

## Deploy Mule Application to CloudHub using Maven

### Mule maven plugin

- ◆ We have seen how we can do manual deployment of mule application to CloudHub in previous sessions. By using the Mule maven plugin, we can automate our Mule application deployment to CloudHub.
- ◆ We need to add cloud hub deployment configuration to mule-maven-plugin under build section of pom.xml and using below commands we can deploy and un-deploy the applications to CloudHub.
  - ◆ To deploy : `mvn clean package deploy -DmuleDeploy`
  - ◆ To un-deploy: `mvn mule:undeploy`

>> Configure cloudhub details in pom.xml

Get the config details from link

Link : <https://docs.mulesoft.com/mule-runtime/latest/deploy-to-cloudhub-2>

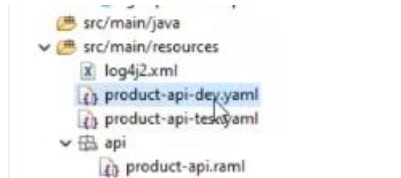
```
<configuration>
  <cloudHubDeployment>
    <uri>https://anypoint.mulesoft.com</uri>
    <muleVersion>4.3.0</muleVersion>
    <username>chinna04</username>
    <password>Realtime@123</password>
    <applicationName>dev-hello-mule-api</applicationName>
    <environment>Sandbox</environment>
    <region>us-east-1</region>
    <workers>1</workers>
    <workerType>MICRO</workerType>
    <objectStoreV2>true</objectStoreV2>
    <properties>
      <http.port>8081</http.port>
    </properties>
  </cloudHubDeployment>
</configuration>
</plugin>
</plugins>
```

To deploy: go to project location on local and open cmd and run “mvn clean deploy -DmuleDeploy”

### Mule maven plugin for each environment

- ◆ In last session we have seen how we can automate the CloudHub deployment using mule maven plugin.
- ◆ If we have multiple environments, then to deploy to each environment we need to refactor the build properties again and again.
- ◆ To avoid this issue, we can configure each environment using profiles of pom.xml.
- ◆ You must have completed Anypoint platform section before this demo.

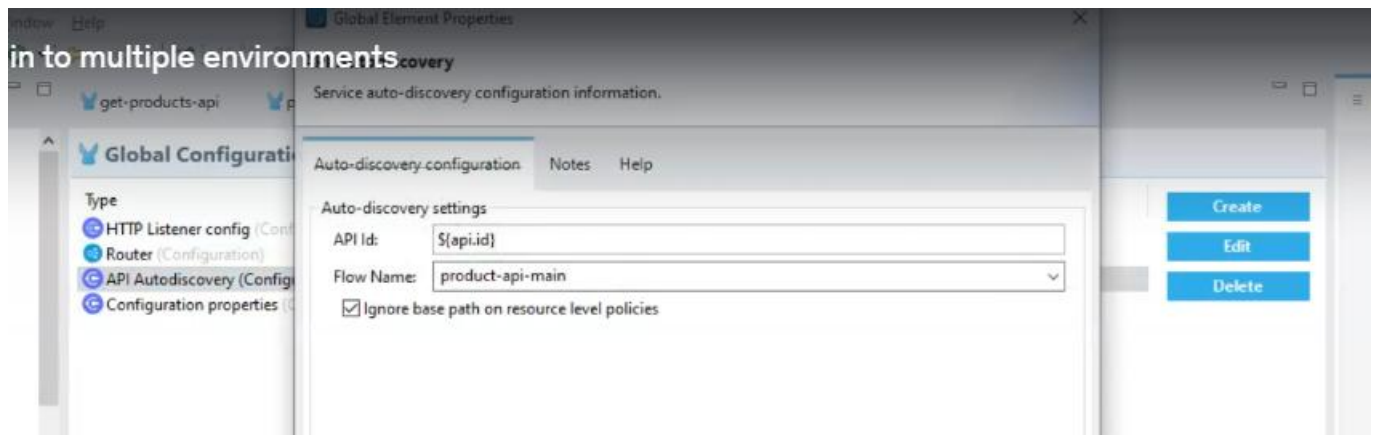
For each env we need to create yaml file under src/main/resources



In those files, we will put all the properties/ configurations according to env(DEV / TEST / PROD)



In configuration properties, It will refer to yaml file, which is dev / test. We need to configure env in dev and test pom.xml . so it will refer to the correct environment.



