

< CONTINUOUS INTEGRATION & DELIVERY OF JAVA WEB
APPLICATION [Jenkins & Ansible] >

IMPLEMENTATION PLAN

VERSION 1.0 | 10/06/2022 – 15/06/2022

Purpose

This section, describe the Continuous Integration & Delivery using Jenkins and Ansible.

Project Scenario

- Agile SDLC.
- Developer makes regular code changes.
- These commits need to be regularly Build and Tested.
- Build and Release Team will do this job.
- Or Developer job to merge and integrate code.

Problem Statement

- PACKAGE/SOFTWARE/ARTIFACT Deployment on server.
- Software testing & integration testing after deployment.
- Test report gets evaluated and approved each time for production deployment.
- In Agile SDLC, there will be frequent code changes.
- Manual code deployment is time consuming.
- Involves task assignment/ticketing/approvals.
- Dependency on Ops & Build & Release Team.

Solution

- Build & test for every commit.
- Automated process.
- Notify for every build status.
- Fix code if bugs or error found instantly rather than waiting.

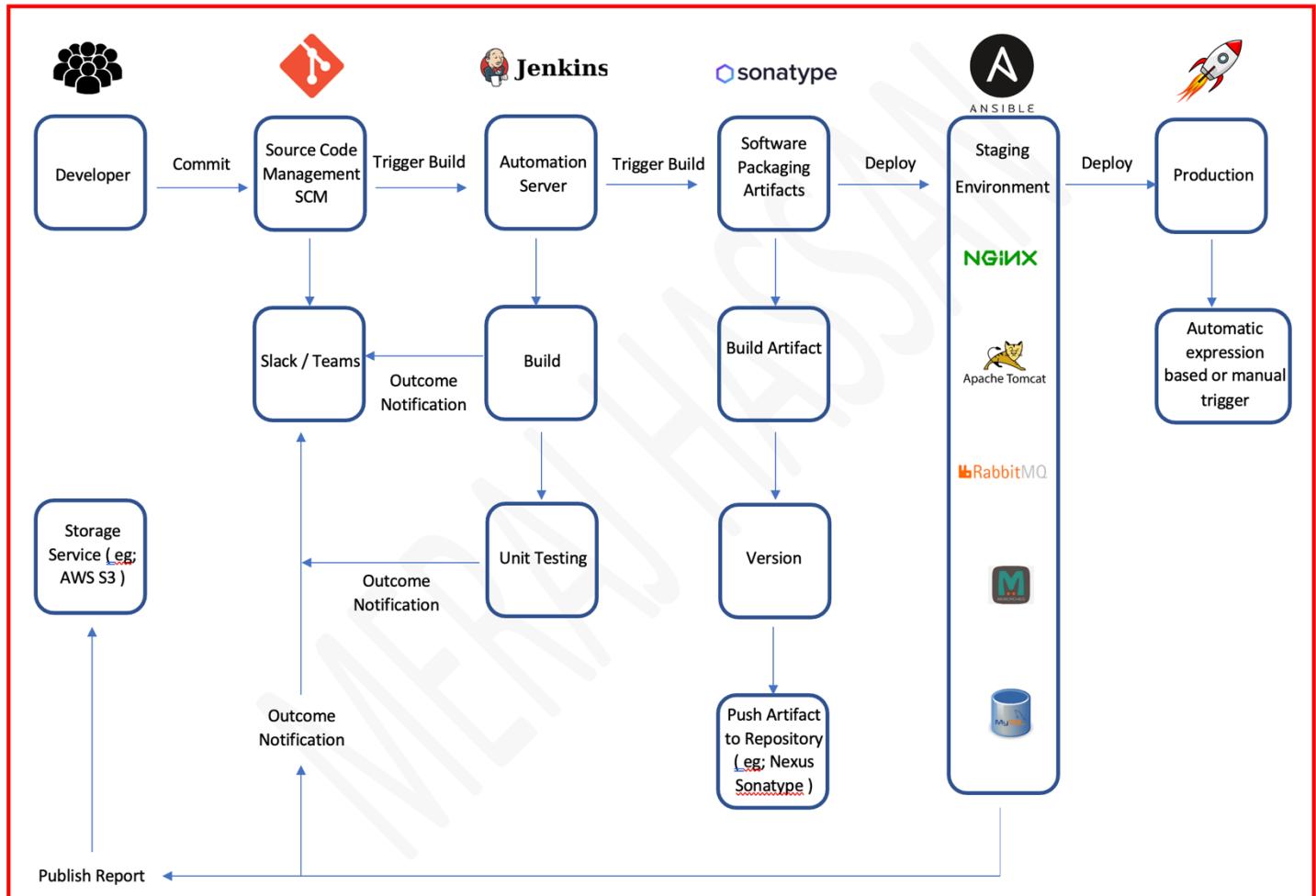
Tools

- Jenkins – Continuous Integration Server
- Ansible – Configuration Management Tool
- GIT – Version Control System
- Maven – Build Tool
- CheckStyle – Code Analysis Tool
- Slack – Notification
- Nexus – Artifact/Software Repository
- Sonarqube – Code Analysis Server
- Tomcat – Java Web Application Server
- AWS – EC2 Resource

Objective

- Automation deployment
- Short MTTR
- Fast turnaround on feature changes
- Less disruptive

Architecture Continuous Integration Pipeline



Flow of Execution

- *Login to AWS Account*
- *Create Login Key*
- *Create Security Group*
 - *Jenkins*
 - *Nexus*
 - *Sonar*
- *Create EC2 instance with user data*
 - *Jenkins*
 - *Sonarqube*
 - *Nexus*
- *Jenkins post installation*
- *Nexus repository setup*
 - *3 Repos*
- *Sonarqube Post Installation*
- *Jenkins Steps*
 - *Build Job*
 - *Setup Slack Notification*
 - *Checkstyle code analysis job*
 - *Setup Sonar integration*
 - *Sonar Code analysis job*
 - *Artifact upload job*
- *Connect all job together with BuildPipeline*
- *Set automatic build trigger*
- *Test with GIT*
- *Create SG*
 - *Windows Server*
