Project 7 Report

# Application Overview

Diagram

Description automatically generated

# System Requirements

|  |  |
| --- | --- |
| **Server Name** | **Details** |
| Jenkins server | t3.large EC2 instance – as in the assignment |
| Docker Bench for Security | Use an Ubuntu 20.04 LTS VM |
| <<add more system requirements as we add more features>> |  |

# Install Jenkins On Jenkins Server

|  |
| --- |
| # Connect to Jankins Server EC2 instance  ssh -i "ts-Swati.pem" ubuntu@ec2-65-0-120-189.ap-south-1.compute.amazonaws.com  sudo apt update  # install java  sudo apt install openjdk-11-jre-headless  #check if java is installed by checking version  java -version  # install jenkins  # Add the Jenkins repository key to the EC2 instance  wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg  # Append the Debian package repository list to server's source.list  sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'  sudo apt update  # install Jenkins and its dependencies  sudo apt install jenkins  # check the status of Jenkins  sudo systemctl status jenkins.service  # Enable the port 8080 on Jenkins EC2 server  #Goto <<EC2publicIp>>:8080 and login as admin using -- retrieve initial admin password from /var/lib/jenkins/secrets/initialAdminPassword  sudo cat /var/lib/jenkins/secrets/initialAdminPassword    # Install the preselected and add on to it the GitHub pluggin    # Create the first user, after the pluggins are installed, from the UI    # Set the Jenkins URL using private IP    # Create a freestyle job |

# Git hub account details

url: <https://github.com/swatiAga/>

### Generate a github personal access classic token

(A Github Token is required for 1. login to github while pushing changed code to github. 2. Jenkins Github plugin configuration)

username: [swati.mridu@gmail.com](mailto:swati.mridu@gmail.com)

Token generated: ghp\_mOhwzDlJ6uUdJXnmySyZMpORGNBFiI33DScl

# Feature 1 Implementation – Integrate GitHub with Jenkins pipeline

Trigger a jenkins job when code is pushed to github using “GitHub hook trigger for GITScm polling”

## Use case diagram

Diagram

Description automatically generated

## Configure the “GitHub plugin” with github accounts on Jenkins

Go to “Manage Jenkins” –> “Configure System” –> Locate “Github” section and “Add Github Server” and do following modification.

## Install BlueOcean Plugin for Jenkins to get a UI for pipeline visualization

## Configure Jenkins pipeline on which you want push trigger should be effective

Goto Dashboard >> [[project pipeline]] >> Configuration

## Configure “webhook” on github for the specific repository

Use the public ip of the EC2 to configure the webhook on github>> repository>>settings>>webhoot

The private ip does not work.

## Clone the github repo in user’s directory

sudo git clone <https://github.com/swatiAga/project7-sourcecode.git>

## Make changes to readme.md file and push the changes to github

After the git Jenkins Integration -> Make a code change in the working directory -> commit to the git repo and push changes to

git push <https://github.com/swatiAga/project7-sourcecode.git>

## Reference

<https://www.devopsschool.com/blog/how-to-build-when-a-change-is-pushed-to-github-in-jenkins/>

