Copyright © 2018 Nexenta Systems™, ALL RIGHTS RESERVED

Notice: No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose, without the express written permission of Nexenta Systems (hereinafter referred to as “Nexenta”).

Nexenta reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. Nexenta products and services only can be ordered under the terms and conditions of Nexenta Systems’ applicable agreements. All of the features described in this document may not be available currently. Refer to the latest product announcement or contact your local Nexenta Systems sales office for information on feature and product availability. This document includes the latest information available at the time of publication.

Nexenta, NexentaStor, NexentaEdge, NexentaFusion, and NexentaConnect are registered trademarks of Nexenta Systems in the United States and other countries. All other trademarks, service marks, and company names in this document are properties of their respective owners.

This document applies to the following product versions:

<table>
<thead>
<tr>
<th>Product</th>
<th>Versions supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>NexentaEdge™</td>
<td>2.1.3</td>
</tr>
<tr>
<td>Ubuntu Linux</td>
<td>Ubuntu 16.04.4 LTS</td>
</tr>
<tr>
<td>CentOS</td>
<td>7.4 with ELRepo LT kernel only</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td></td>
</tr>
<tr>
<td>Chef</td>
<td>12.x</td>
</tr>
<tr>
<td>OpenStack Swift</td>
<td>Newton, Ocata, Pike</td>
</tr>
<tr>
<td>OpenStack Cinder</td>
<td></td>
</tr>
<tr>
<td>Docker</td>
<td>1.13</td>
</tr>
<tr>
<td>S3cmd tool</td>
<td>1.6.1</td>
</tr>
</tbody>
</table>
# Contents

1 Installation Overview ............................................. 1
   About This Installation Guide .................................. 1
   Components of a NexentaEdge Cluster .......................... 1
   NexentaEdge Deployment Options ............................... 4
      Deployment Within Docker Containers ....................... 4
      Deployment Using Packages in an ISO File .................. 4
   NexentaEdge Installation Procedure ......................... 5

2 Prerequisites ...................................................... 6
   Networking Requirements ....................................... 6
      Switch Hardware Requirements ............................... 6
      Replicast Network Requirements ............................. 6
   Data and Gateway Node Requirements .......................... 6
      Hardware Requirements ...................................... 7
      Software Requirements ...................................... 7
   Deployment Workstation Requirements ........................ 8
      Requirements for the Chef Environment ...................... 8

3 Installing NexentaEdge ........................................... 9
   Before You Begin ............................................... 9
   Planning Server Node Deployment .............................. 9
   Network Switch Configuration ................................ 10
   Failure Domains in a NexentaEdge Cluster .................... 10
      Disk-Level Failure Domain .................................. 11
      Server-Level Failure Domain ................................. 11
      Zone-Level Failure Domain .................................. 12
   Downloading the NexentaEdge Deployment and Administration Tools ........................................... 14
      Extracting the NEDEPLOY and NEADM Archives .......... 14
      Preparing Storage Devices for NexentaEdge Deployment ........................................... 15
      Deploying NexentaEdge ....................................... 15
      RedHat/CentOS Considerations ............................... 16
8 Upgrading NexentaEdge to the Latest Release .............................................. 48
  Overview ........................................................................................................ 48
    Disruptive Upgrade ..................................................................................... 48
    Non-Disruptive Upgrade ........................................................................... 48
  Upgrade Procedure ......................................................................................... 49
    Upgrading NexentaEdge Using the NEDEPLOY Tool ................................... 49
    Upgrading NexentaEdge Using Ubuntu or Red Hat / CentOS Commands ...... 50
    Syncing the NexentaEdge GUI With the Upgraded Nodes ....................... 52
Preface

This documentation presents information specific to Nexenta products. The information is for reference purposes and is subject to change.

Intended Audience

This documentation is intended for Object Storage Administrators and Network Administrators and assumes that you have a working knowledge of UNIX. The document also assumes that you have experience with data storage concepts, such as object storage, ZFS, iSCSI, NFS, CIFS, and so on.

Documentation History

The following table lists the released revisions of this documentation.

Table 1: Documentation Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000-nedge-1.0-000010-A</td>
<td>July, 2015</td>
<td>1.0 GA release</td>
</tr>
<tr>
<td>2000-nedge-1.0-000010-B</td>
<td>September, 2015</td>
<td>1.0.1 release</td>
</tr>
<tr>
<td>2000-nedge-1.1-000010-A</td>
<td>February, 2016</td>
<td>1.1.0 GA release</td>
</tr>
<tr>
<td>2000-nedge-1.1-FP1-000010-B</td>
<td>May, 2016</td>
<td>1.1.0 FP1 release</td>
</tr>
<tr>
<td>2000-nedge-1.1-FP2-000010-A</td>
<td>May, 2016</td>
<td>1.1.0 FP2 release</td>
</tr>
<tr>
<td>2000-nedge-1.1-FP3-000010-A</td>
<td>June, 2016</td>
<td>1.1.0 FP3 release</td>
</tr>
<tr>
<td>2000-nedge-2.0-000010-A</td>
<td>April, 2017</td>
<td>2.0.0 GA release</td>
</tr>
<tr>
<td>2000-nedge-2.1-000010-A</td>
<td>December, 2017</td>
<td>2.1.2 GA release</td>
</tr>
<tr>
<td>2000-nedge-2.1.3-000010-A</td>
<td>April, 2018</td>
<td>2.1.3 release</td>
</tr>
</tbody>
</table>

Comments

Your comments and suggestions to improve this documentation are greatly appreciated. Send any feedback to doc.comments@nexenta.com and include the documentation title, number, and revision. Refer to specific pages, sections, and paragraphs whenever possible.
Installation Overview

This chapter includes the following topics:

- NexentaEdge Installation Procedure
- Components of a NexentaEdge Cluster
- NexentaEdge Installation Procedure

About This Installation Guide

This guide contains the procedures for installing the basic components of a NexentaEdge cluster. It details the requirements for the servers and switches, and contains procedures for configuring the servers to be part of a NexentaEdge cluster.

After performing the tasks in this guide, you will have a NexentaEdge cluster that is capable of accepting requests from clients running storage services that NexentaEdge supports, including iSCSI block storage, OpenStack Cinder, Amazon S3, and NFS.

See the NexentaEdge User Guide for information about adding servers and storage devices to your cluster and other administrative tasks.

Components of a NexentaEdge Cluster

From a physical perspective, a NexentaEdge cluster is a collection of server devices connected via a high-performance 10 Gigabit switch. From a logical perspective, a NexentaEdge cluster consists of data nodes and gateway nodes that communicate over a Replicast network. The cluster provides storage services over an external network using the protocols that NexentaEdge supports, including OpenStack Swift, OpenStack Cinder, Amazon S3, iSCSI, and NFS. Note that the term “NexentaEdge cluster” in this manual refers to a logical NexentaEdge cluster.

A NexentaEdge deployment consists of a single physical cluster and one or more logical clusters. Each logical cluster may have multiple namespaces configured for different tenants.

Figure 1-1 shows the components of a NexentaEdge cluster.
A NexentaEdge cluster consists of the following components. A given device may have multiple roles assigned to it; for example, a server may be configured as a data node, gateway node and management controller.

- **Data nodes**

  The data nodes collectively provide the storage for the NexentaEdge cluster. Objects are broken into chunks and distributed across the data nodes using the Replicast protocol. The set of data nodes where the chunks are stored or retrieved is determined based on server load and capacity information.

  Data nodes must be configured with interfaces to an IPv6 Replicast network for data distribution and storage and to an IPv4 network (either an external network or dedicated management network) for initial configuration with the NEDEPLOY tool and subsequent administration with the NEADM tool.

  After initial configuration, data nodes require only a connection to the Replicast network, since administration of the data nodes is done by the deployment workstation via the management controller node, which has connectivity to both the management network and the Replicast network.
• Gateway nodes

Gateway nodes provide the connection between external clients and the data stored in the NexentaEdge cluster. Gateway nodes accept and respond to client requests, translating them into actions performed in the NexentaEdge cluster. Gateway nodes are provisioned with interfaces to the external network, Replicast network, and management network (if different from the external network).

When you configure a NexentaEdge cluster, you indicate which storage service(s) you want to provide for a given tenant, then specify which should be the gateway nodes for that cluster/tenant/service combination. Gateway nodes run the services within lightweight containers.

• Replicast network

The Replicast network is an isolated IPv6 VLAN used for communication and data transfer among the data nodes and gateway nodes in the NexentaEdge cluster. The Replicast protocol provides the means for efficient storage and retrieval of data in the cluster.

• Deployment workstation

The deployment workstation is the system from which you deploy and configure the NexentaEdge software to the other nodes. NexentaEdge uses the Chef environment for installation. You deploy NexentaEdge using a Chef Solo instance packaged with the NexentaEdge software.

To deploy NexentaEdge to the nodes in the cluster using the NEDEPLOY tool, the deployment workstation must have IPv4 network connectivity to the nodes, either through a management network or an external network.

• Management controller node

One of the data nodes in the cluster is designated the management controller node. The management controller translates external cluster-wide behavior into internal component-specific configuration, and may provide the connection between the deployment workstation and the data nodes in the Replicast network. Only one of the data nodes in the cluster can be a management controller. The management controller node needs to have network connectivity to both the deployment workstation and to the other nodes in the cluster.

• External network

External clients store and retrieve data in the NexentaEdge cluster by communicating with gateway nodes on the external network. The external network may use IPv4 or IPv6 and is likely to carry traffic unrelated to NexentaEdge.

• Management network

To aid in deploying and administering the NexentaEdge cluster, you may elect to place the deployment workstation and data nodes in a dedicated IPv4 management network. The NEDEPLOY and NEADM tools, running on the deployment workstation, send configuration information to and receive status information from the data nodes over this network.

• External clients

External clients are end-user machines that access data stored in the NexentaEdge cluster via gateway nodes. External clients access data in the cluster using APIs of the storage services NexentaEdge supports: OpenStack Swift and Amazon S3 via HTTP/REST, NFS or iSCSI block storage.
From a client perspective, the NexentaEdge cluster acts as an OpenStack Swift or Amazon S3 object storage system. To accommodate applications that expect block storage, the NexentaEdge cluster can act as an iSCSI or OpenStack Cinder target, or NFS share.

**NexentaEdge Deployment Options**

NexentaEdge supports a variety of deployment options, including deployment using Docker containers and installing from an ISO file.

**Deployment Within Docker Containers**

By default, NexentaEdge nodes are deployed within Docker containers, so that multiple logical nodes reside on a single physical server.

Since multiple Docker containers can reside on a single physical server, a given server can have multiple nodes deployed to it, which allows the size of the cluster to scale to the full storage capacity of the server. In this kind of configuration, all gateway services run as Docker containers.

