Configuring Ambari Authentication with LDAP/AD

Date of Publish: 2018-07-15

http://docs.hortonworks.com

Contents

Configuring Ambari Authentication for LDAP/AD

By default Ambari uses an internal database as the user store for authentication and authorization. If you want to configure LDAP or Active Directory (AD) external authentication, you must configure Ambari to authenticate external users, configure Ambari to use an LDAP/AD datastore, and synchronize your LDAP users and groups.

Recommended datastores for users are Active Directory (AD), FreeIPA, and Open LDAP.

Related Information

Configuring Ambari to authenticate external users

Preparing for LDAPS integration

Synchronize LDAP Users and Groups

LDAP Authentication and Authorization Testing

Configuring Ambari to authenticate external users

By default, Ambari uses an internal database as the user store for authentication and authorization. You can configure Ambari to authenticate external users stored in LDAP, Active Directory (AD), or FreeIPA datastores.

About this task

For each case, you must run the ambari-server setup-ldap command line utility on the Ambari host, and be prepared to provide information for each prompt described in the following table. Generally, you will run this cli utility and provide values appropriate for your environment and external user datastore. The wizard sets default configuration values appropriate for the LDAP type you select. You must then customize the default configuration values to optimize or tune your environment.

Procedure

1. On the Ambari Server host, open ambari-server setup-ldap with a command line editor.
2. Respond to each prompt.

   Prompts marked with an asterisk (*) are required values.

Prompt	Description
Please select the type of LDAP you want to use *	Selecting a datastore type for the LDAP integration with helps Ambari provide the most appropriate default configuration values for the upcoming, default configurations. Supported options are AD, IPA and “Generic LDAP”.
Primary URL Host*	The fully qualified hostname for the LDAP server.
Primary URL Port*	The port for the LDAP server. By default, secured LDAPs runs on port 636. Unsecured LDAP runs on port 389.
Secondary URL Host	The hostname for an additional LDAP server, which should be a replica of your primary. This value is optional.
Secondary URL Port	The port for a secondary LDAP server. This value is optional.
Use SSL*	Set to true to connect to your LDAP server over a secured connection. A secured connection requires

Prompt	Description
	your LDAP server to present a valid certificate from a trusted CA to the Ambari server.
Do you want to provide custom TrustStore for Ambari [y/n]	If your LDAP server certificate was signed by a well-known CA, you can rely on the default java truststore to contain the public certs of those CAs. Otherwise, you must explicitly add the public certificate of the CA that signed the LDAP server's certificate to the default java truststore on the Ambari host. Alternatively, create a custom truststore, and then use this option to configure Ambari to use it.
TrustStore type	Format of the truststore [jks/jceks/pkcs12]
Path to TrustStore	Path on the Ambari host where you placed the custom truststore that Ambari should use.
Password for TrustStore	Password for the custom truststore. Default is changeit.
User object class*	The object class you define for users.
User name attribute*	The attribute you define for username.
Group object class*	The object class you define for groups.
Group name attribute*	The attribute you define for group name.
Group member attribute*	The attribute you define for group membership.
Distinguished name attribute*	The attribute you define for the distinguished name.
Search Base*	The root search base in the directory for both users and groups.
Referral method*	Enter follow or ignore for your LDAP referrals.
Bind anonymously*	If true, bind to the LDAP server anonymously. If false, bind to the LDAP server non-anonymously.
Bind DN*:	If you set Bind anonymously to false, enter the Distinguished Name ("DN") for the LDAP service account that can be used to search. This account should exist in LDAP and have sufficient privilege to search the directory tree, but does not require any administrative or login privileges.
Bind DN Password*:	Enter the password for your LDAP manager DN. If this password eventually expires or gets changed in the LDAP server, it should be updated here also.
Handling behavior for username collisions*:	When Ambari finds duplicate user accounts, what strategy should Ambari use. (convert/skip).
	Skip — Avoids importing any users from ldap that already exist as local users.
	Convert — Merges the lists of local and ldap user credentials. In this case, the default username "admin" allows login with either the local

Prompt	Description
	password or the ldap password. Further, the Convert option prevents managing the local users overlapped with ldap users.
	To avoid overlapping local users with ldap users, choose the SKIP option. If you choose the Convert option, consider changing the default password for potentially overlapped local users, before running ldap-sync.
Force lower-case user names	Standardizes all username characters on lowercase, such that Admin==admin.
Results from LDAP are paginated when requested	Determines when pagination should be used when reading responses to ldapsearch operations. Generally, we recommend pagination for large Active Directory trees. You should ensure that your LDAP implementations support pagination.
Disable endpoint identification during SSL handshake.	This option is available in Ambari 2.7.3+ and can be used to bypass the newer java requirements that the certificate of the LDAP server contains the IP of the server as a SAN. Equivalent of passing in this flag on jvm startup options: -Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true

Preparing for LDAPS integration

If you are using LDAPs, the certificate authority that signed the certificate for your LDAP server must be present in the truststore used by Ambari.

