Installing DataPlane

Date of Publish: 2018-05-18

http://docs.hortonworks.com

Contents

DP Platform support requirements

You should review the support requirements for the DP Platform to ensure your environment meets those requirements. Additionally, you must consider various aspects of your HDP clusters and prepare those clusters as part of your DataPlane installation in order to register the clusters with DataPlane.

Important:

The specific DP Apps you plan to install into your environment might bring additional requirements. Review the App-specific documentation thoroughly to ensure you can meet the App-specific requirements. For example, depending on your choice of Apps, your cluster requirements might change. This includes (but is not limited to) a minimal HDP version, setup and configuration of Knox, and other cluster requirements.

Support Matrix information

You can find the most current information about interoperability for this release on the Support Matrix. The Support Matrix tool provides information about:

- Operating Systems
- Databases
- Browsers
- JDKs

To access the tool, go to: https://supportmatrix.hortonworks.com.

DP Platform Host requirements

DP Platform must be installed on a dedicated host that is not part of an existing HDP cluster, to prevent potential port conflicts with other HDP services.

DataPlane is certified for use with specific versions of CentOS and RHEL. These operating systems include support for Docker 1.12 or higher.

Table 1: Requirements for the DP Platform host

Item	Versions
Container infrastructure	Docker 1.12.x or higher
Processing and Memory Requirements	• Multicore processor, with minimum 8 cores • Minimum 16 GB RAM
Software	• yum, rpm • wget • tar • bash shell
Authentication	Existing LDAP or Active Directory (AD)
Environment	Disable SELinux
Ports	• 443 (Used by DataPlane for SSL access) • 80 (Redirected to port 443 for SSL) • 8500 (Used by Consul which handles the Docker container networking) When configuring the software repositories, to avoid a potential port conflict, be sure you either use a distinct host or do not host the local repositories on port 80.

SmartSense requirements

A SmartSense ID is required to enable any service app on DP Platform. You can retrieve the SmartSense ID from the Hortonworks Support Portal, under the Tools tab.

Related Tasks

General requirements for clusters

Related Information

DLM support requirements

DSS support requirements

Hortonworks Support Matrix

Installation overview

DP Platform and its associated DP Apps are installed on a single host. The DataPlane Platform and the DP Apps run as a set of “containers” on Docker on this host. It is recommend this is a dedicated host distinct from other software or cluster hosts. We will refer to this host as your DP Instance (or “DP Host”).

Hosts from clusters that you plan to register into DataPlane must be accessible from this host. The hostname of a cluster node must be DNS addressable from the DataPlane host. In addition, for any DP Apps you plan to use with these clusters, you must install the requisite Cluster Agent for that DataPlane App (for example: DLM Engine or DSS Profiler). Be sure your clusters meet the hardware and software requirements for that particular Agent. See the DataPlane Support Matrix and the support matrix for each of the DP Apps that you want to install.

You are strongly encouraged to read completely through this entire document before starting the installation process, to that you understand the interdependencies and order of the steps.

Installation prerequisites

Product binaries

Prior to starting the DataPlane Installation, you must download the DataPlane repository tarballs (i.e. the “product binaries”) from the Hortonworks Customer Portal following the instructions provided as part of the subscription fulfillment process. DP Platform and the DP Apps (and related Agents) are provided as RPMs in tarball repositories.

Check DNS

Your system must be configured for both forward and reverse DNS.

Every host name used with DataPlane must be resolvable by DNS or configured in the /etc/hosts file on the DataPlane container, so that host names can be resolved between all cluster nodes. Using a DNS server is the recommended method, but if the instances are added to /etc/hosts, you must explicitly register the cluster host names within the DataPlane Docker containers. It is not sufficient to have the host names included in the /etc/hosts file on the DP Platform host. See the DP Platform Administration guide for instructions.

> **Note:** If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

Check ports

The DP Host requires the following ports to be available on the host:

Port	Description
80, 443	The DP Instance Web UI. Port 80 is redirected to port 443 for SSL. Refer to *Configuring DP Platform* to configure your own certificate.
8500	Used by Consul which handles the Docker container networking.

Disable SELinux

You must disable SELinux enforcement of permissions before installing the DataPlane service, due to an issue with SELinux and Docker.

If you do not disable SELinux, then DataPlane will not install and run properly, and you will have to destroy and reinstall the containers.

```
setenforce 0    #A zero, not a letter
sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/sysconfig/selinux
```

Important: The second command prevents SELinux from being automatically re-enabled after a reboot.

Install Docker

Docker containers are used to install each DataPlane service. You must install either Docker Enterprise Edition (EE) or Community Edition (CE). You might be required to reboot your system after installing Docker.

