Configuring Linux Host for iSCSI with FlashArray

This document covers the configuration and best practices to configure iSCSI in Linux. In this example, we used Red Hat Enterprise Linux 6, but this procedure has also been tested on Ubuntu. This procedure also works on SUSE/SLES systems. The following steps use commands with example IP addresses and IQNs. When running the commands, replace the IPs and the IQNs with those from your own environment.

Linux Host Configuration

1. Make sure that you are following Pure Storage Linux Recommended Settings before proceeding.
   
   Note: If multiple interfaces exist on the same subnet in RHEL, your iSCSI initiator may fail to connect to Pure Storage target. In this case, you need to set sysctl's net.ipv4.conf.all.arp_ignore to 1 to force each interface to only answer ARP requests for its own addresses. Please see RHEL KB for Issue Detail and Resolution Steps (requires Red Hat login).

2. Install the iscsi-initiator-utils package as root user:
   
   $ sudo su
   # yum install iscsi-initiator-utils

3. Start the iscsi service and enable it to start when the system boots:
   
   For RHEL6:
   
   # service iscsi start
   # chkconfig iscsi on

   For RHEL7:
   
   # systemctl start iscsid.socket
   # systemctl enable iscsi

   iscsid.socket would start iscsid.service if stopped. At this stage, the status of iscsi service service iscsi status might be seen as active or started. After the discovery command, the service starts.

4. Before setting up DM Multipath on your system, ensure that your system has been updated and includes the device-mapper-multipath package:
   
   # yum install device-mapper-multipath device-mapper-multipath-libs

5. Enable default multipath configuration file and start the multipath daemon:
The Multipath Policy defines how the host distributes IOs across the available paths to the storage. The Round Robin (RR) policy distributes IOs evenly across all Active/Optimized paths. A newer MPIO policy, queue-length, is similar to round-robin in that IOs are distributed across all available Active/Optimized paths (it also provides additional benefits). The queue-length path selector bias IOs towards paths that are servicing IO quicker (paths with shorter queues). In the event that one path becomes intermittently disruptive or is experiencing higher latency, queue-length will prevent the utilization of that path reducing the effect of the problem path.

The following are recommended entries to existing multipath.conf files (/etc/multipath.conf) for Linux OSes. Add the following to the existing section for controlling Pure devices.

Please note that fast_io_fail_tmo and dev_loss_tmo do not apply to iSCSI.

**RHEL 8.0**

RHEL 8.0 and several of latest distributions of Linux have newer versions of device-mapper-multipath that made some changes to default settings. Many of the previous pure settings have been covered by these new defaults that are in place. For Distributions using these newer versions of multipath, it is only necessary to add the following settings to multipath.conf. Please check the release-notes of your distribution to see if this applies to you.
fast_io_fail_tmo 10
path_grouping_policy "group_by_prio"
failback "immediate"
prio "aluad"
hardware_handler "1 alua"
max_sectors_kb 4096
}
}

RHEL 7.3

No manual changes required. The RHEL OS should configure this file automatically provided that the dm-multipath version is device-mapper-multipath-0.4.9-99.el7.x86_64. See the RHEL KB: https://access.redhat.com/solutions/2772111. The dm-multipath config shown below for PURE is default with the device-mapper version included in RHEL / Oracle Linux 7.3

```bash
device {
    vendor "PURE"
    product "FlashArray"
    path_grouping_policy "multibus"
    path_selector "queue-length 0"
    path_checker "tur"
    features "0"
    hardware_handler "0"
    prio "const"
    failback immediate
    fast_io_fail_tmo 10
    dev_loss_tmo 60
    user_friendly_names yes
}
```

Included in RHEL 7.3+ is device-mapper-multipath-0.4.9-99. Support added for PURE FlashArray - With this release, multipath has added built-in configuration support for the PURE FlashArray (BZ#1300415).

Supporting Info:
- **RHEL KB** - Standard dm-multipath configuration added for Pure Storage
- **Bug 1300415** - Add PURE to multipath-tools on RHEL
- **RHEL 7.3 Release Notes** - Support added for PURE FlashArray

RHEL 6.2 and above, SLES 12, and supporting kernels

```bash
defaults {
    polling_interval 10
    find_multipaths yes
}
devices {
```
device {
  vendor                "PURE"
  path_selector         "round-robin 0"
  path_grouping_policy  multibus
  rr_min_io             1
  path_checker          tur
  fast_io_fail_tmo      10
  dev_loss_tmo          60
  no_path_retry         0
  hardware_handler      "1 alua"
  prio                  alua
  failback              immediate
}

RHEL 5.x - 6.1 and supporting kernels

defaults {
  polling_interval 10
}
devices {
  device {
    vendor                "PURE"
    path_selector         "round-robin 0"
    path_grouping_policy  multibus
    path_checker          tur
    fast_io_fail_tmo      10
    dev_loss_tmo          60
    no_path_retry         0
  }
}

RHEL 5.6 and below, and supporting kernels

defaults {
  polling_interval 10
}
devices {
  device {
    vendor                "PURE"
    path_selector         "round-robin 0"
    path_grouping_policy  multibus
    rr_min_io             1
    path_checker          tur
    no_path_retry         0
  }
}
Oracle VM Server

```bash
device {
    vendor                "PURE"
    product               "FlashArray"
    path_selector         "queue-length 0"
    path_grouping_policy  group_by_prio
    path_checker          tur
    fast_io_fail_tmo      10
    dev_loss_tmo          60
    no_path_retry         0
    hardware_handler      "1 alua"
    prio                  alua
    failback              immediate
    user_friendly_names   no
}
```

More information on multipath settings can be found here: RHEL Documentation.

