RHEL

ONTAP SAN Host Utilities

NetApp
January 30, 2026

Table of Contents

RHEL

Configure RHEL 10.x for FCP and iSCSI with ONTAP storage

The Linux Host Utilities software provides management and diagnostic tools for Linux hosts that are connected to ONTAP storage. When you install the Linux Host Utilities on a Red Hat Enterprise Linux (RHEL) 10.x host, you can use the Host Utilities to help you manage FCP and iSCSI protocol operations with ONTAP LUNs.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot LUN and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Install the Linux Host Utilities

NetApp strongly recommends installing the Linux Host Utilities to support ONTAP LUN management and assist technical support with gathering configuration data.

Install Linux Host Utilities 8.0.

> Installing the Linux Host Utilities doesn't change any host timeout settings on your Linux host.

Step 3: Confirm the multipath configuration for your host

You can use multipathing with RHEL 10.x to manage ONTAP LUNs.

To ensure that multipathing is configured correctly for your host, verify that the `/etc/multipath.conf` file is defined and that you have the NetApp recommended settings configured for your ONTAP LUNs.

Steps

1. Verify that the `/etc/multipath.conf` file exits. If the file doesn't exist, create an empty, zero-byte file:

   ```
   touch /etc/multipath.conf
   ```

2. The first time the `multipath.conf` file is created, you might need to enable and start the multipath

services to load the recommended settings:

```
systemctl enable multipathd
```

```
systemctl start multipathd
```

3. Each time you boot the host, the empty `/etc/multipath.conf` zero-byte file automatically loads the NetApp recommended host multipath parameters as the default settings. You shouldn’t need to make changes to the `/etc/multipath.conf` file for your host because the operating system is compiled with the multipath parameters that recognize and manage ONTAP LUNs correctly.

   The following table shows the Linux OS native compiled multipath parameter settings for ONTAP LUNs.

   Show parameter settings

Parameter	Setting
detect_prio	yes
dev_loss_tmo	"infinity"
failback	immediate
fast_io_fail_tmo	5
features	"2 pg_init_retries 50"
flush_on_last_del	"yes"
hardware_handler	"0"
no_path_retry	queue
path_checker	"tur"
path_grouping_policy	"group_by_prio"
path_selector	"service-time 0"
polling_interval	5
prio	"ontap"
product	LUN
retain_attached_hw_handler	yes
rr_weight	"uniform"
user_friendly_names	no
vendor	NETAPP

4. Verify the parameter settings and path status for your ONTAP LUNs:

```
multipath -ll
```

The default multipath parameters support ASA, AFF, and FAS configurations. In these configurations, a single ONTAP LUN shouldn't require more than four paths. Having more than four paths can cause problems during a storage failure.

The following example outputs show the correct parameter settings and path status for ONTAP LUNs in an ASA, AFF, or FAS configuration.

ASA configuration

An ASA configuration optimizes all paths to a given LUN, keeping them active. This improves performance by serving I/O operations through all paths at the same time.

Show example

```
# multipath -ll
3600a098038314e535a24584e4b496252 dm-32 NETAPP,LUN C-Mode
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 11:0:0:41 sdan 66:112  active ready running
  |- 11:0:1:41 sdcb 68:240  active ready running
  |- 14:0:2:41 sdfd 129:240 active ready running
  `- 14:0:0:41 sddp 71:112  active ready running
```

AFF or FAS configuration

An AFF or FAS configuration should have two groups of paths with higher and lower priorities. Higher priority Active/Optimized paths are served by the controller where the aggregate is located. Lower priority paths are active but non-optimized because they are served by a different controller. Non-optimized paths are only used when optimized paths aren't available.

The following example displays the output for an ONTAP LUN with two Active/Optimized paths and two Active/Non-Optimized paths:

Show example

```
# multipath -ll
3600a0980383149764b5d567257516273 dm-0 NETAPP,LUN C-Mode
size=150G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 16:0:3:0  sdcg 69:64  active ready running
| `- 10:0:0:0  sdb  8:16   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 10:0:1:0  sdc  8:32   active ready running
  `- 16:0:2:0  sdcf 69:48  active ready running
```

Step 4: Confirm the iSCSI configuration for your host

Ensure that iSCSI is configured correctly for your host.

