# Mule 4: ARM based deployment

Contents

[Mule 4: ARM based deployment 1](#_Toc38548486)

[Scope 2](#_Toc38548487)

[Description 2](#_Toc38548488)

[Nexus repository set up 2](#_Toc38548489)

[Maven Set up 3](#_Toc38548490)

[ARM server registration 5](#_Toc38548491)

[Sample Mule 4 App and changes in POM and settings.xml 8](#_Toc38548492)

Scope: This document will help the audience to understand the ARM based Mule4 app deployments using Nexus repository being referenced in the POM xml of the project. This document is created assuming the audience with no prior set up of Nexus/ARM/Maven repository set up.

Description:

ARM based deployments is divided into following parts:

1. Nexus repository setup
2. Maven set up
3. ARM setup of local Mule Runtime to AnyPoint Platform
4. A Demo App and deployment into cloudhub based control plane.

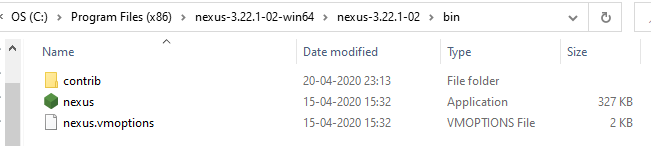
### Nexus repository set up

Step 1. Download the Nexus repository from the [Nexus official site](https://www.sonatype.com/download-oss-sonatype). Since This PoC is conducted on a Windows 64 bit machine intended Nexus set up will be nexus-3.22.1-02-win64.

Depending on the license version of the organization, Nexus Pro version can also be used. Here we will be using the free OSS version of Nexus repository(Nexus Repository Manager OSS 3.x).

Step 2: Once downloaded, unzip the downloaded file to C:\Program Files (x86). It will create two folders : nexus-3.22.1-02-win64 and sonatype-work.

Step 2a: Lets go inside of C:\Program Files (x86)\nexus-3.22.1-02-win64\nexus-3.22.1-02\bin and see if the following structure is available:



Step 2b: Open the command prompt as Administrator and go to above bin location to run :

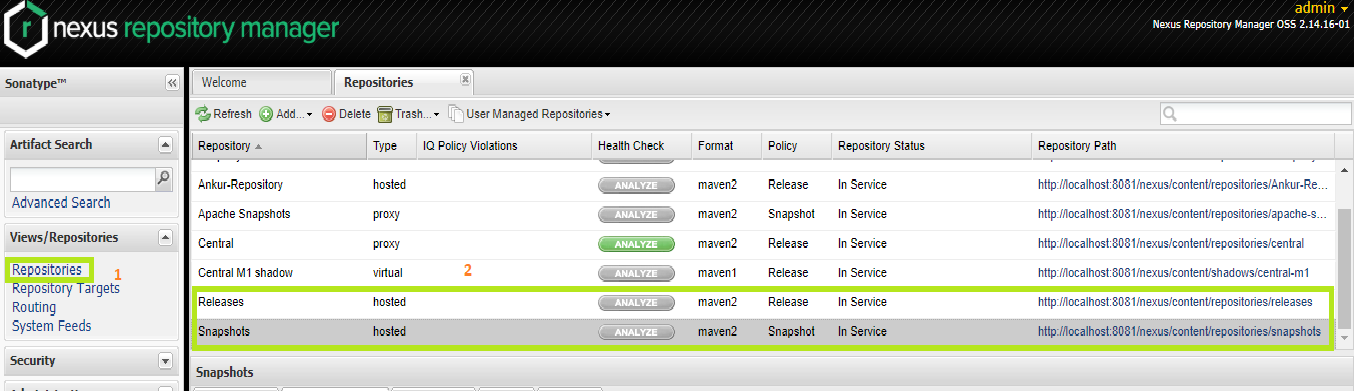
Nexus start

Step 2c: You should be able to see the following nexus service getting installed as windows service under services.msc:



Step 3: Console of the Nexus repository: by default Nexus gets installed at <http://localhost:8081/nexus> with default credentials as username: admin password: admin123

Login with same:



Go to “Repositories” (1) and locate the two repositories “Releases” and “Snapshots” on right hand side (2).

Make a note of the Repositories paths of the respective repositories.

