

# Openfiler Lab 3: SAN Configuration

<b>OPENFILER LAB 3: SAN CONFIGURATION</b>	<b>1</b>
<b>References</b>	<b>1</b>
<b>Background</b>	<b>1</b>
<b>Objectives</b>	<b>1</b>
<b>Procedures</b>	<b>1</b>
<b>Part 1: Configuring Openfiler iSCSI Target</b>	<b>2</b>
Task 1: Openfiler iSCSI Target	2
<b>Part 2: Configuring Linux iSCSI Initiator</b>	<b>3</b>
Task 1: iSCSI Software	3
Task 2: Configure Storage	3
Task 3: iSCSI Software installation and configuration using Microsoft Windows OS	4

## References

- [1] Openfiler Overview: <http://www.openfiler.com/products>
- [2] Openfiler Architecture: <http://www.openfiler.com/products/openfiler-architecture>
- [3] Openfiler Feature Summary: <http://www.openfiler.com/products/feature-summary/feature-summary>
- [4] Microsoft iSCSI Initiator Step-by-Step Guide: <http://technet.microsoft.com/en-us/library/ee338476%28WS.10%29.aspx>
- [5] nixCraft Windows iSCSI Configuration: <http://www.cyberciti.biz/tips/windows-2003-iscsi-initiator-software-connection.html>

## Background

Openfiler is a flexible, open source enterprise storage solution with support for a variety of common, industry standard access protocols. Because it is based on the Linux operating system, Openfiler can be run on most modern hardware without issue. Openfiler supports client access to storage at both the file and block levels. At the file level, Openfiler can export storage using such network-attached storage (NAS) protocols as NFS and CIFS, among others. Openfiler supports iSCSI and Fibre Channel storage area network (SAN) protocols for block level data access [1]. For a more detailed discussion of the Openfiler architecture, refer to the Openfiler website [2].

In order to simplify deployment, Openfiler provides a powerful web-based GUI to configure and control its various services. The GUI includes support for managing storage, shares, user accounts, quotas, and network protocols such as NFS and CIFS. Ideally, users should be able to configure Openfiler to meet their needs without ever running a single Linux command or editing a configuration file by hand [3].

## Objectives

Upon completing this lab, students should understand how to use Openfiler to accomplish the following tasks:

- Configure access control for SAN services
- Configure iSCSI targets and start the iSCSI target service
- Install iSCSI initiator software on Linux clients
- Configure Linux clients for access to iSCSI devices

**Note:** Openfiler Lab 1: Storage & Authentication Configuration is a prerequisite for this lab and should be completed before continuing further.

## Procedures

Follow the steps below to perform the lab. Take a screenshot or screenshots where noted in red to demonstrate successful completion. Also, there are questions throughout the lab that you are required to answer. Please take time to think about your response as these questions are weighted heavily in the grading rubric.

**Note:** IP addresses used in the lab are from network **192.168.255.0/24**. Addresses used in instruction need to be modified appropriately.

## Part 1: Configuring Openfiler iSCSI Target

iSCSI is a common SAN protocol used in enterprise environments. In this part of the lab you will configure Openfiler as an iSCSI target. You will also configure an ACL to control access to the iSCSI target.

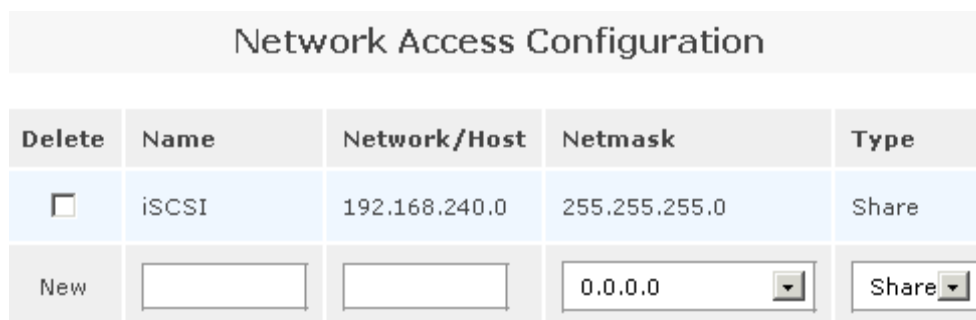
### Task 1: Openfiler iSCSI Target

#### Step 1: Create iSCSI Network ACL - Screenshot(s) 5%

Open a web browser on the Windows host and connect to the Openfiler web interface at <https://192.168.240.100:446/>, then login with the username `openfiler` and the password `password`.