About this task

You perform the following steps on the iSCSI host.

Steps

1. Verify that the iSCSI initiator package (iscsi-initiator-utils) is installed:

```
rpm -qa | grep iscsi-initiator-utils
```

   You should see an output similar to the following example:

```
iscsi-initiator-utils-6.2.1.11-0.git4b3e853.el9.x86_64
```

2. Verify the iSCSI initiator node name, which is located in the `/etc/iscsi/initiatorname.iscsi` file:

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. Configure the iSCSI session timeout parameter located in the `/etc/iscsi/iscsid.conf` file:

```
node.session.timeo.replacement_timeout = 5
```

   The iSCSI `replacement_timeout` parameter controls how long the iSCSI layer should wait for a timed-out path or session to reestablish itself before failing any commands on it. You should set the value of `replacement_timeout` to 5 in the iSCSI configuration file.

4. Enable the iSCSI service:

```
$systemctl enable iscsid
```

5. Start the iSCSI service:

```
$systemctl start iscsid
```

6. Verify that the iSCSI service is running:

```
$systemctl status iscsid
```

Show example

```
● iscsid.service - Open-iSCSI
     Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
enabled; preset: disabled)
     Active: active (running) since Tue 2025-12-02 11:36:21 EST; 2
weeks 1 day ago
TriggeredBy: ● iscsid.socket
       Docs: man:iscsid(8)
             man:iscsiuio(8)
             man:iscsiadm(8)
   Main PID: 2263 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1 (limit: 816061)
     Memory: 18.5M
        CPU: 14.480s
     CGroup: /system.slice/iscsid.service
             └─2263 /usr/sbin/iscsid -f -d2
```

7. Discover the iSCSI targets:

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

show example

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.30.87
192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
```

8. Log in to the targets:

```
$iscsiadm --mode node -l all
```

9. Set iSCSI to log in automatically when the host boots:

```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n
node.startup -v automatic
```

You should see an output similar to the following example:

```
iscsiadm --mode node -T iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 -p
192.168.30.87:3260 -o update -n node.startup -v automatic
```

10. Verify the iSCSI sessions:

```
$iscsiadm --mode session
```

Show example

```
iscsiadm --mode session
tcp: [1] 192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [2] 192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [3] 192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [4] 192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
```

Step 5: Optionally, exclude a device from multipathing

If required, you can exclude a device from multipathing by adding the WWID for the unwanted device to the "blacklist" stanza for the `multipath.conf` file.

Steps

1. Determine the WWID:

```
/lib/udev/scsi_id -gud /dev/sda
```

"sda" is the local SCSI disk that you want to add to the blacklist.

An example WWID is `360030057024d0730239134810c0cb833`.

2. Add the WWID to the "blacklist" stanza:

```
blacklist {
         wwid   360030057024d0730239134810c0cb833
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
        devnode "^cciss.*"
}
```

Step 6: Customize multipath parameters for ONTAP LUNs

If your host is connected to LUNs from other vendors and any of the multipath parameter settings are overridden, you need to correct them by adding stanzas later in the `multipath.conf` file that apply specifically to ONTAP LUNs. If you don't do this, the ONTAP LUNs might not work as expected.

Check your `/etc/multipath.conf` file, especially in the defaults section, for settings that might be overriding the default settings for multipath parameters.

> You shouldn't override the recommended parameter settings for ONTAP LUNs. These settings are required for optimal performance of your host configuration. Contact NetApp support, your OS vendor, or both for more information.

The following example shows how to correct an overridden default. In this example, the `multipath.conf` file defines values for `path_checker` and `no_path_retry` that aren't compatible with ONTAP LUNs, and you can't remove these parameters because ONTAP storage arrays are still attached to the host. Instead, you correct the values for `path_checker` and `no_path_retry` by adding a device stanza to the `multipath.conf` file that applies specifically to the ONTAP LUNs.

Show example

