Install JAVA
sudo amazon-linux-extras install java-openjdk11 -y
sudo update-alternatives --config java
vim ~/.bash_profile
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-11.0.11.0.9-1.amzn2.0.1.x86_64/bin/java"
source ~/.bash_profile
echo $JAVA_HOME

Install Jenkins on linux

sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade -y
sudo yum install jenkins java-11-openjdk-devel -y
sudo systemctl daemon-reload

sudo service jenkins start
sudo service jenkins status

---
sudo systemctl status Jenkins
sudo systemctl start jenkins.service
---

sudo yum install httpd -y **(global config error path error maven http error 403)** **(ERROR1)**

Install docker on slave (confirm if it needs to be installed on master)

sudo amazon-linux-extras install docker -y

sudo service docker start

sudo docker info

Enter the public ip of the master Jenkins server
Public_ip_of_master:8080
Get the initial admin password of Jenkins
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Install suggested plugins
Create a jenkins user named master from jenkins dashboard

Install git and maven on master & slave
sudo yum install git -y
sudo yum install maven -y

Repo clone and pushed to github
https://github.com/laxapatiakshaylearning/cicdpipelinejenkins

Setup master slave

Slave setup on master
Manage jenkins → Manage node & clouds → New Node → give node name linux_node1 → permanent

Remote root directory
/home/ec2-user/

Tick on (use websocket) & save

Click on the slave node & right click on agent.jar then copy link address

On slave machine
Download the agent.jar file at /home/ec2-user using wget
wget http://54.198.69.61:8080/jnlpJars/agent.jar

Join from slave using below command
java -jar agent.jar -jnlpUrl http://54.198.69.61:8080/computer/linux_slave1/jenkins-agent.jnlp -secret aa224035b5f55c28c926847bcf9bf60d4cf2dcf75b53be7fb350581a0413b874 -workDir "/home/ec2-user/"

Configure global config
Manage jenkins → Global tool configuration

Jdk
Add jdk
Name: java
Untick install automatically
Java_home =/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64

Git

To find the path of git below is the command
Whereis git
/usr/bin/git **(ERROR1)**

Maven add maven
Name : maven
Untick install automatically

To find the path of maven below is the command
Mvn -v
/usr/share/maven

Name : maven
Path : /usr/share/maven

Click Save

SETUP SONAR ON NEW SERVER

Install java
sudo amazon-linux-extras install java-openjdk11 -y

Check java path

sudo update-alternatives --config java

/usr/lib/jvm/java-11-openjdk-11.0.11.0.9-1.amzn2.0.1.x86_64/bin/java

Set environment variable

vim ~/.bash_profile
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-11.0.11.0.9-1.amzn2.0.1.x86_64/bin/java"

source ~/.bash_profile

Install sonarqube

download binary of sonar

wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.1.0.31237.zip
unzip sonarqube-8.1.0.31237.zip

cd sonarqube-8.1.0.31237/bin/linux-x86-64/
./sonar.sh console

Login using admin as username & password

Navigate to admin logo which is at top right corner → click on my account → security → enter token name : sonaradmin → click on generate & copy token
8f0684f5499a51b6e1f14b2ca7535c90d1e1d0f4

Go to jenkins master server

Install sonar plugin
Managejenkins → manage plugins → available plugins → search plugin

SonarQube Scanner

& install without restart

Manage Jenkins → Configure system

SonarQube servers

Tick

Environment variables Enable injection of SonarQube server configuration as build environment variables

Click Add sonarqube
Name : sonarqube
http://publicip_of_sonarqube:9000
http://34.203.248.145:9000

Add credentials for sonarqube
Add → jenkins → secret text in kind → Secret -- paste the token that was copied from sonar server → Description : sonar token → click on none & select sonar token

Tools Path configuration
Manage jenkins → manage nodes & clouds → click on settings of node → tools location add →

GIT
To find the path of git below is the command
Whereis git (Execute this command on slave node)
/usr/bin/git

JAVA
To find the path of git below is the command
update alternatives command (Execute this command on slave node)
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64

MAVEN
To find the path of maven below is the command
mvn -v (Execute this command on slave node)
/usr/share/maven

Create a freestyle project

Create a job → name for the job : cicdpipeline → Freestyle project

Add the git repo
https://github.com/laxapatiakshaylearning/cicdpipelinejenkins.git
I have not added credentials as the repo is public

Add webhook in the git portal
Go to git web repo name
Settings webhook add webhook
Payload url :
http://public_ip_of_jenkins:8080/github-webhook/
http://54.162.56.96:8080/github-webhook/

Content type
application/json

configure
Let me select individual events
Tick
Branch or tag creation
Commit comments
Forks
Issues
Pull request review comments
Pull request review threads
Pull request reviews
Pull requests
Pushes
Active

Click on add/update webhook
Confirm the dot is replaced by tick, by this we can confirm that the connection is established between github & jenkins master server. The connection is checked by sending a packet to the jenkins server & upon receiving the response from the jenkins server it confirms the connectivity

CLICK on the below tick on master jenkins server

GitHub hook trigger for GITScm polling

Build
Add build step
Invoke top level maven targets

Replace default with maven from drop down menu

Goal
clean install
Save → build now
(check all the components are working properly then add sonar)

Now after checking that the build is successfully build add sonar scanner

Tick on Prepare SonarQube Scanner environment & Modify maven goal to
clean install sonar:sonar

Check if the webhook is triggered from git

Make some changes into the repo and commit

Now check if jenkins job is getting triggered by webhook below is the text that can be observed in the console output of the build number
Started by GitHub push by laxapatiakshaylearning

Check if we are able to see results on sonarqube server as well as on jenkins

Add jacoco plugin

Manage plugins
JaCoCo
Install without restart

Configure on job

Post-build Actions

Record jacoco coverage report

Apply and save → click on build & check if report is generated → coverage report

DOCKER

Add docker plugins

Docker

docker- build-step

Cloudbees

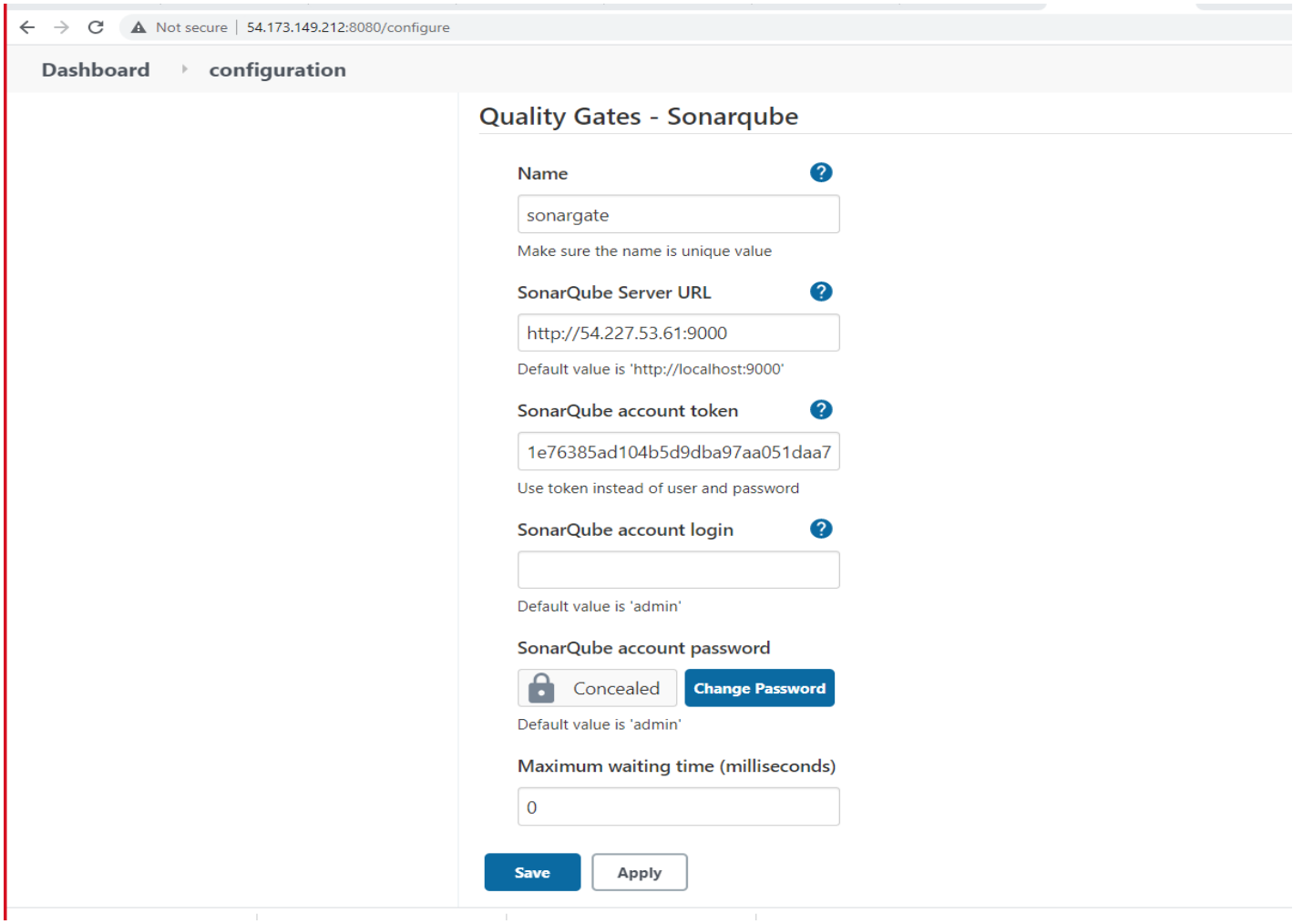
Configure system

Quality Gates - Sonarqube

Name : sonargate
SonarQube Server URL: http://54.227.53.61:9000
SonarQube account token : add token

Username : admin
Password : admin

Minimum & maximum waiting time remove 0



CREATE CUSTOM QUALITY GATE

Quality gate → give name

Select overall code

Coverage 96 (if code coverage is less than 96 percent the gate must fail)

Lines to cover 20 (if the lines

Uncovered line

NOW install docker on linux machines
Install docker on master slave jenkins

sudo amazon-linux-extras install docker -y

sudo service docker start

sudo docker info

Configure project on master jenkins portal

Add build step
Docker build and publish

Create a public docker repo name in docker hub & add below in jenkins
Repo name
laxapatiakshaylearning/cicdnewjenkins

Tag
$BUILD_ID

We build docker image on the jenkins server itself so we do not add server url

Add registry credentials

Username : laxapatiakshaylearning
Password:
Description: docker credentials
Select docker credentials save & build the job

IF we get the docker error as below

[cicdpipeline] $ docker build -t laxapatiakshaylearning/cicdnewjenkins:7 --pull=true /var/lib/jenkins/workspace/cicdpipeline
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=de

fault&pull=1&rm=1&shmsize=0&t=laxapatiakshaylearning%2Fcicdnewjenkins%3A7&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
Build step 'Docker Build and Publish' marked build as failure
Finished: FAILURE

Do the following

Execute the below command on master
sudo chmod 777 /var/run/docker.sock

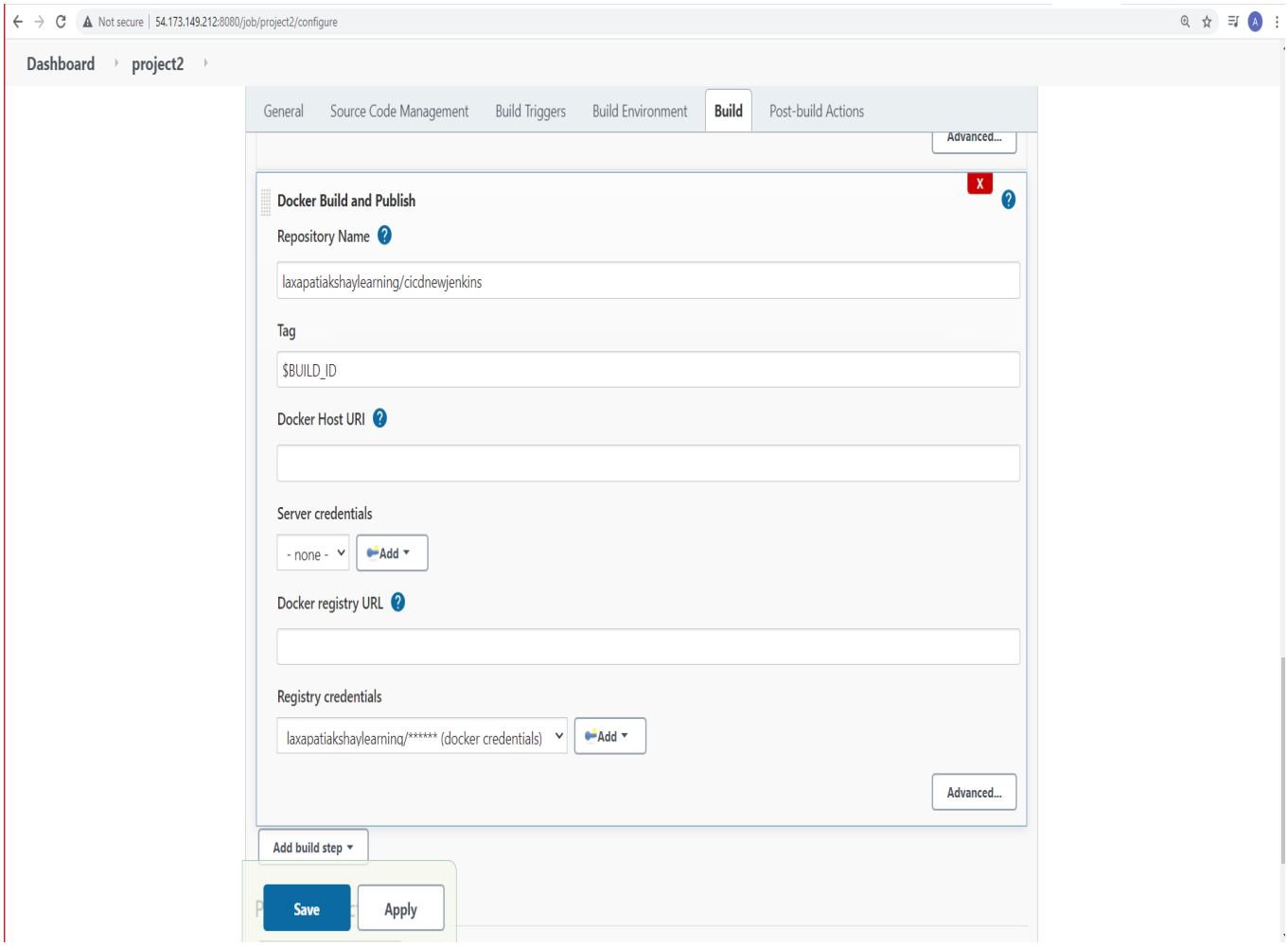
Now we can confirm that the job is running fine on jenkins master & we are able to see image in docker hub
Note : that the job id is appended to the tag of docker hub

Check on docker hub if the repo is pushed with the image id tag of build number

CREATE A JOB 2 named project2 and copy it from project 1

Then edit the projet & add the build step
Build step

Docker build & publish



Configure project on master jenkins portal

Add build step
Docker build and publish

Create a public docker repo name in docker hub & add below in jenkins
Repo name
laxapatiakshaylearning/cicdnewjenkins

Tag
$BUILD_ID

We build docker image on the jenkins server itself so we do not add server url

Add registry credentials

Username : laxapatiakshaylearning
Password:
Description: docker credentials
Select docker credentials save & build the job

TRY TO RUN JOB2 manually & check if the image is pushed

Then edit project 1 & add post build action
Build other projects
Project to build project2
Trigger only if build is stable

Add the default quality gate in sonar & check if project 1 triggers project 2 & check if the image is pushed to docker hub

Then edit the quality gate in sonarqube to the new quality gate and the build must fail

Create a third project for cd job

Execute shell
docker pull laxapatiakshaylearning/cicdnewjenkins:latest
docker run -d --name jenkins laxapatiakshaylearning/cicdnewjenkins:latest

We have to run the cd job manually

Check if we are able to run the job on slave machine

Manage
Restrict where this project can be run
Linux_node1

Connect slave to master
java -jar agent.jar -jnlpUrl http://54.162.56.96:8080/computer/linux_node1/jenkins-agent.jnlp -secret a9d897b2885c1db922ddfd27d29d3a618b08226c9612e6a514116e04e2efddfb -workDir "/home/ec2-user/"