Huraay ! Nexus set up completed !!

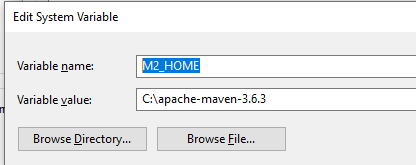
### Maven Set up

Step 1: Download the suitable and attest stable version from the [maven](https://maven.apache.org/download.cgi) official site. We will be using apache-maven-3.6.3 as part of this PoC.

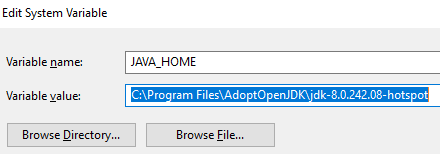
Step 2a:Unzip the maven zip to C:\apache-maven-3.6.3 and look at the contents:

Step 2b: Create env variables for maven:

Step 2b.1: Create system variable as M2\_HOME:



Step 2b.2: Make sure JAVA\_HOME is set to correct version of jdk:

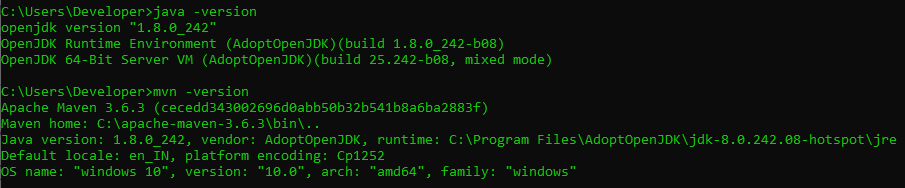


Step 2b.3: Add following entries to Path system variable:



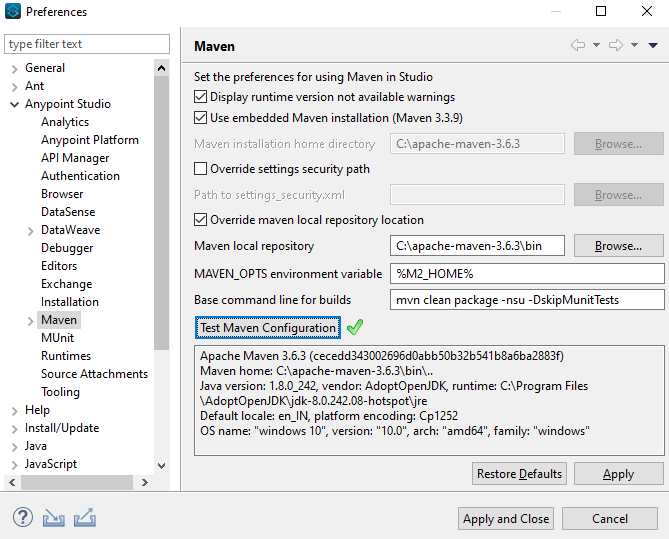
Step 2b.4: Is it all set up correctly ?

Go to cmd and check by yourself:



Step 3: Changes to AnyPoint Studio to configure customized maven :

Open Anypoint Studio. Windows->Preferences ->”Anypoint Studio->Maven”. Click on “Test Maven configuration”



