**Project Internship**

**WEEK - 1**

* Linux
  + Basic To Advance Commands
    - File Management
    - User Management
    - Process Management
    - Resource Management

**WEEK - 2**

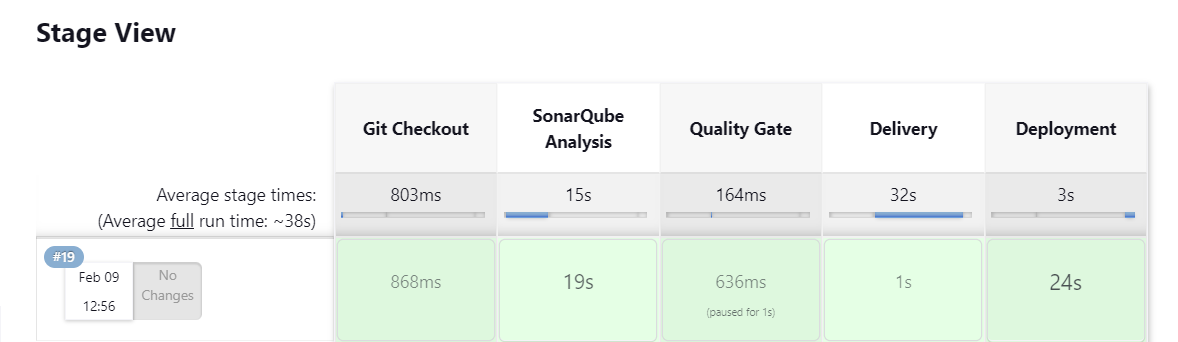
* AWS
  + Basics Of Cloud
  + AWS Basics:-
    - VPC
    - NAT Gateway
    - Internet Gateway
    - Subnets & Routing
    - Security Groups
  + AWS Services:-
    - EC2
    - S3
    - AMI
    - Elastic Load Balancer

