iSCSI Target & Initiator Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 49 Agenda:

iSCSI Target & Initiator Setup Using Ansible:

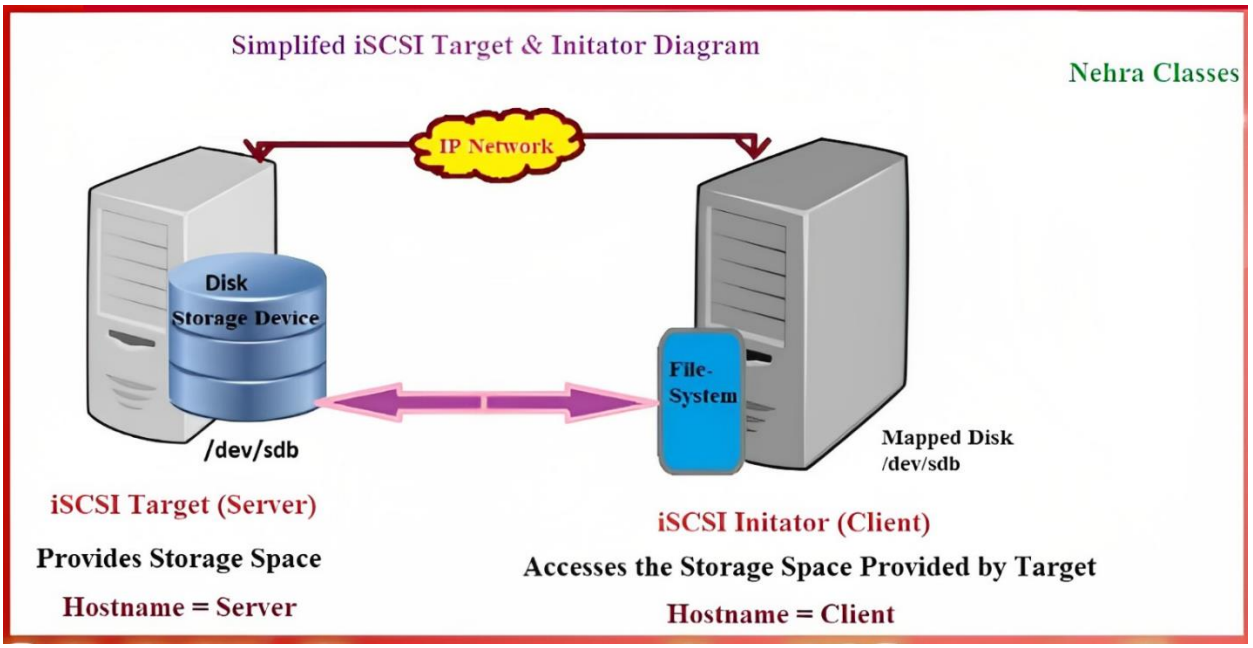
iSCSI (Internet Small Computer System Interface) is a protocol that allows SCSI commands and data to be transmitted over an IP network, enabling block-level storage communication between computers. In the context of iSCSI, "initiator" and "target" are two fundamental roles that devices can play. Let's break down each role:

iSCSI Initiator:

An iSCSI initiator is a device (usually a server or a computing system) that initiates the connection to the iSCSI target. It sends SCSI commands over the IP network to the target to access storage resources. In other words, the initiator is responsible for requesting data from the target, treating the remote storage as if it were a local drive. The initiator's operating system typically includes iSCSI initiator software that manages the connection and handles the SCSI commands and responses.

iSCSI Target:

An iSCSI target is a device (often a storage system or device) that provides access to its storage resources over the IP network to iSCSI initiators. It exposes its storage volumes as iSCSI LUNs (Logical Unit Numbers), which the initiators can access. The target is responsible for processing the incoming iSCSI requests from initiators, executing the requested SCSI commands on the storage, and sending back the appropriate responses.



Let's create a shell script to configure the iscsi-target server, which we will push on the ansible managed node where we need to configure the iSCSI target server.

$ vim iscsi-target.sh

```
#!/bin/bash

# Install required packages
sudo dnf install targetcli iscsi-initiator-utils -y

# Start & enable the iscsid service.
sudo systemctl enable --now iscsid

# Enable and start the target service
sudo systemctl enable target --now

# Configure the iSCSI target
sudo targetcli backstores/fileio create disk01 /opt/iscsi_disk.img 5G

sudo targetcli iscsi/ create iqn.1994-05.com.redhat:26b1519e5e1c
sudo targetcli iscsi/iqn.1994-05.com.redhat:26b1519e5e1c/tpg1/luns/ create /backstores/fileio/disk01
sudo targetcli iscsi/iqn.1994-05.com.redhat:26b1519e5e1c/tpg1/acls/ create #Client-IQN
```

iSCSI Target & Initiator Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
# Configure firewall rules
sudo firewall-cmd --add-service=iscsi-target --permanent
sudo firewall-cmd --add-port=3260/tcp --permanent
sudo firewall-cmd --reload

echo "iSCSI target server setup complete!"
```

Apply the execute permission on the script file.
$ chmod +x iscsi-target.sh

Let's create a shell script to configure the iscsi-**initiator** client, which we will push on the ansible managed node where we need to configure the iSCSI target server.
$ vim iscsi-initiator.sh

