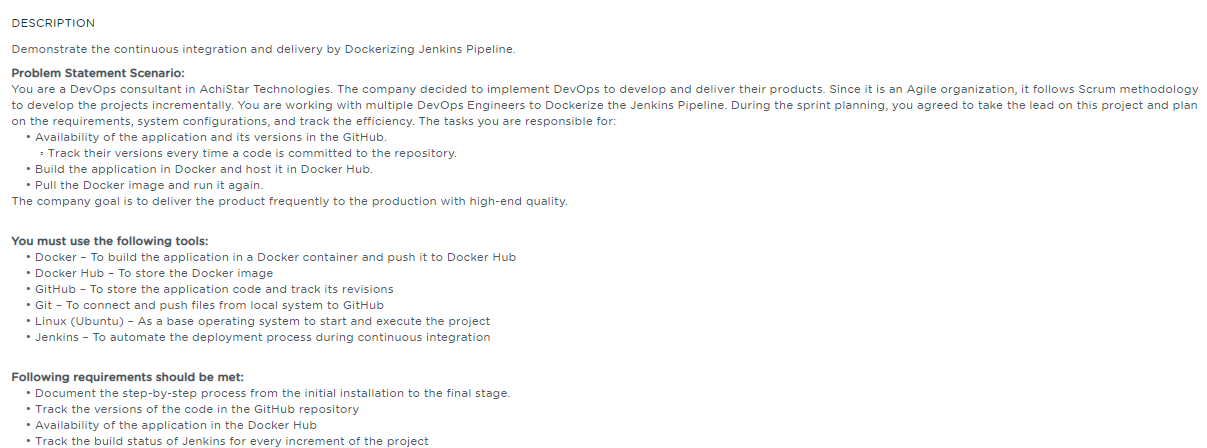
**Dockerizing Jenkins Pipeline - Assessment**



# Overview

This assessment is intended to demonstrate the Continuous Integration and Continuous Delivery of Node JS application by Dockerizing the Jenkins pipeline

# Steps

* **Step 1** - Install Git and setup your GitHub account
  + **Step 1.1** - Install Git
  + **Step 1.2** - Setup GitHub account
  + **Step 1.3** - Login from Git local to remote GitHub
  + **Step 1.4** – Initialize Git and add the Project into the Git repo
* **Step 2** - Create a Project repository in GitHub
  + **Step 2**.1 – Create a new project repository
  + **Step 2.2 –** Creation of SSH Key and adding it to GitHub.
* **Step 3** - Install and configure Jenkins
  + **Step 3.1** - Add a key to use Debian repository
  + **Step 3.2** - Make changes to /etc/opt/sources.list
  + **Step 3.3** - Update local package index
  + **Step 3.4** - Run Jenkins in browser
* **Step 4** - Create a build pipeline in Jenkins
* Docker - To build the application in a Docker container and push it to DockerHub
* Creating an account in the DockerHub – To store the docker images

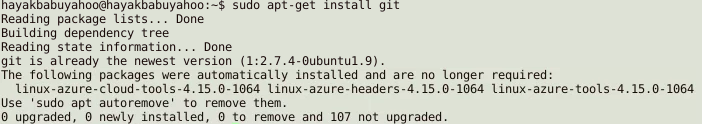
# Step 1 - Install Git and setup your GitHub account

## Install Git

If Git is already installed in your machine, you can check the version of git by executing the below command in the terminal.



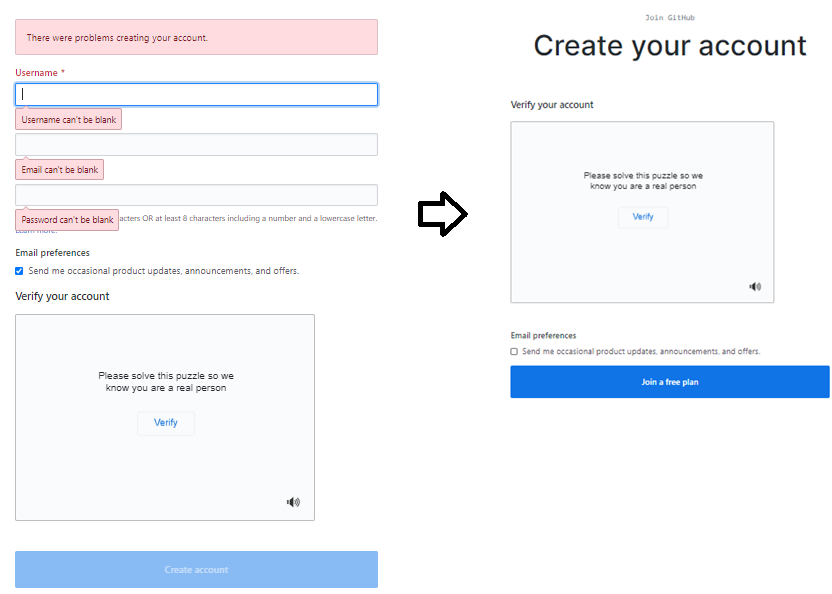
If git is not installed, then you can follow the below steps to install git