**Deployment Using Packages in an ISO File**

During the normal installation and deployment process for NexentaEdge, packages are downloaded from the Internet and installed on the nodes. However, in some installations, downloading packages is not possible, since the nodes may be blocked from the Internet due to security requirements.

To accommodate this kind of installation, Nexenta provides a method to install the operating system and all required packages from a repository contained in an ISO file instead of downloading them over the Internet. See [Deploying NexentaEdge Without Internet Access](#) for installation details.
NexentaEdge Installation Procedure

The following table lists the tasks you perform to install the NexentaEdge software and initialize the cluster. For more information about each task, click the link in the right column.

Table 1-1: NexentaEdge Installation Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Verify that your servers and switches meet the requirements for</td>
<td>• For server hardware requirements and supported operating systems, see <a href="#">Data and Gateway Node Requirements</a>.</td>
</tr>
<tr>
<td>NexentaEdge.</td>
<td>• For switch requirements, see <a href="#">Networking Requirements</a>.</td>
</tr>
<tr>
<td></td>
<td>• See the document “Configuration Guidelines for NexentaEdge” for specific hardware recommendations for various types of NexentaEdge deployments.</td>
</tr>
<tr>
<td>2. Determine how many gateway and data nodes your NexentaEdge cluster</td>
<td>Planning Server Node Deployment</td>
</tr>
<tr>
<td>will have.</td>
<td></td>
</tr>
<tr>
<td>3. Configure the switches that support the NexentaEdge servers.</td>
<td>Network Switch Configuration</td>
</tr>
<tr>
<td>4. Download and extract the NexentaEdge Cluster Deployment (NEDEPLOY) and Administration (NEADM) tools to the deployment workstation.</td>
<td>Downloading the NexentaEdge Deployment and Administration Tools Extracting the NEDEPLOY and NEADM Archives</td>
</tr>
<tr>
<td>5. Run the NEDEPLOY tool to add nodes to the NexentaEdge cluster.</td>
<td>Deploying NexentaEdge</td>
</tr>
<tr>
<td>6. Configure failure domains for the data nodes in the NexentaEdge</td>
<td>Failure Domains in a NexentaEdge Cluster</td>
</tr>
<tr>
<td>cluster.</td>
<td></td>
</tr>
<tr>
<td>7. Install and activate your product license.</td>
<td>Installing the NexentaEdge License</td>
</tr>
<tr>
<td>8. Use the NEADM tool to create one or more logical clusters.</td>
<td>Creating a Logical Cluster</td>
</tr>
<tr>
<td>9. Add one or more tenants to the logical clusters.</td>
<td>Creating a Tenant</td>
</tr>
<tr>
<td>10. Configure NexentaEdge to work with storage services.</td>
<td>Configuring NexentaEdge Storage Service Groups iSCSI Deployment Example Amazon S3 Deployment Example OpenStack Swift Deployment Example NFS Share Deployment Example Integrating NexentaEdge with OpenStack Cinder</td>
</tr>
</tbody>
</table>
This chapter includes the following topics:

- Networking Requirements
- Data and Gateway Node Requirements
- Deployment Workstation Requirements

Networking Requirements

This section lists the requirements for the Replicast network, which connects the data and gateway nodes in the NexentaEdge cluster.

Switch Hardware Requirements

In general, the network switches that you use for the Replicast network must meet the following requirements:

- 10 Gigabit Ethernet non-blocking enterprise-class switch.
- Jumbo frame support (9236 MTU).
- 802.3 Flow Control must be enabled for all Replicast ports (switch and host).

Replicast Network Requirements

- The Replicast network must be on a separate untagged VLAN, or be a physically separate network
- The Replicast network must be a single IPv6 subnet that has no ingress or egress routes
- The Replicast network must support jumbo frames (9236 MTU)

Data and Gateway Node Requirements

The data and gateway nodes must meet the hardware and software requirements described in this section.
Hardware Requirements

- CPU requirements
  - x86-64bit architecture with a CPU that supports AVX and SSE4.2 (full SIMD instruction set extension). These are standard on modern Xeon CPU architectures.

- Network interface requirements
  Each node requires multiple network interfaces:
  - Management network interface – IPv4, static or DHCP-assigned address
  - Replicast network interface – IPv6 unconfigured; jumbo frame support (can be configured by setting up IPv4 with 9000 MTU)
  - Client access network interface – Each gateway node must have network connectivity to the clients that will use it for data storage (for example, as an iSCSI target).
  - 802.3 Flow Control must be enabled for all Replicast ports (switch and host)
  
  During NexentaEdge deployment, at least one interface on each node must have Internet access, so that required components can be downloaded. If this is not possible (for security reasons, for example), see Deploying NexentaEdge Without Internet Access.

- Memory per node: Nexenta recommends at least 64GB RAM per data node, and 64GB per gateway node. If a node will function both as a data node and gateway node, Nexenta recommends at least 128GB RAM.

- To store the NexentaEdge core files, the size of the root partition for each node should be twice the amount of installed RAM, or you can mount /opt/nedge/var/cores to a partition or NFS share at least this size.

- All attached SATA/SAS disks or SSDs (other than the Linux operating system disk) must be fully allocated for the CCOW storage system.

- Each gateway node must have a dedicated 10 Gigabit Ethernet port for gateway services, in addition to a dedicated interface port for the CCOW storage system.

- The minimum number of nodes for a NexentaEdge deployment are four data nodes and one gateway node. Each data node must have minimum of four storage devices.

- For hybrid (SSD and HDD) deployments, all disks on a given storage node must have the same logical sector size.

Software Requirements

Before being deployed in a NexentaEdge cluster, the data and gateway nodes must be pre-provisioned with an operating system.

- The following Linux distributions have been tested to work with NexentaEdge:
  - Ubuntu 16.04 LTS, 16.04.4 LTS
  - RedHat Enterprise Linux 7.4 with Enterprise Linux Repository (ELRepo) LT kernel only
  - Community Enterprise Linux (CentOS) 7.4 with ELRepo LT kernel only
• All data and gateway nodes must be configured with a management IP address and SSH access. The `/etc/sudoers` on each node must allow all root privileges to the user ID that will be deploying NexentaEdge.

• It is important that the time be consistent on all the nodes in the cluster. Nexenta recommends that you configure each node to synchronize with a Network Time Protocol (NTP) server.

Deployment Workstation Requirements

The deployment workstation is the system from which you deploy the components of NexentaEdge to the data and gateway nodes. It must have one of the following operating systems installed:

• Ubuntu 16.04 LTS
• CentOS 7.4
• Red Hat Enterprise Linux 7.4

The user ID with which you deploy NexentaEdge must have password-less SSH access to all of the nodes to be added to the cluster, as well as all root privileges set in `/etc/sudoers` on each node.

Requirements for the Chef Environment

The deployment workstation makes use of the Chef environment (`www.getchef.com`) to install the NexentaEdge software. You deploy NexentaEdge using a Chef Solo instance packaged with the NexentaEdge software. A Chef Solo deployment does not require uploading cookbooks to a Chef server.
Installing NexentaEdge

Before You Begin

Before you start to deploy NexentaEdge, verify that your environment meets the hardware and software requirements described in the Prerequisites chapter.

Planning Server Node Deployment

Use the following guidelines when planning deployment of the server nodes in the NexentaEdge cluster:

• Determine the number of gateway and data nodes to be deployed.

  Plan to create at least four data nodes and one gateway node. You can increase the number of nodes as needed, up to a maximum of 2,000. In general, each physical server is configured as a single data or gateway node, although a given server can be configured with both data and gateway roles.

  It is possible to configure virtual machines as data or gateway nodes, but this is not recommended for a production deployment.

• Determine the roles of the nodes: data node, gateway node, or management controller.

  You must have at least one gateway node and one management controller, and at least four data nodes in total. A given server can combine all of these roles.
• Verify that each node in the cluster has access to the Internet.

A server node may require Internet access for cluster provisioning using Chef. You may need to use a proxy if some or all nodes do not have Internet access, or you can grant the nodes access to specific URLs that are required for the deployment.

In the current release, these URLs include http://www.chef.io and https://prodpkg.nexenta.com, and http://docker.com.

If it is not possible for the nodes to have Internet access (for security reasons, for example), see Deploying NexentaEdge Without Internet Access.

• For each node, allocate storage devices for the NexentaEdge cluster.

By default, during server node deployment, NexentaEdge allocates all raw (unpartitioned) devices visible on the server. Note that if any data exists on these devices, it will be lost during the deployment process.

• On each node, stop and disable the firewall, using the commands `systemctl stop firewalld`, `systemctl disable firewalld`, and `systemctl status firewalld` commands to do so. Restart Docker after disabling the firewall.

• For larger deployments (larger than 132 TB) the nodes must be deployed using Docker containers (that is, with the NEDEPLOY --data-container option). In this kind of deployment, Docker containers are allocated in maximum blocks of 132 TB, depending on the size of the drives on the node.

For example, if a node has 36 drives of 8 TB, the deployment would be 3 Docker containers: 2 containers with 16 disks (128 GB) and 1 container with 4 disks (24 GB).

Network Switch Configuration

Prior to configuring a NexentaEdge cluster, you need to configure the network switches that support the NexentaEdge servers. A node can be deployed after its switch port has been configured. Nexenta recommends that you add the nodes to the cluster only after all the switch port ranges have been configured.

The Replicast network, which connects the nodes in the cluster, must be a separate VLAN with no ingress or egress routes. It is also possible to set up multiple IPv6 subnets that only route IPv6 traffic between themselves. Consult with Nexenta if your deployment requires the use of IPv6 routers.

Failure Domains in a NexentaEdge Cluster

In the context of a NexentaEdge cluster, a failure domain refers to a group of devices or servers that for failure considerations can be treated as a single domain. For example, a group of servers in a rack may receive power from a single source. If that power source should fail, all of the servers in the rack will lose power and fail. In this example, the rack is the failure domain.

When NexentaEdge replicates data across the nodes in the cluster, it makes three copies of each chunk and stores each copy in a different location. To ensure constant availability of the data, each replicated chunk should be stored in a location in a different failure domain. This ensures that if the devices or servers in one failure domain fail, the data is still accessible from a node in one of the other failure domains.
In a NexentaEdge cluster, the failure domain can be at the Disk level, Server level, or Zone level. Each type of failure domain is explained below.

**Disk-Level Failure Domain**

*Figure 3-1* shows a single-node NexentaEdge cluster that uses a Disk-level failure domain.

*Figure 3-1: Disk-Level Failure Domain*

In this example, each chunk is replicated across three physical disks on the data node. Each disk is a failure domain. If one of the disks should fail, the copies of the chunk are still available from the other two disks.

