ATTUNITY Compose for Data Lakes

Attunity Compose Agent Installation Guide

Version 6.3

January 06, 2019

Contents

Introduction

When defining an Attunity Compose for Data Lakes for Spark project, the Attunity Compose for Data Lakes Agent must be installed on the remote Spark machine, which may either be ephemeral (i.e. part of an Amazon EMR, Microsoft Azure HDInsight, or Google Dataproc cluster) or non-ephemeral.

The installation procedure differs according to whether your Hadoop cluster is ephemeral or non-ephemeral.

Hadoop Cluster Type	Topic
Non-Ephemeral Cluster	See Installing Attunity Compose for Data Lakes Agent in a Non-Ephemeral Environment
Ephemeral Cluster	See one of the following topics: » Launching an Amazon EMR Cluster with Attunity Compose for Data Lakes Agent » Setting up a Microsoft Azure HDInsight Cluster with Attunity Compose Agent » Launching a Google Dataproc Cluster with Attunity Compose for Data Lakes Agent

Which Installation Package Do I Need?

The Compose Agent package you need to install depends on the Hadoop target platform.

The available platforms are as follows:

Platform	Required Package
Hortonworks	compose-agent-<version>-<build>.x86_64.rpm
Amazon EMR	compose-agent-<version>-<build>.x86_64.rpm
Cloudera	compose-agent-<version>-<build>.x86_64.rpm
Microsoft Azure HDInsight	compose-agent-<version>-<build>.amd64.deb
Google Cloud Storage (Dataproc)	compose-agent-<version>-<build>.amd64.deb

After installing the Attunity Compose for Data Lakes Agent, you need to provide the connection settings to the Spark machine or to your ephemeral cluster.

For more information on providing the connection settings, see Compose Agent Settings.

Installing Attunity Compose for Data Lakes Agent in a Non-Ephemeral Environment

This topic explains how to install the Attunity Compose for Data Lakes Agent in a non-ephemeral Hadoop cluster environment, which may exist either on-premises or in the cloud. The package you need to install depends on your environment. For more information, see Which Installation Package Do I Need?

> **Note** Requires Java runtime 1.8 and above.

Installing the RPM Package

To install the Attunity Compose for Data Lakes Agent:

Run the following command:

```
[user=usename] [group=groupname] [verbose=true] [debug=true]
password=password rpm -ivh AttunityComposeForDataLakes_Agent _<Version>_
Linux_X64.rpm
```

To upgrade the Attunity Compose for Data Lakes Agent:

Run the following command:

```
rpm -Uvh AttunityComposeForDataLakes_Agent _<Version>_Linux_X64.rpm
```

To uninstall the Attunity Compose for Data Lakes Agent:

Run the following command:

```
rpm -e compose-agent
```

Installing the Debian Package

To install the Attunity Compose for Data Lakes Agent:

Run the following command:

```
[user=usename] [group=groupname] [verbose=true] [debug=true]
password=password dpkg -i AttunityComposeForDataLakes_Agent _<Version>_
Linux_X64.deb
```

To upgrade the Attunity Compose for Data Lakes Agent:

```
dpkg -i AttunityComposeForDataLakes_Agent _<Version>_Linux_X64.deb
```

To uninstall the Attunity Compose for Data Lakes Agent:

Run the following command:

```
dpkg -r compose-agent
```

Optional Parameters

Parameter	Description
[user=*usename*]	Overrides the default user under which the Java service runs. The default user name is "Compose".
[group=*groupname*]	Overrides the default group under which the Java service runs. The default group name is "Compose". **Note** Only the root user and the specified user can run the service. Other users in the group cannot run the service.
[verbose=true]	Sets the logging mode to verbose.
[debug=true]	Sets the logging mode to debug.
--prefix=/installation_dir Not supported with Debian.	Prefixes the installation directory with the specified path. For example, if you specified: `--prefix=/mydir1/mydir2` The Compose Agent would be installed here: **/mydir1/mydir2/attunity/acompose**

Installing the Hortonworks JDBC Driver for Apache Hive

Perform the steps described in Installing the Hortonworks JDBC Driver for Apache Hive.

Configuration Options

» The **site_compose-agent_login.sh** file under the **bin** directory is a site specific process environment configuration file that you can modify as required. This may be useful, for example, if you want Compose for Data Lakes to run with a specific Java

version (for instance, when several Java versions are installed).

» **Spark home:** When using an on-premises Hadoop cluster, you need to specify the location of the Spark_Home variable (or $SPARK_HOME on Linux). This is not required when using Amazon EMR.

Changing the Compose Agent Password

If your cluster is active for an extended period, best practice is to periodically change the Compose Agent password.

To do this:

Run the following command from <INSTALL_DIR>\bin:

```
acjs.sh server setadminpassword new_password old_password
```

Example:

```
acjs.sh server setadminpassword 745hghTUYIIOJNOGO34 RE9R0EJVJFMA0GIW068
```

Launching an Amazon EMR Cluster with Attunity Compose for Data Lakes Agent

The procedure below explains how to launch an Amazon EMR cluster with Attunity Compose Agent.

