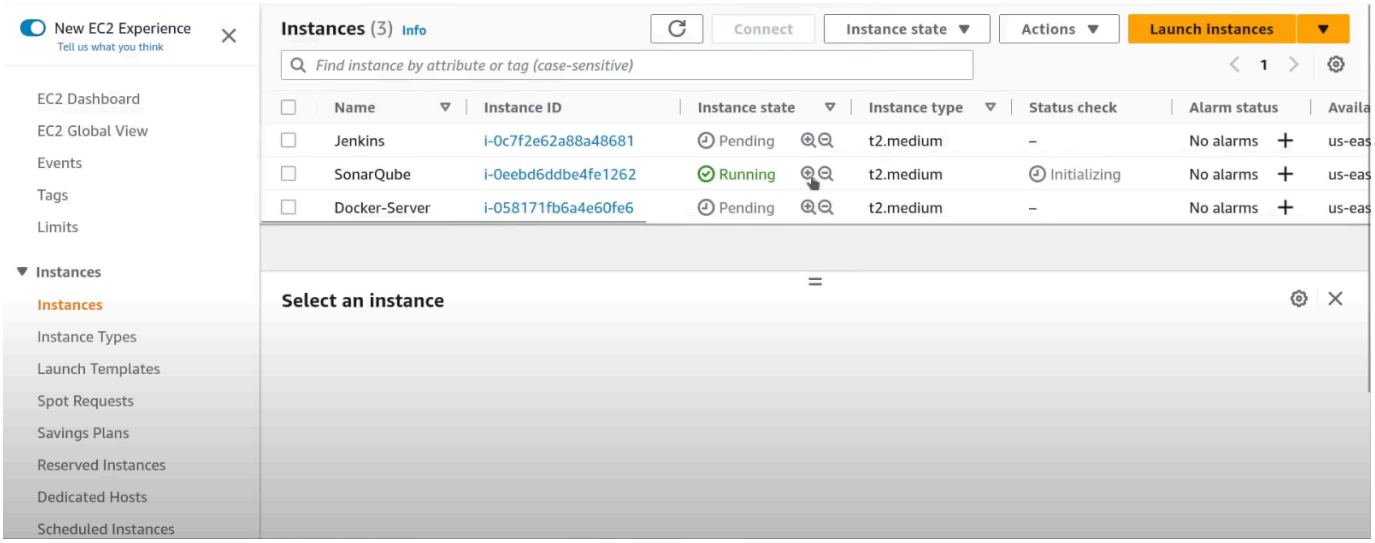
Deploying Code to Github -> Jenkins CI-CD Pipeline On AWS

Using Jenkins/SonarCube/Docker:
- Create a GitHub Repository and push your code to the Repository.

Requirements:
3-EC2 Instances: one for Jenkins, one for SonarCube, and another for Docker.

Go to your Amazon Console:
Launched 1st EC2 Instance -> Chosen Name as Jenkins -> Chosen Ubuntu as Operating System -> Chosen Instance type as t2.micro -> Created a new key pair (helps to SSH into the instance, downloads your SSH key. pem file) -> Launch Instance.
Launched 2nd EC2 Instance -> Chosen Name as SonarCube -> Chosen Ubuntu as Operating System -> Chosen Instance type as t2.medium(it Consumes a lot of memory) -> Used earlier created key pair -> Launch Instance.
Launched 3rd EC2 Instance -> Chosen Name as Docker-Server -> Chosen Ubuntu as Operating System -> Chosen Instance type as t2.micro -> Used earlier created key pair -> Launch Instance.

Output:



SSH Into EC2 Instances:

Jenkins:
Open your terminal and navigate to the SSH-Key-Pair folder. Before SSH, Check for permissions and Give permissions using the below command, if required.
- chmod 400 SSH-Key-Pair.pem
- ssh -i SSH-Key-Pair. pem ubuntu@ip
- hostname change: sudo hostnamectl set-hostname Jenkins

- sudo apt update (It will update)

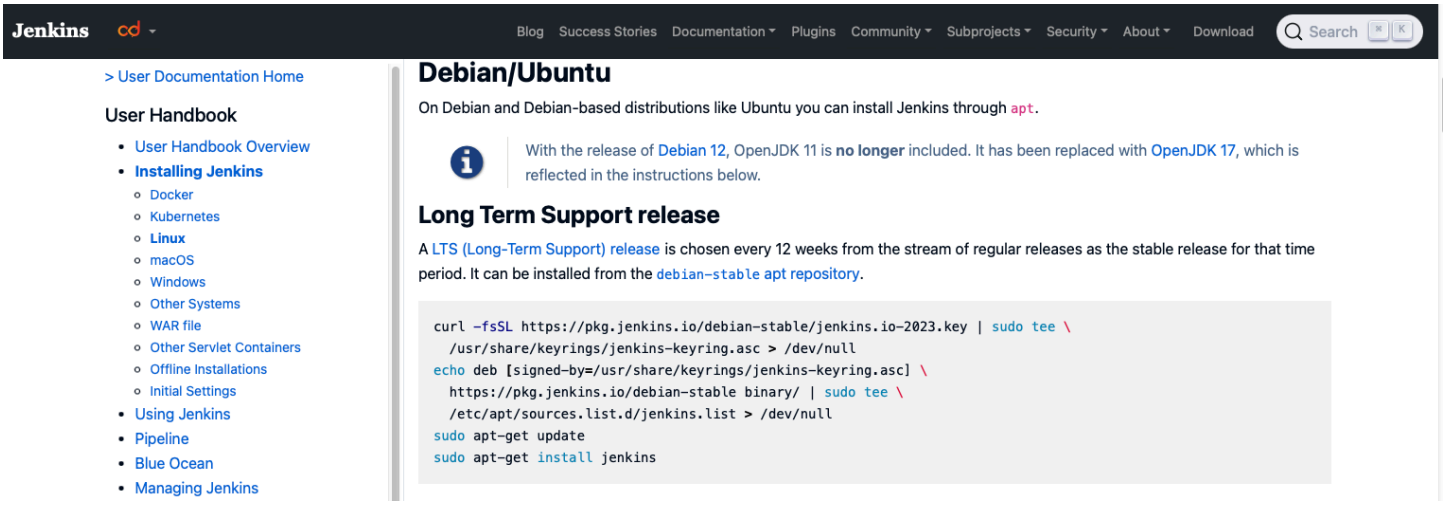
Installing Jenkins:

To install Jenkins, we need to have a Java 11 runtime environment. We can install this using the below command.

- sudo apt install openjdk-11-jre(it will make your environment ready).

Go to the below link(Jenkins.io) and copy the bash command to install Jenkins:

- https://www.jenkins.io/doc/book/installing/linux/#debianubuntu



**- curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee **
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] **
**https://pkg.jenkins.io/debian-stable binary/ | sudo tee **
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins

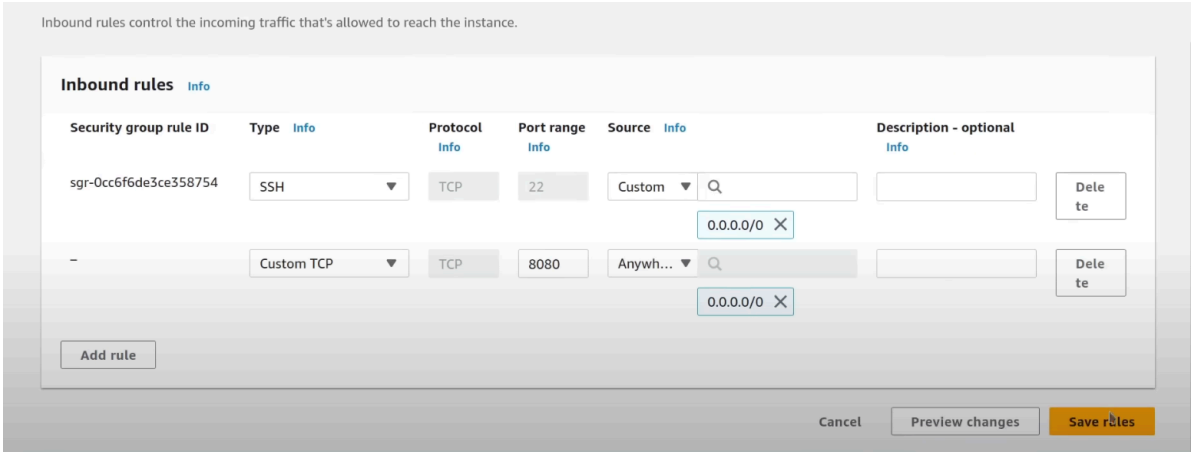
This will install updated Jenkins on your Ec2 Instance.

To verify whether Jenkins is installed or not use the below command

- systemctl status Jenkins

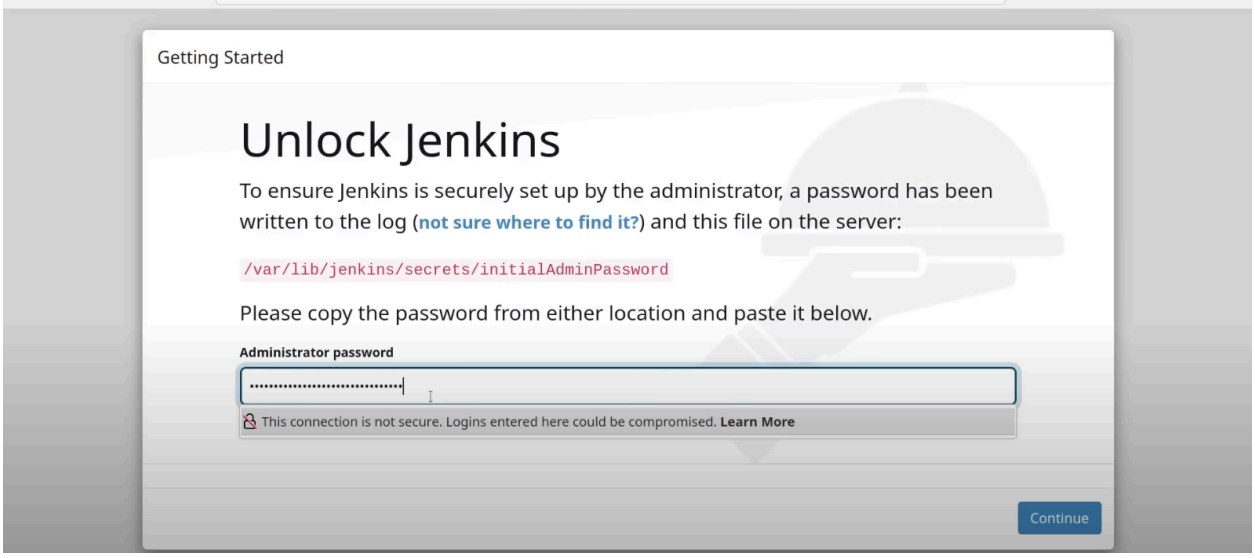
To access this instance, you need to enable default port 8080.

- EC2 instance for Jenkins -> Security Groups -> Edit Inbound Rules -> Add Rule with 8080 as shown below.



- Manage Jenkins -> Manage Plugins -> Search & Install SonarCube Scanner, SSH2 easy plugin.(it will be used later for sonar qube).

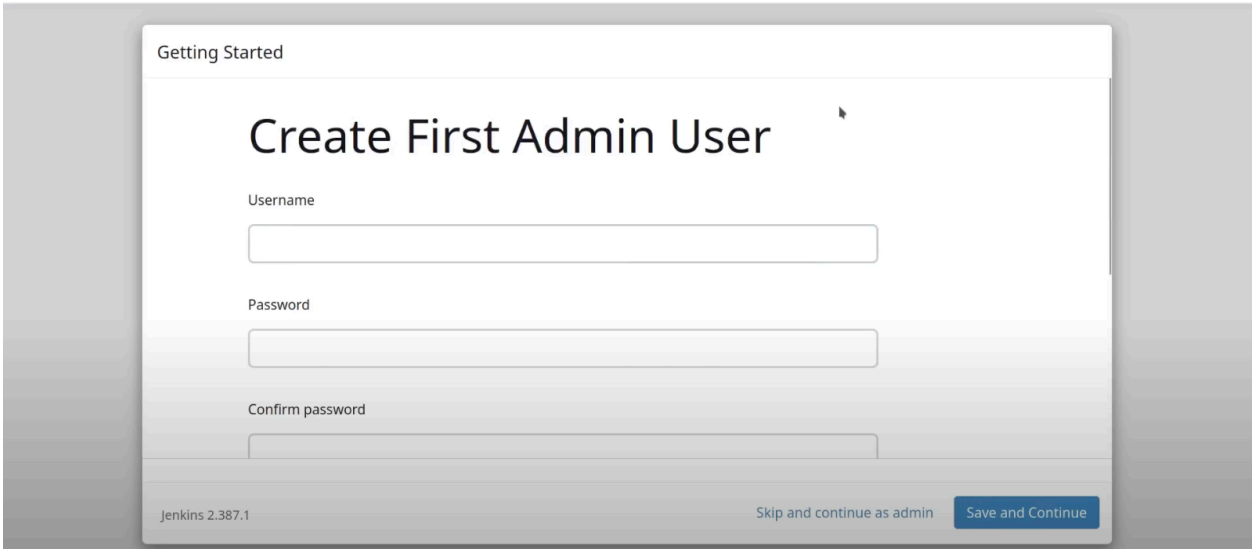
One can copy the IP address of the instance & port, and paste it in the browser. x.x.x.x:8080:



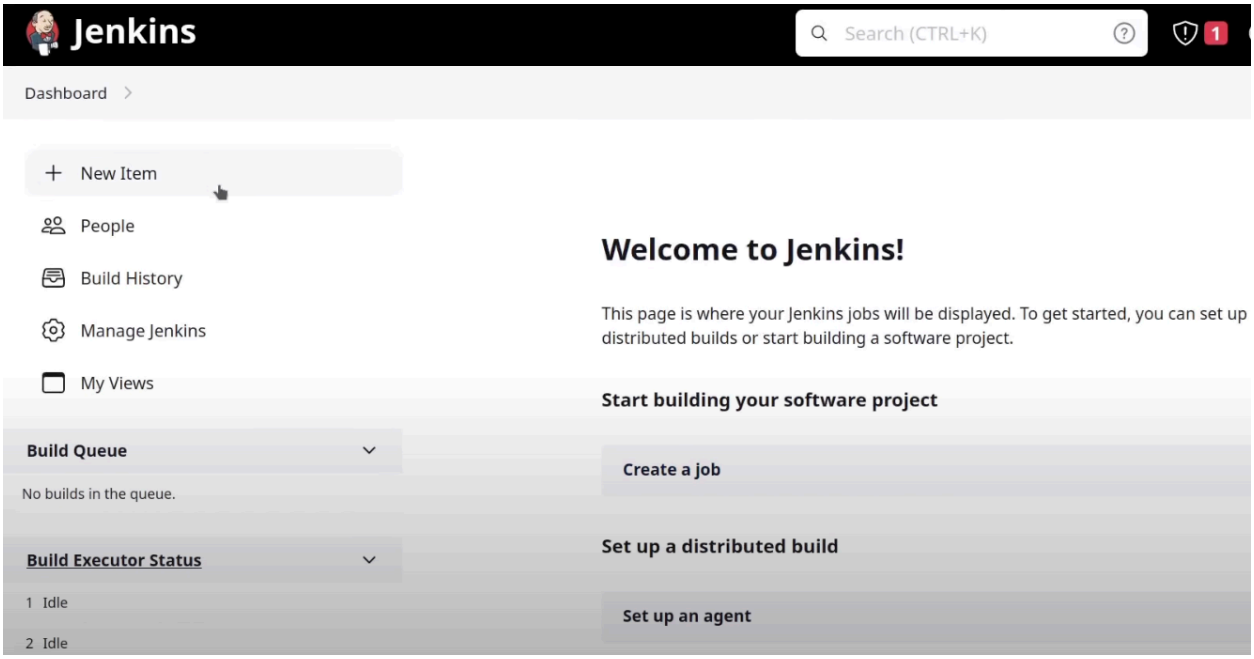
As suggested, navigate through the path and copy the encrypted key/password & paste it here.

- sudo cat /var/lib/jenkins/secrets/initialAdminPassword
- Install all the suggested plugins.

Now Create a User with the necessary details and start Jenkins as shown below:



Jenkins UI:



Create a pipeline:

- New Item -> Name(ex: Automated-Pipeline) -> Select Project(ex:- Freestyle Project).
- Once the project is created -> Source Code Management -> Git -> Give the git details.
- After that go to Configure -> enable GitHub hook triggers for GITScm polling (Enabling hooks triggers whenever there is a change in your git hub repository).
- Go to Repository settings, Enable WebHooks using Jenkins ip: port & push, pull events.

SonarCube:

Open your terminal and navigate to the SSH-Key-Pair folder.

- ssh -i SSH-Key-Pair. pem ubuntu@ip
- hostname change: sudo hostnamectl set-hostname Jenkins
- sudo apt update (It will update)

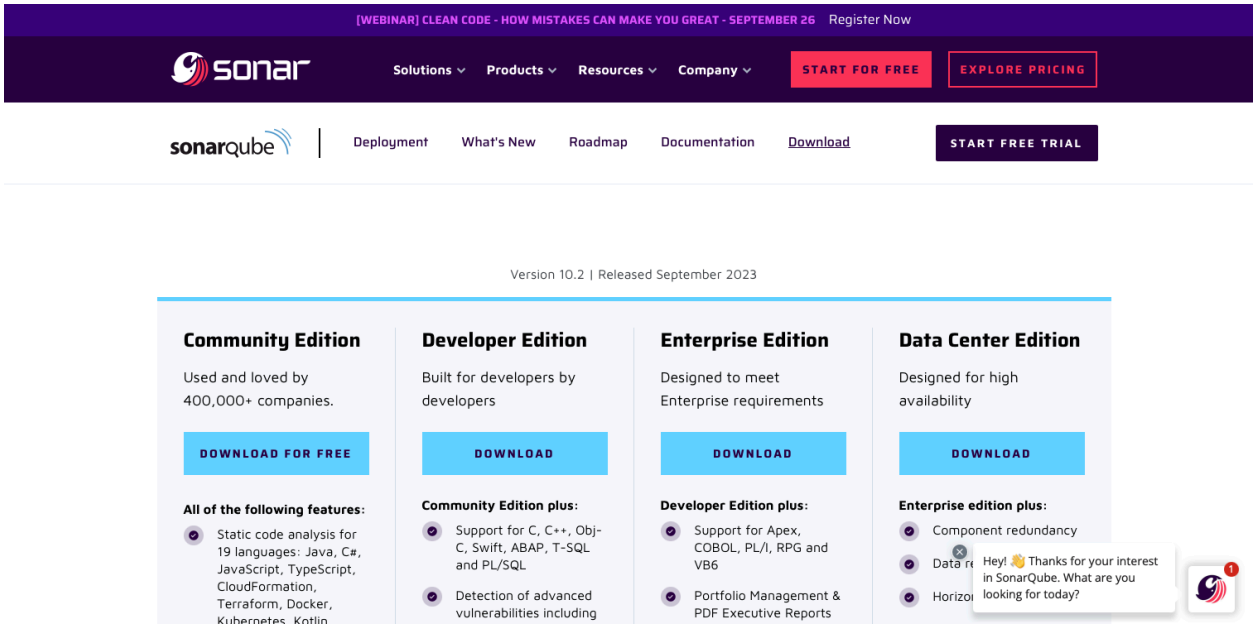
Installing SonarCube:

To install SonarCube, we need to have a Java 17 runtime environment. We can install this using the below command.

- sudo apt install openjdk-17-jre(it will make your environment ready).

Go to the below link(SonarCube) and copy the link to install SonarCube Community edition:

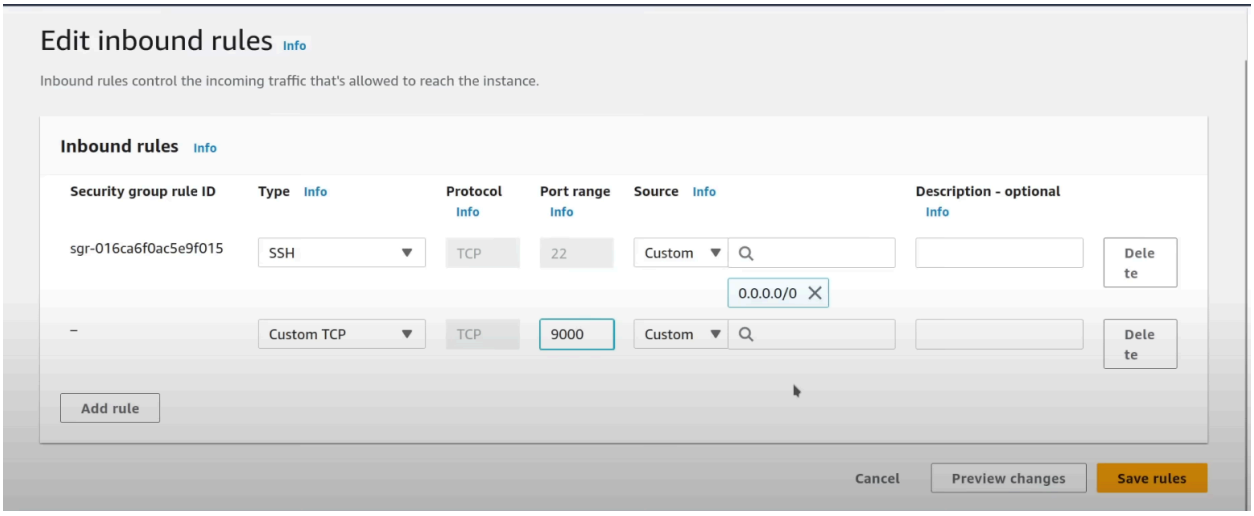
- https://www.sonarsource.com/products/sonarqube/downloads/success-download-community-edition/
- wget url -> will download the sonar cube.
- unzip sonarcube.zip



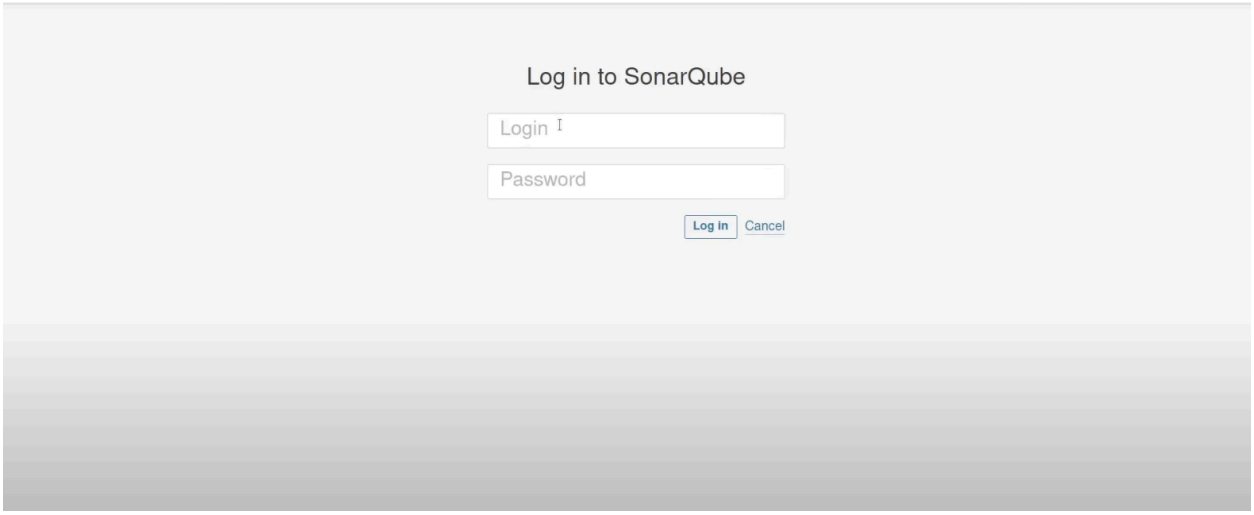
- Go to the bin folder and choose OS to download.
- ./sonar.sh, Console(To see logs of working Sonar).

This will install the updated SonarCube on your Ec2 Instance.
To verify whether SonarCube is installed or not use the below command.
- systemctl status SonarCube.

To access this instance, you need to enable default port 9000.
- EC2 instance for SonarCube -> Security Groups -> Edit Inbound Rules -> Add Rule with 9000 as shown below.



One can copy the IP address of the instance & port, and paste it in the browser. x.x.x.x:9000:
- By default username: admin, password: admin & then create a new password.



Create Manual Project:
- Give the project name, key as you like, and corresponding branch.
- Now Using Continuous Integration CI Jenkins, Choose the DevOps platform as GitHub.
- Configure Analysis -> Continue -> Choose Build based on your requirements(Other).
- In Security, create a SonarCube token.
- Go to Jenkins -> Manage Jenkins -> Global Tool Configuration -> Add Sonar Qube Scanner.
- Go to Configure System of Manage Jenkins -> Add SonarQube Server with IP: Port.
- Try Build Now

DOCKER:

Docker:
Open your terminal and navigate to the SSH-Key-Pair folder. Before SSH, Check for permissions and Give permissions using the below command, if required.
- chmod 400 SSH-Key-Pair.pem
- ssh -i SSH-Key-Pair. pem ubuntu@ip
- hostname change: sudo hostnamectl set-hostname docker
- sudo apt update (It will update)

Installing Docker:

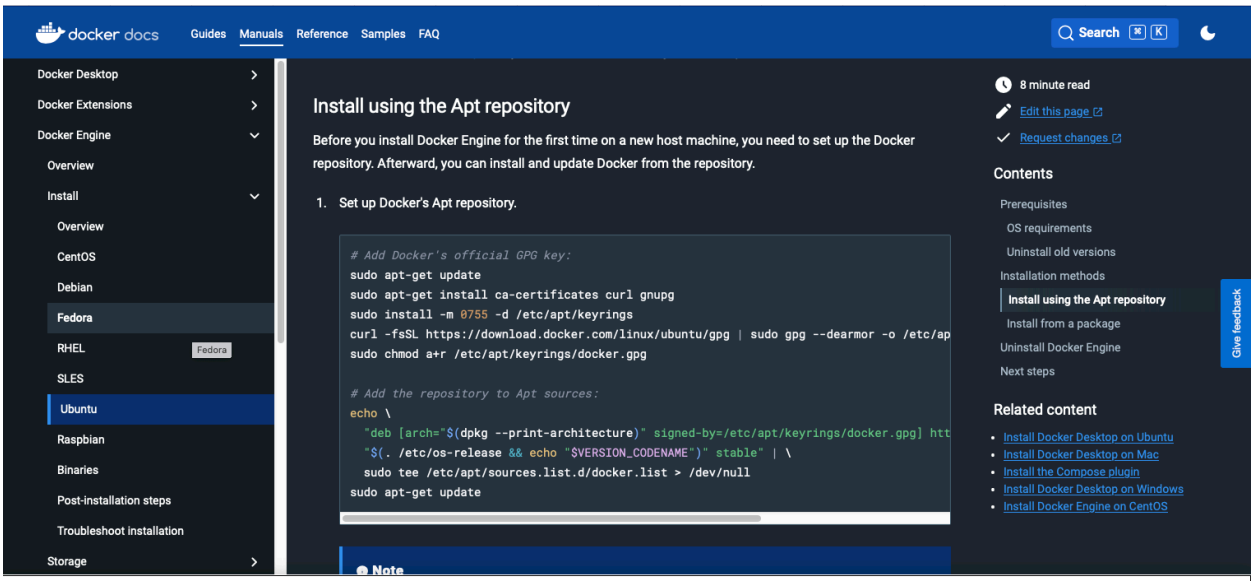
To install Docker, Use the below command.

- Switch to another user: sudo su Jenkins
- # Add Docker's official GPG key:

sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

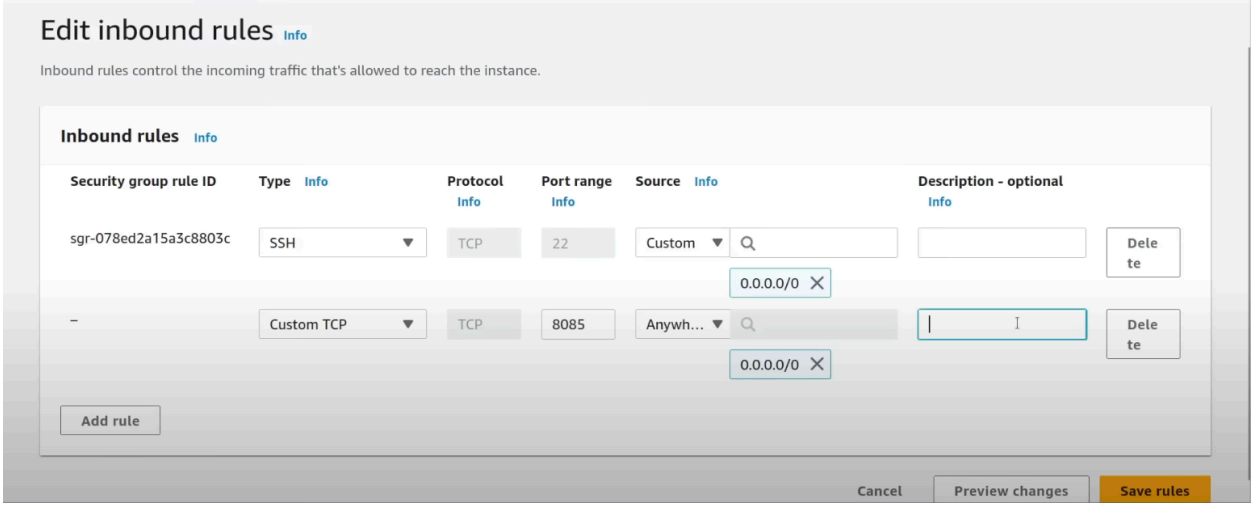
Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

- sudo apt-get install docker-ce docker-ce-cli containerd.io docker-build-plugin docker-compose-plugin.
- Create a password for your docker user. (- passwd ubuntu)

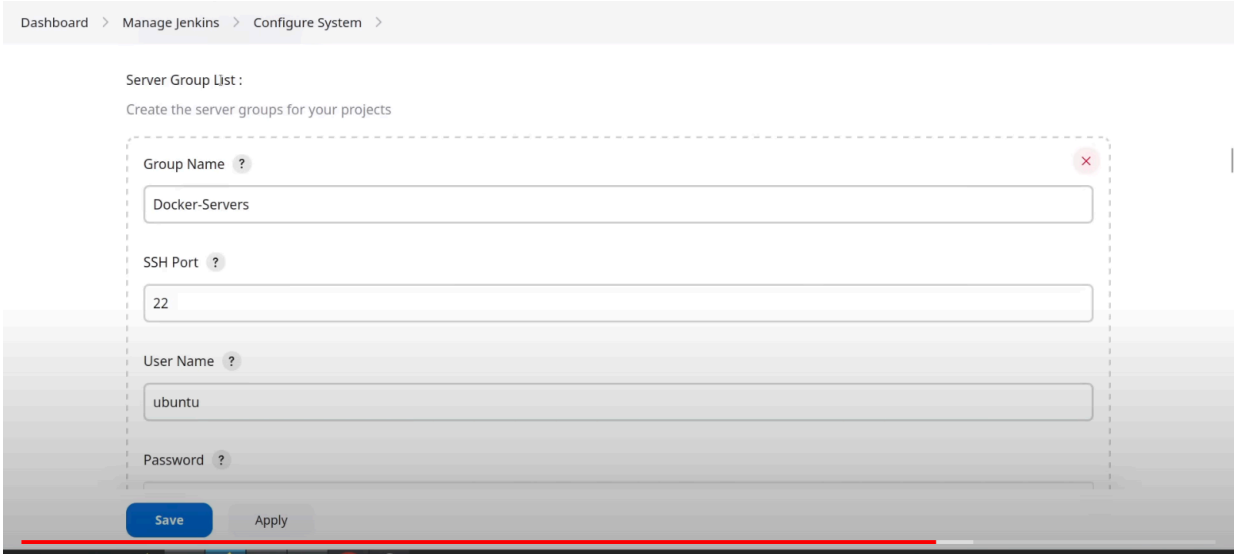


This will install an updated Docker on your Ec2 Instance.
To verify whether Docker is installed or not use the below command
- systemctl status docker

To access this instance and website, you need to enable default port 22, 8085.
- EC2 instance for Docker -> Security Groups -> Edit Inbound Rules -> Add Rule with 22, 8085 as shown below.

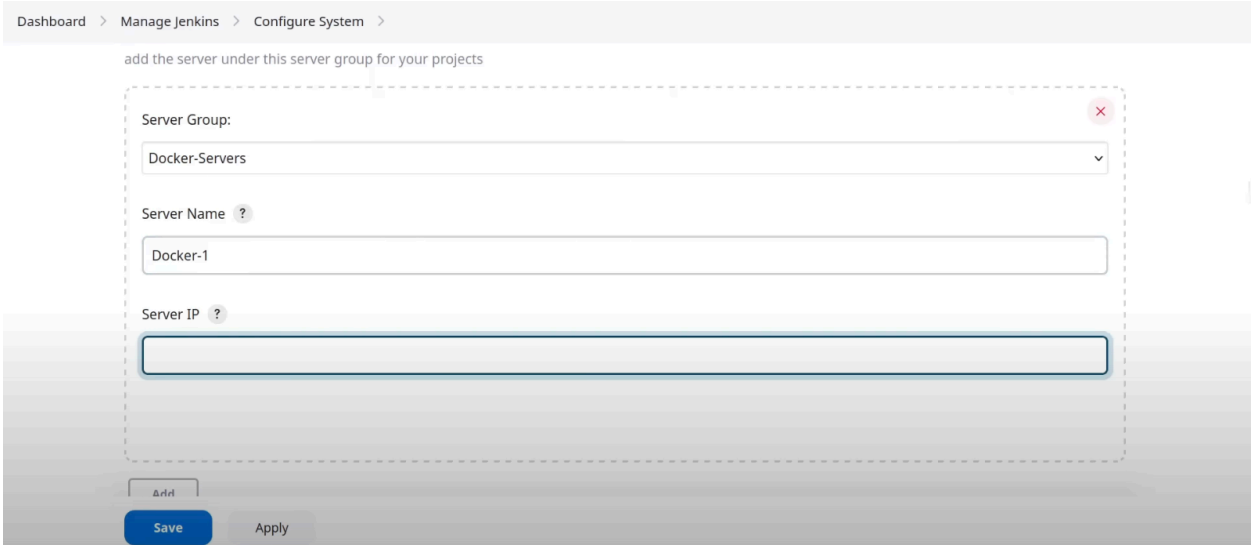


Integrating Jenkins & Docker:
- Check, if you can ssh from Jenkins to Docker.
- Edit sshd config in docker, to give permissions for Jenkins to access.
- nano /etc/ssh/sshd_config
- Make sure your PubkeyAuthentication Yes uncommented & PasswordAuthentication YES
- restart ssd service. systemctl restart sshd.
- Now you can ssh from Jenkins to Docker. (Your in Jenkins Console).
- Generate Keypair public/private (ssh-keygen).
- ssh-copy-id ubuntu@dockerIP from Jenkins Console.



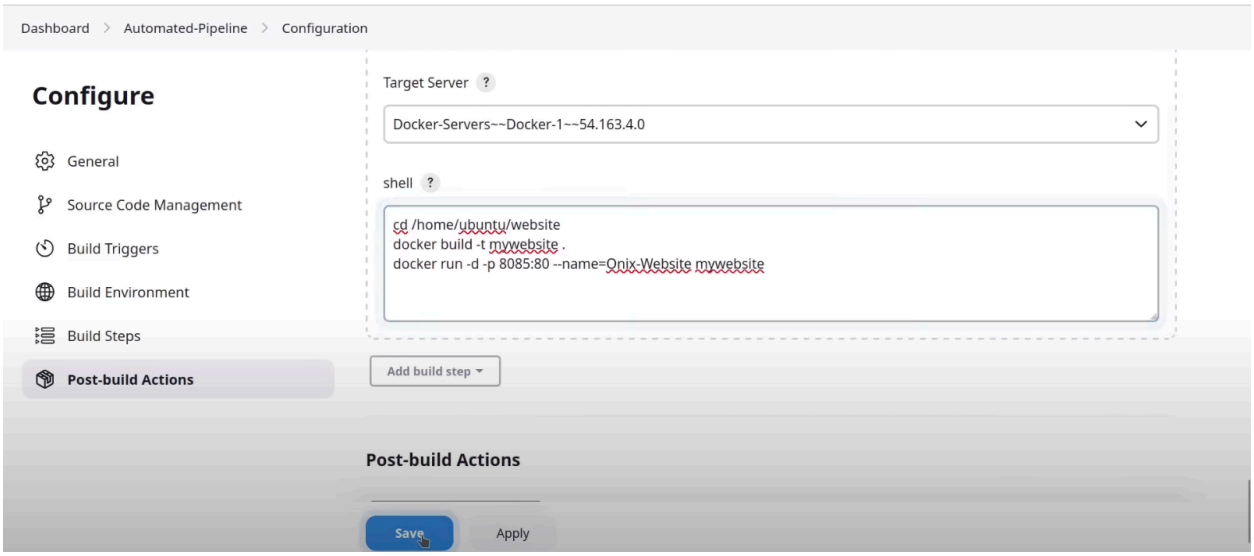
- Now, Jenkins -> Manage jenkins -> configure system -> Under server group center -> create a Docker Server Group(port 22) -> Add the server under server list.

Add Server under Server list:



- Go to Post-build Actions and look for Remote Shell, Select the Docker server and create a file using touch to check whether everything is working correctly or not.
- Add the current user to the docker group, so that it will. Give permissions to Execute Shell.
- sudo usermod -aG docker ubuntu & then newgrp docker & then docker ps should work.

Docker post build Actions:



- Under Excuse shell -> give a path to your directory -> docker build -t name -> docker run as shown above.
- you should able to see your website with dockerIP:8085.