Since this is a single-node cluster, if the data node itself should fail, then all three of the disks and therefore the data replicated across the disks would be unavailable. In addition, to upgrade the Nexenta software, you would need to shut down NexentaEdge services on the data node, disrupting client access to the data; see **Upgrading NexentaEdge to the Latest Release**.

**Server-Level Failure Domain**

*Figure 3-2* shows a three-node NexentaEdge cluster that uses a Server-level failure domain.
In this example, each chunk is replicated on a disk on one of the three physical servers configured as a data node. Each server is a failure domain. If one of the servers should fail, the copies of the chunk are still available from the other two servers.

To upgrade the Nexenta software to a later version, you can either shut down NexentaEdge services on all three data nodes and upgrade them (disruptive upgrade), or you can shut down NexentaEdge services on one data node and upgrade it, leaving the remaining two data nodes to provide service to clients (non-disruptive upgrade). See Upgrading NexentaEdge to the Latest Release.

**Zone-Level Failure Domain**

*Figure 3-3* shows a NexentaEdge cluster that uses a Zone-level failure domain.
In this example, each chunk is replicated on a disk on a data node located in a specified zone. A zone can be made up of a group of servers in a rack that receive power from a single source. If that power source should fail, all of the servers in the zone will lose power and fail. Assuming the servers in the other two zones are not powered from the same source that failed, the data is still available from servers in the remaining zones.

To upgrade the Nexenta software to a later version, you can either shut down NexentaEdge services on all the data nodes in all zones and upgrade them (disruptive upgrade), or you can shut down NexentaEdge services on one data node and upgrade it, leaving the remaining data nodes to provide service to clients (non-disruptive upgrade). See Upgrading NexentaEdge to the Latest Release.

By default, if you do not assign the data nodes to zones (as in Figure 3-1 and Figure 3-2), all of the nodes are considered to be part of a single zone, zone 0. NexentaEdge distributes the chunks across the nodes in the cluster without regard to zone assignment.

When data nodes are deployed in multiple Docker containers on a single physical server, all of the data nodes on that server are part of the same zone to which the server is assigned.

If you configure a Zone-level failure domain, it requires a minimum of three zones, each consisting of one or more data nodes. You can assign data nodes to zones either at initial NexentaEdge deployment, using the NEDEPLOY tool, or you assign data nodes to zones after deployment using the NEADM tool. In either case, if you plan to use zones, it is important that you do the configuration prior to system initialization.
Downloading the NexentaEdge Deployment and Administration Tools

To install and configure NexentaEdge, you download two software tools: NEDEPLOY and NEADM. The NexentaEdge Cluster Deployment (NEDEPLOY) tool deploys NexentaEdge to the nodes in the cluster. The NexentaEdge Administration (NEADM) tool allows you to administer the NexentaEdge cluster and obtain information about its performance.

You download and run the NEDEPLOY and NEADM tools on the system you designated as the NexentaEdge deployment workstation.

The user ID with which you deploy NexentaEdge must have password-less SSH access to all of the nodes to be added to the cluster, as well as all root privileges set in /etc/sudoers on each node.

Nexenta provides the NEDEPLOY and NEADM tools for the following operating systems (x64 bit only):

- Ubuntu Linux LTS 16.04
- Red Hat Enterprise Linux 7.4
- CentOS 7.4

**Note:** The example commands in this guide use Ubuntu Linux LTS 16.04.

Download the NEDEPLOY and NEADM archives appropriate for your deployment workstation:

- nedeploy-ubuntu_linux_latest_x64.tar.gz and neadm-ubuntu_linux_latest_x64.tar.gz (Ubuntu Linux)
- nedeploy-rhel_linux_latest_x64.tar.gz and neadm-rhel_linux_latest_x64.tar.gz (Red Hat Enterprise Linux)

Extracting the NEDEPLOY and NEADM Archives

After you download the NEDEPLOY and NEADM tools to your deployment workstation, extract the archives to any directory.

- To extract the NEDEPLOY and NEADM archives (using Ubuntu Linux):
  1. Log in to the deployment workstation.
  2. Extract the NEDEPLOY and NEADM archive files to the home directory:
     Example:
     ```
     $ cd ~
     $ tar -xzf <download-directory>/nedeploy-ubuntu_linux_latest_x64.tar.gz
     $ tar -xzf <download-directory>/neadm-ubuntu_linux_latest_x64.tar.gz
     ```
  3. Set the PATH environment variable to use the directories where you extracted the archives.
     Example:
     ```
     $ PATH=$PATH:~/nedeploy
     ```
$ PATH=$PATH:~/neadm

4. Verify that the NEDEPLOY and NEADM tools are now operational.

   Example:

   $ nede
   NexentaEdge Deployment Tool
   (additional help text follows)
   $ neadm
   NexentaEdge Cluster Administration Tool
   (additional help text follows)

Preferable Storage Devices for NexentaEdge Deployment

During deployment, NexentaEdge allocates all raw (unpartitioned) devices visible on the node to
NexentaEdge cluster storage. Note that if any data exists on these devices, it will be lost during the
deployment process.

It is possible that the node to which you are deploying NexentaEdge already has disks that are formatted and/
or mounted. In order for NexentaEdge to allocate these disks to cluster storage, the partition tables for the
disks need to be erased.

   To erase the partition table for devices to be used for NexentaEdge cluster storage:

   1. For each device to be used as NexentaEdge cluster storage, enter the following commands:

      $ dd if=/dev/zero of=<device> bs=1M count=100
      $ hdparm -z <device>

      Example:

      $ dd if=/dev/zero of=/dev/sdat bs=1M count=100
      $ hdparm -z /dev/sdat

      On CentOS systems, the hdparm utility may not be installed by default. In this case, use the
      following command to install it:

      $ yum install hdparm -y

Note: You can optionally exclude specific disks from usage in the NexentaEdge cluster. See
the descriptions of the -x and -X parameters for NEDEPLOY.

Deploying NexentaEdge

The NEDEPLOY tool performs the following configuration tasks:

- Ensuring the nodes meet the system requirements for NexentaEdge
- Adding NexentaEdge server nodes to the NexentaEdge cluster
• Allocating devices on the nodes for use as cluster storage
• Initializing the physical NexentaEdge cluster

You run NEDEPLOY from the NexentaEdge deployment workstation. If errors occur during the deployment process, you can view information about them in the 
~nedeploy/nedeploy_logs/nedge-deploy.log<timestamp> file.

Before running NEDEPLOY, verify that the network switch topology to be used in the NexentaEdge cluster is set up correctly.

RedHat/CentOS Considerations

For RedHat/CentOS installations, Nexenta recommends the following prior to running NEDEPLOY:

• Pre-configure the networking interfaces used for the Replicast network with IPv4 addresses.
• Disable the Network Manager controller.
• Reboot the node to ensure the interface name is not automatically changed.
• Make sure that the ifup and ifdown commands work correctly for specified interface names.
• The built-in firewall must be configured to allow IPv6 multicast traffic on the Replicast interface(s). This can be achieved by adding the interface(s) to the list of trusted interfaces for the firewall service, or by stopping (systemctl stop firewalld command) and disabling the firewall service (systemctl disable firewalld command).

Note that for RedHat/CentOS 7.4 installations, the ELRepo repository is required.
For nodes running RedHat/CentOS 7.4, enter the following commands to upgrade to the ELRepo kernel:

$ rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
$ yum --enablerepo=elrepo-kernel install kernel-lt kernel-lt-headers
$ reboot

Following the reboot, if the new kernel version is listed first, then use the following command to set grub to use 0, which configures the system to boot off the newer kernel. You may want to leave some of the older 3.10.x kernels on the system in case you decide to revert to the stock RedHat/CentOS 7 kernel. You can always modify the default kernel to use later on.

$ grub2-set-default 0

Using the Command Line

Use the following procedures to run NEDEPLOY from the command line. These procedures check that the nodes meet the system requirements and deploy the NexentaEdge software.

Running the Pre-Check Utility

- To verify that a node meets system requirements for NexentaEdge:

  1. Log in to the NexentaEdge deployment workstation (that is, the workstation where the NEDEPLOY tool is installed).
2. Run the pre-check utility to ensure that the node meets the requirements for being added to the cluster.

$ nedeploy precheck <ip-address> <username:password> -i <interface> [-t <profile>] [-x <disks-to-exclude>] [-X <disks-to-reserve>]

Where:

<ip-address> Is the management IP address of the node.

username Is a user account on the node that has administrative privileges; for example, root.

If you specify a different account than root, it must have password-less SSH access to the node, as well as all root privileges set in /etc/sudoers on the node.

password Is the password for this user account.

-i <interface> Is the Ethernet interface to be used for communication between nodes within the NexentaEdge cluster (that is, the Replicast network).

-t <profile> Has to do with how metadata is distributed among the hard disk drives (HDDs) and solid state drives (SSDs) on the node and whether journaling operations are enabled. You can specify one of the following profiles:

capacity NexentaEdge uses all of the available HDDs and SSDs as one large storage pool.

performance NexentaEdge offloads the majority of metadata to SSD. This is the default profile.

gateway NexentaEdge configures the node as a gateway with no disks allocated for cluster storage.

all-flash NexentaEdge optimizes the node for use in an all-flash cluster.

-x <disks-to-exclude> Specifies a comma-separated list of one or more devices that NexentaEdge will not use as cluster storage. NexentaEdge adds these disks to a list of devices that it will never allocate to cluster storage. In this list, you can specify disks that you are using for other applications besides NexentaEdge.

-X <disks-to-reserve> Specifies a comma-separated list of one or more devices that NexentaEdge will not allocate to cluster storage for this deployment. The difference between this list and the <disks-to-exclude> list is that the disks in the <disks-to-exclude> list are permanently excluded from use in the NexentaEdge cluster, while disks in the <disks-to-reserve> can be added to the cluster at a later time.

Note that the pre-check utility includes the <disks-to-reserve> when calculating the total memory required per node.
Example:

$ nedeploy precheck 10.3.32.12 root:password -i eth1 -t capacity

System response:

10.3.32.12 - Prechecking node
10.3.32.12 - Connecting
10.3.32.12 - Operating System: Ubuntu 16.04 LTS
10.3.32.12 - Maximum transmission unit (MTU): 1500
10.3.32.12 - Network Interface Speed: 10000
10.3.32.12 - Total Memory: 8175256kB
10.3.32.12 - Raw Disks:
10.3.32.12 -       HDD /dev/sdb         40G
10.3.32.12 -       HDD /dev/sdc         40G
10.3.32.12 -       HDD /dev/sdd         40G
10.3.32.12 -       HDD /dev/sde         40G
10.3.32.12 - Total Disk Space: 160GB
CPU instructions detected: fpu vme de pse tsc mtc pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 cflush dts mmx fxsr sse sse2 ss ht syscall
nx rdtscp lm constant_tsc arch_perfmon pebs bts nopl xt6ology
  tsc_reliable nonstop_tsc aperfmperf pni pclmulqdq sse3 cx16 sse4_1
  sse4_2 popcnt aes hypervisor lahf_lm epb dtherm ida arat
CPU must support AVX + SSE_4.2+
10.3.32.12 - PRE-CHECK FAILED

If the pre-check utility indicates that the node does not meet the requirements for
NexentaEdge, correct the issue if possible and re-run the utility.