```
defaults {
   path_checker      readsector0
   no_path_retry     fail
}

devices {
   device {
      vendor          "NETAPP"
      product         "LUN"
      no_path_retry   queue
      path_checker    tur
   }
}
```

Step 7: Review the known issues

There are no known issues.

What's next?

- Learn about using the Linux Host Utilities tool.
- Learn about ASM mirroring

  Automatic Storage Management (ASM) mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate failure group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM doesn't mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites. See Oracle databases on ONTAP for further information.

- Learn about Red Hat Linux Virtualization (KVM)

  Red Hat Linux can serve as a KVM host. This enables you to run multiple virtual machines on a single physical server using the Linux Kernel-based Virtual Machine (KVM) technology. The KVM host doesn't require explicit host configuration settings for ONTAP LUNs.

Configure RHEL 9.x for FCP and iSCSI with ONTAP storage

The Linux Host Utilities software provides management and diagnostic tools for Linux hosts that are connected to ONTAP storage. When you install the Linux Host Utilities on a Red Hat Enterprise Linux (RHEL) 9.x host, you can use the Host Utilities to help you manage FCP and iSCSI protocol operations with ONTAP LUNs.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot LUN and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Install the Linux Host Utilities

NetApp strongly recommends installing the Linux Host Utilities to support ONTAP LUN management and assist technical support with gathering configuration data.

Install Linux Host Utilities 8.0.

Installing the Linux Host Utilities doesn’t change any host timeout settings on your Linux host.

Step 3: Confirm the multipath configuration for your host

You can use multipathing with RHEL 9.x to manage ONTAP LUNs.

To ensure that multipathing is configured correctly for your host, verify that the `/etc/multipath.conf` file is defined and that you have the NetApp recommended settings configured for your ONTAP LUNs.

Steps

1. Verify that the `/etc/multipath.conf` file exits. If the file doesn’t exist, create an empty, zero-byte file:

```
touch /etc/multipath.conf
```

2. The first time the `multipath.conf` file is created, you might need to enable and start the multipath services to load the recommended settings:

```
systemctl enable multipathd
```

```
systemctl start multipathd
```

3. Each time you boot the host, the empty `/etc/multipath.conf` zero-byte file automatically loads the NetApp recommended host multipath parameters as the default settings. You shouldn’t need to make changes to the `/etc/multipath.conf` file for your host because the operating system is compiled with the multipath parameters that recognize and manage ONTAP LUNs correctly.

   The following table shows the Linux OS native compiled multipath parameter settings for ONTAP LUNs.

Show parameter settings

Parameter	Setting
detect_prio	yes
dev_loss_tmo	"infinity"
failback	immediate
fast_io_fail_tmo	5
features	"2 pg_init_retries 50"
flush_on_last_del	"yes"
hardware_handler	"0"
no_path_retry	queue
path_checker	"tur"
path_grouping_policy	"group_by_prio"
path_selector	"service-time 0"
polling_interval	5
prio	"ontap"
product	LUN
retain_attached_hw_handler	yes
rr_weight	"uniform"
user_friendly_names	no
vendor	NETAPP

4. Verify the parameter settings and path status for your ONTAP LUNs:

```
multipath -ll
```

The default multipath parameters support ASA, AFF, and FAS configurations. In these configurations, a single ONTAP LUN shouldn’t require more than four paths. Having more than four paths can cause problems during a storage failure.

The following example outputs show the correct parameter settings and path status for ONTAP LUNs in an ASA, AFF, or FAS configuration.

ASA configuration

An ASA configuration optimizes all paths to a given LUN, keeping them active. This improves performance by serving I/O operations through all paths at the same time.

Show example

```
multipath -ll
3600a098038314c4a433f577471797958 dm-2 NETAPP,LUN C-Mode
size=180G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 14:0:0:0  sdc  8:32   active ready running
  |- 17:0:0:0  sdas 66:192 active ready running
  |- 14:0:3:0  sdar 66:176 active ready running
  `- 17:0:3:0  sdch 69:80  active ready running
```

AFF or FAS configuration

An AFF or FAS configuration should have two groups of paths with higher and lower priorities. Higher priority Active/Optimized paths are served by the controller where the aggregate is located. Lower priority paths are active but non-optimized because they are served by a different controller. Non-optimized paths are only used when optimized paths aren't available.

The following example displays the output for an ONTAP LUN with two Active/Optimized paths and two Active/Non-Optimized paths:

Show example

