



## **Cisco Extensible Network Controller Deployment Guide, Release 1.5**

**First Published:** February 24, 2014

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-31558-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)



The Java logo is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. or other countries.

© 2014 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### Preface

#### Preface v

Audience v

Document Conventions v

Obtaining Documentation and Submitting a Service Request vi

---

### CHAPTER 1

#### Cisco XNC Overview 1

About Cisco Extensible Network Controller 1

System Requirements for Cisco XNC 2

Supported Web Browsers for Cisco XNC 2

---

### CHAPTER 2

#### Deploying Cisco XNC 3

Installing Cisco XNC 3

Installing or Upgrading Cisco XNC Software 3

Installing the Cisco XNC Application 4

Upgrading the Cisco XNC Application 5

Installing Additional Cisco XNC Applications 6

Starting the Cisco XNC Application 7

Verify That Cisco XNC is Running 8

Using the TLS KeyStore and TrustStore Files 8

Creating the TLS KeyStore File 8

Creating the TLS TrustStore File 9

Running the TLS KeyStore Password Configuration Script 9

Logging in to the Cisco XNC GUI 10

Configuring Cisco XNC 10

Configuring High Availability Clusters 10

Password Protecting the High Availability Clusters 11

Editing the Configuration Files for Cisco Nexus 3000 Series Switches 12

Running the Backup and Restore Script	13
Running the Password Recovery Script	13
Uninstalling the Cisco XNC Application	13



## Preface

---

This preface contains the following sections:

- [Audience, page v](#)
- [Document Conventions, page v](#)
- [Obtaining Documentation and Submitting a Service Request, page vi](#)

## Audience

This guide is intended for site administrators who will manage Cisco Smart-enabled software installation and licensing.

## Document Conventions

Command descriptions use the following conventions:

Convention	Description
<b>bold</b>	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x   y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x   y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y   z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
<i>variable</i>	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
<code>screen font</code>	Terminal sessions and information the switch displays are in screen font.
<b><code>boldface screen font</code></b>	Information you must enter is in boldface screen font.
<i><code>italic screen font</code></i>	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



#### Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



#### Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*, at: <http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.







## CHAPTER

# 1

## Cisco XNC Overview

---

This chapter contains the following sections:

- [About Cisco Extensible Network Controller, page 1](#)
- [System Requirements for Cisco XNC, page 2](#)
- [Supported Web Browsers for Cisco XNC, page 2](#)

## About Cisco Extensible Network Controller

Cisco Extensible Network Controller (Cisco XNC) is a software platform that serves as an interface between the network elements (southbound) and third-party applications (northbound). Cisco XNC is a JVM-based application that runs on a Java Virtual Machine (JVM). Cisco XNC is based on a highly available, scalable, and extensible architecture that supports a network. Cisco XNC is built for extensibility using the Open Services Gateway initiative (OSGi) framework, which allows new functionality to be added.

Cisco XNC can support multiple protocol plugins in the southbound direction. In the current release, Cisco Plug-in for OpenFlow 1.0 is available.

Cisco XNC provides the following:

- Multiprotocol capability with the Cisco Plug-in for OpenFlow version 1.0 available in this release.
- Functionality to support network visibility and programmability, such as network topology discovery, network device management, forwarding rules programming, and access to detailed network statistics.
- A Service Abstraction Layer (SAL) that enables modular southbound interface support, such as OpenFlow.
- Consistent management access through the GUI or through Java or Representational State Transfer (REST) northbound APIs.
- Security features, such as role-based access control (RBAC), and integration with an external Active Directory using RADIUS or TACACS for authentication, authorization, and accounting (AAA) functions.
- Troubleshooting tools, such as analytics gathering and diagnostic packet injection.
- Cisco advanced features such as Topology Independent Forwarding (TIF), which enables the administrator to customize the path a data flow takes through the network.