For general information about installing Docker, see Install Docker.

For Docker installation instructions for your operating system, access the appropriate Docker instructions:

- Get Docker EE for Red Hat Enterprise Linux
- Get Docker EE for CentOS
- Get Docker EE for Oracle Linux
- Get Docker CE for CentOS

LDAP

You need access to an enterprise LDAP setup when configuring DataPlane. Refer to *Enterprise LDAP requirements* for more information on the LDAP settings and options.

External Database

By default, DP Platform includes an embedded PostgreSQL instance for testing and evaluation purposes only. You should configure your DP instance to use an external PostgreSQL instance. We strongly recommend configuring DP Platform with an existing external database when running in production, and not use the embedded PostgreSQL. Refer to *Configuring DP Platform* for more information.

Related Tasks

Add host entries to the DPS environment

Related reference

DP Platform support requirements

Enterprise LDAP requirements

Related Information

DataPlane Administration

Setting up the local repository for DataPlane

You must download the DataPlane repository tarballs from the Hortonworks Customer Portal following the instructions provided as part of the subscription fulfillment process. DataPlane Platform and the DP Apps (and related Agents) are provided as RPMs in tarball repositories.

Prior to installing DataPlane, you must set up a server to host the RPMs in a local repository that can then be used to install the product binaries.

> **Note:** You can create the local repository on the same host as your DP Instance, but do not host the local repository on port 80 since that will conflict with your DP Instance.

Prepare the web server for the local repository

Before setting up your local repository, you must properly configure an HTTP web server, on which you will create the repositories.

Before you begin

Prior to preparing the web server, you must have:

- Selected a server host that runs a supported operating system. This will be the local repository host.
- Enabled network access from your target DP Instance host to local repository host.
- Ensured that the web server is not using port 80. This port is used by DataPlane and will cause a conflict if in use by your web server.
- Ensured that the hosts have a package manager installed such as yum (for RHEL, CentOS, or Oracle Linux).

Procedure

1. Create an HTTP server:
   a) On the local repository host, install an HTTP server (such as Apache httpd) using the instructions provided on the Apache community website.
   b) Activate the server.
   c) Ensure that any firewall settings allow inbound HTTP access from your cluster nodes to your local repository host.

      Note:

      If you are using Amazon EC2, make sure that SELinux is disabled.
2. On your local repository host, create a directory for your web server.

```
mkdir -p /var/www/html/
```

3. Optional: If you are using a symlink, enable the followsymlinks on your web server.

What to do next

You next must set up your local repository.

Related Information

Downloading the Apache HTTP Server

Set up a local repository for DataPlane

Setting up a local repository involves moving the tarball to the selected mirror server and extracting the tarball to create the repository.

Before you begin

- Ensure that you have downloaded the required tarball from the customer portal, following the instructions provided as part of the product procurement process.
- You must have completed the preparatory tasks before setting up a repository.

Procedure

1. Copy the repository tarballs to the web server directory and expand (uncompress) the archive file:
   a) Navigate to the web server directory you previously created.

   ```
   cd /var/www/html/
   ```

   All content in this directory is served by the web server.
   b) Move the tarball to the current directory and expand each of the repository tarball.
   Replace <filename> with the actual name of the RPM tarball that you are expanding.

   ```
   tar zxvf <filename>.tar.gz
   ```

   During expansion of the tarball, subdirectories are created in /var/www/html/, such as DP/centos7. These directories contain the repositories.

   Expanding the tarball takes several seconds.
2. Confirm that you can browse to the newly created local repositories by using the *Base URL*:

   ```
   http://<your_webserver>:port/<repo_name>/<OS>/<version>
   ```

   - <your_webserver>:port

     This is the FQDN and port of the web server host.
   - <repo_name>

     The repository name, usually the abbreviated name of the DataPlane component, for example DP for *DP Platform*.
   - <OS>

     The operating system.
   - <version>

     The version number of the downloaded component.

   DP Platform Base URL example:

   ```
   http://<your_webserver>:port/DP/centos7/1.1.0.0
   ```

   Remember this base URL. You need it to set up the repository configuration file in subsequent steps.
3. If you have multiple repositories configured in your environment, deploy the following plugin on all the nodes in your cluster.

   ```
   yum install yum-plugin-priorities
   ```

4. Edit the /etc/yum/pluginconf.d/priorities.conf file to add the following values:

```
[main]
enabled=1
gpgcheck=0
```

Results
The local repository is now set up and ready for use.

What to do next
Create the repository configuration file for the newly created local repository.

Create the repository configuration file

A repository configuration file (.repo file) must be created on the DataPlane host. The file is required to identify the path to the repository data, establish whether a GPG signature check should be performed on the repository packages, etc. Only one repository configuration file is needed.