# Feature 2: Docker Build –

## Install the “docker pipeline” plugin for Jenkins

Graphical user interface, text, application, email

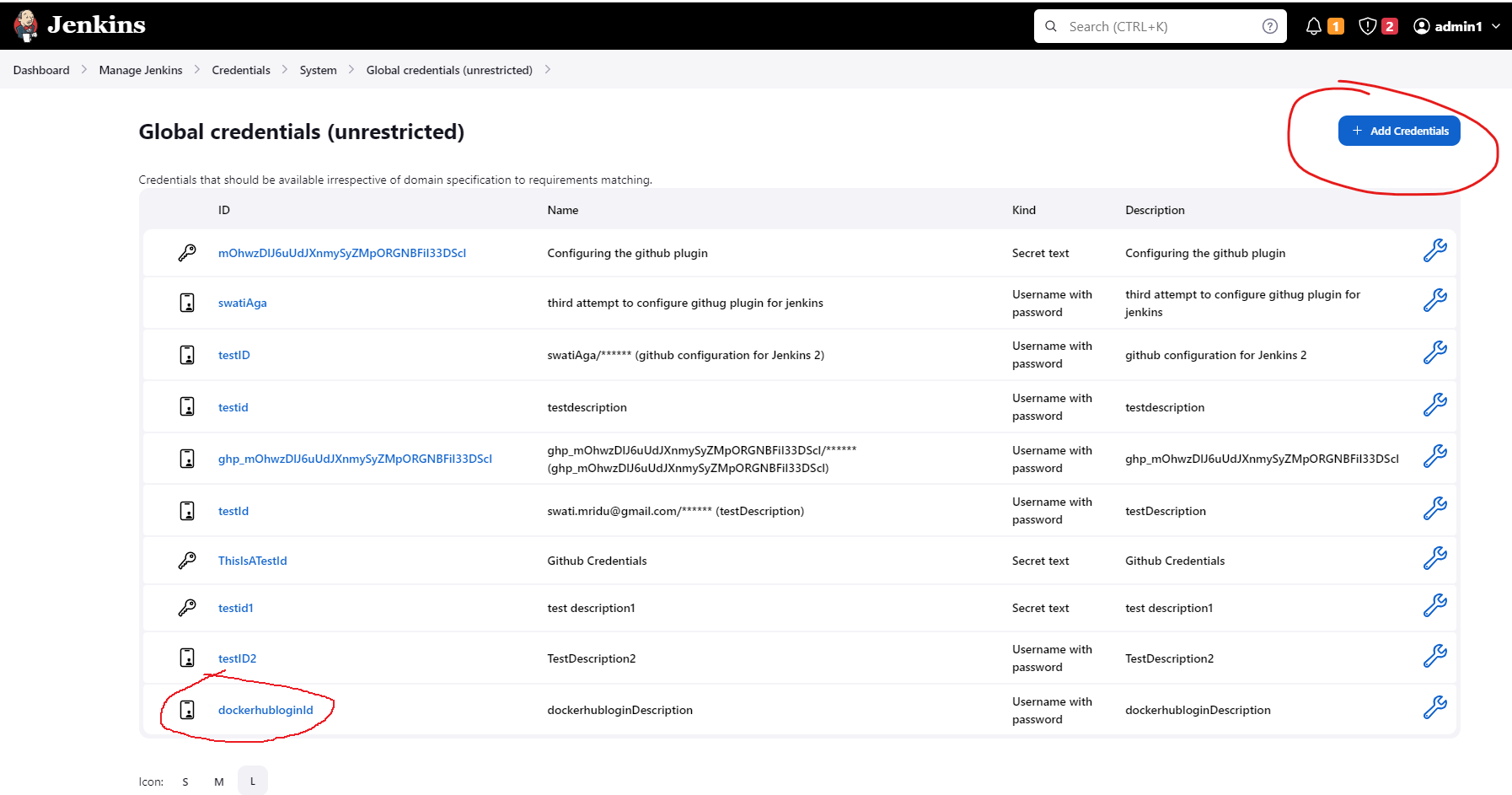
Description automatically generated

## Install “Kubernetes Continuous Deployment” plugin for Jenkins

Download [Kubernetes Continuous Plugin 1.0 version](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa3N1MVdsMUJVbDd3aXpma3FpWUxHY1o2LTE3Z3xBQ3Jtc0trUGFkaFQ5QmhuZlY1bW1Cd2tDX3BFTGZzY3l2WGw5cV9WWkFGTGlXNUpFc2wyRl9janZKV2k1ZkstaUdKd2x0WkZwTDFpR1Y5VDE5bzRUcTc5V3daRldWOHNhZGdnRjZ4ZjNPU1gwOWNlSVFWVG12OA&q=https%3A%2F%2Fupdates.jenkins.io%2Fdownload%2Fplugins%2Fkubernetes-cd%2F1.0.0%2Fkubernetes-cd.hpi&v=XE_mAhxZpwU)

Upload the plugin file “Manage Jenkins >> advanced >> upload file”

## Add DockerHub credentials to Jenkins



## Reference

<https://www.youtube.com/watch?v=XE_mAhxZpwU>

# Possible Improvements

## Jenkins suggested improvement on deploying agent and build server separately.

Graphical user interface, text, application

Description automatically generated

Reference link - <https://www.jenkins.io/doc/book/security/controller-isolation/>

# Tools used

Blue Ocean plugin for Jenkins