```
multipath -ll
3600a0980383149764b5d567257516273 dm-0 NETAPP,LUN C-Mode
size=150G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 16:0:3:0  sdcg 69:64  active ready running
| `- 10:0:0:0  sdb  8:16   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 10:0:1:0  sdc  8:32   active ready running
  `- 16:0:2:0  sdcf 69:48  active ready running
```

Step 4: Confirm the iSCSI configuration for your host

Ensure that iSCSI is configured correctly for your host.

About this task

You perform the following steps on the iSCSI host.

Steps

1. Verify that the iSCSI initiator package (iscsi-initiator-utils) is installed:

```
rpm -qa | grep iscsi-initiator-utils
```

   You should see an output similar to the following example:

```
iscsi-initiator-utils-6.2.1.11-0.git4b3e853.el9.x86_64
```

2. Verify the iSCSI initiator node name, which is located in the `/etc/iscsi/initiatorname.iscsi` file:

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. Configure the iSCSI session timeout parameter located in the `/etc/iscsi/iscsid.conf` file:

```
node.session.timeo.replacement_timeout = 5
```

   The iSCSI `replacement_timeout` parameter controls how long the iSCSI layer should wait for a timed-out path or session to reestablish itself before failing any commands on it. You should set the value of `replacement_timeout` to 5 in the iSCSI configuration file.

4. Enable the iSCSI service:

```
$systemctl enable iscsid
```

5. Start the iSCSI service:

```
$systemctl start iscsid
```

6. Verify that the iSCSI service is running:

```
$systemctl status iscsid
```

Show example

```
● iscsid.service - Open-iSCSI
     Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
enabled; preset: disabled)
     Active: active (running) since Tue 2025-12-02 11:36:21 EST; 2
weeks 1 day ago
TriggeredBy: ● iscsid.socket
       Docs: man:iscsid(8)
             man:iscsiuio(8)
             man:iscsiadm(8)
   Main PID: 2263 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1 (limit: 816061)
     Memory: 18.5M
        CPU: 14.480s
     CGroup: /system.slice/iscsid.service
             └─2263 /usr/sbin/iscsid -f -d2
```

7. Discover the iSCSI targets:

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

show example

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.30.87
192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
```

8. Log in to the targets:

```
$iscsiadm --mode node -l all
```

9. Set iSCSI to log in automatically when the host boots:

```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n
node.startup -v automatic
```

You should see an output similar to the following example:

```
iscsiadm --mode node -T iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 -p
192.168.30.87:3260 -o update -n node.startup -v automatic
```

10. Verify the iSCSI sessions:

```
$iscsiadm --mode session
```

Show example

```
iscsiadm --mode session
tcp: [1] 192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [2] 192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [3] 192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [4] 192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
```

Step 5: Optionally, exclude a device from multipathing

If required, you can exclude a device from multipathing by adding the WWID for the unwanted device to the "blacklist" stanza for the `multipath.conf` file.

Steps

1. Determine the WWID:

```
/lib/udev/scsi_id -gud /dev/sda
```

"sda" is the local SCSI disk that you want to add to the blacklist.

An example WWID is `360030057024d0730239134810c0cb833`.

2. Add the WWID to the "blacklist" stanza:

```
blacklist {
          wwid   360030057024d0730239134810c0cb833
         devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
         devnode "^hd[a-z]"
         devnode "^cciss.*"
}
```

Step 6: Customize multipath parameters for ONTAP LUNs

If your host is connected to LUNs from other vendors and any of the multipath parameter settings are overridden, you need to correct them by adding stanzas later in the `multipath.conf` file that apply specifically to ONTAP LUNs. If you don't do this, the ONTAP LUNs might not work as expected.

Check your `/etc/multipath.conf` file, especially in the defaults section, for settings that might be overriding the default settings for multipath parameters.

> You shouldn't override the recommended parameter settings for ONTAP LUNs. These settings are required for optimal performance of your host configuration. Contact NetApp support, your OS vendor, or both for more information.

The following example shows how to correct an overridden default. In this example, the `multipath.conf` file defines values for `path_checker` and `no_path_retry` that aren't compatible with ONTAP LUNs, and you can't remove these parameters because ONTAP storage arrays are still attached to the host. Instead, you correct the values for `path_checker` and `no_path_retry` by adding a device stanza to the `multipath.conf` file that applies specifically to the ONTAP LUNs.

