

## System Provisioning and Configuration management

What we will do:

1. Create a small MERN Stack application, i.e. a web application.
2. Push this application on a specific folder of given GitHub repository.
3. Create a job in Jenkins and make a build of this application.
4. Using Jenkins and Docker Plugin, we will create an Image of developed web application.
5. Finally, we will use a Terraform script to deploy the image on AWS Docker platform.

### Creating MERN Stack Application

Here is the URL for the complete source code.

<https://github.com/7Aishwarya/Simple-Web-Application1.git>

To directly use the application, clone it to your local repo and follow the below steps:

```
# git clone "https://github.com/7Aishwarya/Simple-Web-Application1.git"
```

Go to the directory and type the following commands in terminal.

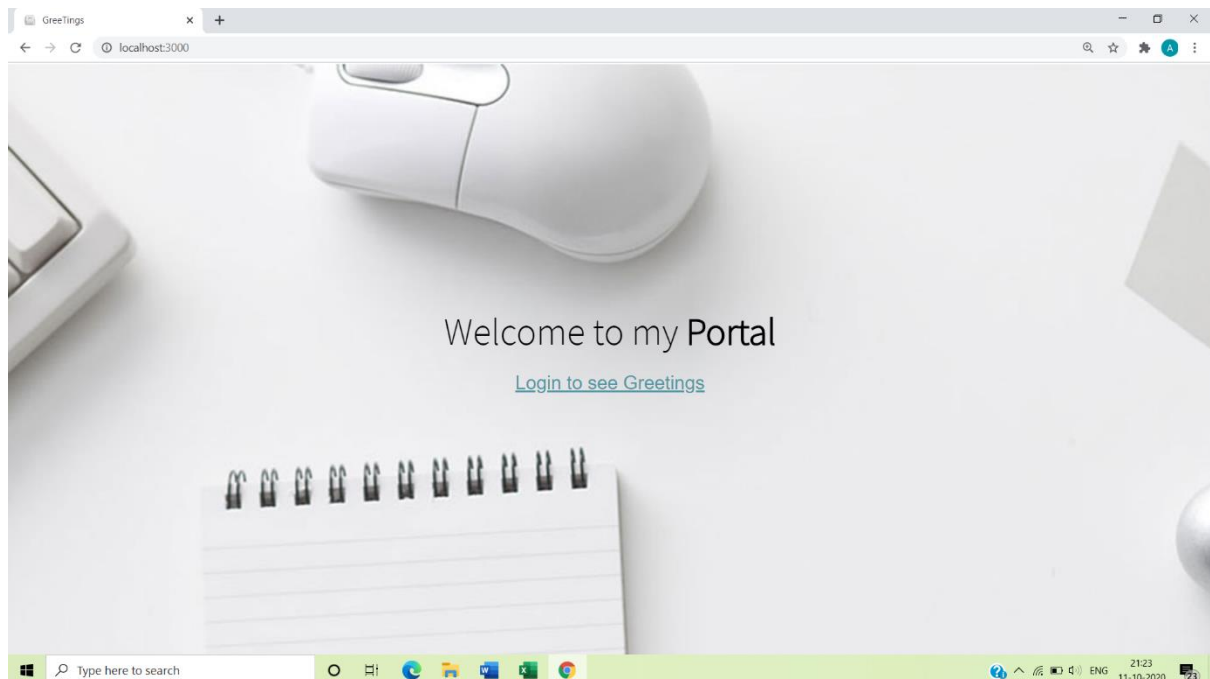
```
# npm install
```

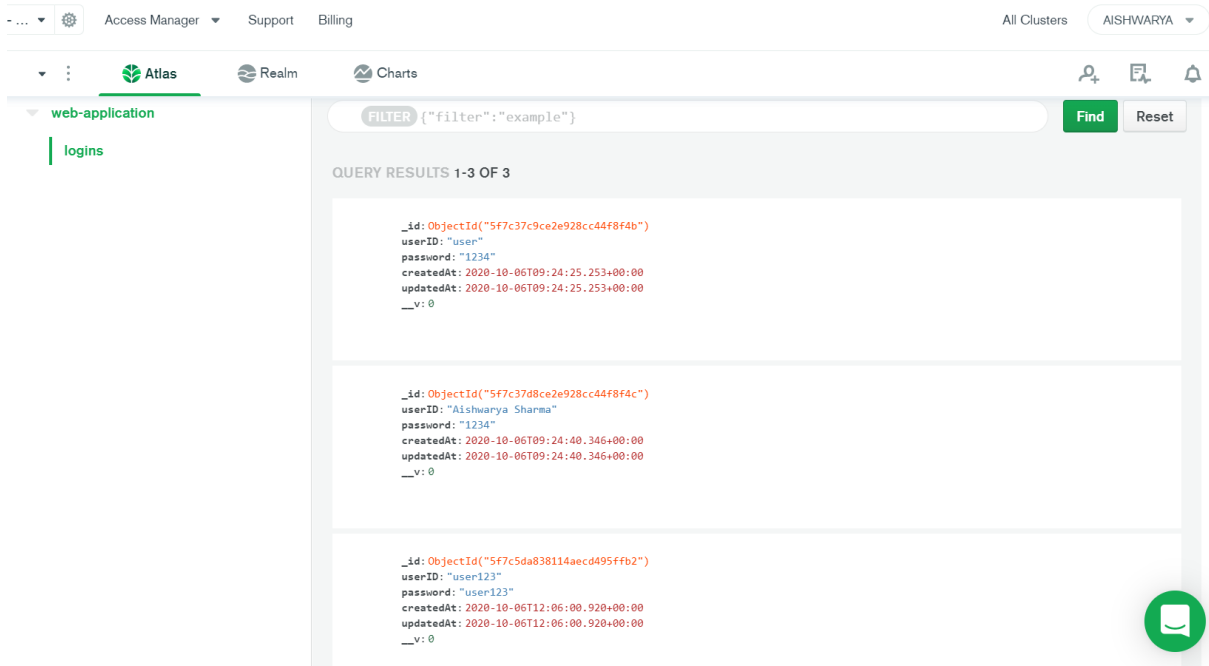
```
# npm start
```

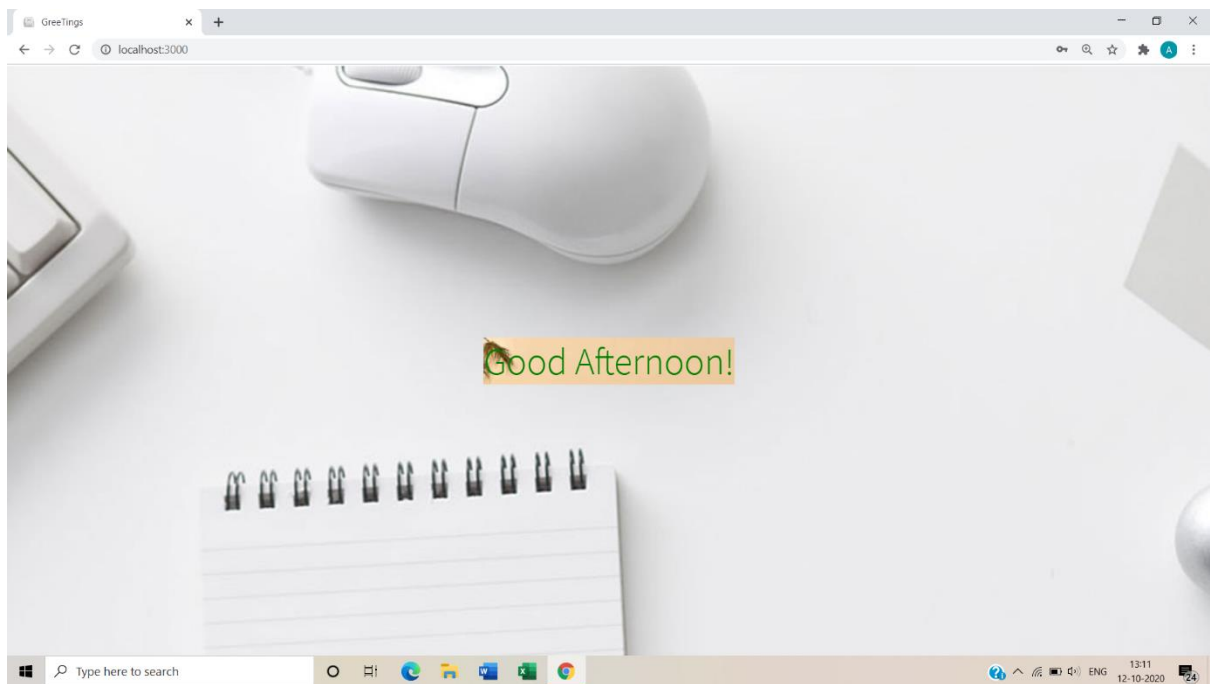
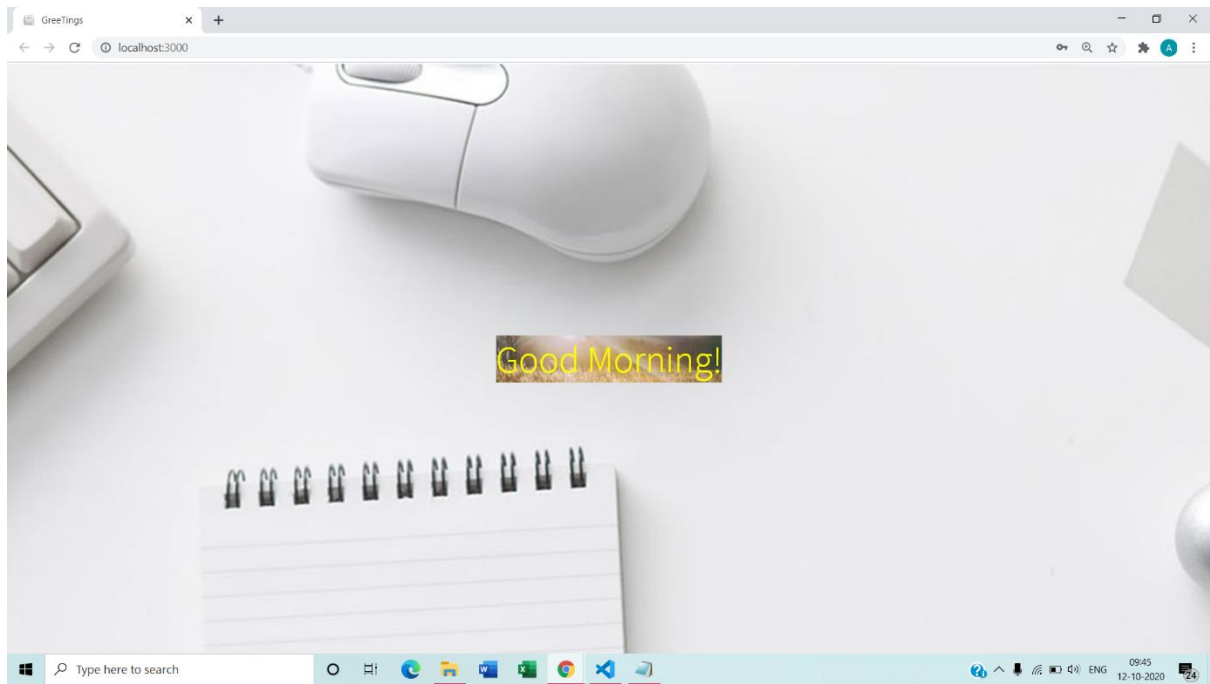
Open new terminal, and then

