Hive Password Encryption

MEP 4.0 introduces default configuration for Hive Metastore password encryption using the MapR Installer. The password is stored in the `hive-site.xml` file.

MEP 4.0 introduces default configuration for Hive Metastore password encryption using the MapR Installer. The password is stored in the `hive-site.xml` file.

Note: For Hive-2.1 (MEP-5.0.0 and later) and Hive-2.3 (MEP-6.0.0 and later) installed using the MapR Installer, `javax.jdo.option.ConnectionPassword` is automatically encrypted.

```
<property>
  <name>javax.jdo.option.ConnectionPassword<name>
  <value>{password}<value>
<property>
```

The `hadoop.security.credential.provider.path` configuration property replaces the `javax.jdo.option.ConnectionPassword` property in the `hive-site.xml` file that contains the path to the keystore file created by the Hadoop Crediential Provider. Credential providers store and protect passwords out of clear text for the underlying database. By default, the MapR Installer creates the keystore file in MapR file system. `/user/${MAPR_USER}/hivemetastore.jceks` .

Note: Starting from Hive-2.3 MEP 6.0.0, SSL keystore passwords, `hive.server2.webui.keystore.password` , `hive.server2.keystore.password` , and `templeton.keystore.password` , are automatically read from the `/opt/mapr/conf/ssl-client.xml` file without any additional steps from your side. But you can still encrypt them manually and store them in the `*jceks` files.

Reset MapR Installer Default Configuration

To remove changes made by the MapR Installer and reset Hive to its default setting:

1. Open the `hive-site.xml` file.
2. Delete the `hadoop.security.credential.provider.path` property.
3. Add the `javax.jdo.option.ConnectionPassword` property.
4. Save and close the `hive-site.xml` file.

Manual Password Encryption

Note: For any user to use Hive, the keystore file requires read permission (644). To limit keystore file access to a smaller number of Hive users, modify permissions as necessary.

To encrypt a password manually:

1. Create the keystore file using the Hadoop Credential Provider as follows:

   ```
   hadoop credential create javax.jdo.option.ConnectionPassword -provider <path-to-keystore>
   ```

   Where `<path-to-keystore>` is `jceks://<file-system-name>/<path-to-keystore>` .

   For example, `jceks://maprfs/user/mapr/hivemetastore.jceks` .

2. Delete the `javax.jdo.option.ConnectionPassword` property in the `hive-site.xml` file:

   ```
   <property>
     <name>javax.jdo.option.ConnectionPassword</name>
     <value>{yourpassword}</value>
   </property>
   ```

3. Add the `hadoop.security.credential.provider.path` property to the `/opt/mapr/hive/hive-2.3/conf/hive-site.xml` file:

   ```
   <property>
     <name>hadoop.security.credential.provider.path</name>
     <value>jceks://maprfs/user/mapr/hivemetastore.jceks</value>
     <description>password to use against metastore database</description>
   </property>
   ```

4. Restart the Hive services to update the configuration:

   ```
   maprcli node services -name hivemeta -action restart -nodes `hostname -f`
   maprcli node services -name hs2 -action restart -nodes `hostname -f`
   maprcli node services -name hcat -action restart -nodes `hostname -f`
   ```