First steps for your IBM® BigInsights on Cloud cluster

Follow the steps in this document to secure your cluster. You will change the initial passwords using LDAP, establish SSL communications, and change the root SSH (Secure Shell) keys. This document also provides an overview and a set of ongoing security tasks that you can complete using LDAP.

LDAP Overview

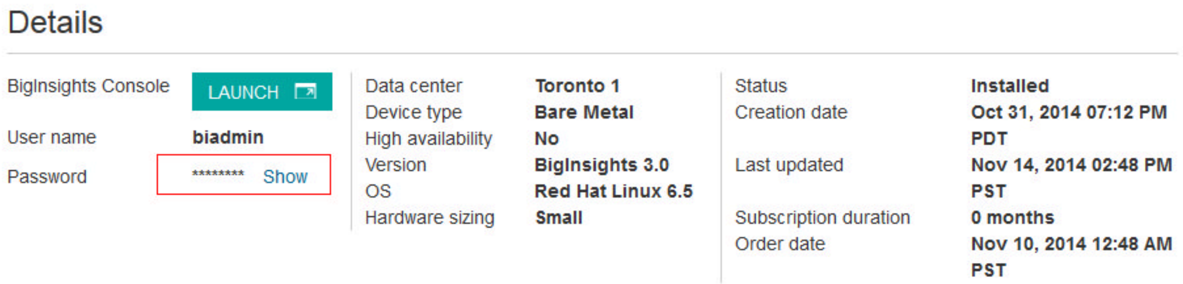
LDAP (Lightweight Distributed Access Protocol) is used over an IP network to manage distributed directory information. Directories in the LDAP sense refer to a collection of information—attributes—that pertain to the particular object being examined. This directory information is organized in a hierarchical, object-oriented structure that allows for inheritance and the easy retrieval of directory information.

Your BigInsights™ on Cloud cluster utilizes LDAP for user and group management. Your provisioned cluster includes all necessary BigInsights system users created as LDAP objects and tied in as a PAM (Pluggable Authentication Module), which allows the OS to recognize these users and groups just as if they had been created through the traditional POSIX methods that you're likely familiar with. 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 will be available on all nodes immediately and a home directory is created for the user on each individual node when they first log in to the node.

Though LDAP is often used by system administrators due to its unique and powerful combination of database and object-oriented features, learning the protocol can be a bit tedious and time consuming. If you're not yet an LDAP pro, no worries! Your BigInsights on Cloud cluster comes preloaded with an administration script that can greatly simplify your interactions with the service. This guide covers some of the common things that you can do with the script to ensure you're spending less time on administrative tasks and more time getting the most out of your powerful BigInsights on Cloud cluster. We recommend reading the next 'Getting Started' section and then jumping around to whatever operations you're looking to perform.

Step 1: Find the biadmin password and necessary scripts

When you first receive your cluster, navigate to the cluster Details page to obtain your cluster information, including the biadmin user password (pictured below). This password doubles as the initial LDAP root bind password for the cluster.



Details

BigInsights Console	LAUNCH	Data center	Toronto 1	Status	**Installed**
		Device type	**Bare Metal**	Creation date	**Oct 31, 2014 07:12 PM PDT**
User name	**biadmin**	High availability	**No**		
Password	******** Show	Version	**BigInsights 3.0**	Last updated	**Nov 14, 2014 02:48 PM PST**
		OS	**Red Hat Linux 6.5**		
		Hardware sizing	**Small**	Subscription duration	**0 months**
				Order date	**Nov 10, 2014 12:48 AM PST**

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). This 'root' password allows you to 'bind', or connect, to the LDAP hierarchical structure as the root user with full permissions to modify any aspect of the tree-like LDAP directory structure.

To navigate to the cluster Details page:

1. From your IBM Bluemix™ dashboard, click the BigInsights service.
2. From the BigInsights service page, click **Manage Clusters**.
3. In the table your clusters, find the cluster you want. 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.

Now that you've found the LDAP bind password, the next step is to locate the LDAP administration script (ehaas_ldap.sh). BigInsights on Cloud by default places the ehaas_ldap.sh script in the /opt/ehaas/bin directory.

