Cisco Extensible Network Controller Deployment Guide, Release 1.6

First Published: June 11, 2014

Last Modified: July 25, 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-32188-01

CONTENTS

Preface

This preface contains the following sections:

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
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	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.
`variable`	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
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	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:

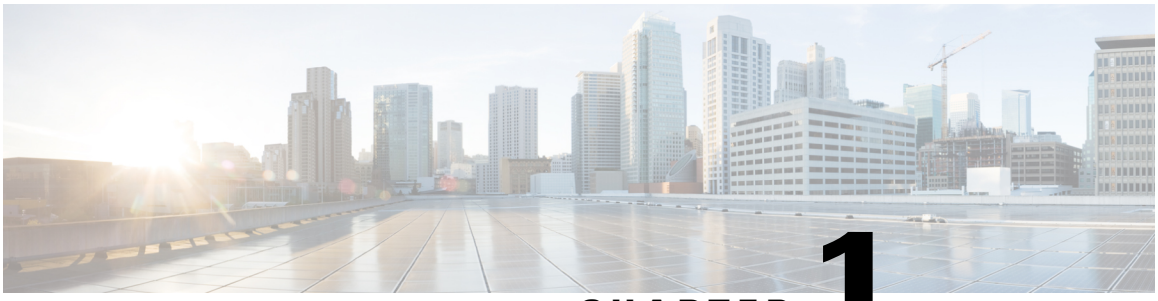
Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

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/c/en/us/td/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 Extensible Network Controller Overview

This chapter contains the following sections:

About Cisco Extensible Network Controller

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

Cisco XNC can support multiple protocol plugins in the southbound direction. In Release 1.6, Cisco Plug-in for OpenFlow 1.0 and the Cisco One Platform Kit (onePK) are supported.

Cisco XNC provides the following:

- Multiprotocol capability with the Cisco Plug-in for OpenFlow.
- 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 Cisco Monitor Manager, which 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 Release 1.6 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 command line interface (CLI) framework for Cisco XNC.
- The Virtual Patch Panel Application (Port-to-Port Forwarding application) provides port-to-port traffic management within a switch or across the network without any need for physical connection changes or rewiring.
- Access to the Cisco XNC northbound API content from the application menu bar that enables you to view the API definitions and related calls.

Guidelines and Limitations for Cisco XNC

Cisco Extensible Network Controller (XNC) runs in a Java Virtual Machine (JVM). As a Java-based application, Cisco XNC can run on any x86 server. For best results, we recommend the following:

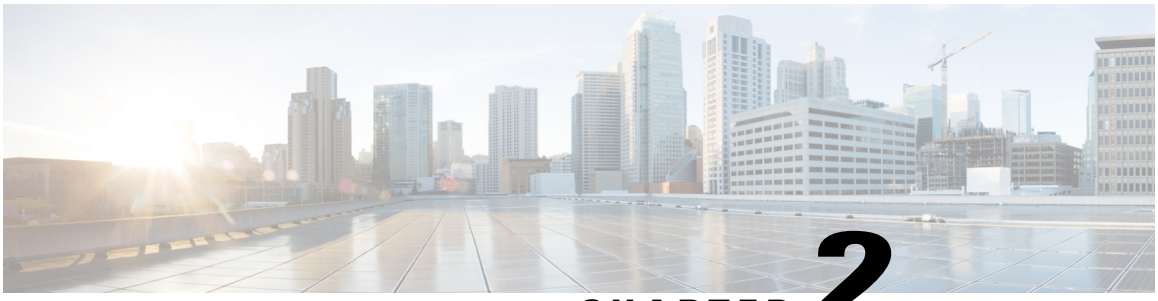
- One 6-core CPU at 2 GHz or higher.
- A minimum of 8 GB of memory.
- A minimum of 40 GB of free hard disk space must be available on the partition where you will be installing the Cisco XNC application.
- A 64-bit Linux distribution with Java, such as the following:
  - Ubuntu Linux
  - Red Hat Enterprise (RHEL) Linux
  - Fedora Linux
- Java Virtual Machine 1.7.x
- A $JAVA_HOME environment variable in your profile set to the path of the JVM.