- Cisco network applications such as Network Slicing that allows logical partitioning of the network using flow specification, and Monitor Manager, that provides visibility into the network traffic.
- High-availability clustering to provide scalability and high availability.
- The Cisco Open Network Environment Platform Kit (Cisco onePK) version 1.1.0 is supported in this release of Cisco XNC. The Cisco onePK plug-in communicates with the onePK agent.
- Support for onePK devices in the network and the ability to install TIF rules on onePK devices.
- A CLI framework for Cisco XNC.
- Virtual Patch Panel Application (P2P Forwarding application) provides port-to-port traffic management within a switch or across the network without any need for physical connection changes or rewiring.

## System Requirements for Cisco XNC

Cisco XNC runs in a JVM. As a Java-based application, Cisco XNC can run on any Linux-based x86 server. For best results, we recommend the following:

- One 6-core CPU at 2 GHz or better.
- A minimum of 8 GB of memory.
- A minimum of 40 GB of free hard disk space must be available on the partition where you will be installing the Cisco XNC application.
- A recent 64-bit Linux distribution that supports Java, such as the following:
  - Ubuntu Linux
  - Red Hat Enterprise (RHEL) Linux
  - Fedora Linux
- Java Virtual Machine 1.7 or later releases
- A \$JAVA\_HOME environment variable in your profile set to the path of the JVM.
- Python 2.7.3 to support the backup and restore script.

## Supported Web Browsers for Cisco XNC

The following web browsers are supported for Cisco XNC:

- Firefox 18.x and later versions
- Chrome 24.x and later versions

**Note**

Javascript 1.5 or a later version must be enabled in your browser.



## Deploying Cisco XNC

---

This chapter contains the following sections:

- [Installing Cisco XNC, page 3](#)
- [Using the TLS KeyStore and TrustStore Files, page 8](#)
- [Logging in to the Cisco XNC GUI, page 10](#)
- [Configuring Cisco XNC, page 10](#)
- [Running the Backup and Restore Script, page 13](#)
- [Running the Password Recovery Script, page 13](#)
- [Uninstalling the Cisco XNC Application, page 13](#)

## Installing Cisco XNC

### Installing or Upgrading Cisco XNC Software

- To complete a new installation of Cisco XNC, follow the steps in the [Installing the Cisco XNC Application, on page 4](#).
- To upgrade Cisco XNC, version 1.0 to Cisco XNC, version 1.5, follow the steps in [Upgrading the Cisco XNC Application, on page 5](#)

## Installing the Cisco XNC Application

- 
- Step 1** In a web browser, navigate to [Cisco.com](https://Cisco.com).
- Step 2** Under **Support**, click **All Downloads**.
- Step 3** In the center pane, click **Cloud and Systems Management**.
- Step 4** If prompted, enter your Cisco.com username and password to log in.
- Step 5** In the right pane, click **Network Controllers and Applications**, and then click **Cisco Extensible Network Controller (XNC)**.
- Step 6** Download the Cisco XNC application bundle and any additional applications that you have purchased.
- Step 7** Create a directory in your Linux machine where you plan to install Cisco XNC.  
For example, in your Home directory, create `CiscoXNC`.
- Step 8** Copy the Cisco XNC zip file into the directory that you created.
- Step 9** Unzip the Cisco XNC zip file.  
The Cisco XNC software is installed in a directory called `xnc`. The directory contains the following:
- `xncbundle` file—The Cisco XNC application bundle.
  - `runxnc.sh` file—The file that you use to launch Cisco XNC on Linux or UNIX systems.
  - `version.properties` file—The Cisco XNC build version.
  - `captures` directory—The directory that contains output dump files from analytics run in Cisco XNC.
  - `configuration` directory—The directory that contains the Cisco XNC basic initialization files.  
This directory also contains the `startup` subdirectory where configurations are saved.
  - `bin` directory—This directory contains the following scripts:
    - `adminpasswordreset.sh` file—The script to reset the default network-admin user password to the factory default.
    - `backup.py` file—The Cisco XNC backup script.
    - `configkeystorepwd.sh` file—The TLS KeyStore password configuration script.
  - `etc` directory—The directory that contains profile information.
  - `lib` directory—The directory that contains the Cisco XNC Java libraries.
  - `logs` directory—The directory that contains the Cisco XNC logs.
- Note** The `logs` directory is created after the Cisco XNC application is started.
- `plugins` directory—The directory that contains the OSGi plugins.
  - `ObjectStore` directory—The directory that contains the Cisco XNC objects.
  - `work` directory—A webserver working directory that is created after the Cisco XNC application is started.
-

## Upgrading the Cisco XNC Application

Cisco provides an upgrade script to upgrade a Cisco XNC 1.0 installation to Cisco XNC 1.5. The upgrade process replaces all of the version 1.0 product files.

