CONFIGURING THE ISCSI TARGET USING ANSIBLE

The Ceph iSCSI gateway is the iSCSI target node and also a Ceph client node. The Ceph iSCSI gateway can be a standalone node or be colocated on a Ceph Object Store Disk (OSD) node. Completing the following steps will install, and configure the Ceph iSCSI gateway for basic operation.

Requirements:

- A running Ceph Luminous (12.2.x) cluster or newer
- RHEL/CentOS 7.5; Linux kernel v4.16 or newer; or the Ceph iSCSI client test kernel
- The `ceph-iscsi-config` package installed on all the iSCSI gateway nodes

Installing:

1. On the Ansible installer node, which could be either the administration node or a dedicated deployment node, perform the following steps:
    1. As `root`, install the `ceph-ansible` package:

        ```
        # yum install ceph-ansible
        ```

    2. Add an entry in `/etc/ansible/hosts` file for the gateway group:

        ```
        [ceph-iscsi-gw]
        ceph-igw-1
        ceph-igw-2
        ```

> **Note:** If co-locating the iSCSI gateway with an OSD node, then add the OSD node to the `[ceph-iscsi-gw]` section.

Configuring:

The `ceph-ansible` package places a file in the `/usr/share/ceph-ansible/group_vars/` directory called `ceph-iscsi-gw.sample`. Create a copy of this sample file named `ceph-iscsi-gw.yml`. Review the following Ansible variables and descriptions, and update accordingly.

Variable	Meaning/Purpose
`seed_monitor`	Each gateway needs access to the ceph cluster for rados and rbd calls. This means the iSCSI gateway must have an appropriate `/etc/ceph/` directory defined. The `seed_monitor` host is used to populate the iSCSI gateway's `/etc/ceph/` directory.
`cluster_name`	Define a custom storage cluster name.
`gateway_keyring`	Define a custom keyring name.
`deploy_settings`	If set to `true`, then deploy the settings when the playbook is ran.
`perform_system_checks`	This is a boolean value that checks for multipath and lvm configuration settings on each gateway. It must be set to true for at least the first run to ensure multipathd and lvm are configured properly.
`gateway_iqn`	This is the iSCSI IQN that all the gateways will expose to clients. This means each client will see the gateway group as a single subsystem.
`gateway_ip_list`	The ip list defines the IP addresses that will be used on the front end network for iSCSI traffic. This IP will be bound to the active target portal group on each node, and is the access point for iSCSI traffic. Each IP should correspond to an IP available on the hosts defined in the `ceph-iscsi-gw` host group in `/etc/ansible/hosts`.
`rbd_devices`	This section defines the RBD images that will be controlled and managed within the iSCSI gateway configuration. Parameters like `pool` and `image` are self explanatory. Here are the other parameters: `size` = This defines the size of the RBD. You may increase the size later, by simply

	changing this value, but shrinking the size of an RBD is not supported and is ignored. `host` = This is the iSCSI gateway host name that will be responsible for the rbd allocation/resize. Every defined `rbd_device` entry must have a host assigned. `state` = This is typical Ansible syntax for whether the resource should be defined or removed. A request with a state of absent will first be checked to ensure the rbd is not mapped to any client. If the RBD is unallocated, it will be removed from the iSCSI gateway and deleted from the configuration.
`client_connections`	This section defines the iSCSI client connection details together with the LUN (RBD image) masking. Currently only CHAP is supported as an authentication mechanism. Each connection defines an `image_list` which is a comma separated list of the form `pool.rbd_image[,pool.rbd_image]`. RBD images can be added and removed from this list, to change the client masking. Note that there are no checks done to limit RBD sharing across client connections.

Note: When using the `gateway_iqn` variable, and for Red Hat Enterprise Linux clients, installing the `iscsi-initiator-utils` package is required for retrieving the gateway's IQN name. The iSCSI initiator name is located in the `/etc/iscsi/initiatorname.iscsi` file.

Deploying:

On the Ansible installer node, perform the following steps.

1. As `root`, execute the Ansible playbook:

```
# cd /usr/share/ceph-ansible
# ansible-playbook ceph-iscsi-gw.yml
```

Note: The Ansible playbook will handle RPM dependencies, RBD creation and Linux IO configuration.

2. Verify the configuration from an iSCSI gateway node:

```
# gwcli ls
```

Note: For more information on using the `gwcli` command to install and configure a Ceph iSCSI gateaway, see the Configuring the iSCSI Target using the Command Line Interface section.

