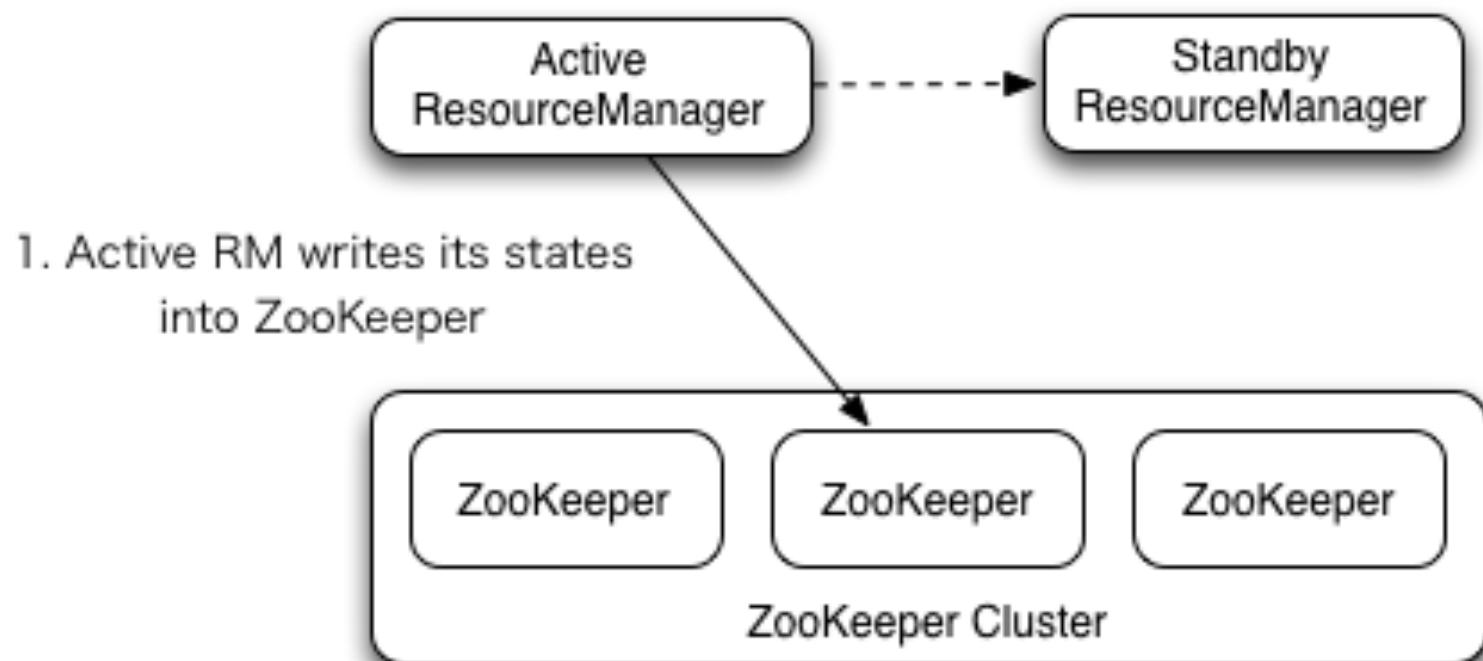


ResourceManager High Availability

Architecture

2. Fail-over if the Active RM fails
(fail-over can be done by auto/manual)



YARN (MRv2) ResourceManager High Availability

RM → tracking the resources in a cluster and scheduling applications .
Can be a single point of failure in a YARN cluster.

HA features:

- Adds redundancy in the form of an active-standby.
- 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

Uses:

- 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.

Architecture

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

Enable ResourceManager high availability

At least three:

- hosts in your cluster
- Apache ZooKeeper servers running

1. In **Ambari Web**, browse to **Services > YARN > Summary**.

2. Select **Service Actions** and choose **Enable ResourceManager HA**.

The **Enable ResourceManager HA** wizard launches, describing a set of automated and manual steps that you must take to set up ResourceManager high availability.

3. On **Get Started**, read the overview of enabling ResourceManager HA.

ENABLE
RESOURCEMANAGER HA
WIZARD

Get Started

Select Host

Review

Configure Components

Get Started

This wizard will walk you through enabling ResourceManager HA on your cluster.

Once enabled, you will be running a Standby ResourceManager in addition to your Active ResourceManager. This allows for an Active-Standby ResourceManager configuration that automatically performs failover.

You should plan a cluster maintenance window and prepare for cluster downtime when enabling ResourceManager HA.

Next →

Enable ResourceManager high availability

Select Host, accept the default selection or choose an available host. →

On **Review Selections**, expand YARN if necessary, to review all the configuration changes proposed for YARN. →

On **Configure Components**, click **Complete**.

Sample configurations

```
<property>  
  <name>yarn.resourcemanager.ha.enabled</name>  
  <value>true</value>  
</property>  
<property>  
  <name>yarn.resourcemanager.cluster-id</name>  
  <value>cluster1</value>  
</property>  
<property>  
  <name>yarn.resourcemanager.ha.rm-ids</name>  
  <value>rm1,rm2</value>  
</property>
```

Sample configurations..

```
<property>
  <name>yarn.resourcemanager.hostname.rm1</name>
  <value>master1</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname.rm2</name>
  <value>master2</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm1</name>
  <value>master1:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm2</name>
  <value>master2:8088</value>
</property>
```


Sample configurations..

```
<property>
```

```
  <name>yarn.resourcemanager.zk-address</name>
```

```
  <value>zk1:2181,zk2:2181,zk3:2181</value>
```

```
</property>
```

Admin commands

```
$ yarn rmadmin -getServiceState rm1  
active
```

```
$ yarn rmadmin -getServiceState rm2  
standby
```

```
$ yarn rmadmin -transitionToStandby rm1
```

Automatic failover is enabled for org.apache.hadoop.yarn.client.RMHAServiceTarget@1d8299fd

Refusing to manually manage HA state, since it may cause a split-brain scenario or other incorrect state.

If you are very sure you know what you are doing, please specify the forcemanual flag.

Lab : ResourceManager high availability