Show example

```
defaults {
   path_checker      readsector0
   no_path_retry     fail
}

devices {
   device {
      vendor          "NETAPP"
      product         "LUN"
      no_path_retry   queue
      path_checker    tur
   }
}
```

Step 7: Review the known issues

RHEL 9.x with ONTAP storage has the following known issues.

9.3

NetApp Bug ID	Title	Description	JIRA ID
1508554	NetApp Linux Host Utilities CLI requires additional library package dependencies to support Emulex host bus adapter (HBA) adapter discovery	In RHEL 9.x, the NetApp Linux SAN Host Utilities CLI `sanlun fcp show adapter -v` fails because the library package dependencies to support Emulex host bus adapter (HBA) discovery cannot be found.	Not applicable
1593771	A Red Hat Enterprise Linux 9.3 QLogic SAN host encounters loss of partial multipaths during storage mobility operations	During the ONTAP storage controller takeover operation, half of the multipaths are expected to go down or switch to a failover mode and then recover to full path count during the giveback workflow. However, with a Red Hat Enterprise Linux (RHEL) 9.3 QLogic host, only partial multipaths are recovered after a storage failover giveback operation.	RHEL 17811

9.2

NetApp Bug ID	Title	Description
1508554	NetApp Linux Host Utilities CLI requires additional library package dependencies to support Emulex HBA adapter discovery	In RHEL 9.2, the NetApp Linux SAN Host Utilities CLI `sanlun fcp show adapter -v` fails because the library package dependencies to support HBA discovery cannot be found.
1537359	A Red Hat Linux 9.2 SAN booted host with Emulex HBA encounters stalled tasks leading to kernel disruption	During a storage failover giveback operation, a Red Hat Linux 9.2 SAN booted host with an Emulex host bus adapeter (HBA) encounters stalled tasks leading to kernel disruption. The kernel disruption causes the operating system to reboot and if `kdump` is configured, it generates the `vmcore` file under the `/var/crash/` directory. The issue is being triaged with the `lpfc` driver but it cannot be reproduced consistently.

9.1

NetApp Bug ID	Title	Description
1508554	NetApp Linux Host Utilities CLI requires additional library package dependencies to support Emulex HBA adapter discovery	In RHEL 9.1, the NetApp Linux SAN Host Utilities CLI `sanlun fcp show adapter -v` fails because the library package dependencies to support HBA discovery cannot be found.

What's next?

- Learn about using the Linux Host Utilities tool.
- Learn about ASM mirroring

  Automatic Storage Management (ASM) mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate failure group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM doesn't mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites. See Oracle databases on ONTAP for further information.

- Learn about Red Hat Linux Virtualization (KVM)

  Red Hat Linux can serve as a KVM host. This enables you to run multiple virtual machines on a single physical server using the Linux Kernel-based Virtual Machine (KVM) technology. The KVM host doesn't require explicit host configuration settings for ONTAP LUNs.

Configure RHEL 8.x for FCP and iSCSI with ONTAP storage

The Linux Host Utilities software provides management and diagnostic tools for Linux hosts that are connected to ONTAP storage. When you install the Linux Host Utilities on a Red Hat Enterprise Linux (RHEL) 8.x host, you can use the Host Utilities to help you manage FCP and iSCSI protocol operations with ONTAP LUNs.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot LUN and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Install the Linux Host Utilities

NetApp strongly recommends installing the Linux Host Utilities to support ONTAP LUN management and assist technical support with gathering configuration data.

Install Linux Host Utilities 8.0.

ⓘ Installing the Linux Host Utilities doesn't change any host timeout settings on your Linux host.

Step 3: Confirm the multipath configuration for your host

You can use multipathing with RHEL 8.x to manage ONTAP LUNs.

To ensure that multipathing is configured correctly for your host, verify that the `/etc/multipath.conf` file is defined and that you have the NetApp recommended settings configured for your ONTAP LUNs.

Steps

1. Verify that the `/etc/multipath.conf` file exits. If the file doesn't exist, create an empty, zero-byte file:

   ```
   touch /etc/multipath.conf
   ```

2. The first time the `multipath.conf` file is created, you might need to enable and start the multipath services to load the recommended settings:

   ```
   systemctl enable multipathd
   ```

   ```
   systemctl start multipathd
   ```

3. Each time you boot the host, the empty `/etc/multipath.conf` zero-byte file automatically loads the NetApp recommended host multipath parameters as the default settings. You shouldn't need to make changes to the `/etc/multipath.conf` file for your host because the operating system is compiled with the multipath parameters that recognize and manage ONTAP LUNs correctly.