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.



CHAPTER 2

Deploying Cisco Extensible Network Controller

This chapter contains the following sections:

Installing Cisco XNC

Installing or Upgrading the Cisco Extensible Network Controller Software

Important There is no direct upgrade path from Cisco XNC Release 1.0 to Cisco XNC Release 1.6. If you have Cisco XNC Release 1.0 installed and you want to update to Cisco XNC Release 1.6, you must first upgrade to Cisco XNC Release 1.5. See the *Cisco Extensible Network Controller Deployment Guide, Release 1.5*for the procedure.

- To complete a new installation of Cisco XNC, see Installing the Cisco Extensible Network Controller Application, on page 4.
- To upgrade Cisco XNC Release 1.5 to Cisco XNC Release 1.6, see Upgrading Cisco Extensible Network Controller Release 1.5 to Release 1.6, on page 5.

Installing the Cisco Extensible Network Controller Application

Step 1 In a web browser, navigate to 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:

- `runxnc.sh` file—The file that you use to launch Cisco XNC.
- `version.properties` file—The Cisco XNC build version.
- `captures` directory—The directory that contains output dump files from analytics run in Cisco XNC.

  Note The `captures` directory is created after you execute the Cisco XNC analytics tool.

- `configuration` directory—The directory that contains the Cisco XNC initialization files.

  This directory also contains the `startup` subdirectory where configurations are saved.

- `bin` directory—The directory that contains the following script:
  - `xnc` file—This script contains the Cisco XNC common CLI.
- `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.
- `work` directory—The webserver working directory.

  Note The work directory is created after the Cisco XNC application is started.

Upgrading Cisco Extensible Network Controller Release 1.5 to Release 1.6

You can use the **upgrade** command to upgrade a Cisco XNC Release 1.5 installation to Cisco XNC Release 1.6. This upgrade is called an in-place upgrade, which means that the product bits are replaced. A backup archive is created to restore your original installation, if necessary.

When you execute the **upgrade** command, the installation and the configuration are upgraded. However, any changes you made to the shell scripts or configuration files, for example,`runxnc.sh` and `config.ini`, are overwritten. After you complete the upgrade process, you must manually reapply your changes to those files.

Before You Begin

- Use the same type of installer that you used when installing Cisco XNC Release 1.5. 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.
- If you have upgraded from Cisco XNC Release 1.0 to Cisco XNC Release 1.5, reset the password, start the controller and save the configuration using the **Save** button at the top of the menu bar in Cisco XNC.
- Stop all controller instances that use the Cisco XNC 1.5 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.

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 Cisco Extensible Network Controller (XNC) Release 1.6 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 Release 1.6 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 Release 1.5 software.

Step 9 Stop all running Cisco XNC Release 1.5 processes.

Step 10 Backup the Cisco XNC Release 1.5 installation using your standard backup procedures.

Step 11 Navigate to the `xnc/bin` directory in the temporary directory that you created for the Release 1.6 upgrade software.

Step 12 Upgrade the application by entering the **./xnc upgrade --perform --target-home** {*xnc_directory_to_be_upgraded*} **[--verbose] [--backupfile** {*xnc_backup_location_and_zip_filename*}**]** command.
You can use one of the following options:

Option	Description
--perform --target-home {*xnc_directory_to_be_upgraded*}	Upgrades the Cisco XNC installation.
--perform --target-home {*xnc_directory_to_be_upgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*}	Upgrades the Cisco XNC installation and creates a backup .zip file in the directory path that you set. **Note** You must provide the name of the backup file and the .zip extension.
--rollback --target-home {*xnc_directory_to_be_upgraded*}	Rolls back to the previous Cisco XNC installation.
--rollback --target-home {*xnc_directory_to_be_upgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*}	Rolls back to the previous Cisco XNC installation using the backup file in the absolute path that you set.
--verbose	Displays detailed information to the console. This option can be used with any other option and is disabled by default.
--validate --target-home {*xnc_directory_to_be_upgraded*}	Validates the Cisco XNC installation.
./xnc help upgrade	Displays the options for the command.

