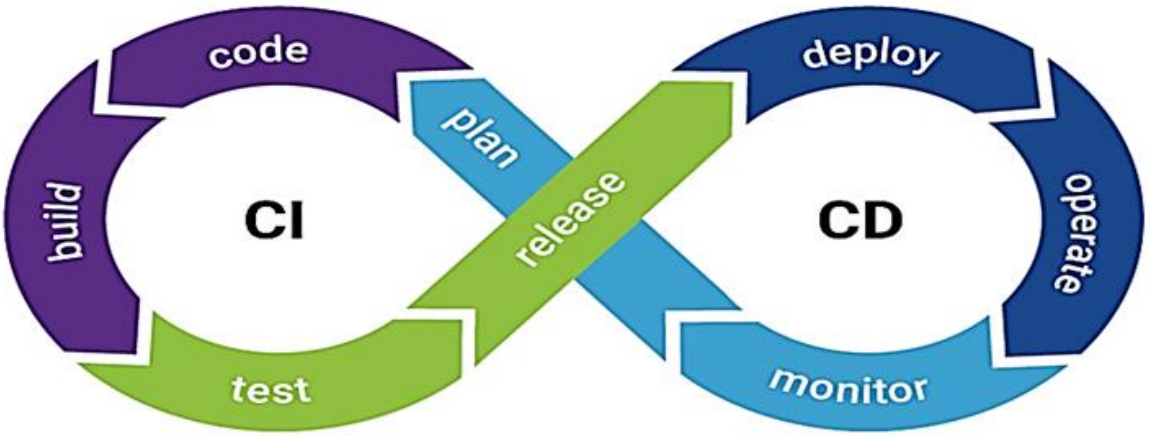
As part of this article let's have deep dive on Continuous Integration and Continuous Delivery with AnyPoint Platform - CloudHub.

Before we begin to understand CICD in CloudHub let's first touch base on few jargons being use in the article.

- **CloudHub** is an integration platform as a service (iPaaS) where you can deploy cross-cloud integration applications in the cloud, create new APIs on top of existing data sources, integrate on-premises applications with cloud services, and much more.
- **Continuous Integration** An approach to be continually validating the state of a codebase through automated testing. Achieved through integration with source control/version control and artifact store.
- **Continuous Delivery / Deployment** An approach to regularly deploying artifacts that successfully pass the CI phase to ensure confidence around the deployment
- **API Auto Discovery** API Autodiscovery is a mechanism that manages an API from API Manager by pairing the deployed application to an API created on the platform. API Management includes tracking, enforcing policies if you apply any, and reporting API analytics. Critical to the Autodiscovery process is identifying the API by providing the API name and version.
- **API Policies** Policies enable you to enforce regulations to help manage security, control traffic, and improve adaptability of your APIs. For example, a policy can control authentication, access, allotted consumption, and service level access (SLA). You can implement all these regulations with no modification to the code implementation. Mulesoft provides ready-to-use default policies that are shipped with the product.



Pre requisites :

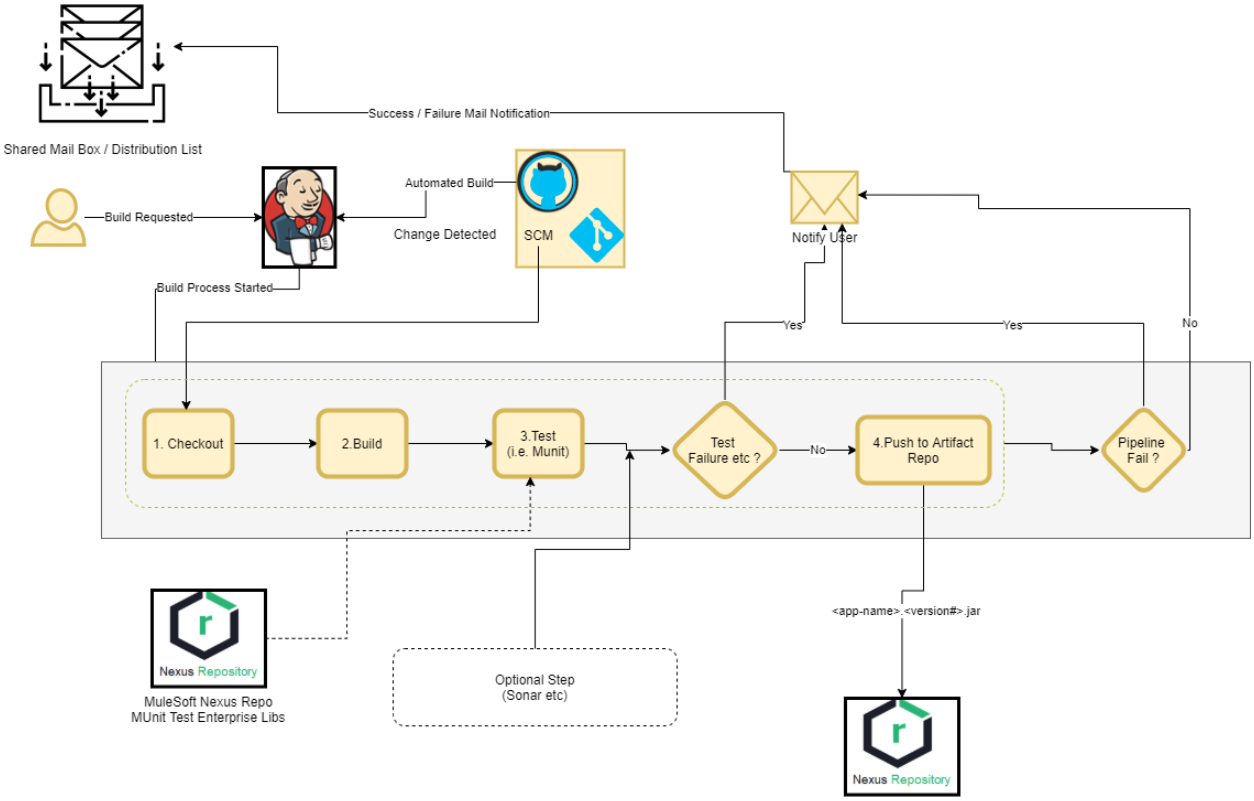
CICD is a culture set of best practices to be followed to have code changes deployed more frequently and reliably.