For auto discovery also we need to refer in pom.xml



>> api.id --->> instance id

To enable autodiscovery, we need to provide runtime properties that are client id and client secret.

Link: <https://docs.mulesoft.com/mule-gateway/mule-gateway-org-credentials-mule4>

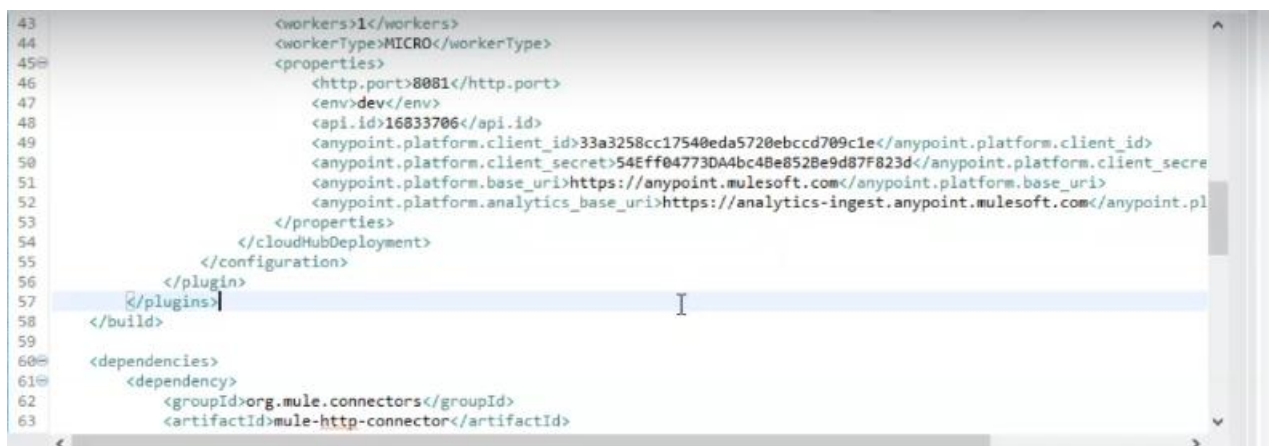
```
<anypoint.platform.client_id>XXXXXXX</anypoint.platform.client_id>
<anypoint.platform.client_secret>XXXXXXX</anypoint.platform.client_secret>
<anypoint.platform.base_uri>=https://anypoint.mulesoft.com</anypoint.platform.base_uri>
<anypoint.platform.analytics_base_uri>=https://analytics-
ingest.anypoint.mulesoft.com</anypoint.platform.analytics_base_uri>
```

Paste this in pom.xml

To get client id and secret

Go to anypoint > Access management > Environments > click on dev to see client id and secrets

Same do this for test and prod env



```
43         <workers>1</workers>
44         <workerType>MICRO</workerType>
45         <properties>
46             <http.port>8081</http.port>
47             <env>dev</env>
48             <api.id>16833706</api.id>
49             <anypoint.platform.client_id>33a3258cc17540eda5720ebccd709c1e</anypoint.platform.client_id>
50             <anypoint.platform.client_secret>54Eff04773DA4bc4Be852Be9d87F823d</anypoint.platform.client_secret>
51             <anypoint.platform.base_uri>https://anypoint.mulesoft.com</anypoint.platform.base_uri>
52             <anypoint.platform.analytics_base_uri>https://analytics-ingest.anypoint.mulesoft.com</anypoint.pl
53         </properties>
54     </cloudHubDeployment>
55 </configuration>
56 </plugin>
57 </plugins>
58 </build>
59
60 <dependencies>
61     <dependency>
62         <groupId>org.mule.connectors</groupId>
63         <artifactId>mule-http-connector</artifactId>
```