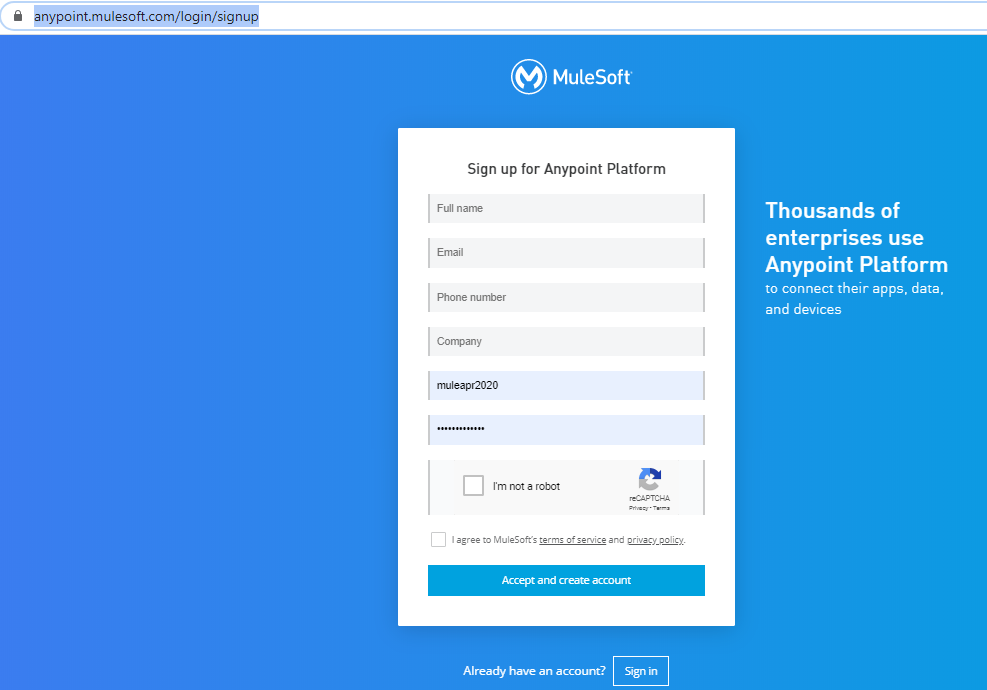
Apply and close.

Congratulations for setting up Maven successfully.

### ARM server registration

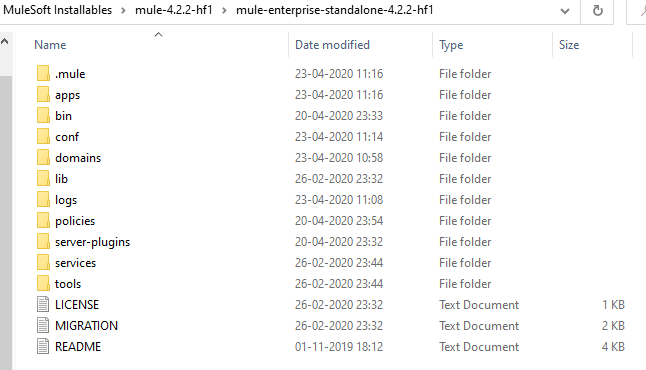
Step 1: This document assumes audience has an active AnyPoint platform license to login with active subscription to Runtime manager. Otherwise, Feel free to go through rest of steps :

Step 1a: Create an account on the Anypoint platform <https://anypoint.mulesoft.com/login/signup>

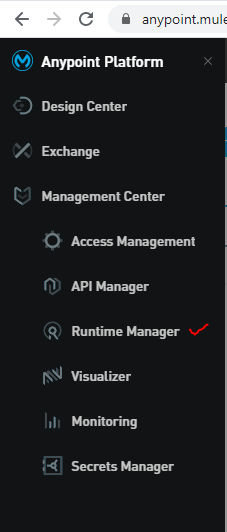


Step 1b: Make sure you have Mule Runtime downloaded already on your system. In case required, you can either download it from https://help.mulesoft.com Or <https://www.mulesoft.com/lp/dl/mule-esb-enterprise>

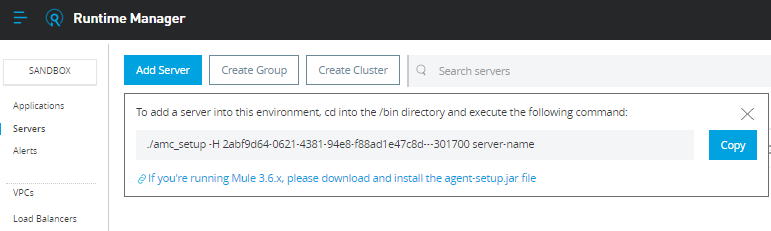
Step 1c: Unzip the downloaded file to C:\Mule-runtime and make sure you are able to see following entries:



Step 1d: Login to AnyPoint platform using the credentials created in previous steps. Go to Runtime Manager:



Step 1e: Go to “Servers” and click on “Add Server” :



Step 1f: Copy the above amc set up command and change the server-name to your server name and hit enter in command prompt:



In windows env, remove ‘./’

Step 1g: Check on Anypoint Platform-> RunTime manager -> Server.