Deploying NexentaEdge to the Nodes

To add nodes using NEDEPLOY from the command line:

1. From the NexentaEdge deployment workstation, use the following command to deploy the
   NexentaEdge software to the nodes:

   $ nedeploy deploy solo ({<ip-address>:<port> | <nodename>})
   <username:password> -i <interface> [-I <interface>] [-t <profile>]
   [-x <disks-to-exclude>] [-X <disks-to-reserve>] [-z <zone>]
   [-F <filesystem-type>] [-m] [--docker] [-d <TRs>]
   [-P <number-of-partitions>] [--upgrade] [--wipeout-datastores]

   Where:

   <ip-address>
   Is the management IP address of the node. You can optionally
   specify the SSH port if it is not the default of 22.

   <nodename>
   Is the name of the node to add to the cluster. This will be the
   name recorded in the Chef database for the node.

   You can assign any name to the node, although Nexenta
   recommends that you use the host name of the node.
**username**

Is a user account on the node that has administrative privileges; for example, root.

If you specify a different account than root, it must have password-less SSH access to the node, as well as all root privileges set in `/etc/sudoers` on the node.

**password**

Is the password for this user account.

- **-i <interface>**

  Is the Ethernet interface to be used for communication between nodes within the NexentaEdge cluster (that is, the Replicast network).

- **-I <interface>**

  Optionally specifies the Ethernet interface to be used for the gateway. If a storage service (iSCSI, Swift, S3) is started on this node, it would use this interface. If you do not specify the gateway interface, the storage service uses the interface specified with the `-i` option.

- **-t <profile>**

  Has to do with how metadata is distributed among the hard disk drives (HDDs) and solid state drives (SSDs) on the node and whether journaling operations are enabled. You can specify one of the following profiles:

  - **capacity** NexentaEdge uses all of the available HDDs and SSDs as one large storage pool.
  - **performance** NexentaEdge offloads the majority of metadata to SSD, assuming a hybrid SSD/HDD deployment. This is the default profile.
  - **gateway** NexentaEdge configures the node as a gateway with no disks allocated for cluster storage.
  - **all-flash** NexentaEdge optimizes the node for use in an all-flash cluster.

- **-x <disks-to-exclude>**

  Specifies a comma-separated list of one or more devices that NexentaEdge will not use as cluster storage. NexentaEdge adds these disks to a list of devices that it will never allocate to cluster storage. In this list, you can specify disks that you are using for other applications besides NexentaEdge.

- **-X <disks-to-reserve>**

  Specifies a comma-separated list of one or more devices that NexentaEdge will not allocate to cluster storage for this deployment. The difference between this list and the `<disks-to-exclude>` list is that the disks in the `<disks-to-exclude>` list are permanently excluded from use in the NexentaEdge cluster, while disks in the `<disks-to-reserve>` can be added to the cluster at a later time.

  Note that the pre-check utility includes the `<disks-to-reserve>` when calculating the total memory required per node.
-z <zone> Specifies the zone to which the node belongs. All of the nodes in a given zone are considered to be part of the same failure domain; for example, a group of nodes that receive power from the same source can be placed in the same zone. NexentaEdge ensures that data in the cluster is replicated across multiple zones, so that failure of the nodes in a zone does not result in lost access to the data.

Zone settings must be applied prior to system initialization. See Zone-Level Failure Domain for more information.

-<m> Configures this node as a management controller.

-F <filesystem-type> Optionally specifies the filesystem type to use for LFS drivers. You can specify ext4 or none. The default of none causes the RD (raw disk) driver to be automatically enabled.

--docker Causes deployed services (applications and endpoints) to run inside of Docker containers. This is the default deployment method.

--data-container Deploys NexentaEdge services to the node using Docker containers. If the node has 132 TB or more of disk space, you must specify this option.

-d <TBs> Specifies the maximum capacity in TBs that can be served by an individual data container. After this number of TBs is reached, a new data container is created. Default is 132 TB.

-F <number-of-partitions> Provisions the data node with the specified <number-of-partitions> within a large mounted partition. Default is 1. This option is valid only for all-flash configurations where the ext4 filesystem type is used; that is, the –F ext4 option is included in the nedeploy command.

--upgrade Upgrades the NexentaEdge core files on the node. See Upgrading NexentaEdge to the Latest Release.

--wipeout-datastores If used in conjunction with the --upgrade option, erases all data on the disks used by the previous NexentaEdge version.

Example:
$ nedeploy deploy solo 10.3.30.32 root:password -i eth1 -t capacity
-x sdau,sdaw

2. Repeat the previous step for each node you want to add to the cluster.

Initializing the NexentaEdge Cluster

Important: If you plan to use zones, you must configure them prior to doing the procedure below. See Zone-Level Failure Domain for more information.
To initialize the NexentaEdge cluster:

1. On the NexentaEdge deployment workstation, enter the `neadm system status` command. The first time you enter this command after deployment, NexentaEdge prompts you for the URL for the NexentaEdge management controller node.

   Example:
   ```
   $ neadm system status
   Error: connect ECONNREFUSED
   Unable to reach management node: http://0.0.0.0:8080
   Please enter management node IP with NEDGE port (default 8080)
   Example: http://1.2.3.4:8080
   ? Remote API URL: http://10.3.30.32:8080
   Remote API URL successfully updated. Re-run system status again ...
   ```

2. Verify the system status:

   ```
   $ neadm system status
   System response:
   ```

   Ensure that the nodes you added to the cluster have a state of ONLINE and verify the zone assignments for the nodes. Upon initial deployment, it may take up to two minutes for NexentaEdge to get the status for the nodes. If you see an error message when you enter the `neadm system status` command, wait two minutes and enter the `neadm system status` command again.

   **Note:** Whenever you run the `neadm system status` command, NexentaEdge by default automatically performs a service checkpoint status check. Normally this has no impact on system performance, but on larger deployments, it may take some time to complete. On systems that are larger and more stable, you can disable the service checkpoint status check by changing the `ENABLE_AUTOMATED_CHECKPOINT` setting in the `.neadmrc` file on the NexentaEdge deployment workstation from 1 to 0.

3. If the servers have a status of ONLINE, initialize the cluster:

   ```
   $ neadm system init
   System response:
   ```
   ```
   enter to continue
   Do you agree to the EULA? "YES / NO" : YES
   NAME VALUE
   GenID 1491938187453656
   NumRows 16
   HashMask 0xf
   ServerCount 5
   VdevCount 25
   ```
STATUS: Cluster in consistent state
Setting up initial configuration checkpoint, please wait ...
Service checkpoint created successfully
NexentaEdge cluster initialized successfully.
System GUID: DD3EE6D7-1234-5678-9012-C1B41C7EABC8

Note the System GUID; you may need to use it when you configure your NexentaEdge license key. After initialization, you can display the System GUID using the neadm system license show command.

Installing the NexentaEdge License

To activate NexentaEdge, you must install a license for the product. NexentaEdge supports either online or offline license installation. Both of these methods use the activation key you received when you downloaded the software from Nexenta Systems.

NexentaEdge has the following types of licenses:

- Enterprise license – Enables all NexentaEdge features. Depending on the type of Enterprise license you purchase from Nexenta Systems, there may be limitations on the amount of time NexentaEdge can be used or the amount of storage capacity that can exist in the cluster.

  The following options are available for Enterprise licenses:

  - Trial – Allows full functionality for 45 days after activation. After the trial period expires, you must install a perpetual license in order to keep using the product. A trial license does not have any limitation on the amount of storage in the cluster.
  - Perpetual – Allows full functionality and does not expire. For a perpetual license, you can specify the amount of storage capacity that can be added to the cluster.

If you move from a trial license to a perpetual license, make sure the perpetual license allows at least as much storage capacity in the cluster as you have already deployed with your trial license.

Online License Installation

Use online license installation if the deployment workstation has access to the Internet. You enter the activation key with an NEADM command. NEADM contacts the Nexenta license server, which uses the activation key to generate your license. NEADM downloads your license and installs it in the system.

- To install a NexentaEdge license using online license installation:
  1. Log in to the NexentaEdge deployment workstation.
  2. Use the following command to generate and install the NexentaEdge license:
     
     ```
     $ neadm system license set online <activation_key>
     
     System response:
     
     Online Activation Completed.
     ```
Offline License Installation

Use offline license installation if the deployment workstation does not have access to the Internet. You submit the activation key and System GUID to Nexenta, either by using the Support Portal, or contacting your Nexenta representative. Nexenta sends you a license, which you save as a text file, load onto the deployment workstation, and install in the system using the NEADM tool.

To install a NexentaEdge license using offline license installation:

1. Log in to the NexentaEdge deployment workstation.
2. Display the System GUID.
   
   ```
   $ neadm system license show
   ```
   System response:
   
<table>
<thead>
<tr>
<th>PARAM</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation GUID:</td>
<td>DD3EE6D7-1234-5678-9012-C1B41C7EABC8</td>
</tr>
<tr>
<td>License Type:</td>
<td>ENTERPRISE-TRIAL</td>
</tr>
<tr>
<td>License Serial Number:</td>
<td>XX-XXX-XX-XXXXXXXXX</td>
</tr>
<tr>
<td>License Base Capacity:</td>
<td>0 TB, Raw Capacity</td>
</tr>
<tr>
<td>License Expiration Date</td>
<td>XX/XX/XXXX</td>
</tr>
<tr>
<td>License not activated</td>
<td></td>
</tr>
</tbody>
</table>

3. Copy the value displayed for the Installation GUID. The GUID is also displayed when you initialize the cluster with the `neadm system init` command.

4. Submit the GUID, along with the activation key, to Nexenta. Use the Support Portal or contact your Nexenta representative. Nexenta will use the GUID and activation key to generate your license and send it to you.

5. Save the license as a text file in a location where it is accessible from the deployment workstation.

6. Use the following command to install the NexentaEdge license:
   
   ```
   $ neadm system license set file <license_file_path>
   ```
   System response:
   
   License Set Successfully

Creating a Logical Cluster

A NexentaEdge deployment consists of a single physical cluster and one or more logical clusters. The physical cluster is simply the server devices and network switches used by NexentaEdge. A logical cluster is the collection of data and gateway nodes that make up the storage system. A logical cluster has one or more tenants. NexentaEdge provides services such as iSCSI block storage, OpenStack Swift, and Amazon S3 to specific tenants within a logical cluster.