Step 13 Navigate to the `xnc` directory where you originally installed Cisco XNC.

Step 14 Start the Cisco XNC processes that you previously stopped.

Note Press Ctrl–F5, or press the Cmd, Shift, and R keys simultaneously when you access Cisco XNC through a web UI following an upgrade.

Step 15 If you have any upgrade-related issues, perform the following tasks:

a) Stop all Cisco XNC processes.

b) Navigate to the temporary directory that you created in Step 6.

c) Enter the **./xnc upgrade --rollback --target-home** {*xnc_directory_to_be_downgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*} command.

d) Restart the Cisco XNC processes.

Note Press Ctrl–F5, or press the Cmd, Shift, and R keys simultaneously when you access Cisco XNC through a web UI following a rollback.

Installing Additional Cisco Extensible Network Controller Applications

Before You Begin

You must purchase additional Cisco XNC applications and download the .zip files from Cisco.com. We recommend that you backup your configuration before you install new applications.

Step 1 Open a command window where you installed Cisco XNC.

Step 2 Unzip the application file, and place the `.jar` files into the `xnc/plugins` directory that was created when you installed the software.

Starting the Cisco Extensible Network Controller Application

Step 1 Navigate to the `xnc/bin` directory.

Step 2 Change the default password supplied with Cisco XNC by entering the **./xnc reset-admin-password [--wait-seconds** {*wait_time*} **--password** {*password*}**]** command.
The {*password*} variable resets the administrator password to the value that you specify by restarting the user manager. The {*wait_time*} is the number of seconds to wait while the user manager restarts. The minimum {*wait_time*} value is 5 seconds and the maximum is 60 seconds.

Note
- The password must be from 8 to 256 characters, contain both uppercase and lowercase characters, and have at least one number and one nonalphanumeric character.
- If you leave the password blank, it is reset to the factory default of "admin".
- Each time that you reset the administrative password, make sure that the new password meets these requirements or you will not be able to log in to Cisco XNC.

Step 3 Navigate to the `xnc` directory and start Cisco Extensible Network Controller (XNC) by entering the **./runxnc.sh** command. You can use one of the following options:

Option	Description
no option	Starts Cisco XNC with the **-start** option.
-jmx	Enables Java Management Extensions (JMX) remote access on the Cisco XNC JVM, which can be used to troubleshoot performance issues.
-jmxport *port_number*	Enables JMX remote access on the specified JVM port.
-debug	Enables debugging on the Cisco XNC JVM.
-debugsuspend	Suspends the Cisco XNC startup until a debugger is connected.

Option	Description
-debugport *port_number*	Enables debugging on the specified JVM port.
-start	Starts Cisco XNC and provides Secure Shell (SSH) access to the controller on port 2400. **Note** The SSH server can be accessed by any Cisco XNC user with the network-administrator role.
-start *port_number*	Starts Cisco XNC and provides SSH access to the controller on the specified port number. **Note** The SSH server can be accessed by any Cisco XNCCisco Extensible Network Controller (XNC) user with the network-administrator role. The valid range of values for *port_num* is 1024 through 65535.
-stop	Stops Cisco XNC.
-status	Displays the status of Cisco XNC.
-console	Starts Cisco XNC with the OSGi console.
-help	Displays the options for the **./runxnc.sh** command.
-tls	Enables TLS secure connections between Cisco XNC and OpenFlow or onePK switches. To enable TLS, start the controller by entering the **./runxnc.sh -tls -tlskeystore** *keystore_file_location* **-tlstruststore** *truststore_file_location* command.

Verifying That Cisco Extensible Network Controller 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 Verify that XNC is running by entering the **./runxnc.sh -status** command.
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 configuration guide for your switches.

