|  |  |  |
| --- | --- | --- |
|  | **K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai**  *Accredited ‘A’ Grade by NAAC with 3.21 CGPA*  *3 Programs Accredited by National Board of Accreditation*  *Permanently Affiliated to University of Mumbai,*  *Best College Award by University of Mumbai (Urban Region), ISTE (MH), and CSI (Mumbai)*  *UGC Recognized Institute under Section 2(f) and 12(B) of the UGC Act, 1956* |  |

# Experiment No. 2

## Aim:

* Continuous Integration using Jenkins Pipeline

## Theory:

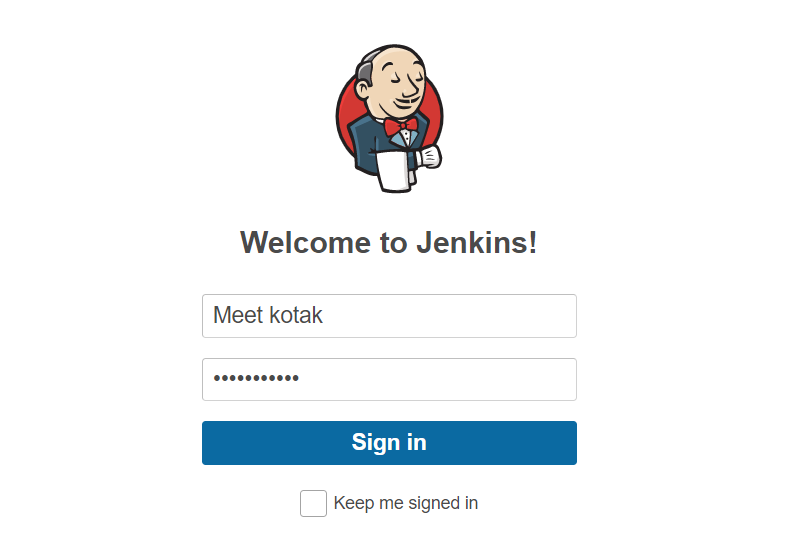
Jenkins is a free and open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. The Jenkins project was originally named Hudson, and was renamed after a dispute with Oracle, which had forked the project and claimed rights to the project name.

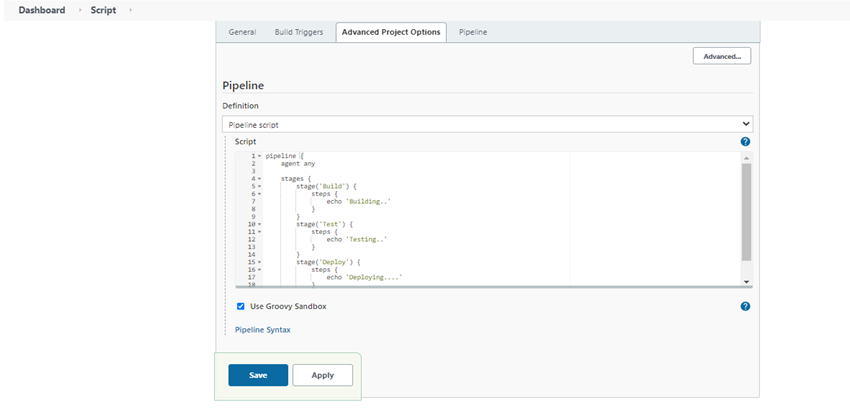
Jenkins Pipeline is a combination of jobs to deliver software continuously using Jenkins. A Jenkins pipeline consists of several states or stages, and they get executed in a sequence one after the other. JenkinsFile is a simple text file that is used to create a pipeline as code in Jenkins. It contains code in Groovy Domain Specific Language (DSL), which is simple to write and human-readable.