   The following table shows the Linux OS native compiled multipath parameter settings for ONTAP LUNs.

Show parameter settings

Parameter	Setting
detect_prio	yes
dev_loss_tmo	"infinity"
failback	immediate
fast_io_fail_tmo	5
features	"2 pg_init_retries 50"
flush_on_last_del	"yes"
hardware_handler	"0"
no_path_retry	queue
path_checker	"tur"
path_grouping_policy	"group_by_prio"
path_selector	"service-time 0"
polling_interval	5
prio	"ontap"
product	LUN
retain_attached_hw_handler	yes
rr_weight	"uniform"
user_friendly_names	no
vendor	NETAPP

4. Verify the parameter settings and path status for your ONTAP LUNs:

```
multipath -ll
```

The default multipath parameters support ASA, AFF, and FAS configurations. In these configurations, a single ONTAP LUN shouldn't require more than four paths. Having more than four paths can cause problems during a storage failure.

The following example outputs show the correct parameter settings and path status for ONTAP LUNs in an ASA, AFF, or FAS configuration.

ASA configuration

An ASA configuration optimizes all paths to a given LUN, keeping them active. This improves performance by serving I/O operations through all paths at the same time.

Show example

```
multipath -ll
3600a098038314c4a433f577471797958 dm-2 NETAPP,LUN C-Mode
size=180G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 14:0:0:0  sdc  8:32   active ready running
  |- 17:0:0:0  sdas 66:192 active ready running
  |- 14:0:3:0  sdar 66:176 active ready running
  `- 17:0:3:0  sdch 69:80  active ready running
```

AFF or FAS configuration

An AFF or FAS configuration should have two groups of paths with higher and lower priorities. Higher priority Active/Optimized paths are served by the controller where the aggregate is located. Lower priority paths are active but non-optimized because they are served by a different controller. Non-optimized paths are only used when optimized paths aren't available.

The following example displays the output for an ONTAP LUN with two Active/Optimized paths and two Active/Non-Optimized paths:

Show example

```
multipath -ll
3600a0980383149764b5d567257516273 dm-0 NETAPP,LUN C-Mode
size=150G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 16:0:3:0  sdcg 69:64  active ready running
| `- 10:0:0:0  sdb  8:16   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 10:0:1:0  sdc  8:32   active ready running
  `- 16:0:2:0  sdcf 69:48  active ready running
```

Step 4: Confirm the iSCSI configuration for your host

Ensure that iSCSI is configured correctly for your host.

About this task

You perform the following steps on the iSCSI host.

Steps

1. Verify that the iSCSI initiator package (iscsi-initiator-utils) is installed:

```
rpm -qa | grep iscsi-initiator-utils
```

   You should see an output similar to the following example:

```
iscsi-initiator-utils-6.2.1.11-0.git4b3e853.el9.x86_64
```

2. Verify the iSCSI initiator node name, which is located in the `/etc/iscsi/initiatorname.iscsi` file:

```
InitiatorName=iqn.YYYY-MM.com.<vendor>:<host_name>
```

3. Configure the iSCSI session timeout parameter located in the `/etc/iscsi/iscsid.conf` file:

```
node.session.timeo.replacement_timeout = 5
```

   The iSCSI `replacement_timeout` parameter controls how long the iSCSI layer should wait for a timed-out path or session to reestablish itself before failing any commands on it. You should set the value of `replacement_timeout` to 5 in the iSCSI configuration file.

4. Enable the iSCSI service:

```
$systemctl enable iscsid
```

5. Start the iSCSI service:

```
$systemctl start iscsid
```

6. Verify that the iSCSI service is running:

```
$systemctl status iscsid
```

Show example

```
● iscsid.service - Open-iSCSI
     Loaded: loaded (/usr/lib/systemd/system/iscsid.service;
enabled; preset: disabled)
     Active: active (running) since Tue 2025-12-02 11:36:21 EST; 2
weeks 1 day ago
TriggeredBy: ● iscsid.socket
       Docs: man:iscsid(8)
             man:iscsiuio(8)
             man:iscsiadm(8)
   Main PID: 2263 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1 (limit: 816061)
     Memory: 18.5M
        CPU: 14.480s
     CGroup: /system.slice/iscsid.service
             └─2263 /usr/sbin/iscsid -f -d2
```