## Setup GitHub account

**About GitHub**: It is a web-based hosting service for version control using Git. It offers plans for public and private repositories. You can add multiple projects by creating multiple public repositories. In this section, you will only demonstrate on the public repository and its usage.

Navigate to **https://github.com/** and click on Sign up for GitHub. Enter the details and click on **Create account.**

****

Click **Join a free plan** to create a free plan. You will receive an email to confirm your account. It is important to confirm your account before you use GitHub. Once confirmed, your GitHub account is set up successfully.

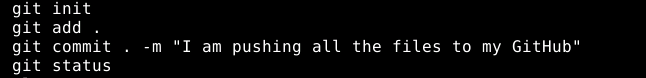
## Login from Git local to remote GitHub

Open the terminal in your lab and execute the below commands by replacing <your\_Email\_Id with your registered email address in GitHub and <Your Username> with your GitHub username.

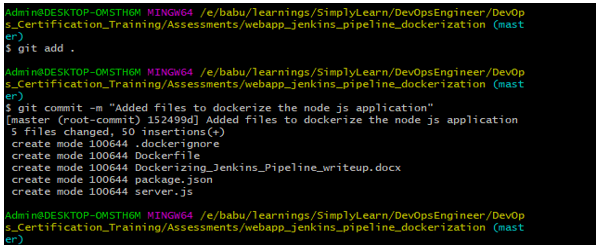


## Initialize Git and add the project into the local Git repo

Since the project (here, Node JS application) files are to be pushed, initialize a .git folder inside the directory by executing the below commands



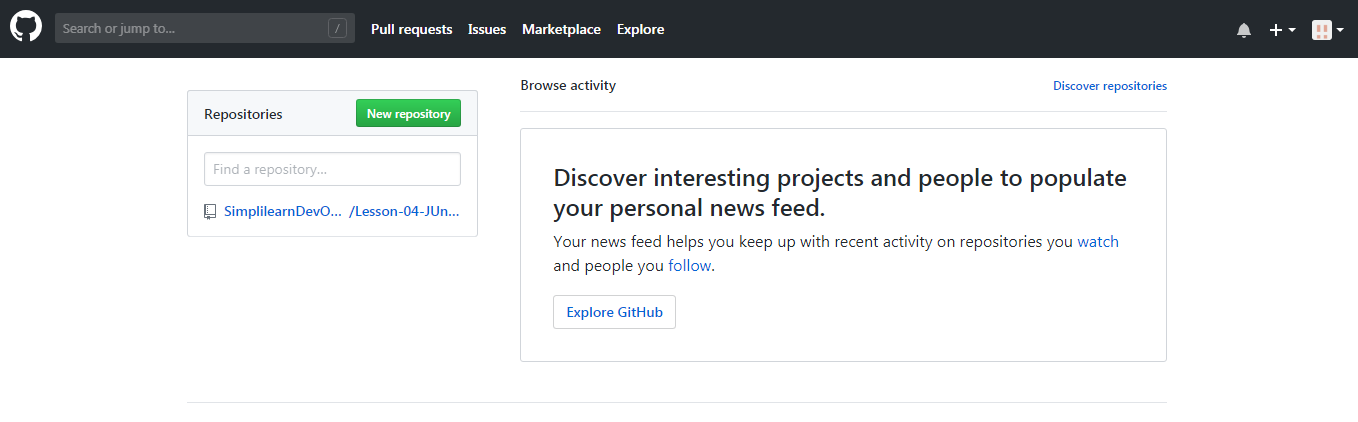
Please follow the below process for step-by-step confirmation of each command execution.