A backup archive is created during the process that can be used to restore the original installation, if required.

Running the upgrade script upgrades the installation and configuration. However, any changes you made to the shell scripts or configuration files, for example, `runxnc.sh` and `config.ini`, will be overwritten. Once you have completed the upgrade process, you must manually reapply your changes to those files.

To upgrade Cisco XNC 1.0 to Cisco XNC 1.5 follow the steps below:

### Before You Begin

- Use the same type of installer that you used when installing Cisco XNC version 1.0. For example, if you have Cisco XNC with the Monitor Manager application installed, you must use the installation software for Cisco XNC with Monitor Manager.
- Stop all controller instances that use the Cisco XNC 1.0 installation. This will avoid conflicts with the file system, which is updated during upgrade.
- If you are using high availability clustering, stop all XNC instances in the cluster to ensure that there are no inconsistencies.

- 
- Step 1** In a web browser, navigate to [Cisco.com](https://www.cisco.com).
- Step 2** Under **Support**, click **All Downloads**.
- Step 3** In the center pane, click **Cloud and Systems Management**.
- Step 4** In the right pane, click **Network Controllers and Applications**, and then click **Cisco Extensible Network Controller (XNC)**.
- Step 5** Download the version 1.5 Cisco XNC application bundle and any additional applications that you currently have installed.
- Step 6** Create a temporary directory in your Linux machine where you plan to upgrade Cisco XNC. For example, in your Home directory, create `CiscoXNC_Upgrade`.
- Step 7** Extract the Cisco XNC 1.5 zip file into the temporary directory that you created.
- Step 8** Navigate to the `xnc` directory that was created when you installed the Cisco XNC version 1.0 software.
- Step 9** Stop all running Cisco XNC 1.0 processes.
- Step 10** Backup the Cisco XNC 1.0 installation using your standard backup procedures.
- Step 11** Navigate to the `xnc/bin` directory in the temporary directory that you created for the 1.5 upgrade software.
- Step 12** Run the upgrade script using the following syntax: `./upgrade [--upgrade | --rollback] <xnc_directory_to_be_upgraded>`  
You can use one of the following options:

Option	Description
<code>--upgrade</code> <code>&lt;xnc_directory_to_be_upgraded&gt;</code>	Upgrades Cisco XNC and creates a default backup.

Option	Description
<b>--upgrade</b> <xnc_directory_to_be_upgraded> <backup_file_path>	Upgrades Cisco XNC and a backup in the directory path that you set.
<b>--rollback]</b> <xnc_directory_to_rollback>	Rolls back to the previous Cisco XNC installation using the default backup.
<b>--rollback]</b> <xnc_directory_to_rollback> <backup_file_path>	Rolls back to the previous Cisco XNC installation using the backup in the directory path that you set.
<b>--check</b> <xnc_directory_to_be_upgraded>	Validates the Cisco XNC installation, but does not perform an upgrade.
<b>--help</b>	Displays the options for the upgrade script.

- Step 13** After the script has successfully completed, navigate to the `xnc` directory where you originally installed Cisco XNC.
- Step 14** Start the Cisco XNC processes that you previously stopped.
- Step 15** Run the password recovery script to reset the admin password. See [Running the Password Recovery Script](#), on page 13.
- Step 16** Log in to Cisco XNC as the admin user, and reset the passwords for all users.  
For more information, see the Users and Roles section of the *Cisco Extensible Network Controller Configuration Guide*.
- Step 17** If there are any upgrade-related issues, perform the following tasks:
- Stop the Cisco XNC processes.
  - Run the rollback command.
  - Restart the Cisco XNC processes.

