ASSIGNMENT 5

AIM : Understand Jenkins and execute pipeline using Jenkins.

THEORY :

What is Jenkins Pipeline?

Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

A continuous delivery (CD) pipeline is an automated expression of your process for getting software from version control right through to your users and customers. Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as progressing the built software (called a "build") through multiple stages of testing and deployment.

Why Pipeline?

Jenkins is, fundamentally, an automation engine which supports a number of automation patterns. Pipeline adds a powerful set of automation tools onto Jenkins, supporting use cases that span from simple continuous integration to comprehensive CD pipelines. By modeling a series of related tasks, users can take advantage of the many features of Pipeline:

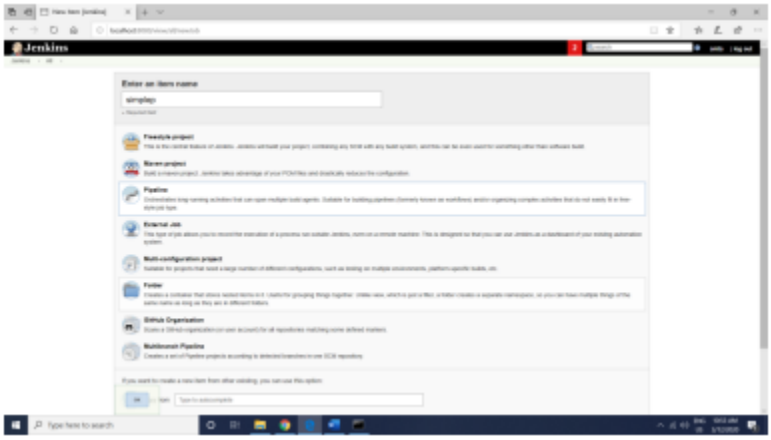
1. Code: Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.
2. Durable: Pipelines can survive both planned and unplanned restarts of the Jenkins master.
3. Pausable: Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.
4. Versatile: Pipelines support complex real-world CD requirements, including the ability to fork/join, loop, and perform work in parallel.
5. Extensible: The Pipeline plugin supports custom extensions to its DSL and multiple options for integration with other plugins.

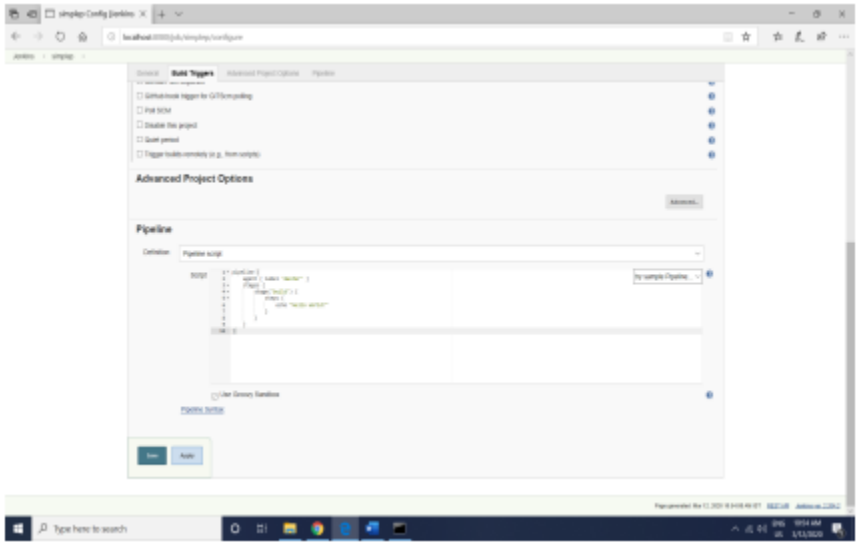
Example : Creation of Pipeline

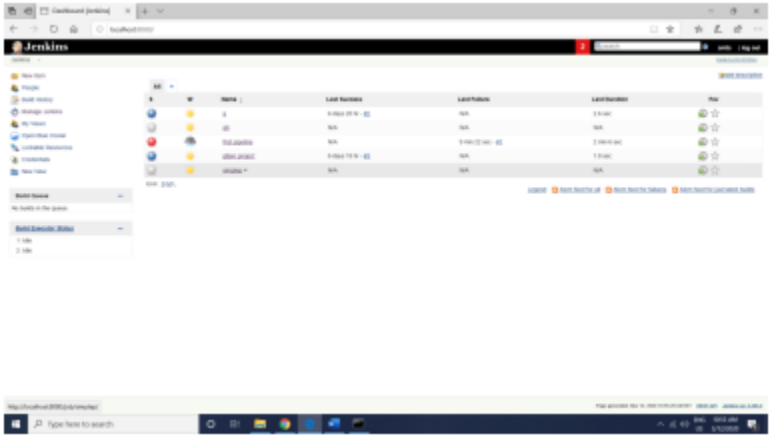
Simple Pipeline:

Select new item

Type Project name and select Pipeline.







Output :

Basic Pipeline Script :

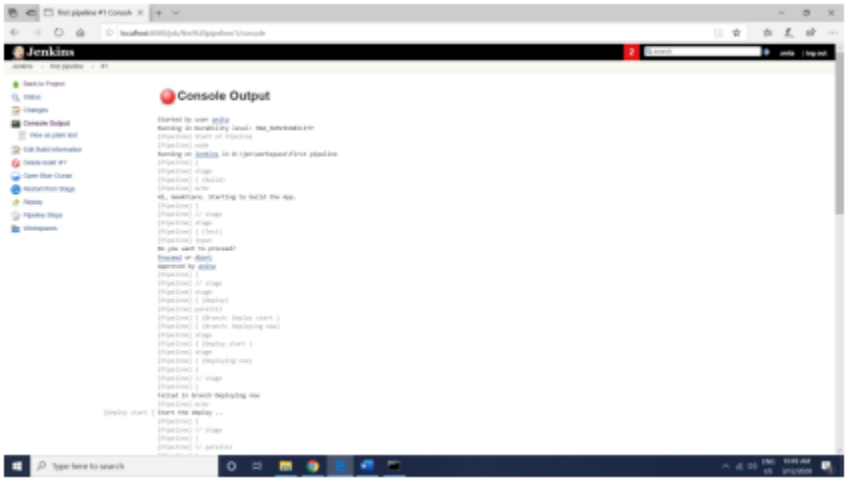
```
pipeline {
  agent any
  stages {
```

```
stage('Build') {
steps {
echo 'Hi, GeekFlare. Starting to build the App.'
}
}
stage('Test') {
steps {
  input('Do you want to proceed?')
}
}
stage('Deploy') {
parallel {
stage('Deploy start ') {
steps {
  echo "Start the deploy .."
}
}
stage('Deploying now') {
agent {
docker {
reuseNode true
image ‘nginx’
}
}
steps {
echo "Docker Created"
}
}
```

```
}
}
stage('Prod') {
steps {
echo "App is Prod Ready"
}
}
}
}
```

Click on Build now :

Console Output :



CONCLUSION : In this experiment we had created pipeline using scripting language.