While NexentaEdge supports an unlimited number of logical clusters, a typical deployment has one logical cluster.
To create a NexentaEdge cluster:

1. Log in to the deployment workstation.
2. Create a cluster:
   
   ```bash
   neadm cluster create <cluster_name>
   ```
   
   Example:
   
   ```bash
   neadm cluster create clu1
   ```
   
   System response:
   
   Cluster clu1 created successfully.

3. Proceed to **Creating a Tenant**

Creating a Tenant

A tenant is a group of users, or an account, that shares resources such as containers or virtual machines. Typically, a tenant is also the entity that is billed for storage services.

You must have at least one cluster created before you can create a tenant. After creating the tenant, you can create buckets and specify the storage services that NexentaEdge provides for this tenant.

To create a tenant:

1. Log in to the deployment workstation.
2. Create a tenant:
   
   ```bash
   neadm tenant create <cluster-name>/<tenant-name>
   ```
   
   Example:
   
   ```bash
   neadm tenant create clu1/ten1
   ```
   
   System response:
   
   Tenant ten1 created successfully.

3. Proceed to **Creating a Bucket**

Creating a Bucket

A bucket is a container for objects. For example, if you deploy an iSCSI storage service group, you can create a bucket and add LUNs to it.

Buckets are created on a per-tenant basis; to create a bucket, you must have first created a cluster and the tenant to which you want to assign the bucket.

Note that OpenStack Swift and Amazon S3 services cannot share the same namespace, so you cannot use the same bucket names if you are using both services.
To create a bucket:

1. Log in to the deployment workstation.
2. Create a bucket:
   
   ```
   $ neadm bucket create <cluster-name>/<tenant-name>/<bucket-name>
   ```
   
   Example:
   
   ```
   $ neadm bucket create clu1/ten1/buk1
   ```
   
   System response:
   
   Bucket buk1 created successfully.
3. Proceed to Configuring NexentaEdge Storage Service Groups
Configuring NexentaEdge Storage Service Groups

This chapter includes the following topics:

- About NexentaEdge Storage Service Groups
- Before You Begin
- iSCSI Deployment Example
- Amazon S3 Deployment Example
- OpenStack Swift Deployment Example
- NFS Share Deployment Example

About NexentaEdge Storage Service Groups

NexentaEdge enables you to easily provision one or multiple enterprise-class object and block storage services. External clients access data stored in the NexentaEdge cluster via gateway nodes using protocols of the storage services NexentaEdge supports.

NexentaEdge includes support for the following:

- Object protocols: OpenStack Swift, Amazon S3
- Block protocols: iSCSI, OpenStack Cinder, Network Block Device (NBD)
- File protocol: NFS

To configure a NexentaEdge cluster, you indicate which storage service(s) you want to provide for a given tenant, then specify the servers that should serve as the gateway nodes for that storage service. The data and gateway nodes that provide the storage service are known as a storage service group.

To complete the installation, you may need to perform some additional configuration tasks that are specific to the storage services to which you are connecting the NexentaEdge cluster. For example, if you are configuring an iSCSI storage service group, you can create LUNs that are made available to iSCSI initiators via the iSCSI target enabled by the iSCSI storage service group. These tasks are covered in the NexentaEdge User Guide.

Before You Begin

Before attempting the procedures in this chapter, make sure you have already deployed a NexentaEdge cluster with at least one tenant.
iSCSI Deployment Example

The following example describes how to configure an iSCSI storage service group with one iSCSI LUN.

For the example, the following naming conventions are used:

- **Cluster Name**: clu1
- **Tenant**: ten1
- **Bucket**: buk1
- **iSCSI storage service group name**: isc01
- **LUN**: LUN01
- **Gateway node**: D586BF84009C4230BFE31CAC961197D

1. **To configure an iSCSI storage service group:**
   - Add an iSCSI storage service group to the cluster:
     
     ```
     $ neadm service create iscsi isc01
     ```
     
     System response:
     
     ```
     Service isc01 created
     ```

2. **Verify the server ID that you want to add to the cluster:**
   
   ```
   $ neadm system status
   ```
   
   System response:

<table>
<thead>
<tr>
<th>ZONE:HOST:CID</th>
<th>SID</th>
<th>UTIL</th>
<th>CAP</th>
<th>CPU</th>
<th>MEM</th>
<th>DEVS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:node32:</td>
<td>B5DEF7B9B3C690...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.2@2.34Ghz</td>
<td>4.16G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node33:</td>
<td>[M] B3EF9C3OB6F79F...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.18G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node34:</td>
<td>15581CC9F314A4E...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.18G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node35:</td>
<td>9BB624B94C8E2C...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.15G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node36:</td>
<td>F88C6C3180C34E...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.15G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
</tbody>
</table>

3. **Copy the server ID.**

4. **Associate the server ID with the iSCSI service group:**

   ```
   $ neadm service add isc01 node32
   ```
   
   System response:
   
   ```
   Service isc01 added to data02-ab
   ```

   If you are deploying the iSCSI storage service group in a high-availability (HA) configuration, you can specify two server nodes to be gateway nodes by configuring a virtual IP address (VIP) for the storage service group; otherwise only one node can be a gateway node. See the *NexentaEdge User’s Guide* for more information.

5. **For Docker deployments, you must configure the Docker network that links the containers to the node on which they are contained.**
The following commands show an example of how to configure a Docker network named client-net using the Macvlan network driver. You can use other network drivers if necessary.

```
$ ifconfig enp2s0f1 192.168.10.201/24 up
$ modprobe macvlan
$ docker network create -d macvlan --subnet 192.168.10.0/24 --gateway 192.168.10.1 -o parent=enp2s0f1 client-net
$ docker run --net=client-net --ip=192.168.10.11 -it alpine /bin/sh
```

6. For Docker deployments, list the available Docker networks and determine which one you want to attach to the iSCSI storage service group.

Example:

```
$ docker network ls
```

System response:

```
<table>
<thead>
<tr>
<th>NETWORK ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>7fca4eb8c647</td>
<td>client-net</td>
</tr>
<tr>
<td>bridge</td>
<td></td>
</tr>
<tr>
<td>9f904ee27bf5</td>
<td>none</td>
</tr>
<tr>
<td>null</td>
<td></td>
</tr>
<tr>
<td>cf03ee007fb4</td>
<td>host</td>
</tr>
<tr>
<td>host</td>
<td></td>
</tr>
<tr>
<td>78b03ee04f4c4</td>
<td>multi-host</td>
</tr>
<tr>
<td>overlay</td>
<td></td>
</tr>
</tbody>
</table>
```

7. For Docker deployments, determine subnet information for the selected Docker network.

Example:

```
$ docker network inspect client-net
```

System response:

```
[
 {
   "Name": "client-net",
   "Id": "b2b1a2cba71761d988383fd68218cf70bbbd17d3284896085f7c98b0f",
   "Scope": "local",
   "Driver": "bridge",
   "IPAM": {
      "Driver": "default",
      "Config": [
         {
            "Subnet": "192.168.200.1/16",
            "Gateway": "192.168.200.21"
         }
      ]
   }
 },
```

8. For Docker deployments, display the ID of the gateway node configured for the storage service group.

Example:

```
$ neadm service show isc01
```
9. Using the ID from the previous step, add the Docker network to the gateway node. Note the use of quotes between the network name and IP address.

Example:

```
$ neadm service configure isc01 X-Container-Network-D586BF84009C4230BFEE31CAC961197D "client-net --ip 192.168.200.1"
```

To specify multiple Docker networks, separate them with semicolons; for example:

```
$ neadm service configure isc01 X-Container-Network-D586BF84009C4230BFEE31CAC961197D "client-net --ip 192.168.200.1;local-net --ip 192.168.1.23"
```

Note that the client network IP address ranges must be on the same subnet as the physical client interfaces.

10. Apply the iSCSI storage service group to the cluster/tenant:

```
$ neadm service serve isc01 clu1/ten1
```

System response:

```
Success
```

11. Enable the iSCSI storage service group:

```
$ neadm service enable isc01
```

System response:

```
Service isc01 enabled
```

12. Verify that the iSCSI storage service group is enabled.

```
$ neadm service list
```
13. Add an iSCSI LUN:


Where:

- `<service-group>` is the iSCSI storage service group for which the LUN is being created.
- `<lun-path>` associates the LUN with a specific cluster/tenant/bucket combination for this iSCSI storage service group.
- `<size>` is the size of the new LUN. To specify a size in gigabytes, use the `GB` suffix (for example, `2GB` to specify 2 gigabytes). If you do not specify a suffix, a size in bytes is assumed. Maximum LUN size is 16 petabytes.
- `-b <block-size>` is the block size for the new LUN. The default is 4096 bytes. You can change this in increments of 512 bytes, with a minimum block size of 512 bytes. For ESX, this should be set to 512 bytes.
- `-s <chunk-size>` sets the size of the chunks in the object that backs the LUN; NexentaEdge breaks objects stored for this LUN into chunks of this size.
  
  You can specify the chunk size in $2^n$-byte increments; for example, 8192, 16384 and so on. The minimum chunk size is 8,192 bytes, and the maximum is 1,048,576 bytes.
- `-n <lun-number>` is the LUN number for the new LUN. You can specify a number from 1–255.
- `-o <option>` specifies CSV input of metadata.
- `-v <vip-address>/<netmask>` is the VIP the new LUN will be associated with. See the NexentaEdge User Guide.
- `-r <replication-count>` sets the number of times the LUN object is replicated across the cluster. This can be set to 2, 3, or 4. The default replication count for a LUN object is 3.
- `-t <IOPS-limit>` sets a limit in 4K normalized IOPS for the LUN object.

For the `<size>`, `<block-size>`, and `<chunk-size>` parameters, you can optionally specify a suffix to indicate the units (B, KB, MB, GB, or TB). For example, to specify a size in gigabytes, use the `GB` suffix. If you do not specify a suffix, a size in bytes is assumed.

Example:

$ neadm iscsi create isc01 clu1/ten1/buk1/LUN01 1G

System response:

iSCSI LUN clu1/ten1/buk1/LUN01 created.
14. Log in to your iSCSI initiator and discover the iSCSI target that you added to the service group.

Additional iSCSI Management Commands

- **To view the list of iSCSI LUNs:**
  
  ```bash
  $ neadm iscsi list isc01
  ```
  
  System response:
  
<table>
<thead>
<tr>
<th>LUN HOST</th>
<th>SIZE</th>
<th>BLOCK</th>
<th>CHUNK</th>
<th>REPCOUNT</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>newnode101</td>
<td>1G</td>
<td>4K</td>
<td>32K</td>
<td>3</td>
<td>clu1/ten1/buk1/LUN01</td>
</tr>
</tbody>
</table>

- **To create a snapshot of a LUN:**
  
  ```bash
  $ neadm iscsi snapshot create isc01 clu1/ten1/buk1/LUN01@snap01
  ```
  
  System response:
  
  Snapshot snap01 created.