## Installing Additional Cisco XNC Applications

### Before You Begin

You must purchase additional Cisco XNC applications and download the .zip files from [Cisco.com](#). We recommend backing up your configuration before installing new applications.

- Step 1** Open a command window where you installed Cisco XNC.
- Step 2** Unzip the application file, and place the .jar file into the `xnc/plugins` directory that was created when you installed the software.

## Starting the Cisco XNC Application

**Step 1** Navigate to the `xnc` directory that was created when you installed the software.

**Step 2** Start Cisco XNC using the following syntax: `./runxnc.sh`

You can use one of the following options:

Option	Description
no option	Starts Cisco XNC with the <b>-start</b> option.
<b>-jmx</b>	Enables JMX remote access on the Cisco XNC JVM, which can be used to troubleshoot performance issues.
<b>-jmxport</b> <i>num</i>	Enables JMX remote access on the specified JVM port.
<b>-debug</b>	Enables debugging on the Cisco XNC JVM.
<b>-debugsuspend</b>	Suspends the Cisco XNC startup until a debugger is connected.
<b>-debugport</b> <i>port_number</i>	Enables debugging on the specified JVM port.
<b>-start</b>	Starts Cisco XNC and provides Secure Shell (SSH) access to the controller on port 2400. <b>Note</b> The SSH server can be accessed by any Cisco XNC user with the network-administrator role.
<b>-start</b> <i>port_num</i>	Starts Cisco XNC and provides SSH access to the controller on the specified port number. <b>Note</b> The SSH server can be accessed by any Cisco XNC user with the network-administrator role.
<b>-stop</b>	Stops Cisco XNC.
<b>-status</b>	Displays the status of Cisco XNC.
<b>-console</b>	Starts Cisco XNC with the OSGi console.
<b>-help</b>	Displays the options for the <code>runxnc.sh</code> script.
<b>-tls</b>	Enables TLS secure connections between Cisco XNC and OpenFlow or onePK switches.  To enable TLS, start the controller with the following options: <code>./runxnc.sh -tls -tlskeystore keystore_file_location -tlstruststore truststore_file_location</code>

## Verify That Cisco XNC is Running

- 
- Step 1** Open a command window where you installed Cisco XNC.
- Step 2** Navigate to the `xnc` directory that was created when you installed the software.
- Step 3** Run the following script: `./runxnc.sh -status`  
 The controller outputs the following, which indicates that the controller is running the java process with PID 21680:
- ```
Controller with PID:21680 -- Running!
```
- 

### What to Do Next

Connect the switches to the controller. For more information, see the appropriate configuration guide.

## Using the TLS KeyStore and TrustStore Files

Enabling the Transport Layer Security (TLS) connections between Cisco XNC and OpenFlow or onePK switches requires TLS KeyStore and TLS TrustStore files.

- The TLS KeyStore file contains the private key and certificate information used by Cisco XNC.
- The TLS TrustStore file contains the Certification Authority (CA) certificates used to sign the certificates on the connecting switches.

Both the TLS KeyStore and TLS TrustStore files are password protected.

If you choose to use TLS connections in your Cisco XNC implementation, all of the connections in the network must be TLS encrypted, and you must run Cisco XNC with TLS enabled. After you provide both the TLS KeyStore and TLS TrustStore files, you can run the TLS KeyStore password configuration script to provide the passwords for Cisco XNC to unlock the KeyStore files.

## Creating the TLS KeyStore File

- 
- Step 1** Provide the following files:
- `xnc-private.pem`—A .pem file that contains the Cisco XNC private key.
  - `xnc-cert.pem`—A .pem file that contains the Cisco XNC certificate.
- Step 2** Run the following command: `cat xnc-privkey.pem xnc-cert.pem > xnc.pem`  
 The `xnc.pem` file is created with the private key and certificate.
- Step 3** Run the following command: `openssl pkcs12 -export -out xnc.p12 -in xnc.pem`
- Step 4** Enter a password at the prompt.



**Note** You must use the same password for Step 4 and Step 6. The password must contain at least six characters

The xnc.pem file is converted to a password-protected .p12 file.

**Step 5** Run the following command: **keytool -importkeystore -srckeystore xnc.p12 -srcstoretype pkcs12 -destkeystore tlsKeyStore -deststoretype jks**

**Step 6** Enter a password at the prompt.

**Note** You must use the same password for step 4 and step 6. The password must contain at least six characters

The xnc.p12 is converted to a password-protected Java KeyStore file.

---

## Creating the TLS TrustStore File

---

**Step 1** Create a file called sw-cacert.pem to contain the CA certificate for the switch.

**Step 2** Run the following command: **keytool -import -alias swca1 -file sw-cacert.pem -keystore tlsTrustStore**

**Step 3** Enter a password at the prompt.

The sw-cacert.pem file is converted into a password-protected Java TrustStore file.

**Step 4** If the switches in your network use more than one CA certificate, repeat Step 1 through Step 3 for each CA certificate that is used.

---

## Running the TLS KeyStore Password Configuration Script

The configkeystorepwd.sh script allows you to input the TLS KeyStore passwords so that the KeyStore files can be unlocked and used by Cisco XNC.

### Before You Begin

Ensure that the cURL program is installed.

---

**Step 1** Ensure Cisco XNC is running with TLS enabled.

**Step 2** Open a command window where you installed Cisco XNC.

**Step 3** Navigate to the xnc directory that was created when you installed the software.

**Step 4** Run the following command: **./configkeystorepwd.sh**

**Step 5** At the prompt, enter the following information:

- The Cisco XNC username
- The Cisco XNC password

- The TLS KeyStore password
  - The TLS TrustStore password
- 

## Logging in to the Cisco XNC GUI

You can log into the Cisco XNC GUI using HTTP or HTTPS:

- The default HTTP web link for the Cisco XNC GUI is `http://Controller_IP:8080`
- The default HTTPS web link for the Cisco XNC GUI is `https://Controller_IP:8443`

**Note**

Before you can use HTTPS, you must manually specify the `https://` protocol in your web browser.

---

---

**Step 1** In your web browser, enter the Cisco XNC GUI web link.

**Step 2** On the launch page, do the following:

- a) Enter your username and password.  
The default username and password is admin/admin.
- b) Click **Log In**.

---

## Configuring Cisco XNC

### Configuring High Availability Clusters

Cisco XNC supports high availability clustering in active/active mode with up to five controllers. To use high availability clustering with Cisco XNC, you must edit the `config.ini` file for each instance of Cisco XNC.

**Before You Begin**

- All IP addresses must be reachable and capable of communicating with each other.
- All switches in the cluster must connect to all of the controllers.
- All controllers must have the exact same HA clustering configuration information in the `config.ini` files.
- All controllers must have the exact same information in the `xnc/configuration/startup` directory.

- If using cluster passwords, all controllers must have the exact same password configured in the `xncjgroups.xml` file. See [Password Protecting the High Availability Clusters](#), on page 11.

- 
- Step 1** Ensure that Cisco XNC is not running on any of the instances in the cluster.
- Step 2** Open a command window on one of the instances in the cluster.
- Step 3** Navigate to the `xnc/configuration` directory that was created when you installed the software.
- Step 4** Use any text editor to open the `config.ini` file.
- Step 5** Locate the following text:
- ```
# HA Clustering configuration (colon-separated IP addresses of all controllers that are part of the
cluster.)
# supernodes=<ip1>:<ip2>:<ip3>:<ipn>
```
- Step 6** Remove the comments on the `# supernodes` line, and replace `<ip1>:<ip2>:<ip3>:<ipn>` with the IP addresses for each instance of Cisco XNC in the cluster. You can enter from two to five IP addresses.
- Example:**
- ```
# HA Clustering configuration (colon-separated IP addresses of all controllers that are part of the
cluster.)
supernodes=<10.1.1.1>:<10.2.1.1>:<10.3.1.1>:<10.4.1.1>:<10.5.1.1>
```
- Step 7** Save the file and exit the editor.
- Step 8** Repeat Step 3 through Step 7 for each instance of Cisco XNC in the cluster.
- Step 9** Restart Cisco XNC.
- 

## Password Protecting the High Availability Clusters

You can password protect your HA clusters with the `xncjgroups.xml` file. This file must be exactly the same for each instance of Cisco XNC.

