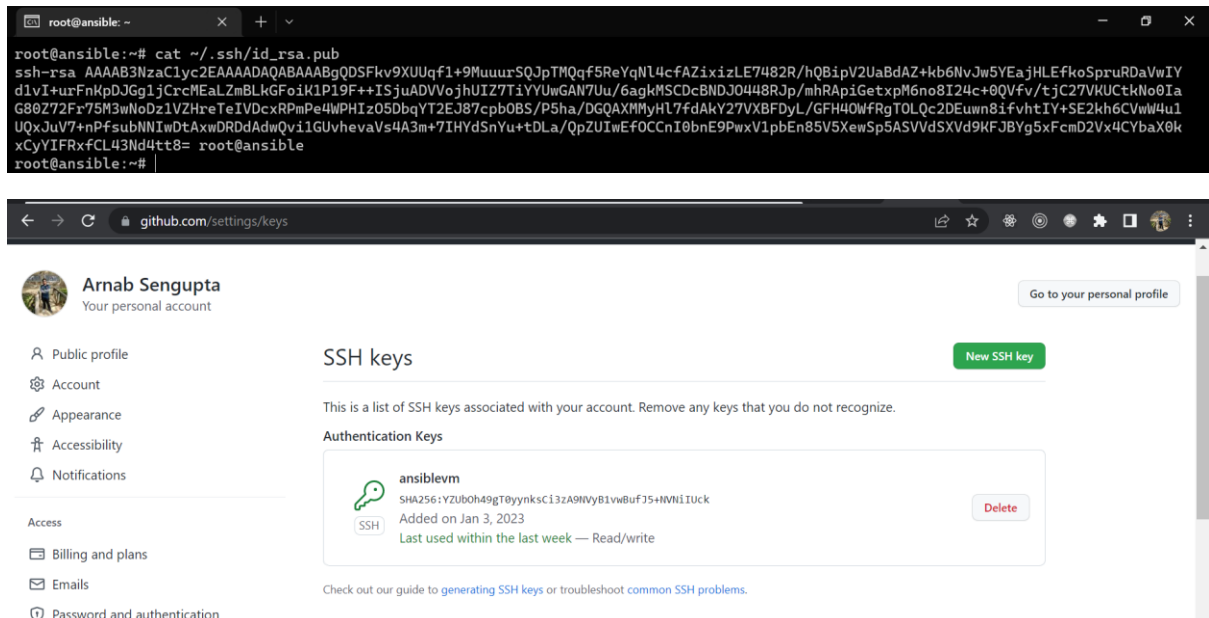


Capstone Project

1. Building an entire CI/CD pipeline in Jenkins

Jenkins-CI-Pipeline

a. generating SSH key and uploading it to GitHub



b. Create a Jenkins pipeline – integrate the private repo with Jenkins that will be first stage

c. Second stage compile the project using maven but exclude test

d. Third stage do code testing using unittest checkstyle codecoverage

e. Add all this stages in the pipeline

f. Then integrate Jenkins with SonarQube. Provide the difference between sonarqube testing and the previous testing framework

g. Finally upload the jar file inside nexus

h. While uploading jar file pipeline syntax generator provide you with static value.

i. Autmatically find out the jar file and upload the same

Jenkins-Cl-Pipeline

```
pipeline {
    agent any
    stages {
        stage('Poll Code Repository') {
            steps {
                git credentialsId: 'git-key', url:
'git@github.com:arnabcs10/springboot-chat-app.git'
            }
        }
        stage('Build') {
            steps {
                sh 'mvn -B -DskipTest clean package'
            }
        }
        stage('Test') {
            steps {
                junit 'target/surefire-reports/*.xml'
            }
        }
        stage('Checkstyle') {
            steps {
                sh 'mvn checkstyle:checkstyle'
                recordIssues(tools: [checkStyle(pattern: '**/checkstyle-
result.xml')])
            }
        }
        stage('Code Coverage') {
            steps {
                jacoco()
            }
        }
        stage('Sonarqube Analysis'){
            steps {
                sh 'mvn clean verify sonar:sonar \
-Dsonar.projectKey=springboot-chat-app \
-Dsonar.host.url=http://20.185.62.113:9000 \

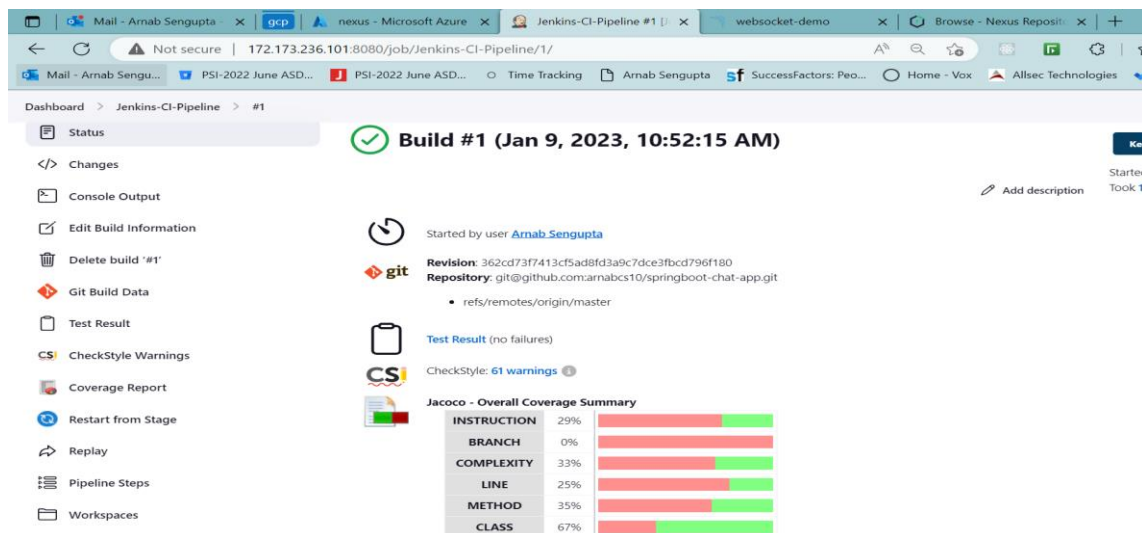
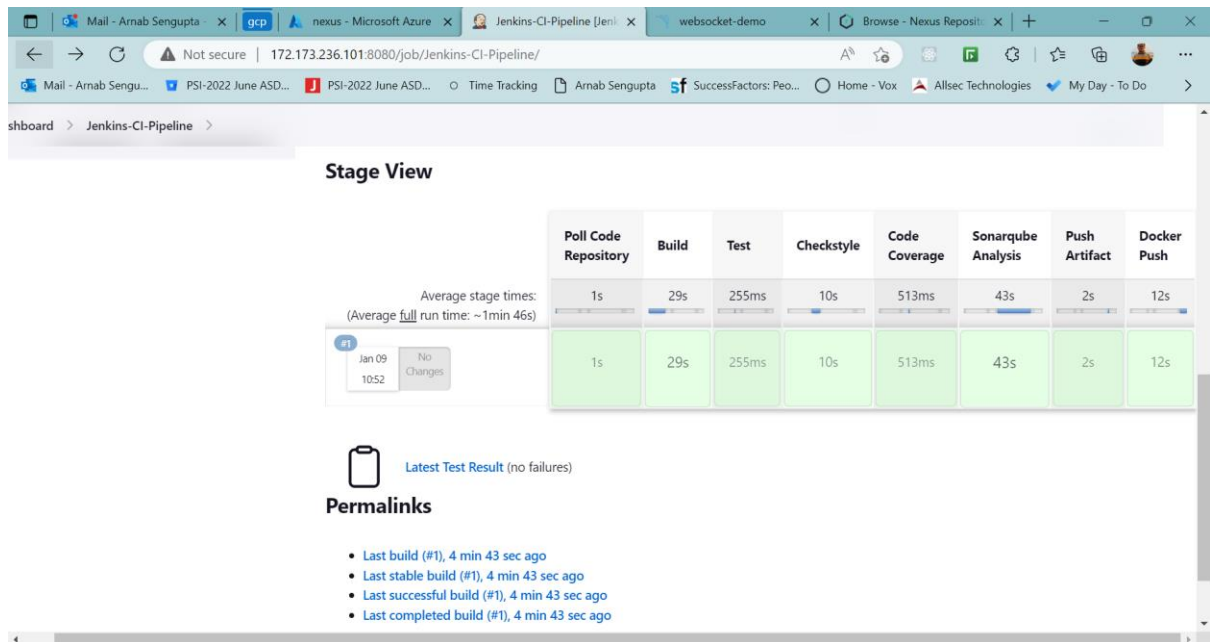
```

```

-
Dsonar.login=sqp_ea1ed8fedff3c8dc8a5fb91351068cea2769aa91'
    }
  }
  stage('Push Artifact'){
    steps{
      script {
        pom = readMavenPom file: "pom.xml";
        filesByGlob = findFiles(glob:
"target/*.${pom.packaging}");
        echo "${filesByGlob[0].name}
${filesByGlob[0].path} ${filesByGlob[0].directory}
${filesByGlob[0].length} ${filesByGlob[0].lastModified}"
        artifactPath = filesByGlob[0].path;
      }
      nexusArtifactUploader artifacts: [[artifactId:
pom.artifactId, classifier: '', file: artifactPath, type: pom.packaging,
type: 'war']], credentialsId: 'nexus-cred', groupId: pom.artifactId,
nexusUrl: '40.117.186.85:8081', nexusVersion: 'nexus3', protocol: 'http',
repository: 'maven-snapshots', version: pom.version
    }
  }
  stage('Docker Push') {
    steps {
      script {
        withDockerRegistry(credentialsId: 'nexus-cred',
url: 'http://40.117.186.85:8085') {
          // some block
          sh 'mvn compile jib:build -
Djib.allowInsecureRegistries=true -DsendCredentialsOverHttp'
        }
      }
    }
  }
}