Important: Attempting to use the `targetcli` tool to change the configuration will result in the following issues, such as ALUA misconfiguration and path failover problems. There is the potential to corrupt data, to have mismatched configuration across iSCSI gateways, and to have mismatched WWN information, which will lead to client multipath problems.

Service Management:

The `ceph-iscsi-config` package installs the configuration management logic and a Systemd service called `rbd-target-gw`. When the Systemd service is enabled, the `rbd-target-gw` will start at boot time and will restore the Linux IO state. The Ansible playbook disables the target service during the deployment. Below are the outcomes of when interacting with the `rbd-target-gw` Systemd service.

```
# systemctl <start|stop|restart|reload> rbd-target-gw
```

- `reload`

  A reload request will force `rbd-target-gw` to reread the configuration and apply it to the current running environment. This is normally not required, since changes are deployed in parallel from Ansible to all iSCSI gateway nodes

- `stop`

  A stop request will close the gateway's portal interfaces, dropping connections to clients and wipe the current LIO

configuration from the kernel. This returns the iSCSI gateway to a clean state. When clients are disconnected, active I/O is rescheduled to the other iSCSI gateways by the client side multipathing layer.

Administration:

Within the `/usr/share/ceph-ansible/group_vars/ceph-iscsi-gw` file there are a number of operational workflows that the Ansible playbook supports.

Warning: Before removing RBD images from the iSCSI gateway configuration, follow the standard procedures for removing a storage device from the operating system.

I want to…	Update the `ceph-iscsi-gw` file by…
Add more RBD images	Adding another entry to the `rbd_devices` section with the new image.
Resize an existing RBD image	Updating the size parameter within the `rbd_devices` section. Client side actions are required to pick up the new size of the disk.
Add a client	Adding an entry to the `client_connections` section.
Add another RBD to a client	Adding the relevant RBD `pool.image` name to the `image_list` variable for the client.
Remove an RBD from a client	Removing the RBD `pool.image` name from the clients `image_list` variable.
Remove an RBD from the system	Changing the RBD entry state variable to `absent`. The RBD image must be unallocated from the operating system first for this to succeed.
Change the clients CHAP credentials	Updating the relevant CHAP details in `client_connections`. This will need to be coordinated with the clients. For example, the client issues an iSCSI logout, the credentials are changed by the Ansible playbook, the credentials are changed at the client, then the client performs an iSCSI login.
Remove a client	Updating the relevant `client_connections` item with a state of `absent`. Once the Ansible playbook is ran, the client will be purged from the system, but the disks will remain defined to Linux IO for potential reuse.

Once a change has been made, rerun the Ansible playbook to apply the change across the iSCSI gateway nodes.

```
# ansible-playbook ceph-iscsi-gw.yml
```

Removing the Configuration:

The `ceph-ansible` package provides an Ansible playbook to remove the iSCSI gateway configuration and related RBD images. The Ansible playbook is `/usr/share/ceph-ansible/purge_gateways.yml`. When this Ansible playbook is ran a prompted for the type of purge to perform:

lio :

In this mode the LIO configuration is purged on all iSCSI gateways that are defined. Disks that were created are left untouched within the Ceph storage cluster.

all :

When `all` is chosen, the LIO configuration is removed together with **all** RBD images that were defined within the iSCSI gateway environment, other unrelated RBD images will not be removed. Ensure the correct mode is chosen, this operation will delete data.

Warning: A purge operation is destructive action against your iSCSI gateway environment.

Warning: A purge operation will fail, if RBD images have snapshots or clones and are exported through the Ceph iSCSI gateway.

```
[root@rh7-iscsi-client ceph-ansible]# ansible-playbook purge_gateways.yml
Which configuration elements should be purged? (all, lio or abort) [abort]: all
```

```
PLAY [Confirm removal of the iSCSI gateway configuration] **********


GATHERING FACTS ***************************************************************
ok: [localhost]


TASK: [Exit playbook if user aborted the purge] *******************************
skipping: [localhost]


TASK: [set_fact ] *************************************************************
ok: [localhost]


PLAY [Removing the gateway configuration] *************************************


GATHERING FACTS ***************************************************************
ok: [ceph-igw-1]
ok: [ceph-igw-2]


TASK: [igw_purge | purging the gateway configuration] *************************
changed: [ceph-igw-1]
changed: [ceph-igw-2]


TASK: [igw_purge | deleting configured rbd devices] ***************************
changed: [ceph-igw-1]
changed: [ceph-igw-2]


PLAY RECAP ********************************************************************
ceph-igw-1                 : ok=3    changed=2    unreachable=0    failed=0
ceph-igw-2                 : ok=3    changed=2    unreachable=0    failed=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0
```