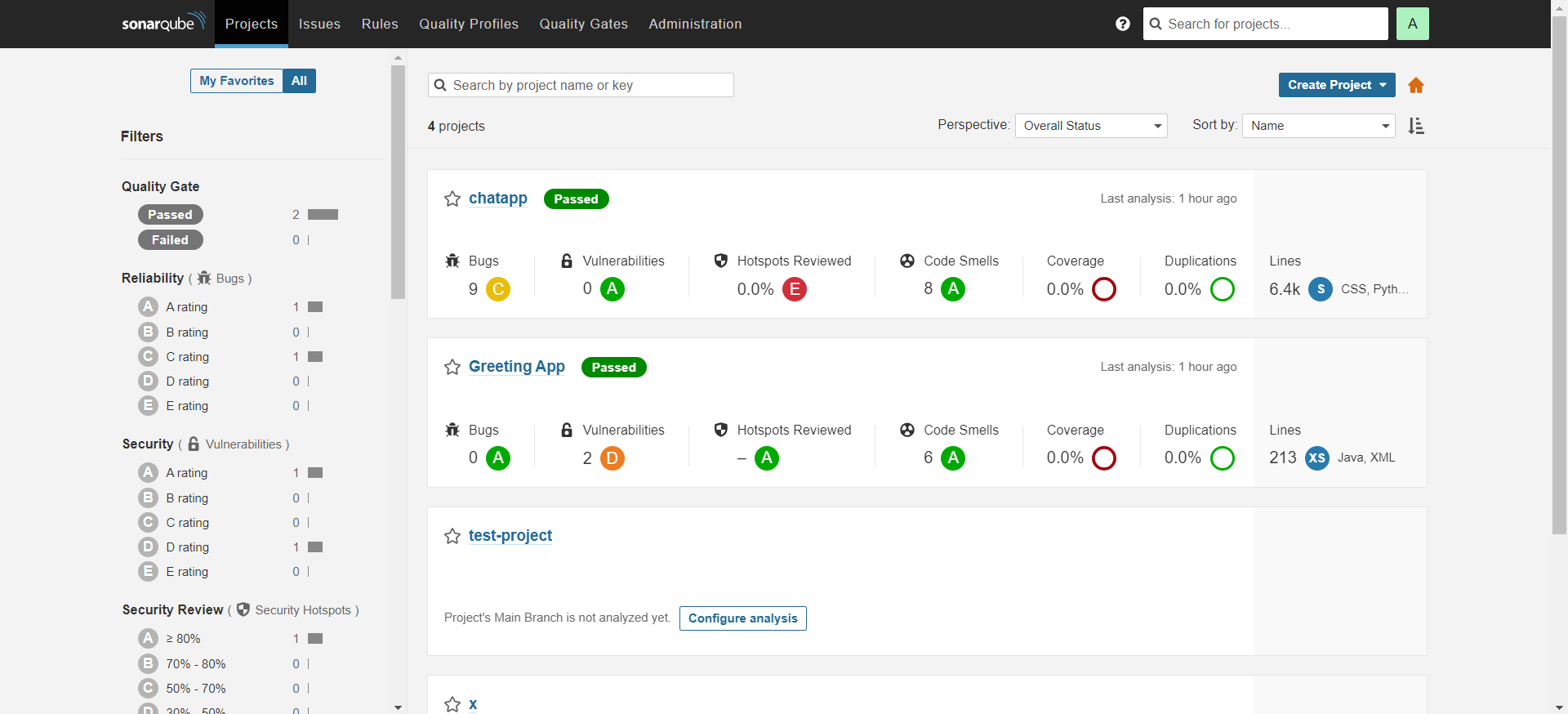
**WEEK - 3**

* + AWS IAM (Identity & Access Management):-
    - User Management
    - Group Management
    - Policy
    - Roles
    - Cross Account Access
  + Jenkins:- Automation tool for CICD.
    - CI/CD & software development life cycle.
    - Freestyle Job Configuration.
    - Pipeline Job Configuration.
    - Master-Agent configuration: Running jobs on agent node & job distribution among different nodes.
    - Automating the deployment process of Java Spring application:
      * Git Checkout
      * Project Build
      * Project Test
      * Delivery & Deployment
    - Plugins in Jenkins:
      * SonarQube

*SonarQube is an open-source platform developed by Sonar-Source for* ***continuous inspection*** *of* ***code quality*** *to perform automatic reviews with static analysis of code to detect bugs and code smells on 29 programming languages.*

* + - * Grafana
      * CloudWatch
    - Automating the deployment process of NodeJS application.
    - Automating the deployment process of DJANGO application:
      * Git Checkout
      * SonarQube Analysis
      * Quality Gate Check
      * Project Build
      * Project Delivery
      * Project Deployment



****

**SCRIPT:**

pipeline {

agent {

label 'jenkins-agent-maven'

}

stages {

stage('Git Checkout') {

steps {

git changelog: false, poll: false, url: 'https://github.com/RavenClaaaw/chat-app'

}

}

stage('SonarQube Analysis') {

environment {

SCANNER\_HOME = tool 'sonar-general'

}

steps {

script{

withSonarQubeEnv() {

sh "${SCANNER\_HOME}/bin/sonar-scanner"

}

}

}

}

stage("Quality Gate") {

steps {

timeout(time: 1, unit: 'HOURS') {

waitForQualityGate abortPipeline: true

}

}

}

stage('Delivery'){

steps{

sh 'rsync -avz /home/jenkins/workspace/chatapp-pipeline whoisroop@10.0.10.197:/tmp/'

}

}

stage('Deployment'){

steps{

sh 'ssh jenkins@10.0.10.197 "sudo -u whoisroop /usr/local/bin/chatapp-deploy.sh"'

}

}

}

}

**Week – 4 (12-2-24 TO 16-2-24)**

* + Cloud Monitoring:
    - GRAFANA (Server)- *Grafana is a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.*
    - PROMETHEUS (Server)- *An open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and modern alerting approach.*
    - NODE-EXPORTER (Instance): *It can collect and store node-level metrics as time-series data, recording information with a timestamp.*
  + DOCKER:
    - Docker installation & commands
    - Docker concepts: -
      * Images
        + Pulling images from docker registry
        + Building on top of registry images: DOCKERFILE
        + ECR (Elastic Container Registry) – AWS DOCKER PRIVATE REGISTRY SERVICE
      * Containers
        + Docker Networks
        + Docker Volumes
        + Docker Compose
    - Project: - Containerizing The 3 Tier Django Application (Docker Compose):
      * Frontend – Webserver
        + (NGINX – Base Image)
      * Backend – GIT, PYTHON Django, G-Unicorn
        + (Ubuntu:18.04 – Base Image)
      * Database – MySQL
        + (MySQL – Base Image)
  + KUBERNETES

**PROJECT**

* + Week – 2
    - Deployment of 3 tier application on AWS (LINUX):
      * Frontend – Webserver (NGINX)
      * Backend – GIT, PYTHON Django, G-Unicorn
      * Database – MySQL
      * Cloning instances using amazon machine images & setting up elastic load balance to distribute the load among different frontend – application load balancer & backend – network load balancer.
  + Week - 3
    - Deployment of 3 tier application on AWS (LINUX):
      * Frontend - Webserver (NGINX)
      * Backend – GIT, Node-JS, PM2 (Process Manager - 2)
      * Database – MongoDB