, db1 and log1 volumes are used to store a database's data and log files respectively. When cloned or reverted to the same timestamp, the resulting volumes will be consistent to that timestamp.

Recovering Data

Volumes can be reverted or cloned from any point-in-time in the history timeline of a volume:

Clone	Create a new volume that is based on another volume from a selected timestamp.
	For example, consider the following timeline:
	Jan 19, 2015 9:34:00am - volume db1 created. Jan 19, 2015 10:05:00am - volume db1_clone created, based on db1, at timestamp 9:55:04am.
	The cloned volume db1_clone will contain the contents of db1 as it was in 9:55:04am. Concurrently with volume db1_clone, volume db1 continues to function as before – including all updates it received since the moment of the cloning.
Revert	Revert a volume to a selected timestamp. The volume contents will be changed in-place; any changes beyond the timestamp will be lost.
	For example, consider the following timeline:
	Jan 19, 2015 9:34:00am - volume db1 created. Jan 19, 2015 10:05:00am - volume db1 is reverted to timestamp 9:55:04am. All changes beyond 9:55:04am are lost.
	Note: A volume must be dismounted from the host before a revert operation, as the host may contain cache information or file system metadata that were not committed to storage yet, which may conflict with the reverted volume. Once the revert is completed, the volume can be safely remounted.

Recovering Instances

Overview

The Reduxio Cinder driver is integrated with the BackDating technology. Cinder snapshots can be captured using Reduxio bookmarks, and volumes can also be cloned or reverted to any second in the past as well.

Recovery using Cinder Snapshots

To create a Cinder snapshot:

```
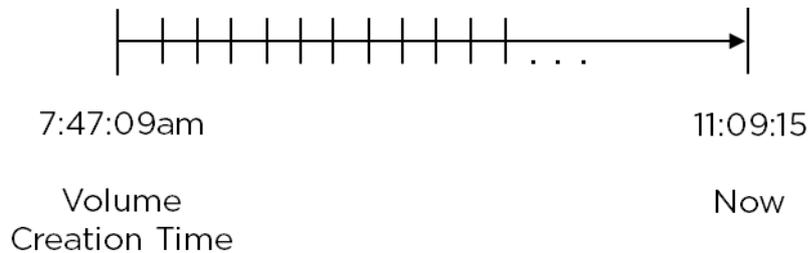$ cinder snapshot-create --name backup1 vol1
+-------------+--------------------------------------+
|   Property  |                Value                 |
+-------------+--------------------------------------+
|  created_at |      2016-04-10T08:07:04.865626      |
| description |                 None                 |
|      id     | cfa58882-e4db-400c-94f1-6a26ed4834fb |
|   metadata  |                  {}                  |
|     name    |               backup1                |
|     size    |                  20                  |
|    status   |               creating               |
|  updated_at |                 None                 |
|  volume_id  | 85f91996-f54c-4bef-8484-aed494ac510f |
+-------------+--------------------------------------+
```

A new manual bookmark is created on the Reduxio system named snapshot-<id>. To list the Reduxio bookmarks, run the following in ReduxioCLI:

```
rdxadmin@mango:/ > volumes list-bookmarks volume-85f91996-f54c-4bef-8484
```

```
name                             start_time               type
-------------------------------------------------------------------
snapshot-cfa58882-e4db-400c-94   04/10/2016-01:07:06 -0700  manual
```

To list the Cinder snapshots:

```
$ cinder snapshot-list
+------------------------------+------------------------------+-----------+---------+------+
|              ID              |          Volume ID           |  Status   |  Name   | Size |
+------------------------------+------------------------------+-----------+---------+------+
| cfa58882-e4db-400c-94f1-6a26…| 85f91996-f54c-4bef-8484-…    | available | backup1 |  20  |
+------------------------------+------------------------------+-----------+---------+------+
```

To create a new volume from a snapshot:

```
$ cinder create --snapshot-id cfa58882-e4db-400c-94f1-6a26ed4834fb
```

Recovery using BackDating

It is possible to create volume clones from a past timestamp by using the cinder create metadata field to pass the requested timestamp to the Reduxio driver. This enables the user to clone instances and data volumes to any second in the past, based on the available history.

To create a new volume from a BackDating timestamp:

```
$ cinder create --source-volid e64fc6de-fac6-427d-89a1-4c467c696561 --metadata
backdate="04/03/2016-12:51:00"
```

Conclusion

Reduxio storage systems based on TimeOS, offer breakthroughs in efficiency, performance and unique data management capabilities far exceeding anything in the market today.

The Reduxio Cinder driver integrates Reduxio storage with the OpenStack environment, and introduces data recovery to any second using Reduxio BackDating to the OpenStack cloud.

Troubleshooting

This section provides information on troubleshooting problems with the Reduxio Cinder driver.

Collecting Support Logs

To troubleshoot problems with the Reduxio Cinder driver, collect the following log files and provide to Reduxio Support if needed:

1. **Cinder Logs:**

   `/opt/stack/logs/cinder-volume.log`
   `/opt/stack/logs/cinder-api.log`
   `/opt/stack/logs/cinder-scheduler.log`

2. **Nova Logs:**

   `/opt/stack/logs/n-sch.log`

Common Problems

ERROR: Unable to establish connection to
http://172.18.3.43:9696/v2.0/quotas/admin.json
ERROR: ERROR (ClientException): Unexpected API Error.

Explanation
Not all OpenStack services are running or operational.

Solution
Restart OpenStack services.

In Red Hat OpenStack:

```
$ openstack-service stop
$ openstack-service start
```

Policy doesn't allow volume_extension:volume_manage to be performed. (HTTP 403) (Request-ID: req-d1d7513c-9804-450e-a87a-da69f354f4ea)

Explanation
OS_USERNAME environment variable is not set to the correct username.

Solution
Verify correct environment variable settings in openrc file.

ERROR: Policy doesn't allow volume_extension:volume_manage to be performed. (HTTP 403) (Request-ID: req-d1d7513c-9804-450e-a87a-da69f354f4ea)

Explanation
OS_USERNAME environment variable is not set to the correct username.

Solution
1. Verify correct environment variable settings in openrc file.
2. Run:

```
$ source openrc admin admin
```

ERROR: Requested volume or snapshot exceeds allowed gigabytes quota. Requested 10240G, quota is 1000G and 360G has been consumed.

Explanation
The gigabytes quota has been exceeded.

Solution
Update the gigabytes quota in the relevant project. Refer to the Update Cinder Quotas section for more information.

ERROR: Service not found. (HTTP 404) when running cinder manage

Explanation
The cinder manage command fails with the following error:

```
$ cinder manage --name vol1 devstack vol1
ERROR: Service not found. (HTTP 404) (Request-ID: req-7befb450-5697-4cc6-b980-d2a32e6ed0e0)
```

The cinder manage/unmanage commands require the host@backend#pool format for the host argument.

Solution
Run the following command instead:

```
$ cinder manage --name vol1 devstack@REDUXIO_VOLUME_TYPE#pool1 vol1
```

Reduxio Documentation

- *Reduxio TimeOS™ Administration Guide*

OpenStack Documentation

- *[OpenStack website](#)*