Lab: YARN Management

Managing & Using YARN

Lab: YARN Management

1. Table of Contents

2. Submitting Application And Viewing Application Status

The goal of this lab is to demonstrate the process of submitting job to YARN, and how to monitor status and locate job logs.

1. Login to cockpit and ssh to root@edge.cluster and switch user to admin

```
ssh root@edge.cluster
su - admin
```

2. Submit a sample Pi application

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
  -Dmapreduce.reduce.java.opts='-Xmx1024m' \
  16 1024
```

3. On another cockpit tab, ssh to root@edge.cluster and switch user to admin. While application is running, view status of submitted YARN application

```
yarn top
yarn application -list
yarn container -list $APP_ID
yarn queue -status root
```

4. While application is running, login to Ambari and navigate to YARN service. Click Quick Links > Resource Manager UI
5. Navigate and explore status and log of running application

3. Configuring YARN Capacity Scheduler

The goal of this lab is for student to be able to configure the capacity scheduler, and submit job to a particular Queue.

Lets create a queue hierarchy

1. Login to Ambari as user admin, and navigate to YARN Queue Manager view
2. Click the 'root' queue.
3. Click add queue and create queue 'demo'
4. Set minimum allocation to 75% and reduce llap queue allocation to 15% and default queue to 10%
5. Select 'demo' queue, click add queue, call it 'demo1'
6. Set minimum allocation to 50% and max allocation to 100% on 'demo1'
7. Add another queue called 'demo2', and set same allocation as like 'demo1' queue
    - Take note that you should end up with a tree of:

```
root
 `- demo
     |- demo1
     `- demo2
```

8. Click Actions > Save and Refresh Queues to apply changes
9. Lets submit the earlier example program, but on a queue demo1

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
  -Dmapreduce.reduce.java.opts='-Xmx1024m' \
  -Dmapreduce.job.queuename=demo1 \
  16 1024
```

10. Lets see yarn status, note on queue name and current capacity usage

```
yarn application -list
yarn queue -status demo1
yarn top
```

Lab: YARN Management

Lets allow user to scale capacity usage to 100% of 'default' queue.

1. Login to Ambari as user admin, and navigate to YARN Queue Manager view
2. Click 'demo1' queue
3. Increase user limit factor to 2
4. Save and refresh queue
5. Lets submit example program again.

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
  -Dmapreduce.reduce.java.opts='-Xmx1024m' \
  -Dmapreduce.job.queue.name=demo1 \
  16 1024
```

6. Lets see yarn status, note on queue name current capacity usage

```
yarn application -list
yarn queue -status demo1
yarn top
```

4. Configuring YARN Queue ACL

The goal of this lab is to demonstrate queue mapping, and configure queue access control on Ranger

Lets configure queue mapping to ensure user admin always get sent to demo1 queue

1. Login to Ambari as user admin, and navigate to YARN Queue Manager view
2. On the left bar, on Queue Mapping field, set

```
u:admin:demo1
```

3. Save and refresh queue
4. Run sample application without specifying queue name

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
  -Dmapreduce.reduce.java.opts='-Xmx1024m' \
  16 1024
```

5. Lets see yarn status, note on queue name current capacity usage

```
yarn application -list
yarn queue -status demo1
yarn top
```

Lab: YARN Management

Lets deny user admin from submitting to any other queue except ones it have been granted permission to

1. Login to Ambari as admin
2. Navigate to YARN service, ensure this option is added in Custom yarn-security:
   - ranger.add-yarn-authorization = false
3. Restart necessary services
4. Navigate to Ranger service, and through quick links, navigate to Ranger Admin UI
5. Under YARN, click on service SandboxCluster_yarn
6. Add a new policy
   - Name: allow demo1
   - Queue: root.demo.demo1
   - User: admin
   - Permission: all
   - Click Save
7. And another policy for the default queue
   - Name: allow default
   - Queue: root.default
   - Group: hadoop, users, public, hdfs, slider
   - User: hive, admin, jeff
   - Permission: submit-app
   - Click Save
8. Try to submit application to demo2 queue

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
  -Dmapreduce.reduce.java.opts='-Xmx1024m' \
  -Dmapreduce.job.queue.name=demo2 \
  16 1024
```

9. Try to submit application to demo1 queue

```
yarn jar \
  /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \
  pi -Dmapreduce.map.java.opts='-Xmx1024m' \
```

```
      -Dmapreduce.reduce.java.opts='-Xmx1024m' \
      -Dmapreduce.job.queue.name=demo1 \
      16 1024
```

5. Setting Up Resource Manager HA

The goal of this lab is to demonstrate the process of setting up ResourceManager HA

1. Login to Ambari as user admin and navigate to YARN service configuration
2. Click Service Actions > Enable ResourceManager HA
3. Assign host to run RM HA, review configuration, and proceed with installation