About this task

If the LDAP server has has a certificate signed by a "well known" CA, no further action is needed as the default Java truststore contains a list of public CAs. If you are using an organizational CA or self-signed certificate, there are two ways of meeting this requirement:

A) Tell Ambari to use a custom truststore that already contains the certificate of the CA that signed the LDAP host certificate. The ambari-server setup-ldap cli utility provides options that support secure and custom truststores, but the custom truststore must be created in advance and available for Ambari to use. The ambari-server setup-ldap cli utility provides options that support secure and custom truststores, but the custom truststore must be created in advance and available for Ambari to use.

B) Import the public certificate of the CA that signed the LDAP host certificate into the default Java truststore. ($JAVA_HOME/jre/lib/security/cacerts) This option may be less secure if the LDAP server uses a self-signed certificate that will become a trusted CA by all processes running on the Ambari host. In addition, since the default Java truststore is tied to the specific version of Java, updating the Java version will require the CA cert to be reconfigured into the newer Java's truststore.

Note:

The truststore information is still stored in the ambari.properties file, and not the ambari database along with the remaining LDAP settings. Configuring a custom truststore of modifying the existing truststore requires a restart of the Ambari server, for the settings to take effect.

Before you begin

Obtain the public certificate of the CA that signed the LDAP server certificate, and choose one of the paths below depending on your truststore management strategy.

Path A - Use a Custom Truststore

- If you are using Active Directory as your LDAP provider, obtain the public certificate of the CA that signed the AD certificate and create a new truststore to import the CA cert (or the ldap host if self-signed) into.

  If necessary, convert the SSL certificate to X.509 format:

  ```
  openssl x509 -in ad-ca.pem -out ad-ca.crt

  $JAVA_HOME/bin/keytool -import -trustcacerts -alias root -file
  $PATH_TO_YOUR_LDAPS_CERT -keystore /etc/security/ldaps-truststore.jks
  ```

  when prompted, enter a password. you will use this during setup.
- If you are using FreeIPA as your LDAP provider and have registered the ipa-client on the Ambari host with the same IPA instance, a preconfigured truststore that contains the “well-known” CAs alongside IPA’s CA public cert should exist in /etc/pki/java/cacerts You can verify this by listing the contents of this file.

  ```
  $JAVA_HOME/bin/keytool -list -keystore /etc/pki/java/cacerts -storepass
   changeit |
  grep ipaca
  ```

Path B - Import to default Java truststore

You can Import an SSL certificate to the existing keystore, such as the default jre certificates store, by typing the following command after setting your JAVA_HOME:

If necessary, convert the SSL certificate to X.509 format:

```
openssl x509 -in slapd.pem -out slapd.crt

$JAVA_HOME/bin/keytool -import -trustcacerts -file slapd.crt -keystore
$JAVA_HOME/jre/lib/security/cacerts
```

Note: Be sure to restart Ambari server to have it pick up the modified truststore.

Procedure

1. On the Ambari Server host, run ambari-server setup-ldap and respond to each prompt.
2. If you set Use SSL* = true, the following prompt appears: Do you want to provide custom TrustStore for Ambari?:
3. If you are using IPA and have installed the ipa-client and registered the Ambari host with IPA, type y.

   When you select this option, enter:

   - At the TrustStore type prompt, enter jks.
   - At the Path to TrustStore file prompt, enter /etc/pki/java/cacerts
   - At the Password for TrustStore prompt, type changeit, unless you changed it, in which case you should provide the current password.
4. If you AD/LDAP and have precreated a custom truststore using the steps above, type y.

When you select this option, enter:

- At the TrustStore type prompt, enter jks.
- At the Path to TrustStore file> prompt, enter /etc/security/ldaps-truststore.jks.

At the Password for TrustStore prompt, type the password that you defined for the keystore.

