

Oneway synch using batch job, vm queues and mule runtime cluster

In this step, you will be working on **08-onewaysynch-using-vms-batch-cluster-solution** project
So, import it

Truncate products table in db2 now

Open one-way-sync-usingvms-batch-cluster.xml and observe that we have split the single flow in earlier application to 2 flows.

First flow is publishing each message to a VM queue and seconf flow is listening for messages in same vm queue and starting a batch job.

Click on “foreach” scope and observe that we have configured a batch size of 200. So, a list of size 200 will be published to queue. And for those records in a list of size 200, a batch job will be created.

So, if there are 6000 total records, 30 batch jobs will be started in this example.

If we deploy this application in a cluster of size 2, batch jobs will be distributed across nodes as I explained the video

Now let us create a cluster first.

Download mule runtime from the mule website or use the mule runtime given by me.

copy this zip file into a folder with name cluster and extract it.

Rename this extracted folder as mule1

Again extract the same zip file and rename the extracted folder with name mule2

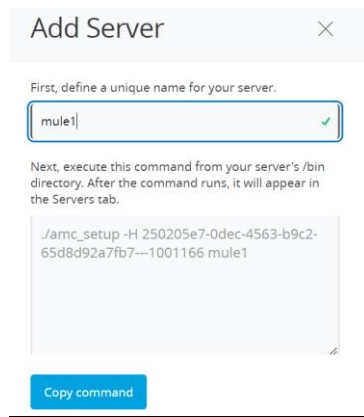
Got to <https://anypoint.mulesoft.com/login/> and login with your credentials.

Go to runtime manager from left menu

Click on servers link in left menu

Click on Add Server button and give servername as mule1.

copy the command shown.



Open command prompt.

Go to mule1/bin folder in command prompt and execute the above command. If you are running on windows machine, remove “.” from the copied command

This command will setup Runtime Manager Agent for your mule runtime.

Now go to mule1/bin folder and execute following command:

mule -M-Dservername=mule1

Open the mule configuration file and observe that we have used \${servername}.yml as externalized file name.

That is the reason we are passing -M-Dservername=mule1 so that it will pick from mule1.yml

Open mule1.yml in src/main/mule and observe that http.port is 8081. So, http Listener start on port 8081

You should see that mule1 server is registered in runtime manager

Click on Add Server button and give servername as mule1.

copy the command shown.

Open another command prompt

Go to mule2/bin folder and execute the above copied command. If you are running on windows machine, remove “./” from the copied command

Now go to mule2/bin folder and execute following command:

mule -M-Dservername=mule2

Open mule2.yml and observe that http.port is 8082. So, http listener deployed on mule2 server listens on port 8082

You should see that mule2 server is registered in runtime manager

Now click on Create Cluster button.

In the next screen, give cluster name as “mycluster” , select mule1 and mule2.

Select clustertype as multicast

Click on Create Cluster

Both mule servers will restart and form a cluster.

Now, right click on the project and export the deployable jar file.

Now, through runtime manager,deploy application to cluster.

Truncate db1.product table.

Give a request to <http://localhost:8081/start>

This will hit http flow in mule1 server. And that flow will publish all records to vmq which is in hazel cast distributed shared memory.

Those records will be received by “VM Listener flows” in both mule1 and mule2.

Open log file with same name as app name in logs folder of mule1 and mule2 servers.

Remember that logs will not be seen on consoles.

You have to look for logs in the log files only.

From the logs you will observe that messages are executed on both the nodes.

So, load is distributed in cluster.