Build the job

Confirm that we get the same error that we got for master:

[cicdpipeline] $ docker build -t laxapatiakshaylearning/cicdnewjenkins:9 --pull=true /home/ec2-user/workspace/cicdpipeline

Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B

%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&pull=1&rm=1&shmsize=0&t=laxapatiakshaylearning%2Fcicdnewjenkins%3A9&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
Build step 'Docker Build and Publish' marked build as failure

Finished: FAILURE

Stop the slave connection & run the below command on slave
sudo chmod 777 /var/run/docker.sock

Build & try if we are able to build the job, if not follow the below steps

Restart the linux slave server (reboot) → after reboot sudo service docker status → sudo service docker start → sudo chmod 777 /var/run/docker.sock → and then reconnect the slave to master → build the project again & check if its executed properly

Click on Download now & install after restart → restart

NOW install docker on linux machines
Install docker on master slave jenkins

sudo amazon-linux-extras install docker -y

sudo service docker start

sudo docker info

Configure project on master jenkins portal

Add build step
Docker build and publish

Create a public docker repo name in docker hub & add below in jenkins
Repo name
laxapatiakshaylearning/cicdnewjenkins

Tag
$BUILD_ID

We build docker image on the jenkins server itself so we do not add server url

Add registry credentials

Username : laxapatiakshaylearning
Password:
Description: docker credentials
Select docker credentials save & build the job

IF we get the docker error as below

[cicdpipeline] $ docker build -t laxapatiakshaylearning/cicdnewjenkins:7 --pull=true /var/lib/jenkins/workspace/cicdpipeline
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&pull=1&rm=1&shmsize=0&t=laxapatiakshaylearning%2Fcicdnewjenkins%3A7&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
Build step 'Docker Build and Publish' marked build as failure
Finished: FAILURE

Do the following

Execute the below command on master
sudo chmod 777 /var/run/docker.sock

Now we can confirm that the job is running fine on jenkins master & we are able to see image in docker hub
Note : that the job id is appended to the tag of docker hub

Check if we are able to run the job on slave machine

Manage
Restrict where this project can be run
Linux_node1

Connect slave to master
java -jar agent.jar -jnlpUrl http://54.162.56.96:8080/computer/linux_node1/jenkins-agent.jnlp -secret a9d897b2885c1db922ddfd27d29d3a618b08226c9612e6a514116e04e2efddfb -workDir "/home/ec2-user/"

Build the job

Confirm that we get the same error that we got for master:

[cicdpipeline] $ docker build -t laxapatiakshaylearning/cicdnewjenkins:9 --pull=true /home/ec2-user/workspace/cicdpipeline

Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&pull=1&rm=1&shmsize=0&t=laxapatiakshaylearning%2Fcicdnewjenkins%3A9&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied Build step 'Docker Build and Publish' marked build as failure

Finished: FAILURE

Stop the slave connection & run the below command on slave
sudo chmod 777 /var/run/docker.sock

Build & try if we are able to build the job, if not follow the below steps

Restart the linux slave server (reboot) → after reboot sudo service docker status → sudo service docker start → sudo chmod 777 /var/run/docker.sock → and then reconnect the slave to master → build the project again & check if its executed properly

BUILD SHOULD FAIL AFTER THE QUALITY GATE FAILS:

Install plugin on jenkins master
Sonar Quality Gates Plugin

Go the project build and manage
Sonar Quality Gates Plugin

Post-build Actions
Project Key: com.example:java-maven-junit-helloworld
Job status when analysis fails: FAILED

##
BELOW TASK IS PENDING AS AFTER FAILING ALSO THE BUILD IS EXECUTED

Now check the code coverage fails by adding quality gate

To add quality gate go to sonar dashboard

Quality gates---> create ---> add rules ----> all overall code 96 percent set

And the code which we have used has code coverage 94.6

It is showing failed in sonar but the build is getting passed check this
Show on jenkins dashboard that it is getting fail but the job is run successfully (check this)

Tried adding webhook at sonarqube

http://54.162.56.96:8080/sonarqube-webhook/

Install plugins