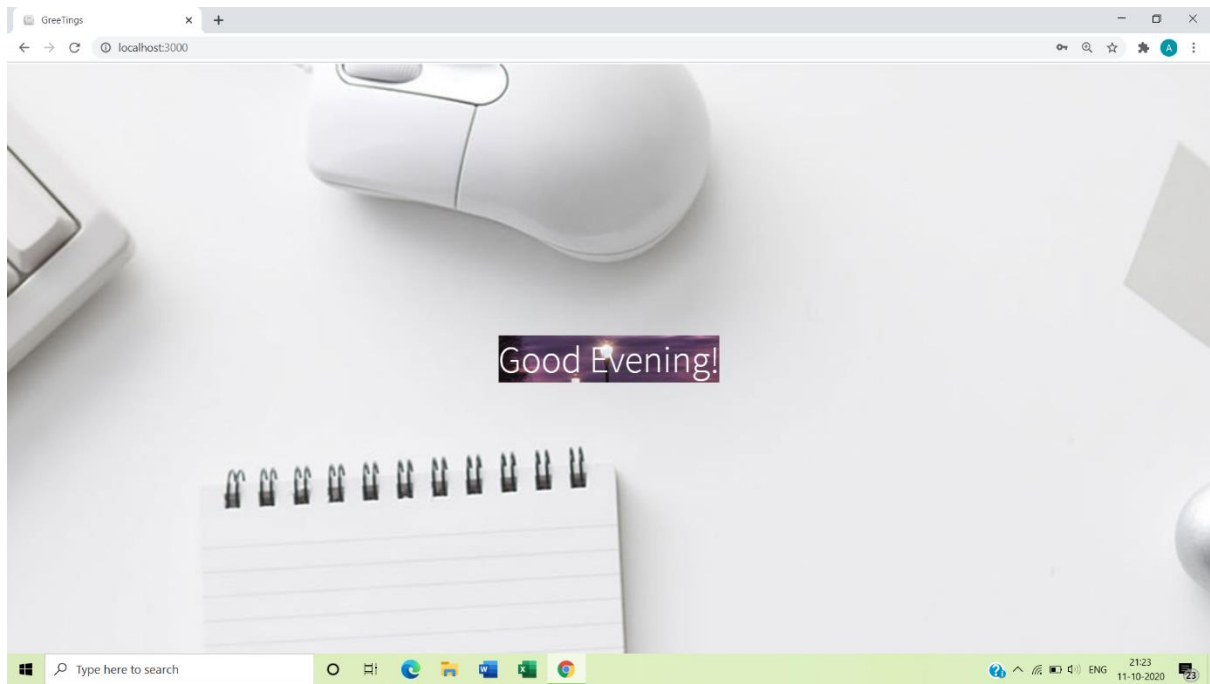
```
# cd backend
```

```
# nodemon server
```









## Setting up Jenkins

### Steps:

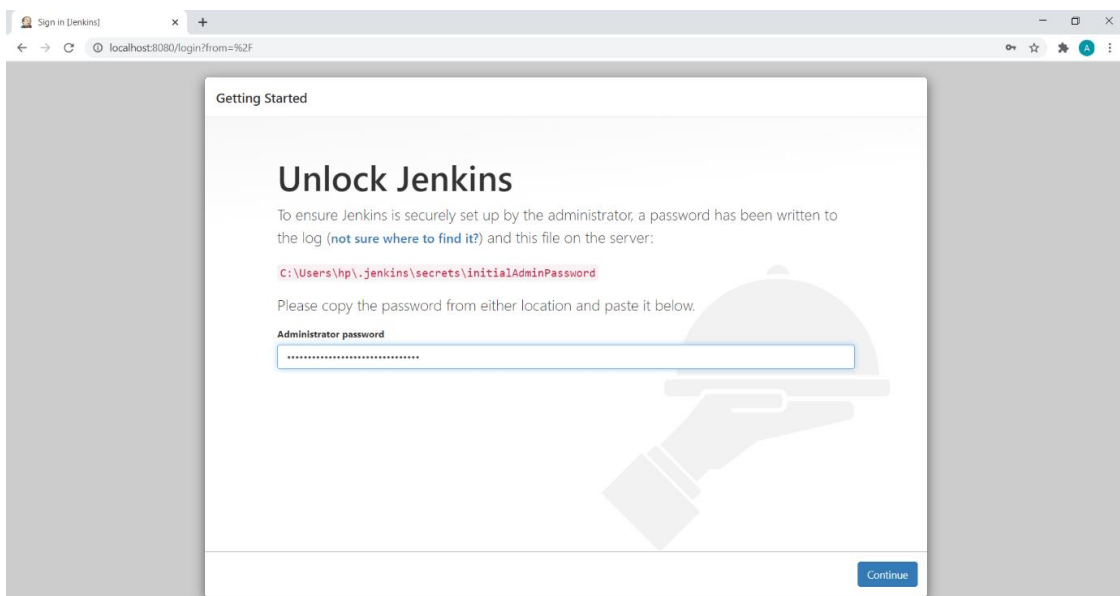
- Ensure that Java JDK and SRE are installed and the path is set in environment variables.
- Download jenkins.war file.
- Go to the directory where you have downloaded the file and open command prompt.
- Type the following command.  
# java -jar jenkins.war