Click `System` in the top menu bar. From this menu you can manage various components of the Openfiler server. You will first create an ACL to control client access to iSCSI.

Scroll down to the `Network Access Configuration` section. Enter `iSCSI` as the Name of the ACL, enter `192.168.240.0` as the Network/Host, select `255.255.255.0` as the Netmask, and select `share` as the Type. Click `Update`. You will be presented with the screen in Figure 1.



Delete	Name	Network/Host	Netmask	Type
<input type="checkbox"/>	iSCSI	192.168.240.0	255.255.255.0	Share

New

Figure 1. iSCSI Network ACL

**Note:** The iSCSI protocol assumes that only one host is connected to a given iSCSI target, and as such does not include a file locking mechanism. If more than one host is connected, data on the iSCSI target will be corrupted. The ACL you created allows access for an entire network, because the client's IP address could change via DHCP. If you suspect that multiple hosts might attempt to connect to a single iSCSI target, however, you should restrict the ACL to a single host.

#### Step 2: Start iSCSI Service - Screenshot(s) 5%

Click `Services` in the top menu bar. From this menu you can enable and disable the various services that Openfiler uses. You will use this menu to enable the iSCSI target server.

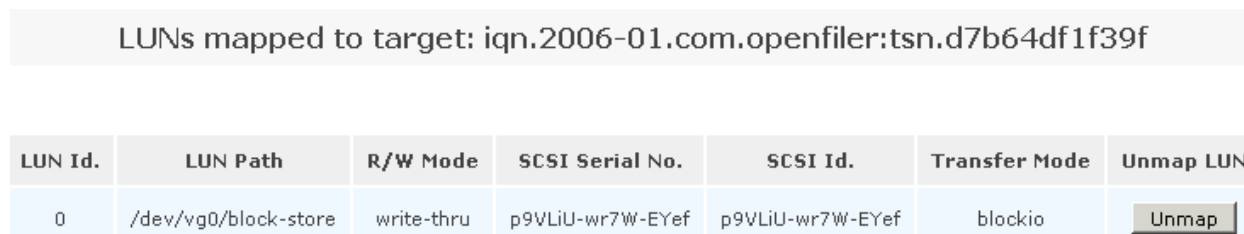
Under the `Manage Services` section, click `Enable` next to `iSCSI target server`. The page will reload, and you will see that the iSCSI target server has been enabled.

#### Step 3: Create iSCSI Target - Screenshot(s) 5%

Click `Volumes` in the top menu bar. Under the `Volumes` section menu, click `iSCSI Targets`. From here you can create and configure Openfiler iSCSI targets.

Under the `Add new iSCSI target` section, accept the default `Target IQN` and click `Add`. You will be presented with a list of settings for your new iSCSI target.

Click the `LUN Mapping` tab. Under the `Map New LUN to Target` section, accept the default settings and click `Map`. You will be presented with the screen in Figure 2.



LUNs mapped to target: iqn.2006-01.com.openfiler:tsn.d7b64df1f39f						
LUN Id.	LUN Path	R/W Mode	SCSI Serial No.	SCSI Id.	Transfer Mode	Unmap LUN
0	/dev/vg0/block-store	write-thru	p9VLIU-wr7W-EYef	p9VLIU-wr7W-EYef	blockio	<input type="button" value="Unmap"/>

Figure 2. iSCSI Target LUN Mapping

Click the `Network ACL` tab. Set the `Access` to `Allow` for the `iSCSI` ACL, and click `Update`. You will be presented with the screen in Figure 3.

iSCSI host access configuration for target "iqn.2006-01.com.openfiler:tsn.d7b64df1f39f"

Name	Network/Host	Netmask	Access
iSCSI	192.168.240.0	255.255.255.0	Allow

Figure 3. iSCSI Network ACL Applied

**Question 1:** Click the CHAP Authentication tab. Why would enabling CHAP authentication for iSCSI be beneficial? Would CHAP protect against unauthorized users intercepting sensitive data? What else should you do in order to secure iSCSI? - **35%**

## Part 2: Configuring Linux iSCSI Initiator

iSCSI initiators function as the client in the iSCSI protocol, and are available for a variety of operating systems, including versions of Windows and Linux. In this part of the lab you will configure an iSCSI initiator on Linux. You will also partition, format, and mount the iSCSI target.

**Note:** To configure a Windows host as an iSCSI initiator, please refer to Microsoft's User's Guide for iSCSI [4], or to the nixCraft guide [5].

