Getting started with IBM BigInsights on Bluemix v1.0

Contents

Getting started with BigInsights

Use this service to provision enterprise-scale, multi-node big data clusters on the IBM® SoftLayer cloud, using IBM's big data solution InfoSphere® BigInsights®. Once provisioned, these clusters can be managed and accessed from this same service.

About this task

InfoSphere BigInsights is IBM's Hadoop offering, which combines open source technology with extra features (for example Big SQL and Text Analytics) to provide industry-leading performance, scale, and reliability. By using the IBM BigInsights Hadoop for Bluemix™ service, you can access all the power of InfoSphere BigInsights for your enterprise without having to install and configure your clusters or manage your hardware. You can begin using your cluster productively as soon as you are notified that it is ready.

The BigInsights service is similar to another IBM Bluemix service that is called Analytics for Hadoop. The Analytics for Hadoop service, by contrast, provisions a single-node, VM cluster for your use on a limited license and is good to try the features. BigInsights instead provides a way to request multi-node, IBM SoftLayer clusters, which are built on proven a reference architecture for high performance and reliability. You can optionally request high availability on the NameNode so that you can use this cluster for mission-critical business transactions.

The following steps provide a roadmap of the tasks you can complete from the BigInsights service.

Procedure

1. Provision a cluster.
2. Complete the first steps for your provisioned cluster.
3. Manage your cluster or clusters. You can get cluster details from the BigInsights service Manage Clusters page. Open the InfoSphere BigInsights console to configure or change it.
   a. Optional: Add nodes to a cluster.
   b. Optional: Delete a cluster.
4. Import data into the cluster. You can import your smaller data sets (smaller than 2 GB) directly from the InfoSphere BigInsights console; the linked section describes various tools for different types (in motion or at rest), from different sources (Swift, HDFS, and so forth) and sizes of data sets (larger than 2 GB).
5. Get started with analyzing your data. The InfoSphere BigInsights console includes a welcome page that describes a number of starting tasks. Use one of the many tools provided by InfoSphere BigInsights to analyze your data. These tools include (links go to the corresponding tutorials):
   - BigSheets. Use BigSheets in the InfoSphere BigInsights console to manipulate and discover data trends and perform extensive text analytics in a familiar spreadsheet format.
   - Big SQL. Big SQL enables IT professionals to create tables and query data in BigInsights using familiar SQL statements. To do so, programmers use standard SQL syntax and, in some cases, SQL extensions created by IBM to

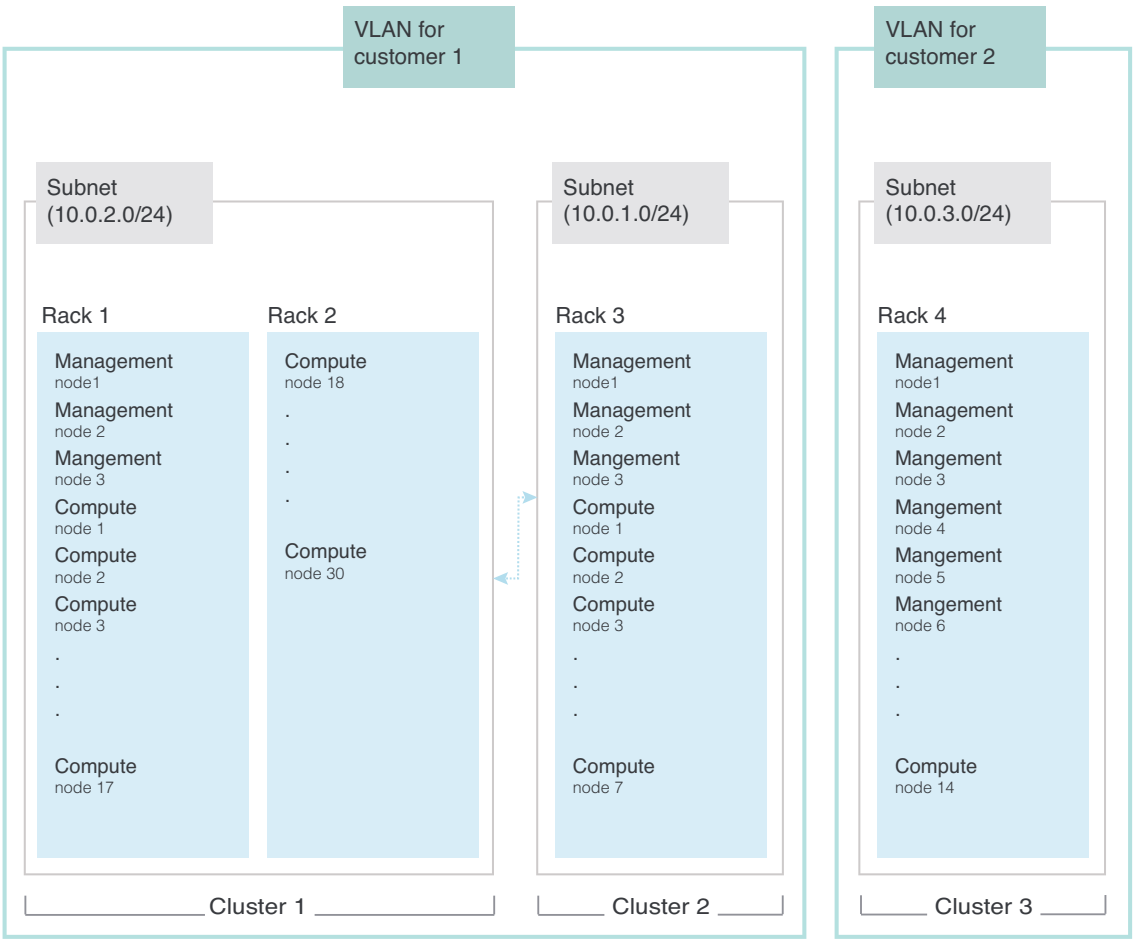
make it easy to exploit certain Hadoop-based technologies. You can open the Big SQL console to run Big SQL queries (under **Quick Links** on the Welcome page).

- Developing an application. Deploy and run applications from the Applications page (click **Manage**). The sample applications are deployed for you, but you can deploy your own applications too.
- Big R. Big R is a library of functions that provide end-to-end integration with the R language and InfoSphere BigInsights. Use Big R for comprehensive data analysis on the InfoSphere BigInsights server, hiding some of the complexity of manually writing MapReduce jobs.
- Text Analytics. Use Text Analytics to analyze large volumes of text and produce annotated documents that provide valuable insights into unconventional data. Use Text Analytics to search for customer web browsing patterns in clickstream log files, find fraud indicators through email analytics, or assess customer sentiment from social media messages.

Architectural overview of your cluster

With the BigInsights service, you can provision and manage InfoSphere BigInsights clusters on the cloud. Your cluster is within a private VLAN.

The following diagram shows the topology for three provisioned BigInsights clusters. *Provisioned cluster* refers to every group of BigInsights nodes that were provisioned as part of an order from a customer. Customers can place orders either through the IBM Bluemix catalog or by IBM or IBM Business Partner sales channels. You can see in the diagram that each customer has a separate VLAN that is protected from other customer VLANs. You can have more than one cluster in your VLAN, as shown for Customer 1.



Each different BigInsights customer has a private VLAN.

The network of nodes is configured as follows:

Management nodes

- These nodes are accessible on a public network
- Inbound traffic is controlled by the iptables firewall. Only the following ports are open:

Tool or function	Port
InfoSphere BigInsights web console	8443
HttpFS	14443
Big SQL 1.0	7052
Big SQL (all new in InfoSphere BigInsights 3.0; to understand the differences between Big SQL 1.0 and Big SQL see Which version of Big SQL should you use?)	51000
Hive	10000
DNS	53
Curl	443
SSH (SSH communication is open only to the management nodes)	22

- Outbound traffic is not restricted.

Compute nodes

- These nodes are within a private network and accessible only through SSH protocol from the management nodes.
- Inbound traffic from any entity outside the cluster is blocked.
- Outbound traffic to any entity outside the cluster is blocked.

All nodes

Fail2Ban software is installed on all nodes to prevent bot attacks.

Security overview

One of the major benefits of BigInsights is its security. The following list summarizes the access and authentication for your cluster. After you receive notification that your cluster is provisioned, complete the first steps to secure your cluster.