```

Stages in pipeline



Jacoc Coverage Report

Overall Coverage Summary

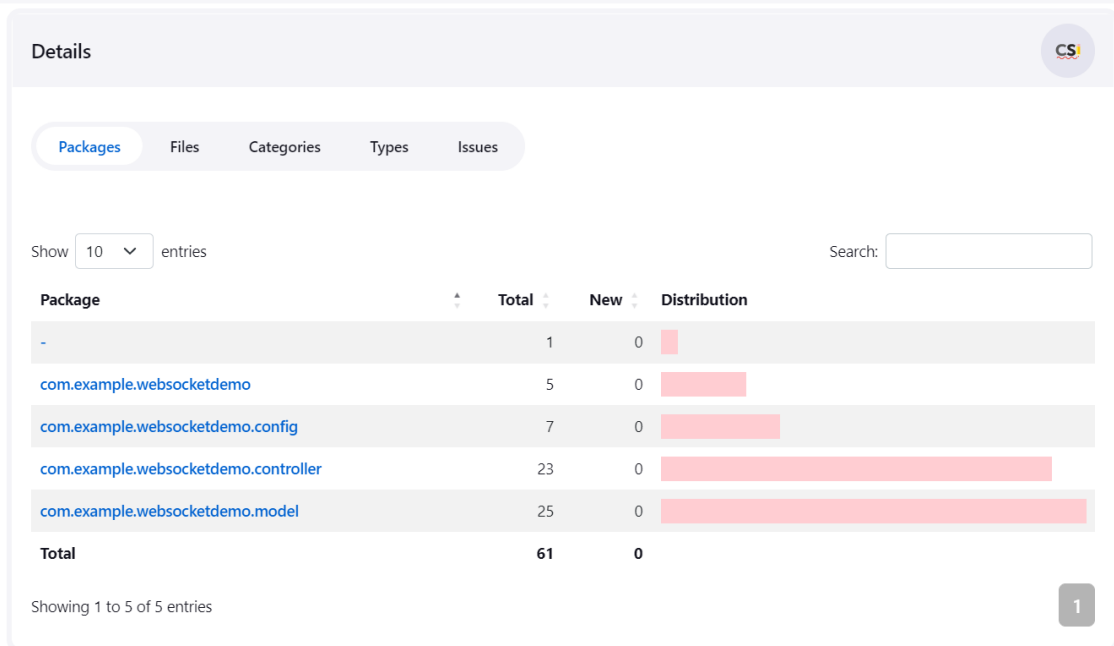
name	instruction	branch	complexity	line	method	class
all classes	29% M: 110 C: 46	0% M: 2 C: 0	33% M: 14 C: 7	25% M: 30 C: 10	35% M: 13 C: 7	67% M: 2 C: 4

Coverage Breakdown by Package

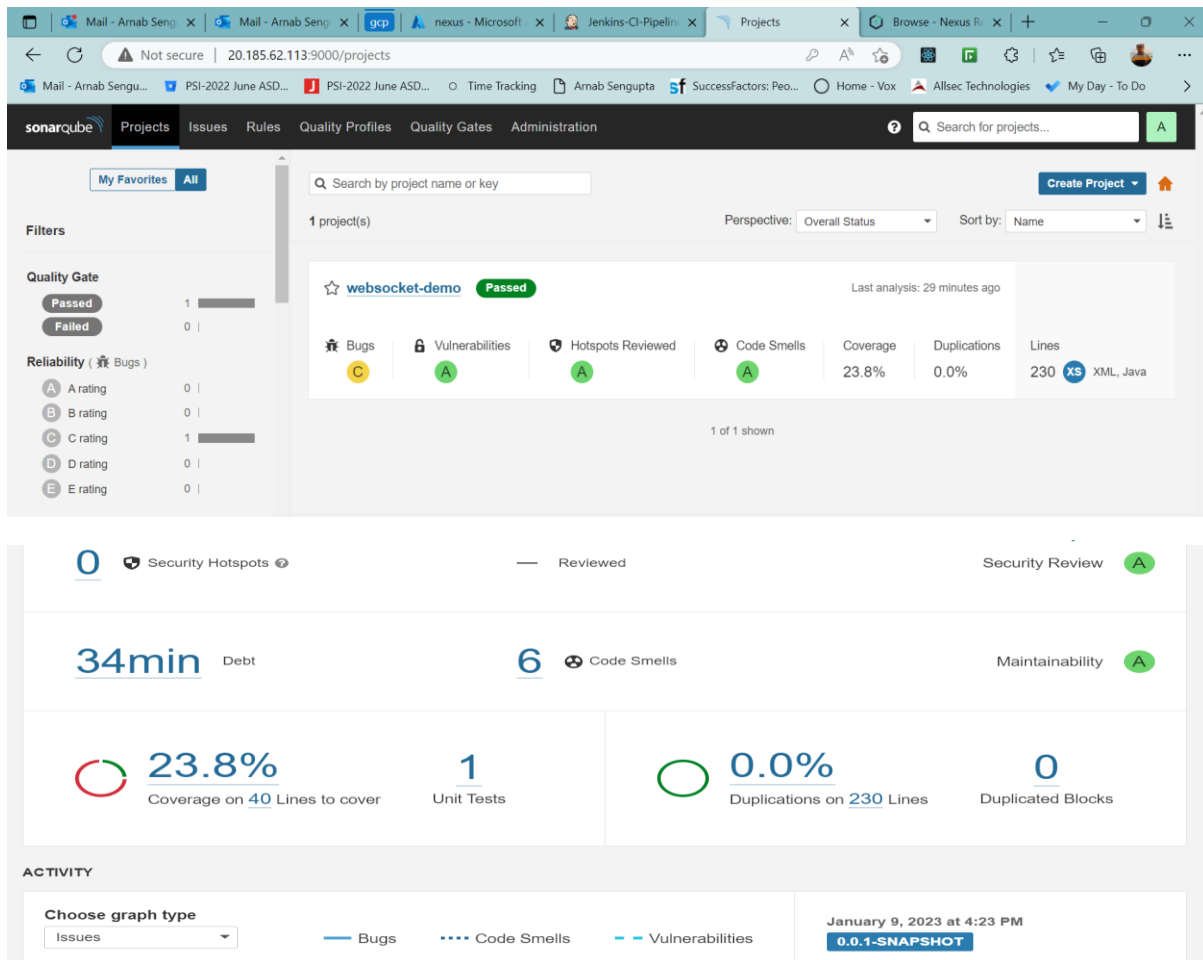
name	instruction	branch	complexity	line	method	class
com.example.websocketdemo	M: 5 C: 3 38%	M: 0 C: 0 100%	M: 1 C: 1 50%	M: 2 C: 1 33%	M: 1 C: 1 50%	M: 0 C: 1 100%
com.example.websocketdemo.config	M: 0 C: 33 100%	M: 0 C: 0 100%	M: 0 C: 3 100%	M: 0 C: 6 100%	M: 0 C: 3 100%	M: 0 C: 1 100%
com.example.websocketdemo.controller	M: 47 C: 10 18%	M: 2 C: 0 0%	M: 5 C: 3 38%	M: 14 C: 3 18%	M: 4 C: 3 43%	M: 0 C: 2 100%
com.example.websocketdemo.model	M: 58 C: 0 0%	M: 0 C: 0 100%	M: 8 C: 0 0%	M: 14 C: 0 0%	M: 8 C: 0 0%	M: 2 C: 0 0%

CheckStyle Report

CheckStyle Warnings



SonarQube Report



Nexus Artifact Registry

The screenshot shows the Sonatype Nexus Repository Manager OSS 3.45.0-01 interface. The left sidebar contains a 'Browse' menu with options: Welcome, Search, Browse (selected), and Upload. The main content area is titled 'Browse' and 'Browse assets and components'. It displays a table of repository components:

Name	Type	Format	Status	URL	Health check	IQ Policy Vi...
docker-images	hosted	docker	Online	copy	Health check	IQ Policy Vi...
maven-central	proxy	maven2	Online - Ready to ...	copy	Analyze	IQ Policy Vi...
maven-public	group	maven2	Online	copy	Health check	IQ Policy Vi...
maven-releases	hosted	maven2	Online	copy	Health check	IQ Policy Vi...
maven-snapshots	hosted	maven2	Online	copy	Health check	IQ Policy Vi...

Below the table, the 'Browse' page for 'maven-snapshots' is shown. It displays a tree view of the repository structure, including 'websocket-demo' and '0.0.1-SNAPSHOT'.

Docker Image

The screenshot shows the Sonatype Nexus Repository Manager OSS 3.45.0-01 interface. The left sidebar contains a 'Browse' menu with options: Welcome, Search, Browse (selected), and Upload. The main content area is titled 'Browse' and 'Browse assets and components'. It displays a table of repository components:

Name	Type	Format	Status	URL	Health check	IQ Policy Vi...
docker-images	hosted	docker	Online	copy	Health check	IQ Policy Vi...
maven-central	proxy	maven2	Online - Ready to ...	copy	Analyze	IQ Policy Vi...
maven-public	group	maven2	Online	copy	Health check	IQ Policy Vi...
maven-releases	hosted	maven2	Online	copy	Health check	IQ Policy Vi...
maven-snapshots	hosted	maven2	Online	copy	Health check	IQ Policy Vi...

