Lab: Running a YARN Application

About this Lab

Objective: To run a YARN application.

File locations: n/a

Successful outcome: You will have executed the DistributedShell YARN application.

Before you begin: Your HDP 2.3 cluster should be up and running within your VM.

Related lesson: ***Hadoop 2 and YARN***

Lab Steps

1) Run a `DistributedShell` Application

a. If not already done, open a Terminal in your VM and type "ssh sandbox".
b. In a terminal window change directories to the `/usr/hdp/current/hadoop-yarn-client` folder:

```
# cd /usr/hdp/current/hadoop-yarn-client/
```

c. Run the following command, which runs a sample `YARN` application that ships with HDP 2.x:

```
# yarn jar hadoop-yarn-applications-distributedshell.jar
org.apache.hadoop.yarn.applications.distributedshell.Client -jar
hadoop-yarn-applications-distributedshell.jar -shell_command cal
```

The `DistributedShell` command allows you to run a script or `shell` command on your cluster. The example above runs the Unix "`cal`" command, which displays a text calendar.

d. Wait for the YARN job to finish.

2) Verify the Result

a. Enter the following command (all on a single line):

```
# yarn application -list -appStates FINISHED | grep Dist
```

You should see the application ID of the `DistributedShell` command that you just ran:

```
application_1378331467073_0004 DistributedShell    YARN      yarn
default    FINISHED   SUCCEEDED        100%
```

b. Copy and paste the application `ID` of your DistributedShell command and check its status using the following command (but replacing the ID shown here with your actual application ID):

```
# yarn application -status application_1378331467073_0004
```

Notice that the details of the job are displayed. This was a simple application, so there is not a lot of information to analyze:

```
Application Report :
   Application-Id : application_1378331467073_0004
   Application-Name : DistributedShell
   Application-Type : YARN
   User : root
   Queue : default
   Start-Time : 1408060384688
   Finish-Time : 1408060391340
   Progress : 100%
   State : FINISHED
   Final-State : SUCCEEDED
   Tracking-URL : N/A
   RPC Port : -1
   AM Host : sandbox.hortonworks.com/172.16.173.149
   Diagnostics :
```

Note

The YARN application command also has a -kill option (followed by the application's ID) that kills a running YARN job. This is a great tool when you have submitted a job and then need to stop it before it runs to completion.

3) View the Log File

a Enter the following command to view the output for this YARN application that you just executed:

```
# yarn logs -applicationId application_1378331467073_0004
```

Important
The `-applicationId` must have the correct case for every letter, or else you will see an error message stating that it is missing. The only capital letter in the option is the I. The d, and all other letters, are lower case.
Somewhere in the log file, you should see a text calendar of the current month. For example:

```
LogType: stdout
LogLength: 150
Log Contents:
   August 2014
Su Mo Tu We Th Fr Sa
2
4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

4) Optional: Run the Job in Six Containers

a. The DistributedShell application allows you to specify how many containers the ApplicationMaster uses. Add the following arguments to the end of the `YARN` command from Step 1.2:

```
-num_containers 6 -container_memory 20
```

b. Now find the `applicationID` and view the aggregated log file:

```
# yarn application -list -appStates FINISHED | grep Dist

# yarn logs -applicationId <applicationId>
```

You should see six calendars this time, one from each container.

c. Notice that this also demonstrates how the log output from multiple containers is aggregated into a single, convenient log file.

Result

In this lab you ran a simple YARN application called the DistributedShell (that ships with HDP 2.x). You also saw how to view the output of the aggregated log file using the YARN logs command.