- LDAP authentication is configured with the policy to lock out users after five consecutive attempts to log in with an incorrect password.
- LDAP policies are configured to require that passwords change after 90 days for the following users of the service:
  - LDAP administrator (`Manager`)
  - BigInsights administrator (`biadmin`)
  - BigInsights catalog user (`catalog`)
  - Big SQL user (`bigsql`)
- Provided SSL certificates are self-signed by IBM.
- JDBC access to Big SQL is configured for SSL with a self-signed certificate.

After your cluster is ready, IBM does not continue to maintain the servers. You must maintain them in accordance with your own security standards. IBM retains the right to shut down the clusters if the way they are being used is in violation of the hosting center's terms of use.

Provisioning a new cluster

From the BigInsights service on IBM Bluemix, you can plan for and request a new cluster for your enterprise on IBM SoftLayer.

Procedure

To plan and provision a new cluster:

1. Log in to IBM Bluemix.
2. From the IBM Bluemix catalog, add the BigInsights service, leaving it unbound under **App**. Click **CREATE**. The BigInsights service page opens.
3. On the BigInsights service page, click **Define Clusters**. The Define Clusters page opens. You can try different combinations to estimate the cost of your cluster.
4. Provide a unique name for the cluster. This name is used in the Manage Clusters page, but also during interactions with IBM Sales. For example, `Sales` or `Analytics Sandbox`.

5. Estimate the cluster requirements. Follow the information that is provided in the window. General examples of node usage are provided; if you click **Compare hardware**, you can see a detailed view of the configuration for **Small**, **Medium**, and **Large** choices.

   You can preview the cluster to see an interactive graphic of the cluster with a list of installed open source and IBM components and services on each node.
6. If you are familiar with Hadoop clusters, you can specify how many compute nodes you need. Alternatively you can specify the amount of data you think you need to store at any time in the cluster and the number of nodes is selected for you. The ratio of storage to compute nodes differs based on your selection of **Small**, **Medium**, or **Large** for your hardware needs.
7. Click **REVIEW CONFIGURATION** to review a detailed configuration and a preliminary cost estimate for your cluster that is based on your selections. The cost estimate is based on the combination of your selections: hardware size, number of compute nodes, and high availability (selecting high availability increases the number of management nodes).

   By clicking **CANCEL**, you can return to the previous page to adjust your settings until you're satisfied.

   Restriction: Do not use the browser **Back** button. If you do, you cannot reuse the same cluster name.
8. After you're happy with your cluster definition, click **CONTINUE**. Your credit card that is in the IBM Bluemix system *will not* be charged. IBM Sales is contacted with your subscription request. You also receive an email that notifies you that your request was received. IBM Sales contacts you to complete your order.

   After you complete the order with IBM Sales, you will receive email to notify you of the progress of your cluster.

What to do next

You can check the progress of your cluster at any time by clicking **Manage Clusters** from the BigInsights service page in IBM Bluemix and finding the **Status** field. Statuses include:

- Cluster Requested
- Approved for Provisioning
- Acquiring Hardware
- Installing

Managing your BigInsights clusters

After your cluster or clusters are provisioned, you can manage them from the BigInsights service. You can also open the InfoSphere BigInsights web console where you can work with big data.

Procedure

1. From your IBM Bluemix dashboard, click the BigInsights service.
2. From the BigInsights service page, click **Manage Clusters**. The Manage Clusters page opens where you can see a filterable list of your clusters.
3. Open the web console or examine cluster details.

Option	Description
Click the cluster name or its launch icon	Open the InfoSphere BigInsights web console in a new tab. From this console, you can manage your servers and clusters.
Click anywhere else in the row	The cluster details page opens. Here you can view the status for the cluster or for individual nodes in the cluster. You can also: • Request more nodes for the cluster • Delete the cluster • Open the web console by clicking **Launch**.

Adding nodes to your BigInsights cluster

After your cluster or clusters are provisioned, you can add one or more nodes to a cluster.

Procedure

1. From your IBM Bluemix dashboard, click the BigInsights service.
2. From the BigInsights service page, click **Manage Clusters**. The Manage Clusters page opens where you can see a filterable list of your clusters.
3. In the list of your clusters, find the cluster that you want to add to. Open the cluster details by clicking somewhere in the row; do not click the cluster name because that opens the InfoSphere BigInsights web console instead. The cluster details page opens, where you can view the individual nodes in the cluster.
4. Click **Actions** > **Request Add Nodes** to request more nodes for the selected cluster.
5. Enter the number of nodes you want and click **SEND REQUEST**. Although you can request one node at a time, requesting all nodes that you think you want at the same time is more efficient.

Results

Your IBM Sales representative is notified of your request so that they can work with you to add it to your contract and approve it. You are notified by email after the nodes are approved (subject line: “Your request to add nodes has been approved”) and then again when the nodes are available (subject line: “Your additional BigInsights nodes are available”).

Deleting your cluster

You can delete a provisioned cluster. Back up any data that you want to save before you make the request.

Procedure

1. Back up any data that you want to keep from the cluster. Use one of the methods for importing or exporting your data by pushing or pulling the data from the cluster into another cluster or an external system.
2. From your IBM Bluemix dashboard, click the BigInsights service.
3. From the BigInsights service page, click **Manage Clusters**. The Manage Clusters page opens where you can see a filterable list of your clusters.

4. In the list of your clusters, find the cluster that you want to delete. Open the cluster details by clicking somewhere in the row; do not click the cluster name because that opens the InfoSphere BigInsights web console instead. The cluster details page opens.
5. Click **Actions** > **Request Delete Cluster**.
6. Confirm your request by clicking **DELETE CLUSTER**.

Results

Your IBM Sales representative is notified of your request so that they can work with you to approve it. After approval, you are notified and then again when the nodes are available.

First steps for your provisioned cluster

First steps for your provisioned cluster

You've been notified by email that your cluster is ready. So now what? Before you can use it productively, you need to properly secure it.

Before you begin

These tasks require that you can establish a Secure Shell (SSH) with the cluster nodes. Windows users require an SSH client such as PuTTY.

About this task

You need to change the default passwords for your cluster, establish SSL communications, and change the root SSH (Secure Shell) keys. You also need to authorize users and groups to the cluster. Authorization for BigInsights is managed by LDAP and you can complete the LDAP tasks by using the provided script.

Procedure

1. Find the `biadmin` password and LDAP script.
2. Change default system passwords
3. Add new users and groups.
4. Configure SSL connections.
5. Decide whether you want to use SMTP for BigInsights alerts. If you do, contact IBM Support for assistance with getting this set up in your IBM SoftLayer cluster.

Step 1: Finding the biadmin password and LDAP script

`biadmin` is the default cluster administrator for BigInsights clusters. Find the `biadmin` user password on the cluster Details page. Log in to your cluster to find the LDAP script.

About this task

The `biadmin` password is also the initial LDAP root bind password for the cluster. You'll need this LDAP root bind password for all LDAP operations, so make a note of it (you can change it to whatever you want). With this root password, you can bind (connect) to the LDAP hierarchical structure as the root user with full permissions to modify anything in the LDAP directory structure.

Procedure

1. From your IBM Bluemix dashboard, click the BigInsights tile to open the service.
2. From the IBM BigInsights for Hadoop service page, click **MANAGE CLUSTERS**.
3. In the list of your clusters, find the cluster. Open the cluster details by clicking somewhere in the row; do not click the cluster name because that opens the InfoSphere BigInsights web console instead.
4. Find the `biadmin` password on the cluster details page as shown.

   Click **Show** to display the password.
5. Establish an SSH with the cluster management node and log in with the `biadmin` credentials.
6. Locate the LDAP script and configuration information. By default, the LDAP script (`ehaas_ldap.sh`) and the associated configuration file (`ehaas_ldap.conf`) are stored in `/opt/ehaas/bin`.

   By default, the LDAP script log, `bi_ldap.log`, is saved in `/opt/ehaas/log`. You can change this location by modifying the $LOG_FILE variable in the LDAP script (`ehaas_ldap.sh`).

Step 2: Changing default system passwords

Change the default passwords for the following users: `root`, LDAP administrator (`Manager`), BigInsights administrator (`biadmin`), BigInsights catalog user (`catalog`), and Big SQL user (`bigsql`).

About this task

The following user IDs are created for you with default passwords:

- LDAP administrator (`Manager`)
- BigInsights administrator (`biadmin`)
- BigInsights catalog user (`catalog`)
- Big SQL user (`bigsql`)

Change these user passwords as soon as possible to properly secure your cluster. You also need to change the default system `root` password.