See RHEL documentation for /etc/multipath.conf attribute descriptions.

7. Restart multipath service for multipath.conf changes to take effect.

```
# service multipathd restart
```

Prepare the FlashArray with the Host, Volume, and Host IQN

1. On the Linux host, collect the IQN:
   ```bash
   # cat /etc/iscsi/initiatorname.iscsi
   ```

2. On FlashArray, create a host:
   ```bash
   purehost create <Linux hostname>
   where
   <Linux hostname> is the desired hostname.
   ```

3. Configure FlashArray host with IQN:
   ```bash
   purehost setattr --addiqnlist <IQN number> <Linux hostname>
   where
   <IQN number> is the initiator IQN number gathered in step 1.
   <Linux hostname> is the hostname created in step 2.
   ```
4. On the FlashArray, create a volume:

```
$ purevol create <volume name> --size <size>
```

where

- `<volume name>` is the desired volume name.
- `<size>` is the desired volume size (GB or TB suffix).

5. Connect the host to volume:

```
$ purevol connect <volume name> --host <host name>
```

where

- `<volume name>` is the name of the volume.
- `<host name>` is the name of the host.

6. On the FlashArray, collect iSCSI interface IPs:

```
$ pureport list
```

7. On Linux Host, discover the target iSCSI portals:

```
# iscsiadm -m discovery -t st -p <FlashArray iSCSI IP>:3260
```

where

- `<FlashArray iSCSI IP>` is the iSCSI interface IP address from either collected in step 6.

8. From your Linux Host, log in to the FlashArray iSCSI target portals on both controllers:

```
# iscsiadm -m node -p <FlashArray iSCSI IP CT0> --login
# iscsiadm -m node -p <FlashArray iSCSI IP CT1> --login
```

where

- `<FlashArray iSCSI IP CT0>` is the iSCSI interface IP address of controller 0 collected from step 6.
- `<FlashArray iSCSI IP CT1>` is the iSCSI interface IP address of controller 1 collected from step 6.

9. Add automatic iSCSI login on boot:

```
# iscsiadm -m node -L automatic
```

10. Confirm the FlashArray volume has multiple paths with `multipath -ll`. A multipathed volume should be represented by a device-mapped ID, as shown in green in the example below:

```
# multipath -ll
3624a93702b60622e2b014a2200011011 dm-1 PURE,FlashArray
size=2.0T features='0' hwhandler='1 alua' wp=rw
```
Mount Volume and Provision Filesystem

1. Create a mount point on the Linux host.

   # mkdir /mnt/store0

2. Provision filesystem on the PURE dm device using the device-mapped ID.

   # mkfs.ext4 /dev/mapper/<device-mapped ID>

   where

   <device-mapped ID> is the device-mapped ID from step 10.

To enable automatic unmap for our thin-provisioning array, use the `-o discard` option when provisioning the filesystem.

   # mkfs.ext4 -o discard /dev/sdb5

   This will cause the RHEL 6.x to issue the UNMAP command, which in turn causes space to be released back to the array for any deletions in that ext4 file system. This only works on Physical RDM datastores, discard will not work on a disk mapped virtually via ESX.

3. Mount PURE dm device to mount point:

   # mount/dev/mapper/<device-mapped ID> <mount point>
where
<device-mapped ID> is the device-mapped ID collected from step 10.

<mount point> is the mount point created in step 1.

or

```
# mount -a
```

or if you require to mount the partition as read-only:

```
# mount -o rw /mnt/store0
```

Verify the partition is mounted (this will also list the options for the mounted partition. i.e. "/dev/sdb5 on /data type ext4 (rw,_netdev)"):  

```
# mount
```

Confirm that the /mnt/iscsi folder is connected to the partition:

```
# df -h /mnt/store0
```

---

**Create Additional Interfaces (Optional)**

Open iSCSI initiator (i.e iscsiadm) utility provides a feature to create multiple interfaces:

```
# iscsiadm -m iface -I <iface name> -o new
```

You may then take the "-l" off the above command to display info about the iSCSI target:

```
# iscsiadm -m node -Tiqn.2010-06.com.purestorage:flasharray.38e69528198fee76 -p 10.124.3.159
# iscsiadm -m node -Tiqn.2010-06.com.purestorage:flasharray.38e69528198fee76 -p 10.124.3.158
```

Now update the newly created interface with a unique initiator name:

```
# iscsiadm -m iface -I <iface name> -o update -n iface.initiatorname -v <initiator name>
```

Rediscover paths from the new interface:

```
# iscsiadm -m discovery -t st -p 10.124.3.159:3260
```

Log in to the target IP with this newly created interface:

```
# iscsiadm -m node -p <FlashArray iSCSI IP CT0> --login
```

---

**Note:** To make iSCSI device mount persistent across reboots, you will need to add an entry in /etc/fstab following RHEL KB.
To verify the existing iscsi session:

```bash
iscsiadm -m session
```
You can use -P 0|1|2 for more verbosity on the sessions like initiator to target IP mapping, session timeout etc

#### Helpful Links

- [iscsiadm do not allow to discovery the same LUN with second NIC.](#)
- [Why does Red Hat Enterprise Linux 6 and above invalidate / discard packets when the route for outbound traffic differs from the route of incoming traffic?](#)