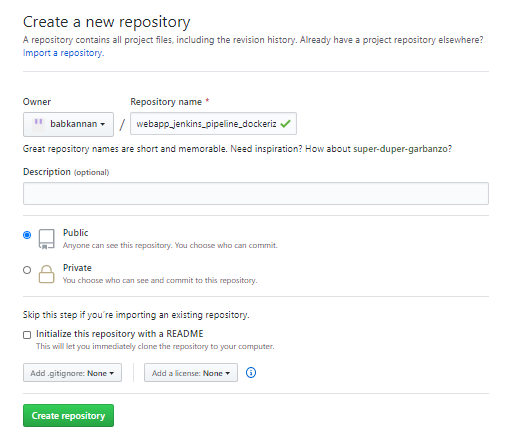
The above commands stage all the project files and pushed into the local Git repository.

## Create a Project repository in GitHub account

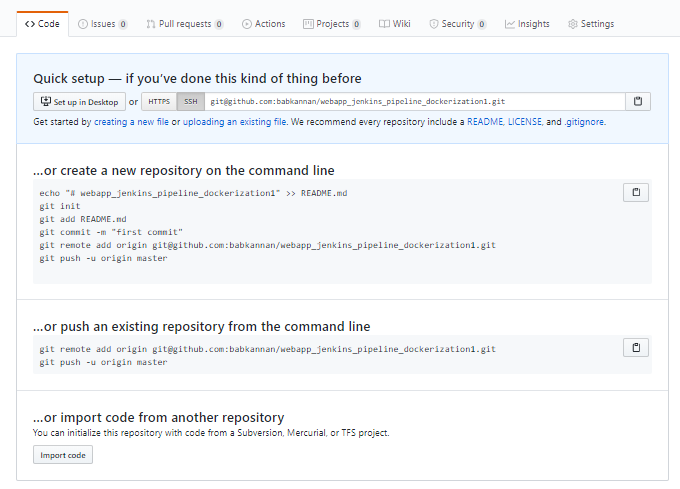
Go to the homepage of GitHub.com and click on **New Repository** as shown below.



Enter the name as “**webapp\_jenkins\_pipeline\_dockerization**” and click on **Create repository**.



You will be redirected to a quick guide page and you will be navigated automatically inside the directory you have created.

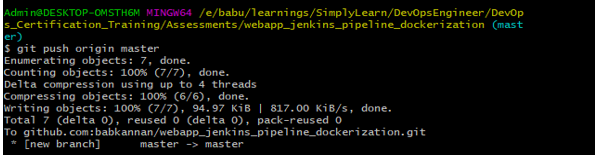


Since a repository is already created, **“…or create a new repository on the command line**” should be skipped. Click on **SSH** to change the instructions from **HTTPS** to **SSH**.

Copy the git remote add origin <URL\_of\_Your\_GitHub\_Repository> and execute it in the terminal.

|  |
| --- |
| git remote add origin git@github.com:babkannan/webapp\_jenkins\_pipeline\_dockerization.git  git push origin master |

The screenshot below shows that the Node js application is successfully pushed into the Git repository



If you’re unable to push the files to your Github.com account, then follow the below steps:

**Creation of SSH Key and adding it to GitHub.**

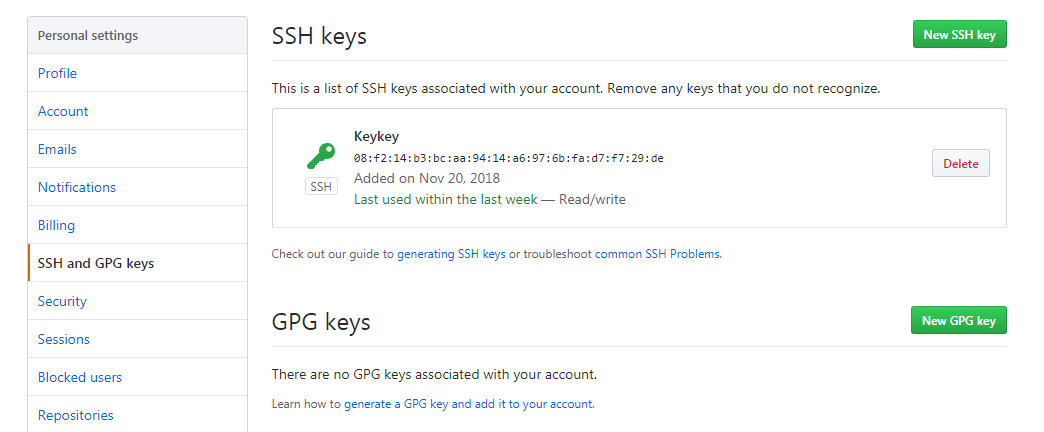
Switch the current directory to ssh by executing below command.

