

Configuring Oozie on a Secure Cluster

The default configuration for Oozie on a secure MapR cluster uses [MapR tickets](#) (../SecurityGuide/Tickets.html) to authenticate between the Oozie client and server. The Oozie server uses MapR tickets to authenticate the connection between the YARN components and the JobClient embedded in the Oozie server. Starting in the MEP 4.0 release, for secure clusters, Oozie is also configured to use SSL encryption. This default configuration is in place once Oozie is installed and the security features for your cluster are enabled. See [Enabling and Disabling Security](#) (../SecurityGuide/Enable-wire-level-security.html). No further configuration is required. See [User Impersonation for Oozie](#) (UserImpersonationforOozie.html) to enable user impersonation for Oozie.

The sections below provide instructions to manually configure Oozie features on a secure MapR cluster:

▼ Configuring the Oozie Server to use SSL

Oozie traffic that uses HTTP is not encrypted by default for non-secure clusters and clusters earlier than MEP 4.0. To enable Secure Sockets Layer (SSL) encryption for Oozie, follow these steps:

1. Shut down Oozie.
2. For Oozie 4.3, specify the path to the keystore and password at `/opt/mapr/oozie/oozie-<version>/conf/oozie-env.sh` :


```
export OOZIE_HTTPS_KEYSTORE_PASS=password
export OOZIE_HTTPS_KEYSTORE_FILE=${HOME}/.keystore
```

3. As the `cluster admin` , run the following command:

```
# /opt/mapr/oozie/oozie-<version>/bin/oozie-setup.sh -hadoop <version> /opt/mapr/hadoop/hadoop-<version> -secure
```

4. Restart Oozie. After the restart, Oozie listens on port 11443 instead of 11000. Change the value of the `OOZIE_URL` environment variable by running the following command:

```
$ export OOZIE_URL="https://<fqdn>:11443/oozie"
```

 **Warning:** Using `localhost` instead of the Oozie server's fully qualified domain name may generate SSL handshake exceptions.

▼ Configuring Oozie Clients to Use SSL (Oozie 4.2.0)

To configure the Oozie clients, follow these steps:

1. Export the certificate from the Oozie server node:

```
sudo -u mapr keytool -exportcert -alias <alias> -file /opt/mapr/conf/certificate.cert -keystore /opt/mapr/conf/ssl_keystore
```

2. Copy `/opt/mapr/conf/certificate.cert` to all Oozie client nodes.
3. On the Oozie client nodes, import the certificate:

```
sudo keytool -importcert -alias my.cluster.com -file /opt/mapr/conf/certificate.cert -keystore
${JRE_cacerts}
```

▼ Configuring Oozie Clients to Use SSL (Oozie 4.3.0)

To configure the Oozie clients, follow this step:

1. Specify the path to the keystore and password at `/opt/mapr/oozie/oozie-<version>/conf/oozie-client-env.sh` :

```
export OOZIE_CLIENT_OPTS='-
Djavax.net.ssl.trustStore=/path/to/oozie.truststore'
```

▼ Using Kerberos to Securely Authenticate Between the Oozie Client and Server

Oozie can use Kerberos to secure authentication between the Oozie client and server. The Oozie server uses the Kerberos principal and keytab information specified in the Java Authentication and Authorization (JAAS) configuration file at `/opt/mapr/conf/mapr.login.conf` . Generate a Kerberos principal of the form `http/<fqdn>@<realm>` and store the keytab in the cluster's keytab file. The default keytab file location is `/opt/mapr/conf/mapr.keytab` .

To use Kerberos authentication on a specific invocation of Oozie without modifying your client, use the `-auth KERBEROS` option when you start Oozie, as in the following example:

```
$ bin/oozie admin -status -auth KERBEROS
```

▼ Defining a Custom Principal and Keytab File

You can use custom Kerberos principals and keytab files if you wish. To specify the locations of these custom Kerberos principals and keytab files, make the following modifications to the `oozie-site.xml` file:

- Explicitly change the authentication type to Kerberos.

```
<property>
  <name>oozie.authentication.type</name>
  <value>kerberos</value>
  <description>
    Defines authentication used for Oozie HTTP endpoint.
    Supported values are: simple | kerberos | #AUTHENTICATION_HANDLER_CLASSNAME#
  </description>
</property>
```

- Modify the following entries to use your custom principals and keytab. The principal takes the form `HTTP/<hostname>`, where *hostname* is the URL used by the client to connect to the server.

```
<property>
  <name>oozie.service.HadoopAccessorService.keytab.file</name>
  <value>/opt/mapr/conf/mapr.keytab</value>
  <description>
    Location of the Oozie user keytab file.
  </description>
</property>

<property>
  <name>local.realm</name>
  <value>{local.realm}</value>
  <description>
    Kerberos Realm used by Oozie and Hadoop. Using 'local.realm' aligns with Hadoop configuration
  </description>
</property>

<property>
  <name>oozie.service.HadoopAccessorService.kerberos.principal</name>
  <value>mapr/<hostname>@${local.realm}</value>
  <description>
    Kerberos principal for Oozie service.
  </description>
</property>

<property>
  <name>oozie.authentication.kerberos.principal</name>
  <value>HTTP/<hostname>@${local.realm}</value>
  <description>
    Indicates the Kerberos principal to be used for the HTTP endpoint. The principal MUST start with 'HTTP/' per the Kerberos HTTP SPNEGO specification.
  </description>
</property>
```

- Optional: If you plan to run Oozie actions that require talking to external services, add the `oozie.credentials.credentialclasses` to `oozie-site.xml`. For more details, see the Oozie [documentation](https://oozie.apache.org/docs/4.2.0/DG_ActionAuthentication.html) (https://oozie.apache.org/docs/4.2.0/DG_ActionAuthentication.html).

```
<property>
  <name>oozie.credentials.credentialclasses</name>
  <value>
    hcat=org.apache.oozie.action.hadoop.HCatCredentials,
    hbase=org.apache.oozie.action.hadoop.HbaseCredentials,
    hive2=org.apache.oozie.action.hadoop.Hive2Credentials
  </value>
</property>
```

 **Note:** No specific configuration is required for configuring Oozie to use MapR-SASL.

▼ Disabling Cached Tokens

After a client authenticates to Oozie, the authentication token received by the client is cached in the user's home directory in the `.oozie-auth-token` file. As long as the cached token remains valid, future authentication requests from the same client use that token and succeed, even if the client's Kerberos or MapR credentials have expired or have been revoked. You can disable use of the cache file by using the `oozie` command-line interface with the `-Doozie.auth.token.cache false` option.