Managing TLS Certificate, KeyStore, and TrustStore Files

About the TLS Certificate, KeyStore, and TrustStore Files

Note To support onePK devices, all connections to Cisco XNC that use onePK or OpenFlow agents require Transport Layer Security (TLS).

Enabling the TLS connections between Cisco XNC and the OpenFlow or onePK switches requires TLS KeyStore and TrustStore files. The TLS KeyStore and TLS TrustStore files are password protected.

Cisco Nexus 3000 and 3100 Series switches require additional credentials, including Private Key, Certificate, and Certificate Authority (CA).

- 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.

If TLS connections are required 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 (see Starting Cisco XNC with TLS Enabled , on page 16). After Cisco XNC is started with TLS, you must run the TLS KeyStore password configuration command (see Providing the TLS KeyStore and TrustStore Passwords, on page 16) to provide the passwords for Cisco XNC to unlock the KeyStore files.

Preparing to Generate the TLS Credentials

OpenFlow and Cisco onePK switches require cryptographic configuration to enable TLS.

Caution Self-signed certificates are appropriate only for testing in small deployments. For additional security, as well as more granular controls over individual certificate use and revocation, you should use certificates generated by your organization's Certificate Authority. In addition, you should never use the keys and certificates generated by this procedure in a production environment.

Before You Begin

Ensure that OpenSSL is installed on the Linux host where these steps will be performed.

Step 1 Create a TLS directory, and then navigate to it:
mkdir -p TLS

cd TLS

Step 2 Create three directories under `mypersonalca` and two prerequisite files:
mkdir -p mypersonalca/certs

mkdir -p mypersonalca/private

mkdir -p mypersonalca/crl

echo "01" > mypersonalca/serial

touch mypersonalca/index.txt

Step 3 Create the CA configuration file (ca.cnf).
The following is an example of the content of the ca.cnf file:

```
[ ca ]
default_ca = mypersonalca

[ mypersonalca ]
#
# WARNING: if you change that, change the default_keyfile in the [req] section below too
# Where everything is kept
dir = ./mypersonalca

# Where the issued certs are kept
certs = $dir/certs

# Where the issued crl are kept
crl_dir = $dir/crl

# database index file
database = $dir/index.txt

# default place for new certs
new_certs_dir = $dir/certs

#
# The CA certificate
certificate = $dir/certs/ca.pem

# The current serial number
serial = $dir/serial

# The current CRL
crl = $dir/crl/crl.pem

# WARNING: if you change that, change the default_keyfile in the [req] section below too
# The private key
private_key = $dir/private/ca.key

# private random number file
RANDFILE = $dir/private/.rand

# The extentions to add to the cert
x509_extensions = usr_cert

# how long to certify for
default_days = 365

# how long before next CRL
```

```
default_crl_days= 30

# which md to use; people in comments indicated to use sha1 here
default_md = sha1

# keep passed DN ordering
preserve = no

# Section names
policy = mypolicy
x509_extensions = certificate_extensions

[ mypolicy ]
# Use the supplied information
commonName = supplied
stateOrProvinceName = optional
countryName = optional
emailAddress = optional
organizationName = optional
organizationalUnitName = optional

[ certificate_extensions ]
# The signed certificate cannot be used as CA
basicConstraints = CA:false

[ req ]
# same as private_key
default_keyfile = ./mypersonalca/private/ca.key

# Which hash to use
default_md = sha1

# No prompts
prompt = no

# This is for CA
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer
string_mask = utf8only
basicConstraints = CA:true
distinguished_name = root_ca_distinguished_name
x509_extensions = root_ca_extensions

[ root_ca_distinguished_name ]
# EDIT THOSE
commonName = Controller
stateOrProvinceName = Mass
countryName = US
emailAddress = root_ca_userid@cisco.com
organizationName = Cisco

[ root_ca_extensions ]
basicConstraints = CA:true
```

What to Do Next

Create the TLS certificate file.

Creating the TLS Private Key, Certificate, and Certification Authority

Before You Begin

Complete the steps in Preparing to Generate the TLS Credentials, on page 9.