The /opt/ehaas/conf directory includes the ehaas_ldap.conf file, which contains the LDAP configuration information related to your cluster. Because this directory is different from the one that contains the LDAP administration script, you must specify the full path name when running the configuration script by applying the –c flag as shown in the tasks below.

The default configuration information for LDAP is as follows:

LDAP installation parameters. These parameters generally should not be changed. If you'd like to upgrade your cluster to be Highly Available, contact us.

- LDAP_NODES contains a listing of all nodes in the cluster.
- LDAP_MASTER_HOSTNAME should be set to the FQDN of the 'MasterManger' node - the initial LDAP master node.
- LDAP_MASTER_STANDBYS if your cluster is configured for HA, this variable will contain a comma-separated list of LDAP standby nodes. These nodes will instantly receive all updates from the current LDAP master and will be ready to take over LDAP operations should the main LDAP server go down.
- IP_FAILOVER this variable should be set to 'true' or 'false' indicating whether or not the cluster is configured for HA with a failover IP address.
- VIRTUAL_IP if configured for HA, this variable will correspond to the virtual IP to be used.

- VIRTUAL_INTERFACE the network interface on which to listen for the VIRTUAL_IP, should generally be the private network interface.
- EXPIRE_SYSTEM_PASSWORDS if set to ‘true’, default system accounts (biadmin, catalog, bigsql, and so on) will feature expiring passwords.

LDAP ‘run time’ parameters:

- EXPIRE_USER_PASSWORDS if set to ‘true’, newly created users will have to reset their passwords every ‘DAYS_PASSWORD_VALID’ number of days.
- DAYS_PASSWORD_VALID if passwords are set to expire, they will expire periodically according to the number specified here.
- DAYS_WARN_BEFORE_EXPIRATION upon login, users will see a warning that their passwords are to expire soon beginning by this number of days subtracted from ‘DAYS_PASSWORD_VALID’. If users do not change their passwords during this ‘warning’ period they will be forced to upon their next log in after ‘DAYS_PASSWORD_VALID’
- LOCKOUT_USER_ACCOUNTS if set to ‘true’, newly created users will have the lockout password policy set up by default. By default, users are allowed three attempts to log in with an invalid password before being locked out. After they are locked out, the lockout remains in effect for fifteen minutes.

Note that the variables specified above as ‘run time’ will affect newly created users. They are not retroactive and will not modify values set for previously-created users.

Step 2: Change initial passwords

Changing user passwords

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

1. LDAP Admin (admin)
2. BigInsights administrator (biadmin)
3. BigInsights catalog user (catalog)
4. Big SQL user (bigsql)

Change these user passwords as soon as possible to properly secure your cluster.

To change a user’s password for them, you can invoke the LDAP script with the ‘-w/-W’ option to specify the root bind password and then the ‘-p’ option to indicate the user whose password you wish to change.

So to change the password of user ‘testuser’ you would issue the following command:

```
./ehass_ldap.sh -w <rootBindPW> -p testuser -c /opt/ehaas/conf/ehaas_ldap.conf
```

Note that changing certain system user passwords (biadmin, catalog, and bigsql) requires you to run other BigInsights scripts afterwards so that all BigInsights daemons get updated with the new passwords. This process runs automatically when you use the ‘-p’ option of the

LDAP script, but must be done manually if you change those passwords with ‘passwd’. *It is therefore highly recommended to change these system passwords with the LDAP script*.

Regular users can change their passwords just as normal—by executing the ‘passwd’ command, because your cluster’s LDAP installation is integrated with PAM.

Changing the root bind password

The LDAP root bind password is every bit as powerful as the normal root password into a system and should be protected as such. You should change the initial root password as soon as you can to properly secure your cluster.

To change the LDAP root bind password now, and periodically in the future (it will not expire on its own), you can use the ‘-r’ option to the LDAP script:

```
./ehaas_ldap.sh –r -c /opt/ehaas/conf/ehaas_ldap.conf
```

You will be prompted for the current LDAP root bind password and then what you’d like to change it to.

