WebHCat InstallWebHCat

WebHCat Installation

WebHCat Installed with Hive

> **Version**
>
> WebHCat and HCatalog are installed with Hive, starting with Hive release 0.11.0.

If you install Hive from the binary tarball, the WebHCat server command `webhcat_server.sh` is in the `hcatalog/sbin` directory.

Hive installation is documented here.

WebHCat Installation Procedure

Note: WebHCat was originally called Templeton. For backward compatibility the name still appears in URLs, log file names, variable names, etc.

1. Ensure that the required related installations are in place, and place required files into the Hadoop distributed cache.
2. Download and unpack the HCatalog distribution.
3. Set the `TEMPLETON_HOME` environment variable to the base of the HCatalog REST server installation. This will usually be same as `HCATALOG_HOME`. This is used to find the WebHCat (Templeton) configuration.
4. Set `JAVA_HOME`, `HADOOP_PREFIX`, and `HIVE_HOME` environment variables.
5. Review the configuration and update or create `webhcat-site.xml` as required. Ensure that site-specific component installation locations are accurate, especially the Hadoop configuration path. Configuration variables that use a filesystem path try to have reasonable defaults, but it's always safe to specify a full and complete path.
6. Verify that HCatalog is installed and that the `hcat` executable is in the `PATH`.
7. Build HCatalog using the command `ant jar` from the top level HCatalog directory.
8. Start the REST server with the command "`hcatalog/sbin/webhcat_server.sh start`" for Hive 0.11.0 releases and later, or "`sbin/webhcat_server.sh start`" for installations prior to HCatalog merging with Hive.
9. Check that your local install works. Assuming that the server is running on port 50111, the following command would give output similar to that shown.

```
% curl -i http://localhost:50111/templeton/v1/status
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.6.0.v20120127)

{"status":"ok","version":"v1"}
%
```

Server Commands

- **Start the server:** `sbin/webhcat_server.sh start` (HCatalog 0.5.0 and earlier – prior to Hive release 0.11.0)
  - `hcatalog/sbin/webhcat_server.sh start` (Hive release 0.11.0 and later)

- **Stop the server:** `sbin/webhcat_server.sh stop` (HCatalog 0.5.0 and earlier – prior to Hive release 0.11.0)
    - `hcatalog/sbin/webhcat_server.sh stop` (Hive release 0.11.0 and later)
- **End-to-end build, run, test:** `ant e2e`

Requirements

- Ant, version 1.8 or higher
- Hadoop, version 1.0.3 or higher
- ZooKeeper is required if you are using the ZooKeeper storage class. (Be sure to review and update the ZooKeeper-related WebHCat configuration.)
- HCatalog, version 0.5.0 or higher. The `hcat` executable must be both in the `PATH` and properly configured in the WebHCat configuration.
- Permissions must be given to the user running the server. (See below.)
- If running a secure cluster, Kerberos keys and principals must be created. (See below.)
- Hadoop Distributed Cache. To use Hive, Pig, or Hadoop Streaming resources, see instructions below for placing the required files in the Hadoop Distributed Cache.

Hadoop Distributed Cache

The server requires some files be accessible on the Hadoop distributed cache. For example, to avoid the installation of Pig and Hive everywhere on the cluster, the server gathers a version of Pig or Hive from the Hadoop distributed cache whenever those resources are invoked. After placing the following components into HDFS please update the site configuration as required for each.

- **Hive:** Download the Hive tar.gz file and place it in HDFS. For example, for Hive version 0.11.0:

```
hadoop fs -put /tmp/hive-0.11.0.tar.gz /apps/templeton/hive-0.11.0.
tar.gz
```

- **Pig:** Download the Pig tar.gz file and place it into HDFS. For example, for Pig version 0.11.1:

```
hadoop fs -put /tmp/pig-0.11.1.tar.gz /apps/templeton/pig-0.11.1.tar.
gz
```

- **Hadoop Streaming:** Place `hadoop-streaming-*.jar` into HDFS. Use the following command:

```
hadoop fs -put <hadoop streaming jar> \
  <templeton.streaming.jar>/hadoop-streaming-*.jar
```

  where *<templeton.streaming.jar>* is a property value defined in `webhcat-default.xml` which can be overridden in the `webhcat-site.xml` file, and *<hadoop streaming jar>* is the Hadoop streaming jar in your Hadoop version:

    - `hadoop-1.*/contrib/streaming/hadoop-streaming-*.jar` in the Hadoop 1.x tar
    - `hadoop-2.*/share/hadoop/tools/lib/hadoop-streaming-*.jar` in the Hadoop 2.x tar
      For example,

```
hadoop fs -put hadoop-2.1.0/share/hadoop/tools/lib/hadoop-
streaming-2.1.0.jar \
  /apps/templeton/hadoop-streaming.jar
```