Below the table, the 'Browse' page for 'docker-images' is shown. It displays a tree view of the repository structure, including 'v2' and 'blobs'. A detailed view of the 'v2/chatapp/manifests/sha256:2c392d1fda...' asset is shown on the right, including a 'Delete asset' button and a table of asset details:

Property	Value
Last downloaded	has not been downlo aded
Locally cached	true
Blob reference	default@24FF1460-94 949C75-084E2653-F 2D3AC53-91ED4897: 4dc1da4e-124a-4c9e- 85ea-8cf5eb23a5e3
Containing repo	docker-images
Uploader	admin
Uploader's IP Address	172.173.236.101

Jenkins CD-pipeline

a. Create an ansible playbook which install docker in Ubuntu system

Playbook for installing Docker in Ubuntu system

```
---
- name: Install Docker in Ubuntu system
  hosts: all
  become: true
  tasks:
    - name: Update and upgrade apt packages
      apt:
        upgrade: yes
        update_cache: yes
    - name: Install required system packages
      apt:
        pkg:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: Add Docker GPG apt Key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker Repository
      apt_repository:
        repo: 'deb https://download.docker.com/linux/ubuntu
bionic stable'
        state: present
```

```

- name: Update apt and install docker-ce
  apt:
    name: docker-ce
    state: latest
    update_cache: yes

- name: Install Docker Module for Python
  pip:
    name: docker

- name: Allow insecure registries
  copy:
    dest: /etc/docker/daemon.json
    content: |-
      {
        "insecure-registries" : ["40.117.186.85:8085"]
      }

- name: Enable the docker service and start
  service:
    name: docker
    enabled: yes
    state: restarted

- name: Check if docker is Installed
  shell:
    docker -v

```

b. Create another ansible playbook which will deploy the private image from nexus repo

Playbook for deploying docker image, pulling from nexus private repository

```

---
- name: Deploy Docker Image
  hosts: all
  become: true
  vars_files:

```



```

- cred.yaml
tasks:
  - name: Log into private registry and force re-
authorization
    docker_login:
      registry: 40.117.186.85:8085
      username: "{{ username }}"
      password: "{{ password }}"
      reauthorize: yes

  - name: Running Docker Container
    docker_container:
      name: chatapp
      image: 40.117.186.85:8085/chatapp
      state: started
      pull: yes
      ports:
        - "8080:8080"

  - name: Connect to app server on port 8080 and check
status 200 - Try 5 times
    tags: test
    uri:
      url: http://localhost:8080
    register: result
    until: "result.status == 200"
    retries: 5
    delay: 10

```

Encrypt nexus credentials in cred.yaml file, which will be used as authentication while pulling docker image from private nexus registry

```

root@ansible:~/springboot-chat-app/ansible# ansible-vault create cred.yaml
New Vault password:
Confirm New Vault password:
root@ansible:~/springboot-chat-app/ansible# cat cred.yaml
$ANSIBLE_VAULT;1.1;AES256
33386261653866646361643931643730653037386561393338343236316230356565326434313031
3566366462316637353830663364353065653234333339390a373066623164353637623738616634
36306232653261316663313231346661333432633537383665616438613162373237343863313832
3865643631343233310a356463333933633564666263336133373164313036646536303233613832
30393132343331383632623933353335363461343930383064643438376534333863326535326663
3836373432323537663136396430326138616338373434646462
root@ansible:~/springboot-chat-app/ansible# git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      cred.yaml

nothing added to commit but untracked files present (use "git add" to track)
root@ansible:~/springboot-chat-app/ansible# git add .
root@ansible:~/springboot-chat-app/ansible# git commit -m "added encrypted files for nexus auth"
[master a678068] added encrypted files for nexus auth

```

Final Jenkins-CD-Pipeline

```

pipeline {
    agent any
    stages {
        stage('Git Checkout') {
            steps {
                git credentialsId: 'git-key', url:
'git@github.com:arnabcs10/springboot-chat-app.git'
            }
        }

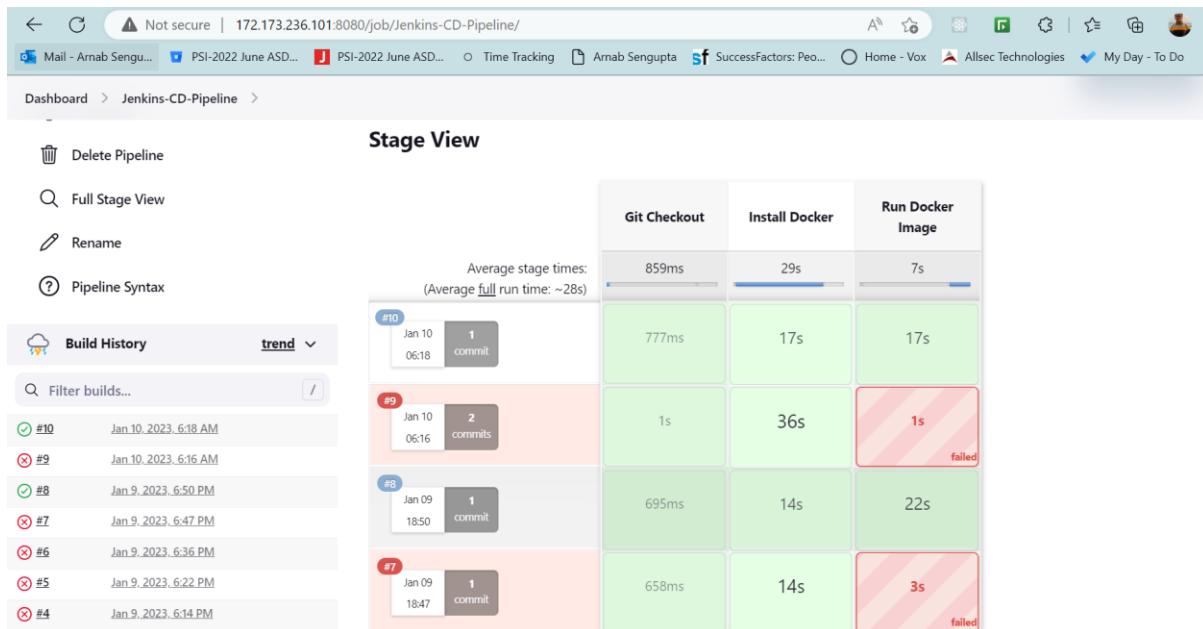
        stage('Install Docker'){
            steps {
                ansiblePlaybook colored: true,
credentialsId: 'git-key', disableHostKeyChecking: true,
inventory: 'ansible/dev.inv', playbook: 'ansible/docker-
install.yaml'
            }
        }

        stage('Run Docker Image'){
            steps {
                ansiblePlaybook colored: true,
credentialsId: 'git-key', disableHostKeyChecking: true,
inventory: 'ansible/dev.inv', playbook: 'ansible/deploy-
docker-image.yaml', vaultCredentialsId: 'vault-pass'
            }
        }
    }
}

```

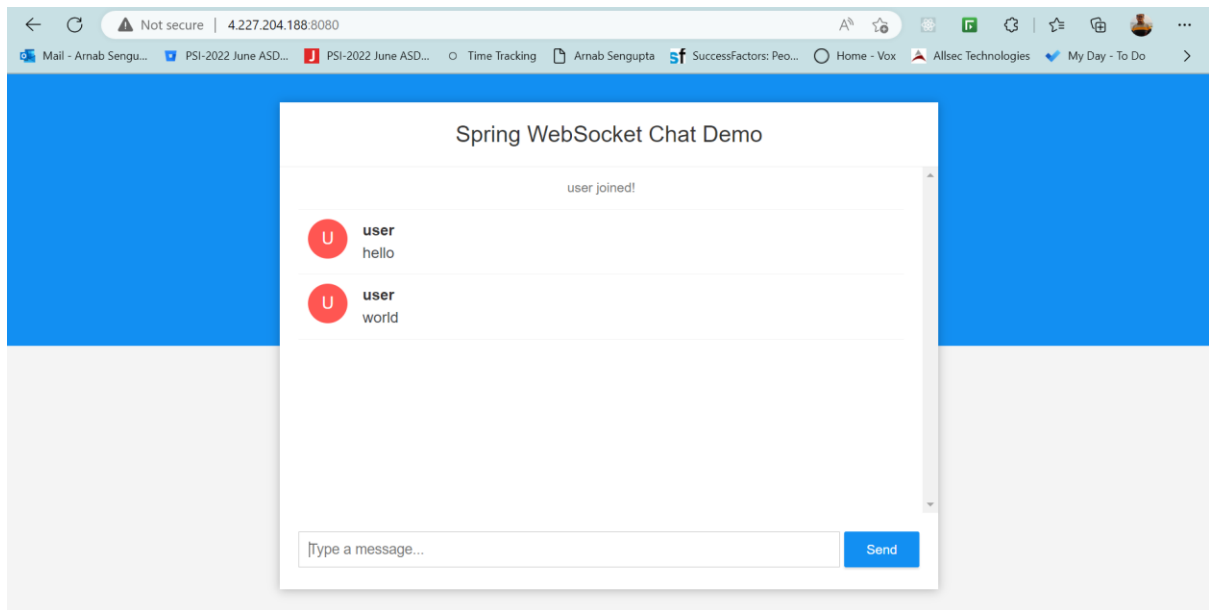
}

Jenkins pipeline stages



Docker installed in Client machine and container running on port 8080:

```
root@client:~#
root@client:~#
root@client:~# docker -v
Docker version 20.10.22, build 3a2c30b
root@client:~#
root@client:~#
root@client:~# docker images
REPOSITORY          TAG         IMAGE ID      CREATED      SIZE
40.117.186.85:8085/chatapp latest      90b41df86f0c  53 years ago 237MB
root@client:~#
root@client:~#
root@client:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED      STATUS      PORTS                  NAMES
fb4a97a40470   40.117.186.85:8085/chatapp "java -cp /app/resou..." 12 hours ago Up 8 minutes 0.0.0.0:8080->8080/tcp chatapp
root@client:~#
root@client:~#
```