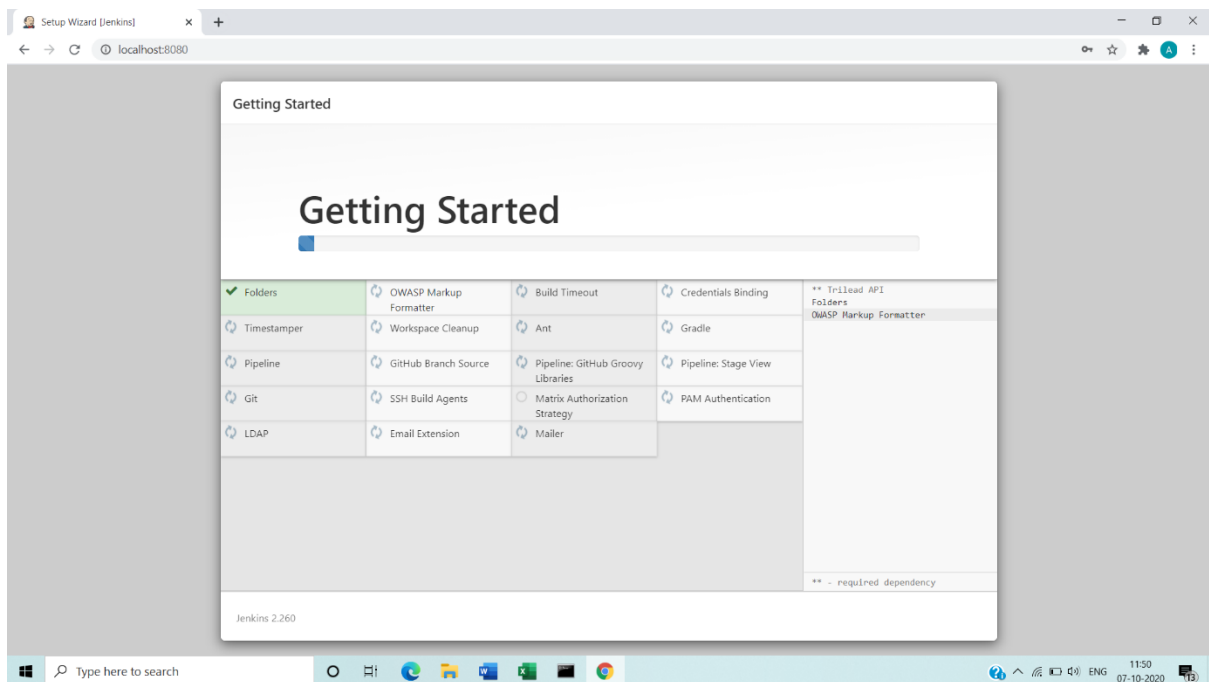
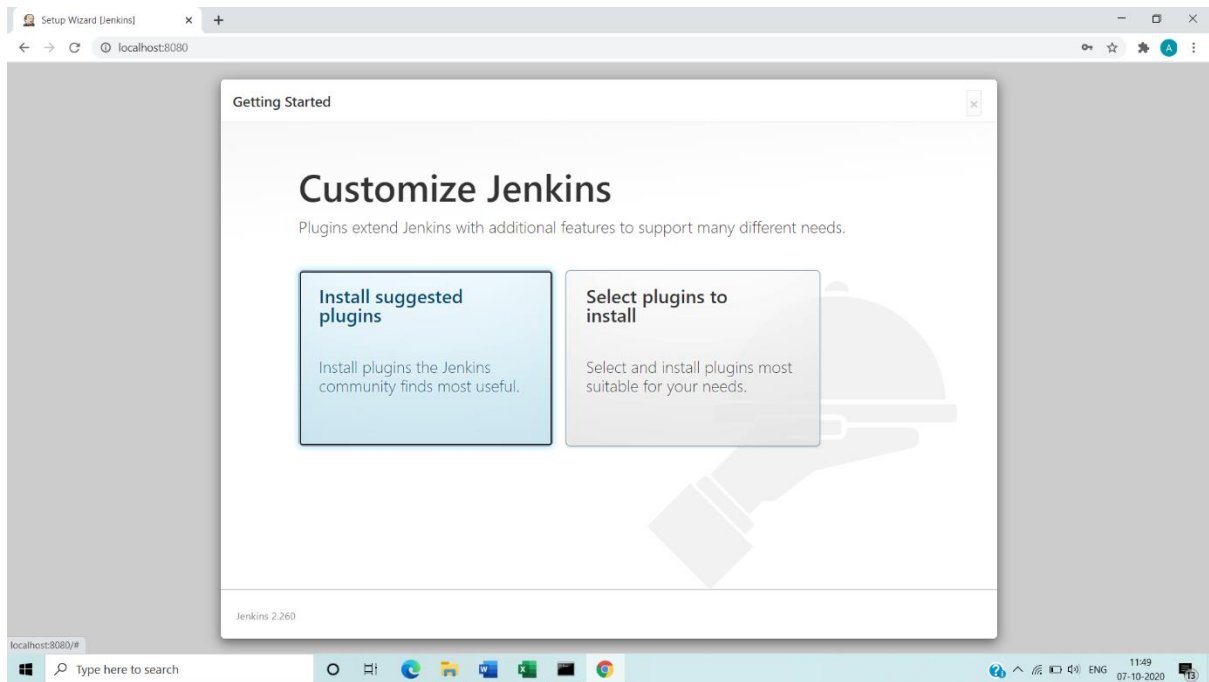
```
Select C:\Windows\System32\cmd.exe - java -jar jenkins.war --enable-future-java
D:\Jenkins>java -jar jenkins.war
Oct 07, 2020 11:38:28 AM Main verifyJavaVersion
SEVERE: Running with Java class version 57 which is not in the list of supported versions: [52, 55]. Run with the --enable-future-java flag to enable such behavior. See https://jenkins.io/redirect/java-support/
java.lang.UnsupportedClassVersionError: 57.0
    at Main.verifyJavaVersion(Main.java:174)
    at Main.main(Main.java:142)

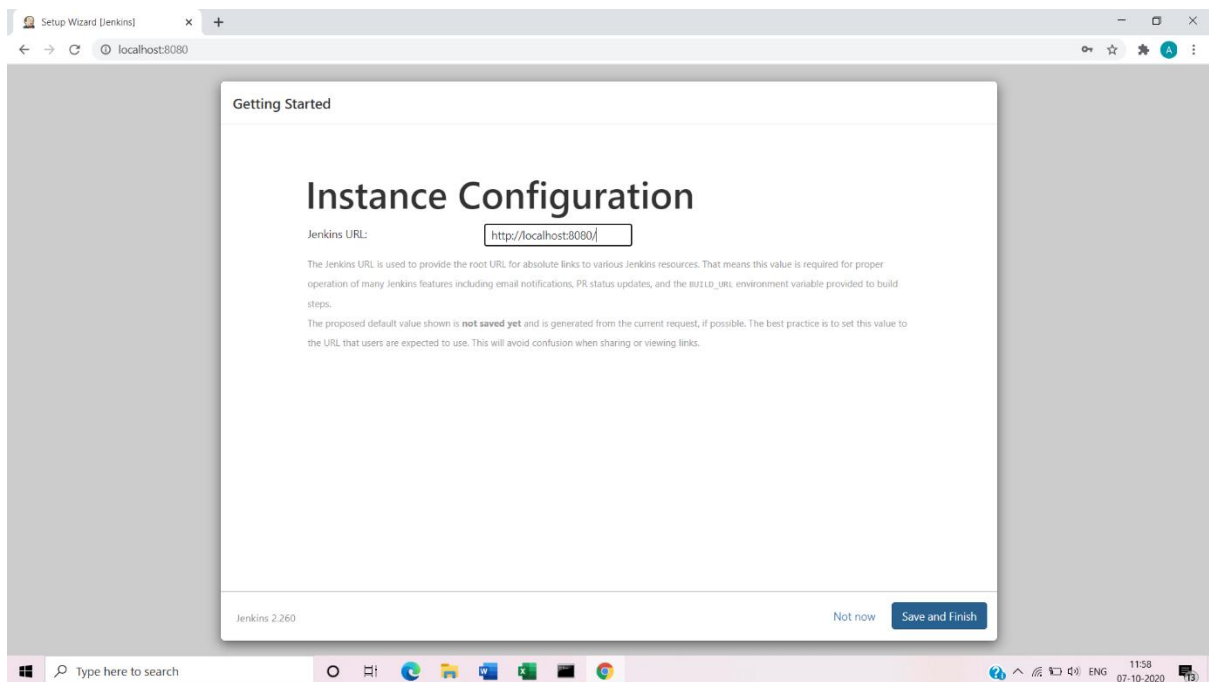
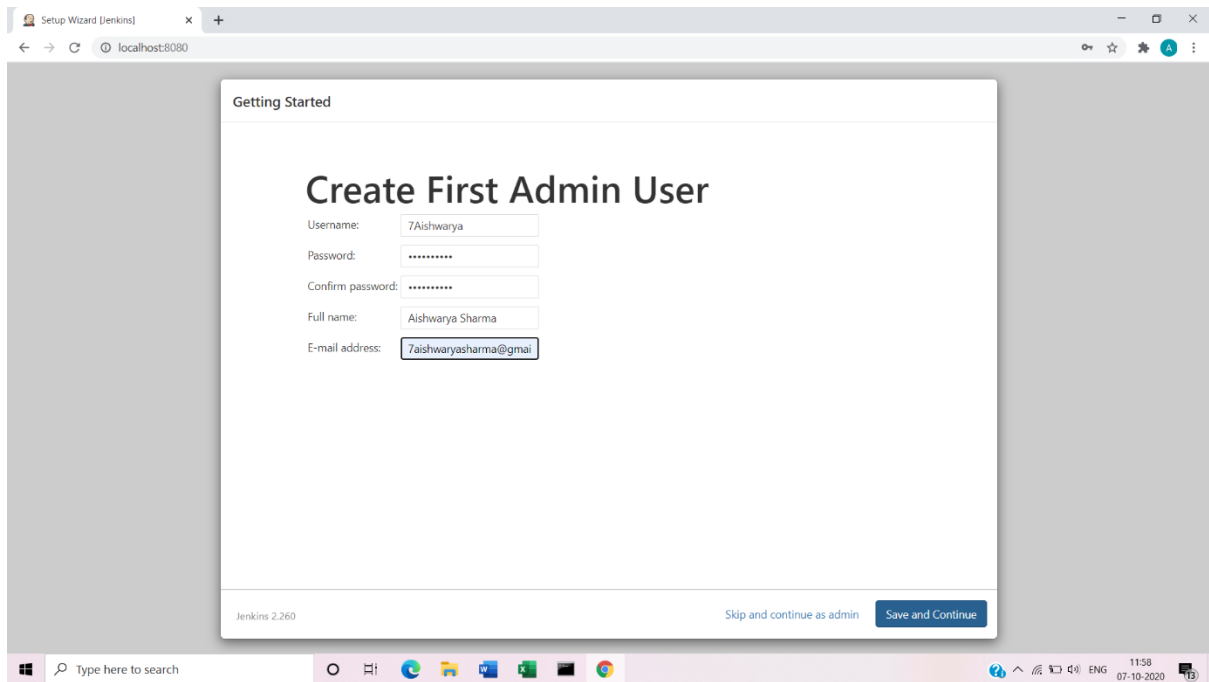
Jenkins requires Java versions [8, 11] but you are running with Java 13 from D:
java.lang.UnsupportedClassVersionError: 57.0
    at Main.verifyJavaVersion(Main.java:174)
    at Main.main(Main.java:142)

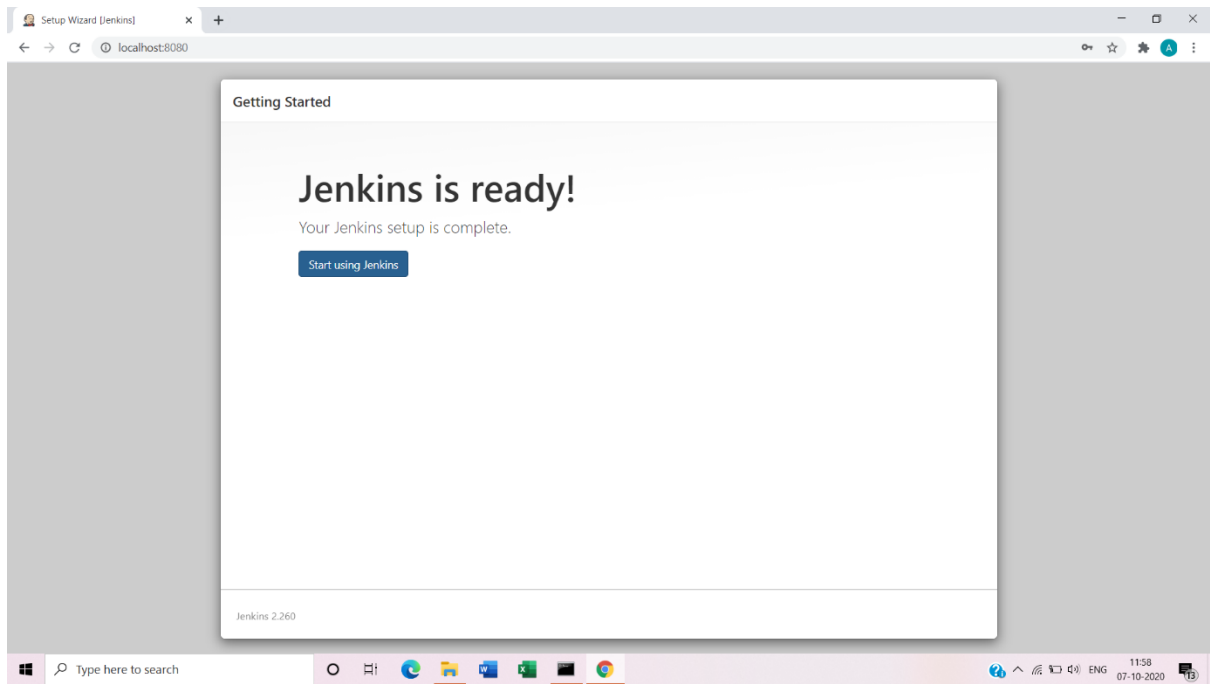
D:\Jenkins>java -jar jenkins.war --enable-future-java
Oct 07, 2020 11:39:06 AM Main verifyJavaVersion
WARNING: Running with Java class version 57 which is not in the list of supported versions: [52, 55]. Argument --enable-future-java is set, so will continue. See https://jenkins.io/redirect/java-support/
Running from: D:\Jenkins\jenkins.war
webroot: $user.home/.jenkins
2020-10-07 06:09:07.574+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: logging initialized @1260ms to org.eclipse.jetty.util.log.JavaUtilLog
2020-10-07 06:09:07.761+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2020-10-07 06:09:12.696+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2020-10-07 06:09:12.829+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.38.v20200611; built: 2020-06-11T12:34:51.929Z; git: 271836ed4cf4612f12b7bb13ef5a92a927634b0d; jvm 13.0.2+8
2020-10-07 06:09:14.059+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#wisitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2020-10-07 06:09:14.175+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=node0
2020-10-07 06:09:14.175+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defaults
2020-10-07 06:09:14.180+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every 60000ms
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.thoughtworks.xstream.core.util.Fields (file:/C:/Users/hp/.jenkins/war/WEB-INF/lib/xstream-1.4.7-jenkins-1.jar) to field java.util.TreeMap.comparator
WARNING: Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2020-10-07 06:09:15.165+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\hp\.jenkins found at: $user.home/.jenkins
2020-10-07 06:09:15.730+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.@96f4da(Jenkins v2.200/, file:///C:/Users/hp/.jenkins/war/, AVAILABLE)(C:\Users\hp\.jenkins\war)
2020-10-07 06:09:16.277+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@7a67e3c6(HTTP/1.1, (http/1.1))(0.0.0.0:8080)
2020-10-07 06:09:16.281+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @9968ms
2020-10-07 06:09:16.289+0000 [id=26] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2020-10-07 06:09:16.463+0000 [id=31] INFO Jenkins.InitReactorRunner$1onAttained: Started initialization
2020-10-07 06:09:16.506+0000 [id=31] INFO Jenkins.InitReactorRunner$1onAttained: Listed all plugins
2020-10-07 06:09:19.449+0000 [id=36] INFO Jenkins.InitReactorRunner$1onAttained: Prepared all plugins
2020-10-07 06:09:19.472+0000 [id=29] INFO Jenkins.InitReactorRunner$1onAttained: Started all plugins
2020-10-07 06:09:19.495+0000 [id=33] INFO Jenkins.InitReactorRunner$1onAttained: Augmented all extensions
2020-10-07 06:09:21.417+0000 [id=33] INFO Jenkins.InitReactorRunner$1onAttained: System config loaded
2020-10-07 06:09:21.435+0000 [id=33] INFO Jenkins.InitReactorRunner$1onAttained: System config adapted
2020-10-07 06:09:21.440+0000 [id=33] INFO Jenkins.InitReactorRunner$1onAttained: Loaded all jobs
2020-10-07 06:09:21.444+0000 [id=35] INFO Jenkins.InitReactorRunner$1onAttained: Configuration for all jobs updated
2020-10-07 06:09:21.552+0000 [id=49] INFO hudson.model.AsyncPeriodicWork$1#doRun$0: Started download metadata
2020-10-07 06:09:21.668+0000 [id=49] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2020-10-07 06:09:23.956+0000 [id=31] INFO o.s.c.s.AbstractApplicationContext#prepareRefresh: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@fe37e46: display name [Root WebApp
licationContext]; startup date [Wed Oct 07 11:39:23 IST 2020]; root of context hierarchy
2020-10-07 06:09:23.981+0000 [id=31] INFO o.s.c.s.AbstractApplicationContext#obtainFreshBeanFactory: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationConte
x]
Type here to search
```

- Now, Jenkins is fully up and running. Copy the password that is generated and use it for further installation.
- Then, go to browser and navigate to localhost:8080
- After that follow the steps as shown in the screenshots below and paste the password under the column administrator password.











## Pushing code to GitHub

Steps:

- Create a GitHub repository and clone it on your local system
- Copy all your files and folders of MERN stack application to this folder.
- Type the following commands:
  - # git add .
  - # git commit -m "commit message"
  - # git push -u origin master

```
hp@LAPTOP-007S0DF9 MINGW64 ~/Desktop/Simple-web-application (main)
$ git add .

hp@LAPTOP-007S0DF9 MINGW64 ~/Desktop/Simple-web-application (main)
$ git commit -m "simple web application"
[main ec8d45b] simple web application
23 files changed, 13847 insertions(+)
create mode 100644 greetings-portal/.gitignore
create mode 100644 greetings-portal/backend/.env
create mode 100644 greetings-portal/backend/models/login.model.js
create mode 100644 greetings-portal/backend/routes/page1.js
create mode 100644 greetings-portal/backend/server.js
create mode 100644 greetings-portal/package-lock.json
create mode 100644 greetings-portal/package.json
create mode 100644 greetings-portal/public/index.html
create mode 100644 greetings-portal/src/welcome.js
create mode 100644 greetings-portal/src/components/App.js
create mode 100644 greetings-portal/src/components/Form.js
create mode 100644 greetings-portal/src/components/Greeting.js
create mode 100644 greetings-portal/src/components/Login.js
create mode 100644 greetings-portal/src/components/afternoon.jpg
create mode 100644 greetings-portal/src/components/evening.jpeg
create mode 100644 greetings-portal/src/components/images/H.ico
create mode 100644 greetings-portal/src/components/images/afternoon.jpg
create mode 100644 greetings-portal/src/components/images/evening.jpeg
create mode 100644 greetings-portal/src/components/images/helpdesk.jpg
create mode 100644 greetings-portal/src/components/images/morning.jpg
create mode 100644 greetings-portal/src/components/morning.jpg
create mode 100644 greetings-portal/src/components/styles.css
create mode 100644 greetings-portal/src/index.js

hp@LAPTOP-007S0DF9 MINGW64 ~/Desktop/Simple-web-application (main)
$ git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/7Aishwarya/Simple-web-application.git'

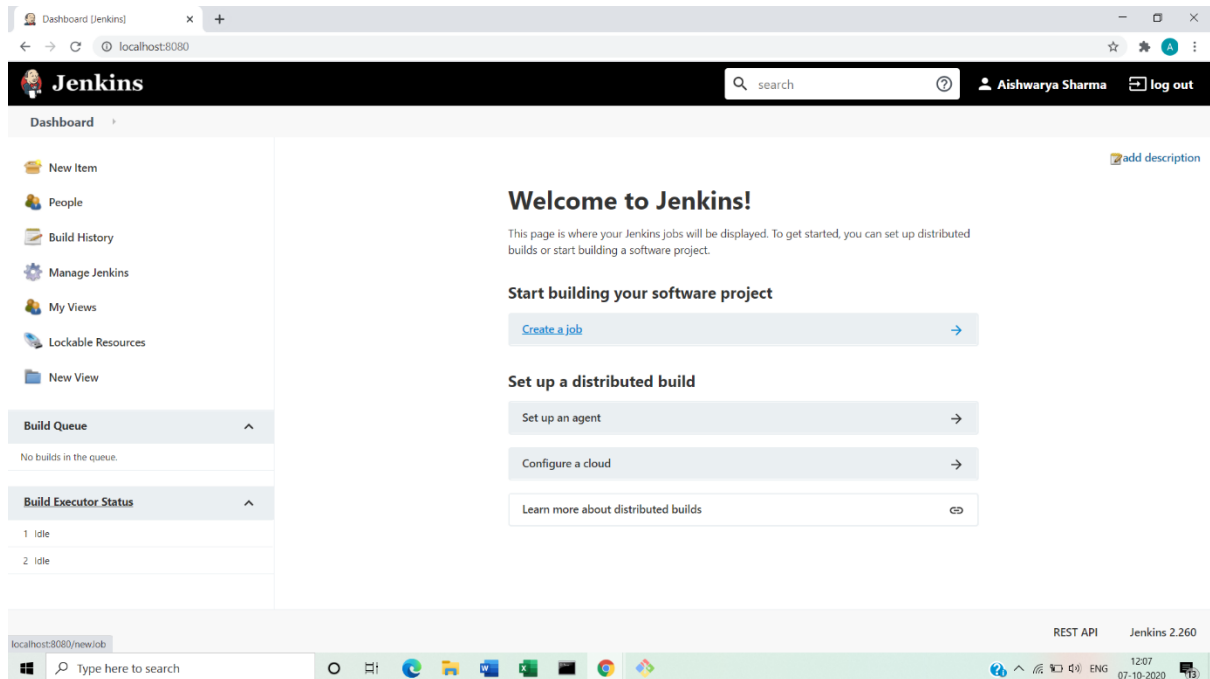
hp@LAPTOP-007S0DF9 MINGW64 ~/Desktop/Simple-web-application (main)
$ git push -u origin main
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 4 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (30/30), 338.89 KiB | 5.06 MiB/s, done.
Total 30 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/7Aishwarya/Simple-web-application.git
   cf55bc7..ec8d45b  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

hp@LAPTOP-007S0DF9 MINGW64 ~/Desktop/Simple-web-application (main)
$
```

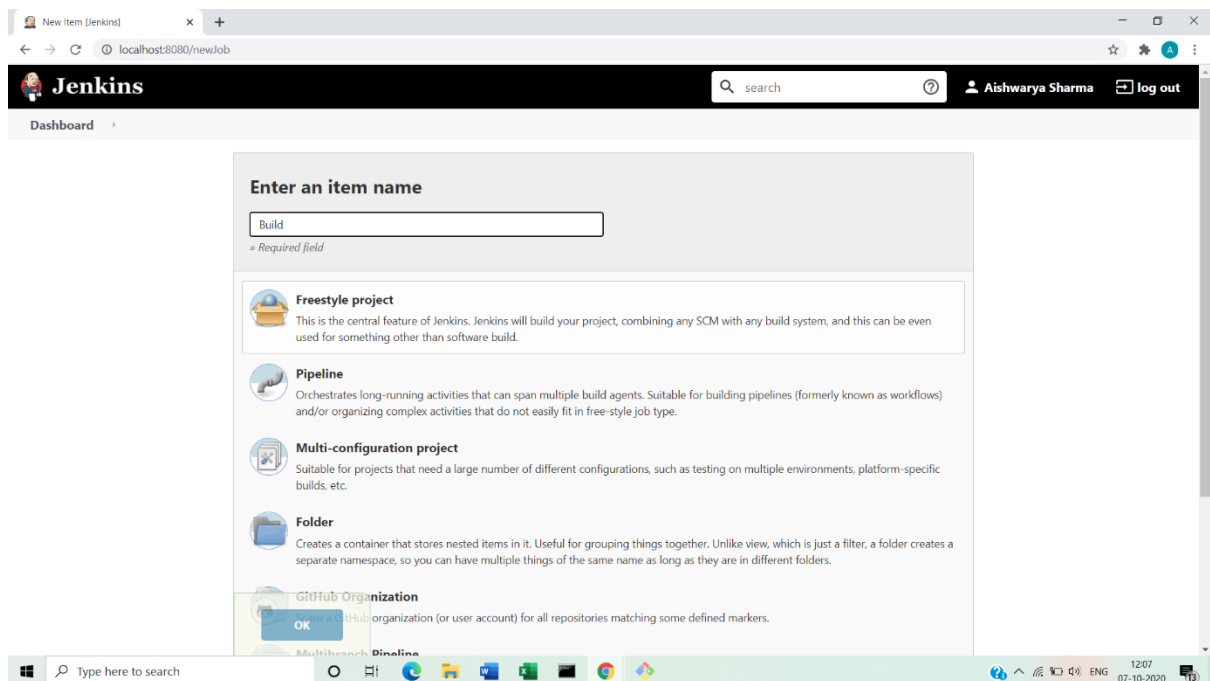
## Creating Job in Jenkins

Steps:

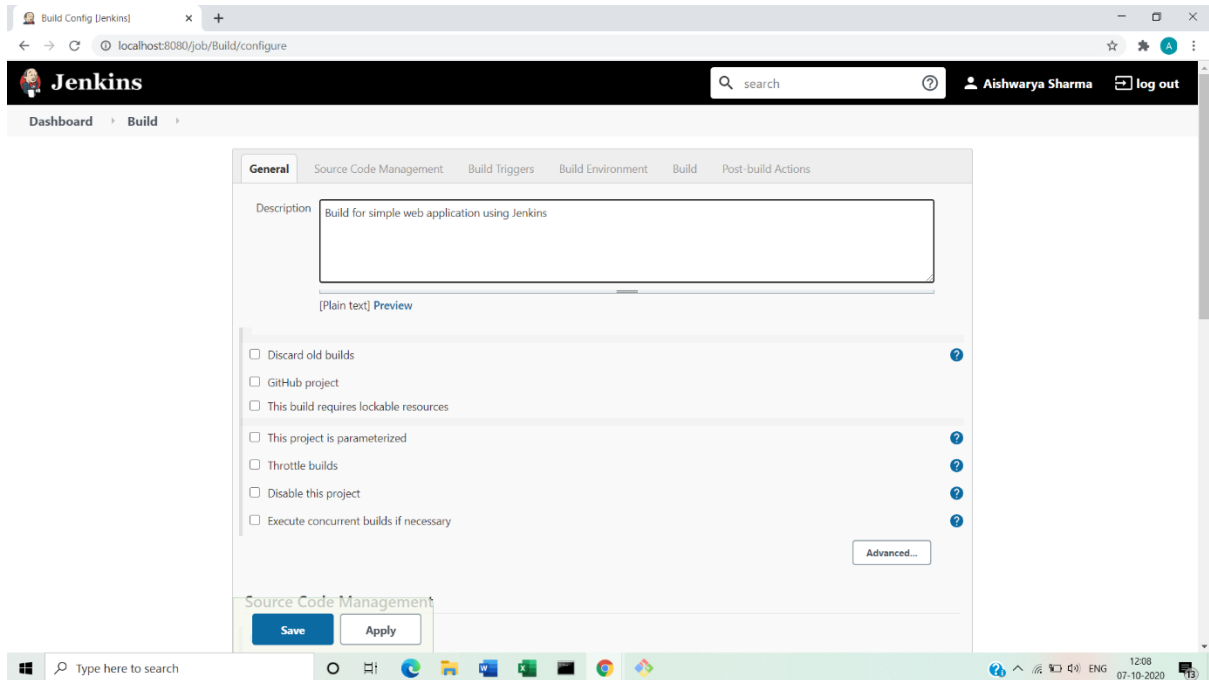
- Click on 'Create a Job'



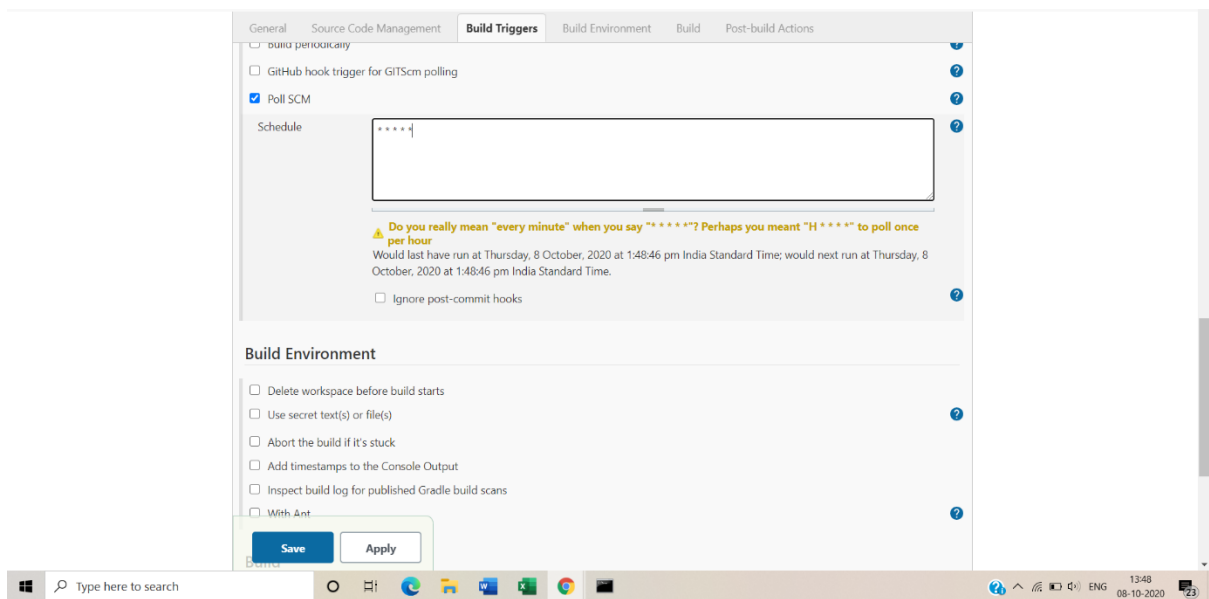
- Select free style project and click OK.



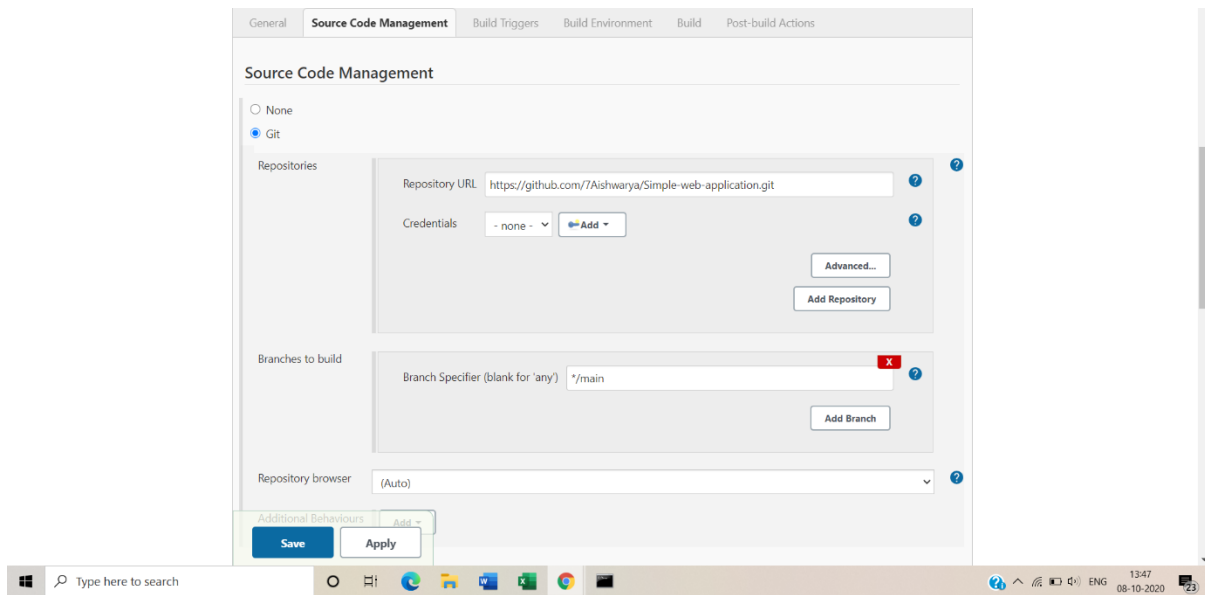
- Follow the steps as shown in the screenshots below.
- Add description



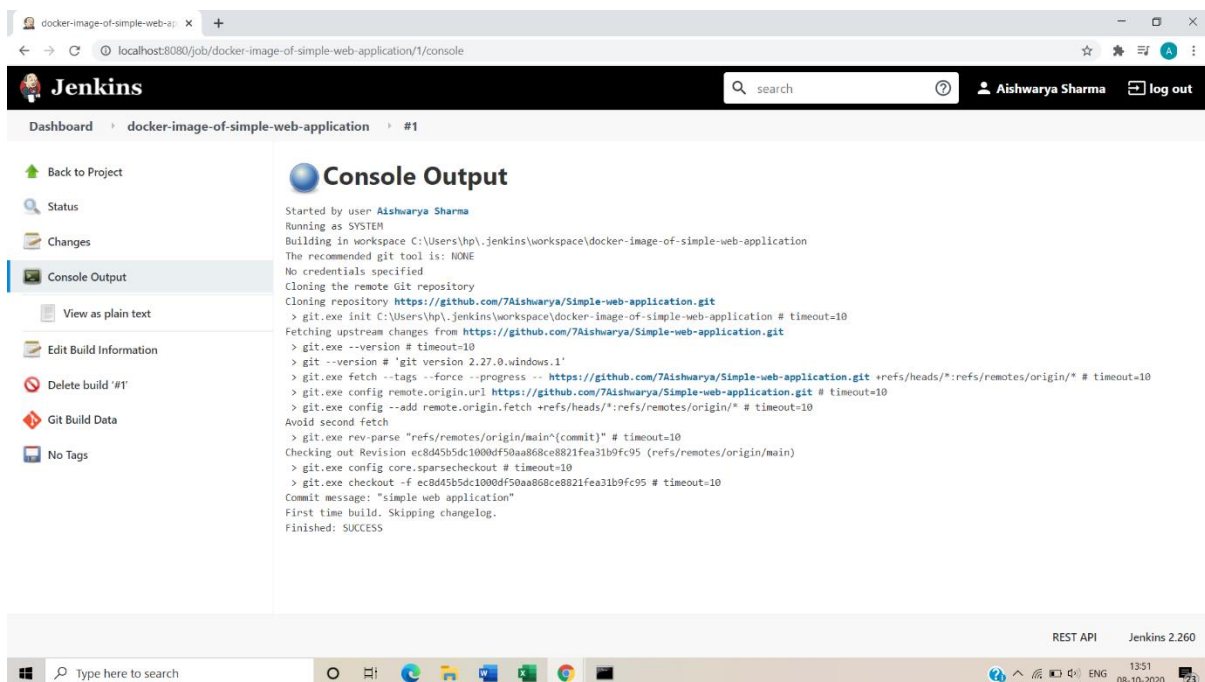
- Build Triggers: Poll SCM and type 5 stars with a space in between.
- (With this step on every commit a build is created)



Source Code Management: Select Git and add the URL of your GitHub repo. Apply and save



- **Create build** and check the output under Console output section on the left side of the dashboard.



## Integrating Git and GitHub with Jenkins + Creating and pushing docker image on Docker Hub.

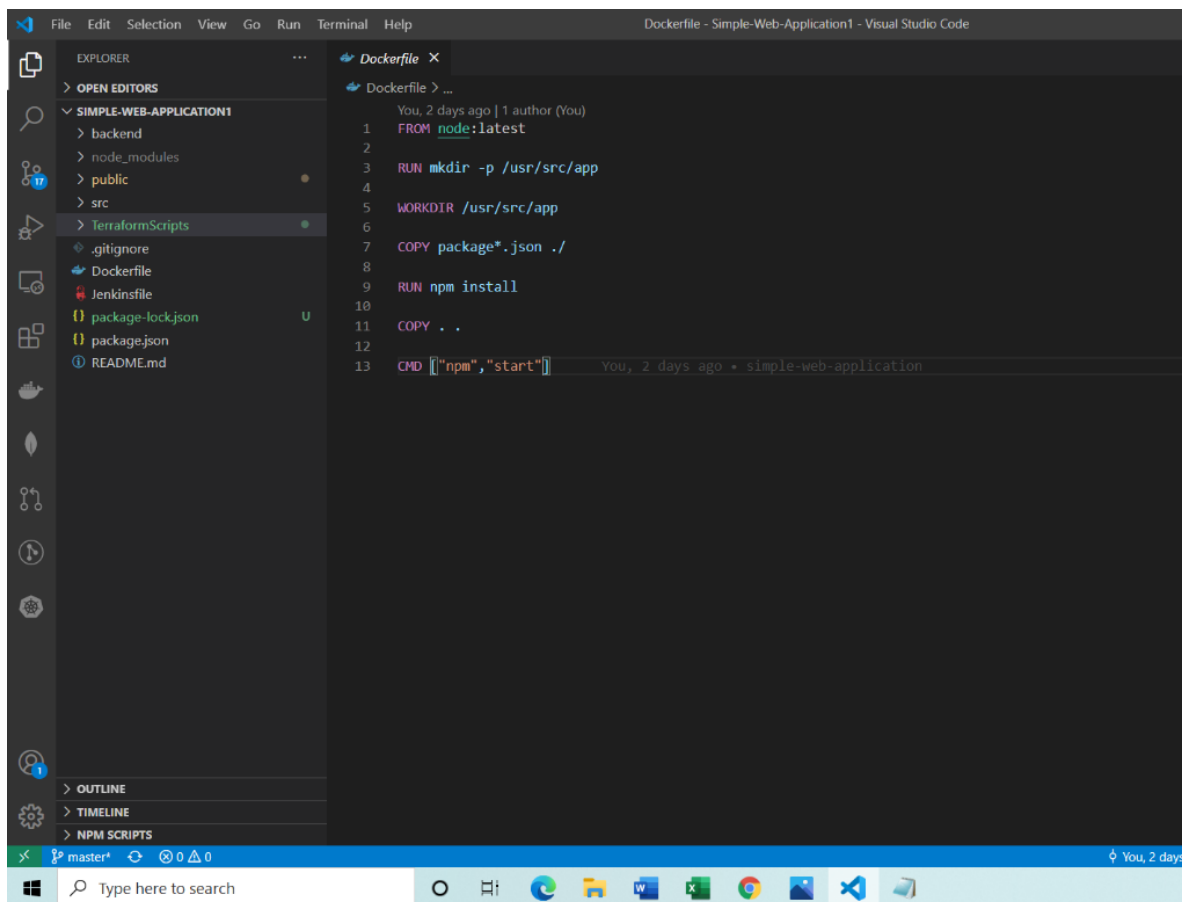
First add Docker File and Jenkins file to your project directory and push them to GitHub using the following commands:

```
# git add .
```

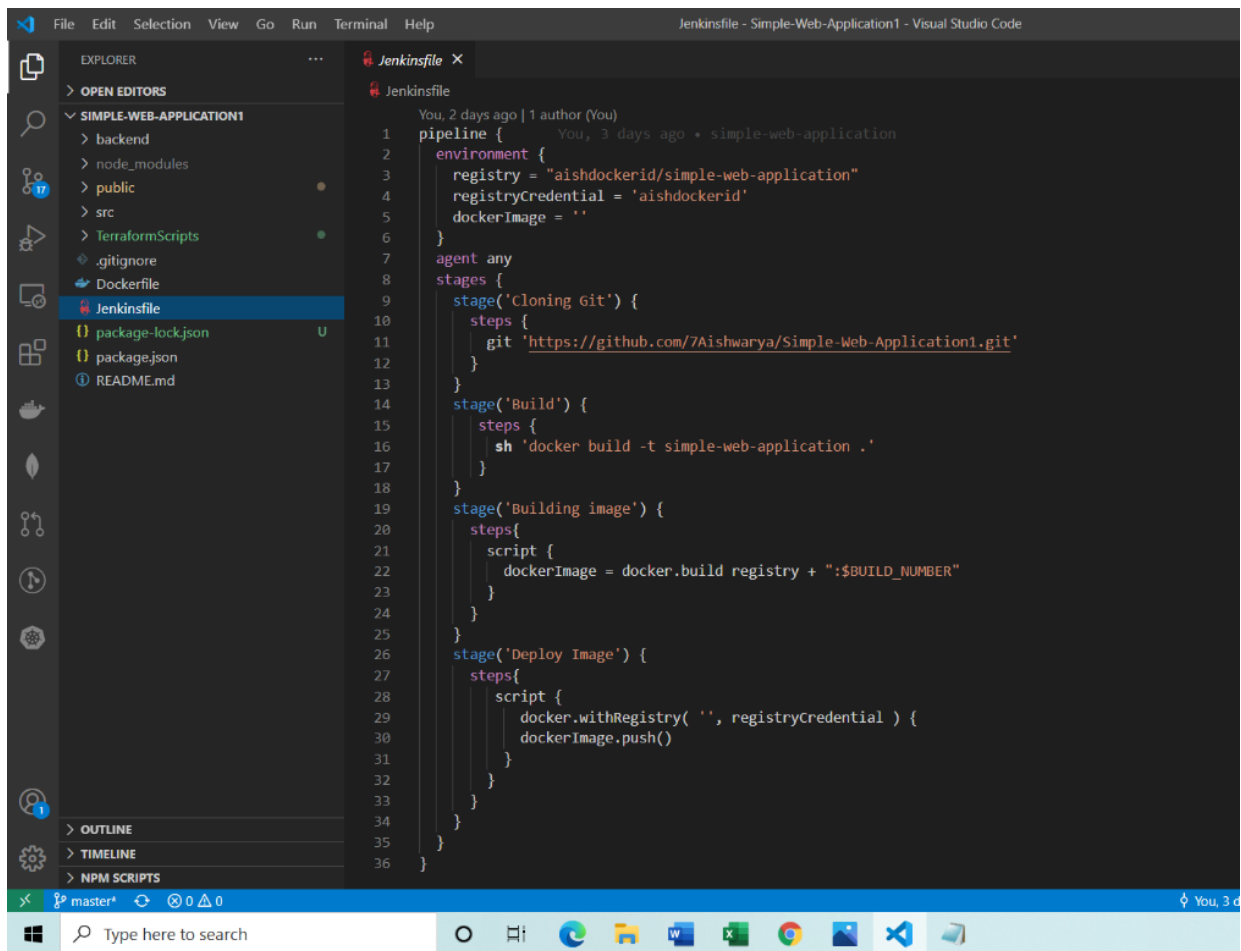
```
# git commit -m "Dockerfile and Jenkinsfile added"
```

```
# git push -u origin master
```

### Dockerfile



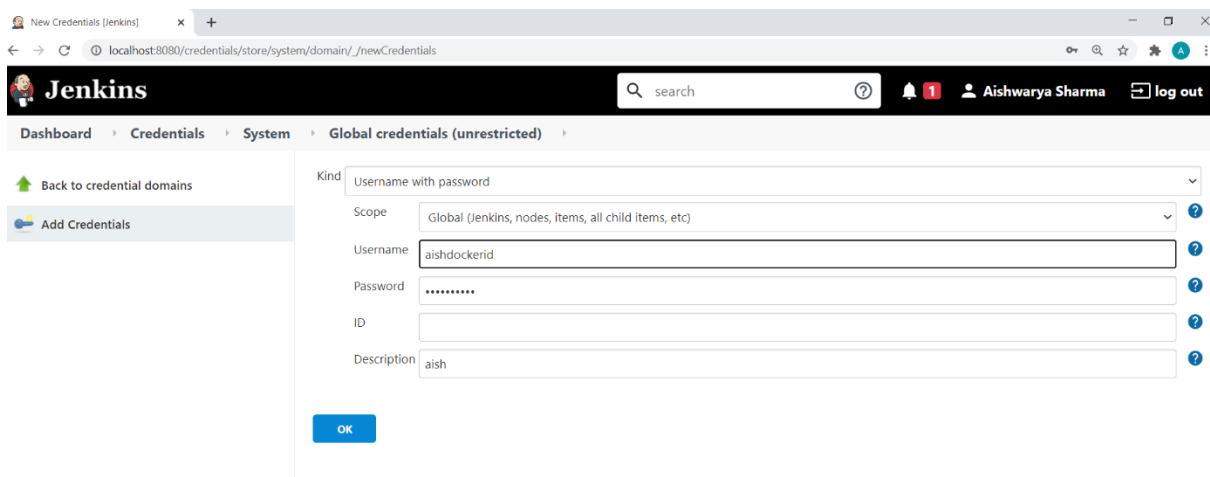
## Jenkinsfile

A screenshot of a Visual Studio Code editor window showing a Jenkinsfile. The Explorer sidebar on the left shows a project named 'SIMPLE-WEB-APPLICATION1' with files like 'package-lock.json', 'package.json', 'README.md', and 'Jenkinsfile'. The Jenkinsfile content is as follows:

```
1 pipeline {
2   You, 2 days ago | 1 author (You)
3   You, 3 days ago * simple-web-application
4   environment {
5     registry = "aishdockerid/simple-web-application"
6     registryCredential = 'aishdockerid'
7     dockerImage = ''
8   }
9   agent any
10  stages {
11    stage('Cloning Git') {
12      steps {
13        git 'https://github.com/7Aishwarya/Simple-Web-Application1.git'
14      }
15    }
16    stage('Build') {
17      steps {
18        sh 'docker build -t simple-web-application .'
19      }
20    }
21    stage('Building image') {
22      steps {
23        script {
24          dockerImage = docker.build registry + ":$BUILD_NUMBER"
25        }
26      }
27    }
28    stage('Deploy Image') {
29      steps {
30        script {
31          docker.withRegistry( '', registryCredential ) {
32            dockerImage.push()
33          }
34        }
35      }
36    }
37  }
38 }
```

Add docker hub credentials to Jenkins

Dashboard -> Credentials -> Global credentials -> Add credentials

A screenshot of the Jenkins 'New Credentials' web interface. The breadcrumb trail shows 'Dashboard > Credentials > System > Global credentials (unrestricted)'. The form is for adding a new credential of type 'Username with password'. The 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'aishdockerid'. The 'Password' field is masked with dots. The 'ID' field is empty. The 'Description' field contains 'aish'. There is an 'OK' button at the bottom.

New Credentials [jenkins] x +

localhost:8080/credentials/store/system/domain/\_/newCredentials

**Jenkins** search ? 1 Aishwarya Sharma log out

Dashboard > Credentials > System > Global credentials (unrestricted) >

Back to credential domains

Add Credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc) ?

Username: aishdockerid ?


Password: ..... ?

ID: ?

Description: aish ?

OK

## Create pipeline

 **Jenkins**


search


Dashboard > All >


### Enter an item name


webdockerpipeline


» 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.

**GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

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

If you want to create a new item from other existing, you can use this option:

 Copy from

Type to autocomplete

OK

Dashboard > webdockerpipeline >

General Build Triggers **Advanced Project Options** Pipeline

### Advanced Project Options

Advanced...

### Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL 

Please enter Git repository.

Credentials 

- none -

 Add

Advanced...  
Add Repository

Branches to build

Branch Specifier (blank for 'any') 

Add Branch

Repository browser (Auto)

Additional Behaviours 

Add

Script Path

Lightweight checkout ☒

[Pipeline Syntax](#)

Jenkins

search

Dashboard

webdockerpipeline

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History

trend

find

11 Oct 2020, 01:12

Atom feed for all

Atom feed for failures

Pipeline webdockerpipeline

Recent Changes

Stage View

Average stage times:  
(Average full run time: ~12min  
10s)

Declarative:  
Checkout  
SCM

Cloning Git

Build

Building  
image

Deploy  
Image

3s

1s

8min 57s

13s

2min 36s

Oct 11  
01:12

No  
Changes

3s

1s

8min 57s

13s

2min 36s

Permalinks

Last build (#1), 14 min ago

Last stable build (#1), 14 min ago

Last successful build (#1), 14 min ago

Last completed build (#1), 14 min ago

## Console Output

Started by user [Aishwarya Sharma](#)

```
Obtained Jenkinsfile from git https://github.com/7Aishwarya/Simple-Web-Application1.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/webdockerpipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/7Aishwarya/Simple-Web-Application1.git
> git init /var/lib/jenkins/workspace/webdockerpipeline # timeout=10
Fetching upstream changes from https://github.com/7Aishwarya/Simple-Web-Application1.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/7Aishwarya/Simple-Web-Application1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/7Aishwarya/Simple-Web-Application1.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 82811e0e50450f0f3ef620b94cff565402208684 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 82811e0e50450f0f3ef620b94cff565402208684 # timeout=10
Commit message: "simple-web-application"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning Git)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/7Aishwarya/Simple-Web-Application1.git # timeout=10
Fetching upstream changes from https://github.com/7Aishwarya/Simple-Web-Application1.git
> git --version # timeout=10
```



Dashboard	webdockerpipeline	#1	<pre> FILE 20190905190505_MSL_LINUX_1.0 - PLAIN No credentials specified &gt; git rev-parse --is-inside-work-tree # timeout=10 Fetching changes from the remote Git repository &gt; git config remote.origin.url https://github.com/7Aishwarya/Simple-Web-Application1.git # timeout=10 Fetching upstream changes from https://github.com/7Aishwarya/Simple-Web-Application1.git &gt; git --version # timeout=10 &gt; git --version # 'git version 2.25.1' &gt; git fetch --tags --force --progress -- https://github.com/7Aishwarya/Simple-Web-Application1.git +refs/heads/*:refs/remotes/origin/* # timeout=10 &gt; git rev-parse refs/remotes/origin/master^{commit} # timeout=10 Checking out Revision 82811e0e50450f03ef629b94cff565402208684 (refs/remotes/origin/master) &gt; git config core.sparsecheckout # timeout=10 &gt; git checkout -f 82811e0e50450f03ef629b94cff565402208684 # timeout=10 &gt; git branch -s -v --no-abbrev # timeout=10 &gt; git checkout -b master 82811e0e50450f03ef629b94cff565402208684 # timeout=10 Commit message: "simple-web-application" [Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (Build) [Pipeline] sh * docker build -t simple-web-application . Sending build context to Docker daemon 789.5kB  Step 1/7 : FROM node:latest ----&gt; c0a69c6687d2 Step 2/7 : RUN mkdir -p /usr/src/app ----&gt; Using cache ----&gt; 27eb64b916d Step 3/7 : WORKDIR /usr/src/app ----&gt; Using cache ----&gt; 909b813e7f79 Step 4/7 : COPY package*.json ./ ----&gt; 359b8d41ca98 Step 5/7 : RUN npm install ----&gt; Running in 8d528370d1b6 </pre>
Dashboard	webdockerpipeline	#1	<pre> \$ docker login -u aishdockerid -p ***** https://index.docker.io/v1/ WARNING! Using --password via the CLI is insecure. Use --password-stdin. WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/webdockerpipeline@tmp/9147ed6c-2468-417d-b574-79fe66459264/config.json. Configure a credential helper to remove this warning. See https://docs.docker.com/engine/reference/commandline/login/#credentials-store  Login Succeeded [Pipeline] { [Pipeline] isUnix [Pipeline] sh * docker tag aishdockerid/simple-web-application:1 aishdockerid/simple-web-application:1 [Pipeline] isUnix [Pipeline] sh * docker push aishdockerid/simple-web-application:1 The push refers to repository [docker.io/aishdockerid/simple-web-application] 0aa614212ce0: Preparing 7744bea74119: Preparing 1b75f28143c6: Preparing 74976214cccb: Preparing 74ef25c221ef: Preparing 193b269f2c3f: Preparing 4a0e6e0a73a2: Preparing 193b269f2c3f: Waiting 174e334f3f46: Preparing 4a0e6e0a73a2: Waiting cbe6bb0c86f: Preparing ef5de533cb53: Preparing 174e334f3f46: Waiting a4c504f73441: Preparing ef5de533cb53: Waiting e8847c2734e1: Preparing b323b70996e4: Preparing e8847c2734e1: Waiting b323b70996e4: Waiting a4c504f73441: Waiting </pre>

```

//440ea/4119: PUSHed
1: digest: sha256:d1f0ebabe05f2a55e11ca990f74d7c97f33c74cbf79c0f6201d298c619471c97 size: 3051
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

We've updated our Terms of Service. [Learn more.](#)

docker hub

Search for great content (e.g., mysql)

Explore

Repositories

Organizations

Get Help ▾

aishdockerid ▾

Repositories

aishdockerid / simple-web-application >

Using 0 of 1 private repositories. [Get more](#)

General

Tags

Builds

Timeline

Collaborators

Webhooks

Settings

aishdockerid / simple-web-application

This repository does not have a description

Last pushed: 7 minutes ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push aishdockerid/simple-web-application:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 1 tag(s).

TAG	OS	PUSHED
1		7 minutes ago

[See all](#)

Recent builds

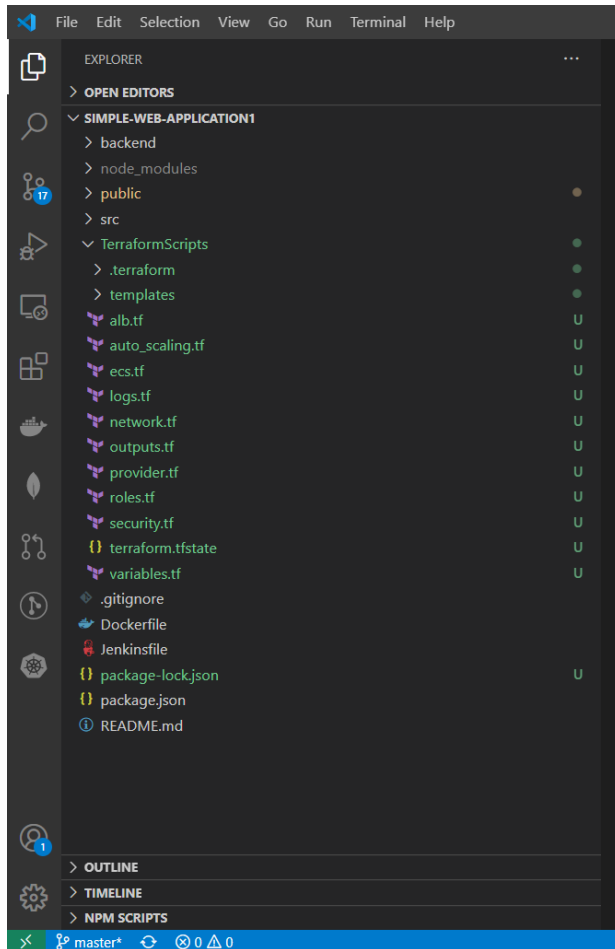
[Link a source provider and run a build to see build results here.](#)

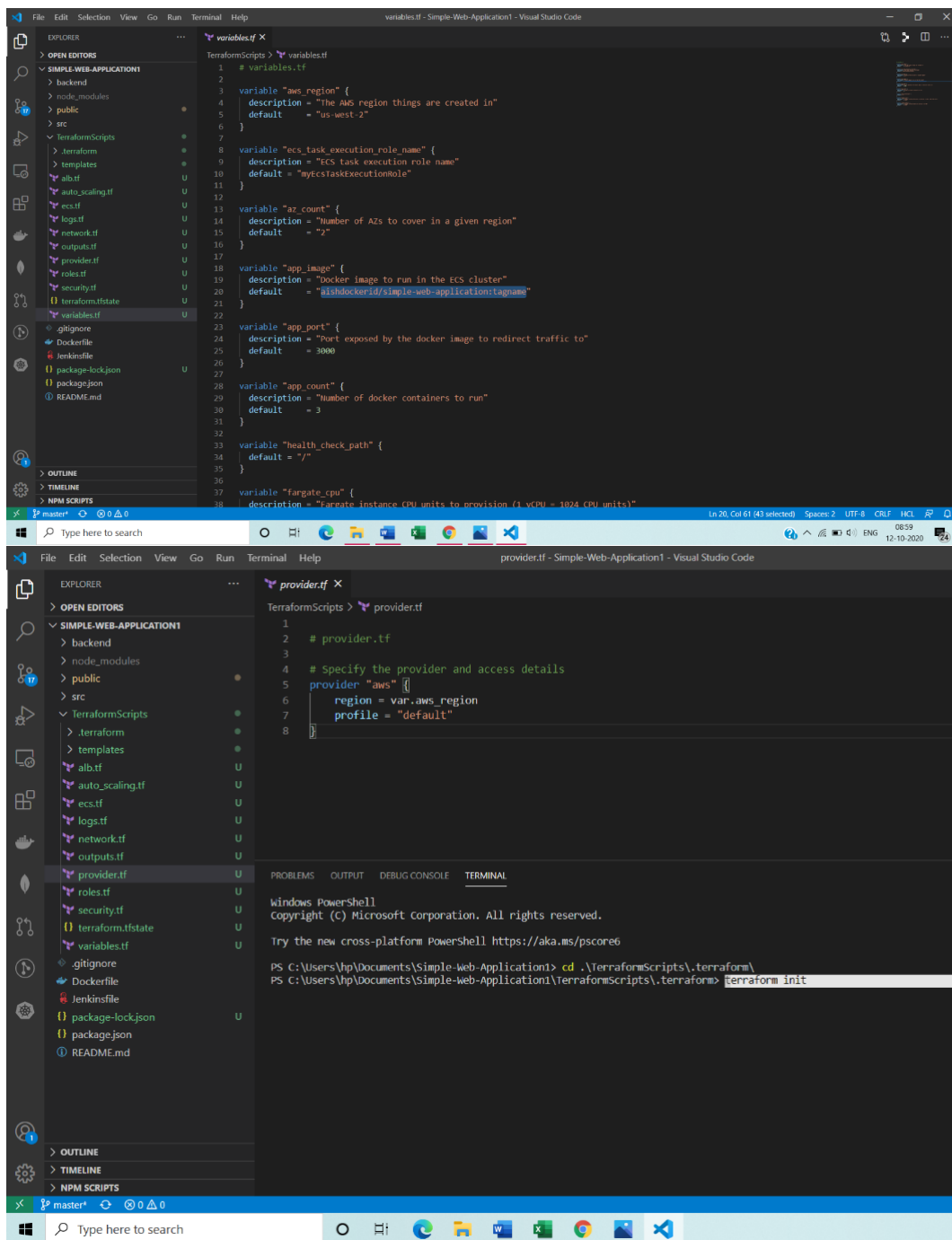
Readme

Repository description is empty. Click [here](#) to edit.

## Deploying Image using Terraform and AWS platform

Add Terraform scripts to your project directory





Refer to the screenshots above to make changes to variables.tf and provider.tf file.

You have to provide your dockerhub credentials and github repo in variables.tf

In provider.tf in place of default you have to enter the [name] as mentioned in credentials file inside .aws folder.

Then initialize the working directory with terraform configuration files using the following command.

# terraform init

# configure aws

//provide credentials (you get the credentials from your aws console)

Go to AWS console -> select your username at the top -> My security credentials -> Access keys -> Create new Access keys or use existing keys

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and user information (7Aishwarya, N. Virginia, Support). The main content area is titled 'AWS Management Console' and features a 'Find Services' search bar, a 'Recently visited services' list, and a grid of service categories including Compute, Satellite, Security, Identity, & Compliance, Containers, Quantum Technologies, and Management & Governance. The right sidebar contains links to 'My Account', 'My Organization', 'My Service Quotas', 'My Billing Dashboard', 'My Security Credentials', 'Sign Out', and 'Learn more'. Below this, there are sections for 'Explore AWS' (Move to Managed File Storage, AWS Certification) and 'Free Digital Training'.

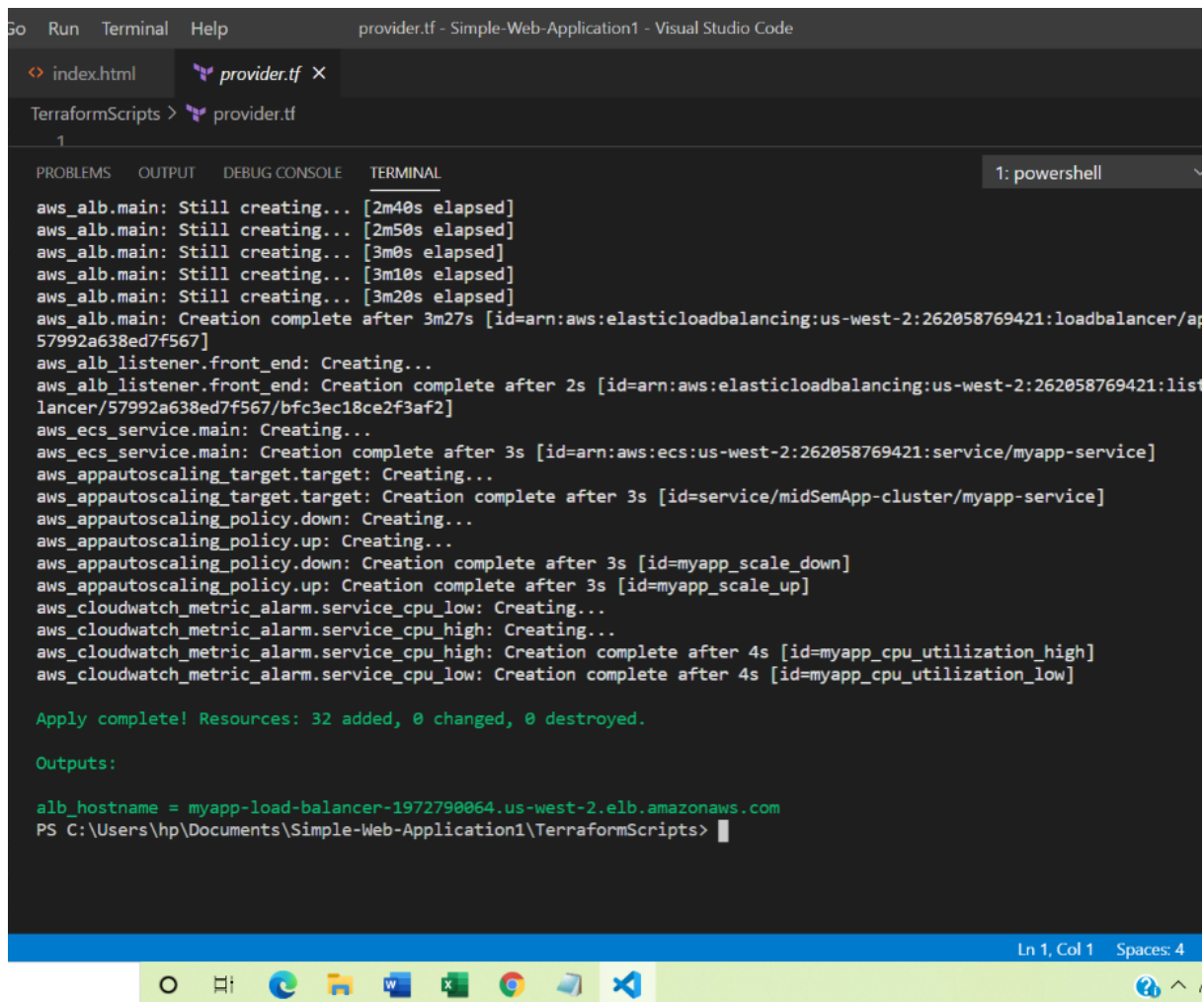
The bottom section of the screenshot shows the 'IAM Management Console' with the 'Your Security Credentials' page. The page title is 'Your Security Credentials' and it includes a description: 'Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console. To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.' The page lists three credential types: Password, Multi-factor authentication (MFA), and Access keys (access key ID and secret access key). The 'Access keys' section is expanded, showing a table of access keys:

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Oct 12th 2020	AKIAJHWRGKCDXCTGWSNQ	2020-10-18 18:37 UTC+0530	us-west-2	elasticloadbalancing	Inactive	Make Active   Delete
Oct 18th 2020	AKIAIYZDX3S2DYKXP4MA	2020-10-18 21:05 UTC+0530	ap-south-1	cloudwatch	Active	Make Inactive   Delete

Below the table is a 'Create New Access Key' button. A warning box states: 'Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. Learn more'. The left sidebar shows the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Access reports, and Credential report.

# terraform plan

# terraform apply



The screenshot shows a Visual Studio Code window with a terminal open. The terminal title is "provider.tf - Simple-Web-Application1 - Visual Studio Code". The terminal content shows the output of a Terraform apply command. It lists the creation of various AWS resources including an Elastic Load Balancing (ALB), an ALB Listener, an ECS Service, an Auto Scaling Target, an Auto Scaling Policy, and CloudWatch Metric Alarms. The output concludes with "Apply complete! Resources: 32 added, 0 changed, 0 destroyed." and displays the output variable "alb\_hostname" as "myapp-load-balancer-1972790064.us-west-2.elb.amazonaws.com". The terminal prompt is "PS C:\Users\hp\Documents\Simple-Web-Application1\TerraformScripts>".

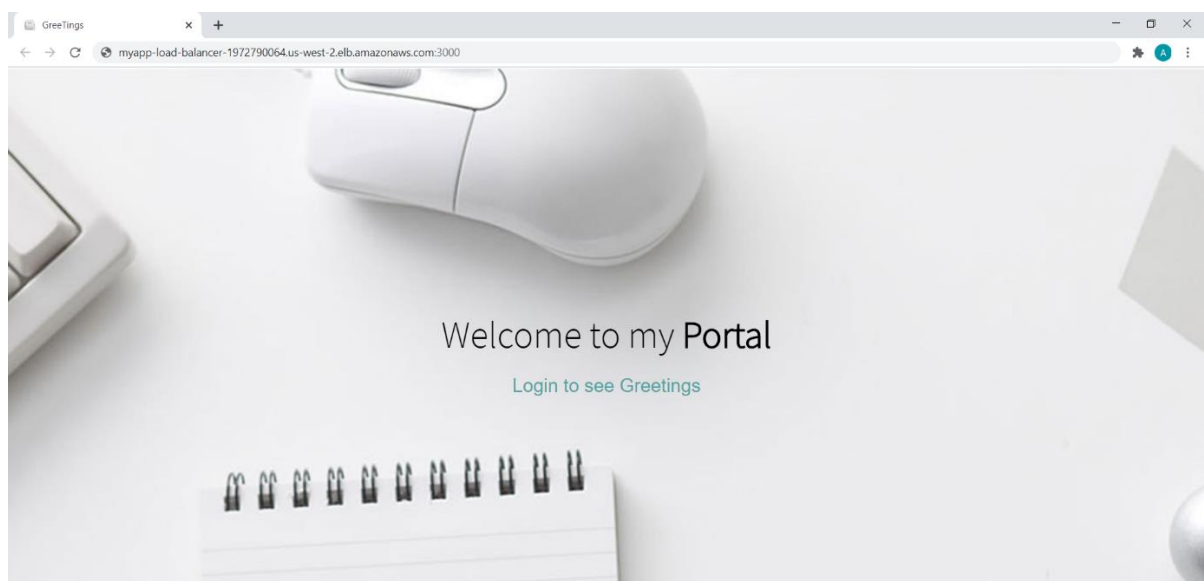
```
aws_alb.main: Still creating... [2m40s elapsed]
aws_alb.main: Still creating... [2m50s elapsed]
aws_alb.main: Still creating... [3m0s elapsed]
aws_alb.main: Still creating... [3m10s elapsed]
aws_alb.main: Still creating... [3m20s elapsed]
aws_alb.main: Creation complete after 3m27s [id=arn:aws:elasticloadbalancing:us-west-2:262058769421:loadbalancer/ap
57992a638ed7f567]
aws_alb_listener.front_end: Creating...
aws_alb_listener.front_end: Creation complete after 2s [id=arn:aws:elasticloadbalancing:us-west-2:262058769421:listen
lancer/57992a638ed7f567/bfc3ec18ce2f3af2]
aws_ecs_service.main: Creating...
aws_ecs_service.main: Creation complete after 3s [id=arn:aws:ecs:us-west-2:262058769421:service/myapp-service]
aws_appautoscaling_target.target: Creating...
aws_appautoscaling_target.target: Creation complete after 3s [id=service/midSemApp-cluster/myapp-service]
aws_appautoscaling_policy.down: Creating...
aws_appautoscaling_policy.up: Creating...
aws_appautoscaling_policy.down: Creation complete after 3s [id=myapp_scale_down]
aws_appautoscaling_policy.up: Creation complete after 3s [id=myapp_scale_up]
aws_cloudwatch_metric_alarm.service_cpu_low: Creating...
aws_cloudwatch_metric_alarm.service_cpu_high: Creating...
aws_cloudwatch_metric_alarm.service_cpu_high: Creation complete after 4s [id=myapp_cpu_utilization_high]
aws_cloudwatch_metric_alarm.service_cpu_low: Creation complete after 4s [id=myapp_cpu_utilization_low]

Apply complete! Resources: 32 added, 0 changed, 0 destroyed.

Outputs:

alb_hostname = myapp-load-balancer-1972790064.us-west-2.elb.amazonaws.com
PS C:\Users\hp\Documents\Simple-Web-Application1\TerraformScripts>
```

Now you can access your application through the link which you get as alb\_hostname.



Everything is done! Cheers :D