To change users' passwords for them, run the LDAP script. Use the **`-w`** option to specify the root bind password and then the **`-p`** option to specify the user whose password you want to change. Users can change their own passwords as normal by running the **`passwd`** command, because your cluster's LDAP installation is integrated with PAM.

Recommendation: Changing certain system user passwords (`biadmin`, `catalog`, and `bigsql`) requires you to run other BigInsights scripts afterward so that all BigInsights daemons get updated with the new passwords. This process runs automatically when you run the LDAP script with the **`-p`** option, but must be done

manually if you change the passwords with the **passwd** command. It is therefore highly recommended to change these system passwords with the LDAP script.

Procedure

1. Log in to the LDAP master node as `biadmin`.
2. Switch to `root`. You must run the script as `root`.
   ```
   su
   ```
3. Change the password first for the LDAP administrator (`Manager`). In this case, you are changing the root bind password with the **-r** option.
   ```
   ehaas_ldap.sh –r
   ```
   You are prompted to enter the existing password and the new password.
4. Change the password for the user `biadmin`.
   ```
   ehass_ldap.sh -w rootbindpw -p biadmin
   ```
5. Run the script twice more to change the other BigInsights system users' passwords. These users are: `catalog` and `bigsql`.
6. Change the `root` password on all cluster nodes.
   ```
   passwd
   ```
   You are prompted to enter the new password.
7. Back up the root SSH key folder. The changes in the subsequent steps remove all authorized keys from establishing SSH communication with your system. They also change your keys if you shared anything with a remote system.
   ```
   mv /root/.ssh /root/.ssh_backup
   ```
8. Create new SSH keys.
   ```
   ssh-keygen -t rsa
   ```
   Accept the default values at each prompt.
9. Optional: Establish passwordless SSH across nodes in your cluster. As `root`, share the new public key with the communication node.
   ```
   ssh root@Y "cat>>/root/.ssh/authorized_keys" < /root/.ssh/id_rsa.pub
   ```
   If `cat` is not in the path, find the correct path for `cat` and insert it into the syntax, for example,
   ```
   ssh root@Y "/sbin/cat>>/root/.ssh/authorized_keys" < /root/.ssh/id_rsa.pub
   ```
10. Log in to each node in your cluster, switch to `root`, and repeat steps 6 - 9 on each node.

Step 3: Adding new users and groups

Add new users to the system by using the LDAP script. When you add a user, you also assign it to a primary group. You can add more groups and assign users to these secondary groups as well.

Before you begin

By default, users that you create have passwords that expire. These users are also locked out for 15 minutes after three unsuccessful attempts to log in. If you want to change these default settings, do so before you create new users because the default settings apply to users created after you change the settings. Changing these settings does not retroactively change the settings for users who were created earlier. You can change the default LDAP settings by editing the `/opt/ehaas/bin/ehaas_ldap.conf` file.

About this task

Procedure

1. Log in as the BigInsights administrator (`biadmin`).
2. Switch to `root` to run the LDAP script. You must run the script as `root`. Enter:
   ```
   su
   ```
3. Add one or more users. To add a single user, run the LDAP script with the **-w|-W** and **-u** options at a minimum. Optionally add the **-j**, **-k**, and **-o** options. The syntax for the **-u** option is `-u` *username* The full syntax with all options is:
   ```
   -w rootbindpw -u username -j groupID -k userID -o password
   ```

   -w|-W
   Specify the LDAP root bind password to authorize the changes that are specified on subsequent options. When you specify **-w**, the password is visible. When you specify **-W**, the password is hidden.

   -u Add a POSIX user to LDAP. A numeric LDAP user ID is assigned for the user unless you specify one with the **-k** option. Users are assigned to the default group ID of 1003 unless you specify the **-j** option.

   -j Assign an LDAP group ID. Use this option when creating users (**-u** option) or groups (**-g** option). LDAP group IDs are numeric. When you are creating users, the default group ID is 1003, which is the `bi_user_group`. Accept the default value unless you do not want the user to be able to access the InfoSphere BigInsights web console. The following groups are available by default, but none of these groups can log in to the InfoSphere BigInsights web console.

   biadmin
   123

   bi_sys_admin_group
   1000

   bi_data_admin_group
   1001

   bi_app_admin_group
   1002

   If you assign a group ID that does not exist, the group is created with the number that you specify. This group cannot access the InfoSphere BigInsights web console. When you use **-j** with the **-u** option, the default group name is the user name that you specify.

   -k Assign an LDAP user ID when you are creating a new user with the **-u** option. LDAP user IDs are numeric and some users were created for you already.

   -o Assign an initial password. As in other POSIX systems, if you do not supply a password, the user has no password and cannot be reached except by **su**.

   Run the script repeatedly to add multiple users. In this example, the new user, `testuser`, is added with the first available LDAP user ID above 500 and the default group ID `1003`. The group ID `1003` is the `bi_user_group`; members of this group can access the InfoSphere BigInsights web console.
   ```
   ehaas_ldap.sh -w rootbindpw -u testuser
   ```

In this example, the new user, `testuser`, is added with the first available LDAP user ID above 500 and the default group ID `1003`. The user is assigned the password `testuserpw`.

```
ehaas_ldap.sh -w rootbindpw -u testuser -o testuserpw
```

In this example, the new user, `testuser`, is added with the LDAP user ID `1000`, the LDAP group ID `1002`. The LDAP group ID `1002` is the `bi_admin_group`; members of this group cannot access the InfoSphere BigInsights web console, even if you subsequently add them to group `1003`. The user is assigned the password `testuserpw`.

```
ehaas_ldap.sh -w rootbindpw -u testuser -j 1002 -k 1000 -o testuserpw
```

4. Optional: Add one or more secondary groups to your cluster. To add a single group, run the LDAP script with the **-w|-W** and **-g** options at a minimum. Optionally, add the **-j** option. The syntax for the **-g** option is `-g groupname` The full syntax with all options is:

   ```
   -w rootbindpw -g groupname -j groupID
   ```

   -g Add a POSIX group to LDAP. When you specify this option, you must specify the POSIX group name. Optionally assign a group ID with the **-j** option. If you do not specify a group ID, the first available number over 500 is assigned and the new group does not have access to the InfoSphere BigInsights web console.

   To assign specific privileges to the new group, you must use PAM. Run the script repeatedly to add multiple groups. In this example, the new group, `testgroup`, is added with the first available group ID above 500. Members of this new group cannot access the InfoSphere BigInsights web console unless they are already a member of group `1003`.

   ```
   ehaas_ldap.sh -w rootbindpw -g testgroup
   ```

   In this example, the new group, `testgroup`, is added with the LDAP group ID `600`. Members of this new group cannot access the InfoSphere BigInsights web console unless they are already a member of group `1003`.

   ```
   ehaas_ldap.sh -w rootbindpw -g testgroup -j 600
   ```

5. Add new users to more groups. To add a single user to a single group, run the LDAP script with the **-w|-W** and **-a** options. The syntax for the **-a** option is `-a username:groupname`

   -a Add a POSIX user to an existing LDAP group. When you specify this option, you must specify the POSIX user name and the LDAP group name. Separate each value with a colon (:).

   Run the script repeatedly to assign the user to other groups or to assign other users to groups. In this example, the user `testuser` is added to the group `testgroup`.

   ```
   ehaas_ldap.sh -w rootbindpw -a testuser:testgroup
   ```

Results

Your new users and groups exist. Users can now log in to any node in the cluster. When a user logs in to a node for the first time, a home directory is created on that node. Users who belong to group 1003 (`bi_user_group`) can log in to the console.

Step 4: Configuring SSL connections

Configure Secure Sockets Layer (SSL) connections with Big SQL and the InfoSphere BigInsights console to ensure that your cluster is secure.

Procedure

1. Download the self-signed certificate that is provided by BigInsights for SSL server connections.
   a. Log in to master management node as `biadmin`.
   b. Export the certificate from the BigInsights keystore in PKCS#12 format. Run the following command:

      ```
      keytool -v -importkeystore -srckeystore /opt/ibm/biginsights/console/wlp/usr
      /servers/waslp-server/resources/security/biginsights.jks -srcalias
      biginsights -destkeystore /tmp/bi-cert.p12 -deststoretype PKCS12
      ```

      Specify `biadmin` as the password for both the source and destination keystores.
2. Apply the certificate to configure SSL for the JDBC connection with Big SQL.
3. Enable the browser for SSL connections with browser-based clients.