Jenkins CD pipeline with tomcat

Create an ansible playbook which will install tomcat

Ansible playbook for installing tomcat to remote server client

```
---
- name: Install Tomcat to remote server
  hosts: all
  become: yes
  vars:
    download_url: https://dlcdn.apache.org/tomcat/tomcat-
10/v10.0.27/bin/apache-tomcat-10.0.27.tar.gz
  tasks:
    - name: Update and upgrade apt packages
      apt:
        upgrade: yes
        update_cache: yes

    - name: Download Open JDK
      apt:
        name: default-jdk
        update_cache: yes
        state: present
```

```
- name: Check if Java is Installed
  shell:
    java -version

- name: Create Group
  group:
    name: tomcat
    state: present

- name: Create User
  user:
    name: tomcat
    state: present

- name: Create a Directory /opt/tomcat
  file:
    path: /opt/tomcat
    state: directory
    mode: 0755
    owner: tomcat
    group: tomcat

- name: Download Tomcat using unarchive
  unarchive:
    src: "{{download_url}}"
    dest: /opt/tomcat
    remote_src: yes
    extra_opts: [--strip-components=1]
- name: Change ownership of tomcat directory
  file:
    path: /opt/tomcat
    owner: tomcat
    group: tomcat
    mode: "u+rwX,g+rx,o=rx"
    recurse: yes
    state: directory

- name: Creating a service file
```

```

copy:
  src: templates/tomcat.service.j2
  dest: /etc/systemd/system/tomcat.service

- name: Reload the SystemD to re-read configurations
  systemd:
    daemon-reload: yes

- name: Enable the tomcat service and start
  systemd:
    name: tomcat
    enabled: yes
    state: started

- name: Connect to Tomcat server on port 8080 and check
status 200 - Try 5 times
  tags: test
  uri:
    url: http://localhost:8080
  register: result
  until: "result.status == 200"
  retries: 5
  delay: 10

```

tomcat.service.j2

```

[Unit]
    Description=Tomcat Service
    Requires=network.target
    After=network.target

    [Service]
    Type=forking
    User=tomcat
    Group=tomcat

    Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-
openjdk-amd64"

```

```

        Environment="JAVA_OPTS=-
Djava.security.egd=file:///dev/urandom"
        Environment="CATALINA_PID=/opt/tomcat/logs/tomcat.pi
d"
        Environment="CATALINA_BASE=/opt/tomcat"
        Environment="CATALINA_HOME=/opt/tomcat"
        Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -
server -XX:+UseParallelGC"

        ExecStart=/opt/tomcat/bin/startup.sh
        ExecStop=/opt/tomcat/bin/shutdown.sh
        Restart=always
        RestartSec=10

        [Install]
        WantedBy=multi-user.target

```

Ansible playbook for deploying WAR file to Tomcat server in Client

```

---
- name: Install Tomcat to remote server
  hosts: all
  become: yes
  vars_files:
    - cred.yaml
  tasks:
    - name: Update and upgrade apt packages
      apt:
        upgrade: yes
        update_cache: yes

    - name: Install lxml
      pip:
        name: lxml

    - name: Download artifact from nexus private repository
      maven_artifact:

```

```

    group_id: websocket-demo
    artifact_id: websocket-demo
    version: 0.0.1-SNAPSHOT
    repository_url:
'http://40.117.186.85:8081/repository/maven-snapshots'
    username: "{{ username }}"
    password: "{{ password }}"
    dest: /opt/tomcat/webapps/chatapp.war
    mode: "0644"

- name: Restart Tomcat
  systemd:
    name: tomcat
    enabled: yes
    state: restarted

```

Jenkins-CD-Pipeline-Tomcat

Jenkins pipeline for installing tomcat and deploying war file on tomcat server in client

```

pipeline {
    agent any
    stages {
        stage('Git Checkout') {
            steps {
                git credentialsId: 'git-key', url:
'git@github.com:arnabcs10/springboot-chat-app.git'
            }
        }

        stage('Install Tomcat'){
            steps {
                ansiblePlaybook colored: true,
credentialsId: 'git-key', disableHostKeyChecking: true,
inventory: 'ansible/dev.inv', playbook:
'ansible/tomcat.yaml'
            }
        }
    }
}

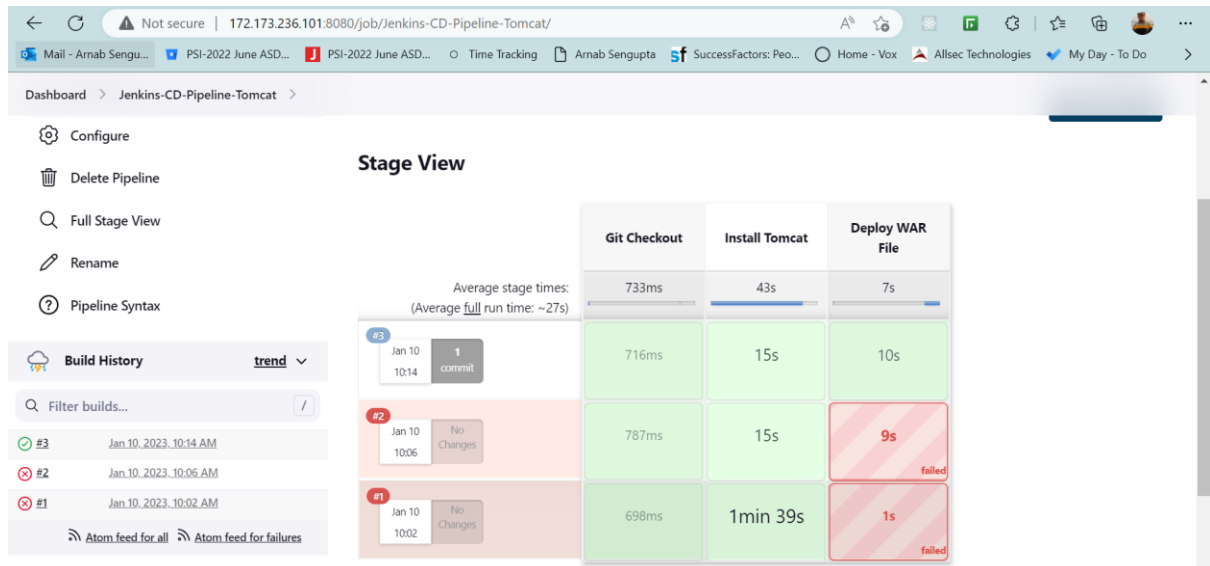
```



```

stage('Deploy WAR File'){
    steps {
        ansiblePlaybook colored: true,
credentialsId: 'git-key', disableHostKeyChecking: true,
inventory: 'ansible/dev.inv', playbook: 'ansible/deploy-war-
file.yaml', vaultCredentialsId: 'vault-pass'
    }
}
}
}

```



```

root@client:/opt/tomcat/webapps#
root@client:/opt/tomcat/webapps# systemctl status tomcat
● tomcat.service - Tomcat Service
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-01-10 10:23:45 UTC; 1h 22min ago
     Process: 15201 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 15208 (java)
      Tasks: 30 (limit: 9530)
     Memory: 431.3M
    CGroup: /system.slice/tomcat.service
            └─15208 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties

Jan 10 10:23:45 client systemd[1]: Starting Tomcat Service...
Jan 10 10:23:45 client startup.sh[15201]: Tomcat started.
Jan 10 10:23:45 client systemd[1]: Started Tomcat Service.
lines 1-13/13 (END)

```

←↻⚠ Not secure | 4.227.204.188:8080/manager/html/deploy?org.apache.catalina.filters.CSRF_NONCE=EE9FF84D0BC42E...A🔖🔧🔍🏠👤...

📧 Mail - Arnab Sengu...📧 PSI-2022 June ASD...📧 PSI-2022 June ASD...🕒 Time Tracking📄 Arnab Sengupta📄 SuccessFactors: Peo...🔊 Home - Vox🔥 Allsec Technologies🔖 My Day - To Do>

Manager

[List Applications](#)[HTML Manager Help](#)[Manager Help](#)[Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/chatapp	None specified		true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/manager	None specified	Tomcat Manager Application	true	1	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/websocket-demo-0.0.1-SNAPSHOT	None specified		true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>

←↻⚠ Not secure | 4.227.204.188:8080/chatapp/home/🔖🔧🔍🏠👤...

📧 Mail - Arnab Sengu...📧 PSI-2022 June ASD...📧 PSI-2022 June ASD...🕒 Time Tracking📄 Arnab Sengupta📄 SuccessFactors: Peo...🔊 Home - Vox🔥 Allsec Technologies🔖 My Day - To Do>

Enter your username

Username

Start Chatting

Jenkins CD Pipeline terraform

1. Authenticate Jenkins server to Azure CLI
2. Create terraform manifest file for provisioning resources.

```
root@ansible:~/new# cd springboot-chat-app/
root@ansible:~/new/springboot-chat-app# ls
Readme.md  azure-pipelines.yml  deploy.yml  infrastructure  mvnw  pom.xml  src
ansible    build.sh             developer.jpg k8s-deployment.yaml  mvnw.cmd  screenshot.png  start.sh
root@ansible:~/new/springboot-chat-app# cd infrastructure/
root@ansible:~/new/springboot-chat-app/infrastructure# tree
.
├── input-variables.tf
├── locals.tf
├── network-interface.tf
├── null-provisioner.tf
├── output.tf
├── provider.tf
├── resource-group.tf
├── script.sh
├── virtual-network.tf
├── vm.tf
├── vnet-input-variables.tf
└── web-subnet.tf

0 directories, 12 files
root@ansible:~/new/springboot-chat-app/infrastructure#
```