Step 1 Generate the TLS private key and Certification Authority (CA) files by entering the **openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -out mypersonalca/certs/ca.pem -outform PEM -keyout mypersonalca/private/ca.key** command.
This step generates the TLS private key in PEM format with a key length of 2048 bits, and the CA file.

Step 2 Generate the certificate key and certificate request files by entering the **openssl req -newkey rsa:2048 -keyout cert.key -keyform PEM -out cert.req -outform PEM** command.
This step generates the controller key (cert.key) and certificate request (cert.req) files in PEM format.

Note You must specify a common name in this step to complete Step 3. An example of a common name is the hostname of the server whereCisco XNC is running.

Step 3 Generate the certificate file by entering the **openssl ca -batch -notext -in cert.req -out cert.pem -config ca.cnf** command.
This step generates the certificate (cert.pem) file in PEM format using the certificate request (cert.req) and the certificate configuration (ca.cnf) files as inputs, and creates the certificates file (cert.pem) as output.
The following is an example of the console response:

```
Using configuration from ca.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AU'
stateOrProvinceName   :ASN.1 12:'Some-State'
organizationName      :ASN.1 12:'Internet Widgits Pty Ltd'
commonName            :ASN.1 12:'localhost'
```

What to Do Next

Generate and import the certificate files on your Cisco Nexus 3000 or 3100 Series switch.

Configuring the Cryptographic Keys on the Switch

Before You Begin

Create the TLS certificate.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **ip domain-name** *domain-name*	Configures the domain name for the switch.
Step 2	switch(config)# **crypto key generate rsa label myKey2 exportable modulus 2048**	Generates the cryptographic key.
Step 3	switch(config)# **crypto ca trustpoint myCA**	Enters the trustpoint configuration mode and installs the trustpoint file on the switch.
Step 4	switch(config-trustpoint)# **rsakeypair myKey2**	Installs the key files on the switch.
Step 5	switch(config-trustpoint)# **exit**	Exits trustpoint configuration mode.
Step 6	switch# **show crypto ca trustpoints**	(Optional) Verifies creation of the trustpoint files.
Step 7	switch# **show crypto key mypubkey rsa**	(Optional) Verifies creation of the key files.
Step 8	From the console, enter the **cat mypersonalca/certs/ca.pem** command.	Displays the certificate file on the machine hosting the generated TLS certificates.
Step 9	switch(config)# **crypto ca authenticate myCA**	Copies the CA certificate (`ca.pem`) to the switch to use as input. **Note** When copying the CA certificate, include the lines `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`. End the input with a line that contains only `END OF INPUT`.
Step 10	switch(config)# **crypto ca enroll myCA**	Generates the certificate request on the switch.
Step 11	From the console, enter the **openssl ca -in n3k-cert.req -out newcert.pem -config ./ca.cnf** command.	Copies the certificate request from the switch to the file `n3k-cert.req` on your Linux machine, and then uses it to generate the switch certificate.
Step 12	switch(config)# **crypto ca import myCA certificate**	Copies the certificate (`newcert.pem`) to the switch.
Step 13	From the console, enter the **cat newcert.pem** command.	Displays the certificate on the Linux console.
Step 14	switch# **show crypto ca certificates**	Displays the certificates on the switch.

What to Do Next

Enable TLS for Cisco onePK and OpenFlow switches.