Configuring SSL connections for JDBC with Big SQL

Configure secure communication with the Big SQL database by applying the self-signed certificate that is provided by BigInsights or your own certificate.

About this task

Procedure

1. Log in to the InfoSphere BigInsights console to identify the node (host name) that is running the Big SQL head node. Go to the Cluster Status page and select **Big SQL** from the left navigation. On the right under **Big SQL Head Node**, find the host name for **Big SQL**.

   Tip: InfoSphere BigInsights 3.0 includes both Big SQL 1.0 and an all new Big SQL. This step refers to the all-new Big SQL, not to Big SQL 1.0. To understand the differences between Big SQL 1.0 and Big SQL, see Which version of Big SQL should you use? in the InfoSphere BigInsights documentation.
2. Open a Secure Shell environment (SSH) to the head node that you identified in step 1 as the user `bigsql`. The user `bigsql` is the Big SQL instance owner. Verify that current the directory is `/home/bigsql`.

Option	Description
If you are using the self-signed certificate that was provided by BigInsights	Copy the certificate on master management node to the `/tmp/db2sslcert` directory.
If you are using your own SSL certificate	Copy the certificate in PKCS#12 format to the `/tmp/db2sslcert` directory.

3. Set up the environment to run IBM Global Security Kit (GSKit) commands to generate the DB2® keystore.

   ```
   export PATH=$PATH:/home/bigsql/sqllib/gskit/bin
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/bigsql/sqllib/lib64/gskit:/home
   /bigsql/sqllib/lib32/gskit
   ```
4. Create the DB2 keystore.
   a. Create the directory (`ssl-keystore`) to store the keystore files.

      ```
      mkdir ssl-keystore
      ```
   b. Generate the keystore.

      ```
      gsk8capicmd_64 -keydb -create -db "/home/bigsql/ssl-keystore/keystore.kdb"
      -pw "passw0rd" -type cms -stash –empty
      ```
   c. Verify that the following files are created in `ssl-keystore` directory.

keystore.kdb, keystore.rdb, keystore.sth and keystore.crl

5. Import the SSL certificate from the /tmp/db2sslcert directory into the keystore that was created in the previous step.

```
gsk8capicmd_64 -cert -import -target "/home/bigsql/ssl-keystore/keystore.kdb"
-target_pw "passw0rd" -file "/tmp/db2sslcert/bi-cert.p12" -pw "biadmin"
```

   To verify, list the certificate in the keystore by using this command:

```
gsk8capicmd_64 -cert -list -db "/home/bigsql/ssl-keystore/keystore.kdb"
-pw "passw0rd"
```

   It lists a certificate with label biginsights.

6. Update SSL-related DB2 configurations.
   a. Specify keystore database to be used.

```
db2 update dbm cfg using SSL_SVR_KEYDB "/home/bigsql/ssl-keystore
/keystore.kdb"
```

   b. Specify the stash file to be used.

```
db2 update dbm cfg using SSL_SVR_STASH "/home/bigsql/ssl-keystore
/keystore.sth"
```

   c. Specify the label of certificate to be used.

```
db2 update dbm cfg using SSL_SVR_LABEL "biginsights"
```

   d. Specify the port (for example, 52000) where the SSL daemon listens.

```
db2 update dbm cfg using SSL_SVCENAME 52000
```

   e. Specify the DB2 communication protocol.

```
db2set DB2COMM=TCPIP,SSL
```

7. Refresh the DB2 configuration by restarting DB2.

```
db2stop force; db2start
```

8. Verify that a DB2 daemon is listening on port 52000.

```
netstat -nltp | grep 52000
```

9. Verify Big SQL health by running the BigInsights health check command from the master management node.

```
su – biadmin 'healthcheck.sh bigsql'
```

   All inbound public communication is blocked, except on ports 22, 8443, 10000, 14443, 51000, and 7052. You want to communicate with port 52000 where the SSL daemon is listening. Modify the iptables to enable communication on port 52000 using the public interface (eth1) of the head node that was identified in step 1.

```
su – biadmin
sudo iptables -I INPUT -i eth1 -p tcp --dport 52000 -j ACCEPT
sudo iptables -I INPUT -i eth1 -p udp --dport 52000 -j ACCEPT
```

Results

SSL communication with Big SQL is now configured.

Enabling your browser for SSL connections

BigInsights uses a web client (the InfoSphere BigInsights web console) and you want to ensure that client communications are using SSL.

About this task

On Firefox, go to the BigInsights web console. The first time you do this you see a message that the connection is untrusted. Click **Add Exception** to permanently trust the connection. You are then connected to the console.

For Chrome, complete the following steps to enable an SSL connection with the web console.

Procedure

1. Download the self-signed certificate that is provided by BigInsights for use with the web console.
   a. Open the InfoSphere BigInsights web console in your browser. You'll receive the message that `Your connection is not private`. In the address bar the lock icon has a red X over it and the https has a red strikethrough.
   b. In the address bar, click the lock icon. Identify information for the server is displayed in a pop-up dialog.
   c. Click the **Connection** tab and then click **Certificate information**.
   d. In the Certificate dialog, click the **Details** tab and then click **Copy to File**.
   e. In the Certificate Export Wizard, accept the default **DER encoded binary X.509 (.CER)**.
   f. Specify a **File name** and path and click **Next**.
   g. Click **Finish**.
   h. Close the dialog.
2. Apply the certificate in the browser so that it applies to the InfoSphere BigInsights web console.
   a. Open the menu and click **Settings**.
   b. At the bottom of the page, click **Show advanced settings**.
   c. Under **HTTPS/SSL** click **Manage certificates**.
   d. In the Certificates window, click **Trusted Root Certification Authorities** and click **Import**.
   e. In the Certificate Import Wizard, browse to select the certificate that you downloaded in step 1.
   f. Ensure that you are placing the certificate in the **Trusted Root Certification Authorities** store.
   g. Click **Finish**. A security warning displays. Click **Yes** to install the certificate.
   h. Click **Close**.
   i. Restart Chrome.

Managing user authentication

You can expire users' passwords, unlock locked users, and add or remove lockout protection to a user. By default, new users are created with passwords that expire and lockout protection. You can change the default LDAP settings by editing the `/opt/ehaas/bin/ehaas_ldap.conf` file.

Before you begin

After you log in, switch to `root` to run the script.

Procedure

- Manually expire a user's password. This option is valuable if you think that a user's password was compromised and you want to force a password change. To expire a single user's password before the time period specified on the DAYS_PASSWORD_VALID configuration parameter, run the LDAP script with the **-w|-W** and **-e** options. The syntax for the **-e** option is `-e` *username*

  -w|-W
  Specify the LDAP root bind password to authorize the changes that are specified on subsequent options. When you specify **-w**, the password is visible. When you specify **-W**, the password is hidden.

  -e Manually expire an LDAP user's password. At next login, the user will be forced to change the password.

  Run the script repeatedly to expire multiple passwords. In this example, the user `testuser`'s password is immediately expired. The next time `testuser` logs in, they are prompted to change the password.

  ```
  ehaas_ldap.sh -w rootbindpw -e testuser
  ```

- Unlock a locked user. By default, users can be locked out after three unsuccessful login attempts. To unlock a locked-out user, run the LDAP script with the **-w|-W** and **-f** options. The syntax for the **-f** option is `-f` *username*

  Run the script repeatedly to unlock multiple users. After running this example command, the user `testuser` can now log in to the system with the existing password.

  ```
  ehaas_ldap.sh -w rootbindpw -f testuser
  ```

- Add lockout protection to a user that currently cannot be locked out. By default, new users are created with lockout protection. If you've changed the default setting for the LOCKOUT_USER_ACCOUNTS configuration parameter, users can be created without lockout protection. You might want to reestablish protection for these users or for users from which you removed lockout protection.

  To add lockout protection to a single user, run the LDAP script with the **-w|-W** and **-l** options. The syntax for the **-l** option is `-l` *username*

  -l Add lockout protection to an LDAP user. By default, with this protection, users are locked out for 15 minutes after three unsuccessful login attempts. You can change the 15-minute default in the configuration file. LDAP users are created by default with lockout protection. If the default was changed or the lockout protection was manually removed, you can add it back for a specific user with this option.