cd ~/.ssh

Generate an RSA key for the registered email Id. (An example is available below)

|  |
| --- |
| ssh key-gen -t rsa -C “<your email\_id>”  gedit id\_rsa.pub |

Copy the entire key from the clipboard. Choose ***Your avatar* > settings** > **SSH & GPG Keys** and click on **New SSH** **key** and paste the key and **save** it.

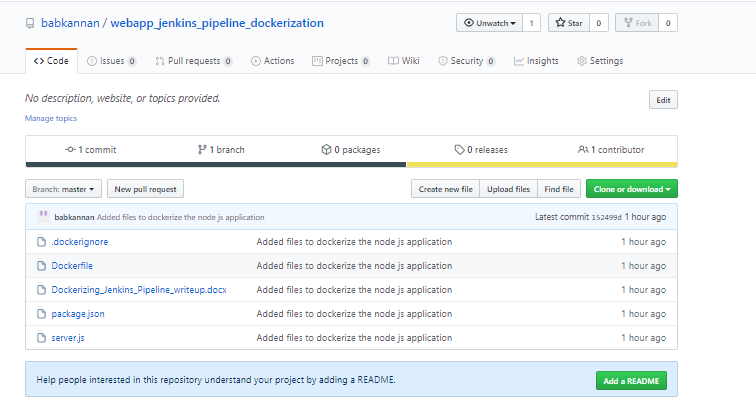


In the terminal, execute **ssh-add** to save the key and link it with local git.

Copy the git remote add origin <URL\_of\_Your\_GitHub\_Repository> and execute it in the terminal.

|  |
| --- |
| git remote add origin git@github.com:babkannan/webapp\_jenkins\_pipeline\_dockerization.git  git push -u origin master |

Reload your GitHub.com account to confirm the output shown below.



# Install and configure Jenkins

Go to <https://pkg.jenkins.io/debian-stable/> to understand how to install Jenkins on a Ubuntu/Debian machine.

## Add a key to use Debian repository

To use the Debian repository of Jenkins to automate installation and upgrade, first add the key to your system using the following command:

|  |
| --- |
| wget -q -O – <https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add – |

## Make changes to /etc/opt/sources.list

Add the following entry in your **/etc/apt/sources.list**:

|  |
| --- |
| sudo vi /etc/apt/sources.list |

Now add the following command

|  |
| --- |
| deb [https://pkg.jenkins.io/debian-stable binary/](https://pkg.jenkins.io/debian-stable%20binary/) |

## Update local package index

|  |
| --- |
| sudo apt-get update |

Make sure you have jdk installed. If not, install it using the following command:

|  |
| --- |
| sudo apt-get install openjdk-8-jdk |

Then install Jenkins.