- 
- Step 1** Ensure that Cisco XNC is not running on any of the instances in the cluster.
- Step 2** Open a command window on one of the instances in the cluster.
- Step 3** Navigate to the `xnc/configuration` directory that was created when you installed the software.
- Step 4** Use any text editor to open the `xncjgroups.xml` file.
- Step 5** Locate the following text:
- ```
<!-- <AUTH auth_class="org.jgroups.auth.MD5Token" auth_value="ciscoXNC" token_hash="MD5"></AUTH> -->
```
- Step 6** Remove the comments from the AUTH line.
- Example:**
- ```
<AUTH auth_class="org.jgroups.auth.MD5Token" auth_value="ciscoXNC" token_hash="MD5"></AUTH>
```
- Step 7** (Optional) Change the password in the `auth_value` attribute.

By default, the cluster is protected with the password "ciscoXNC". You can change this password to whatever value you want, provided you make the same change on all machines in the cluster.

**Step 8** Save the file and exit the editor.

**Step 9** Repeat Step 4 through Step 8 for each instance of Cisco XNC in the cluster.

**Step 10** Restart Cisco XNC.

## Editing the Configuration Files for Cisco Nexus 3000 Series Switches

The following configuration settings can improve scalability when connecting to Cisco Nexus 3000 Series switches.

**Step 1** Navigate to the `xnc/configuration` directory that was created when you installed the software.

**Step 2** Use any text editor to open the `config.ini` file.

**Step 3** Update the following parameters:

| Name                        | Default Value | Recommended Value |
|-----------------------------|---------------|-------------------|
| of.messageResponseTimer     | 2000          | 60000             |
| of.switchLivenessTimeout    | 60500         | 120500            |
| of.flowStatsPollInterval    | 10            | 240               |
| of.portStatsPollInterval    | 10            | 240               |
| of.descStatsPollInterval    | 60            | 240               |
| of.barrierMessagePriorCount | 100           | 50                |
| of.discoveryInterval        | 300           | 300               |
| of.discoveryTimeoutMultiple | 2             | 2                 |

**Step 4** Save the file and exit the editor.

**Step 5** Restart Cisco XNC.

## Running the Backup and Restore Script

The backup script allows you to backup your Cisco XNC configurations and restore them later.

- 
- Step 1** Open a command window where you installed Cisco XNC.
- Step 2** Navigate to the `xnc` directory that was created when you installed the software.
- Step 3** Run the following command: **python backup.py**
- Step 4** At the prompt, perform one of the following tasks:
- To save the last configuration, enter **backup**.  
The script creates a timestamped tar file in the `xnc` directory with the following format:  
`xnc-yy-mm-dd_time.tar`
  - To restore a saved backup file, enter **restore filename**.  
The *filename* is the backup tar file.  
If prompted, choose Y to overwrite the existing configuration.
  - To exit the program, enter **exit**.
- Step 5** If you are restoring a configuration, stop and restart Cisco XNC for the configuration to take effect.
- 

## Running the Password Recovery Script

The password recovery script resets the password for the Cisco XNC Network Administrator user to the factory default.

- 
- Step 1** Open a command window where you installed Cisco XNC.
- Step 2** Navigate to the `xnc` directory that was created when you installed the software.
- Step 3** Run the following command: **./adminpasswordreset.sh**
- Step 4** At the prompt, choose y to reset the network administrator password.
- 

## Uninstalling the Cisco XNC Application

- 
- Step 1** Navigate to the directory where you created the Cisco XNC installation.  
For example, if you installed the controller in `Home/CiscoXNC`, navigate to the `Home` directory.

**Step 2** Delete the `CiscoXNC` directory.

---