  Run the script repeatedly to add lockout protection for multiple users. After running this example command, the user `testuser` can now be locked out.

  ```
  ehaas_ldap.sh -w rootbindpw -l testuser
  ```

- Remove lockout protection from a user that currently can be locked out after unsuccessful login attempts. By default, new users are created with lockout protection. You can change the default setting for the LOCKOUT_USER_ACCOUNTS configuration parameter to create subsequent users without lockout protection.

  To remove lockout protection from an existing user, run the LDAP script with the **-w|-W** and **-n** options. The syntax for the **-n** option is `-n` *username*

  -n Manually remove lockout protection from a specific LDAP user. For example, this option is valuable if you are changing passwords and the existing password is cached somewhere in the cluster, for example with the

`catalog` system user. You can temporarily remove the lockout protection so that you can change the password in all continuously running services where it needs to be reset.

Run the script repeatedly to remove lockout protection for multiple users. After running this example command, the user `testuser` can no longer be locked out.

```
ehaas_ldap.sh -w rootbindpw -n testuser
```

Deleting users or groups

You can delete users and groups by using the LDAP script.

Before you begin

After you log in, switch to `root` to run the script.

Procedure

- Delete a single user. To delete a single user, run the LDAP script with the **-w|-W** and **-d** options. The syntax for the **-d** option is `-d username`

  -w|-W
  Specify the LDAP root bind password to authorize the changes that are specified on subsequent options. When you specify **-w**, the password is visible. When you specify **-W**, the password is hidden.

  -d Delete a specified user from LDAP. The user is removed from all groups to which it belongs.

  Run the script repeatedly to delete multiple users. After running this example command, the user `testuser` is deleted from the system.

  ```
  ehaas_ldap.sh -w rootbindpw -d testuser
  ```

- Delete a single group. To delete a single group, run the LDAP script with the **-w|-W** and **-D** options. The syntax for the **-D** option is `-D username`

  -D Delete a specified group from LDAP. Included users are removed from the group, but are not deleted from the system.

  Run the script repeatedly to delete multiple groups. After running this example command, the group `testgroup` is deleted from the system.

  ```
  ehaas_ldap.sh -w rootbindpw -D testgroup
  ```

What is LDAP?

Although LDAP is often used by system administrators because of its unique and powerful combination of database and object-oriented features, learning the protocol can be a bit tedious and time-consuming. BigInsights provides a script to simplify your work with LDAP.

LDAP (Lightweight Distributed Access Protocol) manages distributed directory information over an IP network. *Directories* in LDAP refer to a collection of information or attributes of a particular object. (Each user, for example, is represented as an object.) This directory information is organized in a hierarchical, object-oriented structure that allows for inheritance and the retrieval of the information.

Your BigInsights cluster uses LDAP for user and group management. Your provisioned cluster includes all of the required BigInsights system users that are created as LDAP objects. These objects are delivered as a PAM (Pluggable Authentication Module). With the PAM, the operating system can recognize these users and groups as if they were created through the traditional POSIX methods.

A further benefit of LDAP is that after a user (represented in LDAP as a directory object) is created, that user is instantly "created" on all nodes in the cluster. The user's ID information is available on all nodes immediately. A home directory is created for the user on each individual node when they first log in to the node.

LDAP script reference

The LDAP script (`ehaas_ldap.sh`) and associated configuration file (`ehaas_ldap.conf`) are both stored in `/opt/ehaas/bin`. After you log in, switch to `root` to run the script. Run the script to create and delete users and groups and manage authentication.

LDAP script syntax

The following syntax is case-sensitive.

```
ehaas_ldap.sh -w|-W rootbindpw
[-u username]
[-j GID]
[-k UID]
[-o password]
[-g groupname]
[-a username:groupname]
[-p username]
[-r]
[-e username]
[-l username]
[-n username]
[-f username]
[-d username]
[-D groupname]
-c conf_filepath
```

-w|-W
Specify the LDAP root bind password to authorize the changes that are specified on subsequent options. When you specify **-w**, the password is visible. When you specify **-W**, the password is hidden.

-u Add a POSIX user to LDAP. A numeric LDAP user ID is assigned for the user unless you specify one with the **-k** option. Users are assigned to the default group ID of 1003 unless you specify the **-j** option.

-j Assign an LDAP group ID. Use this option when creating users (**-u** option) or groups (**-g** option). LDAP group IDs are numeric. When you are creating users, the default group ID is 1003, which is the `bi_user_group`. Accept the default value unless you do not want the user to be able to access the InfoSphere BigInsights web console. The following groups are available by default, but none of these groups can log in to the InfoSphere BigInsights web console.

biadmin
123

bi_sys_admin_group
1000

bi_data_admin_group
1001

bi_app_admin_group
1002

If you assign a group ID that does not exist, the group is created with the number that you specify. This group cannot access the InfoSphere BigInsights web console. When you use **-j** with the **-u** option, the default group name is the user name that you specify.

-k Assign an LDAP user ID when you are creating a new user with the **-u** option. LDAP user IDs are numeric and some users were created for you already.

-o Assign an initial password. As in other POSIX systems, if you do not supply a password, the user has no password and cannot be reached except by **su**.

-g Add a POSIX group to LDAP. When you specify this option, you must specify the POSIX group name. Optionally assign a group ID with the **-j** option. If you do not specify a group ID, the first available number over 500 is assigned and the new group does not have access to the InfoSphere BigInsights web console.

-a Add a POSIX user to an existing LDAP group. When you specify this option, you must specify the POSIX user name and the LDAP group name. Separate each value with a colon (:).

-p Change a user's password. Specify the user name. You are prompted to enter the existing password and a new password.

-r Change the LDAP root bind password. You are prompted to enter the existing password and a new password.

-e Manually expire an LDAP user's password. At next login, the user will be forced to change the password.

-l Add lockout protection to an LDAP user. By default, with this protection, users are locked out for 15 minutes after three unsuccessful login attempts. You can change the 15-minute default in the configuration file. LDAP users are created by default with lockout protection. If the default was changed or the lockout protection was manually removed, you can add it back for a specific user with this option.

-n Manually remove lockout protection from a specific LDAP user. For example, this option is valuable if you are changing passwords and the existing password is cached somewhere in the cluster, for example with the `catalog` system user. You can temporarily remove the lockout protection so that you can change the password in all continuously running services where it needs to be reset.

-f Manually unlock a locked LDAP user. The user password remains the same.

-d Delete a specified user from LDAP. The user is removed from all groups to which it belongs.

-D Delete a specified group from LDAP. Included users are removed from the group, but are not deleted from the system.

-c The `ehaas_ldap.conf` file is also in `/opt/ehaas/bin/`. If you decide to move the file to a different directory from the script, you can use the **-c** option. The **-c** option specifies where to find the `.conf` file. For example:`-c /opt/ehaas/conf/ehaas_ldap.conf`

LDAP configuration parameters

LDAP is configured with a set of default values. You can change the default LDAP settings by editing the `/opt/ehaas/bin/ehaas_ldap.conf` file.

LDAP parameters that you can change

If you change the following parameters, the changes affect newly created users. Changing these values does not retroactively change the settings for users who were created earlier.

EXPIRE_USER_PASSWORDS
When this parameter value is `true`, newly created users have passwords that expire after the number of days that are specified on the DAYS_PASSWORD_VALID parameter. Users must change their passwords before they expire. This parameter value does not affect system passwords.

The default value is `true`.

DAYS_PASSWORD_VALID
Specifies the number of days passwords are valid before they expire. This parameter is used when EXPIRE_USER_PASSWORDS is `true`.

The default value is 90 days. If you change this value, it does not affect system users. There is no maximum value, although changing passwords frequently is recommended for security.

DAYS_WARN_BEFORE_EXPIRATION
When users log in to BigInsights, they are warned that their passwords are to expire. The number on this parameter is subtracted from DAYS_PASSWORD_VALID. If a user does not change a password during the warning period, the user is forced to do so the next login after expiration.

The default value is 15 days. There is no maximum value for the number, although make it consistent with DAYS_PASSWORD_VALID.

LOCKOUT_USER_ACCOUNTS
When this parameter value is `true`, newly created users have the lockout password policy set by default. By default, that policy allows users three attempts to log in with an invalid password before the user is locked out. After users are locked out, the lockout remains in effect for 15 minutes.

The default value is `true`.

LOG_FILE
Specifies the location of the log files that are produced by the script when it runs. By default these files are stored in `/tmp/bi_ldap/log`.