|  |
| --- |
| sudo apt-get install jenkins |

## Run Jenkins in browser

Once Jenkins is installed, open a browser on your local machine and enter the

URL: x.x.x.x:8080

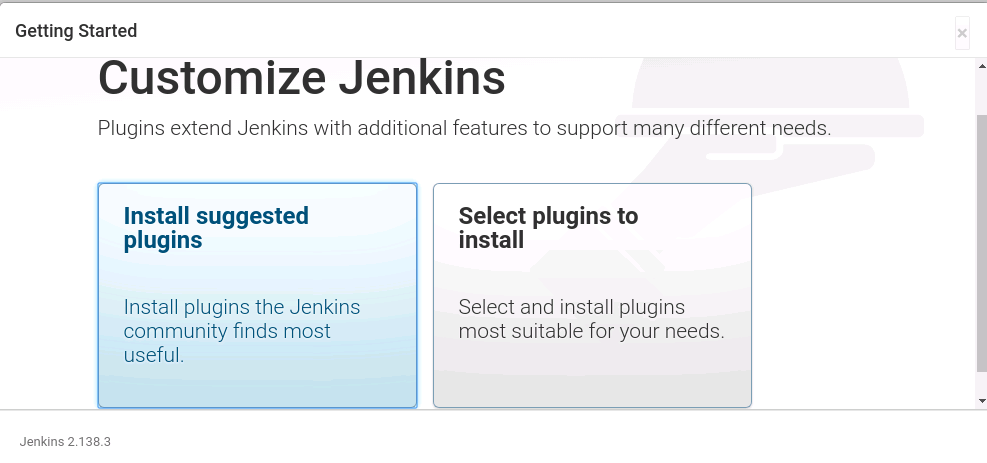
replacing x.x.x.x with the external IP address of your virtual machine or run(or localhost)

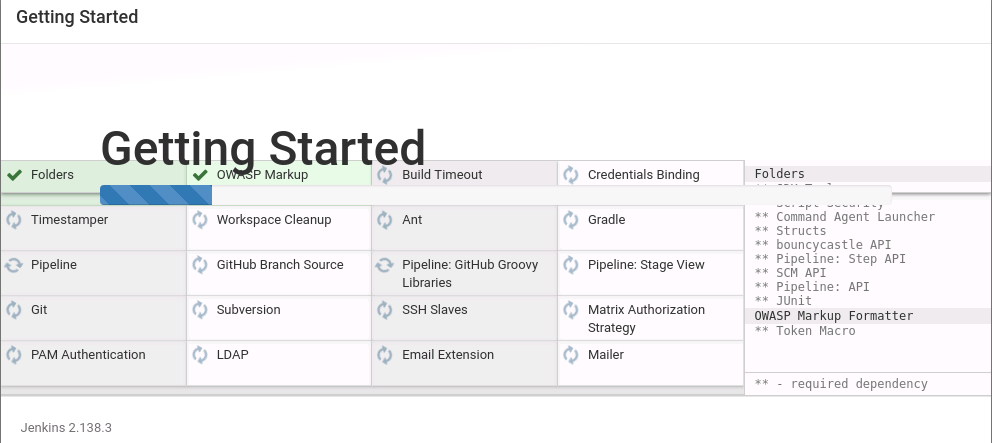
You will need to enter the admin password.

To find the password, type the following command:

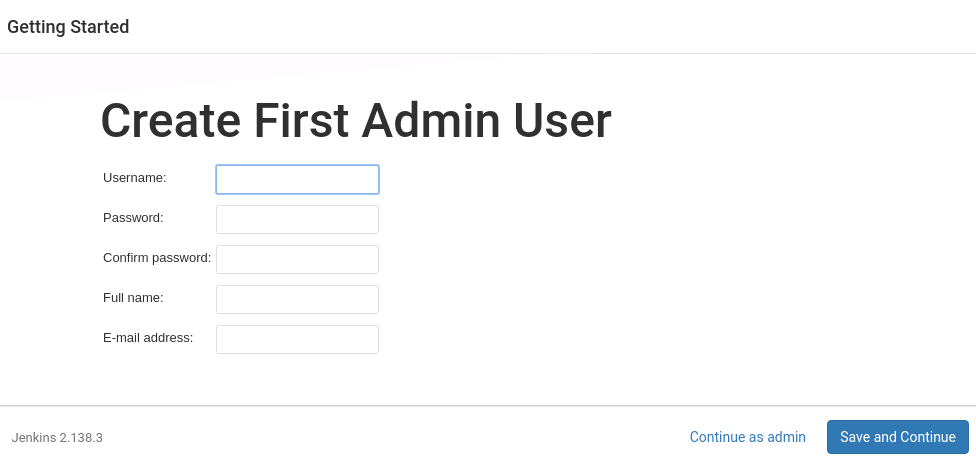
|  |
| --- |
| sudo cat /var/lib/jenkins/secrets/initialAdminPassword |

Use the string of numbers as password. Once you are logged in, you will be redirected to the page below:

 Select **Install suggested plugins**.You will be redirected to the page below:



After installing recommended plugins, you can create “First Admin User” or continue as Admin by filling the required details as shown in the screenshot.



Once that is done, start using Jenkins.