Before understanding the overall CICD with AnyPoint Platform - CloduHub we should have knowledge of how API Deployment , Auto Discovery and Policy works in CloudHub.

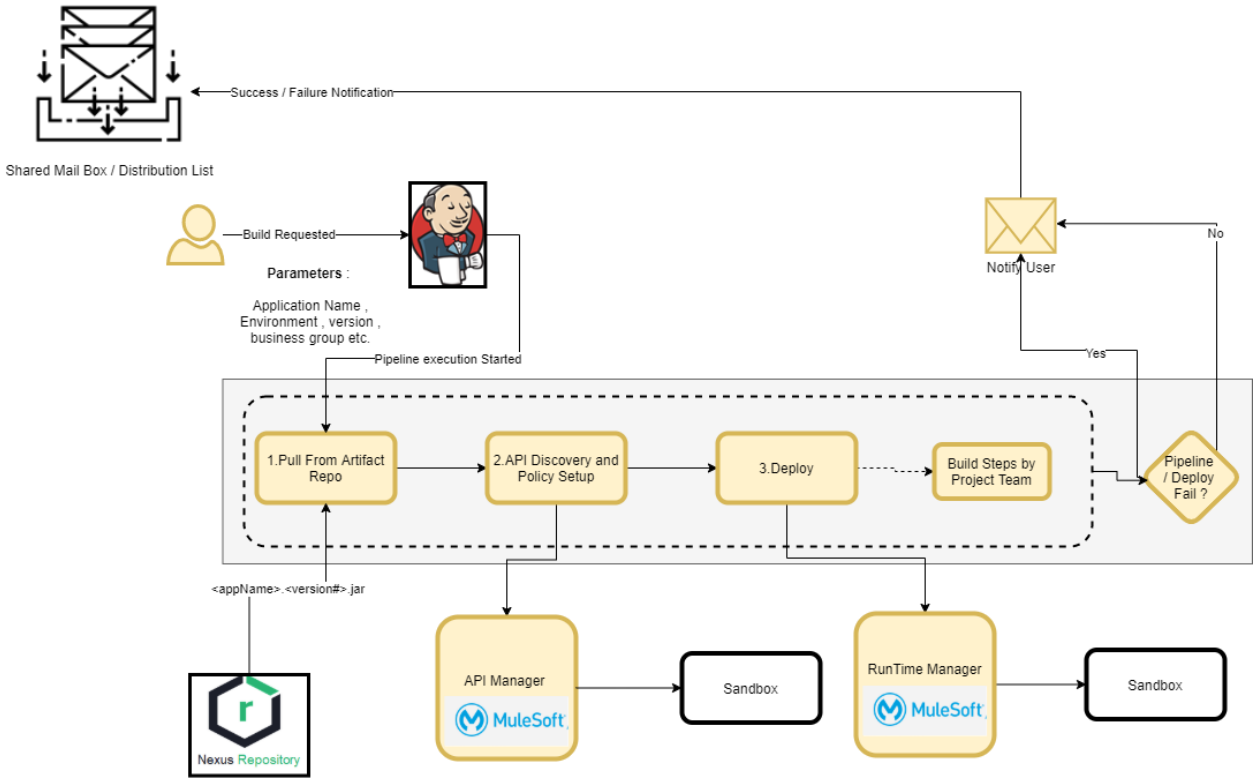
Refer these topics for more details on API Deployment , Auto Discovery and Policies works in CloudHub

Basic understanding of Anypoint platform

CI Flow


Shared Mail Box / Distribution List
Success / Failure Mail Notification
Build Requested
Automated Build
Change Detected
SCM
Notify User
Build Process Started
Yes
Yes
No
1. Checkout
2.Build
3.Test
(i.e. Munit)
Test
Failure etc ?
No
4.Push to Artifact
Repo
Pipeline
Fail ?
Nexus Repository
MuleSoft Nexus Repo
MUnit Test Enterprise Libs
Optional Step
(Sonar etc)
<app-name>.<version#>.jar
Nexus Repository

