NiFi Authentication

Date of Publish: 2020-12-15

https://docs.cloudera.com/

Contents

NiFi Authentication

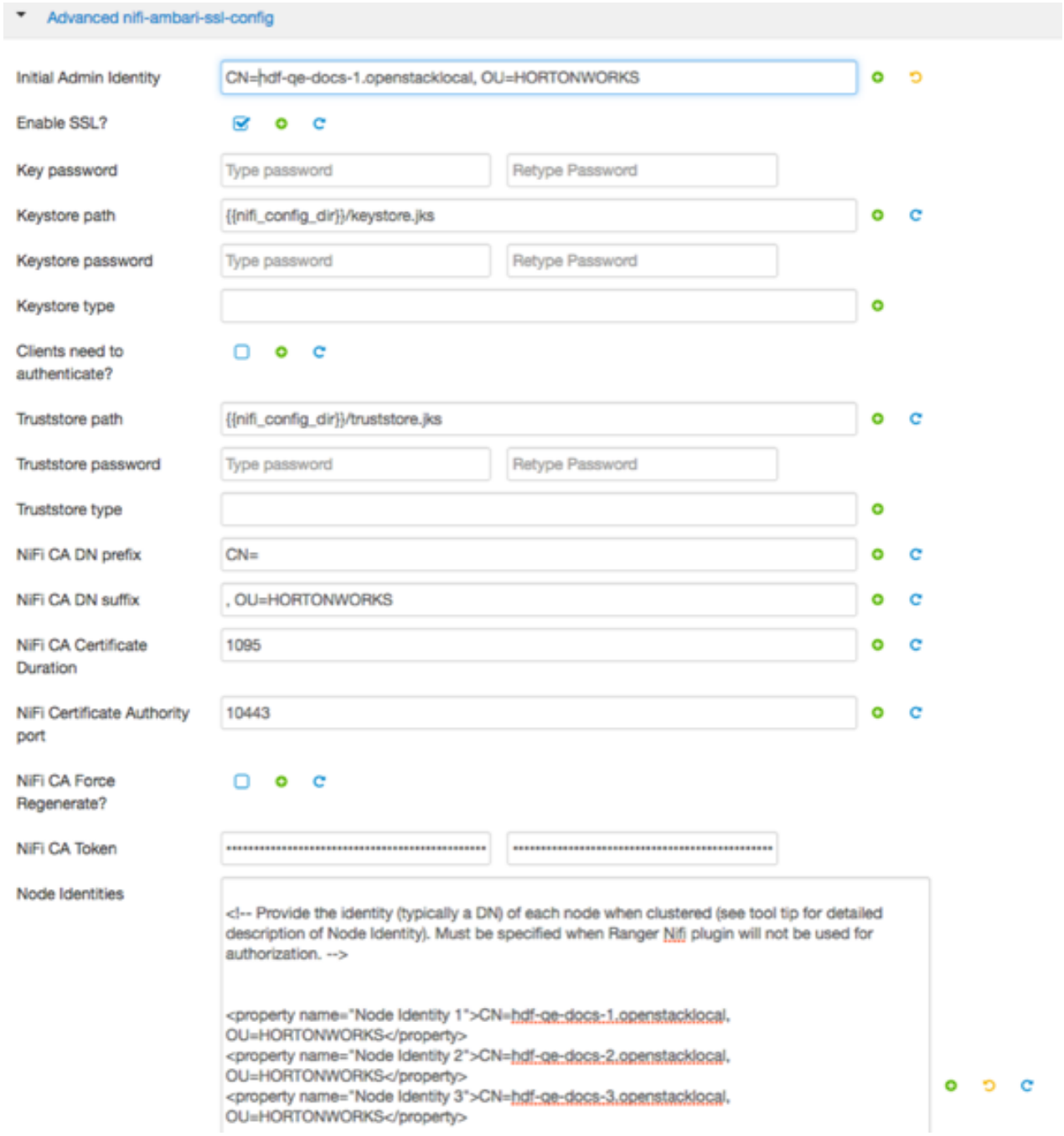
After you have installed Ambari and the HDF stack, you have a 2 options for enabling SSL for your NiFi services.