Default LDAP parameters

These parameters are set only when your cluster is installed and generally do not change. The information below is provided for your reference. If you'd like to upgrade your cluster to High Availability (HA), contact IBM Support.

LDAP_NODES
Lists all nodes in the cluster.

LDAP_MASTER_HOSTNAME
The fully qualified domain name of the "MasterManger" node, the initial LDAP master node.

LDAP_MASTER_STANDBYS
If your cluster is configured for HA, this parameter contains a comma-separated list of LDAP standby nodes. These nodes instantly receive all updates from the current LDAP master node and are ready to take over LDAP operations if the main LDAP server goes down.

IP_FAILOVER
If your cluster is configured for HA, the parameter is set to `true`. If your cluster is not configured for HA, this parameter value is `false`.

VIRTUAL_IP
If your cluster is configured for HA, this parameter value is the virtual IP to use.

VIRTUAL_INTERFACE
If your cluster is configured for HA, this parameter value is the network interface on which to listen for the VIRTUAL_IP. This interface is usually the private network interface. The default value is set by IBM SoftLayer if you do not specify a value (currently `eth0`).

EXPIRE_SYSTEM_PASSWORDS
When this parameter value is `true`, default system accounts (`biadmin`, `catalog`, `bigsql`, and so on) have passwords that expire after 90 days. The default value is `true`.

LOCKOUT_AFTER_FAILED_ATTEMPTS
After this number of unsuccessful login attempts, users are locked for the number of minutes specified by the LOCKOUT_MINUTES parameter. The current default number of attempts is three.

LOCKOUT_MINUTES
The number of minutes a user is locked out. The current default value is 15 minutes. An administrator can run the LDAP script (`ehaas_ldap.sh`) with the **`-f`** option to remove a lock before the number of minutes is reached.

LOCKOUT_FAILURE_INTERVAL_IN_SECS
The amount of time, in seconds, within which failed login attempts are tallied and compared to the number set on the LOCKOUT_AFTER_FAILED_ATTEMPTS parameter. The current default value is 120.

Importing data

You can import data into BigInsights by either pushing or pulling with various tools.

Recommended tools for importing data

Some tools or services are better suited for certain types of data than for others. Before you determine which one to use, carefully consider the availability of the data that you want to import into BigInsights.

Table 1. Recommended tools for importing data into BigInsights

If you want to import:	The recommended way to import the data, depending on size:
Data at rest • Flat files • Semi-structured files • Unstructured files	• Aspera Point-to-Point (recommended for large data sets, consisting of one file, or many large files, or many small files) • Distributed File Copy application (for files greater than 2 GB) • Distribution Copy (distcp) (for files greater than 2 GB) • Hadoop shell commands: copyfromlocal and copytolocal (for files greater than 2 GB) • InfoSphere BigInsights Console Files page Import / Export • Web REST Import application (limited to files smaller than 2 GB) • Web Crawler application • HttpFS
Data in motion • Semistructured data • Constantly updated data • Web server of application logs • Twitter data	• Flume. You can add similar Flume configurations to push data. • Import from Twitter Decahose. Import a 10% sample of data from Twitter.
Data from a data warehouse • Structured data • Relational tables	• Database Import application You can use this application with the related Database Export application. • Sqoop: DB2, Netezza®, other databases
Data from Swift object store	Import Export Object Store application (for files smaller that 4.5 GB)

Push and pull methods

With push and pull, the direction of data flow is relative to the initiator, whereas import and export refer to absolute direction from a single location. *Pull* brings data toward the initiator; *push* sends data away from the initiator. *Import* moves data into BigInsights and *export* moves data out of BigInsights.

BigInsights supports importing and export data using both push and pull methods.

Import by Push
Writes data into BigInsights from an external system. The external system initiates the request.

Import by Pull
Reads data into BigInsights from an external system.BigInsights initiates the request.

Export by Push
Writes data into an external system from BigInsights. BigInsights initiates the request.

Export by Pull
Read data into an external system from BigInsights. The external system initiates the request.

Sample applications for importing data into BigInsights

Several sample applications are provided for copying data within BigInsights or pulling data into BigInsights. From BigInsights, open the InfoSphere BigInsights console and you can see them on the Applications page.

Sample executions of the Distributed File Copy application and the Web Crawler application are provided in BigInsights. You can review these sample executions to see how to pull data into BigInsights. Although the run samples are available only on BigInsights, the sample applications are also available in InfoSphere BigInsights if you have it installed on premises.

If you installed InfoSphere BigInsights, you can use the information from the following sample executions to push data from InfoSphere BigInsights into BigInsights.

Distributed File Copy application
The following named version of this sample was run for you on BigInsights: **Pull data into HDFS**.This run sample copies data from the BigInsights master node to BigInsights HDFS. For example, this sample copies a BigInsights log file to HDFS. (This sample is also used to complete the data import using Aspera when you copy the data from the staging node to HDFS.)

See the full InfoSphere BigInsights documentation for additional information about Distributed File Copy application.

Web Crawler application
This sample execution, Pull data into HDFS from web sample, uses the Web Crawler sample application to import specified web pages into BigInsights HDFS.

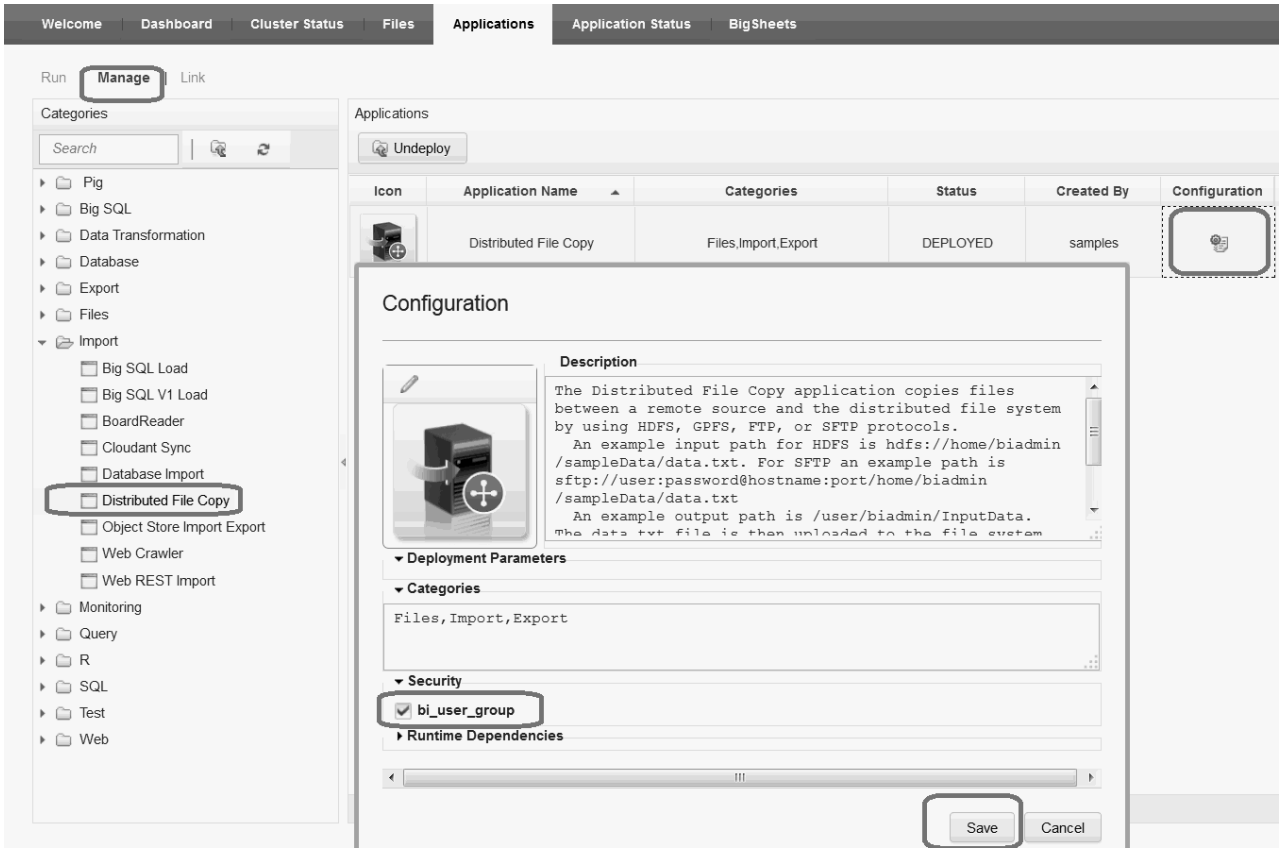
Import Export Object Store
This sample application is available to copy data between HDFS and a cloud-based object store using HDFS and Swift protocols. You can also use this application to back up your data from HDFS into Swift storage before you delete your cluster.