7. Discover the iSCSI targets:

```
$iscsiadm --mode discovery --op update --type sendtargets --portal
<target_IP>
```

show example

```
iscsiadm --mode discovery --op update --type sendtargets --portal
192.168.30.87
192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23
```

8. Log in to the targets:

```
$iscsiadm --mode node -l all
```

9. Set iSCSI to log in automatically when the host boots:

```
$iscsiadm --mode node -T <target_name> -p <ip:port> -o update -n
node.startup -v automatic
```

You should see an output similar to the following example:

```
iscsiadm --mode node -T iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 -p
192.168.30.87:3260 -o update -n node.startup -v automatic
```

10. Verify the iSCSI sessions:

```
$iscsiadm --mode session
```

Show example

```
iscsiadm --mode session
tcp: [1] 192.168.30.87:3260,1139 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [2] 192.168.31.97:3260,1142 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [3] 192.168.31.87:3260,1141 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
tcp: [4] 192.168.30.97:3260,1140 iqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:vs.23 (non-flash)
```

Step 5: Optionally, exclude a device from multipathing

If required, you can exclude a device from multipathing by adding the WWID for the unwanted device to the "blacklist" stanza for the `multipath.conf` file.

Steps

1. Determine the WWID:

```
/lib/udev/scsi_id -gud /dev/sda
```

"sda" is the local SCSI disk that you want to add to the blacklist.

An example WWID is `360030057024d0730239134810c0cb833`.

2. Add the WWID to the "blacklist" stanza:

```
blacklist {
         wwid   360030057024d0730239134810c0cb833
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
        devnode "^cciss.*"
}
```

Step 6: Customize multipath parameters for ONTAP LUNs

If your host is connected to LUNs from other vendors and any of the multipath parameter settings are overridden, you need to correct them by adding stanzas later in the `multipath.conf` file that apply specifically to ONTAP LUNs. If you don't do this, the ONTAP LUNs might not work as expected.

Check your `/etc/multipath.conf` file, especially in the defaults section, for settings that might be overriding the default settings for multipath parameters.

> You shouldn't override the recommended parameter settings for ONTAP LUNs. These settings are required for optimal performance of your host configuration. Contact NetApp support, your OS vendor, or both for more information.

The following example shows how to correct an overridden default. In this example, the `multipath.conf` file defines values for `path_checker` and `no_path_retry` that aren't compatible with ONTAP LUNs, and you can't remove these parameters because ONTAP storage arrays are still attached to the host. Instead, you correct the values for `path_checker` and `no_path_retry` by adding a device stanza to the `multipath.conf` file that applies specifically to the ONTAP LUNs.

Show example

```
defaults {
   path_checker      readsector0
   no_path_retry     fail
}

devices {
   device {
      vendor          "NETAPP"
      product         "LUN"
      no_path_retry   queue
      path_checker    tur
   }
}
```

Step 7: Review the known issues

RHEL 8.x with ONTAP storage has the following known issues.

8.1

NetApp Bug ID	Title	Description
1275843	Kernel disruption might occur on Red Hat Enterprise Linux 8.1 with QLogic QLE2672 16GB FC HBA during storage failover operation	Kernel disruption might occur during storage failover operations on the Red Hat Enterprise Linux 8.1 kernel with a QLogic QLE2672 Fibre Channel (FC) host bus adapter (HBA). The kernel disruption causes Red Hat Enterprise Linux 8.1 to reboot, leading to application disruption. If the kdump mechanism is enabled, the kernel disruption generates a vmcore file located in the/var/crash/ directory. You can check the vmcore file to determine the cause of the disruption.A storage failover with the QLogic QLE2672 HBA event affects the "kmem_cache_alloc+131" module. You can locate the event in the vmcore file by finding the following string: " [exception RIP: kmem_cache_alloc+131]"After the kernel disruption, reboot the Host OS and recover the operating system. Then restart the applications