3. Create Null provisioner to transfer ssh public key to remote host and installing required packages and dependencies

```
resource "null_resource" "null_copy_ssh_key_to_vm" {
  depends_on = [
    azurerm_linux_virtual_machine.webserver
  ]
  connection {
    type      = "ssh"
    host      =
    azurerm_linux_virtual_machine.webserver.public_ip_address
    user      =
    azurerm_linux_virtual_machine.webserver.admin_username
    private_key = file("~/ssh/id_rsa")
  }
  #file provisioner which will upload my key
  provisioner "file" {
    source      = "~/ssh/id_rsa"
    destination = "/tmp/id_rsa"
  }
  provisioner "file" {
    source      = "script.sh"
  }
}
```

```

    destination = "/tmp/script.sh"
}
provisioner "remote-exec" {
  inline = [
    "sudo chmod 400 /tmp/id_rsa",
    "chmod +x /tmp/script.sh",
    "/tmp/script.sh"
  ]
}
}
}

```

Script.sh

```

#!/bin/bash

# Update apt repository
echo Update apt repository
sudo apt-get update -y

# Install Docker
echo Install Docker
sudo apt-get install -y ca-certificates curl gnupg lsb-
release
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) 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-compose-plugin -y

# Install kind
echo Install kind

```

```

curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.17.0/kind-
linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind
kind --version

# Install kubectl
echo Install kubectl
sudo apt-get install -y apt-transport-https
sudo curl -fsSL https://packages.cloud.google.com/apt/doc/apt-
keyring.gpg | sudo tee /etc/apt/keyrings/kubernetes-archive-
keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-archive-
keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial
main' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubectl
kubectl version --short

# Create kind cluster
echo Create kind cluster
sudo kind create cluster
sudo kubectl get nodes
sudo kubectl get pods

# Install helm
echo Install helm
wget https://get.helm.sh/helm-v3.10.3-linux-amd64.tar.gz
tar zxvf helm-v3.10.3-linux-amd64.tar.gz
cd linux-amd64/
sudo cp helm /usr/bin
helm version

```

4. Create Jenkins pipeline

```

pipeline {
    agent any

```

```
tools {
    terraform 'terraform'
}
environment {
    RESOURCE_GROUP = 'rg-default'
    HOST_VM = 'web-linuxvm'
    USERNAME = 'azureuser'
}
stages {
    stage('Poll Code Repository') {
        steps {
            git credentialsId: 'git-key', url:
'git@github.com:arnabcs10/springboot-chat-app.git'
        }
    }
    stage('Terraform Init') {
        when {
            expression{action == "apply"}
        }
        steps{
            script {
                dir('infrastructure/') {
                    sh 'terraform init'
                }
            }
        }
    }
    stage('Terraform Syntax Check') {
        when {
            expression{action == "apply"}
        }
        steps{
            script {
                dir('infrastructure/') {
                    sh 'terraform validate'
                }
            }
        }
    }
}
```

```

    }

    stage('Terraform Plan') {
        when {
            expression{action == "apply"}
        }
        steps{
            script {
                dir('infrastructure/') {
                    sh 'terraform plan'
                }
            }
        }
    }

    stage('Terraform Apply') {
        when {
            expression{action == "apply"}
        }
        steps{
            script {
                dir('infrastructure/') {
                    sh 'terraform apply --auto-approve'
                }
            }
        }
    }

    stage('SSH Connection to VM') {
        when {
            expression{action == "apply"}
        }
        steps{
            sh'''
                PUBLIC_IP=$(az vm show -d -g $RESOURCE_GROUP
-n $HOST_VM --query publicIps -o tsv)
                ssh -tt -o "StrictHostKeyChecking no"
$USERNAME@$PUBLIC_IP <<'EOT'
                echo "They are executed by: $( whoami )"
                exit
            '''
        }
    }

```

```

        '''
    }
}
stage("Install Istio"){

    when {
        expression{action == "apply"}
    }
    steps{
        sh '''
            PUBLIC_IP=$(az vm show -d -g $RESOURCE_GROUP
-n $HOST_VM --query publicIps -o tsv)
            ssh -tt -o "StrictHostKeyChecking no"
$USERNAME@$PUBLIC_IP <<'EOT'
            curl -L https://istio.io/downloadIstio | sh
-

            cd istio-1.16.1
            export PATH=$PWD/bin:$PATH
            sudo istioctl install --set profile=demo -y
            sudo kubectl label namespace default istio-
injection=enabled
            exit
            '''
    }
}
stage('Installing Helm Chart') {
    when {
        expression{action == "apply"}
    }
    steps{
        sh'''
            PUBLIC_IP=$(az vm show -d -g $RESOURCE_GROUP
-n $HOST_VM --query publicIps -o tsv)
            ssh -tt -o "StrictHostKeyChecking no"
$USERNAME@$PUBLIC_IP <<'EOT'
            sudo helm repo add bitnami
https://charts.bitnami.com/bitnami

```



```

        sudo helm install my-nginx-release
bitnami/nginx
        sudo kubectl get pod -o
jsonpath='{.items[0].metadata.name}'
        exit
    ...
    }
}

stage("Terraform destroy"){
    when {
        expression{action == "destroy"}
    }
    steps{
        sh 'terraform destroy -auto-approve'
    }
}

}
}

```

Not secure | 172.173.236.101:8080/job/Jenkins-CD-Pipeline-Terraform/

Mail - Arnab Sengu... | PSI-2022 June ASD... | PSI-2022 June ASD... | Time Tracking | Arnab Sengupta | SuccessFactors: Peo... | Home - Vox | Allsec Technologies | My Day - To Do

Dashboard > Jenkins-CD-Pipeline-Terraform >

Build with Parameters
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax

Build History trend

Filter builds...

Jan 13, 2023 5:06 AM
Jan 12, 2023 9:45 AM
Jan 12, 2023 9:42 AM
Jan 12, 2023 7:08 AM
Jan 12, 2023 7:04 AM
Jan 12, 2023 7:02 AM

Atom feed for all | Atom feed for failures

Stage View

Average stage times:
(Average full run time: ~5min 43s)

	Declarative: Tool Install	Poll Code Repository	Terraform Init	Terraform Syntax Check	Terraform Plan	Terraform Apply	SSH Connection to VM	Install Istio	Installing Helm Chart	Terraform destroy
Jan 13 05:06	1s	2s	7s	3s	27s	4min 22s	4s	24s	8s	0ms

Permalinks

- Last build (#6), 12 min ago
- Last stable build (#6), 12 min ago
- Last successful build (#6), 12 min ago
- Last failed build (#3), 22 hr ago
- Last unsuccessful build (#3), 22 hr ago
- Last completed build (#6), 12 min ago

Helm chart deployed on remote server

```
Dashboard > Jenkins-CD-Pipeline-Terraform > #6
```

```
----- my-nginx -----
LAST DEPLOYED: Fri Jan 13 05:12:15 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: nginx
CHART VERSION: 13.2.21
APP VERSION: 1.23.3

** Please be patient while the chart is being deployed **

NGINX can be accessed through the following DNS name from within your cluster:

    my-nginx-release.default.svc.cluster.local (port 80)

To access NGINX from outside the cluster, follow the steps below:

1. Get the NGINX URL by running these commands:

NOTE: It may take a few minutes for the LoadBalancer IP to be available.
      Watch the status with: 'kubectl get svc --namespace default -w my-nginx-release'

export SERVICE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].port}" services my-nginx-release)
export SERVICE_IP=$(kubectl get svc --namespace default my-nginx-release -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo "http://${SERVICE_IP}:${SERVICE_PORT}"

azureuser@web-linuxvm:~$
<ctl get pod -o jsonpath='{.items[0].metadata.name}'
my-nginx-release-7597b974f5-29h3azsazureuser@web-linuxvm:~$          exit
logout
Connection to 20.158.216.80 closed.
```

Created Resource:

The screenshot shows the Microsoft Azure portal interface. At the top, the browser address bar displays the URL: <https://portal.azure.com/#@publicisgroupe.net/resource/subscriptions/70c6ff6d-44c7-461c-a2b1-c1a1bbd37...>. The portal header includes the Microsoft Azure logo and a search bar. The main content area is titled 'Resource groups' and shows the 'rg-default' resource group selected. The left sidebar contains a navigation menu with options like 'Home', 'Resource groups', 'Create', 'Manage view', and 'Filter for any field...'. The 'Overview' tab is selected for the 'rg-default' resource group. The right pane displays a list of resources within the group, including 'retail-dev-vnet-default', 'web-linuxvm', 'web-linuxvm_OsDisk_1_8c9b179aebf94...', and 'websubnet-nsg'. The bottom of the page shows a pagination bar indicating 'Page 1 of 1'.