Enabling TLS for onePK and OpenFlow Switches

Before You Begin

- Create the TLS certificate.
- Configure the cryptographic keys on the switch.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **onep**	Enters onePK configuration mode on the switch.
Step 2	switch(config-onep)# **transport type tls**	Enables TLS for onePK switches.
Step 3	switch# **exit**	Exits onePK configuration mode.
Step 4	switch# **show onep status**	(Optional) Displays the onePK configuration.
Step 5	switch(config)# **openflow**	Enters OpenFlow agent configuration mode on the switch.
Step 6	switch(config-ofa)# **switch 1**	Enters OpenFlow agent configuration mode for switch 1.
Step 7	switch(config-ofa)# **tls trust-point local myCA remote myCA**	Enables TLS certificate authority on the switch.
Step 8	switch(config-ofa-switch)# **pipeline** {*201*}	Configures the pipeline. Enter **201** for Cisco Nexus 3000 and 3100 Series switches.
Step 9	switch(config-ofa-switch)# **controller ipv4** {*A.B.C.D*} **port 6653 vrf management security tls**	Enables TLS for OpenFlow switches. *A.B.C.D* is the IP address of the controller. **Note** For more information about configuring TLS for OpenFlow (Cisco Nexus 3000 and 3100 Series switches), see the configuration guide for the switches in your environment.

What to Do Next

Create the TLS KeyStore file.

Creating the TLS KeyStore File

Note The TLS KeyStore file should be placed in the `configuration` directory of Cisco XNC.

Before You Begin

Complete the steps in Configuring the Cryptographic Keys on the Switch.

Step 1 Copy cert.key to xnc-privatekey.pem.
This command copies the cert.key file that was generated in the "Creating the TLS Private Key, Certificate, and Certificate Authority" section. This file contains the Cisco XNC private key.

Step 2 Copy `mpersponal/certs/cert.pem` to xnc-cert.pem.
This command makes a copy of the cert.pem file that was generated in the "Creating the TLS Private Key, Certificate, and Certificate Authority" section. This file contains the Cisco XNC certificate.

Step 3 Create the xnc.pem file, which contains the Cisco XNC private key and certificate, by entering the **cat xnc-privatekey.pem xnc-cert.pem > xnc.pem** command.

Step 4 Convert the PEM file xnc.pem to the PKCS#12 file xnc.p12 file by entering the **openssl pkcs12 -export -out xnc.p12 -in xnc.pem** command.

Step 5 Enter a password at the prompt.

Note The password must contain at least six characters, for example, cisco123. You must use the same password for this step and for Step 7.

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

Step 6 Convert the xnc.p12 to a Java KeyStore (tslTrustStore) file by entering the **keytool -importkeystore -srckeystore xnc.p12 -srcstoretype pkcs12 -destkeystore tlsKeyStore -deststoretype jks** command.
This command converts the xnc.p12 file to a password-protected tlsTrustStore file

Step 7 Enter a password at the prompt.

Note Use the same password that you entered in Step 5.

Creating the TLS TrustStore File

Note The TLS TrustStore file should be placed in the configuration directory of Cisco XNC.

Step 1 Copy the ca.pem file to sw-cacert.pem.

Step 2 Convert the sw-cacert.pem file to a Java TrustStore (tlsTrustStore) file by entering the **keytool -import -alias swca1 -file xnc-cert.pem -keystore tlsTrustStore** command.

Step 3 Enter a password at the prompt.
The xnc-cert.pem file is converted into a password-protected Java TrustStore (tlsTrustStore) file.

Note The password must be at least six characters long, for example, **cisco123**.

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

Starting Cisco XNC with TLS Enabled

Before You Begin

- Generate and import certificate files on the switches.
- Enable TLS on the OpenFlow or onePK switches.
- Create and deploy TLS KeyStore and TLS TrustStore files for the Cisco XNC application.
- Make sure that the TLS KeyStore (tlsKeyStore) and TLS TrustStore (tlsTrustStore) files are located in the `./configuration` directory.

From the console, start Cisco XNC by entering the **./runxnc.sh -tls -tlskeystore ./configuration/tlsKeyStore -tlstruststore ./configuration/tlsTrustStore** command.
Cisco XNC is started with TLS enabled.

Providing the TLS KeyStore and TrustStore Passwords

The TLS KeyStore and TrustStore passwords are sent to the Cisco Extensible Network Controller (XNC) so that it can read the password-protected TLS KeyStore and TrustStore files.

Step 1 Open a command window where you installed Cisco XNC.

Step 2 Navigate to the `xnc/bin` directory.