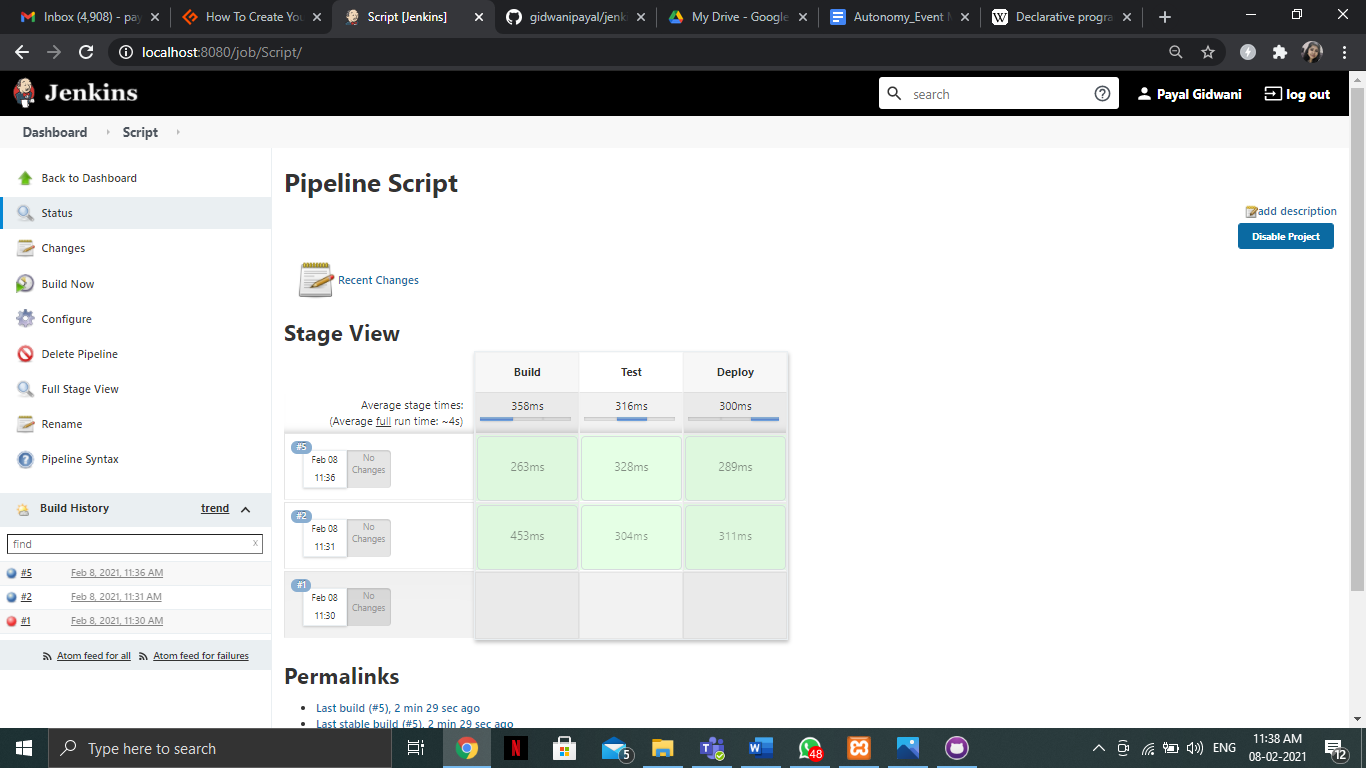
Either you can run JenkinsFile separately, or you can run the pipeline code from Jenkins Web UI also. There are two ways you can create a pipeline using Jenkins.

* Declarative – a new way of creating Jenkins Pipeline. Here you write groovy code containing “pipeline” blocks, which is checked into an SCM (Source Code Management)
* Scripted – way of writing groovy code where the code is defined inside “node” blocks.