Destroying the Infrastructure:

	Declarative: Tool Install	Poll Code Repository	Terraform Init	Terraform Syntax Check	Terraform Plan	Terraform Apply	SSH Connection to VM	Install Istio	Installing Helm Chart	Terraform destroy
Average stage times: (Average <u>full</u> run time: ~1min 24s)	520ms	1s	4s	3s	24s	2min 20s	4s	15s	6s	33s
#13 Jan 13 05:44 No Changes	412ms	883ms								2min 43s
#12 Jan 13 05:42 No Changes										
#11 Jan 13 05:37 No Changes	376ms									759ms
#10 Jan 13 05:35 No Changes	393ms	933ms	2s	2s	20s	17s	4s	7s	4s	

Jenkins CD Pipeline terraform AKS

1. Authenticate Jenkins server to Azure CLI
2. Create terraform manifest file for provisioning resources AKS Cluster

```
root@ansible:~/new/springboot-chat-app# cd aks-cluster-infrastructure/
root@ansible:~/new/springboot-chat-app/aks-cluster-infrastructure# ls
aks-cluster.tf  outputs.tf  provider.tf
root@ansible:~/new/springboot-chat-app/aks-cluster-infrastructure# tree
.
├── aks-cluster.tf
├── outputs.tf
└── provider.tf

0 directories, 3 files
root@ansible:~/new/springboot-chat-app/aks-cluster-infrastructure# |
```

3. Create Jenkins pipeline

```
pipeline {
    agent any
    tools {
        terraform 'terraform'
    }
    environment {
        CHART_NAME = 'nginx-service'
    }
    stages {
        stage('Poll Code Repository') {
            steps {
                git credentialsId: 'git-key', url:
'git@github.com:arnabcs10/springboot-chat-app.git'
            }
        }
        stage('Terraform Init') {
            when {
                expression{action == "apply"}
            }
            steps{
                script {
                    dir('aks-cluster-infrastructure/') {
                        sh 'terraform init'
                    }
                }
            }
        }
    }
}
```

```

    }
}
stage('Terraform Syntax Check') {
    when {
        expression{action == "apply"}
    }
    steps{
        script {
            dir('aks-cluster-infrastructure/') {
                sh 'terraform validate'
            }
        }
    }
}

stage('Terraform Plan') {
    when {
        expression{action == "apply"}
    }
    steps{
        script {
            dir('aks-cluster-infrastructure/') {
                sh 'terraform plan'
            }
        }
    }
}

stage('Terraform Apply') {
    when {
        expression{action == "apply"}
    }
    steps{
        script {
            dir('aks-cluster-infrastructure/') {
                sh 'terraform apply --auto-approve'
            }
        }
    }
}

```

```

    }
    stage('Configure Kubectl') {
        when {
            expression{action == "apply"}
        }
        steps {
            script {
                dir('aks-cluster-infrastructure/') {
                    sh'''
                        az aks get-credentials --resource-
group $(terraform output -raw resource_group_name) --name
$(terraform output -raw kubernetes_cluster_name)
                        kubectl get nodes
                        ...
                    '''
                }
            }
        }
    }

    stage('Installing Helm Chart') {
        when {
            expression{action == "apply"}
        }
        steps{
            sh'''
                helm list
                # helm repo add bitnami
                https://charts.bitnami.com/bitnami
                # helm install $CHART_NAME
                bitnami/nginx
                ...
            '''
        }
    }

    stage('Verify Deployments') {
        when {
            expression{action == "apply"}
        }
        steps{

```

```

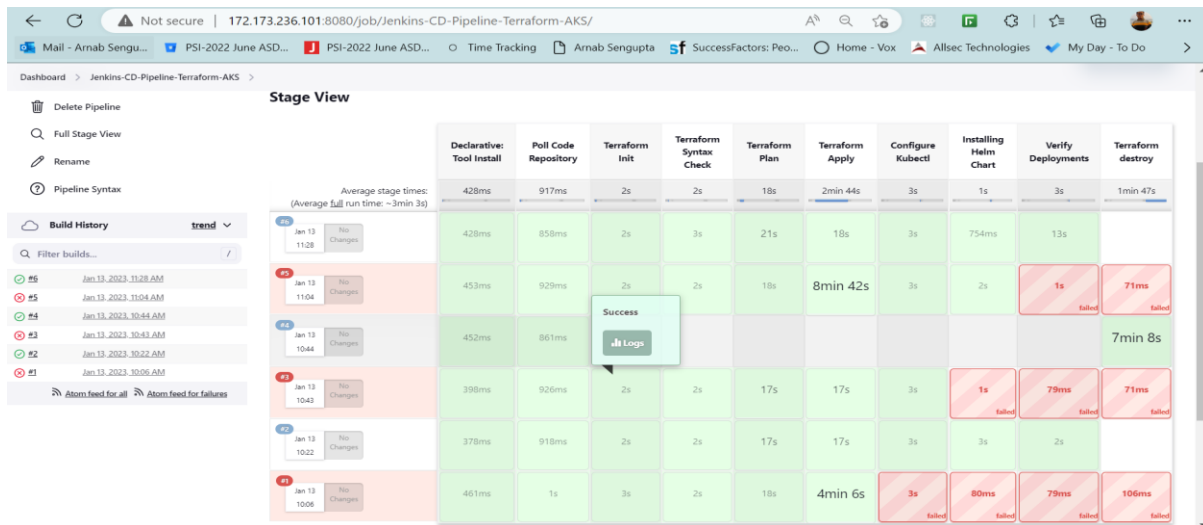
        sh'''
        kubectl get pods
        kubectl get deployments
        kubectl get services

        echo "Waiting for end point..."
        sleep 10
        EXTERNAL_IP=$(kubectl get svc $CHART_NAME -o
yaml | grep -oP '(?<=ip: )[0-9].+')
        echo 'End point ready:' && echo $EXTERNAL_IP
        echo "URL: http://$EXTERNAL_IP"
        '''
    }
}

stage("Terraform destroy"){
    when {
        expression{action == "destroy"}
    }
    steps {
        script {
            dir('aks-cluster-infrastructure/') {
                sh 'terraform destroy --auto-
approve'
            }
        }
    }
}
}

```

Build Pipeline:



```
[Pipeline] sh
```

```
+ terraform output -raw resource_group_name
```

```
+ terraform output -raw kubernetes_cluster_name
```

```
+ az aks get-credentials --resource-group topical-hen-rg --name topical-hen-aks
```

```
WARNING: Merged "topical-hen-aks" as current context in /var/lib/jenkins/.kube/config
```

```
+ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-default-21914257-vmss000000	Ready	agent	18m	v1.24.6
aks-default-21914257-vmss000001	Ready	agent	21m	v1.24.6

```
+ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-service-75647c85d-j8p5z	1/1	Running	0	16m

```
+ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-service	1/1	1	1	16m

```
+ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	22m
nginx-service	LoadBalancer	10.0.4.249	20.88.166.10	80:31085/TCP	16m

```
+ echo Waiting for end point...
```

```
Waiting for end point...
```

```
+ sleep 10
```

```
+ kubectl get svc nginx-service -o yaml
```

```
+ grep -oP (?<=ip: )[0-9].+
```

```
+ EXTERNAL_IP=20.88.166.10
```

```
+ echo End point ready:
```

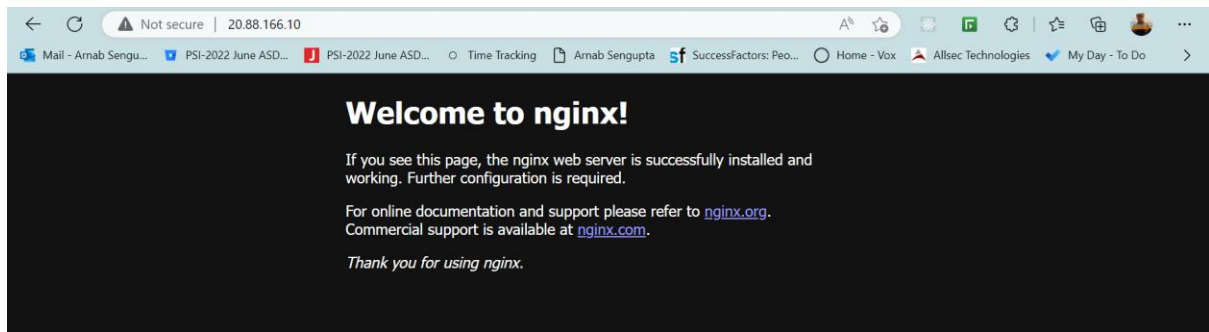
```
End point ready:
```

```
+ echo 20.88.166.10
```

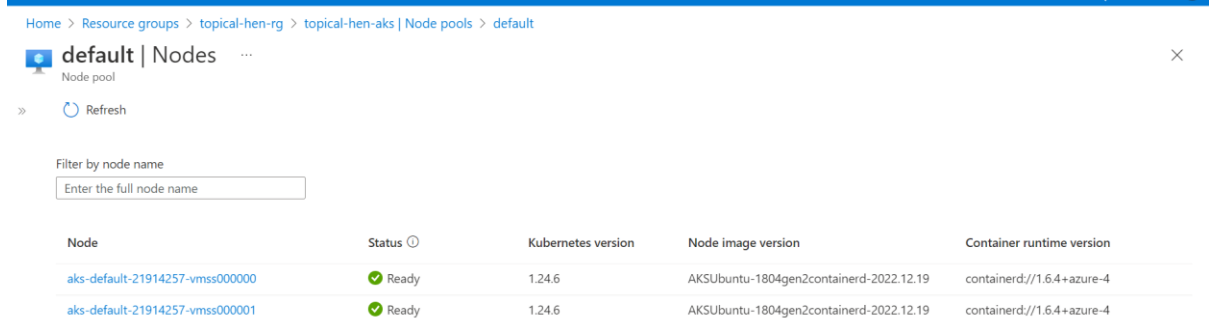
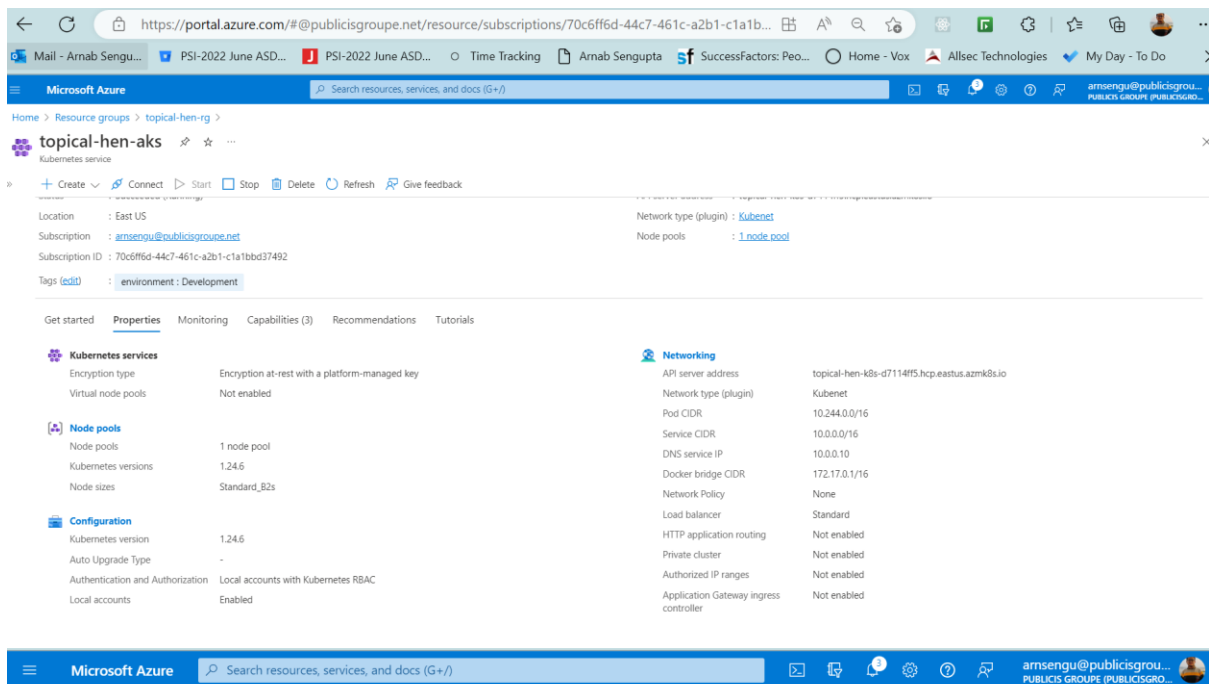
```
20.88.166.10
```

```
+ echo URL: http://20.88.166.10
```

```
URL: http://20.88.166.10
```