- Enabling SSL with the NiFi CA
- Enabling SSL with existing certificates

You can use the NiFi service Configs tab Advanced nifi-ambari-ssl-config dialog to configure security for these options.

To access the NiFi SSL configuration dialog:

1. From the Ambari services column, click NiFi.
2. Click the Configs tabs.
3. Click Advanced nifi-ambari-ssl-config.



Enabling SSL with a NiFi Certificate Authority

When you enable SSL with the NiFi Certificate Authority (CA) installed, the NiFi CA generates new client certificates for you through Ambari. If you want to enable SSL with a NiFi CA installed, and are planning to use Ranger to manage authorization:

1. Select the **Enable SSL?** box.
2. Specify the **NiFi CA** token.

If you want to enable SSL with a NiFi CA installed, and are not yet using Ranger to manage authorization:

1. Check the **Enable SSL?** box.
2. Specify the **NiFi CA Token**.

3. Verify that the authorizations.xml file on each node does not contain policies. The authorizations.xml is located in {nifi_internal_dir}/conf. By default, this location is /var/lib/nifi/conf/, and the value of {nifi_internal_dir} is specified in the **NiFi internal dir** field under **Advanced nifi-ambari-config**.

   Note:

   If authorizations.xml does contain policies, you must delete it from each node. If you do not, your **Initial Admin Identity** and **Node Identities** changes do not take effect.
4. Specify the **Initial Admin Identity**. The **Initial Admin Identity** is the identity of an initial administrator and is granted access to the UI and has the ability to create additional users, groups, and policies. This is a required value when you are not using the Ranger plugin for NiFi for authorization.

   The **Initial Admin Identity** format is CN=admin, OU=NIFI.

   After you have added the **Initial Admin Identity**, you must immediately generate certificate for this user.
5. Specify the **Node Identities**. This indicates the identity of each node in a NiFi cluster and allows clustered nodes to communicate. This is a required value when you are not using the Ranger plugin for NiFi for authorization.

```
<property name="Node Identity 1">CN=node1.fqdn, OU=NIFI</property>
<property name="Node Identity 2">CN=node2.fqdn, OU=NIFI</property>
<property name="Node Identity 3">CN=node3.fqdn, OU=NIFI</property>
```

   Replace node1.fqdn, node2.fqdn, and node3.fqdn with their respective fully qualified domain names.

Enabling SSL with Existing Certificates

If you want to enable SSL with existing certificates, and plan to use Ranger for authorization:

1. Check the **Enable SSL?** box.
2. Set **Keystore path**, **Keystore password**, and **Keystore type** values.

   The keystore path is similar to: /etc/security/nifi-certs/keystore.jks
3. Set the **Truststore path**, **Truststore password**, and **Truststore type** values.

   The truststore path is similar to: /etc/security/nifi-certs/truststore.jks
4. Check **Clients need to authenticate?** if you want to ensure that nodes in the cluster are authenticated and are required to have certificates that are trusted by the truststores.

If you want to enable SSL with existing certificates, and are not yet using Ranger for authorization:

1. Check the **Enable SSL?** box.
2. Set **Keystore path**, **Keystore password**, and **Keystore type** values.

   The keystore path is similar to: /etc/security/nifi-certs/keystore.jks
3. Set the **Truststore path**, **Truststore password**, and **Truststore type** values.

   The truststore path is similar to: /etc/security/nifi-certs/truststore.jks
4. Check **Clients need to authenticate?** to ensure that nodes in the cluster are authenticated and are required to have certificates that are trusted by the Truststores.
5. Specify the **Initial Admin Identity**. The **Initial Admin Identity** is the identity of an initial administrator and is granted access to the UI and has the ability to create additional users, groups, and policies. This is a required value when you are not using the Ranger plugin for NiFi for authorization.

   The **Initial Admin Identity** format is CN=admin, OU=NIFI.

