Deploying Cisco XNC

This chapter contains the following sections:

Installing Cisco XNC

Installing the Cisco XNC 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:

- `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.
- `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.
- `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 GUI configurations are saved.
- `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.

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 **-start** option.
-jmx	Enables JMX remote access on the Cisco XNC JVM, which can be used to troubleshoot performance issues.
-jmxport *num*	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.
-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_num*	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 XNC user with the network-administrator role.
-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 script.
-tls	Enables TLS secure connections between Cisco XNC and OpenFlow switches. To enable TLS, start the controller with the following options: **./runxnc.sh -tls -tlskeystore** *keystore_file_location* **-tlstruststore** *truststore_file_location*

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 switches require 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 OpenFlow switches' certificates.

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 7.

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 allows the factory default password for the Cisco XNC Network Administrator user.

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 password.

Uninstalling the Cisco XNC Application

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

Step 2 Delete the `CiscoXNC` directory.