Procedure

1. Navigate to the repository directory.
   cd /etc/yum.repos.d/
2. Create a repository file.
   vi dp.repo

   Alternatively, you can copy an existing repository file to edit.
3. Add the following content to the repository file.

   Important: Be sure to use the *Base URL* you created when setting up the local repository.

```
#VERSION_NUMBER=1.1.0.0
[DP-1.1.0.0]
name=DP  Version - DP-1.1.0.0
baseurl=http://<your_webserver>:port/DP/centos7/1.1.0.0
gpgcheck=1
gpgkey=http://<your_webserver>:port/DP/centos7/1.1.0.0/RPM-GPG-KEY/RPM-
GPG-KEY-Jenkins
enabled=1
priority=1
```

Results
You are now ready to install the DataPlane software.

Related Tasks
Install DataPlane Platform

Installing and Configuring DP Platform

Install DataPlane Platform

When installing DataPlane in a production environment, DP Platform and all associated DataPlane Services must be installed on a separate host that is not part of any cluster.

About this task

DP Platform is required as a baseline, but you can install any combination of DP Apps on top of the platform. After installing the platform, see the installation instructions for each DataPlane app for instructions on how to install and configure the app on the platform.

- You will be installing the DP Platform software using the local repositories.

Before you begin

- You need root user access on the DP Host to perform this task.
- You must have completed the actions identified in *DataPlane installation prerequisites*.
- The host must meet the requirements identified in the *DataPlane Support Matrix*.
- The host must be a dedicated host that is not part of an existing cluster. This prevents potential port conflicts with other cluster services.
- You must have the FQDN or IP address of the host available.

Procedure

1. Log in as root to the host on which you set up the DataPlane repositories.

   ```
   sudo su
   ```

2. Verify that SELinux is disabled.

   ```
   sestatus
   ```

   If SELinux is not disabled, stop now and verify that all installation prerequisites were successfully completed, then continue with the installation.

3. Be sure Docker is installed, configured, and started on the host.

   Refer to *Installation Prerequisites* for more information.

4. Install the RPMs for DP Platform.

   ```
   yum install dp-core
   ```

   A folder is created that contains the Docker image tarball files and a configuration script.

   If the yum command fails, then the local repository was not set up correctly. Check the repository file /etc/yum.repos.d/dp.repo on the DataPlane host.

What to do next

Proceed to initializing the DP Platform.

Related Tasks

Installation prerequisites

Setting up the local repository for DataPlane

Related reference

DP Platform support requirements

Related Information

Hortonworks Support Matrix

(Optional) Configure an external database

Although DataPlane includes an embedded PostgreSQL database, the embedded database is intended for nonproduction use. You should use an external database for production environments. After installing the database following the instructions provided with the database software, you must set up the database for use with DataPlane.

About this task

- You will be configuring an external database or your own SSL certificate.
- Refer to the *DataPlane Support Matrix* for requirements and supported databases.

  Be sure to have the database URI, username, and password available.

Before you begin

- You need root user access on the DP Host to perform this task.
- You must have the proper role to perform this task.
- The PostgreSQL database must have been installed and properly configured for remote access.
- A database must have been created.
- A database user must have been created and assigned permissions for the new database.

Procedure

1. Open the <installer_home>/config.env.sh file for editing.

   ```
   vi /usr/dp/current/core/bin/config.env.sh
   ```

2. Modify the DB Configs settings to add the appropriate connection information.

   ```
   USE_EXTERNAL_DB="yes"
   DATABASE_URI="jdbc:postgresql://<host_name>:5432/<database_name>"
   DATABASE_USER="<user_name>"
   DATABASE_PASS="<password>"
   ```

Results

Your external database is now set up so that you can configure it for DataPlane during DataPlane installation.

(Optional) Configure a TLS certificate

If you choose to use the default TLS (formerly SSL) certificates provided with DataPlane, then DataPlane generates self-signed certificates. If using your own certificates, then you must modify certificate configuration settings.

About this task

- DP Platform is required as a baseline, but you can install any combination of DP Apps on top of the platform. After installing the platform, see the installation instructions for each DataPlane app for instructions on how to install and configure the app on the platform.
- DataPlane supports only PEM-encoded certificates and only with OpenSSL 1.0.2k or later.
- You will be configuring your own SSL certificate.

Before you begin

- You need root user access on the DP Host to perform this task.
- DataPlane supports only PEM-encoded certificates and only with OpenSSL 1.0.2k or later.
- Have the full path and file name for the public key and private key (.pem files) and the certificate password available.