Restriction: Use for files smaller than 4.5 GB.

Web REST Import application
This sample execution named Pull data into HDFS using webRESTImport, uses the Web REST Import application to import data from a specified REST URL into BigInsights HDFS.

These applications are all deployed and ready to run. Change the security settings for the applications:

1. On the Applications page in the console, click **Manage**.
2. In the left navigation, select the application that you want to change.
3. On the right side, click **Configuration**.
4. In the Configuration window, expand **Security** and then select **bi_user_group**.
5. Click **Save**.



Pushing data into BigInsights

Pushing data into BigInsights

Pushing data into BigInsights

Use one of several methods to push data from an external system into BigInsights or from one location to another within BigInsights.

Procedure

- Copy object store data on the cloud. Use this application is to copy data from one BigInsights cloud location (stored in HDFS) to a Swift-based object store or vice versa.
- Push data with Aspera Point-to-Point. This method is the best for importing your largest data sets, whether you have a one or many large files or one or many small ones.
- Push data with the Distributed File Copy sample application. For example, from the HDFS of installed InfoSphere BigInsights Enterprise Edition to BigInsights or from HDFS of one BigInsights instance to another cloud instance. Use this method if you have files that are greater than 2 GB. The linked instructions describe how to start from BigInsights and copy or pull data from there.
- Import data in motion with Flume. Continuously push data from an external system into BigInsights.

Copying object store data on the cloud

Use the BigInsights Import Export Object Store application to copy your data between a cloud-based Swift object store and HDFS. The Import Export Object Store application is available in the InfoSphere BigInsights console and uses HDFS and Swift protocols. Files larger than 4.5 GB cannot be copied with this application.

About this task

As in the final task for pushing data with Aspera, you can use the Import Export Object Store application to copy data between HDFS and the OpenStack Swift based object store on IBM SoftLayer. Common scenarios might be:

- Backing up HDFS data into the Swift object store.
- Pushing or pulling data from external systems into Swift, then using this application to copy the Swift data into HDFS.

The Import Export Object Store application is automatically deployed on BigInsights. In the InfoSphere BigInsights console, you can see it on the Applications page.

Procedure

1. Create or update a credentials file with your object store credentials. Use the InfoSphere BigInsights credstore.sh utility to store credentials for the Swift object store that you plan to copy data to or from.
2. Log in to the console and go the Applications page.
3. Under **Applications** in the left navigation area, click the **Import Export Object Store** icon. On the right, the Import Export Object Store application input form displays.
4. Follow the instructions in the application to provide input and output targets, depending on whether you are copying from HDFS to the object store or vice versa.
5. Specify the name of the object store credentials file that you created.
6. Click **Run**.

Pushing data with Aspera Point-to-Point

Aspera Point-to-Point client is the best method for importing your largest data sets, whether you have many large files or many small ones.

About this task

Aspera provides several products for importing data, such as a Aspera Desktop Client, Aspera Point-to-Point, and others. You can use different Aspera software; contact Aspera Sales to obtain the right product for your business.

The following tasks illustrate how to use the Aspera Point-to-Point client to push data onto a staging node within BigInsights and from there into HDFS on BigInsights, but your staging node can be outside of BigInsights.

Procedure

1. Install and configure Aspera software.
2. Push data onto a staging node.
3. Copy data from the staging node to HDFS.
4. Optional: Delete the data from the staging location.

Installing and configuring Aspera software

Install and configure the Aspera software onto a staging node on the cloud (target) and the external machine where the data is stored that you want to import (source).

About this task

The following steps describe how to use Aspera Point-to-Point to import data into BigInsights. The staging node can be either a BigInsights node or an edge node on the cloud. The steps assume that the staging node is a BigInsights node.

Procedure

1. Configure BigInsights with Aspera Point-to-Point client.
   a. Identify the IP address for the node that you want to use as a staging location in BigInsights. You can delete the data from this staging location after you complete the whole process.
   b. Log in to the node with `root/sudo` privileges.
   c. Download Aspera Point-to-Point client from the link that you received from Aspera Sales. The Aspera site prompts you to enter a user name and password.
   d. Install Aspera Point-to-Point client.
      ```
      rpm -Uvh aspera-scp-p2p-[version].rpm
      ```
   e. Apply the license that you received from Aspera Sales. Copy the license file into `/opt/aspera/etc`. In a terminal window, enter the following command to verify that the license was applied:
      ```
      ascp –A
      ```
      You get a confirmation that the license is installed and applied.
   f. Open a communications port. These commands illustrate the use of the default port. If you prefer to use a different port, see the Aspera documentation.
      ```
      iptables -I INPUT -p tcp --dport 33001 -j ACCEPT
      iptables -I INPUT -p udp --dport 33001 -j ACCEPT
      ```
   g. Create a directory on the BigInsights staging node to receive the data. For example, `/incoming/staging`.

   BigInsights is configured for data transfer with Aspera Point-to-Point client.
2. Install Aspera Point-to-Point Advanced Desktop Transfer Client software on an external client machine. Follow the instructions in the Aspera documentation.

Pushing data onto a staging node

Push data from the external client machine onto the staging node in BigInsights. Your staging node might be outside of BigInsights, but this scenario assumes that it is inside.

Procedure

- On Windows:
  1. Click **Start** > **Aspera Point-to-Point**.
  2. Click **Connections** in the toolbar and create a connection to the BigInsights staging node. Use your BigInsights credentials (biadmin is the user and the password is the one you set for it).

3. Optional: Set up for an encrypted data transfer from the external client machine. On the Security page, select **Encrypt data in transit**.
4. Select the directory on the BigInsights staging node where you want to store the data. For example, `/incoming/staging`.
5. Click **Preferences** in the toolbar and under **Transfers**, configure the bandwidth to the maximum available. For example, change the **Default Target Rate** from **45 Mbps** to `1000 Mbps` or more.
6. Click **Connect** to connect to theBigInsights staging node.
7. Click the arrows to transfer data from the external Windows client (left) to theBigInsights staging node (right).

- On Linux:
  1. Ensure that the directory on theBigInsights staging node exists to receive the data. For example, `/incoming/staging`.
  2. From a terminal window, connect to the external machine that contains the data that you want to push onto the BigInsights staging node.
  3. Push data from the Linux machine to the BigInsights staging node. If you want to encrypt data during transfer, enter:

     ```
     ascp -T --policy=fair -l 1000m -m 1m /var/log/secure
     biadmin@IP_address_BigInsights_staging_node:/incoming/staging
     ```

     If you do not want to encrypt data, remove the `-T` option from the command.

Results

Your data was pushed to theBigInsights staging node.

Copying from the staging location to HDFS

Use the BigInsights Distributed File Copy sample application to copy your data from the staging node to HDFS. The Distributed File Copy sample application is available in the InfoSphere BigInsights console, and uses a MapReduce job to copy data.

About this task

The Distributed File Copy sample application is automatically deployed on BigInsights. In the InfoSphere BigInsights console, you can see it on the Applications page.

Procedure

1. Log in to the console and go the Applications page.
2. Under Applications in the left navigation area, click the **Distributed File Copy** icon. On the right, application details display, including when the application was run.
3. Use the values in the run sample **Pull data into HDFS** as a guide. Create an execution of **Distributed File Copy** to copy your data from your staging node to HDFS. This run sample copies data from the BigInsights master node to BigInsights HDFS. For example, this sample copies a BigInsights log file to HDFS.
4. Click **Run**.

Importing data in motion with Flume into BigInsights

Flume is best for importing data in motion. Use it from an external, source machine to push data into BigInsights. (Use it from BigInsights to pull data from an external, source system.)

Before you begin

The example assumes that the IBM Java™ version 1.6 SR 11 JVM or newer is installed. Ensure that the source machine has Flume installed on it.

About this task

Flume is installed for you on BigInsights. You must configure the source machine to run a Flume agent. You must start a Flume collector agent on the BigInsights master node to communicate with the source machine's Flume agent.