Step 3 Provide the TLS KeyStore and TLS TrustStore passwords by entering the **./xnc config-keystore-passwords** [**--user** {*user*} **--password** {*password*} **--url** {*url*} **--verbose --prompt --keystore-password** {*keystore_password*} **--truststore-password** {*truststore_password*}] command.

Step 4 Enter the following information:

- The Cisco XNC username {*user*}
- The Cisco XNC password {*password*}
- The Cisco XNC web URL {*url*}
- The TLS KeyStore password {*keystore_password*}
- The TLS TrustStore password {*truststore_password*}

Logging in to the Cisco XNC GUI

You must log into the Cisco XNC GUI using HTTPS.

- 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 Extensible Network Controller (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 same HA clustering configuration information in the `config.ini` files.
- All controllers must have the same information in the `xnc/configuration/startup` directory.

- If using cluster passwords, all controllers must have the same password configured in the `xncjgroups.xml` file. See Password Protecting the High Availability Clusters, on page 18.

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 Extensible Network Controller (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.

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, if 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 and 3100 Series Switches

The following configuration settings can improve scalability when connecting to Cisco Nexus 3000 or 3100 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	Predefined Value	Default Value	Recommended Value
of.messageResponseTimer	10000	2000	60000
of.switchLivenessTimeout	—	60500	120500
of.flowStatsPollInterval	120	10	240
of.portStatsPollInterval	300	5	240
of.descStatsPollInterval	—	60	240
of.barrierMessagePriorCount	50	100	50
of.discoveryInterval	—	300	300
of.discoveryTimeoutMultiple	—	2	2

Note Predefined values are the values that Cisco includes in the `config.ini` file that is shipped with Cisco XNC. A em dash ("—") in this column of the table means that unless you explicitly update the value, the default value will be used.

Step 4 Save the file and exit the editor.

Step 5 Restart Cisco XNC.

Backing Up or Restoring Cisco Extensible Network Controller

The backup and restore commands allow you to back up your Cisco Extensible Network Controller (XNC) configurations and restore them.

Step 1 Open a command window where you installed Cisco XNC.

Step 2 Navigate to the `xnc/bin` directory that was created when you installed the software.

Step 3 Back up the configuration by entering the **./xnc config --backup** command.
The **--backup** option creates a backup archive (in .zip format) of the startup configuration in the current `xnc` distribution. The backup archive is stored in `{xncHome}/backup/`. A new archive is created each time that the backup command is entered using a filename with the current timestamp.

Step 4 Restore the configuration by entering the **./xnc config --restore --backupfile** {*zip_filename*} command.
The **--restore** option restores the startup configuration of the current `xnc` distribution from an existing backup archive. The restore action requires the absolute path of the backup archive.

Step 5 If you are restoring a configuration, stop and restart Cisco XNC for the restored configuration to take effect.

Recovering the Administrative Password

The Cisco Extensible Network Controller (XNC) network administrator user can return the administrative password to the factory default.

Note The controller may or may not be running when this command is used. If the controller is not running, the password reset takes effect the next time that it is run.

Step 1 Open a command window where you installed Cisco XNC.

Step 2 Navigate to the `xnc/bin` directory that was created when you installed the software.

Step 3 Reset the administrative password by entering the **./xnc reset-admin-password** [**--wait-seconds** {*wait_time*} **--password** {*password*}] command.
Resets the admin password to the default or specified password by restarting the user manager.

- The **wait-seconds** is the length of time, in seconds, to wait for the user manager to restart. The minimum is 5 seconds and the maximum is 60 seconds.
- The **password** is the administrative password.

Note

- The password must be from 8 to 256 characters, contain both uppercase and lowercase characters, and have at least one number and one nonalphanumeric character.
- If you leave the password blank, it is reset to the factory default of "admin".
- Each time that you reset the administrative password, make sure that the new password meets these requirements or you will not be able to log in to Cisco XNC.

Uninstalling the Cisco Extensible Network Controller Application

Before You Begin

Ensure that your Cisco Extensible Network Controller (XNC) application is stopped before proceeding.

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.