Enable Integration with Cluster High Availability

Contents:

In a Hadoop cluster, **high availability** provides failover support for one or more configured nodes. This section describes how to enable the Trifacta® platform to utilize the highly available set of nodes within the Hadoop cluster. High availability enables access to each node of the cluster configured for it, in the event of machine crash or software installation or upgrade.

If high availability is enabled on the Hadoop cluster, you must enable it on the Trifacta platform, which prevents integration conflicts between Hadoop-specific components in the platform and their cluster equivalents.

Enable HttpFS

The WebHDFS service does not directly support high availability. You must enable the related HttpFS service and specify a WebHDFS namenode to point to the server hosting the HttpFS service.

NOTE: Avoid enabling the HttpFS service on the primary namenode of the cluster. In the event that the node hosting the namenode fails over, the HttpFS service is no longer available. You may be required to manually set the active namenode and restart the Trifacta platform.

For more information, see *Enable HttpFS*.

Enable HA Service

To begin, you must enable the High Availability service in the Trifacta platform for the supported components. In platform configuration, each component has its own feature flag under `feature.highAvailability`.

You can apply this change through the *Admin Settings Page* (recommended) or

`trifacta-conf.json`

. For more information, see *Platform Configuration Methods*.

In the following example configuration, high availability has been disabled for resourcemanagers and enabled for namenodes:

NOTE: In almost all cases, `feature.highAvailability.resourceManager` should be set to `false` . For more information, see *Example - Configure resource manager* below.

```
"feature.highAvailability.namenode": true,
"feature.highAvailability.resourceManager": false,
```

Configure HA for Individual Components

High availability in Hadoop works by specifying a **nameservice** for a highly available component and then enumerating the hosts and ports as **children** of that nameservice node. These values must be explicitly specified in the platform configuration.

Service names and child names should be specified in the file as they appear in the cluster's configuration files.

Example - Configure namenode

In the following example, the nameservice `namenodeha` provides high availability through two namenodes: `nn1` a nd `nn2`. In a high availability environment, these hosts are used for submitting jobs and writing data to HDFS.

```
"hdfs": {
  ...
  "highAvailability": {
    "serviceName": "namenodeha",
    "namenodes": {
      "nn1": {
        "host": "nn1.hadoop.mycompany.org",
        "port": 8020
      },
      "nn2": {
        "host": "nn2.hadoop.mycompany.org",
        "port": 8020
      }
    }
  }
},
```

Example - Configure resource manager

NOTE: Set `feature.highAvailability.resourcemanager=true` only if the cluster file `yarn-site.xml` enables `yarn.resourcemanager.hostname.highlyavailableyarn`. This setting enables the cluster high availability for resourcemanager.

Otherwise, set `feature.highAvailability.resourcemanager=false` for all environments. For HA environments, the resourcemanager hosts specified in the configuration below set the HA servers that are used by the Trifacta platform.

The following example specifies two failover nodes for the resource manager: `rm1` and `rm2`.

```
"yarn": {
 "resourceManagers": {
   "rm1": {
     "host": "rm1.yarn.mycompany.org",
     "port": 8032,
     "schedulerPort": 8030,
     "adminPort": 8033,
     "webappPort": 8042
   },
  "rm2": {
     "host": "rm2.yarn.mycompany.org",
     "port": 8032,
     "schedulerPort": 8030,
     "adminPort": 8033,
     "webappPort": 8042
   }
  }
}
```

Update Active Namenode

The active namenode used by the service must be configured explicitly. This value must be updated whenever the active namenode changes. Otherwise, HDFS becomes unavailable.

NOTE: If the HttpFS service has been tied to the primary namenode of the cluster and that node fails, this setting must be manually configured to the new node and the platform must be restarted. Avoid tying HttpFS to the primary namenode.

In this example, the active namenode has been set to the `nn1` value in the previous configuration:

```
"webhdfs": {
 "proxy": { ... },
 "version": "/webhdfs/v1",
 "port": 14000,
 "httpfs": true
},
...
"namenode": {
 "host": "nn1.hadoop.mycompany.org",
 "port": 8020
 },
},
```

Configure HA in a Kerberized Environment

If you are enabling high availability in a Kerberized environment, additional configuration is required.

NOTE: WebHDFS does not support high availability/failover. You must enable HttpFS instead. For more information, see *Enable HttpFS*.

Steps:

1. If you have not done so already, acquire `httpfs-site.xml` from your Hadoop cluster.
2. Add the following settings to the file, replacing `[hadoop.user` (default=`trifacta)]` with the value appropriate for your environment:

```
<property>
 <name>httpfs.authentication.type</name>
 <value>org.apache.hadoop.security.token.delegation.web.KerberosDelegationTokenAuthenticationHandler<
/value>
</property>
<property>
 <name>httpfs.authentication.delegation-token.token-kind</name>
 <value>WEBHDFS delegation</value>
</property>
<property>
 <name>httpfs.proxyuser.[hadoop.user].hosts</name>
 <value>*</value>
</property>
<property>
 <name>httpfs.proxyuser.[hadoop.user].groups</name>
 <value>*</value>
</property>
```

3. The above change must also be applied to the `httpfs-site.xml` configuration file for the cluster.

Save your changes and restart the platform.

Platform Restart

When high availability has been enabled, you must restart the platform from the command line. For more information, see *Start and Stop the Platform*.