Configuring a MapR Cluster to Access an HDFS Cluster

As of the 3.1 release of the MapR distribution for Hadoop, a MapR cluster can access an external HDFS cluster with the `hdfs://` or `webhdfs://` protocols.

> Use the `hdfs://` protocol for CDH3 clusters.

Prerequisites

- The MapR node accessing the HDFS cluster must have the `mapr-core` or `mapr-client` package installed.
- The HDFS cluster is installed and configured according to the vendor's specifications.
- To use the `hdfs://` protocol, edit the `fs.hdfs.impl` property in the `$HADOOP_HOME/conf/core-site.xml` file to include the value `org.apache.hadoop.hdfs.DistributedFileSystem`.

The following cases provide configuration instructions for various combinations of secure and non-secure MapR and HDFS clusters:

Case 1: Non-Secure MapR and HDFS Clusters

If the MapR and HDFS cluster are both non-secure, verify that the `fs.hdfs.impl` property in the `$HADOOP_HOME/conf/core-site.xml` file has the following value:

```
org.apache.hadoop.hdfs.DistributedFileSystem
```

No additional configuration is required.

Case 2: Secure MapR and HDFS Clusters

If the MapR and HDFS clusters are both secure, complete the following steps:

1. Verify that the `hadoop.rpc.protection` configuration parameter is set to the same value on the MapR and HDFS clusters. This parameter takes one of three values: `authentication`, `integrity`, or `privacy`. The default value for a MapR cluster with security features enabled is `privacy`. Configure this value in the `$HADOOP_HOME/conf/core-site.xml`. After changing this value, you must restart Hadoop services on the node where you changed the configuration.
2. On each node in your MapR cluster that runs the JobTracker or JobClient, create the file `$HADOOP_HOME/conf/hdfs-site.xml` with the following properties:

```
<property>
 <name>hadoop.security.authentication</name>
 <value>kerberos</value>
</property>
<property>
 <name>hadoop.security.authorization</name>
 <value>true</value>
</property>
<property>
 <name>dfs.namenode.kerberos.principal</name>
 <!-- Name Node principal in HDFS cluster -->
 <value>hdfs/_HOST@<KERBEROS_REALM></value>
</property>
<property>
 <name>dfs.web.authentication.kerberos.principal</name>
 <value>HTTP/_HOST@<KERBEROS_REALM></value>
</property>
```

3. For each JobTracker node in the MapR cluster:
   a. Create the `mapr/<FQDN_OF_JOBTRACKER>@<KERBEROS_REALM>` Kerberos principal in the same KDC as the one used by HDFS cluster. Note that the username for the principal *must* be `mapr` .
   b. Generate a keytab file for this principal, such as `mapr.keytab,` copy the keytab to the `/opt/mapr/conf/` directory on the Job Tracker node, and set the keytab file's ownership to `mapr:mapr` and the permission mode bits to 600.
   c. Add the following configuration parameters to the `$HADOOP_HOME/conf/mapred-site.xml` file:

```
<property>
 <!-- Job Tracker principal in MapR cluster -->
    <name>mapreduce.jobtracker.kerberos.principal</name>
    <value>mapr/_HOST@<KERBEROS_REALM></value>
</property>
<property>
  <name>mapreduce.jobtracker.keytab.file</name>
  <value>/opt/mapr/hadoop/hadoop-0.20.2/conf/mapr.keytab</value>
</property>
```

   d. Modify the `HADOOP_JOBTRACKER_OPTS` variable in `/opt/mapr/conf/env.sh` by replacing `${MAPR_LOGIN_OPTS}` with `${HYBRID_LOGIN_OPTS}`.
   e. Restart the JobTracker and TaskTracker services.
4. Launch any commands that need to access the HDFS cluster with the `“HADOOP_OPTS=-Dhadoop.login=hybrid”` Java property.

Case 3: Secure MapR Cluster and Non-Secure HDFS Cluster

If a secure MapR cluster is accessing an non-secure HDFS cluster with the `hdfs://` protocol, modify the `/opt/mapr/conf/env.sh file` on each node to replace `${MAPR_LOGIN_OPTS}` with `“-Dhadoop.login=maprsasl_permissive”` in the `HADOOP_JOBTRACKER_OPTS` and `HADOOP_TASKTRACKER_OPTS` variables.

Launch any commands that need to access the HDFS cluster with the `“HADOOP_OPTS=-Dhadoop.login=hybrid”` Java property.

Case 4: Secure MapR Cluster with Secure and Non-Secure HDFS Cluster

If the MapR cluster is secure and the HDFS cluster is both secure and non-secure, complete the steps described in Step 2, above.

If a secure MapR cluster is accessing an non-secure HDFS cluster with the `hdfs://` protocol, modify the `/opt/mapr/conf/env.sh file` on each node to replace `${MAPR_LOGIN_OPTS}` with `“-Dhadoop.login=maprsasl_permissive”` in the `HADOOP_JOBTRACKER_OPTS` and `HADOOP_TASKTRACKER_OPTS` variables.

Launch any commands that need to access the HDFS cluster with the `“HADOOP_OPTS=-Dhadoop.login=hybrid”` Java property.

Case 5: Non-Secure MapR Cluster with Secure HDFS Cluster

This case is not supported. Non-secure MapR clusters cannot connect to secure HDFS clusters.

Verifying access to HDFS cluster

Use the following commands to verify access to the remote HDFS cluster from the MapR cluster. If the remote HDFS cluster is secure, use `kinit` to obtain the necessary credentials beforehand.

CDH3 Only

```
hadoop fs -ls hdfs://<namenode_host:port>/
```

Other HDFS Versions

```
hadoop fs -ls webhdfs://<namenode_host_running_webhdfs_service>/
```