- **To view the list of LUN snapshots:**
  
  ```bash
  $ neadm iscsi snapshot list isc01 clu1/ten1/buk1/LUN01
  ```
  
  System response:
  
  SNAPSHOTS:
  
  snap01

- **To create a clone of the snapshot:**
  
  ```bash
  $ neadm iscsi snapshot clone isc01 clu1/ten1/buk1/LUN01@snap01 clu1/ten1/buk1/LUN02
  ```
  
  System response:
  
  Snapshot snap01 cloned into LU clu1/ten1/buk1/LUN02.

**Note:** Specify a new LUN to clone the snapshot. If you specify an existing LUN, the procedure fails.

- **To roll back a LUN to a snapshot version:**
  
  ```bash
  $ neadm iscsi snapshot rollback isc01 clu1/ten1/buk1/LUN01@snap01
  ```
  
  System response:
  
  iSCSI LU clu1/ten1/buk1/LUN01 rolled back to Snapshot: snap01

- **To delete a snapshot:**
  
  ```bash
  $ neadm iscsi snapshot delete isc01 clu1/ten1/buk1/LUN01@snap01
  ```
  
  System response:
  
  Snapshot snap01 deleted.

See “Configuring an iSCSI Storage Service” in the *NexentaEdge User Guide* for more information.
Integration with OpenStack Cinder

A NexentaEdge cluster can provide block storage functions to OpenStack Cinder. OpenStack Cinder includes an API that can manage a block storage backend. You can configure this storage backend to be a NexentaEdge iSCSI storage service group.

To do this, you set up an iSCSI storage service group, then install and enable the NexentaEdge Cinder plugin, which configures OpenStack Cinder to use the iSCSI service group for storage. See Integrating NexentaEdge with OpenStack Cinder for configuration details.

Amazon S3 Deployment Example

The following example describes how to configure an Amazon S3 storage service group.

For the example, the following naming conventions are used:

<table>
<thead>
<tr>
<th>Cluster Name</th>
<th>clul</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>ten1</td>
</tr>
<tr>
<td>Amazon S3 storage service group name</td>
<td>s301</td>
</tr>
<tr>
<td>Server ID</td>
<td>B5DEF7B9B3C69055A9079125BC24FD29</td>
</tr>
</tbody>
</table>

1. **To configure an Amazon S3 storage service group:**
   1. Add an Amazon S3 storage service group to the cluster:
      ```
      $ neadm service create s3 s301
      System response:
      Service s301 created
      ```
   2. Verify the server ID that you want to add to the cluster:
      ```
      $ neadm system status
      System response:
      ZONE:HOST:CID        SID              UTIL   CAP  CPU           MEM        DEVs STATE
      0:node32:            B5DEF7B9B3C690... 0%   160G 2/0.2@2.34Ghz 4.16G/7.8G 4/4  ONLINE
      0:node33:            [M] B3EF9C30B6F79F... 0%   160G 2/0.1@2.34Ghz 4.18G/7.8G 4/4  ONLINE
      0:node34:            155B81C291FA4E... 0%   160G 2/0.1@2.34Ghz 4.18G/7.8G 4/4  ONLINE
      0:node35:            9BB624B94C8E2C... 0%   160G 2/0.1@2.34Ghz 4.15G/7.8G 4/4  ONLINE
      0:node36:            F88C6C3180C34E... 0%   160G 2/0.1@2.34Ghz 4.15G/7.8G 4/4  ONLINE
      ```
      3. Copy the server ID.
      4. Associate the server ID with the Amazon S3 storage service group:
      ```
      Example:
      $ neadm service add s301 node32
5. For Docker deployments, you must configure the Docker network that links the containers to the node on which they are contained.

The following commands show an example of how to configure a Docker network named `client-net` using the Macvlan network driver. You can use other network drivers if necessary.

```plaintext
$ ifconfig enp2s0f1 192.168.10.201/24 up
$ modprobe macvlan
$ docker network create -d macvlan --subnet 192.168.10.0/24 --gateway 192.168.10.1 -o parent=enp2s0f1 client-net
$ docker run --net=client-net --ip=192.168.10.11 -it alpine /bin/sh
```

6. For Docker deployments, list the available Docker networks and determine which one you want to attach to the NFS storage service group. In the example below, the network `client-net` is used.

Example:

```plaintext
$ docker network ls
```

System response:

```
<table>
<thead>
<tr>
<th>NETWORK ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>7fca4eb8c647</td>
<td>client-net</td>
</tr>
<tr>
<td>bridge</td>
<td>none</td>
</tr>
<tr>
<td>null</td>
<td>host</td>
</tr>
<tr>
<td>cf03ee007fb4</td>
<td>multi-host</td>
</tr>
<tr>
<td>overlay</td>
<td></td>
</tr>
</tbody>
</table>
```

7. For Docker deployments, determine subnet information for the selected Docker network. In the example below, the network `client-net` is used.

Example:

```plaintext
$ docker network inspect client-net
```

System response:

```
{
  "Name": "client-net",
  "Id": "b2b1a2cba717161d984383fd68218cf70bbbd17d328496885f7c98b0f",
  "Scope": "local",
  "Driver": "bridge",
  "IPAM": {
    "Driver": "default",
    "Config": [
      {
        "Subnet": "192.168.200.1/16",
        "Gateway": "192.168.200.21"
      }
    ]
  }
}
```
8. For Docker deployments, display the IDs of the gateway node configured for the storage service group.

Example:

```bash
$ neadm service show s301
```

System response:

```
FIELD                VALUE
X-Service-Name       s301
X-Service-Type       s3
X-Service-Worker     ccowgws3
X-Number-Of-Versions 1
X-Auth-Type          disabled
X-Description        S3 Object
X-Servers            B5DEF7B9B3C69055A9079125BC24FD29
X-Status             enabled
```

Service Group Objects:
```
[
]
```

9. Using the ID from the previous step, add the Docker network to the gateway node.

Example:

```bash
$ neadm service configure s301 X-Container-Network-B5DEF7B9B3C69055A9079125BC24FD29 "client-net --ip 192.168.200.1"
```

To specify multiple Docker networks, enclose them in quotes and separate them with semicolons; for example:

```bash
$ neadm service configure s301 X-Container-Network-B5DEF7B9B3C69055A9079125BC24FD29 "client-net --ip 192.168.200.1;local-net --ip 192.168.1.23"
```

10. Apply the Amazon S3 storage service group to the cluster/tenant:

```bash
$ neadm service serve s301 clu1/ten1
```

System response:
```
Service s301 now serving path clu1/ten1
```

The combination of cluster and tenant is known as a logical path.

11. Enable the Amazon S3 storage service group:

```bash
$ neadm service enable s301
```

System response:
```
Service s301 enabled
```

After the Amazon S3 storage service group is enabled, you can use the IP address of the gateway node (server ID B5DEF7B9B3C69055A9079125BC24FD29) for Amazon S3 API communication.

12. Verify that the Amazon S3 storage service group is enabled.

```bash
$ neadm service list
```
Additional Configuration

After configuring an Amazon S3 storage service group, you can perform the following optional configuration tasks:

- Configuring authentication for the Amazon S3 storage service group
- Changing the ports for Amazon S3 API calls
- Changing the region setting used for Amazon S3 API calls
- Configuring the Amazon S3 gateway to provide MD5 or vendor-defined checksum values for objects
- Enabling versioning support on the Amazon S3 gateway
- Configuring Amazon S3 multipart upload support

These tasks are covered in the “Configuring an Amazon S3 Storage Service Group” chapter of the NexentaEdge User Guide.

OpenStack Swift Deployment Example

The following example describes how to configure an OpenStack Swift storage service group.

For the example, the following naming conventions are used:

Cluster Name clu1
Tenant AUTH_1234
OpenStack Swift storage service group name sw01
Server ID D586BF84009C4230BFEE31CAC961197D

To configure an OpenStack Swift storage service group:

1. Add an OpenStack Swift storage service group to the cluster:
   $ neadm service create swift sw01
   System response:
   Service sw01 created

2. Verify the server ID that you want to add to the cluster:
   $ neadm system status
   System response:
   ZONE:HOST:CID SID UTIL CAP CPU MEM DEVs STATE
3. Copy the server ID.

4. Associate the server ID with the OpenStack Swift storage service group:

   $ neadm service add sw01 node32

   System response:
   Service sw01 added to B5DEF7B9B3C69055A9079125BC24FD29

5. Apply the OpenStack Swift storage service group to the cluster:

   $ neadm service serve sw01 clu1

   System response:
   Service sw01 now serving path clu1

6. Enable the OpenStack Swift storage service group:

   $ neadm service enable sw01

   System response:
   Service sw01 enabled

7. Verify that the OpenStack Swift storage service group is enabled.

   $ neadm service list

   System response:
   TYPE  NAME  SERVERID                          STATUS
   swift sw01  B5DEF7B9B3C69055A9079125BC24FD29  enabled

After creating an OpenStack Swift storage service group, use the gateway IP with the port number for OpenStack Swift API.

After creating an OpenStack Swift storage service group, you can use the IP address of the gateway node (server ID BC2C6400B2C5D2C69CE0A) for OpenStack Swift API communication.

Additional Configuration

After configuring an OpenStack Swift storage service group, you can perform the following optional configuration tasks:

• Configuring authentication for the OpenStack Swift storage service group
• Changing the ports for OpenStack Swift API calls
• Configuring the OpenStack Swift gateway to provide MD5 or vendor-defined checksum values for objects
• Enabling versioning support on the OpenStack Swift gateway
• Configuring OpenStack Swift large objects support

These tasks are covered in the “Configuring an OpenStack Swift Storage Service Group” chapter of the *NexentaEdge User Guide*.

**NFS Share Deployment Example**

The following example describes how to configure an NFS storage service group and share an NFS filesystem.