### Task 1: iSCSI Software

#### Step 1: Install & Start iSCSI Software - **Screenshot(s) 5%**

Login to the Linux host as a **root** (or use **sudo** command).

In order to install and start the iSCSI initiator software, run the following commands:

```
[root@localhost ~]# yum -y install iscsi-initiator-utils  
<Output Omitted>
```

```
Installed:  
iscsi-initiator-utils.i386 0:6.2.0.871-0.12.el5_4.1  
Complete!
```

```
[root@localhost ~]# service iscsi start ; chkconfig iscsi on  
<Output Omitted>
```

**Note:** Installation and configuration procedure of iSCSI initiator on Ubuntu can be found at:

<http://www.howtoforge.com/using-iscsi-on-ubuntu-10.04-initiator-and-target>

#### Step 2: Discover & Configure IQN - **Screenshot(s) 5%**

You will now use the **iscsiadm** command to discover the iSCSI target IQN. Then, you will configure the iSCSI initiator software with the discovered IQN, and restart the iSCSI service. Run the following command:

```
[root@localhost ~]# iscsiadm -m discovery -t sendtargets -p 192.168.240.100  
192.168.240.100:3260,1 iqn.2006-01.com.openfiler:tsn.d7b64df1f39f
```

Use the text editor of your choice to edit the **/etc/iscsi/initiatorname.iscsi** file and add the discovered IQN. The file should look like this:

```
[root@localhost ~]# cat /etc/iscsi/initiatorname.iscsi  
InitiatorName=iqn.2006-01.com.openfiler:tsn.d7b64df1f39f
```

Run the following command to restart the iSCSI service:

```
[root@localhost ~]# service iscsi restart  
<Output Omitted>
```

**Note:** Your IQN may be different from the example. You should use the IQN returned by the **iscsiadm** command. You should see that the iSCSI software successfully sets up and logs in to the iSCSI target when you restart the iSCSI service.

### Task 2: Configure Storage

#### Step 1: Partition iSCSI Device - **Screenshot(s) 5%**

The new iSCSI device is not yet partitioned. Run the following command to view the newly added iSCSI device:

```
[root@localhost ~]# fdisk -l
<Output Omitted>
Disk /dev/sdb: 2147 MB, 2147483648 bytes
67 heads, 62 sectors/track, 1009 cylinders
Units = cylinders of 4154 * 512 = 2126848 bytes
Disk /dev/sdb doesn't contain a valid partition table
```

Use the **fdisk** command to partition the iSCSI device. Create a new primary partition filling the entire device. Use the default partition type. When you are finished, the **fdisk -l** command should return the following output:

```
[root@localhost ~]# fdisk -l
<Output Omitted>
Disk /dev/sdb: 2147 MB, 2147483648 bytes
67 heads, 62 sectors/track, 1009 cylinders
Units = cylinders of 4154 * 512 = 2126848 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 1 1009 2095662 83 Linux
```

**Note:** Your iSCSI device may differ from the example. You should partition the device that does not have a partition table. Do not partition the device that the Linux operating system is installed on!

### Step 2: Format iSCSI Device - **Screenshot(s) 5%**

Run the following command to format the iSCSI device:

```
[root@localhost ~]# mke2fs -j -L iscsi /dev/sdb1
<Output Omitted>
```

**Note:** Your iSCSI device may differ from the example. You should format the device that you partitioned.

### Step 3: Mount iSCSI Device - **Screenshot(s) 5%**

Run the following command to create a mount point for the iSCSI device:

```
[root@localhost ~]# mkdir /mnt/iscsi
```

Using the text editor of your choice, add the following line to the `/etc/fstab` file on the Linux host to make the iSCSI mount persistent across reboots:

```
LABEL=iscsi /mnt/iscsi ext3 _netdev 0 0
```

Run the following commands to mount the iSCSI target:

```
[root@localhost ~]# mount -a
[root@localhost ~]# mount
<Output Omitted>
/dev/sdb1 on /mnt/iscsi type ext3 (rw,_netdev)
```

## Task 3: iSCSI Software installation and configuration using Microsoft Windows OS

**Step 1: Install & Start iSCSI Software - **Screenshot(s) 5%****

**Step 2: Discover & Configure IQN - **Screenshot(s) 5%****

**Step 3: Partition iSCSI Device - **Screenshot(s) 5%****

**Step 4: Format iSCSI Device - **Screenshot(s) 5%****

**Step 5: Mount iSCSI Device - **Screenshot(s) 5%****