CD Flow

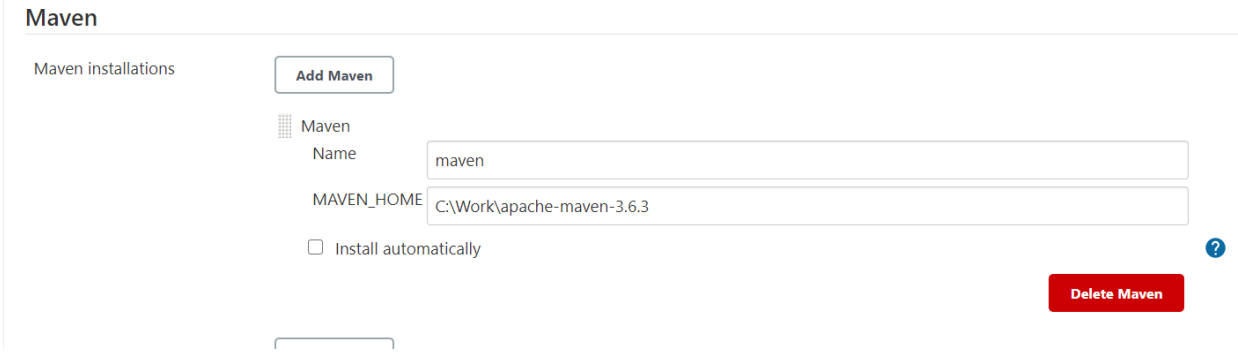


So let's begin the setup

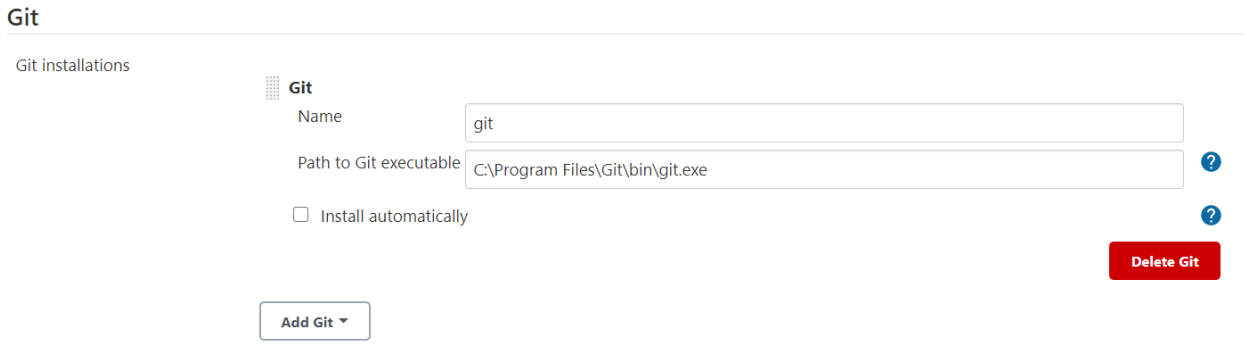
Tools with Initial Instruction to setup the Environment :

- Maven [Download From Here]
- Git [Download From Here]
- Jenkins (make sure you have below plugins installed)
    - [CloudHub Deployer]