5. Review your settings and if they are correct, select y.
6. Start or restart the Ambari server.
   ambari-server restart

Active Directory LDAP setup example

If the users for whom you want to enable authentication into Ambari UI are stored in Active Directory, you should configure Ambari to integrate directly against your AD instance. Selecting AD as an LDAP type helps the wizard configure some smarter defaults for the the attribute values that tend to work in most AD instances.

About this task

Gather details about your AD instance from your AD administrator and provide them as input to the ambari-server setup-ldap cli wizard. Verify the settings before you confirm them as AD instances can be configured in many ways.

To configure LDAP integration against AD using the cli wizard:

Procedure

1. Run ambari-server setup-ldap on the Ambari server host.
2. Provide the following information about your domain.

Prompt	Example value for AD
Please select the type of LDAP you want to use :	AD
Primary URL Host*	ad.hortonworks.site
Primary URL Port	636
Secondary URL Host (optional)	
Secondary URL Port (optional)	
Use SSL*	true
Do you want to provide custom TrustStore for Ambari [y/n]	n
TrustStore type	jks
Path to TrustStore	
Password for TrustStore	
User object class	user
User name attribute*	sAMAccountName
Group object class*	group
Group name attribute*	cn
Group member attribute*	member
Distinguished name attribute*	distinguishedName
Search Base	CN=Users,dc=hortonworks,dc=site

Prompt	Example value for AD
Referral method*	follow
Bind anonymously*	false
Bind DN:	CN=ldapbind,CN=Users,dc=hortonworks,dc=site
Bind DN Password:	
Handling behavior for username collisions:	convert
Force lower-case user names	true
Results from LDAP are paginated when requested	true

3. Verify your default settings.

What to do next
Synchronize your LDAP users and groups.

FreeIPA LDAP setup example

If the users for whom you want to enable authentication into Ambari UI are stored in FreeIPA, you should configure Ambari to integrate directly against your IPA instance. Selecting IPA as an LDAP type helps the wizard configure some smarter defaults for the the attribute values that tend to work in most IPA instances.

About this task
Gather details about your FreeIPA instance from your IPA administrator (or use the Tips below) and provide them as input to the cli wizard. Be sure to provide your own searchbase, and verify the attribute settings before confirming.

To configure LDAP integration against IPA using the cli wizard:

Procedure
1. Run ambari-server setup-ldap on the Ambari server host.
2. Provide the following information about your domain.

Prompt	Example value for IPA
Please select the type of LDAP you want to use :	IPA
Primary URL Host*	ipa.hortonworks.site
Primary URL Port	636
Secondary URL Host (optional)	
Secondary URL Port (optional)	
Use SSL*	true
Do you want to provide custom TrustStore for Ambari [y/n]	y
TrustStore type	jks
Path to TrustStore	/etc/pki/java/cacerts
Password for TrustStore	changeit
User object class	posixaccount
User name attribute*	uid
Group object class*	posixGroup

Prompt	Example value for IPA
Group name attribute*	cn
Group member attribute*	member
Distinguished name attribute*	dn
Search Base	cn=accounts,dc=hortonworks,dc=site
Referral method*	follow
Bind anonymously*	true
Bind DN:	uid=ldapbind,cn=users,cn=accounts,dc=hortonworks,dc=site
Bind DN Password:	
Handling behavior for username collisions:	convert
Force lower-case user names	true
Results from LDAP are paginated when requested	false

3. 

   Note:

   The truststore configuration can leverage the IPA CA created during ipa-client installation at /etc/pki/cacerts/java.

   See Choosing options during ambari-server setup-ldap for more details.

   Restart Ambari Server.

   A restart is required before Ambari can leverage the custom truststore.
4. Verify your default settings.

Example

FreeIPA Tips for determining LDAP Search Properties:

- IPA Clients contain /etc/ipa/default.conf with various LDAP server properties:

```
[root@demo ~]# cat /etc/ipa/default.conf
  basedn = dc=hortonworks,dc=site
  realm = HORTONWORKS.SITE
  domain = hortonworks.site
  server = ipa.hortonworks.site
```