Created AKS Cluster:



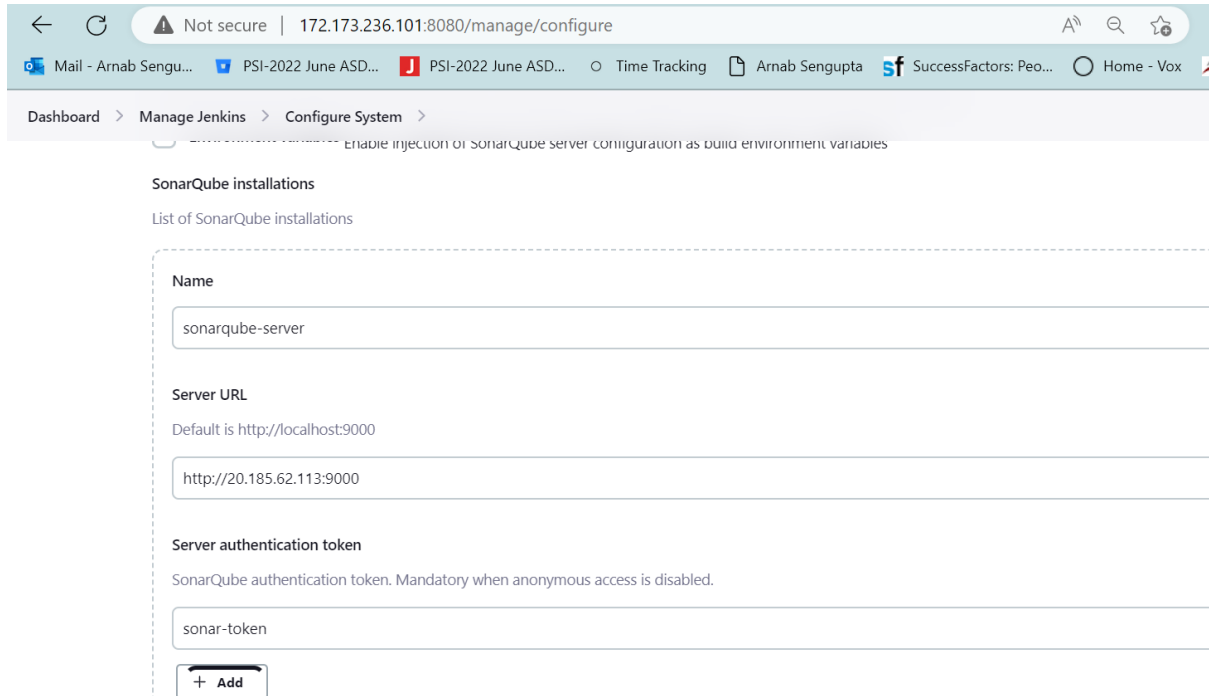
Destroying AKS Cluster:



Jenkins: sonar pipeline. Include SonarQube quality gate enforcement in Jenkins pipeline to be added.

1. Configured Jenkins setting for SonarQube quality gate

Added sonar server URL and authentication token



The screenshot shows the Jenkins 'Configure System' page for 'Manage Jenkins'. Under the 'SonarQube installations' section, there is a list of installations. One installation is configured with the following details:

- Name:** sonarqube-server
- Server URL:** http://20.185.62.113:9000 (Default is http://localhost:9000)
- Server authentication token:** sonar-token (Mandatory when anonymous access is disabled)

At the bottom of the configuration area, there is a '+ Add' button to add more installations.

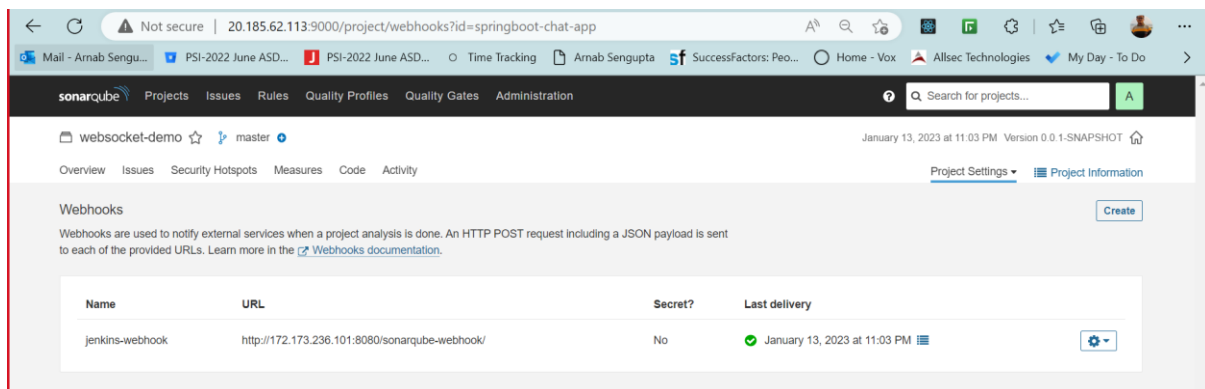
Modified Jenkins pipeline stages for quality gate:

Definition

Pipeline script

```
Script ?
30
31
32 stage('Sonarqube Analysis'){
33     steps {
34         withSonarQubeEnv('sonarqube-server') {
35             sh 'mvn clean verify sonar:sonar -Dsonar.projectKey=springboot-chat-app'
36         }
37     }
38 }
39
40 stage('SonarQube Quality Check'){
41     steps{
42         timeout(time:1, unit:'MINUTES'){
43             waitForQualityGate abortPipeline: true, credentialsId: 'sonar-token'
44         }
45     }
46 }
47
```

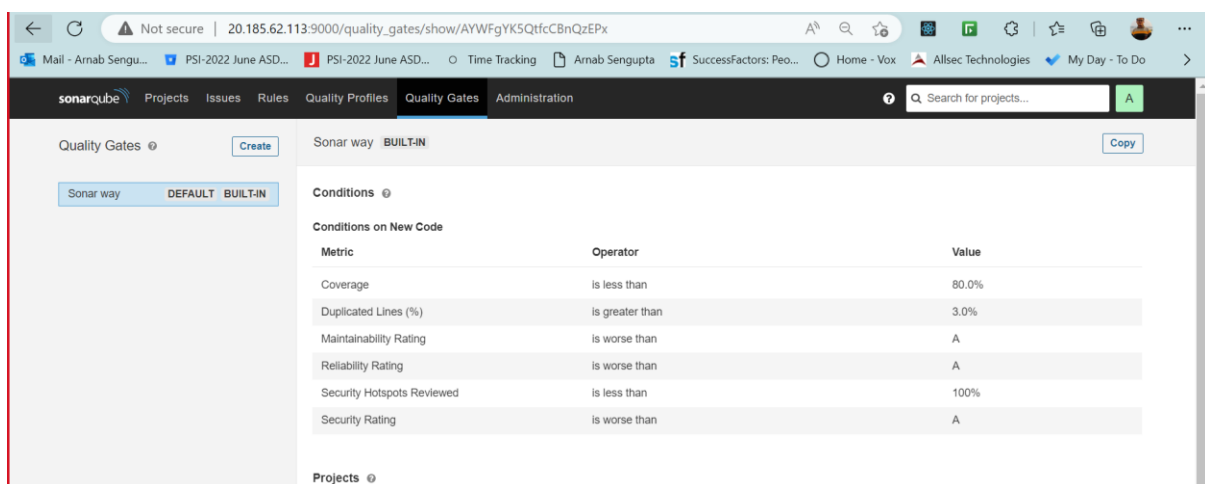
2. Added SonarQube webhook for Jenkins



The screenshot shows the SonarQube interface for configuring webhooks. The 'Webhooks' section is active, displaying a table of configured webhooks. A single webhook named 'jenkins-webhook' is listed with the URL 'http://172.173.236.101:8080/sonarqube-webhook/'. The 'Secret?' column shows 'No', and the 'Last delivery' column shows a successful status with a green checkmark and the timestamp 'January 13, 2023 at 11:03 PM'.