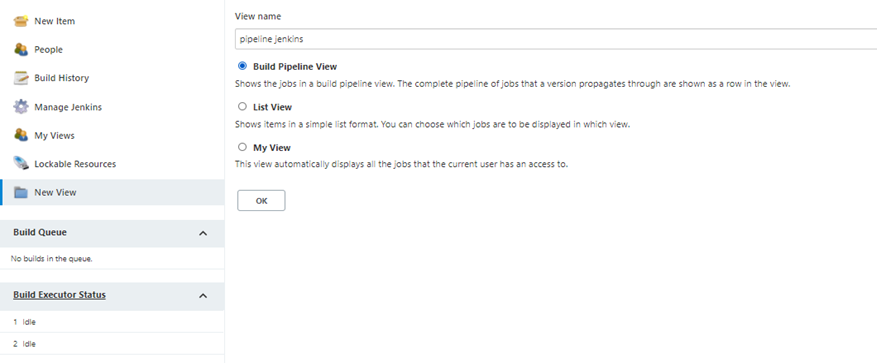
### 1. Using Pipeline Scripting:

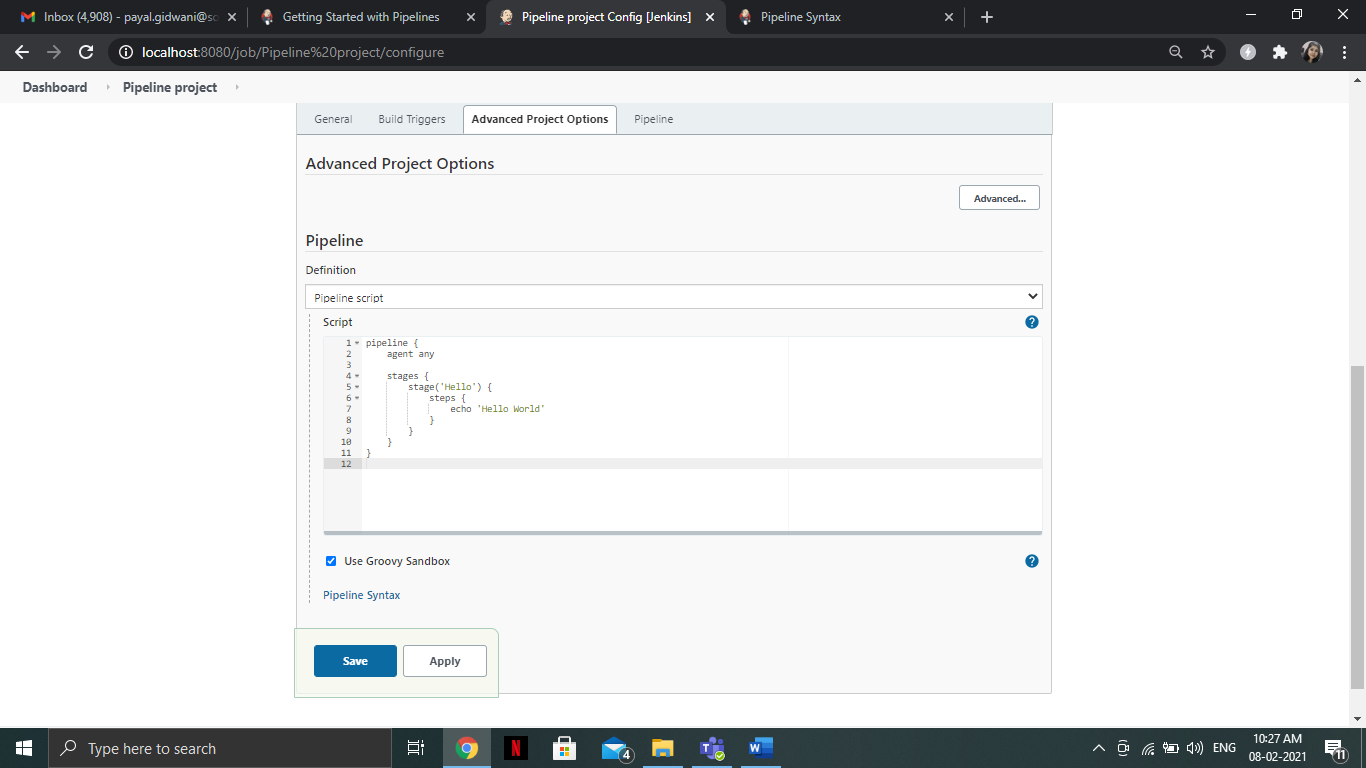






### 2. Example: Hello world







### 3. Using Git Code:

