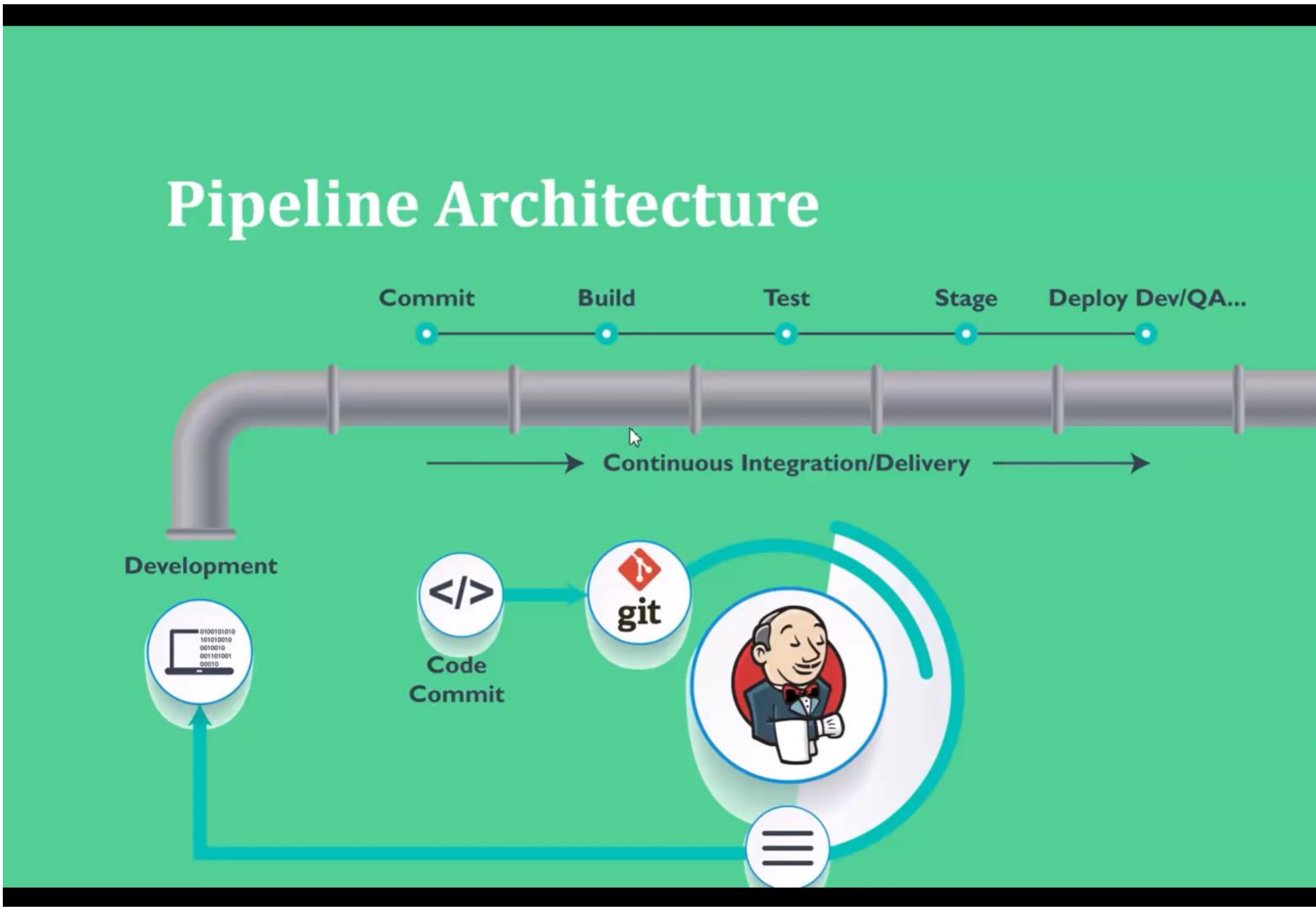
JENKINS

Jenkins is a Java-based open-source automation platform with plugins designed for continuous integration. It is used to continually create and test software projects, making it easier for developers and DevOps engineers to integrate changes to the project and for consumers to get a new build.

PIPELINE



- ➔ There is an external trigger that starts the pipeline.
- ➔ The latest code gets fetched from the source code repository.
- ➔ Then the code gets compiled, built, and packaged.
- ➔ Bunch of tests get run.`
- ➔ Then, the compiled output gets published somewhere.
- ➔ Then, if we do continuous delivery, it gets deployed to live servers.
- ➔ The final output is some sort of notification to say whether the build is succeeded or not.

All the features in the pipeline comes from the plugins. We cannot do this with plain installation of Jenkins.

In Jenkins, all the plugins come from a central online catalog, which also gets used for installation and ongoing updates, which includes both feature updates and to add new functionality to the plugins, but also security fixes if there are any vulnerabilities in the plugins.

If we want to use some functionality available already in another plugin, we can just import the dependency on that plugin.

Due to this, we ended up with a huge dependency graph where we only installed 5-6 plugins but ended up with 50-60 that are deployed or with these interrelated dependencies.

INSTALLATION

Basic steps –

1. Install Java Development Kit (JDK) (minimum jdk version 11 required) -

x64 Installer	159.94 MB	https://www.oracle.com/java/technologies/downloa

2. Set the Path for the Environmental Variable for JDK

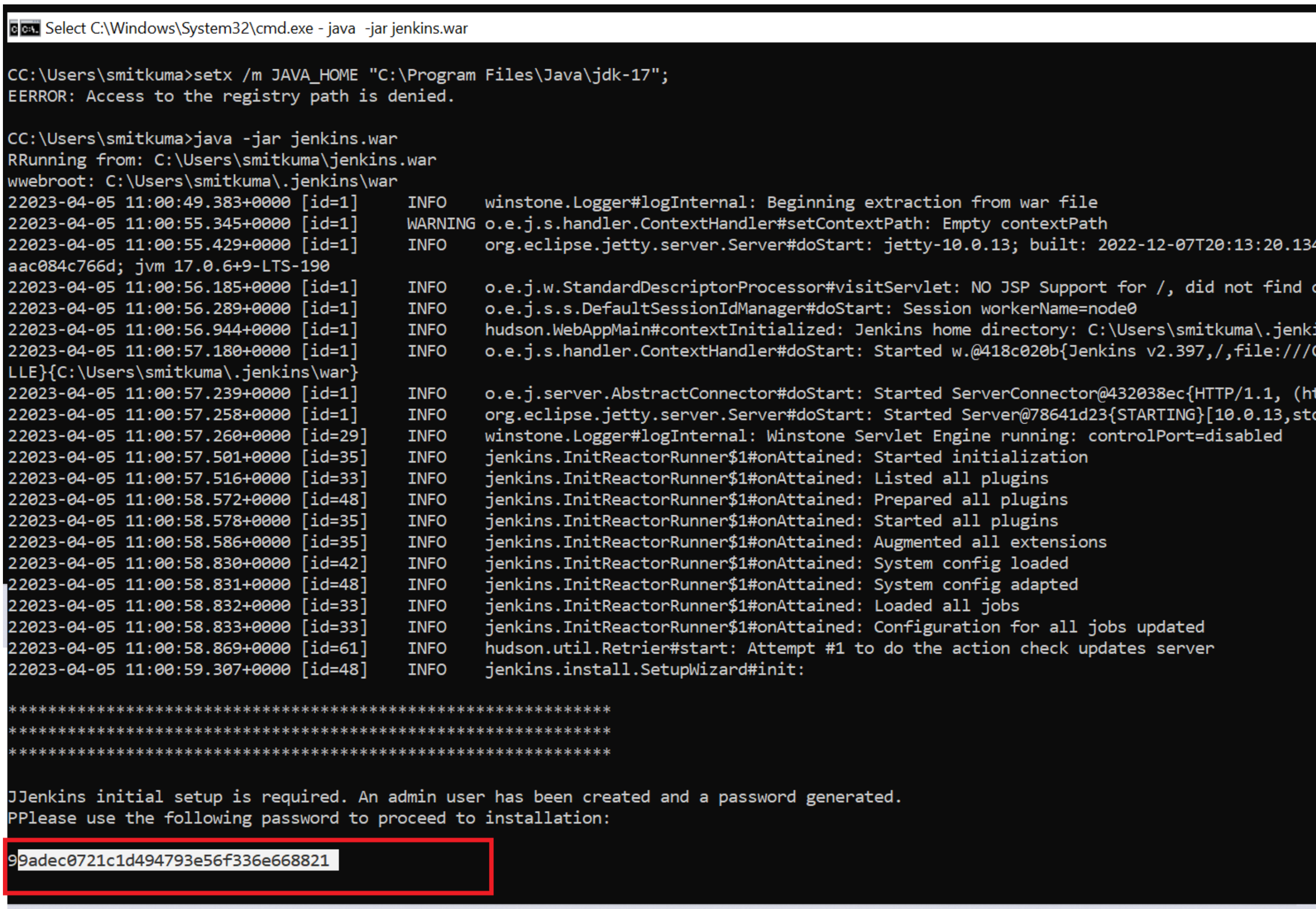
Download and run Jenkins

3. Download Jenkins Generic Java package (.war)
4. Open up a terminal in the download directory
5. Run `java -jar jenkins.war --httpPort=8080`
6. Browse to `http://localhost:8080`
7. Follow the instructions to complete the installation

When the installation is complete, you can start putting Jenkins to work!

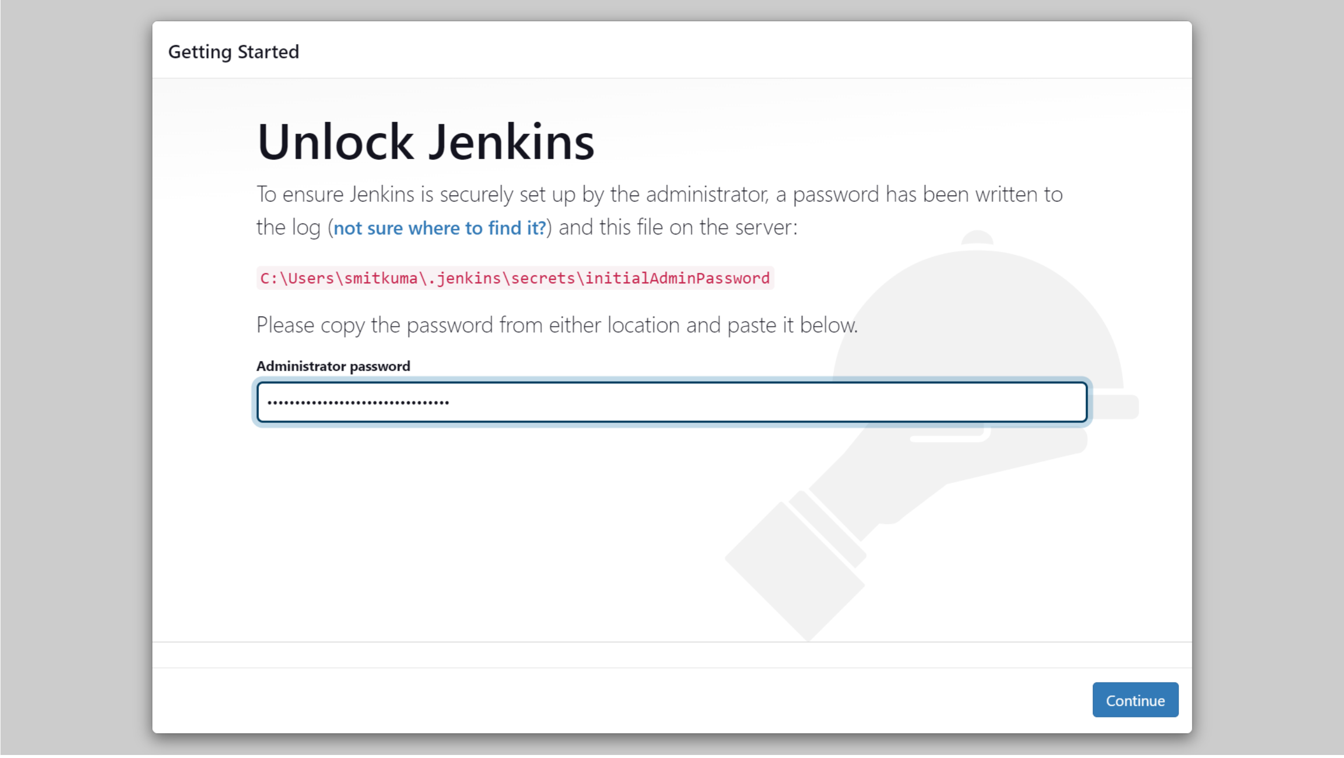
Continue to "Create your first Pipeline"

1. Initialize the jenkins.war file with the command “java -jar jenkins.war” and install it.



```
Select C:\Windows\System32\cmd.exe - java  -jar jenkins.war

CC:\Users\smitkuma>setx /m JAVA_HOME "C:\Program Files\Java\jdk-17";
EERROR: Access to the registry path is denied.

CC:\Users\smitkuma>java -jar jenkins.war
RRunning from: C:\Users\smitkuma\jenkins.war
wwebroot: C:\Users\smitkuma\.jenkins\war
22023-04-05 11:00:49.383+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
22023-04-05 11:00:55.345+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
22023-04-05 11:00:55.429+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-10.0.13; built: 2022-12-07T20:13:20.134
aac084c766d; jvm 17.0.6+9-LTS-190
22023-04-05 11:00:56.185+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find
22023-04-05 11:00:56.289+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
22023-04-05 11:00:56.944+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\smitkuma\.jenki
22023-04-05 11:00:57.180+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@418c020b{Jenkins v2.397,/,file:///
LLE}{C:\Users\smitkuma\.jenkins\war}
22023-04-05 11:00:57.239+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@432038ec{HTTP/1.1, (ht
22023-04-05 11:00:57.258+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started Server@78641d23{STARTING}[10.0.13,st
22023-04-05 11:00:57.260+0000 [id=29]    INFO    winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
22023-04-05 11:00:57.501+0000 [id=35]    INFO    jenkins.InitReactorRunner$1#onAttained: Started initialization
22023-04-05 11:00:57.516+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
22023-04-05 11:00:58.572+0000 [id=48]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
22023-04-05 11:00:58.578+0000 [id=35]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
22023-04-05 11:00:58.586+0000 [id=35]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
22023-04-05 11:00:58.830+0000 [id=42]    INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
22023-04-05 11:00:58.831+0000 [id=48]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
22023-04-05 11:00:58.832+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
22023-04-05 11:00:58.833+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
22023-04-05 11:00:58.869+0000 [id=61]    INFO    hudson.util.Retrier#start: Attempt #1 to do the action check updates server
22023-04-05 11:00:59.307+0000 [id=48]    INFO    jenkins.install.SetupWizard#init:

***************************************************************
***************************************************************
***************************************************************

JJenkins initial setup is required. An admin user has been created and a password generated.
PPlease use the following password to proceed to installation:

99adec0721c1d494793e56f336e668821
```

2. Make sure you copy paste the admin password which has been generated- 99adec0721c1d494793e56f336e668821(This may also be found at: C:\Users\smitkuma\.jenkins\secrets\initialAdminPassword)
3. To use Jenkins run https://localhost:8080 (the default port) and enter the key obtained after installing Jenkins as shown above to unlock Jenkins. Save this key this is our admin password.



Getting Started

Unlock Jenkins

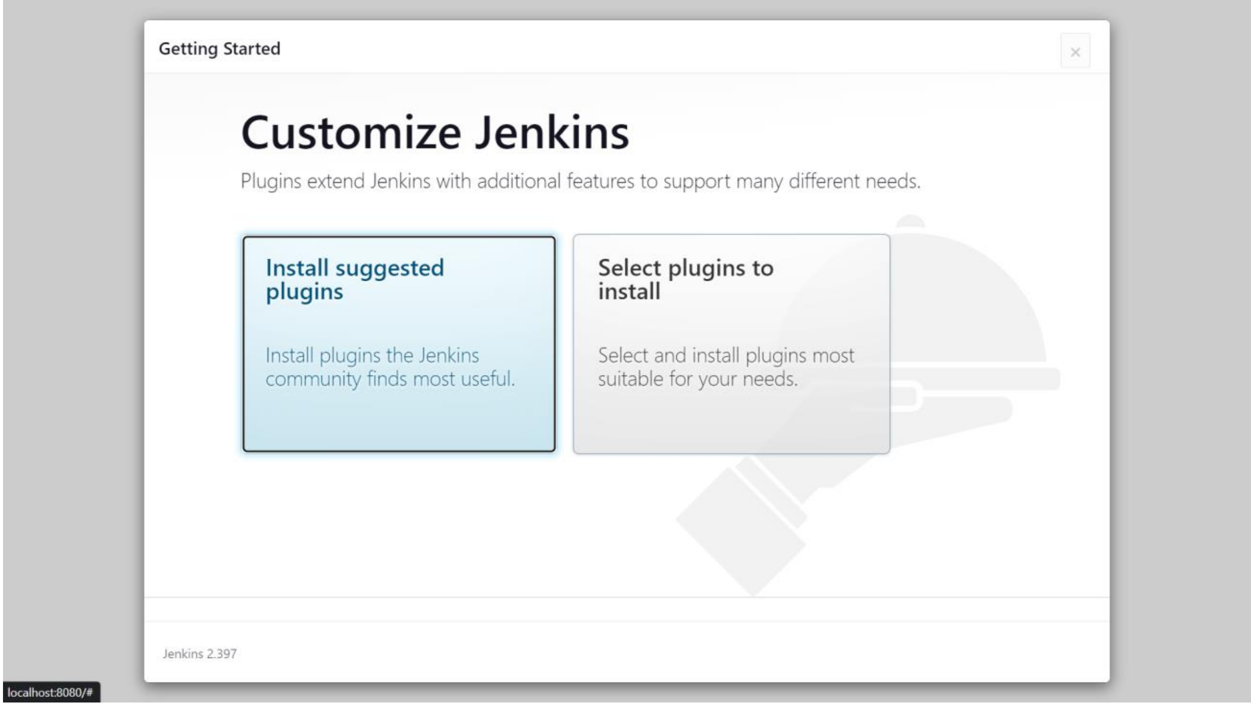
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\Users\smitkuma\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password: ································

4. Select the required plugins and initiate your jenkins profile



Getting Started

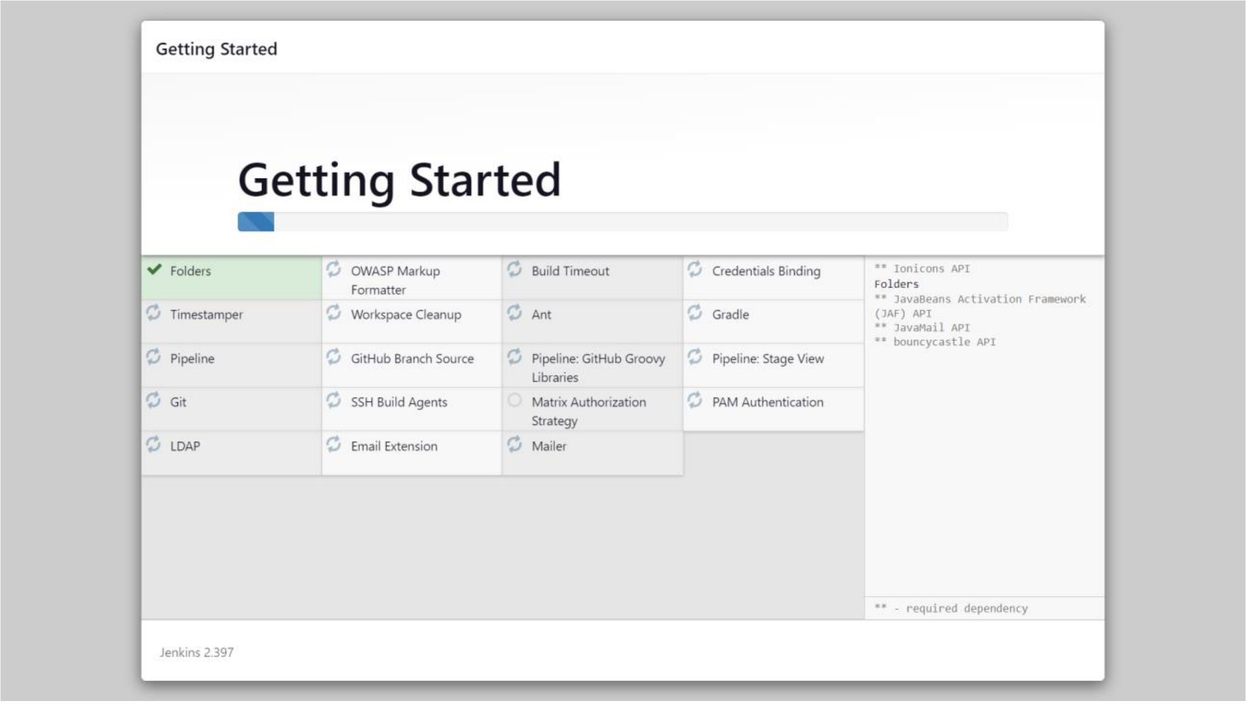
Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.

Select plugins to install
Select and install plugins most suitable for your needs.