- Maven Integration plugin

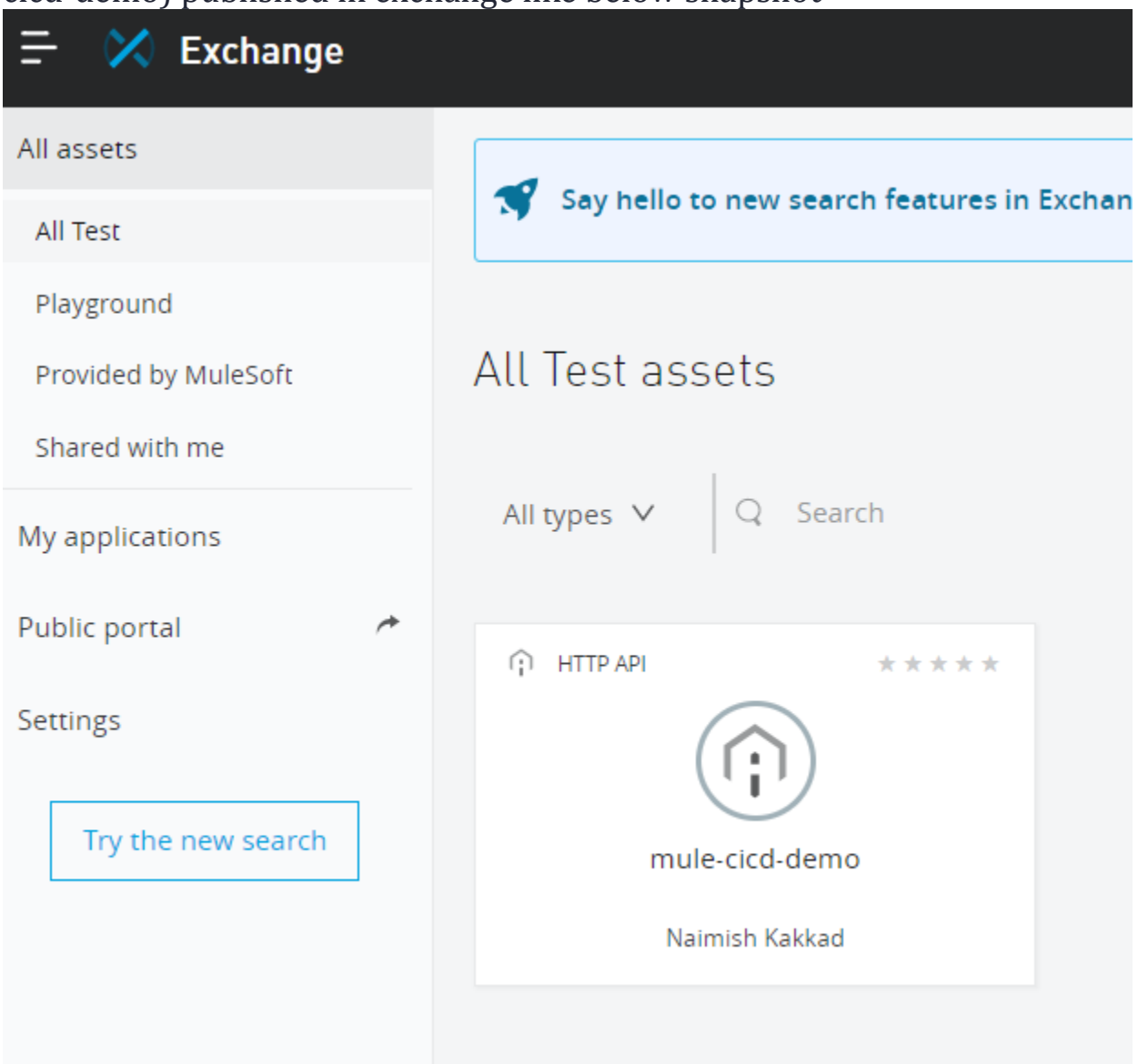


- Groovy
- Nexus Artifact Uploader
- Email Extension Plugin
- Pipeline Utility Steps
- Git plugin



- AnyPoint Studio Download From Here
- Nexus repo Download from here
- Active gmail account
- Active Anypoint Platform Account Sign Up Here

- Make sure you have specification/simple api asset (i.e. mule-cicd-demo) published in exchange like below snapshot



- Github Sign Up Here

Please follow below steps before starting with Jenkins Pipeline configurations

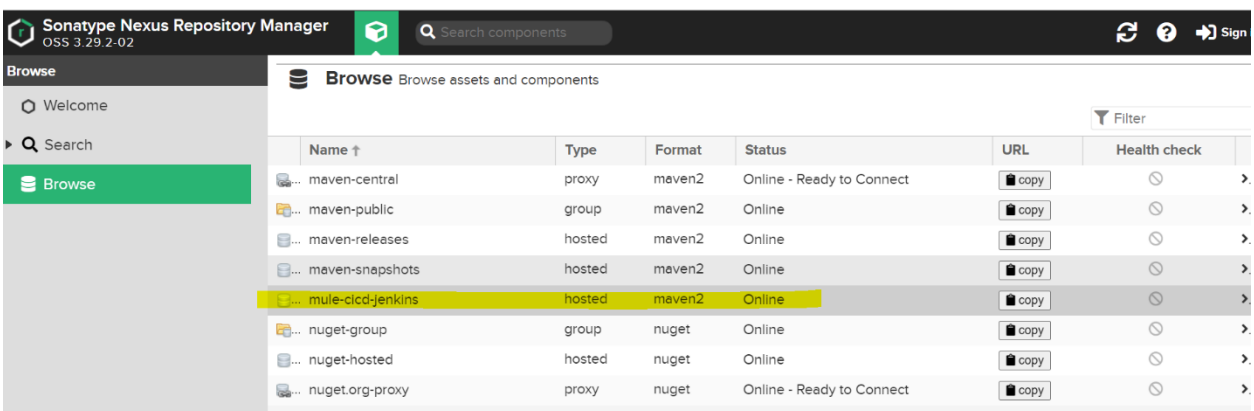
Nexus Repo OSS :

Verify Installation

```
C:\Work\nexus-3.29.2-02-win64\nexus-3.29.2-02\bin>nexus /run
2021-03-23 14:13:41,438+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.pax.logging.NexusLogActivator - start
2021-03-23 14:13:41,954+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.features.internal.FeaturesWrapper - Fast FeaturesService starting
2021-03-23 14:13:43,154+0530 WARN  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4JInitialiser - Your logging framework class org.ops4j.pax.logg
ing.slf4j.Slf4jLogger is not known - if it needs access to the standard println methods on the console you will need to register it by calling registerLoggingSystemPackage
2021-03-23 14:13:43,154+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Package org.ops4j.pax.logging.slf4j registered; all classe
s within it or subpackages of it will be allowed to print to System.out and System.err
2021-03-23 14:13:43,154+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Replaced standard System.out and System.err PrintStreams w
ith SLF4JPrintStreams
2021-03-23 14:13:43,154+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Redirected System.out and System.err to SLF4J for this con
text
```