Procedure

1. If your private key file is in RSA/DES format, run the following command to convert the private key to PCKS8 encrypted key format:

   ```
   openssl pkcs8 -topk8 -in <server.key> -out <pcks8.key>
   ```

   Make sure to replace <server.key> with your key in RAS/DES format and <pcks8.key> with the desired PCKS8 encrypted key.
2. Open the configuration file:

   ```
   /usr/dp/current/core/bin/config.env.sh
   ```
3. Uncomment and modify the following properties:

   ```
   USE_TLS="true"
   USE_PROVIDED_CERTIFICATES="yes"
   DATAPLANE_CERTIFICATE_PUBLIC_KEY_PATH="<absolute_path_of_public_key_file>"
   DATAPLANE_CERTIFICATE_PRIVATE_KEY_PATH="<absolute_path_of_encrypted_private_key_file>
   ```
4. Save the file.

Initialize DP Platform

After installing the RPMs and optionally configuring your external database and TLS certificate, you must initialize DataPlane.

About this task

You will be initializing and configuring the DP Platform.

Before you begin

If you plan to use an external database or use your own TLS (SSL) certificate, be sure to configure those options prior to initializing the DP Platform. Refer to Configure an external database and Configure a TLS certificate for more details.

Procedure

1. Navigate to the folder containing the DataPlane configuration script.

   ```
   cd /usr/dp/current/core/bin
   ```
2. Initialize the software.

   This loads the Docker images into your local system and prompts for configuration options.

   ```
   ./dpdeploy.sh init --all
   ```
3. Create the password for a Super User and a Master Password for the system.

   Ensure that you remember these passwords, as they cannot be retrieved or reset. If you forget the password, contact Hortonworks Support for further assistance.

   ```
   DataPlane Services will now setup a Super User with a username 'admin'.
   This user can configure LDAP, add DataPlane Service Admins, enable services,
    etc.
   Setup a password for this user.

   Enter password for DataPlane Services Super User:
   ```

```
Re-enter password:

DataPlane Services will now setup a Master password that is used to secure
 the secret storage for the system.
Caution: The master password can be setup only once and cannot be reset
 easily.
You will need to provide it for various admin operations. Hence please
 remember what you enter here.

Enter master password for DataPlane Service (Minimum 6 characters long):
Reenter password:
```

4. When the initialization process completes, you can check the status of the docker containers using the following command:

```
docker ps
```

Sample output:

```
CONTAINER ID        IMAGE                                                COMMAND
             CREATED           STATUS              PORTS
                                 NAMES
a9ce135f1b4a    hortonworks/dp-app:1.1.0.0-388              "/bootstrap.sh"
          34 seconds ago    Up 33 seconds    0.0.0.0:80->80/tcp,
 0.0.0.0:443->443/tcp, 9000/tcp                          dp-app
3605425f7337    hortonworks/dp-cluster-service:1.1.0.0-388   "./
docker_service_..."   35 seconds ago    Up 33 seconds    9009-9010/tcp
                                                        dp-cluster-service
80bff7d6d4f4    hortonworks/dp-db-service:1.1.0.0-388        "./
docker_service_..."   35 seconds ago    Up 34 seconds    9000/tcp
                                                        dp-db-service
55658aefe1f5    hortonworks/dp-gateway:1.1.0.0-388           "./
docker_service_..."   36 seconds ago    Up 35 seconds    8762/tcp
                                                        dp-gateway
888df9acb2ea    hortonworks/dp-knox:1.1.0.0-388              "/usr/dp-
scripts/k..."   38 seconds ago    Up 36 seconds    53/udp, 8300-8302/tcp,
 8400/tcp, 8301-8302/udp, 8500/tcp                        knox
4146fe3dc416    consul:1.0.1                                 "docker-
entrypoint..."   39 seconds ago    Up 38 seconds    8300-8302/tcp,
 8301-8302/udp, 8600/tcp, 8600/udp, 0.0.0.0:8500->8500/tcp   dp-consul-
server
d6e1fc2c6d28    postgres:9.6.3-alpine                        "docker-
entrypoint..."   53 seconds ago    Up 51 seconds    5432/tcp
                                                       dp-database
```

Output descriptions:

Docker Container	Description
dp-app	DataPlane Instance application (UI, web, etc.), accessible from port 443 (port 80 redirects to port 443)
dp-cluster-service	Powers how DP Platform talks to clusters
dp-db-service	Backend data store API in support of DP Apps
dp-gateway	Handles routing between DP Platform, DP Apps, Knox, etc.
dp-knox	Runs a Knox instance that wraps the DataPlane instance. This is the AuthN enforcement point for Data (SSO).
consul	Handles the networking of the containers
postgres	The PostgreSQL database instance that the DataPlane instance uses by default. This is not used if you external database.