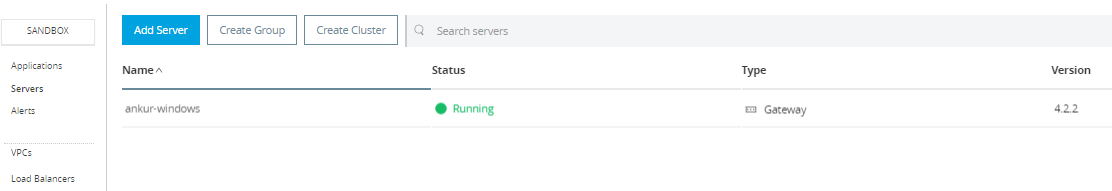
Note: the status will remain in “Created”. You need to start the Mule Runtime from your local machine.

Go to extracted folder of Mule Runtime on your respective filesystem and run the Mule.bat or Command prompt, locate the bin location and run “mule start”.

Once active, you should be able to see following in logs (command prompt):



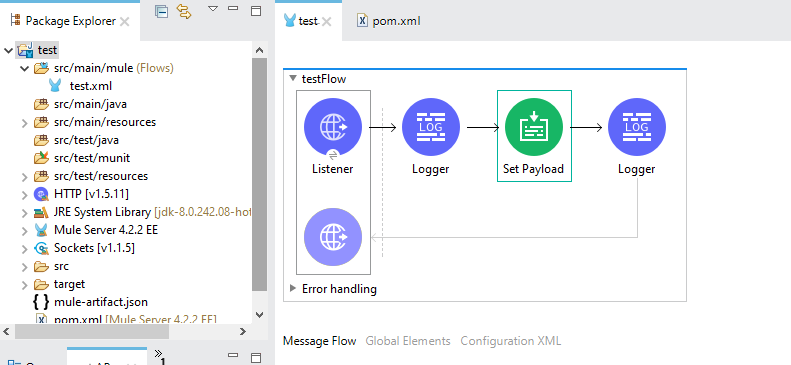
Cross check the status on Anypoint Platform in parallel. It should now changed to “Running”.



Congratulations!! Done with ARM server registration

### Sample Mule 4 App and changes in POM and settings.xml

Step 1: This document assumes that audience has familiarity with basic Anypoint Studio and verse with basic application development.



Pretty much basic, Listener on 8082 port (as Nexus repo keeps 8081 busy) and GET operation with “/test” as url. Logging before and after setting payload.

Config xml:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<mule xmlns:http=*"http://www.mulesoft.org/schema/mule/http"* xmlns=*"http://www.mulesoft.org/schema/mule/core"*

xmlns:doc=*"http://www.mulesoft.org/schema/mule/documentation"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd*

*http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd"*>

<http:listener-config name=*"HTTP\_Listener\_config"* doc:name=*"HTTP Listener config"* doc:id=*"fea1adb1-9113-4e7c-a0d7-8e1ff6420a21"* >

<http:listener-connection host=*"0.0.0.0"* port=*"8082"* />

</http:listener-config>

<flow name=*"testFlow"* doc:id=*"b4f68fb4-8651-454c-8a42-ac17fe011183"* >

<http:listener doc:name=*"Listener"* doc:id=*"00d4b649-250f-422e-a4e4-e6db161b17ff"* config-ref=*"HTTP\_Listener\_config"* path=*"/test"*/>

<logger level=*"INFO"* doc:name=*"Logger"* doc:id=*"467536d1-c282-4aaa-bcae-6be3ed246828"* message=*"#['Flow initiated..']"*/>

<set-payload value=*'#[now() as String {format:"dd-MMM-YYYY HH:mm:ss"}]'* doc:name=*"Set Payload"* doc:id=*"688c5f04-f841-47ff-a5f7-385f1af23c33"* />

<logger level=*"INFO"* doc:name=*"Logger"* doc:id=*"f7f0bb47-88d6-42d9-be2b-1928c7604e0e"* message=*"#['Flow completed...']"*/>

</flow>

</mule>

Step 2: Edit pom.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.mycompany</groupId>

<artifactId>test</artifactId>

<version>1.0.5-SNAPSHOT</version>

<packaging>mule-application</packaging>

<name>test</name>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<app.runtime>4.2.2-hf2</app.runtime>