- **Override Jars:** Place override jars required (if any) into HDFS. *Note:* Hadoop versions prior to 1.0.3 required a patch (HADOOP-7987) to properly run WebHCat. This patch is distributed with WebHCat (located at `templeton/src/hadoop_temp_fix/ugi.jar`) and should be placed into HDFS, as reflected in the current default configuration.

```
hadoop fs -put ugi.jar /apps/templeton/ugi.jar
```

The location of these files in the cache, and the location of the installations inside the archives, can be specified using the following WebHCat configuration variables. (See the Configuration documentation for more information on changing WebHCat configuration parameters.) Some default values vary depending on release number; defaults shown below are for the version of WebHCat that is included in Hive release 0.11.0. Defaults for the previous release are shown in the HCatalog 0.5.0 documentation.

Name	Default (Hive 0.11.0)	Description
templeton.pig.archive	`hdfs:///apps/templeton/pig-0.11.1.tar.gz`	The path to the Pig archive.
templeton.pig.path	`pig-0.11.1.tar.gz/pig-0.11.1/bin/pig`	The path to the Pig executable.
templeton.hive.archive	`hdfs:///apps/templeton/hive-0.11.0.tar.gz`	The path to the Hive archive.
templeton.hive.path	`hive-0.11.0.tar.gz/hive-0.11.0/bin/hive`	The path to the Hive executable.
templeton.streaming.jar	`hdfs:///apps/templeton/hadoop-streaming.jar`	The path to the Hadoop streaming jar file.
templeton.override.jars	`hdfs:///apps/templeton/ugi.jar`	Jars to add to the `HADOOP_CLASSPATH` for all Map Reduce jobs. These jars must exist on HDFS. This is not needed for Hadoop versions 1.0.1 and newer.

Permissions

Permission must be given for the user running the WebHCat executable to run jobs for other users. That is, the WebHCat server will impersonate users on the Hadoop cluster.

Create (or assign) a Unix user who will run the WebHCat server. Call this USER. See the Secure Cluster section below for choosing a user on a Kerberos cluster.

Modify the Hadoop core-site.xml file and set these properties:

Variable	Value
hadoop.proxyuser.USER.groups	A comma-separated list of the Unix groups whose users will be impersonated.
hadoop.proxyuser.USER.hosts	A comma-separated list of the hosts that will run the HCatalog and JobTracker servers.

Secure Cluster

To run WebHCat on a secure cluster follow the Permissions instructions above but create a Kerberos principal for the WebHCat server with the name `USER/host@realm`.

Also, set the WebHCat configuration variables `templeton.kerberos.principal` and `templeton.kerberos.keytab`.

Proxy User Support

Proxy User Support in WebHCat allows the caller of WebHCat to instruct WebHCat to run commands on the Hadoop cluster as a particular user.

The canonical example is Joe using Hue to submit a MapReduce job through WebHCat. For the following description, assume Joe has the Unix name 'joe', Hue is 'hue' and WebHCat is 'hcat'. If Hue specifies 'doAs=joe' when calling WebHCat, WebHCat submits the MR job as 'joe' so that the Hadoop cluster can perform securitiy checks with respect to 'joe'. If the doAs value is not specified, the MR job will be submitted as user 'hue'.

To set up Proxy User Support, make the following edits in configuration files.

In hive-site.xml, set:

Variable	Value
hive.security.metastore.authorization.manager	org.apache.hadoop.hive.ql.security.authorization. StorageBasedAuthorizationProvider
hive.security.metastore.authenticator.manager	org.apache.hadoop.hive.ql.security. HadoopDefaultMetastoreAuthenticator
hive.metastore.pre.event.listeners	org.apache.hadoop.hive.ql.security.authorization. AuthorizationPreEventListener
hive.metastore.execute.setugi	true

In webhcat-site.xml, set:

Variable	Value
webhcat.proxyuser.hue.groups	A comma-separated list of the Unix groups whose users may be impersonated by 'hue'.
webhcat.proxyuser.hue.hosts	A comma-separated list of the hosts which are allowed to submit requests by 'hue'. In the canonical example, this would be the servers running Hue.

In core-site.xml, make sure the following are also set:

Variable	Value
hadoop.proxyuser.hcat.group	A comma-separated list of the Unix groups whose users may be impersonated by 'hcat'.
hadoop.proxyuser.hcat.hosts	A comma-separated list of the hosts which are allowed to submit requests by 'hcat'.

Navigation Links

Previous: Using WebHCat
Next: Configuration

Hive installation: Installing Hive
HCatalog installation: Installation from Tarball

General: WebHCat Manual – HCatalog Manual – Hive Wiki Home – Hive Project Site
Old version of this document (HCatalog 0.5.0): WebHCat Installation