5. Browse to your DP Instance host and proceed to log in using the Super User admin account.

Note:

If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

```
https://<DPS_host_FQDN>
```

As part of the installation process, data collection using cookies and other telemetry mechanisms is turned on by default. To disable all data telemetry, see the topic for disabling data telemetry.

What to do next

You can now complete configuration of the DataPlane Platform.

Related Tasks

(Optional) Configure a TLS certificate

Related Information

Disable and enable data telemetry

Log in and configure DataPlane Platform

To complete your installation, you must log in and configure the DP Instance for LDAP.

About this task

You must configure your DP Instance for LDAP and set up your initial DataPlane (DataPlane) Admin users and groups.

Before you begin

Be sure you have your enterprise LDAP configuration available. Refer to *Enterprise LDAP requirements* for more information.

Procedure

1. Browse to your DP Instance host.

   If you are using AWS, do not use the public DNS to access DataPlane. Use a public IP address or set up and use a DNS (Route 53) fully qualified domain name (FQDN).

   https://<DataPlane-host-FQDN>
2. Log in using the Super User admin account you configured during initialization:

   Username: admin
   Password: Use the password specified during DataPlane initialization

   After login, the initial “Onboarding Welcome” screen displays.
3. Click **Get Started** to proceed with setting up authentication.

   The Onboard/Configure LDAP page displays the Setup Authentication settings.
4. Enter your LDAP information.

   See Enterprise LDAP requirements for more details on these settings and options.

   If you are using LDAPS and a self-signed certificate, be sure to click *Upload certificate* and provide your certificate in order for DataPlane to connect to your LDAP instance. Use the corporate LDAP values that you collected during the preparatory steps.
5. Click **Save**.
   A success message displays on the page.
6. Add users and groups, which will be the initial members of the default DataPlane Admin role.

At least one user must be configured and must have at least the DataPlane (DataPlane) Admin role, which is needed to add other users or groups, assign them roles, configure services in DataPlane, etc.

Tip:

You must click the name of the user when it displays and ensure it appears in the Users field on a dark background.

If the name appears on a white background, it means the name is not recognized and the action fails.

7. Click **Save & Login** to save your changes.
8. Log in as one of the DataPlane Admin users you added.

Results

You are now ready to proceed to managing your DP Platform, including cluster registration and DataPlane App installs.

What to do next

You can now install and configure additional DP Apps and manage your DP Instance.

Related reference

Enterprise LDAP requirements

Configure Knox SSO between DataPlane and HDP

This topic provides an overview of how to configure Knox SSO in your HDP cluster to work with DataPlane. Refer to the HDP documentation for details that might be applicable to your specific HDP configuration and setup.

About this task

- You will be configuring Knox SSO in your HDP cluster to work with your DP instance.

Before you begin

- You must have installed and configured DataPlane.
- You must have configured Knox SSO on your clusters.

  See *Knox SSO with DataPlane* for details.
- Knox SSO, LDAP, and Ranger must have been configured for HDP and Ambari.
- You must have an SSL certificate (such as a .pem file) available and have access to the public key in the file.
- Knox host FQDN must be DNS addressable and available from your DataPlane environment.

  If it is not, the Knox IP address must be in the /etc/hosts file on the DP environment. Refer to the *DataPlane Administration Guide*for details on how to add Knox to the DataPlane environment hosts.

Procedure

1. In a terminal, SSH to the DataPlane host.
2. Navigate to $DP_INSTALL_HOME/certs/.

```
cd /usr/dp/current/core/bin/certs/
```

3. Display the content of the ssl-cert.pem file.

```
cat ssl-cert.pem
```

4. Copy and retain the DataPlane public key displayed in the certificate between “Begin Certificate” and “End Certificate”, because you need it in a succeeding step.

   The public key looks similar to the following:

   -----BEGIN CERTIFICATE-----

```
NIICzTCCAaKjAwIBAgIIVJzHWfmsfP8wDQYJK0ZIhvcNAQEFCQAwXzELMAkGA1UE
BhMCVVMxDTALBgMVBAgTBFRlc4QxDTALBgNVBAcTBFRlc3QxFzANBgNVBAoTBkhH
ZG9vcDETMAsGA1WECxMEVGvzdDESMBAGA4UEAxMJbG9jYWxob3N0MB2XDTE3MDcx
MjEzMTUxMVoXDTE0MDcxMjEzMTUxMVoWxzELMAkGA1UCBhMCVVMxDTALBgNVBAgT
BFRlc9QxDTALBgNVBCcTBFRlc3QxFzANBgNVBAoTBkhhZG9vcDENNAsGA5UECxMF
VGvzdDESMBBGA1UEAxMJbG8jYWxob3N2MIGfMA0GCSqGSIb3DQLBAQUAA4GNAKCB
iQKBgQcYLhQDwCcQa12BZ2+v1gWICsFxOplW+EA6tBCJtMJDs5sNSV/XiomPxY2D
8OU3oY68DiLs/U+la4K2mHp+gvh5+91EuMvXHtkui++7zDtD+cfBmsY5peAFwZ6g
2NBwIjyMsKSiJWtT4syKMnAB5yv2p8xp3Z6c+0GCmZ+EeguWVQyDAQABMA0GCSqG
zIb3DQEBbQUAA9GBAJAeMEFZY1Q4mK+RFq6wbshUOSQR+wB8zDkxAtgPfQINR9tK
5MA8Iy6J90/eBUqGvAoN8PbEnTHU5VsL6m3J0vPmJ4EzFqCwI5VjeWdIMdoPPB/b
QfmRZb0bpriGv6TrNdr9SKDTlchxW2tBbB1PaiR5yi3oEsuAaNKsi7GeT2wa
```

   -----END CERTIFICATE-----

5. On your HDP cluster Knox host, create a token.xml topology file.

```
vi /etc/knox/conf/topologies/token.xml
```

6. Add the required content to the token.xml file:

   a) Add the basic topology content.

   You can copy and paste the following content into the file and modify the content as needed.

```
<?xml version="1.0" encoding="UTF-8"?>
<topology>
   <uri>https://$KNOX_HOSTNAME_FQDN:8443/gateway/token</uri>
   <name>token</name>
   <gateway>
      <provider>
         <role>federation</role>
         <name>SSOCookieProvider</name>
         <enabled>true</enabled>
         <param>
            <name>sso.authentication.provider.url</name>
            <value>https://$KNOX_HOSTNAME_FQDN:8443/gateway/knoxsso/api/
v1/websso</value>
         </param>
         <param>
            <name>sso.token.verification.pem</name>
            <value>
                $ADD_THE_PUBLIC_KEY_HERE
            </value>
         </param>
      </provider>
      <provider>
         <role>identity-assertion</role>
         <name>HadoopGroupProvider</name>
         <enabled>true</enabled>
      </provider>
      <provider>
         <role>authorization</role>
```

```
            <name>XASecurePDPKnox</name>
            <enabled>true</enabled>
         </provider>
      </gateway>

      <service>
         <role>KNOXTOKEN</role>
         <param>
            <name>knox.token.ttl</name>
            <value>500000</value>
         </param>
         <param>
            <name>knox.token.client.data</name>
            <value>cookie.name=hadoop-jwt</value>
         </param>
         <param>
            <name>main.ldapRealm.authorizationEnabled</name>
           <value>true</value>
         </param>
      </service>
</topology>
```

HadoopGroupProvider enables the Hadoop user-group mapping, which identifies the groups to which users belong.

The authorization=XASecurePDPKnox parameter and main.ldapRealm.authorizationEnabled=true parameter enable Ranger authorization with the token topologies in Knox.

b) Replace $KNOX_HOSTNAME_FQDN with the fully qualified domain name of the host.

c) In the sso.token.verification.pem parameter, paste in the public key value that you copied in a previous step, replacing $ADD_THE_PUBLIC_KEY_HERE.

7. Perform a secure copy of the token.xml topology file to a Knox-enabled node on the HDP cluster.

8. Verify that Knox has picked up the files:

a) Log in to the Knox-enabled node.

b) Ensure that a directory called token.topo.<number> is present in the path /var/lib/knox/data-<version>/deployments/.

If the file is not present, verify that the content in the token.xml file is correct. You can check the Knox gateway logs for error information.

9. Log in to each additional cluster used with DataPlane and repeat Step 5 (create a token.xml file) through Step 8 (verify copy of the file).

10. Configure the Knox SSO topology to point to an LDAP instance for DataPlane to use.

a) Open the file <knox_gateway_home>/topologies/knoxsso.xml.

b) Modify the following properties with appropriate values for your environment:

main.ldapRealm.userDnTemplate
main.ldapRealm.contextFactory.url
knoxsso.redirect.whitelist.regex

Sample configuration for using packaged LDAP:

```
<param>
   <name>main.ldapRealm.userDnTemplate</name>
   <value>uid={0},ou=people,dc=hadoop,dc=apache,dc=org</value>
  </param>
  <param>
   <name>main.ldapRealm.contextFactory.url</name>
   <value>ldap://localhost:33389</value>
  </param>
<param>
       <name>knoxsso.redirect.whitelist.regex</name>
```

```
        <value>.*;^/.*$;https?://localhost*$;^http.*$</value>
    </param>
```

Related Concepts

Knox SSO with DPS

Related Tasks

Add host entries to the DPS environment

Related Information

Defining Cluster Topologies

Configure Knox Gateway for DataPlane and HDP

DP Platform communicates with services on the HDP cluster like DataPlane Agents, Ambari, Atlas, Ranger, etc. If you are using TLS wire encryption on your clusters, you must configure Knox Gateway to proxy requests to and from DP Platform. If DataPlane uses proxying, then all DataPlane services need to use it to communicate with DataPlane.

About this task

This topic provides an overview of how to configure Knox Gateway proxy in your HDP cluster to work with DataPlane. Refer to the HDP documentation for details that might be applicable to your specific HDP configuration and setup.

- You will be configuring Knox Gateway proxy in your HDP cluster to work with your DP instance.

Before you begin

- You must have installed and configured DataPlane.
- You must have configured Knox Gateway on your clusters.

  See *Knox Gateway proxying with DataPlane* for details.

Knox host FQDN must be DNS addressable and available from your DataPlane environment. If not, the Knox IP address must be in the /etc/hosts file on the DP environment. Refer to the *DataPlane Administration* guide for details on how to add Knox to the DataPlane environment hosts.

Procedure

1. On your HDP cluster Knox host, navigate to the Knox topologies directory.

   ```
   cd /etc/knox/conf/topologies
   ```

2. Create a DataPlane proxy topology file.

   ```
   vi dp-proxy.xml
   ```

3. Add the host name for each of the services listed in the file, based on where that service is running in your HDP cluster.

   Tip: At this point, you can add to the file the DataPlane service agents that you plan to install, or you can add them later.

   Important:

   - Do not modify the URL in the provider section of the file.
   - Be sure to keep this file updated if you move services or add services in your cluster.

The <localhost> entry in the following example might be something like ctr-e138-1518143905142-369209-01-000005.hwx.site:20070.

```
Topology dp-proxy.xml

<?xml version="1.0" encoding="utf-8"?>
<topology>
  <gateway>
    <provider>
        <role>federation</role>
        <name>SSOCookieProvider</name>
        <enabled>true</enabled>
        <param>
            <name>sso.authentication.provider.url</name>
            <value>https://localhost:8443/gateway/knoxsso/api/v1/websso</
value>
        </param>

    <provider><role>identity-assertion</role>
      <name>Default</name>
      <enabled>true</enabled>
    </provider>
   </gateway>

    <service>
      <role>WEBHDFS</role>
      <url>http://<localhost>:20070/webhdfs</url>
  </service>
  <service>
      <role>WEBHCAT</role>
      <url>http://<localhost>:20111/templeton</url>
  </service>
  <service>
    <role>AMBARI</role>
    <url>http://<localhost>:8080</url>
  </service>
  <service>
    <role>RANGER</role>
    <url>http://<localhost>:6080</url>
  </service>
  <service>
    <role>RANGERUI</role>
    <url>http://<localhost>:6080</url>
  </service>
  <service>
    <role>ATLAS</role>
    <url>http://<localhost>:21000</url>
  </service>
  <service>
    <role>ATLAS-API</role>
    <url>http://<localhost>:21000</url>
  </service>
  <service>
   <role>BEACON</role>    ##The DLM Engine
   <url>http://<localhost>:25968</url>
  </service>
  <service>
   <role>HIVE</role>
   <url>http://<localhost>:10001/cliservice</url>
  </service>
  <service>
   <role>RESOURCEMANAGER</role>
   <url>http://<localhost>:8088/ws</url>
```

```
  </service>

  <service>
   <role>PROFILER-AGENT</role>    ##The DSS Agent
   <url>http://<localhost>:21900</url>
  </service>

</topology>
```

Configure Ranger to restrict access to DataPlane

You must configure a Ranger policy for the new Knox topology, in order to restrict access to only authorized users of DataPlane.

Procedure

1. Navigate to the Ranger UI.
2. Click **Access Manager**, and then click the Knox repository link, for example: **<cluster-name> Policies**.
3. Click **Add New Policy**, and then enter the following values:

Parameter	Value
Policy Type	Access
Knox Topology	token
Knox Service	*

4. Enter groups or user names in **Select Group** or **Select User**.
5. Optional: Under Policy Conditions click **Add Condition** and enter the IP addresses of the DataPlane host.

   This adds an IP-based filter to ensure that only known DataPlane Core hosts can access cluster services through the token topology.
6. Under Permissions, click **Add Permission** and select **Allow**.