NetApp Bug ID	Title	Description
1275838	Kernel disruption occurs on Red Hat Enterprise Linux 8.1 with QLogic QLE2742 32GB FC HBA during storage failover operations	Kernel disruption occurs during storage failover operations on the Red Hat Enterprise Linux 8.1 kernel with a QLogic QLE2742 Fibre Channel (FC) host bus adapter (HBA). The kernel disruption causes Red Hat Enterprise Linux 8.1 to reboot, leading to application disruption. If the kdump mechanism is enabled, the kernel disruption generates a vmcore file located in the/var/crash/ directory. You can check the vmcore file to determine the cause of the disruption. A storage failover with the QLogic QLE2742 HBA event affects the "kmem_cache_alloc+131"module. You can locate the event in the vmcore file by finding the following string: " [exception RIP: kmem_cache_alloc+131]"After the kernel disruption, reboot the Host OS and recover the operating system. Then restart the applications.
1266250	Login to multiple paths fails during the Red Hat Enterprise Linux 8.1 installation on iSCSI SAN LUN	You cannot login to multiple paths during the Red Hat Enterprise Linux 8.1 installation on iSCSI SAN LUN multipath devices. Installation is not possible on multipath iSCSI device and the multipath service is not enabled on the SAN boot device.

8.0

NetApp Bug ID	Title	Description
1238719	Kernel disruption on RHEL8 with QLogic QLE2672 16GB FC during storage failover operations	Kernel disruption might occur during storage failover operations on a Red Hat Enterprise Linux (RHEL) 8 kernel with a QLogic QLE2672 host bus adapter (HBA). The kernel disruption causes the operating system to reboot. The reboot causes application disruption and generates the vmcore file under the /var/crash/directory if kdump is configured. Use the vmcore file to identify the cause of the failure. In this case, the disruption is in the “kmem_cache_alloc+160” module. It is logged in the vmcore file with the following string: "[exception RIP: kmem_cache_alloc+160]". Reboot the host OS to recover the operating system and then restart the application.
1226783	RHEL8 OS boots up to "emergency mode" when more than 204 SCSI devices are mapped on all Fibre Channel (FC) host bus adapters (HBA)	If a host is mapped with more than 204 SCSI devices during an operating systemreboot process, the RHEL8 OS fails to boot up to "normal mode" and enters "emergency mode". This results in most of the host services becoming unavailable.
1230882	Creating a partition on an iSCSI multipath device during the RHEL8 installation is not feasible.	iSCSI SAN LUN multipath devices are not listed in disk selection during RHEL 8 installation. Consequently, the multipath service is not enabled on the SAN boot device.
1235998	The "rescan-scsi-bus.sh -a" command does not scan more than 328 devices	If a Red Hat Enterprise Linux 8 host maps with more than 328 SCSI devices, the host OS command "rescan-scsi-bus.sh -a" only scans 328 devices. The host does not discover any remaining mapped devices.

NetApp Bug ID	Title	Description
1231087	Remote ports transit to a blocked state on RHEL8 with Emulex LPe16002 16GB FC during storage failover operations	Remote ports transit to a blocked state on RHEL8 with Emulex LPe16002 16GB Fibre Channel (FC) during storage failover operations. When the storage node returns to an optimal state, the LIFs also come up and the remote port state should read "online". Occasionally, the remote port state might continue to read as "blocked" or "not present". This state can lead to a "failed faulty" path to LUNs at the multipath layer
1231098	Remote ports transit to blocked state on RHEL8 with Emulex LPe32002 32GB FC during storage failover operations	Remote ports transit to a blocked state on RHEL8 with Emulex LPe32002 32GBFibre Channel (FC) during storage failover operations. When the storage node returns to an optimal state, the LIFs also come up and the remote port state should read "online". Occasionally, the remote port state might continue to read as "blocked" or "not present". This state can lead to a "failed faulty" path to LUNs at the multipath layer.

What's next?

- Learn about using the Linux Host Utilities tool.
- Learn about ASM mirroring

  Automatic Storage Management (ASM) mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate failure group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM doesn't mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites. See Oracle databases on ONTAP for further information.

- Learn about Red Hat Linux Virtualization (KVM)

  Red Hat Linux can serve as a KVM host. This enables you to run multiple virtual machines on a single physical server using the Linux Kernel-based Virtual Machine (KVM) technology. The KVM host doesn't require explicit host configuration settings for ONTAP LUNs.