5. Now it will take few minutes to install the plugins.



Getting Started

Getting Started

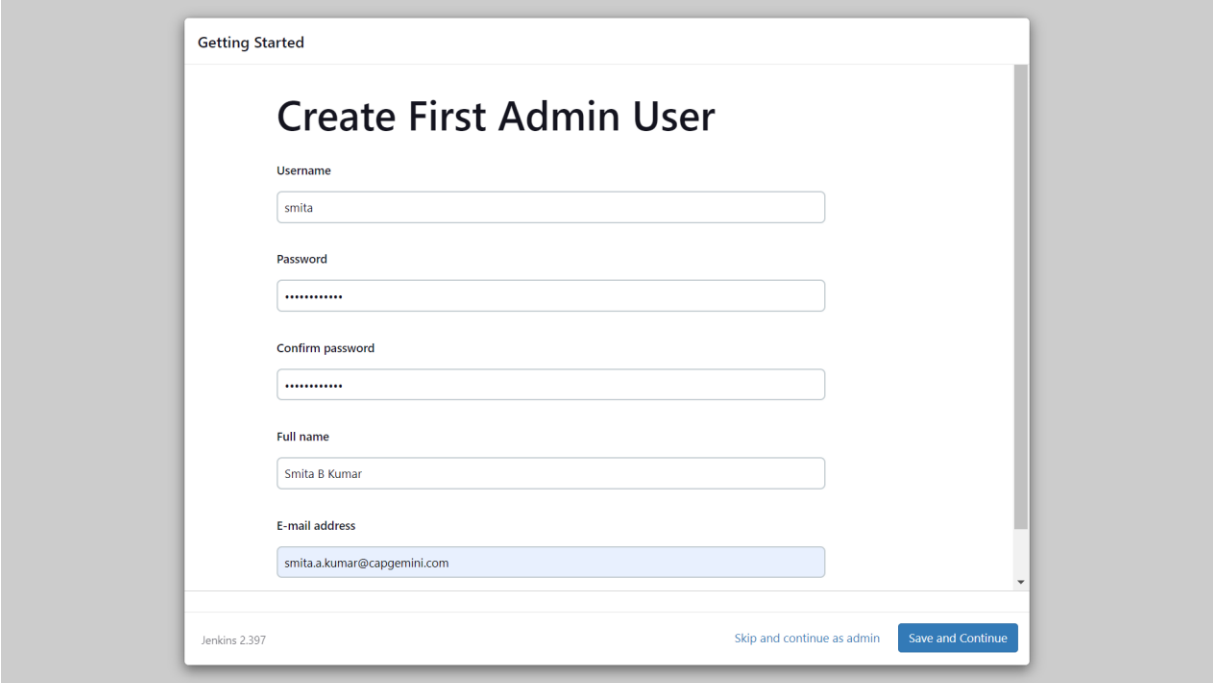
✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	○ Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** Ionicons API
Folders
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
** bouncycastle API
```

```
** - required dependency
```

Jenkins 2.397

6. Now we can create a personal profile or login as admin – give a simple password which you can remember ex: Password@123



Getting Started

Create First Admin User

Username: smita

Password: ••••••••••••

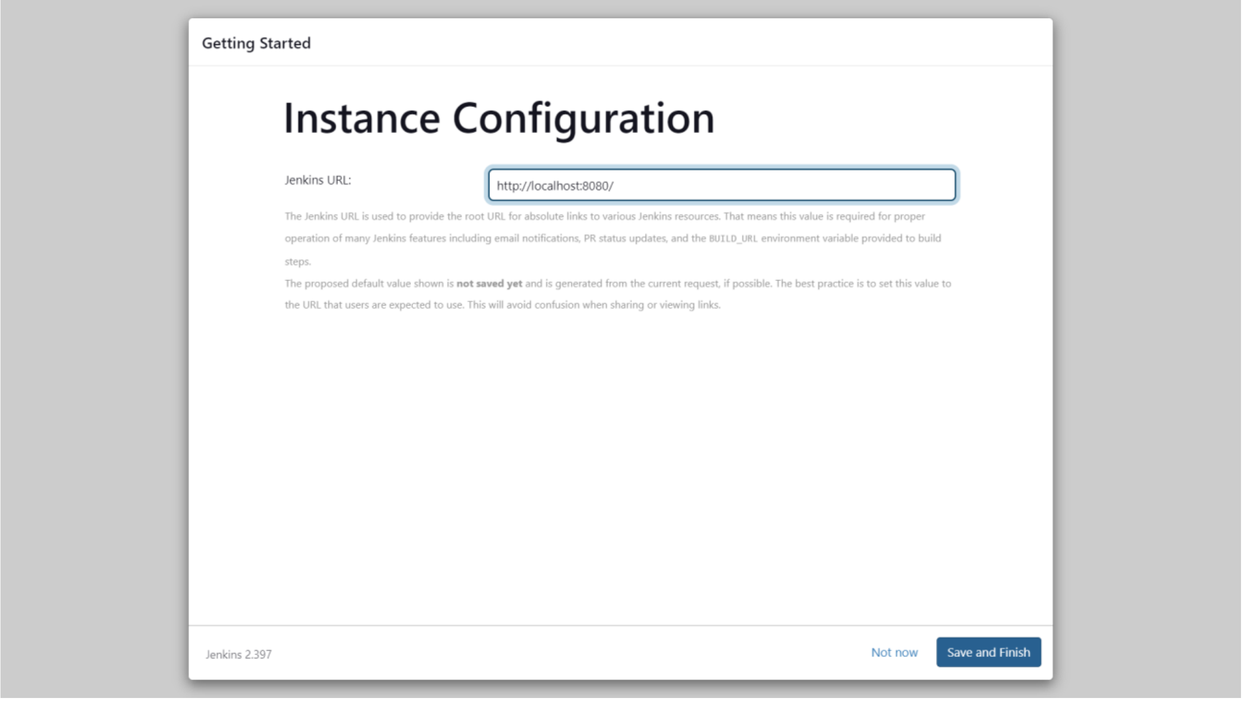
Confirm password: ••••••••••••

Full name: Smita B Kumar

E-mail address: smita.a.kumar@capgemini.com

Jenkins 2.397

7. We can change the instance if needed at this step



Getting Started

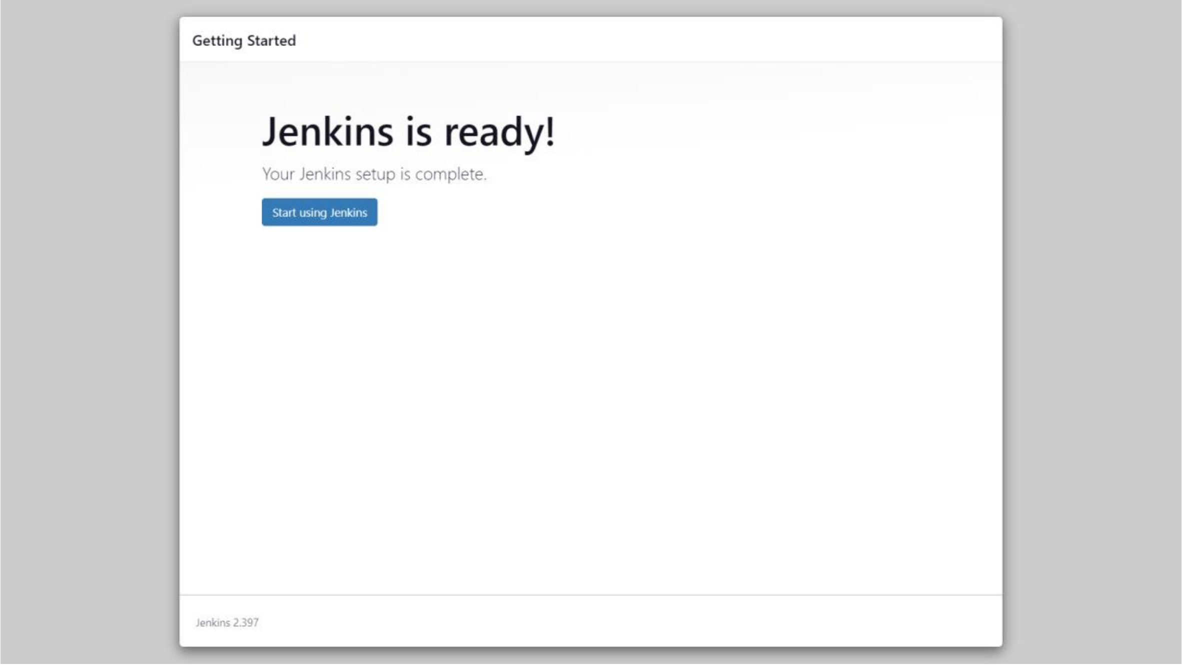
Instance Configuration

Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

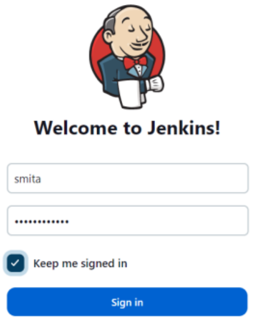
8. Jenkins is installed successfully.



Getting Started

Jenkins is ready!

Your Jenkins setup is complete.



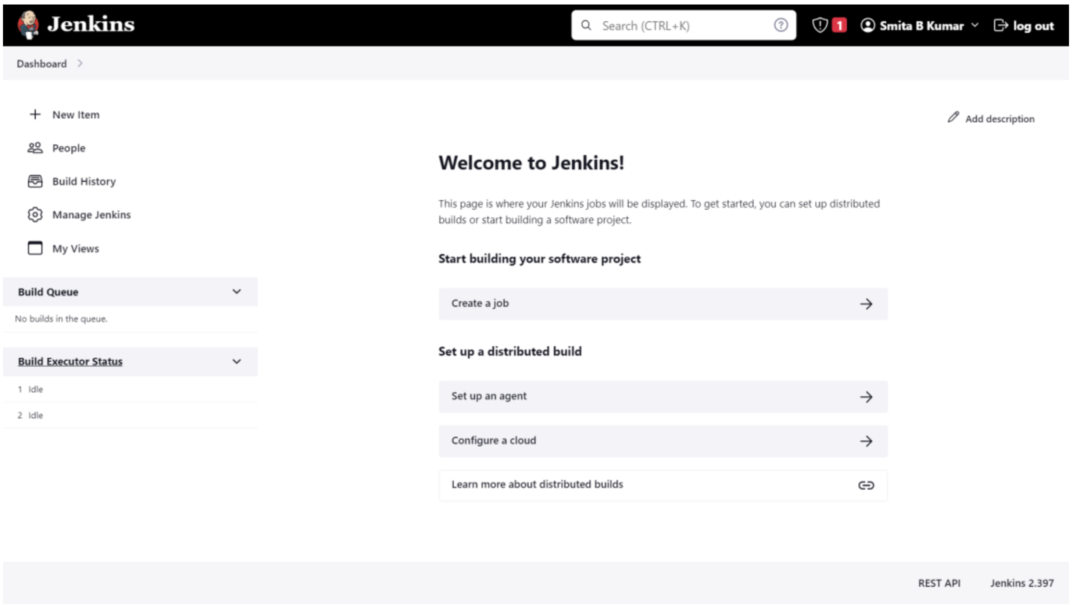
Welcome to Jenkins!

smita

••••••••••••

☑ Keep me signed in

9. Now we can create a job my following these steps and cloning any project.



No builds in the queue.

1 Idle

2 Idle

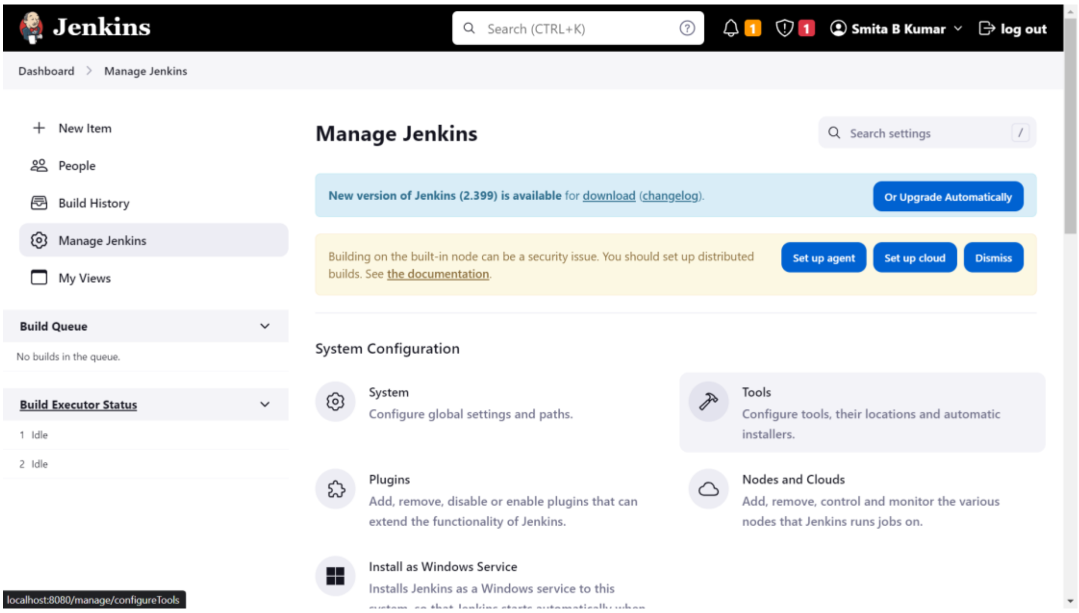
Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

10. Now Click on **Manage Jenkins**-> **Tool** -> Script the **Java Home Path** & **Maven Home Path** and repository URL and the project will be ready to build.



No builds in the queue.

1 Idle

2 Idle

localhost:8080/manage/configureTools

Manage Jenkins

New version of Jenkins (2.399) is available for download (changelog).

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

System Configuration

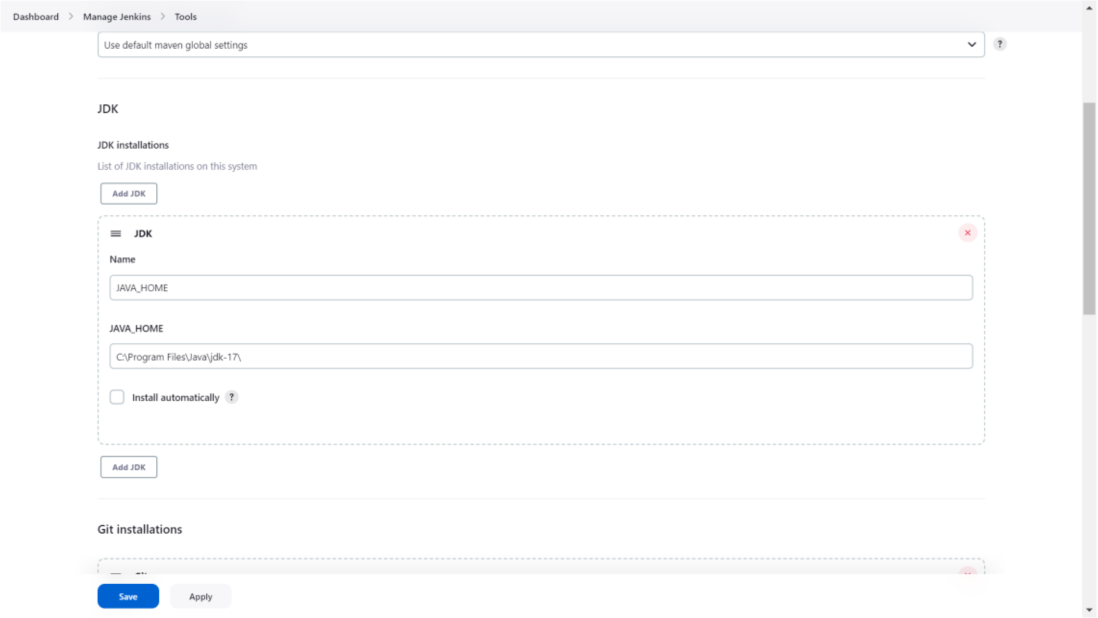
System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Install as Windows Service
Installs Jenkins as a Windows service to this



Use default maven global settings

JDK

JDK installations

List of JDK installations on this system

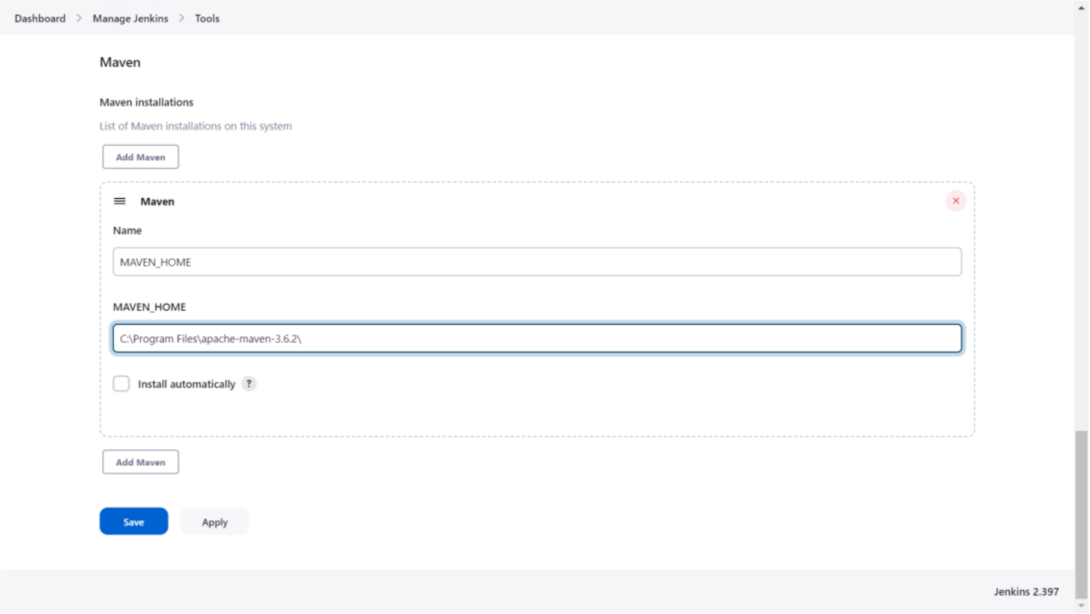
JDK

Name: JAVA_HOME

JAVA_HOME: C:\Program Files\Java\jdk-17\

□ Install automatically

Git installations



Maven

Maven installations

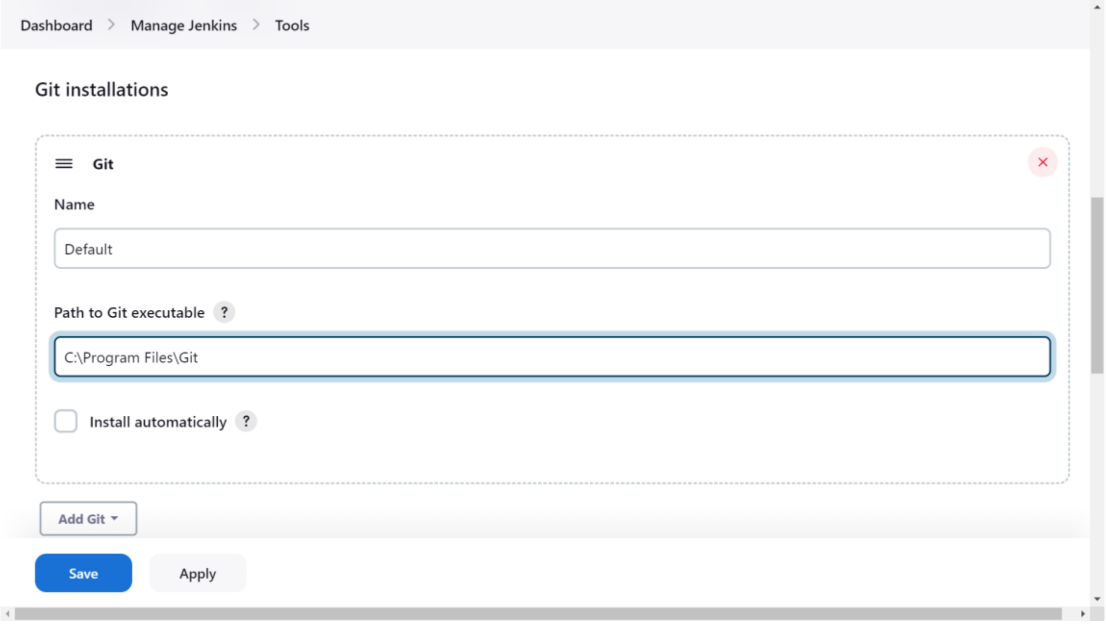
List of Maven installations on this system

Maven

Name: MAVEN_HOME

MAVEN_HOME: C:\Program Files\apache-maven-3.6.2\

□ Install automatically



Git installations

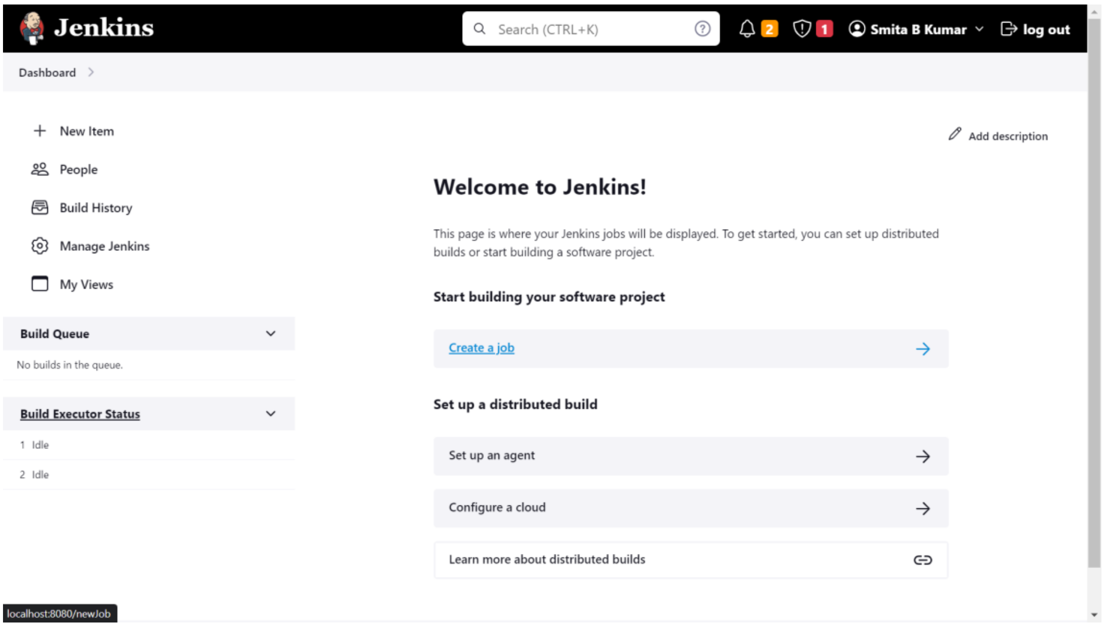
Git

Name: Default

Path to Git executable: C:\Program Files\Git

□ Install automatically

Create a new job



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Build Queue

No builds in the queue.

Build Executor Status

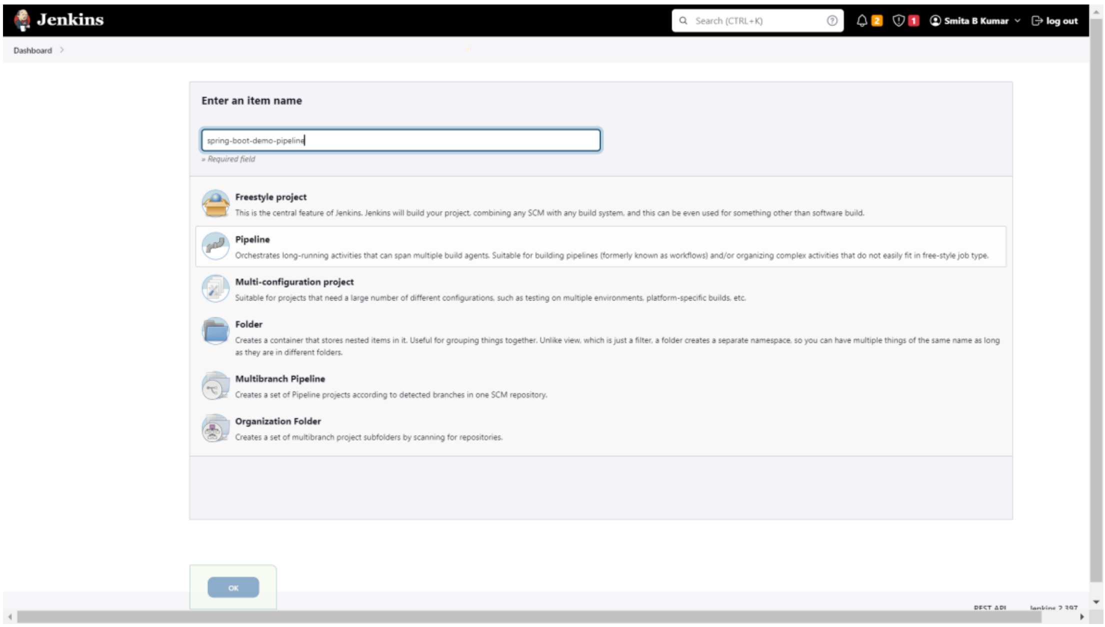
1 Idle

2 Idle

localhost:8080/newJob

Enter the name - **spring-boot-demo-pipeline**

Select **Pipeline**



Enter an item name

spring-boot-demo-pipeline

» *Required field*

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

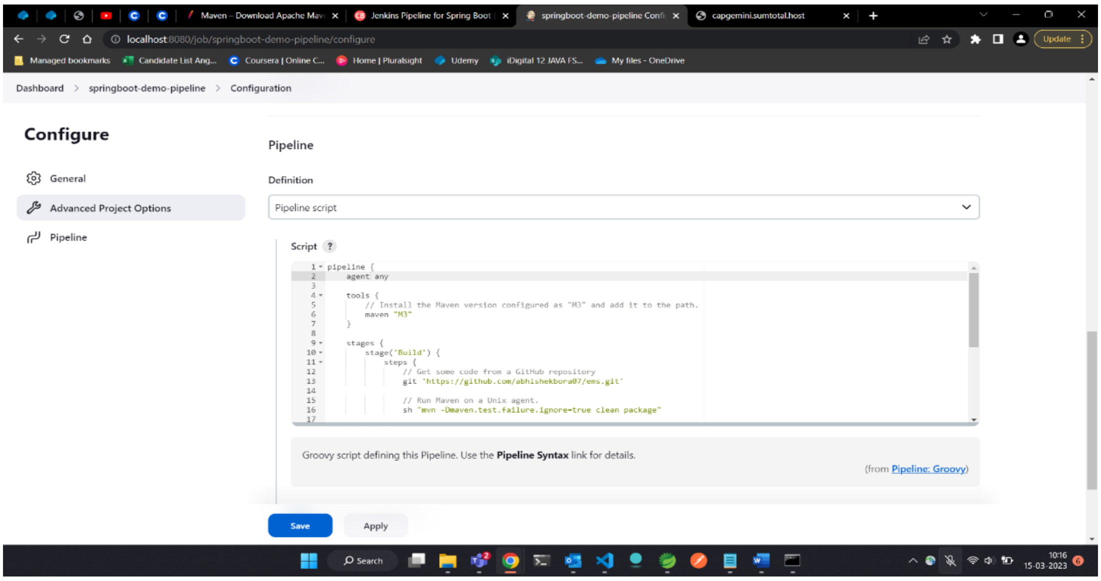
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.



localhost:8080/job/springboot-demo-pipeline/configure
Dashboard > springboot-demo-pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline

Pipeline

Definition

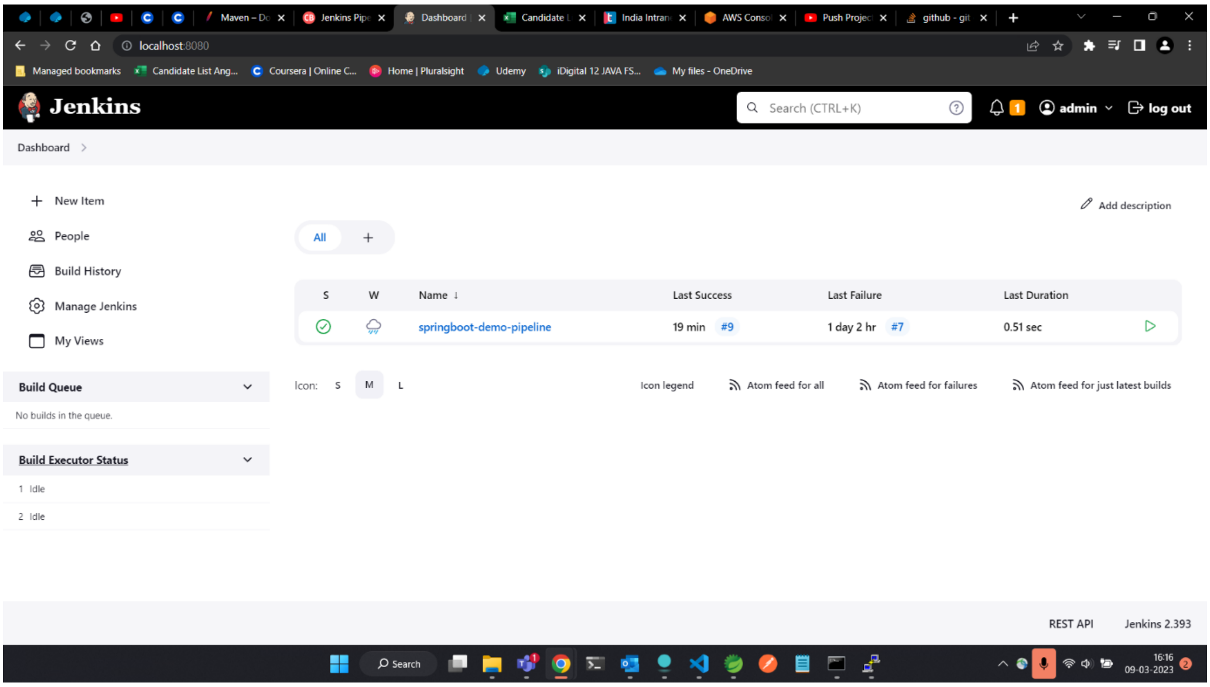
Pipeline script

Script ?

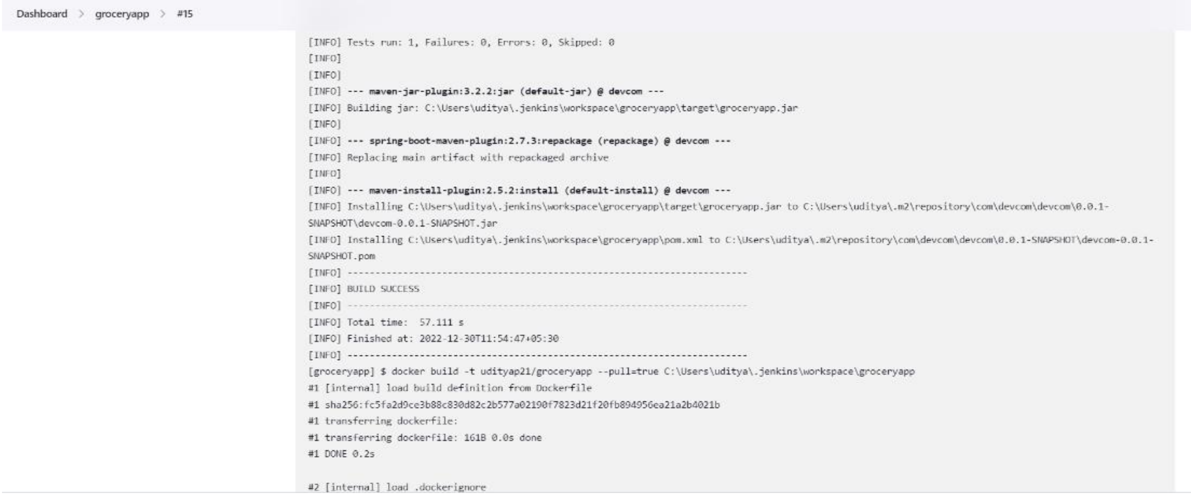
```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/abhishekbora07/ems.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"
```

Groovy script defining this Pipeline. Use the **Pipeline Syntax** link for details.

(from Pipeline: Groovy)



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		springboot-demo-pipeline	19 min #9	1 day 2 hr #7	0.51 sec

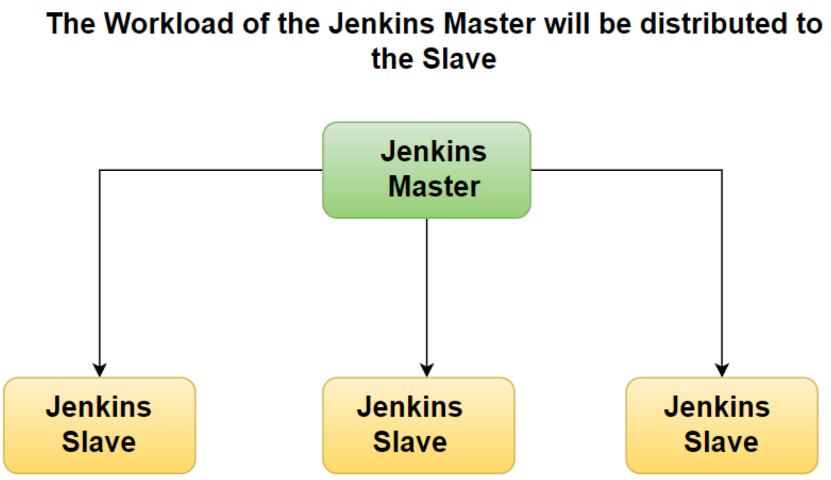


```
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ devcom ---
[INFO] Building jar: C:\Users\uditya\.jenkins\workspace\groceryapp\target\groceryapp.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.3:repackage (repackage) @ devcom ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ devcom ---
[INFO] Installing C:\Users\uditya\.jenkins\workspace\groceryapp\target\groceryapp.jar to C:\Users\uditya\.m2\repository\com\devcom\devcom\0.0.1-
SNAPSHOT\devcom-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\uditya\.jenkins\workspace\groceryapp\pom.xml to C:\Users\uditya\.m2\repository\com\devcom\devcom\0.0.1-SNAPSHOT\devcom-0.0.1-
SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  57.111 s
[INFO] Finished at: 2022-12-30T11:54:47+05:30
[INFO] ------------------------------------------------------------------------
[groceryapp] $ docker build -t udityap21/groceryapp --pull=true C:\Users\uditya\.jenkins\workspace\groceryapp
#1 [internal] load build definition from Dockerfile
#1 sha256:fc5fa2d9ce3b88c830d82c2b577a02190f7823d21f20fb894956ea21a2b4021b
#1 transferring dockerfile:
#1 transferring dockerfile: 161B 0.0s done
#1 DONE 0.2s

#2 [internal] load .dockerignore
```

ARCHITECTURE

- ➔ Developers commit changes to the source code, found in the repository.
- ➔ The Jenkins CI server checks the repository at regular intervals and pulls any newly available code.
- ➔ The Build Server builds the code into an executable file. In case the build fails, feedback is sent to the developers.
- ➔ Jenkins deploys the build application to the test server. If the test fails, the developers are alerted.
- ➔ If the code is error-free, the tested application is deployed on the production server.

The files can contain different code and be very large, requiring multiple builds. However, a single Jenkins server cannot handle multiple files and builds simultaneously; for that, a distributed Jenkins architecture is necessary.


The Workload of the Jenkins Master will be distributed to the Slave
Jenkins Master
Jenkins Slave
Jenkins Slave
Jenkins Slave

- ➔ Jenkins Master

  The Jenkins master is in charge of scheduling the jobs, assigning slaves, and sending builds to slaves to execute the jobs. It'll also keep track of the slave state (offline or online) and retrieve the build result responses from slaves and display them on the console output.

- ➔ Jenkins Slave

  It runs on the remote server. The Jenkins server follows the requests of the Jenkins master and is compatible with all operating systems. Building jobs dispatched by the master are executed by the slave. Also, the project can be configured to choose a specific slave machine.