Troubleshooting DataPlane Installation

Cluster Registration Error Messages

Following are errors you might encounter while registering a cluster in DataPlane on the Add Your Cluster page. Some possible causes and possible resolutions are also included.

Cannot register a cluster, other causes

If you cannot register a cluster with DataPlane and none of the errors mentioned are the cause, the following might apply.

Procedure

1. Verify that the hostname where Knox is running is valid and reachable from the DataPlane host machine. If it is reachable, try adding the hostname resolution to the DataPlane container using the ./dpdeploy.sh utils add-host <ip> <host> command.
2. Verify that network connectivity settings, such as firewalls, are correctly configured.

3. Verify that the username is an Ambari Admin on the cluster. If not, make the user an Ambari Admin user, by logging into Ambari, selecting the user, and providing Admin privileges.
4. See the Hortonworks Support Marix and verify that you are running a supported configuration.

Cluster is not reachable

DataPlane containers are not able to resolve a provided hostname or use the IP address to connect to the machine.

DNS resolution is not setup.

There are firewall or other networking restrictions that are preventing access.

Sample Message:

Failed: This is not a valid Ambari URL.

See the dpdeploy.sh Script Command Reference.

Procedure

1. Verify that the specified hostname or IP address is valid and reachable from the DataPlane host machine.
2. If the hostname or IP address is reachable, try adding the hostname resolution to the DataPlane container using the ./dpdeploy.sh utils add-host <ip> <host> command.
3. Verify if network connectivity settings, such as firewalls, are configured correctly.

Knox is not set up on the HDP cluster, or Ambari credentials are incorrect for 'seeded user' mode

This error occurs when the cluster is reachable, but authentication is failing.

Sample Message:

Unable to connect, please retry. DataPlane could not retrieve cluster information.

Possible Causes:

- Knox is not set up on the cluster.
- The user wants to use the less secure 'seeded user' mode, but the credentials of the seeded user (user name or password) are not setup in DataPlane.

Procedure

1. Validate that Knox is configured correctly as per documentation.
2. If seeded user mode is being used (for evaluation purposes), add the correct credentials to DataPlane using ./dpdeploy.sh utils update-user ambari.

   See the dpdeploy.sh Script Command Reference.

Knox setup is incorrect on the HDP cluster

This error indicates that the cluster being registered has Knox enabled, but the communication from DataPlane to Knox is failing.

Sample Message:

Failed: There was an error fetching information from Ambari.

Possible Causes:

- The Knox token service is not properly configured.
- The public key of DataPlane is not set up correctly in the Knox topology.

Procedure

Validate that the Knox configuration for the token topology is done correctly, following the instructions Configuring DataPlane for Secure Clusters.

Cluster status displays incorrectly on Details page

On the Cluster Details page, sometimes the Status of a cluster displays in gray, instead of red or green.

This generally indicates a timeout issue, in which DataPlane is not able to refresh the cluster details correctly. Manually refreshing the cluster information should fix the problem.

Procedure

1. Click the Actions icon at the end of the row.
2. Click Refresh.

   Refresh of the cluster status might take several seconds.

Logging in using the DataPlane local admin role

The local admin role allows you to perform administrative activities and troubleshoot problems when access through LDAP and Knox is not available. The local admin is also the role you use to log in to DataPlane the first time, before LDAP is configured in DataPlane for SSO.

About this task

When you log in as the local DataPlane Admin, you bypass Knox.

For login, the default username is “admin”. The password you use to log in is set during the installation process.

Procedure

Log in by appending /sign-in to the DataPlane login URL, for example:

```
http://dataplane-host-name/sign-in
```

wget command is not available

Use the command yum install wget to install the wget tool.

Delete and clean up Docker containers

If you have problems with your installation or want to update a DataPlane container, you can delete the Docker containers and then install the new images.

About this task

> **Important:** Performing this task deletes all of your DP Platform database content, so you will have to reconfigure the LDAP and cluster registration settings after reinstalling the Docker containers.

For information about the commands and options supported by ./dpdeploy.sh, use the command-line help.

Before you begin

You must be root user to perform this task.

Procedure

1. cd /usr/dp/current/apps/dlm/bin
2. ./dlmdeploy.sh destroy
3. cd /usr/dp/current/core/bin
4. ./dpdeploy.sh destroy --all
5. docker ps

   This ensures that no containers are running. If you see any, kill them with docker kill.
6. Go to https://docs.hortonworks.com/HDPDocuments/DP/DP-1.1.0/installation/dp_initialize.html#task_o1v_bdd_tdb and run the original DataPlane deployment commands starting with ./dpdeploy.sh init --all.