1. Create an Amazon S3 bucket that your Amazon EMR cluster has read access to.
2. Edit the **compose-agent-<version>-<build>-emr-installer.sh** file and replace the default password (`emr`) with your own password. This is the password that you need to specify in the Compose Agent settings.
3. Upload the following files to this bucket:
   » compose-agent-<version>-<build>.x86_64.rpm (Provided by Attunity)
   » compose-agent-<version>-<build>-emr-installer.sh (Provided by Attunity)
   » HiveJDBC41.jar (Simba Hive JDBC driver)

   To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v1.0.42) from the Hortonworks website.
4. Launch your EMR cluster with the following minimum requirements:
   » **EMR version:**
     » emr-5.15.0
   » **The following services:**
     » Hadoop
     » Spark
     » Hive
     » Tez
5. Add a step of type "Custom JAR" to your EMR definition.
   a. In the **JAR location** field, specify the Amazon **script-runner.jar** for your region (located in s3://region.elasticmapreduce/libs/script-runner/**script-runner.jar**).

   For more information, see:
   https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop-script.html
   b. In the **Arguments** field:
     i. Pass the bucket location (i.e. including the file name) of the **compose-agent-<version>-<build>-emr-installer.sh** script as an argument to the script-runner JAR.
     ii. Pass the bucket folder in which the **compose-agent-<version>-<build>.x86_64.rpm** file resides as an argument to the **compose-agent-<version>-<build>-emr-installer.sh** script. If there are multiple **compose-agent-<version>-<build>.x86_64.rpm** files in the specified location, the script will always take the latest file.
     iii. Pass the bucket location (i.e. including the file name) of the

HiveJDBC41.jar file as an argument to the **compose-agent-<version>-<build>-emr-installer.sh** script.

Make sure to separate the arguments with a space.

Example:

s3://mybucket/compose compose-agent-<version>-<build>-emr-installer.sh s3://mybucket/compose s3://mybucket/compose/HiveJDBC41.jar

> **Note** When you terminate a cluster the associated HDFS storage will also be terminated. Therefore, if you want stored and provisioned data to be retained when a cluster is terminated, set the data store type (i.e storage) and the provisioning target in Compose for Data Lakes to Amazon S3.

Configuring Compose for Data Lakes to work with the Compose Agent on Amazon EMR

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

» Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

Example: 12.3.45.678 amazon.emr.cluster

With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**amazon.emr.cluster** in the above example) specified in the **Compose Agent Settings** window never needs to be changed.

» In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-emr-installer.sh** script as described above.

Setting up a Microsoft Azure HDInsight Cluster with Attunity Compose Agent

You can either launch a Microsoft Azure HDInsight cluster with the Compose for Data Lakes Agent or install the Compose for Data Lakes Agent on an active Microsoft Azure HDInsight cluster. This section explains how to do both as well as how to configure Attunity Compose for Data Lakes to work with the Compose for Data Lakes Agent on a Microsoft Azure HDInsight cluster.

Launching a Microsoft Azure HDInsight Cluster with Attunity Compose Agent

The procedure below explains how to launch a Microsoft Azure HDInsight cluster with Attunity Compose Agent.

1. Create an Microsoft Azure Blob Storage container to which your HDInsight cluster will have read access.
2. Edit the **compose-agent-<version>-<build>-hdi-installer.sh** file and replace the default password (`azure`) with your own password. This is the password that you need to specify in the Compose Agent settings.
3. Upload the following files to the container:
   » compose-agent-<version>-<build>.amd64.deb (provided by Attunity)
   » compose-agent-<version>-<build>-hdi-installer.sh (provided by Attunity)
   » HiveJDBC41.jar (Simba Hive JDBC driver)
      To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v1.0.42) from the Hortonworks website.
4. Launch your Microsoft Azure HDInsight cluster with the following minimum requirements:
   » **Cluster type:** Spark
   » **Version:** Spark 2.1.0 and above
5. In Step 5 of the cluster launch - **Advanced Settings** - click **Script Actions** and then click **Submit New**.
6. In the **Submit script action** window, enter the following values:
   a. **Script type:** - Custom
   b. **Name:** Any
   c. **Bash script UI:** Select the **compose-agent-<version>-<build>-hdi-installer.sh** script in the container you created earlier and copy its URL to this field.
   d. **Node type(s):** Select **Head**.

e. **Parameters:** Select the **HiveJDBC41.jar** and **compose-agent-<version>-<build>.amd64.deb** files in the container you created earlier and copy their URLs to this field.

f. Leave the **Persist this script action rerun when new nodes are added to the cluster** check box selected.

g. Click **Create**.

Installing Attunity Compose Agent on an Active Microsoft Azure HDInsight Cluster

1. Edit the **compose-agent-<version>-<build>-hdi-installer.sh** file and replace the default password (azure) with your own password. This is the password that you need to specify in the Compose for Data Lakes Agent settings in the Compose for Data Lakes console.

