Cloudera Runtime 7.1.9

Configuring Apache Hadoop YARN High Availability

Date published: 2020-02-11
Date modified: 2023-08-29

CLOUDƎRA

Legal Notice

Contents

YARN ResourceManager high availability

The ResourceManager high availability (HA) feature adds redundancy in the form of an active-standby ResourceManager pair.

The YARN ResourceManager is responsible for tracking the resources in a cluster and scheduling applications (for example, MapReduce jobs). The ResourceManager high availability (HA) feature adds redundancy in the form of an active-standby ResourceManager pair to remove this single point of failure. Furthermore, upon failover from the active ResourceManager to the standby, the applications can resume from the last state saved to the state store; for example, map tasks in a MapReduce job are not run again if a failover to a new active ResourceManager occurs after the completion of the map phase. This allows events such the following to be handled without any significant performance effect on running applications:

- Unplanned events such as machine crashes
- Planned maintenance events such as software or hardware upgrades on the machine running the ResourceManager

ResourceManager HA requires ZooKeeper and HDFS services to be running.

YARN ResourceManager high availability architecture

Learn about the architecture of YARN ResourceManager High Availability to use it efficiently.

ResourceManager HA is implemented by means of an active-standby pair of ResourceManagers. On start-up, each ResourceManager is in the standby state; the process is started, but the state is not loaded. When one of the ResourceManagers is transitioning to the active state, the ResourceManager loads the internal state from the designated state store and starts all the internal services. The stimulus to transition to active comes from either the administrator (through the CLI) or through the integrated failover controller when automatic failover is enabled.

ResourceManager Restart

Restarting the ResourceManager allows for the recovery of in-flight applications if recovery is enabled. To achieve this, the ResourceManager stores its internal state, primarily application-related data and tokens, to the RMStateS tore; the cluster resources are re-constructed when the NodeManagers connect. The available alternatives for the state store are MemoryRMStateStore (a memory-based implementation) and ZKRMStateStore (ZooKeeper-based implementation). Note that MemoryRMStateStore will not work for HA.

Fencing

When running two ResourceManagers, a split-brain situation can arise where both ResourceManagers assume they are active. To avoid this, only a single ResourceManager should be able to perform active operations and the other ResourceManager should be "fenced". The ZooKeeper-based state store (ZKRMStateStore) allows only a single ResourceManager to make changes to the stored state, implicitly fencing the other ResourceManager. This is accomplished by the ResourceManager claiming exclusive create-delete permissions on the root znode. The ACLs on the root znode are automatically created based on the ACLs configured for the store; in case of secure clusters, Cloudera recommends that you set ACLs for the root host such that both ResourceManagers share read-write-admin access, but have exclusive create-delete access. The fencing is implicit and does not require explicit configuration (as fencing in HDFS does). You can plug in a custom "Fencer" if you choose to – for example, to use a different implementation of the state store.

Configuration and FailoverProxy

In an HA setting, you should configure two ResourceManagers to use different ports (for example, ports on different hosts). To facilitate this, YARN uses the notion of an ResourceManager Identifier (rm-id). Each ResourceManager has a unique rm-id, and all the RPC configurations (*<rpc-address>*; for example yarn.resourcemanager.add ress) for that ResourceManager can be configured via <rpc-address>.<rm-id>. Clients, ApplicationMasters, and NodeManagers use these RPC addresses to talk to the active ResourceManager automatically, even after a failover.

To achieve this, they cycle through the list of ResourceManagers in the configuration. This is done automatically and does not require any configuration (as it does in HDFS).

Automatic Failover

By default, ResourceManager HA uses ZKFC (ZooKeeper-based failover controller) for automatic failover in case the active ResourceManager is unreachable or goes down. Internally, the *StandbyElector* is used to elect the active ResourceManager. The failover controller runs as part of the ResourceManager.

You can plug in a custom failover controller if you prefer.

Manual Transitions and Failover

You can use the command-line tool yarn rmadmin to transition a particular ResourceManager to active or standby state, to fail over from one ResourceManager to the other, to get the HA state of an ResourceManager, and to monitor an ResourceManager's health.

Configuring YARN ResourceManager high availability

You can use Cloudera Manager to configure YARN ResourceManager high availability (HA).

Cloudera Manager supports automatic failover of the ResourceManager. It does not provide a mechanism to manually force a failover through the Cloudera Manager user interface.

⚠ **Important:** Enabling or disabling High Availability will cause the previous monitoring history to become unavailable.

Enable high availability

You can enable YARN ResourceManager High Availability using Cloudera Manager. When you enable ResourceManager HA in Cloudera Manager, work preserving recovery is also enabled for the ResourceManager by default.

1. In Cloudera Manager, select the YARN service.
2. Click Actions.
3. Select Enable High Availability.

   A screen showing the hosts that are eligible to run a standby ResourceManager displays. The host where the current ResourceManager is running is not available as a choice.
4. Select the host where you want the standby ResourceManager to be installed.
5. Click Continue.

   Cloudera Manager proceeds to run a set of commands that stop the YARN service, add a standby ResourceManager, initialize the ResourceManager high availability state in ZooKeeper, restart YARN, and redeploy the relevant client configurations.

   ResourceManager HA does not affect the JobHistory Server (JHS). JHS does not maintain any state, so if the host fails you can simply assign it to a new host. If you want to enable process auto-restart, do the following:
6. In Cloudera Manager, select the YARN service.
7. Click the Configuration tab.
8. Search for restart.
9. Find the Automatically Restart Process property.
10. Click Edit Individual Values.
11. Select the JobHistory Server Default Group option.
12. Click Save Changes.
13. Restart the JobHistory Server role.

Disable high availability

You can disable YARN ResourceManager High Availability using Cloudera Manager.