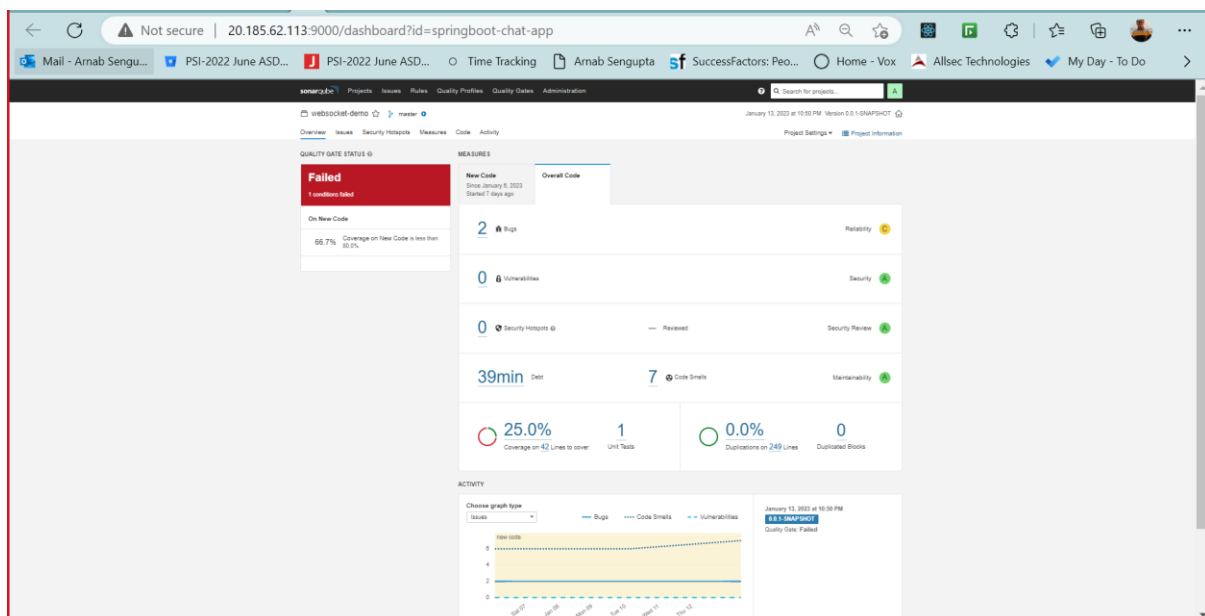
Name	URL	Secret?	Last delivery
jenkins-webhook	http://172.173.236.101:8080/sonarqube-webhook/	No	✓ January 13, 2023 at 11:03 PM

Quality gate: Built In



The screenshot shows the 'Quality Gates' configuration page in SonarQube. The 'BUILT-IN' quality gate is selected. The 'Conditions on New Code' table lists various metrics and their thresholds.

Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A
Reliability Rating	is worse than	A
Security Hotspots Reviewed	is less than	100%
Security Rating	is worse than	A



The screenshot shows the SonarQube dashboard for the 'websocket-demo' project. The 'Quality Gate Status' is 'Failed' (red box). The 'Measures' section shows various metrics: 2 Bugs, 0 Vulnerabilities, 0 Security Hotspots, 39min Debt, 7 Code Smells, 25.0% Coverage on 42 Lines to cover, 1 Unit Tests, 0.0% Duplications on 249 Lines, and 0 Duplicated Blocks. The 'Activity' section shows a graph of 'New Code' over time, with a red line indicating a failure.

Quality Gate Status: Failed

Measures:

- New Code: 2 Bugs, 0 Vulnerabilities, 0 Security Hotspots, 39min Debt, 7 Code Smells
- Coverage: 25.0% (Coverage on 42 Lines to cover)
- Unit Tests: 1
- Duplications: 0.0% (Duplications on 249 Lines)
- Duplicated Blocks: 0

Activity:

Choose graph type: Issues, Bugs, Code Smells, Vulnerabilities

January 13, 2023 at 10:50 PM
Quality Gate Failed

Quality Check failed

Not secure | 20.185.62.113:9000/projects

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects...

My Favorites All

Search by project name or key

1 project(s) Perspective: Overall Status Sort by: Name

websocket-demo Failed Last analysis: 3 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
2	0	-	7	25.0%	0.0%	249

1 of 1 shown

Not secure | 172.173.236.101:8080/job/Jenkins-CI-Pipeline/7/console

Dashboard > Jenkins-CI-Pipeline > #7

```
[Pipeline] { (SonarQube Quality Check)
[Pipeline] timeout
Timeout set to expire in 1 min 0 sec
[Pipeline] {
[Pipeline] waitForQualityGate
Checking status of SonarQube task 'AYWsM2BwQyCLmDEzeXKv' on server 'sonarqube-server'
SonarQube task 'AYWsM2BwQyCLmDEzeXKv' status is 'PENDING'
SonarQube task 'AYWsM2BwQyCLmDEzeXKv' status is 'SUCCESS'
SonarQube task 'AYWsM2BwQyCLmDEzeXKv' completed. Quality gate is 'ERROR'
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Artifact)
Stage "Push Artifact" skipped due to earlier failure(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker Push)
Stage "Docker Push" skipped due to earlier failure(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: Pipeline aborted due to quality gate failure: ERROR
```

```
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Quality Check)
[Pipeline] timeout
Timeout set to expire in 1 min 0 sec
[Pipeline] {
[Pipeline] waitForQualityGate
Checking status of SonarQube task 'AYWsQ25IQyCLmDEzeXK7' on server 'sonarqube-server'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' status is 'IN_PROGRESS'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' status is 'SUCCESS'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' completed. Quality gate is 'OK'
[Pipeline] }
[Pipeline] // timeout
```

3. Added new quality gate to pass build pipeline:

The screenshot shows the SonarQube 'Quality Gates' configuration page for a project named 'My-QualityGate'. The interface includes a sidebar with 'My-QualityGate' selected, and a main area with a table of conditions. The conditions are as follows:

Metric	Operator	Value	Edit	Delete
Coverage	is less than	60.0%		
Duplicated Lines (%)	is greater than	3.0%		
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	A		
Security Rating	is worse than	A		

Build Passed:

The screenshot shows the SonarQube 'Projects' page. A search filter is applied, showing 1 project(s). The project 'websocket-demo' is listed with a 'Passed' status. The last analysis was 1 hour ago. The project's quality gate status is 'Passed'.

Metric	Value	Rating
Bugs	2	C
Vulnerabilities	0	A
Hotspots Reviewed	—	A
Code Smells	7	A
Coverage	25.0%	C
Duplications	0.0%	C
Lines	249	XS

The screenshot shows the SonarQube dashboard for the 'websocket-demo' project. The 'Quality Gate Status' is 'Passed'. The 'Measures' section displays various metrics:

Metric	Value	Rating
New Code	Since January 6, 2023	Started 7 days ago
Bugs	2	C
Vulnerabilities	0	A
Security Hotspots	0	A
Code Smells	7	A
Coverage	25.0%	C
Duplications	0.0%	C

Jenkins Output: Build Success

```
[Pipeline] { (SonarQube Quality Check)
[Pipeline] timeout
Timeout set to expire in 1 min 0 sec
[Pipeline] {
[Pipeline] waitForQualityGate
Checking status of SonarQube task 'AYWsQ25IQyCLmDEzeXK7' on server 'sonarqube-server'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' status is 'IN_PROGRESS'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' status is 'SUCCESS'
SonarQube task 'AYWsQ25IQyCLmDEzeXK7' completed. Quality gate is 'OK'
[Pipeline] }
```

Dashboard > Jenkins-CI-Pipeline >

Rename

CS CheckStyle Warnings

Coverage Trend

Pipeline Syntax

Build History

trend

Filter builds...

#9

Jan 13, 2023, 5:51 PM

#8

Jan 13, 2023, 5:45 PM

#7

Jan 13, 2023, 5:32 PM

#6

Jan 13, 2023, 5:28 PM

#5

Jan 13, 2023, 5:19 PM

#4

Jan 10, 2023, 10:00 AM

#3

Jan 10, 2023, 9:55 AM

#2

Jan 9, 2023, 11:31 AM

#1

Jan 9, 2023, 10:52 AM

CheckStyle Warnings Trend

Code Coverage Trend

Stage View

Not secure | 172.173.236.101:8080/job/Jenkins-CI-Pipeline/

Mail - Arnab Sengupta... PSI-2022 June ASD... PSI-2022 June ASD... Time Tracking Arnab Sengupta SuccessFactors: Peo... Home - Vox Allsec Technologies My Day - To Do

Dashboard > Jenkins-CI-Pipeline >

#6

Jan 13, 2023, 5:28 PM

#5

Jan 13, 2023, 5:19 PM

#4

Jan 10, 2023, 10:00 AM

#3

Jan 10, 2023, 9:55 AM

#2

Jan 9, 2023, 11:31 AM

#1

Jan 9, 2023, 10:52 AM

Atom feed for all Atom feed for failures

CheckStyle Warnings Trend

Code Coverage Trend

Stage View

	Poll Code Repository	Build	Test	Checkstyle	Code Coverage	Sonarqube Analysis	SonarQube Quality Check	Push Artifact	Docker Push
Average stage times: (Average full run time: ~1min 39s)	943ms	25s	184ms	9s	318ms	41s	396ms	921ms	5s
#9 Jan 13 17:51 No Changes	638ms	23s	141ms	8s	195ms	38s	264ms (paused for 3s)	3s	20s
#8 Jan 13 17:45 No Changes	699ms	23s	132ms	8s	282ms	38s	289ms (paused for 5s) failed	84ms failed	76ms failed
#7 Jan 13 17:32 No Changes	697ms	24s	176ms	8s	245ms	40s	388ms (paused for 4s) failed	97ms failed	77ms failed