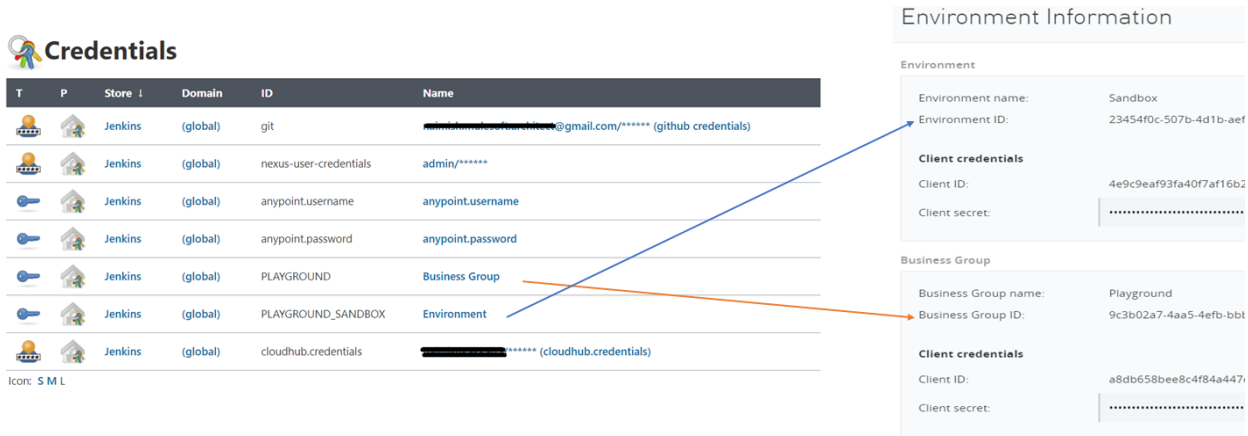
```
 default ValidatorFactory
2021-03-23 14:14:20,831+0530 INFO  [jetty-main-1] *SYSTEM org.sonatype.nexus.siesta.SiestaServlet - Initialized
2021-03-23 14:14:20,837+0530 INFO  [jetty-main-1] *SYSTEM org.sonatype.nexus.repository.httpbridge.internal.ViewServlet - Initialized
2021-03-23 14:14:21,355+0530 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.handler.ContextHandler - Started o.e.j.w.WebAppContext@333cba3c{Sonatype Nexus,/
C:/Work/nexus-3.29.2-02-win64/nexus-3.29.2-02/public/,AVAILABLE}
2021-03-23 14:14:21,408+0530 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.AbstractConnector - Started ServerConnector@406f1159{HTTP/1.1, (http/1.1)}{0.0.0
2021-03-23 14:14:21,408+0530 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.Server - Started @45215ms
2021-03-23 14:14:21,418+0530 INFO  [jetty-main-1] *SYSTEM org.sonatype.nexus.bootstrap.jetty.JettyServer -
-------------------------------------------------

Started Sonatype Nexus OSS 3.29.2-02

-------------------------------------------------
```

- Create maven2 hosted repo to save artifacts produced from CI Flow.



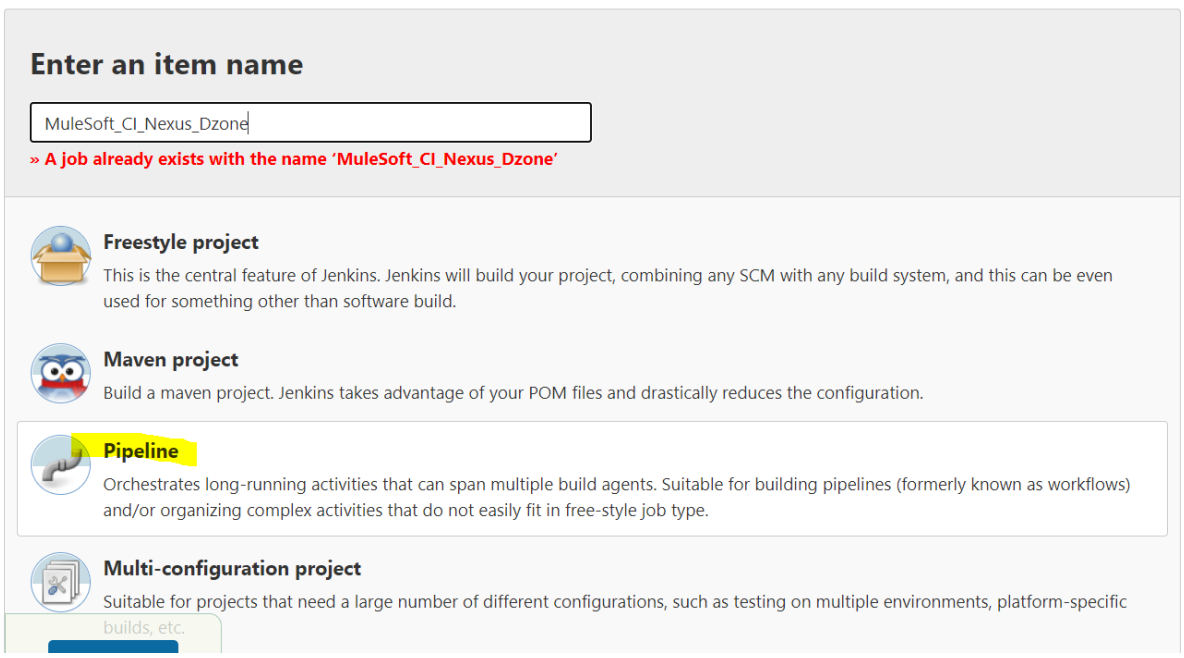
Jenkins :

Credentials Setup :



Create CI Pipeline :

(As I have already have existing job with same name it is giving me error but it will not occur for you as you are creating the same for the first time)



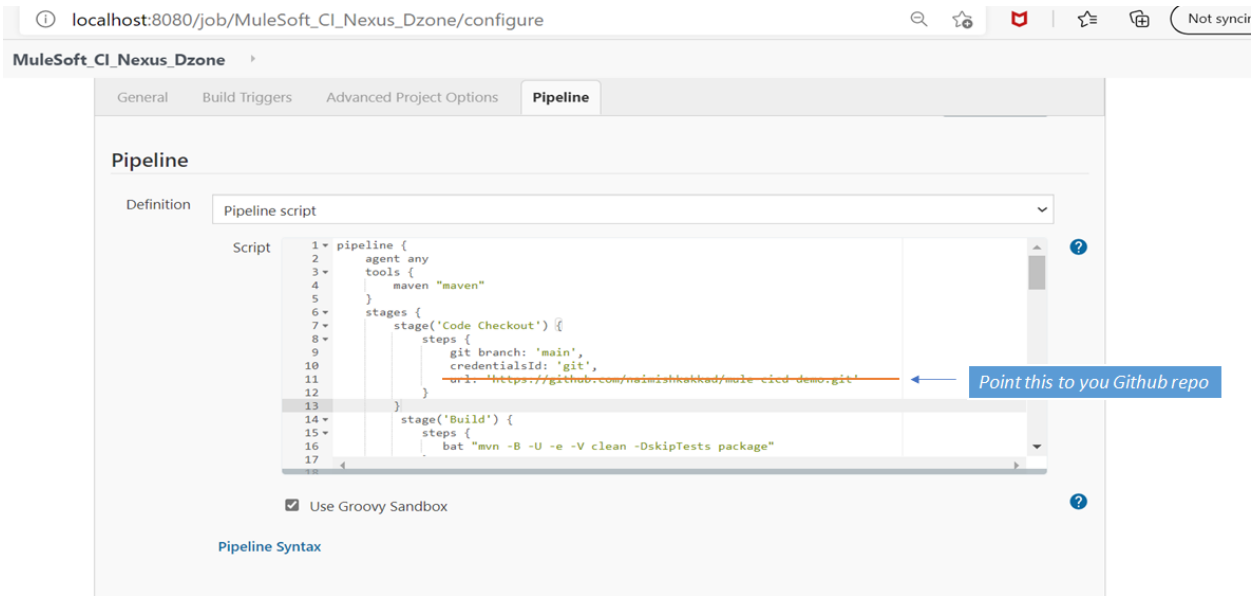
Select Pipeline Script :

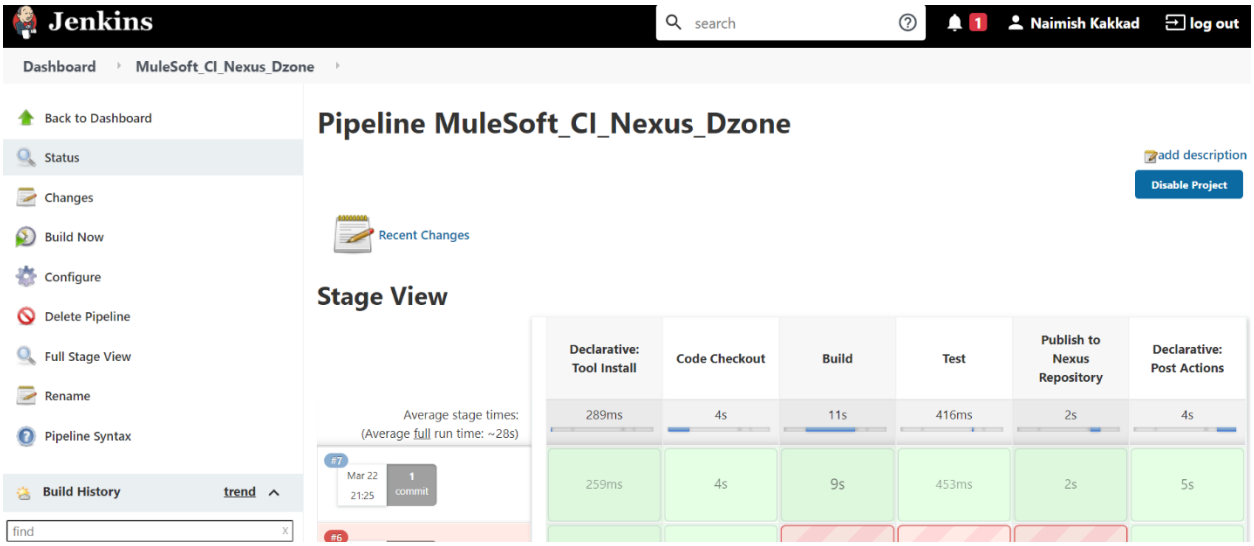
Download Pipeline Script from here :

https://github.com/naimishkakkad/jenkinspipelines/blob/main/CIPipeline.txt

Sample Code - https://github.com/naimishkakkad/mule-cicd-demo

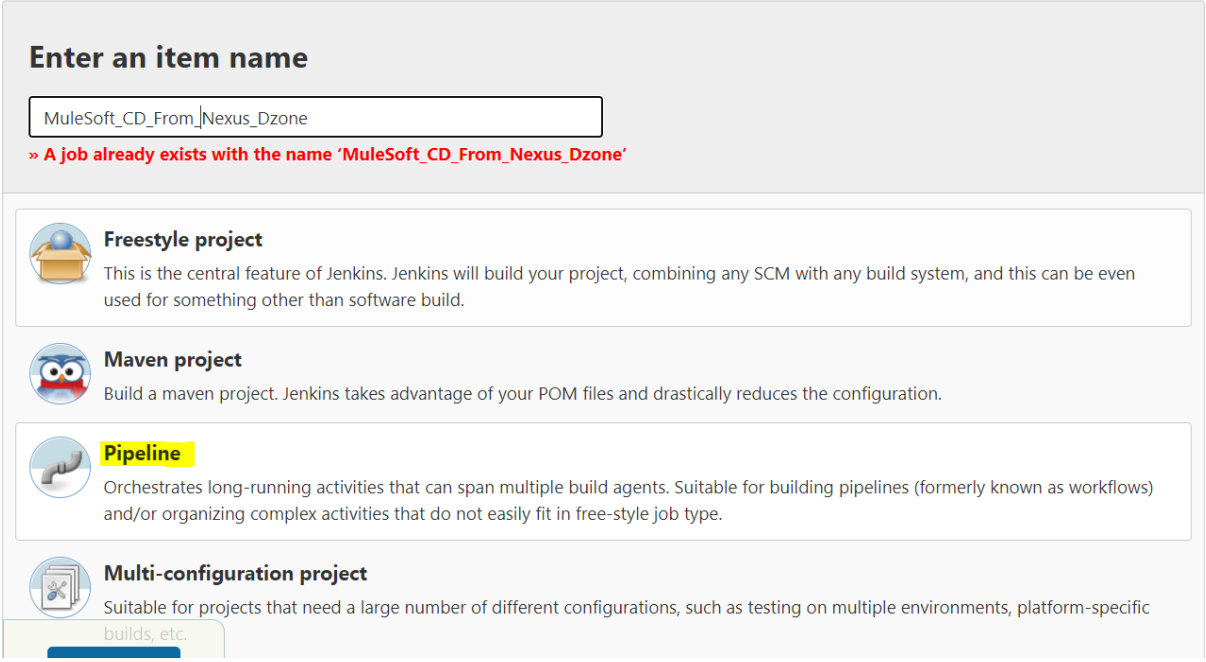
(Clone this and try this in your own GitHub repo)





Create CD Pipeline :

(As I have already have existing job with same name it is giving me error but it will not occur for you as you are creating the same for the first time)



Select Pipeline Script :

Download Pipeline Script from here :

https://github.com/naimishkakkad/jenkinspipelines/blob/main/CDPipeline.txt

Sample Code - https://github.com/naimishkakkad/mule-cicd-demo

(Clone this and try this in your own GitHub repo)

Add Parameters to the pipeline like below

Pipeline MuleSoft_CD_From_Nexus_Dzone

This build requires parameters:

project_name	mule-cicd-demo Project name
version	1.0.1 Project Version
business_group	PLAYGROUND business group
target_env	SANDBOX
cloudhub_app_name	mule-cicd-demo-jenkins
env_client_id	Concealed — Change Password
env_client_secret	Concealed — Change Password

Build

Pipeline

Definition: Pipeline script

Script

```
import org.jenkinsci.plugins.cloudhubdeployer.data.*;

pipeline {
    agent any
    tools {
        maven "maven"
    }
    options {
        skipDefaultCheckout(true)
    }
    stages {
        stage('Fetch Configuration Data') {
            steps {
                cleanWs()
                git branch: 'main',
                credentialsId: 'git',
                url: 'https://github.com/naimishkakkad/mule-cicd-demo.git'
```

☐ Use Groovy Sandbox

Jenkins | search | 1 | Naimish Kakkad | log out

Dashboard › MuleSoft_CD_From_Nexus_Dzone

- Back to Dashboard
- Status
- Changes
- Build with Parameters
- Configure
- Delete Pipeline
- Full Stage View
- Rename
- Pipeline Syntax

Build History — trend

find

Pipeline MuleSoft_CD_From_Nexus_Dzone

add description

Disable Project

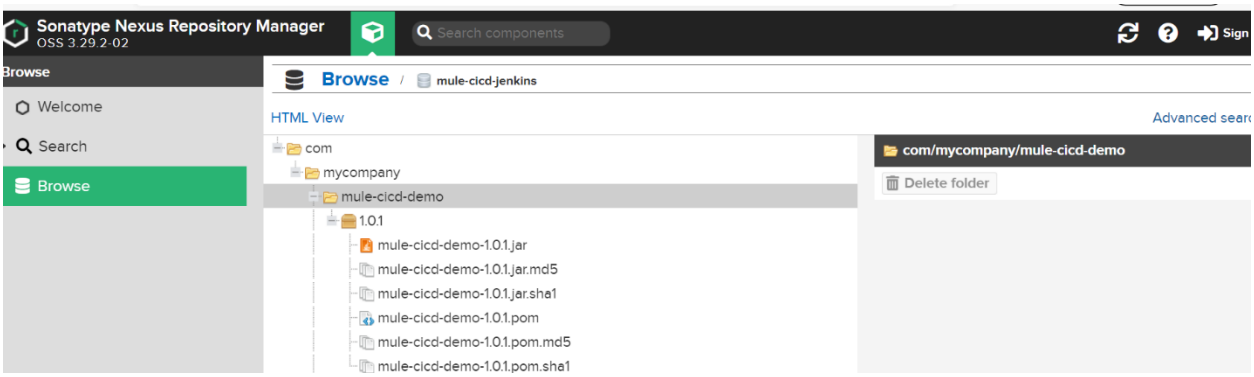
Recent Changes

Stage View

	Declarative: Tool Install	Fetch Configuration Data	Fetch Artifact	API Discovery and Policy Setup	Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~2min 24s)	221ms	4s	9s	5s	5min 5s	3s
#8 Mar 22 22:17 No Changes	188ms	5s	7s	7s	1min 55s	6s
#7						

That's it :)

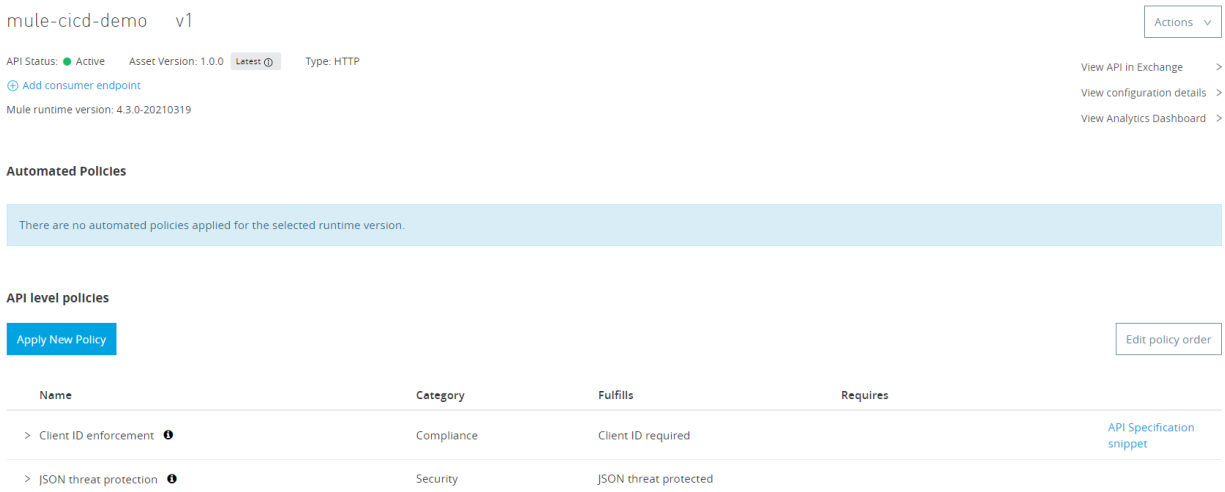
Just Hit the "Build Now" button for CI Pipeline and verify if Jar is published to Nexus repo or not.



Just Hit the "Build with parameters" button for CD Pipeline and verify if application is deployed to CloudHud or not.

Verify API Instance in API Manager

API Name	Version	Status	Client Applications	Creation Date
mule-cicd-demo				1 version
	v1	Active	0	03-22-2021 21:33

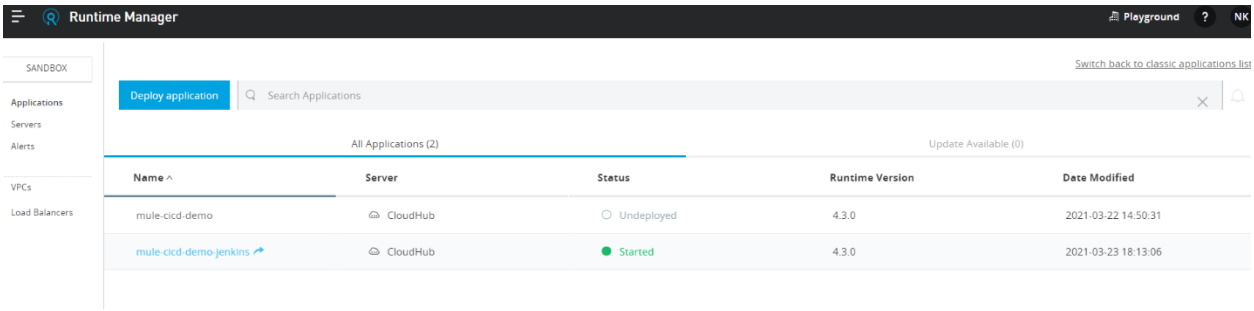


Magic of triggering API Policy setup and API Discovery dynamically via script is happening from [api.platform.properties]

Current support of these script is with JSON Threat protection , client id enforcement and JWT.

As part of this article we are using Client ID Enforcement and JSON Threat Protection.

Verify API Deployment in Runtime Manager



Happy Deploying the API to your Anypoint Platform using these CICD Pipelines along with API Discovery and Policy Setup.

Please find presentation here - [Detailed Presentation]