- Determining valid user attributes (posixaccount, uid, etc): ipa user-show hadoopadmin --raw --all
- Determining valid group attributes (posixgroup, member, memberUid, etc): ipa group-show admins --raw --all
- Verifying ldapbind account and search base using ldapsearch

```
[root@demo ~]# yum install -y openldap-clients

# Test ldap bind properties
AM_LDAP_SEARCHBASE="cn=accounts,dc=hortonworks,dc=site"
AM_LDAP_BINDDN="uid=ldapbind,cn=users,cn=accounts,dc=hortonworks,dc=site"
AM_LDAP_BINDDN_PW="BadPass#1"
AM_LDAP_URL=ldaps://ipa.hortonworks.com:636

# Search for a valid uid and ensure the searchbase, bind dn, and ldapurl
 resolve properly
[root@demo ~]# ldapsearch -D ${AM_LDAP_BINDDN} \
-w ${AM_LDAP_BINDDN_PW} \
-b ${AM_LDAP_SEARCHBASE} \
-H ${AM_LDAP_URL} uid=hadoopadmin

# Tail results of a valid ldapsearch for a single uid:
```

```
numResponses: 2
numEntries: 1
```

Example

Example configuring LDAP integration against IPA:

Using interactive CLI:

```
[root@demo certificates]# ambari-server setup-ldap
Currently 'no auth method' is configured, do you wish to use LDAP instead
 [y/n] (y)?
Please select the type of LDAP you want to use (AD, IPA, Generic LDAP):IPA
Primary LDAP Host (ipa.ambari.apache.org): ipa.hortonworks.com
Primary LDAP Port (636):
Secondary LDAP Host <Optional>:
Secondary LDAP Port <Optional>:
Use SSL [true/false] (true):
Do you want to provide custom TrustStore for Ambari [y/n] (y)?
TrustStore type [jks/jceks/pkcs12] (jks):
Path to TrustStore file (/etc/pki/java/cacerts):
Password for TrustStore:
Re-enter password:
User object class (posixUser):posixaccount
User ID attribute (uid):
Group object class (posixGroup):
Group name attribute (cn):
Group member attribute (memberUid):member
 Distinguished name attribute (dn):
Search Base (dc=ambari,dc=apache,dc=org): cn=accounts,dc=hortonworks,dc=site
Referral method [follow/ignore] (follow):
Bind anonymously [true/false] (false):
Bind DN
 (uid=ldapbind,cn=users,cn=accounts,dc=ambari,dc=apache,dc=org): uid=ldapbind,cn=users,c
Enter Bind DN Password:
Confirm Bind DN Password:
Handling behavior for username collisions [convert/skip] for LDAP sync
 (skip):
Force lower-case user names [true/false]:
Results from LDAP are paginated when requested [true/false]:
```

Note:

In Ambari 2.7.1, the User Object Class and Group Object Class defaults of the IPA defaults must be overwritten.

Using non-interactive CLI:

```
ambari-server setup-ldap \
--ldap-url=ipa.hortonworks.site:636  \
--ldap-user-class=posixAccount \
--ldap-user-attr=uid \
--ldap-group-class=posixGroup \
--ldap-ssl=true \
--ldap-referral="follow" \
--ldap-group-attr=cn \
--ldap-member-attr=member \
--ldap-dn=dn \
--ldap-base-dn=cn=accounts,dc=hortonworks,dc=site \
--ldap-bind-anonym=false \
--ldap-manager-dn=uid=ldapbind,cn=users,cn=accounts,dc=hortonworks,dc=site \
--ldap-manager-password=BadPass#1 \
--ldap-save-settings \
--ldap-sync-username-collisions-behavior=convert  \
```

```
--ldap-force-setup \
--ldap-force-lowercase-usernames=true \
--ldap-pagination-enabled=false \
--ambari-admin-username=admin \
--ambari-admin-password=adminpassword \
--truststore-type=jks \
--truststore-path=/etc/pki/java/cacerts \
--truststore-password=changeit \
--ldap-secondary-host="" \
--ldap-secondary-port=0 \
--ldap-sync-disable-endpoint-identification=true
```

Note:

In Ambari 2.7.1, the ldap-type can must be passed in interactively.

The flag to disable endpoint identification is only available in Ambari 2.7.3 and greater versions.

What to do next

Synchronize your LDAP users and groups.

Related Information

Preparing for LDAPS integration

Generic, Open LDAP setup example

If the users for whom you want to enable authentication into Ambari UI are stored in LDAP, you should configure Ambari to integrate directly against your LDAP instance. Selecting Generic LDAP as an LDAP type helps the wizard configure some smarter defaults for the the attribute values that tend to work in most OpenLDAP instances.

About this task

Gather details about your OpenLDAP instance from your LDAP administrator and provide them as input to the cli wizard. Verify the settings before you confirm them as these instances can be configured in various ways.