1. In Cloudera Manager, select the YARN service.
2. Click Actions.
3. Select Disable High Availability.

    A screen showing the hosts running the ResourceManagers displays.
4. Select the ResourceManager (host) you want to remain as the single ResourceManager.
5. Click Continue.

    Cloudera Manager runs a set of commands that stop the YARN service, remove the standby ResourceManager and the Failover Controller, restart the YARN service, and redeploy client configurations.

Using the yarn rmadmin tool to administer ResourceManager high availability

You can use yarn rmadmin in the command line to manage your ResourceManager HA deployment.

Note: Manually failing over the ResourceManager is not possible when high availability is enabled.

yarn rmadmin has the following options related to ResourceManager high availability:

```
[-getServiceState serviceId]
[-checkHealth <serviceId]
[-help <command>]
```

when *serviceID* is rm-id.

Note: Even though -help lists the -failover option, it is not supported by yarn rmadmin.

Migrating ResourceManager to another host

You can move the YARN ResourceManager role from one host to another using Cloudera Manager.

Before you begin

Cloudera recommends to enable RecourceManager High Availability.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Click Instance.
3. Find the ResourceManager role that you want to migrate to a new host and select it.
4. Select Actions for Selected Stop .
5. Select Actions for Selected Delete .
6. Click Add Role Instances.
7. Use the dropdown menu under ResourceManager to select a hosts for the new ResourceManager role.
8. Click Continue and then click Finish.
9. Select the newly created ResourceManager role.
10. Select Actions for Selected Start and click Start.

11. Restart the YARN service.

Results

The ResourceManager role now run on a new host.

Work preserving recovery for YARN components

With work preserving recovery enabled, if a ResourceManager or NodeManager restarts, no in-flight work is lost.

You can configure work preserving recovery separately for a ResourceManager or NodeManager. You can use work preserving recovery whether or not you use ResourceManager High Availability.

> **Note:** YARN does not support high availability for the JobHistory Server (JHS). If the JHS goes down, Cloudera Manager will restart it automatically.

> **Note:**
>
> After moving the JobHistory Server to a new host, the URLs listed for the JobHistory Server on the ResourceManager web UI still point to the old JobHistory Server. This affects existing jobs only. New jobs started after the move are not affected. For any existing jobs that have the incorrect JobHistory Server URL, there is no option other than to allow the jobs to roll off the history over time. For new jobs, make sure that all clients have the updated mapred-site.xml that references the correct JobHistory Server.

Configuring work preserving recovery on ResourceManager

Work preserving recovery is enabled by default for the ResourceManager. You can disable this feature using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Click the Configuration tab.
3. Search for Enable ResourceManager Recovery.
4. Clear the ResourceManager Default Group checkbox in the Enable ResourceManager Recovery field.
5. Click Save Changes.

Configuring work preserving recovery on NodeManager

Work preserving recovery is enabled by default for the NodeManager. You can disable this feature using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Click the Configuration tab.
3. Search for yarn-site.xml.
4. Find the NodeManager Advanced Configuration Snippet (Safety Valve) for yarn-site.xml.
5. Select the NodeManager and click on the plus icon.
6. Add the following:

   • Name: yarn.nodemanager.recovery.enabled
   • Value: false
7. Click Save Changes.

Example: Configuration for work preserving recovery

Use work preserving recovery properties to enable and configure this feature.

```
<property>
<name>yarn.resourcemanager.work-preserving-recovery.enabled</name>
<value>true</value>
<description>Whether to enable work preserving recovery for the Resource Ma
nager.</description>
</property>

<property>
<name>yarn.nodemanager.recovery.enabled</name>
<value>true</value>
<description>Whether to enable work preserving recovery for the Node Mana
ger.</description>
</property>

<property>
<name>yarn.nodemanager.recovery.dir</name>
<value>/home/cloudera/recovery</value>
<description>The location for stored state on the Node Manager, if work pres
erving recovery is enabled.</description>
</property>

<property>
<name>yarn.nodemanager.address</name>
<value>0.0.0.0:45454</value>
<description>The address of the container manager in the Node Manager.</desc
ription>
</property>

<property>
<name>yarn.resourcemanager.store.class</name>
<value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateSt
ore</value>
 <description>The class name of the state-store to be used for saving app
lication/attempt state and the credentials. </description>
</property>

<property>
<name>hadoop.zk.address</name>
<value>0.0.0.0:3001,0.0.0.0:3002,0.0.0.0:3003</value>
<description>Comma separated list of Host:Port pairs. Each corresponds to a
 ZooKeeper to be used by the Resource Manager for storing Resource Manager s
tate.</description>
</property>

<property>
<name>yarn.resourcemanager.zk-state-store.parent-path</name>
<value>/rmstore</value>
<description>The full path of the root znode where Resource Manager state
will be stored.</description>
</property>
<property>
<name>hadoop.zk.num-retries</name>
<value>500</value>
<description>Number of times Resource Manager tries to connect to ZooKeeper
 server if the connection is lost.</description>
</property>

<property>
```

```
<name>hadoop.zk.retry-interval-ms</name>
<value>120</value>
<description>The interval in milliseconds between retries when connecting to
 a ZooKeeper server.</description>
</property>

<property>
<name>hadoop.zk.timeout-ms</name>
<value>10</value>
<description> ZooKeeper session timeout in milliseconds. Session expiration
 is managed by the ZooKeeper cluster itself and not by the client. This val
ue is used by the cluster to determine when the client's session expires. Ex
pirations happens when the cluster does not hear from the client within the
specified session timeout period (that is, no heartbeat). </description>
</property>

<property>
<name>hadoop.zk.acl</name>
<value>world:anyone:rwcda</value>
<description>ACLs to be used for setting permissions on ZooKeeper znodes.</
description>
</property>
```