**Java web application deployment:**

**Cloud:** AWS

**Tools used:**

**1.Terraform** - To Provision the servers in AWS

**2.Github** - Source Code Management

**3.Jenkins** - Continuous Integration

**4.Maven** - Continuous build

**5.Sonarqube** -Static code analysis and code quality check

**6.Nexus** - Artifact repository

**7.Ansible** - Configuration management and continues deployment

**Workflow:**

1. **InfraCreationstage:**

Creating the ec2 instances on aws using the terraform.

Jenkins pipeline is designed to pull the code the GitHub and using TerraformJenkinsfile where i have mentioned the stages and executed on aws cloud to provision

1. **Git Checkout stage:**

In the next stage the jenkins will pull the SC from GitHub a webhook is configured in GitHub so that ongoing deployment job will trigger automatically when a commit happens on particular branch.

1. **Unit stage:**

The java application source code will be unit test and integration test using maven either in jenkins master or slave based on the target operating system.

1. **Build stage:**

The java application source code will be build using maven either in jenkins master or slave based on the target operating system.

In case any failure in the build step pipeline will be aborted and mail will get trigger to Teams.

1. **Static Code Analysis stage:**

jenkins uses the SonarQube scanner plugin for continues inspection of code quality to perform automatic reviews, static analysis of code to detect bugs, code smells, and security vulnerabilities.

SonarQube offers report duplicated code, coding standards, code coverage, code complicity, bugs, etc. When the code doesn’t meet the quality standards and alert mail trigger to development team with report generated and the pipeline will be aborted.

1. **Quality Gate Status:**

quality gates are set with conditions (with values & operator) like how many bugs are there, if these bugs are reaching to threshold value, it indicates whether your code is clean and can move forward.

Based on the quality gate pass/fail the build is prompted to next stage

* A passing (green) quality gate means the code meets your standard and is ready to be merged. (It will promote)
* A failing (red) quality gate means there are issues to address. (It will not promote)

1. **Nexus Upload stage:**

once the code meets the standards, the java application package (War File) will be pushed to the nexus repository.

1. **Deployment stage:**

Jenkins will execute the ansible playbook on all the servers which will the following tasks to target servers.

1. Download and install java, download and configure tomcat.

2. Pull the war file from Nexus and deploy the war file in tomcat server, finally the java application will deploy into the server.

We can see out web application at **http://localhost:8080**

I have not configured any database, it’s a static web application.