To configure LDAP integration against generic LDAP using the cli wizard:

Procedure

1. Run ambari-server setup-ldap on the Ambari server host.
2. Provide the following information about your domain.

Prompt	Example value for OpenLDAP
Please select the type of LDAP you want to use :	Generic
Primary URL Host*	openldap.hortonworks.site
Primary URL Port	389
Secondary URL Host (optional)	
Secondary URL Port (optional)	
Use SSL*	false
Do you want to provide custom TrustStore for Ambari [y/n]	n
TrustStore type	
Path to TrustStore	
Password for TrustStore	

Prompt	Example value for OpenLDAP
User object class	organizationalPerson
User name attribute*	uid
Group object class*	groupOfNames
Group name attribute*	cn
Group member attribute*	uniquemember
Distinguished name attribute*	
Search Base	ou=people,dc=hortonworks,dc=site
Referral method*	follow
Bind anonymously*	false
Bind DN:	uid=ldapbind,ou=people,dc=hortonworks,dc=site
Bind DN Password:	
Handling behavior for username collisions:	convert
Force lower-case user names	true
Results from LDAP are paginated when requeste	false

3. Verify your default settings.

What to do next
Synchronize your LDAP users and groups.

Synchronize LDAP Users and Groups

After setting up your LDAP integration, you must synchronize LDAP users and groups with Ambari, using the .ambari-server sync-ldap [option] utility.

About this task
The ambari-server sync-ldap [option] utility provides three options for synchronization:

- Specific set of users and groups
- Synchronize the existing users and groups in Ambari with LDAP
- All users and groups

Procedure
Run ambari-server sync-ldap [option] and answer the prompts to initiate the sync.

Option	Description
--users users.txt --groups groups.txt	Specific Set of Users and Groups Use this option to synchronize a specific set of users and groups from LDAP into Ambari. Provide the command a text file of comma-separated users and groups. The comma separated entries in each of these files should be based off of the values in LDAP of the attributes chosen during setup. The "User name attribute" should be used for the users.txt file, and the "Group name attribute" should be used for the groups.txt file. This command

Option	Description
	will find, import, and synchronize the matching LDAP entities with Ambari.
--existing	Existing users and groups After you have performed a synchronization of a specific set of users and groups (above), you use this option to synchronize only those entities that are in Ambari with LDAP. Users will be removed from Ambari if they no longer exist in LDAP, and group membership in Ambari will be updated to match LDAP.
--all	All users and groups Only use this option if you are sure you want to synchronize all users and groups from LDAP into Ambari. If you only want to synchronize a subset of users and groups, use a specific set of users and groups option. This will import all entities with matching LDAP user and group object classes into Ambari.

The users you have just imported are initially granted the Ambari User privilege. Ambari Users can read metrics, view service status and configuration, and browse job information. For these new users to be able to start or stop services, modify configurations, and run smoke tests, they must be Admins. To make this change, as an Ambari Admin, use **Manage Ambari** > **Users** > **Edit**. For instructions, see Modify access levels for users and groups.

Example

Example output after synchronizing LDAP users and groups in Ambari.

```
[root@demo ~]# ambari-server sync-ldap --all
Using python  /usr/bin/python
Syncing with LDAP...
Enter Ambari Admin login: admin
Enter Ambari Admin password:

Fetching LDAP configuration from DB.
Syncing all...

Completed LDAP Sync.
Summary:
  memberships:
    removed = 0
    created = 16
  users:
    skipped = 1
    removed = 0
    updated = 0
    created = 15
  groups:
    updated = 0
    removed = 0
    created = 26

Ambari Server 'sync-ldap' completed successfully.
```

What to do next

Review log files for failed synchronization attempts, at /var/log/ambari-server/ambari-server.log on the Ambari Server host.

Related Information

Modify access levels for users and groups

LDAP Authentication and Authorization Testing

You can use Ambari **Ambari Web** > **Manage Users and Groups** to check whether users and groups appear as expected.

About this task

After you have saved your ldap configurations and synchronized users successfully, verify that the integration worked using the following steps:

Procedure

1. In **Ambari Web** > **Manage Ambari** > **Users**, verify that you can see all imported users, grouped correctly.
2. In **Ambari Web** > **Manage Ambari** > **Groups**, click a group name to assign access roles to imported groups or users.
3. Verify that your imported users can log in with their ldap credentials and perform role-appropirate tasks.

What to do next

If your user integration does not appear as expected, review logs at /var/log/ambari-server/ambari-server.log and review your LDAP configuration settings.

Related Information

Modify access levels for users and groups