<mule.maven.plugin.version>3.3.5</mule.maven.plugin.version>

<target>ankur-windows</target>

<target.type>server</target.type>

<environment>Sandbox</environment>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.mule.tools.maven</groupId>

<artifactId>mule-maven-plugin</artifactId>

<version>${mule.maven.plugin.version}</version>

<extensions>true</extensions>

<configuration>

<armDeployment>

<muleVersion>${app.runtime}</muleVersion>

<uri>https://anypoint.mulesoft.com</uri>

<target>${target}</target>

<targetType>${target.type}</targetType>

<username>muleapr2020</username>

<password>Mysore@125239</password>

<environment>${environment}</environment>

</armDeployment>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.mule.connectors</groupId>

<artifactId>mule-http-connector</artifactId>

<version>1.5.11</version>

<classifier>mule-plugin</classifier>

</dependency>

<dependency>

<groupId>org.mule.connectors</groupId>

<artifactId>mule-sockets-connector</artifactId>

<version>1.1.5</version>

<classifier>mule-plugin</classifier>

</dependency>

</dependencies>

<repositories>

<repository>

<id>anypoint-exchange-v2</id>

<name>Anypoint Exchange</name>

<url>https://maven.anypoint.mulesoft.com/api/v2/maven</url>

<layout>default</layout>

</repository>

<repository>

<id>mulesoft-releases</id>

<name>MuleSoft Releases Repository</name>

<url>https://repository.mulesoft.org/releases/</url>

<layout>default</layout>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>mulesoft-releases</id>

<name>mulesoft release repository</name>

<layout>default</layout>

<url>https://repository.mulesoft.org/releases/</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</pluginRepository>

</pluginRepositories>

<distributionManagement>

<repository>

<id>nexus</id>

<name>Releases</name>

<url>http://localhost:8081/nexus/content/repositories/releases</url>

</repository>

<snapshotRepository>

<id>nexus</id>

<name>Snapshot</name>

<url>http://localhost:8081/nexus/content/repositories/snapshots</url>

</snapshotRepository>

</distributionManagement>

</project>

Properties section: This section can be used to put the env specific details in properties tag and it can even be referred from Project level properties file as well.

<target>ankur-windows</target>

<target.type>server</target.type>

<environment>Sandbox</environment>

armDeployment section: This section contains details specific to armDeployment. As mentioned value specific to tags can be obtained from pom.xml based variables or from .properties from project level:

<armDeployment>

<muleVersion>${app.runtime}</muleVersion>

<uri>https://anypoint.mulesoft.com</uri>

<target>${target}</target>

<targetType>${target.type}</targetType>

<username>muleapr2020</username>

<password>Mysore@125239</password>

<environment>${environment}</environment>

</armDeployment>

distributedElement: this section will contain the details specific to Nexus repository as mentioned in first section of the document:

<distributionManagement>

<repository>

<id>nexus</id>

<name>Releases</name>

<url>http://localhost:8081/nexus/content/repositories/releases</url>

</repository>

<snapshotRepository>

<id>nexus</id>

<name>Snapshot</name>

<url>http://localhost:8081/nexus/content/repositories/snapshots</url>

</snapshotRepository>

</distributionManagement>

Alongwith above, some changes are required in settings.xml as well which can be located under C:\Users\Developer\.m2\settings.xml (or can be copied from C:\apache-maven-3.6.3\conf\settings.xml)

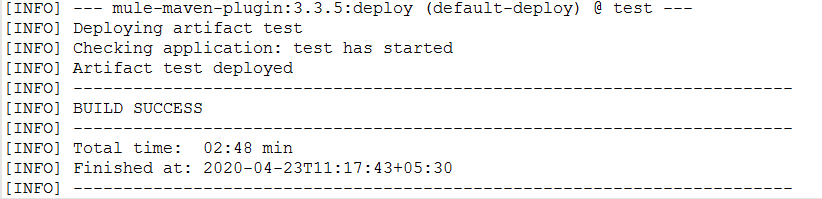




Step 3: We are done with pom.xml and lets try with maven commands. Open command prompt as administrator and point to project location and run