Changing the root password by changing the SSH keys

As part of initial configuration, change your root password for all nodes in your cluster.

1. Log in as biadmin user and sudo su to root.
2. Once when you are logged in as root, enter:

   ```
   passwd
   ```

   and update the password for all the nodes.

3. From the root user, back up the root ssh key folder:

   ```
   mv /root/.ssh /root/.ssh_backup
   ```

   The following changes will remove all authorized keys from establishing SSH communication to your system and also will change your keys if you have shared anything to a remote system.

4. Create new SSH keys:

   ```
   ssh-keygen -t rsa
   ```

   Accept the default values for all.

5. Optional: To establish passwordless SSH as root across nodes in your cluster with your new keys and thus enable passwordless SSH with other nodes in your cluster, share the newly generated public key to the communication node. For example, to establish passwordless SSH between node X and node Y, from node X, run:

   ```
   ssh root@Y "cat >>/root/.ssh/authorized_keys" < /root/.ssh/id_rsa.pub
   ```

   If cat is not in path, find the correct cat path and append, for example:

   ```
   ssh root@Y "/sbin/cat >>/root/.ssh/authorized_keys" < /root/.ssh/id_rsa.pub
   ```

Step 3: Add new users

To add new users to the LDAP service, invoke the LDAP script with the '-w/-W' option to specify the root bind password and then the '-u' option to indicate the attributes of the user to be created.

The format of arguments to '-u' is as follows:
-u <username>:<UID>:<GID>:<password>

To create a new user 'testuser' with a UID of '3333' and a GID of '1003' (corresponding to the bi_user_group, which *will allow the new user to access the BigInsights web console*) you run the following command:

```
./ehaas_ldap.sh -w <rootBindPW> -u testuser:3333:1003:<testuserPassword> -c
/opt/ehaas/conf/ehaas_ldap.conf
```

Note that the last segment of the '-u' argument specifies a default password for the created user and is not mandatory.

Step 4: Add new groups

To add a new group to your cluster, invoke the LDAP script with the '-w/-W' option to specify the root bind password and then the '-g' option to indicate the name and GID of the group to be added.

The format of arguments to '-g' is as follows:
-g <groupname>:<GID>

To create a new group 'testgroup' with a GID of '1111' you run the following command:

```
./ehaas_ldap.sh -w <rootBindPW> -g testgroup:1111 -c /opt/ehaas/conf/ehaas_ldap.conf
```

Step 5: Add users to a group

So you've created a user (Step 3) and a group (Step 4) and now want to add the user to the group. To do so, run the LDAP script with the '-w/-W' option to specify the root bind password and then the '-a' option to indicate you'd like to add the user to the specified group.

To format of arguments to '-a' is as follows:
-a <username>:<groupname>

To add the user 'testuser' to the group 'testgroup', run the following command:

```
./ehaas_ldap.sh -w <rootBindPW> -a testuser:testgroup -c /opt/ehaas/conf/ehaas_ldap.conf
```

Step 6: Configure Secure Sockets Layer (SSL) support for JDBC connections with Big SQL 3.0

BigInsights 3.0 includes both Big SQL 1.0 and an all new Big SQL. The steps below refer to the all-new Big SQL and any user interface references are to 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 IBM Knowledge Center.

BigInsights provides a self-signed certificate for SSL. If you want to download your own certificate, you can.

To secure your JDBC connection with Big SQL with SSL, complete these steps:

1) Log in to the BigInsights console to identify the node (host name) that is running the Big SQL head node. Navigate 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**.
2) Log in to master management node as user 'biadmin'.
3) If you are using your own SSL certificate, skip this step. If you are using the self-signed certificate that comes with BigInsights, complete this step. Run the following command to export the certificate from BigInsights keystore in PKCS#12 format:

   ```
   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.
4) Open a secure shell environment (SSH) to the head node that you identified in step 1 as the Big SQL (DB2®) instance owner, the user name `bigsql`. Verify that current the directory is `/home/bigsql`.

a) If you are using your own SSL certificate, copy the certificate in PKCS#12 format to the `/tmp/db2sslcert` directory.

b) If you are using the self-signed certificate that you generated in step 3, copy the certificate generated on master management node to `/tmp/db2sslcert` directory.

5) Set up the environment to execute 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:/ho
me/bigsql/sqllib/lib32/gskit
```

6) Create the DB2 keystore:

a) Create a new 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.`

7) Import the SSL certificate from the /tmp/db2sslcert directory into the keystore created in step 6:

```
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 using this command:

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

It should list a certificate with label biginsights.

8) 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 on which SSL daemon should listen:

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

e) Specify DB2 communication protocol:

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

9) Refresh DB2 configuration by restarting DB2:

```
db2stop force; db2start
```

10) Verify that a DB2 daemon is listing on port 52000:

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

11) Verify Big SQL health by running the BigInsights health check command:

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

All inbound public communication is blocked, except on ports 22, 8443, 10000, 14443, 51000, and 7052. To communicate with SSL DB2 listener running on port 52000, enable communication on port 52000, public interface (eth1), of the head node (identified in step 1) by modifying the iptables:

```
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
```

SSL communication with Big SQL is now configured.

Ongoing user security (LDAP) tasks

Manually expire a user’s password

To expire a user’s password before the time period specified on the ‘DAYS_PASSWORD_VALID’ parameter, you can run the LDAP script with the ‘-w/-W’ option to provide the LDAP root bind password followed by the '-e' option and the user name whose password is to be expired.

To expire the password of a user 'testuser', you run the LDAP script as follows:

```
./ehaas_ldap.sh -w <rootBindPW> -e testuser -c /opt/ehaas/conf/ehaas_ldap.conf
```

Add lockout protection to a user account

If the parameter 'LOCKOUT_USER_ACCOUNTS' is set to 'true', new accounts will incorporate lockout by default. If you want to add lockout protection to a user account, you can do so with the LDAP script even after the user has been created. To do so, invoke the LDAP script with the '-w/-W' option to specify the root bind password and then the '-l' option with the name of the account to add lockout to.

For example, to add lockout to the 'testuser' account, you run the LDAP script as follows:

```
./ehaas_ldap.sh -w <rootBindPW> -l testuser -c /opt/ehaas/conf/ehaas_ldap.conf
```

Remove lockout protection from an account

Occasionally you might want to remove lockout protection from a user previously protected. To do so, invoke the LDAP script with the '-w/-W' option to specify the root bind password and then the '-n' option with the name of the account to remove lockout protection from.

Note that this operation will *permanently* remove lockout from protection for a user's account and is different than removing a user's current 'locked-out' status (as described in the next section). If you remove lockout protection from an account and want to add it back, you must follow the steps in the previous section, "Add lockout protection to a user account".

For example, to remove lockout protection from the 'testuser' account, you run the LDAP script as follows:

```
./ehaas_ldap.sh -w <rootBindPW> -n testuser -c /opt/ehaas/conf/ehaas_ldap.conf
```

Remove 'locked-out' status from user

If a user's account becomes locked out, but you feel confident that it was not due to a malicious attempt on the account, you can unlock the user's account and permit them to log in again immediately. To do so, run the LDAP script with the '-w/-W' option to specify the root bind password and then the '-f' option with the user name.

For example, to remove a user's (testuser) current 'locked-out' status and allow them to log in once more, run the LDAP script as follows:

./ehaas_ldap.sh -w <rootBindPW> -f testuser -c /opt/ehaas/conf/ehaas_ldap.conf

Delete a user

To remove a created user from your LDAP service, run the LDAP script with the ‘-w/-W’ option to specify the root bind password and then the ‘-d’ option with the user name to delete.

For example, to delete the user ‘testuser’, run the LDAP script as follows:

./ehaas_ldap.sh -w <rootBindPW> -d testuser -c /opt/ehaas/conf/ehaas_ldap.conf

Delete a group

To remove a group from your cluster, run the LDAP script with the ‘-w/-W’ option to specify the root bind password and then the ‘-D’ option with the name of the group you want to delete.

For example, to delete the group ‘testgroup’, invoke the LDAP script as so:

./ehaas_ldap.sh -w <rootBindPW> -D testgroup -c /opt/ehaas/conf/ehaas_ldap.conf

Appendix A: LDAP script quick reference guide

1) Add a POSIX user to the LDAP server:

```
ehaas_ldap.sh -w <ldapPassword> -u <username>:<UID>:<GID>:<(optional)password>
```

2) To add a POSIX group to LDAP server:

```
ehaas_ldap.sh -w <ldapPassword> -g <groupName>:<GID>
```

3) To add a POSIX user to pre-existing LDAP POSIX group:

```
ehaas_ldap.sh -w <ldapPassword> -a <Username>:<Groupname>
```

4) To change a users' password:

```
ehaas_ldap.sh -w <ldapPassword> -p <Username>
```

5) To change the LDAP root's pw:

```
ehaas_ldap.sh –r
```

6) To expire a user's password:

```
ehaas_ldap.sh -w <ldapPassword> -e <user>
```

7) To explicitly add lockout protection to a user's account:

```
ehaas_ldap.sh -w <ldapPassword> -l <user>
```

8) To explicitly remove lockout protection from a user's account:

```
ehaas_ldap.sh -w <ldapPassword> -n <user>
```

9) To remove lockout and allow a user to bind with LDAP again:

```
ehaas_ldap.sh -w <ldapPassword> -f <user>
```

10) To delete a user:

```
ehaas_ldap.sh -w <ldapPassword> -d <userName>
```

11) To delete a group:

```
ehaas_ldap.sh -w <ldapPassword> -D <groupName>
```

Tips:

- If you prefer a hidden input prompt for the LDAP root bind password, use '-W' instead of '-w'.
- ehaas_ldap.conf file is in a different directory than the ehaas_ldap.sh script. The '-c <path>' parameter tells where to find the .conf file when running the LDAP script.

Appendix B: Packages used in LDAP setup

+ openldap-clients-2.4.23-34.el6_5.1.x86_64
+ openldap-2.4.23-34.el6_5.1.x86_64
+ openldap-servers-2.4.23-34.el6_5.1.x86_64
+ nss-pam-ldapd-0.7.5-18.2.el6_4.x86_64
+ pam_ldap-185-11.el6.x86_64
+ authconfig-6.1.12-13.el6.x86_64
+ sshpass-1.05-1.el6.x86_64
+ OpenSSL 1.0.1e-fips
+ perl-TimeDate-1.16-11.1
+ xfsprogs-3.1.1-14
+ cluster-glue-1.0.5-6
+ cluster-glue-libs-1.0.5-6
+ resource-agents-3.9.5-3.1
+ heartbeat-3.0.4-2
+ ntp-4.2.6p5-1
+ oddjob - 0.30-5

Appendix C: Downloading your own self-signed certificate for SSL

BigInsights provides a self-signed certificate that is applied to the management nodes of your cluster. You can independently procure your own certificate. Follow these steps to apply it.

Chrome:

1. Make sure your certificate is available on your local system.

2. In Chrome, open the menu and click **Settings**.
3. At the bottom of the page, click **Show advanced settings**.
4. Under **HTTPS/SSL** click **Manage certificates**.
5. Click **Import** and follow the wizard steps to import the certificate.
6. Click **Close**.

Firefox:

1. Make sure your certificate is available on your local system.
2. Click **Tools > Options**.
3. Click **Advanced**.
4. Click **Certificates**.
5. Click **View Certificates**.
6. Click **Import** and browse to and select your certificate. Click **Open**.
7. Click **OK** twice.

Troubleshooting

Ask questions on the Hadoop Dev site.

LDAP script logs are by default located at /tmp/bi_ldap/log. You can change this location by modifying the $LOG_FILE variable in the LDAP script.

If this does not resolve your issue, please email the author at mrkepple@us.ibm.com. More features and polish are planned to be added to this script and your comments and feature suggestions would be appreciated as well.