   After you have added the **Initial Admin Identity**, you must immediately generate certificate for this user.
6. Specify the **Node Identities**. This indicates the identity of each node in a NiFi cluster and allows clustered nodes to communicate. This is a required value when you are not using the Ranger plugin for NiFi for authorization.

```
<property name="Node Identity 1">CN=node1.fqdn, OU=NIFI</property>
<property name="Node Identity 2">CN=node2.fqdn, OU=NIFI</property>
```

```
<property name="Node Identity 3">CN=node3.fqdn, OU=NIFI</property>
```

Replace node1.fqdn, node2.fqdn, and node3.fqdn with their respective fully qualified domain names.

(Optional) Setting Up Identity Mapping

You can use identity mapping properties to normalize user identities. Once you set up identity mapping, NiFi treats identities authenticated by different identity providers (certificates, LDAP, Kerberos) the same. This allows you to avoid creating duplicate users. Additionally, you only need to set up user-specific configurations such as authorizations once per user.

1. From the NiFi service **Configs** tab, click **Advanced nifi-properties**.
2. Use the Filter box to search for nifi.security.identity.mapping.pattern.
3. Enter the following values:

Table 1: Identity mapping values

Field	Sample value
nifi.security.identity.mapping.pattern.dn	^CN=(.*?), OU=(.*?)$
nifi.security.identity.mapping.value.dn	$1@$2
nifi.security.identity.mapping.pattern.kerb	^(.*?)/instance@(.*?)$
nifi.security.identity.mapping.value.kerb	$1@$2

4. Click **Save**.
5. Restart NiFi using the **Restart all Required** option from the **Action** menu.

Example

The following examples demonstrate normalizing DNs from certificates and principals from Kerberos:

```
nifi.security.identity.mapping.pattern.dn=^CN=(.*?), OU=(.*?), O=(.*?),
 L=(.*?), ST=(.*?), C=(.*?)$
nifi.security.identity.mapping.value.dn=$1@$2
nifi.security.identity.mapping.pattern.kerb=^(.*?)/instance@(.*?)$
nifi.security.identity.mapping.value.kerb=$1@$2
```

Generating Client Certificates

If you are using a CA, you can use the TLS Toolkit provided in the HDF management pack to generate the required client certificates so that you can log into NiFi after enabling SSL.

1. Navigate the TLS Toolkit directory, which will be similar to:

```
cd /var/lib/ambari-agent/cache/common-services/NIFI/1.0.0/package/files/
nifi-toolkit-$version
```

For example:

```
cd /var/lib/ambari-agent/cache/common-services/NIFI/1.0.0/package/files/
nifi-toolkit-1.1.0.2.1.3.0-6
```

2. From the command line, run the following:

```
bin/tls-toolkit.sh client
-c <CA host name>
```

```
-D "<distinguished name>"
-p <CA host port>
-t <NiFi CA token>
-T <keystore type>
```

Your command should look similar to:

```
bin/tls-toolkit.sh client
-c nifi.cert.authority.example.com
-D "CN=admin, OU=NIFI"
-t nifi
-p 10443
-T pkcs12
```

3. To get your keystore password, enter:

```
cat config.json
```

4. Verify that the installation directory contains the following two files:
   - keystore.pkcs12
   - nifi-cert.pem
5. To double-click your keystore file to launch your OS certificate management application, change keystore.pkcs12 to keystore.p12.
6. Import the nifi-cert.pem file as your trusted CA.
7. Import keystore.pkcs12 as the client certificate.

Re-running the TLS Toolkit generates a new set of keystore and configuration files. To avoid having your files overwritten, save the keystore and configuration files to an alternate location before re-running the TLS Toolkit.

Logging into NiFi After Enabling SSL

Now that you have set up SSL, you need to enable logging into NiFi with a certificate:

1. Launch NiFi from the Ambari **Quick Links** menu.
2. Select the certificate you just imported from the browser prompt.
3. Log in with the user name and password you created during installation.

Note:

When you are running NiFi on a host with Ambari and with SSL enabled, the default URL becomes secured https://<local-host>:9091/nifi.