To automate this deployment, we need to create profiles in pom.xml



```
<packaging>mule-application</packaging>
<name>product-api</name>
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <app.runtime>4.3.0-20210119</app.runtime>
    <mule.maven.plugin.version>3.4.2</mule.maven.plugin.version>
</properties>
<profiles>
    <profile>
        <id>dev</id>
    </profile>
</profiles>
```

And copy build section and paste inside profile

```

<app.runtime>4.3.0-20210119</app.runtime>
<mule.maven.plugin.version>3.4.2</mule.maven.plugin.version>
</properties>
<profiles>
  <profile>
    <id>dev</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.0.0</version>
        </plugin>
        <plugin>
          <groupId>org.mule.tools.maven</groupId>
          <artifactId>mule-maven-plugin</artifactId>
          <version>${mule.maven.plugin.version}</version>
          <extensions>true</extensions>
          <configuration>
            <cloudHubDeployment>
              <uri>https://anypoint.mulesoft.com</uri>
              <muleVersion>4.3.0</muleVersion>
              <username>chinna04</username>
              <password>Realtime@123</password>
              <applicationName>abc-dev-product-api</applicationName>
              <environment>dev</environment>
              <region>us-east-1</region>
              <workers>1</workers>
              <workerType>MICRO</workerType>
              <properties>
                <http.port>8081</http.port>
                <env>dev</env>
                <api.id>16833706</api.id>
                <anypoint.platform.client_id>33a3258cc17540eda5720ebccd709c1e</anypoint.platform.client_id>
                <anypoint.platform.client_secret>54Eff04773DA4bc48e8528e9d87F823d</anypoint.platform.client_secret>
                <anypoint.platform.base_uri>https://anypoint.mulesoft.com</anypoint.platform.base_uri>
                <anypoint.platform.analytics_base_uri>https://analytics-ingest.anypoint.mulesoft.com</anypoint.platform.analytics_base_uri>
              </properties>
            </cloudHubDeployment>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>

```

Next copy the whole profile section, under profiles have another profile for test env.

```

</profile>
<profile>
  <id>test</id>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-clean-plugin</artifactId>
        <version>3.0.0</version>
      </plugin>
      <plugin>
        <groupId>org.mule.tools.maven</groupId>
        <artifactId>mule-maven-plugin</artifactId>

```

There are some common things (dependency, plugins) b/w dev and test profiles, we can paste in build (global)

Final global build

```

</profile>
</profiles>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-clean-plugin</artifactId>
      <version>3.0.0</version>
    </plugin>
    <plugin>
      <groupId>org.mule.tools.maven</groupId>
      <artifactId>mule-maven-plugin</artifactId>
      <version>${mule.maven.plugin.version}</version>
      <extensions>true</extensions>
      <configuration>
        <cloudHubDeployment>
          <uri>https://anypoint.mulesoft.com</uri>
          <muleVersion>4.3.0</muleVersion>
          <username>chinna04</username>
          <password>Realtime@123</password>
          <properties>
            <http.port>8081</http.port>
            <anypoint.platform.base_uri>https://anypoint.mulesoft.com</anypoint.platform.base_uri>
            <anypoint.platform.analytics_base_uri>https://analytics-ingest.anypoint.mulesoft.com</anypoint.platform.analytics_base_uri>
          </properties>
        </cloudHubDeployment>
      </configuration>
    </plugin>
  </plugins>
</build>