Procedure

1. On the BigInsights console node, open a port for Flume to communicate with the external, source machine.

   ```
   iptables -I INPUT -p tcp --dport 40101 -j ACCEPT
   iptables -I INPUT -p udp --dport 40101 -j ACCEPT
   ```

   BigInsights is configured for Flume.
2. On the external, source machine, configure the Flume agent.
   a. Get the Flume runtime files. Log in to the BigInsights web console, and under **Quick Links** click **Download client library and development software**. Click **Flume runtime**.
   b. Copy and extract the Flume runtime file onto the external, source machine.
   c. Copy the following code into a file on the external, source machine in the path *flume_runtime_installation_location*/conf. Edit the file to set the preferred source data location and console node host name (both in italics in the following code). In this sample code, the Flume agent is named sendData.

```
# list sources, sinks and channels in the agent
sendData.sources = spooldir
sendData.channels = c1
sendData.sinks=avro-sink

# define the flow
sendData.sources.spooldir.channels = c1
sendData.sinks.avro-sink.channel = c1
sendData.channels.c1.type = memory
sendData.channels.c1.capacity = 1000

# define source and sink
sendData.sources.spooldir.type = spooldir
sendData.sources.spooldir.spoolDir=/var/log/apache/flumespool
sendData.sources.spooldir.channels = c1
sendData.sinks.avro-sink.type = avro
sendData.sinks.avro-sink.hostname =
name_for_BigInsights_console_node
sendData.sinks.avro-sink.port = 40101
```

3. On BigInsights, configure the Flume agent to pull data.
   a. Copy the following code into a file that is named hdfs_remoteSpool_config.properties on the BigInsights console node, in the path *console_node*/$BIGINSIGHTS_HOME/flume/conf. Edit the file to set

the names for the BigInsights console and HDFS nodes (both in italics in the following code). In this sample code, the Flume agent is named `receiveData`.

```
# list sources, sinks and channels in the agent
receiveData.sources = avro-collection-source
receiveData.sinks = hdfs-sink
receiveData.channels = mem-channel

# define the flow
receiveData.sources.avro-collection-source.channels = mem-channel
receiveData.sinks.hdfs-sink.channel = mem-channel
receiveData.channels.mem-channel.type = memory
receiveData.channels.mem-channel.capacity = 1000

# avro source properties
receiveData.sources.avro-collection-source.type = avro
receiveData.sources.avro-collection-source.bind = name_for_console_node
receiveData.sources.avro-collection-source.port = 40101

# hdfs sink properties
receiveData.sinks.hdfs-sink.type = hdfs
receiveData.sinks.hdfs-sink.hdfs.fileType = DataStream
receiveData.sinks.hdfs-sink.hdfs.writeFormat = Text
receiveData.sinks.hdfs-sink.hdfs.rollCount = 0
receiveData.sinks.hdfs-sink.hdfs.rollInterval = 0
receiveData.sinks.hdfs-sink.hdfs.rollSize = 10000000
receiveData.sinks.hdfs-sink.hdfs.batchSize = 100
receiveData.sinks.hdfs-sink.hdfs.path =
hdfs://name_for_HDFS_node:9000/tmp/flume/remoteSpool
```

4. Run the agents.
   a. Connect to the BigInsights console node and start the receiveData agent. Run the command:

      ```
      /opt/ibm/biginsights/flume/bin> ./flume-ng agent  -f
      /opt/ibm/biginsights/flume/conf/hdfs_remoteSpool_config.properties
      -Dflume.root.logger=INFO,console -n receiveData
      ```

   b. Connect to the external, source machine and start the sendData agent. Run the command, substituting the variables in italics:

      ```
      flumeruntime_install_location/bin> ./flume-ng agent  -f
      flumeruntime_install_location/conf/spool_conf.properties
      -Dflume.root.logger=INFO,console -n sendData
      ```

   c. Test. On the external, source machine, copy any data to send to BigInsights into the Flume spool directory. In this example, the directory is `/var/log/apache/flumespool`. The BigInsights HDFS location receives the data. In this example, to directory `/tmp/flume/remoteSpool`.

What to do next

Customize sendData to collect data from various sources.

Pulling data into BigInsights

Use one of these methods from BigInsights to pull data from an external location. In addition to the methods described here, you can also use Flume from BigInsights to pull data in motion.

Pulling data with the Distributed File Copy sample application

Pull data into BigInsights from another instance of BigInsights or an external system. You can also push data from InfoSphere BigInsights installed on premises. Use this method for files that are greater than 2 GB.

About this task

The Distributed File Copy sample application is automatically deployed on BigInsights. In the InfoSphere BigInsights console, you can see it on the Applications page.

Procedure

1. Log in to the console and go the Applications page.
2. Under Applications in the left navigation area, click the **Distributed File Copy** icon. On the right, application details display, including when the application was run.
3. Examine the run sample called Pull data into HDFS. Reuse the sample as needed to copy your data. This run sample copies data from the BigInsights master node to BigInsights HDFS. For example, this sample copies a BigInsights log file to HDFS. (This sample is also used to complete the data import using Aspera when you copy the data from the staging node to HDFS.)
4. Click **Run**.

Importing data in motion from the Twitter Decahose

Use the Import from Twitter Decahose application in the InfoSphere BigInsights console to pull real-time data into BigInsights from Twitter.

Before you begin

Contact IBM Support to get the Twitter properties file that you need to run this application. After you receive the encrypted file, store it in `/user/`*`username`*`/credstore/private/`. (*username* must be the user who plans to run the application.) To store the file, log in to the InfoSphere BigInsights console and go to the **Files** page. Go to `/user/`*`username`*`/credstore/private/` and click the **Upload** button to load the file into the DFS.

About this task

The Twitter data is a random sample of 10% of the Twitter data stream (hence, Decahose) at the time that you run the application from BigInsights.

Procedure

To pull data from the Twitter Decahose:

1. Log in to the InfoSphere BigInsights console and go the Applications page.
2. Under **Applications** in the left navigation area, click the **Import from Twitter Decahose** icon. On the right, the Import from Twitter Decahose application input form displays.
3. In the **Properties file** field, browse to location of the Twitter properties file. By default the file is `twitter_decahose_account` in the path `/user/`*`username`*`/credstore/private/`. If the file does not exist, contact IBM Support.

4. In the **Output path** field, browse to the DFS location where you want to store the Twitter data. The data is stored as DAT or GZIP files. Every 15 minutes, the following three types of files are generated:
   - Decahose data
   - Deleted Tweets
   - Scrub Geo

   Files are named according to time stamps at the beginning and ending of each file.

   Attention: This application returns a continuous stream of data until you manually stop it. Every 15 minutes, three large files are generated. Ensure that you have the necessary space to store this data before you run the application.
5. In the **Compress data** field, enter `Yes` to compress the received data in GZIP format. Generally, a single day of uncompressed data is from 100 GB to 200 GB, but compressed data is approximately 10 GB to 20 GB.

   To decide whether to compress your data, consider both your storage capacity and the relative performance of loading compressed files and decompressing compressed files. Compressed data might take longer to process, so you might want to leave data uncompressed if your server has the capacity to store the data. The smaller file size of the compressed files might mean they take less time to load from storage, even though they might take longer to open.
6. Run the application. Ignore the **Schedule and Advanced settings** section of the application details because the options here are not applicable for this application. This application runs continuously until you stop it. Do not schedule it to run because you might start more than one simultaneous execution of an application that runs continuously. The application continuously downloads Twitter data until you manually stop the application. If the application fails, you can view the errors from the Application History. Scroll down to the **Application History** and in the **Details** column, click the icon next to the failed execution of the application.
7. Stop the application at any time. Scroll down to the **Application History** and click the red stop button in the **Status** column for the running application.

Results

Browse to the output location to locate the files. The files contain JSON data. Each JSON data object is on a new line.

Pulling data from the web

Pull data into BigInsights from the web, either from specific web page URLs or from a REST URL.

About this task

The sample execution, Pull data into HDFS from web sample, uses the Web Crawler sample application to load specified web pages into BigInsights HDFS. The Web Crawler sample application is automatically deployed on BigInsights. In the InfoSphere BigInsights console, you can see it on the Applications page.

Procedure

1. Log in to the console and go the Applications page.
2. Under **Applications** in the left navigation area, click the **Web Crawler** icon. On the right, application details display, including when the application was run.

3. Examine the run sample. Copy or reuse the sample as needed to load your data. The executed sample loads a page from a specified URL. You can add URLs as needed.

   To load data from a Web REST URL, copy the sample and replace the URL with a valid REST URL.

4. Click **Run**.

Index