

# iSCSI Initiator

With [Wikipedia:iSCSI](#) you can access storage over an IP-based network.

The exported storage entity is the **target** and the importing entity is the **initiator**.

This article describes how to access an iSCSI target with the [Open-iSCSI \(https://github.com/open-iscsi/open-iscsi\)](https://github.com/open-iscsi/open-iscsi) initiator.

## Related articles

iSCSI Target

iSCSI Boot

## Contents

- [1 Installation](#)
- [2 Overview](#)
- [3 Configuration](#)
  - [3.1 Start the Service](#)
  - [3.2 Target discovery](#)
  - [3.3 Delete obsolete targets](#)
  - [3.4 Login to available targets](#)

- 3.5 Info
- 3.6 Online resize of volumes
- 4 Tips & Troubleshooting
- 5 See also

## Installation

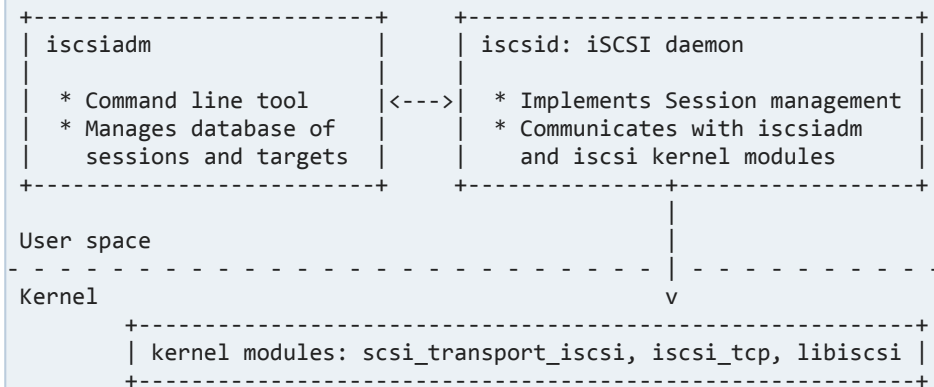
**Install** the **open-iscsi** (<https://www.archlinux.org/packages/?name=open-iscsi>) package from the **official repositories**.

**Note:** An older initiator, **Linux-iSCSI** (<http://sourceforge.net/projects/linux-iscsi/>), was merged with Open-iSCSI in April 2005. This should not be confused with **linux-iscsi.org** (<http://linux-iscsi.org/>), the website for the LIO target.

## Overview

The following diagram shows how the Components work together. A more detailed version can be found here: **Open-iSCSI modules** (<https://archive.is/HHYKR/90a7a1c178a2c069a7cbc0b578b6fb5854f827fa.jpg>)

```
+-----+
| Targets & Sessions configuration Database (DBM based) |
+-----+
```



From the Open-iSCSI **README** (<https://github.com/open-iscsi/open-iscsi>):

Persistent configuration is implemented as a DBM database, which contains two tables:

- Discovery table (`/etc/iscsi/send_targets`)
- Node table (`/etc/iscsi/nodes`)

## Configuration

### Start the Service

`iscsid` is managed by a systemd Unit.

Start `open-iscsi.service` using `systemd`.

If the SCSI target requires authentication by the initiator, the configuration file `/etc/iscsi/iscsid.conf` may need to be updated.

The following parameters are used for authenticating a login session of an initiator to a target and for the target to establish a session back to the initiator

```
node.session.auth.authmethod = CHAP
node.session.auth.username = <username in target>
node.session.auth.password = <password in target>
node.session.auth.username_in = <username in initiator>
node.session.auth.password_in = <password in initiator>
```

The following parameters are used for authenticating a discovery session of an initiator to a target and for the target to establish a session back to the initiator.

```
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = <username in target>
discovery.sendtargets.auth.password = <password in target>
discovery.sendtargets.auth.username_in = <username in initiator>
discovery.sendtargets.auth.password_in = <password in initiator>
```

**Warning:** No two passwords may be the same. This means that you need four unique passwords in the configuration above.

## Target discovery

```
# iscsiadm -m discovery -t sendtargets -p <portalip>
```

## Delete obsolete targets

```
# iscsiadm -m discovery -p <portalip> -o delete
```

## Login to available targets

```
# iscsiadm -m node -L all
```

or login to specific target

```
# iscsiadm -m node --targetname=<targetname> --login
```

logout:

```
# iscsiadm -m node -U all
```

## Info

For running session

```
# iscsiadm -m session -P 3
```

The last line of the above command will show the name of the attached dev e.g

```
Attached scsi disk sdd State: running
```

For the known nodes

```
# iscsiadm -m node
```

## Online resize of volumes

If the iscsi blockdevice contains a partitiontable, you will not be able to do an online resize. In this case you have to unmount the filesystem and alter the size of the affected partition.

1. Rescan active nodes in current session

```
# iscsiadm -m node -R
```

2. If you use multipath, you also have to rescan multipath volume information.

```
# multipathd -k"resize map sdx"
```

3. Finally resize the filesystem.

```
# resize2fs /dev/sdx
```

## Tips & Troubleshooting

You can also check where the attached iSCSI devices are located in the /dev tree with `ls -lh /dev/disk/by-path/*`.

At the server (target) you might need to include the client iqn from `/etc/iscsi/initiatorname.iscsi` in the acl configuration.

Many of the `iscsiadm` operations require that the iSCSI daemon `iscsid` is running. To verify that this is the case, **check the status** of the `open-iscsi.service`.

## See also

- **iSCSI Boot** Booting Arch Linux with / on an iSCSI target.

Retrieved from "[https://wiki.archlinux.org/index.php?title=iSCSI\\_Initiator&oldid=481781](https://wiki.archlinux.org/index.php?title=iSCSI_Initiator&oldid=481781)"

- This page was last edited on 12 July 2017, at 06:49.
- Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.