```

We need to put object store in global build



```

<configuration>
  <cloudHubDeployment>
    <uri>https://anypoint.mulesoft.com</uri>
    <muleVersion>4.3.0</muleVersion>
    <username>chinna04</username>
    <password>Realtime@123</password>
    <objectStoreV2>true</objectStoreV2>
    <properties>
      <http.port>8081</http.port>
      <anypoint.platform.base_uri>https://anypoint.mulesoft.com</anypoint.platform.base_u
      <anypoint.platform.analytics_base_uri>https://analytics-ingest.anypoint.mulesoft.cc
    </properties>
  </cloudHubDeployment>
</configuration>

```

To deploy it: go to project location on local and open cmd and run “mvn clean deploy -DmuleDeploy”

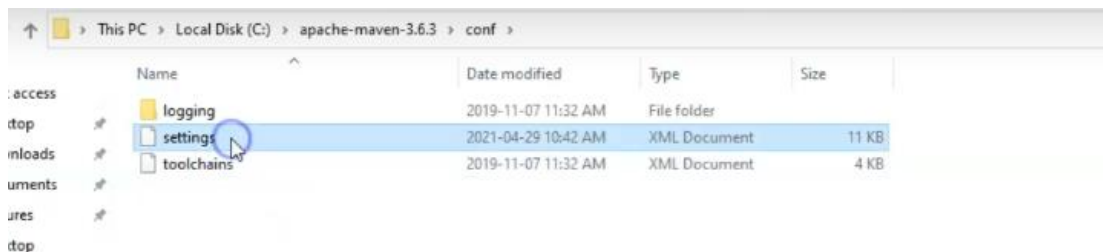
For particular dev env “mvn clean deploy -Pdev -DmuleDeploy”

For test env “mvn clean deploy -Ptest -DmuleDeploy”

## Injecting pom properties via settings.xml

We need to put username and password in settings file in .m2 folder. It will takes these credentials from here

Go to maven installed location > conf > settings file > search for servers



```

<profiles>
  <profile>
    <id>dev</id>
    <properties>
      <username>chinna04</username>
      <password>Realtime@123</password>
      <anypoint.platform.client_id>33a3258cc17540eda5720ebccd709c1e</anypoint.pla
      <anypoint.platform.client_secret>54Eff04773DA4bc4Be852Be9d87F823d</anypoint
    </properties>
  </profile>
</profiles>
<!-- activeProfiles

```

We can refer this to pom.xml

```

<cloudHubDeployment>
  <uri>https://anypoint.mulesoft.com</uri>
  <username>${username}</username>
  <password>${password}</password>
  <muleVersion>4.3.0</muleVersion>
  <objectStoreV2>true</objectStoreV2>
  <properties>
    <http.port>8081</http.port>
    <anypoint.platform.base_uri>https://anypoint.mulesoft.com</any
    <anypoint.platform.analytics_base_uri>https://analytics-ingest

```

```
32 <cloudHubDeployment>
33   <applicationName>abc-dev-product-api</applicationName>
34   <environment>dev</environment>
35   <region>us-east-1</region>
36   <workers>1</workers>
37   <workerType>MICRO</workerType>
38   <properties>
39     <env>dev</env>
40     <api.id>16833786</api.id>
41     <anypoint.platform.client_id>${anypoint.platform.client_id}</anypoint.platform.client_id>
42     <anypoint.platform.client_secret>${anypoint.platform.client_secret}</anypoint.platform.client_secret>
43   </properties>
44 </cloudHubDeployment>
45 </configuration>
```

Do it for test env

```
</profile>
<profile>
  <id>test</id>
  <properties>
    <username>chinna04</username>
    <password>Realtime@123</password>
    <anypoint.platform.client_id>33a3258cc17540eda5720ebccd709c1e</anypoint.pla
    <anypoint.platform.client_secret>54Eff04773DA4bc4Be852Be9d87F823d</anypoint
  </properties>
</profile>
</profiles>
<!-- activeProfiles
| List of profiles that are active for all builds.
|
```

Another way to deploy project to cloudHub using git without using cmd code.

Copy all the source folder to git folder

Upload to git