2. Copy the following files to your cluster head node:

    » compose-agent-<version>-<build>.amd64.deb (provided by Attunity)

    » compose-agent-<version>-<build>-hdi-installer.sh (provided by Attunity)

    » HiveJDBC41.jar (Simba Hive JDBC driver)

       To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v1.0.42) from the Hortonworks website.

3. Open a shell on the cluster head node and run the script using the root user (either through a root shell or using sudo):

```
./compose-agent-<version>-<build>-hdi-installer.sh compose-agent-
<version>-<build>.amd64.deb HiveJDBC41.jar local
```

Configuring Compose for Data Lakes to work with the Compose Agent on a Microsoft Azure HDInsight Cluster

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

» Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

Example: 12.3.45.678 microsoft.hdinsight.cluster

With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**microsoft.hdinsight.cluster** in

the above example) specified in the **Compose Agent Settings** window never needs to be changed.

» In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-hdi-installer.sh** script as described above.

Launching a Google Dataproc Cluster with Attunity Compose for Data Lakes Agent

The procedure below explains how to launch a Google Dataproc Cluster with Attunity Compose Agent.

1. Create a Google Cloud Storage bucket that your Google Dataproc cluster has read access to.

2. Edit the **compose-agent-<version>-<build>-dataproc-installer.sh** file and replace the default password (`google`) with your own password. This is the password that you need to specify in the Compose Agent settings.

3. Upload the following files to the bucket you created earlier:
   » compose-agent-<version>-<build>.amd64.deb (Provided by Attunity)
   » compose-agent-<version>-<build>-dataproc-installer.sh (Provided by Attunity)
   » HiveJDBC41.jar (Simba Hive JDBC driver)

   To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v1.0.42) from the Hortonworks website.

4. From the **Navigation menu** in the Google Cloud Platform console, select **Compute Engine** > **Metadata**.

5. In the **Metadata** window:
   a. Add the following metadata items:
      » att-cmps-package-folder-url
      » att-cmps-hive-jdbc-jar-url
   b. Click **Save**.

6. Return to the **Navigation menu** and select **Dataproc** > **Clusters**.

7. Configure your cluster settings as desired and then configure the following settings which are required for Compose Agent:
   a. Expand the **Advanced options** and click the **Add initialization action** button.
   a. In the **bucket/folder/file** field, browse to the **compose-agent-<version>-<build>-dataproc-installer.sh** file in the bucket you created earlier.

8. Create your Google Dataproc cluster.

Configuring Compose for Data Lakes to work with the Compose Agent on a Google Dataproc Cluster

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

» Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

Example: 12.3.45.678 google.dataproc.cluster

With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**google.dataproc.cluster** in the above example) specified in the **Compose Agent Settings** window never needs to be changed.

» In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-dataproc-installer.sh** script as described above.

Verifying that Compose Agent is Correctly Installed

There are several ways of verifying that Compose Agent is installed correctly.

These are as follows:

» Check the color of the connectivity icon in the upper right corner of the Compose for Data Lakes Console:

  » Green indicates that a successful connection to Compose Agent has been established.

  » Red indicates that there was a problem connecting to Compose Agent.

» Select **Compose Agent Settings** from the **Management** menu in the main window and click **Test Connection**.

» Check for [Error] messages in the **compose_agent.log** which can be accessed as described in the *Attunity Compose for Data Lakes Installation and User Guide*.

Installing the Hortonworks JDBC Driver for Apache Hive

1. Download the Hortonworks JDBC Driver for Apache Hive (v1.0.42) from the Hortonworks website.

2. Extract the **HiveJDBC41.jar** file from the **Simba_HiveJDBC41_1.0.42.1054.zip** file.

3. Copy the **HiveJDBC41.jar** file to the following location(s), depending on where the Compose for Data Lakes Agent is installed:

   » If the Compose for Data Lakes Agent is installed locally (i.e. as part of the Compose for Data Lakes installation), copy the HiveJDBC41.jar file to the following location on the Windows Compose for Data Lakes machine:

   <Compose_Installation_Dir>\java\jdbc

   » If the Compose for Data Lakes Agent is installed remotely (required for Apache Spark projects), copy the HiveJDBC41.jar file to the following location on the Linux Compose for Data Lakes Agent machine:

   <Compose_Installation_Dir>/jdbc

4. If the Compose for Data Lakes Agent is installed locally (i.e. on Windows), restart the Attunity Compose for Data Lakes service.

5. If the Compose for Data Lakes Agent is installed on Linux, restart the Compose for Data Lakes Agent Server by running the following command:

   ```
   ./compose-agent.sh restart
   ```

Compose Agent Settings

> **Note** These settings are relevant for Apache Spark projects only.

In an Apache Spark project, first install the Attunity Compose for Data Lakes Agent as described in this guide and then configure the Compose Agent connection settings in the Compose for Data Lakes Console as described below.

To specify the connection settings:

1. From the **Management** menu in the projects view, select **Compose Agent Settings**.
   The **Compose Agent Settings** window opens.
2. Select **Remote Server** and provide the required connection details. Note that the password is the password you provided when you installed the Compose Agent.
3. Optionally (but recommended), click **Test Connection** to verify the settings.
4. Click **OK** to save your settings.