```
#!/bin/bash

# iSCSI target information
ISCSI_TARGET_IP="192.168.229.129"
ISCSI_TARGET_PORT="3260"
ISCSI_TARGET_IQN="iqn.1994-05.com.redhat:26b1519e5e1c"

# Install the iscsi-initiator packages.
sudo dnf install iscsi-initiator-utils -y

# Start & enable the iscsid service.
sudo systemctl enable --now iscsid

# iSCSI initiator name
ISCSI_INITIATOR_NAME=$(cat /etc/iscsi/initiatorname.iscsi | grep -oE 'InitiatorName=[^"]+' | cut -d '=' -f2)

# Scan & Find the iSCSI Target IQN
iscsiadm -m discovery -t sendtargets -p "$ISCSI_TARGET_IP:$ISCSI_TARGET_PORT"

# Connect the iSCSI Target shared lun in the local machine.
iscsiadm -m node -T "$ISCSI_TARGET_IQN" "$ISCSI_TARGET_IP:$ISCSI_TARGET_PORT" -l

# Wait for the device to be ready
sleep 5

# Find the new LUN device
NEW_DEVICE=$(iscsiadm -m session -P3 | awk '/Attached scsi disk/ { print $4; exit }')

if [ -z "$NEW_DEVICE" ]; then
   echo "Failed to find the new LUN device."
   exit 1
fi

# Partition the device (if needed)
# Replace /dev/sdX with the actual device name (e.g., /dev/sdb)
# echo -e "n\np\n1\n\n\nw" | fdisk /dev/"$NEW_DEVICE"

# Create a filesystem (ext4 in this case)
mkfs.ext4 /dev/"$NEW_DEVICE"

# Create a mount point directory
MOUNT_POINT="/opt/iscsi_mount"
mkdir -p "$MOUNT_POINT"

# Mount the filesystem
mount /dev/"$NEW_DEVICE" "$MOUNT_POINT"
```

iSCSI Target & Initiator Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Apply the execute permission on the script file.
$ chmod +x iscsi-initiator.sh

Now, we can create ansible playbooks for configure the iscsi-target server and iscsi-initiator (client) on the ansible managed nodes. In our case we have two machines node1 and node3, where we will configure iscsi-target & iscsi-initiator respectively.

Let's create an ansible playbook to configure the iscsi-target server first on the ansible managed node1.
$ vim iscsi-target.yml

```
---
- name: iSCSI Target Server Configuration Playbook
  hosts: node1
  become: true
  tasks:
    - name: Run the iscsi-target script which does not exists on the remote node.
      script: /home/vikasnehra/iscsi-target.sh
      args:
        creates: /home/vikasnehra/iscsi-target.sh
...
```

Now, we will create another ansible playbook to configure the iscsi-initiator (client) on the ansible managed node3.
$ vim iscsi-initiator.yml

```
---
- name: iSCSI Client Configuration Playbook
  hosts: node3
  become: true
  tasks:
    - name: Run the iscsi-initiator.sh script which does not exists on the remote node.
      script: /home/vikasnehra/iscsi-initiator.sh
      args:
        creates: /home/vikasnehra/iscsi-initiator.sh
...
```

Now we have total 4 files here, out of these 2 files are shell scripts and 2 are ansible playbooks.
$ ls -lh

First of all, we will execute the iscsi-initiator.yml playbook (to get its IQN) on the managed node3 where we want to attach the lun that will be shared from the target (node1).
$ ansible-playbook iscsi-initiator.yml

The entire shell script will not execute because we haven't configured the iscsi-target yet but it will generate the IQN on the node3 machine, which we can find by running the below ad-hoc command.
$ ansible node3 -m command -a 'sudo cat /etc/iscsi/initiatorname.iscsi'

Now copy this IQN and paste it in the iscsi-target.sh file to authorize the client.
$ vim iscsi-target.sh

Verify the changes.
$ grep IQN iscsi-target.sh

Now, we will execute the iscsi-target.yml on the ansible node1 where we want to configure the iSCSI target so that we can create a lun from the file. This playbook will copy the iscsi-target.sh shell script on node1 and execute it.
$ ansible-playbook iscsi-target.yml

Verify the changes by running the below ansible ad-hoc command, which will show you the image file for the lun.
$ ansible node1 -m command -a 'sudo ls -lh /opt/'

Now, execute the lsblk command on the managed node3 to verify that there is no /dev/sda disk present yet.
$ ansible node3 -m command -a 'sudo lsblk'

iSCSI Target & Initiator Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Now, execute the iscsi-initiator.yml playbook to copy and execute the iscsi-initiator.sh script at node3 machine, which will scan the lun, connect it there and then create the filesystem before mounting it.

$ ansible-playbook iscsi-initiator.yml

Now, you can find additional disk at node3 machine as /dev/sda.

$ ansible node3 -m command -a 'sudo lsblk'

Verify the mount filesystems.

$ ansible node3 -m command -a 'sudo df -hT'

Confirm that the /dev/sda disk is the lun shared from the iscsi-target.

$ ansible node3 -m command -a 'sudo lsscsi'

You have successfully configured iscsi-target & initiator using Ansible & shell scripts.