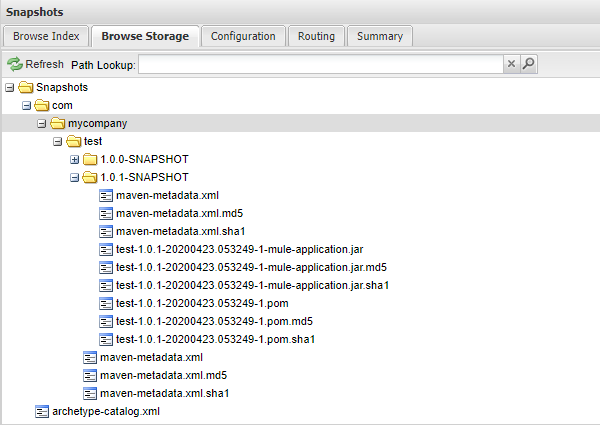
C:\Users\Developer\AnypointStudio\studio-workspace\test>mvn clean package deploy -DmuleDeploy

Command prompt Console logs:

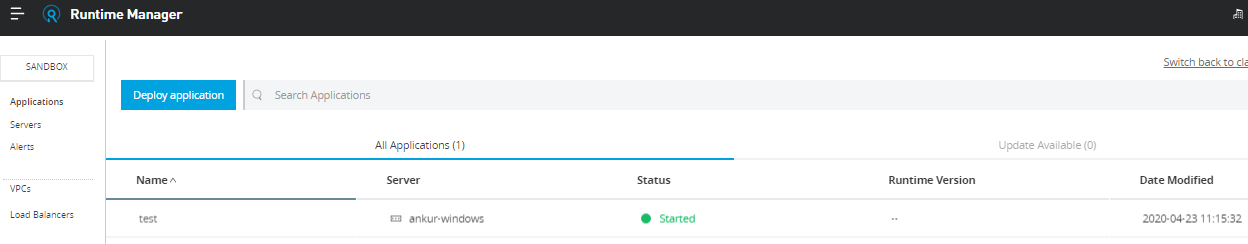




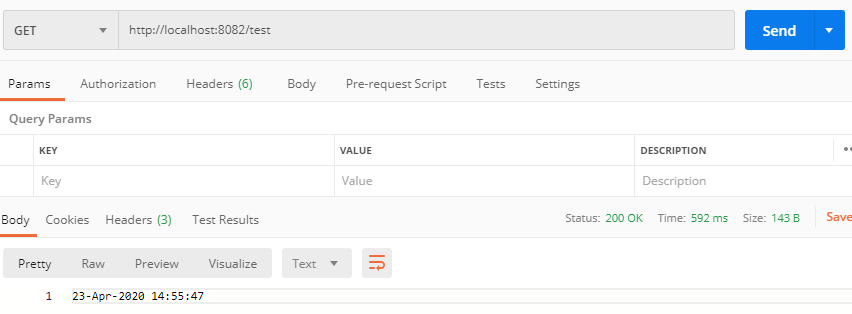
Step 4: Check in nexus snapshot repo, 1.0.1-SNAPSHOT should be available in snapshot repo.



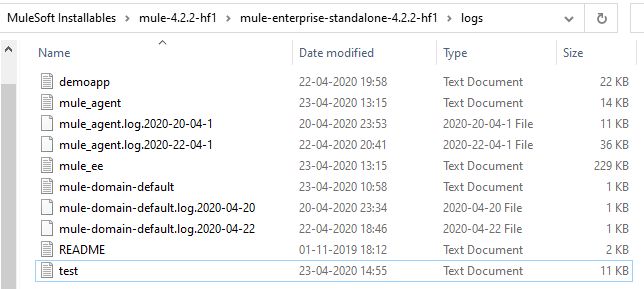
Step 5: Check in AnypointPlatform -> runtime manager -> Applications



Step 6: Test it using PostMan:



Step 7: logs in installed server:



test.txt:

INFO 2020-04-23 14:55:47,916 [[MuleRuntime].cpuLight.08: [test].testFlow.CPU\_LITE @7aeebbe1] [event: 6a7934a0-8544-11ea-841f-6236dd9be254] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: Flow initiated..

INFO 2020-04-23 14:55:47,926 [[MuleRuntime].cpuLight.08: [test].testFlow.CPU\_LITE @7aeebbe1] [event: 6a7934a0-8544-11ea-841f-6236dd9be254] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: Flow completed...