For the example, the following naming conventions are used:

- Cluster Name: clu1
- Tenant: ten1
- Bucket (share) name: buk1
- NFS service group name: nfs01
- Gateway node: B8E8D8A090CB2CB2EF862184A02841D4

*To configure an NFS storage service group:*

1. Add an NFS storage service group to the cluster:

   ```bash
   $ neadm service create nfs nfs01
   ```

   System response:
   
   Service nfs created

2. Verify the server ID that you want to add to the cluster:

   ```bash
   $ neadm system status
   ```

   System response:

<table>
<thead>
<tr>
<th>ZONE:HOST:CID</th>
<th>SID</th>
<th>UTIL</th>
<th>CAP</th>
<th>CPU</th>
<th>MEM</th>
<th>DEVs</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:node32:</td>
<td>B5DEF7B9B3C690...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.2@2.34Ghz</td>
<td>4.16G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node33:</td>
<td>[M] B3EF9C30B6F79F...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.18G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node34:</td>
<td>155B81C291FA4E...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.18G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node35:</td>
<td>9BB624B94C8E2C...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.15G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
<tr>
<td>0:node36:</td>
<td>F88C6C3180C34E...</td>
<td>0%</td>
<td>160G</td>
<td>2/0.1@2.34Ghz</td>
<td>4.15G/7.8G</td>
<td>4/4</td>
<td>ONLINE</td>
</tr>
</tbody>
</table>

3. Copy the server ID.

4. Associate the server ID with the NFS storage service group:

   ```bash
   $ neadm service add nfs01 node34
   ```

   System response:

   Service nfs01 added to 155B81C291FA4E2F8E5B5ED3C1017956

5. Apply the NFS storage service group to the cluster:

   ```bash
   $ neadm service serve nfs01 clu1
   ```
System response:
Service nfs01 now serving path clu1

6. Enable the NFS storage service group:
   
   $ neadm service enable nfs01
   System response:
   Service nfs01 enabled

7. Verify that the NFS storage service group is enabled.
   
   $ neadm service list
   System response:
   
<table>
<thead>
<tr>
<th>TYPE</th>
<th>NAME</th>
<th>SERVERID</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfs</td>
<td>nfs01</td>
<td>155B81C291FA4E2F8E5B5ED3C1017956</td>
<td>enabled</td>
</tr>
</tbody>
</table>

8. Use the following command to export the bucket as an NFS share. The bucket name is used as the name of the NFS share; for example, bucket clu1/ten1/buk1 is exported as hostname:/buk1.
   
   $ neadm nfs share nfs01 clu1/ten1/buk1
   System response:
   Service nfs01 updated successfully.

9. Use the following command to display the status of the servers in the NFS storage service group.
   
   $ neadm nfs status nfs01
   System response:
   
<table>
<thead>
<tr>
<th>Server</th>
<th>Id</th>
<th>Path</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>B8E8D8A090CB2CB2EF862184A02841D4</td>
<td>0</td>
<td>/</td>
<td>PSEUDO</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>/buk1</td>
<td>NEDGE</td>
</tr>
</tbody>
</table>
This chapter includes the following topics:

- Overview
- Setting Up an iSCSI Storage Service for Cinder
- Enabling the NexentaEdge Cinder Plugin

Overview

This chapter describes how to set up a NexentaEdge cluster to provide block storage functions to OpenStack Cinder. OpenStack Cinder includes an API that can manage a block storage backend. You can configure this storage backend to be a NexentaEdge iSCSI storage service.

To do this, you set up an iSCSI storage service, then install and enable the NexentaEdge Cinder plugin, which configures OpenStack Cinder to use the iSCSI service for storage.

If you deploy the iSCSI storage service in a high-availability (HA) configuration, where two gateway nodes provide a single target for initiator requests using a virtual IP (VIP) address, the NexentaEdge Cinder plugin automatically determines the VIP configured for the service.

Setting Up an iSCSI Storage Service for Cinder

To configure an iSCSI storage service to be used as backend storage for Cinder, follow the procedure for setting up an iSCSI storage service, as demonstrated in iSCSI Deployment Example. Create an iSCSI storage service, but do not create any LUNs.

Enabling the NexentaEdge Cinder Plugin

The NexentaEdge Cinder plugin allows the iSCSI storage service to operate as a block storage backend for OpenStack. It manages the interaction between the Cinder API and the iSCSI storage service.

For information about how to obtain the NexentaEdge Cinder plugin, install it, and configure it, see this link.
Deploying NexentaEdge Without Internet Access

This chapter includes the following topics:

- Overview
- Deployment Procedure

Overview

During the normal installation and deployment process for NexentaEdge, packages and other components are downloaded from the Internet and installed on the nodes. However, for some installations, downloading is not possible, since the nodes may be blocked from the Internet due to security requirements.

To accommodate this kind of installation, Nexenta provides an alternative deployment method that requires no Internet access. The operating system, all required packages, and the NEDEPLOY and NEADM tools are deployed on the node using a locally mounted ISO file; no components are downloaded from the Internet.

After the operating system is installed and the required packages and NEDEPLOY and NEADM tools are copied to the local node, you can then install NexentaEdge on the node using the procedures in the Installing NexentaEdge chapter. Any differences between the normal vs. non-Internet methods are noted where applicable.

To deploy NexentaEdge using the non-Internet method, you boot the node from the ISO file, which starts the operating system installer, and also copies the required packages and NEDEPLOY and NEADM tools to a directory on the node. After the operating system is installed, you extract and run the NEDEPLOY tool, specifying that it use the packages in the local directory rather than downloading them from the Internet. You then repeat this process on each node where you are deploying NexentaEdge.

Note that all of the software and hardware requirements listed in the Prerequisites chapter (with the exception of Internet access) apply when deploying NexentaEdge using the non-Internet method.

Deployment Procedure

To deploy NexentaEdge on a node without Internet access:

1. Obtain the NexentaEdge ISO file from Nexenta. The file will have a name similar to `nexentaedge-ubuntu-2.1.3-buildnumber.iso`. Copy the ISO file to the NexentaEdge deployment workstation.
2. Boot the node from the ISO file. This starts the operating system installer program.
3. Install the operating system on the node.
4. After the operating system is installed, extract the NDEPLOY and NEADM tools to the home directory.

Example:

$ cd ~
$ tar -xzf /nexenta/nedeploy-ubuntu_linux_latest_x64.tar.gz
$ tar -xzf /nexenta/neadm-ubuntu_linux_latest_x64.tar.gz

5. Set the PATH environment variable to use the directories where you extracted the archives.

Example:

$ PATH=$PATH:~/nedeploy
$ PATH=$PATH:~/neadm

6. After you extract the NDEPLOY tool, you can use it to deploy NexentaEdge as described in the Installing NexentaEdge chapter, with the following exceptions:

- The NEDEPLOY tool can install NexentaEdge only on the local host. You cannot specify a remote host in the nedeploy deploy solo command.
- You must include the --darksite parameter in the nedeploy deploy solo command, specifying the local directory where the required packages were installed. By default, this directory is /nexenta.

Example:

$ ./nedeploy deploy solo localhost root:password -i eth1 -t capacity -m --darksite=/nexenta/

7. Repeat this procedure on each node you are adding to your NexentaEdge cluster. After deploying NexentaEdge on all of the nodes, you can initialize the cluster as described in Initializing the NexentaEdge Cluster.
Starting the NexentaEdge GUI

This chapter includes the following topics:

- Overview
- Before You Begin
- Deploying the NexentaEdge GUI as a Dashboard Service
- Manually Deploying the NexentaEdge GUI Using Docker

Overview

NexentaEdge features a separately installed graphical user interface (GUI). The NexentaEdge GUI allows you to view information about the physical cluster (status for nodes and storage devices) and view and configure logical clusters (storage services, clusters, tenants, and so on).

If you installed NexentaEdge as a Docker deployment (the default) use the procedure in Deploying the NexentaEdge GUI as a Dashboard Service. Otherwise, use the procedure in Manually Deploying the NexentaEdge GUI Using Docker.

Before You Begin

Before starting the NexentaEdge GUI, do the following:

- Install and initialize your NexentaEdge cluster according to the instructions in this guide. The NexentaEdge GUI is not compatible with NexentaEdge releases prior to 2.0.
- Note the URL for the NexentaEdge management controller node. You can display this setting by viewing the contents of the /neadm/.neadmrc file on the deployment workstation. The setting looks something like this: API_URL=http://10.3.30.32:8080.
Deploying the NexentaEdge GUI as a Dashboard Service

If you deployed NexentaEdge using Docker containers (the default), use the following procedure to deploy the NexentaEdge GUI as a Dashboard service. This procedure creates and enables a Dashboard service called *nedgeui*, which can be monitored using the GUI itself or NEADM.

1. Log in to the NexentaEdge deployment workstation (that is, the workstation where the NEADM tool is installed).
2. Use the following command to create a Dashboard service called *nedgeui*:
   ```bash
   $ neadm service create dashboard nedgeui
   
   System response:
   Service nedgeui created
   
   3. List the IDs of the nodes in the NexentaEdge cluster:
      ```bash
      $ neadm system status
      
      System response:
      
      | ZONE:HOST:CID | SID           | UTIL | CAP    | CPU           | MEM        | DEVs | STATE |
      |---------------|---------------|------|--------|---------------|------------|------|-------|
      | 0:node32:     | B5DEF7B9B3C690... | 0%   | 160G   | 2/0.2@2.34Ghz | 4.16G/7.8G | 4/4   | ONLINE |
      | 0:node33:     | [M] B3EF9C30B6F79F... | 0%   | 160G   | 2/0.1@2.34Ghz | 4.18G/7.8G | 4/4   | ONLINE |
      | 0:node34:     | 155B81C291FA4E... | 0%   | 160G   | 2/0.1@2.34Ghz | 4.18G/7.8G | 4/4   | ONLINE |
      | 0:node35:     | 9BB624B94C8E2C... | 0%   | 160G   | 2/0.1@2.34Ghz | 4.15G/7.8G | 4/4   | ONLINE |
      | 0:node36:     | F88C6C3180C34E... | 0%   | 160G   | 2/0.1@2.34Ghz | 4.15G/7.8G | 4/4   | ONLINE |
      
      4. Copy the ID or hostname of the server node where you want to deploy the Docker container that runs the NexentaEdge GUI.
5. Use the following command to make the node a gateway node for the Dashboard service.
   ```bash
   $ neadm service add nedgeui <server-id>
   
   Example:
   ```bash
   
   $ neadm service add nedgeui AC4F3E7923F2600BFE31CAC961197D
   
   System response:
   Service nedgeui added to AC4F3E7923F2600BFE31CAC961197D
   
   6. Use the following command to enable the Dashboard service:
   ```bash
   $ neadm service enable nedgeui
   
   System response:
   Service nedgeui enabled
   
   7. Use the following command to verify that the Dashboard service is enabled.
   ```bash
   $ neadm service list
System response:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NAME</th>
<th>SERVERID</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>dashboard</td>
<td>nedgeui</td>
<td>AC4F3E7923F2600BFE31CAC961197D</td>
<td>enabled</td>
</tr>
</tbody>
</table>

8. Using a browser, log into the NexentaEdge GUI with a URL in the following format:

```
<nedgegui-container>:<port>
```

where `<nedgegui-container>` is the management IP address or hostname of the server node where you deployed the Docker container that runs the NexentaEdge GUI, and `<port>` is either 3000 for HTTP or 3443 for HTTPS. This opens a NexentaEdge login window in the browser.

9. In the NexentaEdge login window, enter the username/password for the NexentaEdge REST user. The default is `admin/nexenta`. Click **Login**.
10. The NexentaEdge GUI window opens, displaying information about your cluster.

Manually Deploying the NexentaEdge GUI Using Docker

Use the procedure below if you need to manually start the NexentaEdge GUI for your cluster. You may first need to install the Docker Engine on your workstation if it is not already installed.

1. Install the Docker Engine on your workstation.
   For Windows/Mac systems, download and install Docker Toolbox, which is available from this link: https://www.docker.com/products/docker-toolbox
   For Linux systems, follow the instructions appropriate for your system:
   https://docs.docker.com/engine/installation/linux/ubuntu_linux/
   https://docs.docker.com/engine/installation/linux/rhel/
   https://docs.docker.com/engine/installation/linux/cen_tos/

2. Once the Docker Engine is installed, open a Terminal/Console window.

3. Enter the following commands:
   $ docker pull nexenta/nedgeui:2.1.3
   $ docker run -d -p 3000:3000 -e API_ENDPOINT=<url> nexenta/nedgeui:2.1.3

   where <url> is the URL for the NexentaEdge management controller node (that is, the setting for API_URL in /neadm/.neadmrc). This opens a NexentaEdge login window in a browser.
4. In the NexentaEdge login window, enter the username/password for the NexentaEdge REST user. The default is admin/nexenta. Click Login.

5. The NexentaEdge GUI window opens, displaying information about your cluster.
Upgrading NexentaEdge to the Latest Release

This chapter includes the following topics:

- Overview
- Upgrade Procedure

Overview

This chapter describes how to upgrade the nodes in your NexentaEdge cluster to a later release of the software. If the nodes in your NexentaEdge cluster are running version 2.1.2 or higher, you can use the procedure in this chapter to upgrade to the latest release.

This procedure upgrades the NexentaEdge core files on the nodes in your cluster. The upgrade procedure can be either disruptive or non-disruptive to service provided to clients. Whether you can perform a disruptive or non-disruptive upgrade depends on the failure domain configured for your NexentaEdge cluster. See Failure Domains in a NexentaEdge Cluster for a description of each type of failure domain.

Disruptive Upgrade

In a disruptive upgrade, you shut down NexentaEdge services on all of the data nodes in your cluster, upgrade the software, then restart the services on the nodes.

The advantage of this upgrade method is that the stored data on the nodes will be the same after the upgrade as it was before the upgrade; assuming the nodes are restarted at the same time, there will be no data inconsistency across the nodes. The disadvantage is that clients do not have access to data stored in the cluster while the upgrade is in progress.

The disruptive upgrade process is supported for Zone-level, Server-level, and Disk-level failure domains. If your cluster uses the Disk-level failure domain, disruptive upgrade is the only upgrade type supported.

Non-Disruptive Upgrade

In a non-disruptive upgrade, you shut down NexentaEdge services on one of the nodes, then upgrade the software on the stopped node, while leaving the other nodes active to provide service to clients.

The advantage of this method is that service to clients is not interrupted while the upgrade is in progress. While a node is being upgraded, the other active nodes continue to provide access to cluster storage. The disadvantage is that the stored data on the non-upgraded nodes may temporarily be inconsistent with the nodes that have been shut down and upgraded.

The non-disruptive upgrade process is supported on the Zone-level and Server-level failure domain types. It is not supported for the Disk-level failure domain type.
iSCSI Considerations for Non-Disruptive Upgrade

If you are doing a non-disruptive upgrade for an iSCSI storage service group, note the following:

- The iSCSI storage service group must be deployed in a high-availability (HA) configuration.
- The iSCSI storage service group must have more than one gateway node providing iSCSI target functionality for iSCSI initiators.
- Upgrade the data nodes first, one at a time.
- After the data nodes are upgraded, then upgrade the gateway nodes one at a time, so that one of the gateway nodes in the HA pair continues to provide service while the other is being upgraded. When you upgrade the active gateway node, the other node automatically starts providing service to clients.

Amazon S3 Considerations for Non-Disruptive Upgrade

If you are doing a non-disruptive upgrade for an Amazon S3 storage service group, note the following:

- The Amazon S3 storage service group must have more than one gateway node providing service to clients.
- Upgrade the data nodes first, one at a time.
- After the data nodes are upgraded, then upgrade the gateway nodes one at a time, so that at least one of the gateway nodes continues to provide service at all times.

Upgrade Procedure

Use the procedures in this section to upgrade the core files on the nodes in your NexentaEdge cluster. You can perform the upgrade either using the NEDEPLOY tool, or manually using Ubuntu or Red Hat/CentOS commands.

Once the nodes are upgraded, you can sync them with the NexentaEdge GUI so the nodes appear in GUI displays.

Upgrading NexentaEdge Using the NEDEPLOY Tool

If you use the NEDEPLOY tool to perform the upgrade, the procedure is the same for Ubuntu and Red Hat/CentOS systems.

- **To upgrade the nodes in your NexentaEdge cluster using the NEDEPLOY tool**

  1. Determine the version of NexentaEdge installed on the nodes in the cluster. Nodes running NexentaEdge version 2.1.2 or higher can be upgraded to a later version.

     To display the NexentaEdge version on a specific node, log into the deployment workstation and enter the following command:

     ```bash
     $ neadm system version <node_name>
     ```
Example:

$ neadm system version node101

System response:

NexentaEdge Version: 2.1.2-1177
NexentaEdge Build #: 1177

2. Obtain the latest versions of the NEDEPLOY and NEADM tools from Nexenta and extract the archive files to a directory on the NexentaEdge deployment workstation.

Example:

$ tar -xzf <download-directory>/nedeploy-linux_2.1.3-6558_x64.tar.gz
$ tar -xzf <download-directory>/neadm-linux_2.1.3-6558_x64.tar.gz

3. Stop NexentaEdge services on the node(s) to be upgraded. If you are doing a disruptive upgrade, enter this command on all of the nodes in your cluster. If you are doing a non-disruptive upgrade, enter this command on only the node you are currently upgrading.

$ service nedge stop

4. From the NexentaEdge deployment workstation, use the following command to deploy the NexentaEdge software to the nodes:

$ nedeploy deploy solo {<ip-address>[:port] | <nodename>}
<username:password>\ -i <interface> [-I <interface>] [-t <profile>]
[-x <disks-to-exclude>] [-X <disks-to-reserve>] [-z <zone>]
[-F <filesystem-type>] [-m] [--docker] [--data-container] [-d <TBs>]
[-P <number-of-partitions>] --upgrade

You specify the --upgrade option to upgrade the specified node. See Deploying NexentaEdge to the Nodes for a description of the other parameters in the nedeploy command.

Example:

$ nedeploy deploy solo 10.3.30.32 root:password -i eth1 -t capacity
-x sdau,sdaw --upgrade

5. Once all the nodes have been upgraded to the new NexentaEdge version, enter the following commands to verify that all of the services are running:

$ neadm system status
$ neadm service list

Upgrading NexentaEdge Using Ubuntu or Red Hat / CentOS Commands

To upgrade the nodes in your NexentaEdge cluster manually using Ubuntu or Red Hat/CentOS commands

1. Determine the version of NexentaEdge installed on the nodes in the cluster. Nodes running NexentaEdge version 2.1.2 or higher can be upgraded to a later version.

To display the NexentaEdge version on a specific node, log into the deployment workstation and enter the following command:

$ neadm system version <node_name>
Example:

$ neadm system version node101

System response:

NexentaEdge Version: 2.1.2-1177
NexentaEdge Build #: 1177

2. Stop NexentaEdge services on the node(s) to be upgraded. If you are doing a disruptive upgrade, enter this command on all of the nodes in your cluster. If you are doing a non-disruptive upgrade, enter this command on only the node you are currently upgrading.

$ service nedge stop

3. Enter the following commands to verify that all NexentaEdge services have stopped:

$ ps auxf | grep nedge
$ ps auxf | grep ccow

4. If you are using raw disks, check that the process rdhold is no longer running. If it is, kill the process using its PID number. For example:

$ ps auxf | grep rdhold
rdhold        1646  0.0  0.2   4272  1856 pts/1    S+   13:29   0:00
$ kill -9 1646

5. For Ubuntu systems, do the following:
   a. Edit the file /etc/apt/sources.list.d/nedge.list and change the NexentaEdge build number to the version to which you are upgrading.
   b. Enter the following command to obtain the new version of NexentaEdge:

$ apt-get update

   c. Enter the following command to upgrade the node to the new version of NexentaEdge:

$ apt-get install nedge-core

d. Enter Y to confirm the upgrade.

e. After the upgrade is complete, enter the following command to check that the upgraded version of NexentaEdge is in the base OS packaging system.

$ dpkg -s nedge-core | grep "Version:"

System response:

Version: 2.1.3-6558

6. For Red Hat/CentOS systems, do the following:
   a. Edit the file /etc/yum.repos.d/nedge.repo and change the NexentaEdge build number to the version to which you are upgrading.
   b. Enter the following command to upgrade the node to the new version of NexentaEdge:

$ yum upgrade nedge-core

c. Enter Y to confirm the upgrade.
d. After the upgrade is complete, enter the following command to check that the upgraded version of NexentaEdge is in the base OS packaging system.

```
$ yum info nedge-core | grep "Version    :
```

System response:

```
Version    : 2.1.3_3247
```

7. Edit the file `/opt/nedge/etc/ccow/rt-rd.json` and add the following entry:

```
"journal_maxentries": 64
```

8. Enter the following commands to ensure that NexentaEdge uses the new `journal_maxentries` setting.

```
$ source /opt/nedge/env.sh
$ vdevchk -f -F
```

9. Once all the nodes have been upgraded to the new NexentaEdge version, enter the following commands on the deployment workstation to verify that all of the services are running:

```
$ neadm system status
$ neadm service list
```

Syncing the NexentaEdge GUI With the Upgraded Nodes

If you are using the NexentaEdge GUI as a Dashboard service, do the following to sync the GUI with the upgraded nodes.

1. Log in to the NexentaEdge deployment workstation (that is, the workstation where the NEADM tool is installed).

2. Use the following command to disable the existing Dashboard service `nedgeui`:

```
$ neadm service disable nedgeui
```

3. Log into the server node where the Docker container that runs the NexentaEdge GUI is deployed and open a Terminal/Console window.

4. Enter the following command:

```
$ docker pull nexenta/nedgeui:2.1.3
```

5. On the NexentaEdge deployment workstation, use the following command to re-enable the `nedgeui` Dashboard service:

```
$ neadm service enable nedgeui
```