 - *Tomcat & Backend Server*
- *Setup tomcat & backend server on ec2*
- *Create Jenkins job to run Ansible Playbook*
- *Deploy artifact to staging by Ansible*
- *Deploy artifact to production tomcat using Ansible*
- *Connect all the jobs with Build Pipeline*
- *Clean-up*

Prerequisite

1. *AWS Account*
2. *GITHUB Account*
3. *IntelliJ*

1. Prerequisite:

- Install and setup Jenkins, Nexus, Sonar and Slack Servers.
- Create CI pipeline to create, check and upload artifact in Nexus repo.
- Add Slack Notification & Quality gates for artifact.

Link: https://github.com/merajafnan/DevOps_Projects/tree/CI-Jenkins

eereeda-jenkins

+ Add a bookmark

SonarScanner-CodeAnalysis - #4 Success after 47 sec (Open)

Jenkins APP 18:03 Slack/Jenkins plugin: you're all set on <http://44.196.57.136:8080/>

Deploy-to-Nexus - #1 Started by user meraj (Open)

Deploy-to-Nexus - #1 Success after 1.6 sec (Open)

Build - #4 Started by user meraj (Open)

18:07 Build - #4 Success after 9.2 sec (Open)

Test - #2 Started by upstream project "Build" build number 4 (Open)

Test - #2 Success after 11 sec (Open)

Integration Test - #2 Started by upstream project "Test" build number 2 (Open)

Integration Test - #2 Success after 13 sec (Open)

Code-Analysis - #5 Started by upstream project "Integration Test" build number 2 (Open)

Code-Analysis - #5 Success after 7.8 sec (Open)

SonarScanner-CodeAnalysis - #5 Started by upstream project "Code-Analysis" build number 5 (Open)

SonarScanner-CodeAnalysis - #5 Success after 47 sec (Open)

Deploy-to-Nexus - #2 Started by upstream project "SonarScanner-CodeAnalysis" build number 5 (Open)

Deploy-to-Nexus - #2 Success after 1.1 sec (Open)

B I S | ⌂ | ⌂ ⌂ | ⌂ ⌂ | ⌂ ⌂ | ⌂ ⌂ | ⌂ ⌂

Send a message to #eereeda-jenkins

+

meraj afnan you

+ Add teammates

Apps

Jenkins

+ Add apps

eereeda-jenkins

37 new messages

✓ • • 🎉 🚀 React ⏱ Reply ⋮

2. Launch Staging Server

Create SG for Backend services, Tomcat app and Windows server

The screenshot shows the AWS Security Groups page with 7 items listed. The columns include Name, Security group ID, Security group name, VPC ID, Description, and Own. The groups are:

Name	Security group ID	Security group name	VPC ID	Description	Own
Windows-Server-So...	sg-0023cd144150696b5	Windows-Server-SoftT...	vpc-083c92be56c3fe99a	Windows-Server-SoftT...	1628
eereeda-sonar-sg	sg-0516d8f616cd82a2e	eereeda-sonar-sg	vpc-083c92be56c3fe99a	eereeda-sonar-sg	1628
eereeda-nexus-sg	sg-0a1f4f35a73982780	eereeda-nexus-sg	vpc-083c92be56c3fe99a	eereeda-nexus-sg	1628
eereeda-jenkins-sg	sg-0c9b468109242dc85	eereeda-jenkins-sg	vpc-083c92be56c3fe99a	eereeda-jenkins-sg	1628
eereeda-backend-st...	sg-0db940876b822615f	eereeda-backend-stagi...	vpc-083c92be56c3fe99a	eereeda-backend-stagi...	1628
eereeda-app-stagin...	sg-0b7f28b5615ff87c2	eereeda-app-staging-SG	vpc-083c92be56c3fe99a	eereeda-app-staging-SG	1628

Create EC2 Instance (Backend Server & Tomcat Server)

The screenshot shows the AWS Instances page with 5 items listed. The columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Available. The instances are:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available
Jenkins-Server	i-068f5cced042c37b0	Stopped	t2.small	-	No alarms	+
app01-staging-eereeda	i-0964e47d5beb3e032	Running	t2.micro	2/2 checks passed	No alarms	+
be01-staging-eereeda	i-0287d71172522f382	Pending	t2.micro	-	No alarms	+
Nexus-server	i-0e50b0f18d510338d	Stopped	t2.medium	-	No alarms	+
Sonarqube-Server	i-0fb090ab6a389124e	Stopped	t2.medium	-	No alarms	+

Create Playbook for Tomcat Server { [DevOps_Projects/ansible/tomcat_setup.yml](#) }

Create Playbook for Deployment (Tomcat Setup and Deploy Artifact) { [DevOps_Projects/ansible/eereeda-app-setup.yml](#) }

GIT URL: https://github.com/merajafnan/DevOps_Projects/tree/cd-ansible-jenkins

3. Ansible for Jenkins

Install Ansible on Jenkins Server : https://docs.ansible.com/ansible/latest/installation_guide/index.html

Manage Jenkins > Plugin Manager > Ansible > Install without restart

New Item > Deploy-To-Staging-Ansible > Copy from 'Build'

The screenshot shows the Jenkins 'Deploy-To-Staging-Ansible' configuration page. It includes sections for Branch Specifier, Invoke Ansible Playbook, Inventory, Content, Credentials, and Add Extra Variables.

Branch Specifier (blank for 'any')
*/cd-ansible-jenkins

Invoke Ansible Playbook
Playbook path ?
ansible/site.yml

Inventory
 Do not specify Inventory
 File or host list
 Inline content
 Dynamic inventory ?

Content
app01-staging ansible_host=172.31.87.39
|
[appsvrgrp]
app01-staging

Credentials ?
ubuntu (app-stage-ssh-login)

Disable the host SSH key check ?

Add Extra Variables

- {{USER}} admin
- {{PASS}} admin123
- {{nexusip}} 172.31.28.180
- {{reponame}} eereeda-release
- {{groupid}} QA
- {{time}} \$TIME
- {{build}} \$ID
- {{eereeda_version}} \$TIME-\$ID.war

Project Deploy-To-Staging-Ansible

This build requires parameters:

TIME

09:09:391_15-06-2022

ID

6

Build

```
TASK [Start tomcat svc] ****
changed: [app01-staging]

TASK [Wait until ROOT.war is extracted to ROOT directory] ****
ok: [app01-staging]

PLAY RECAP ****
app01-staging : ok=23    changed=9    unreachable=0    failed=0    skipped=6    rescued=0    ignored=0

[Slack Notifications] found #4 as previous completed, non-aborted build
[Slack Notifications] will send OnSuccessNotification because build matches and user preferences allow it
Finished: SUCCESS
```

Screenshot

Deploy-to-Nexus > Configure > Trigger parameterized build on other projects

Post-build Actions

Trigger parameterized build on other projects



X

Build Triggers

Projects to build ?

Deploy-To-Staging-Ansible,

✖ Blank project name in the list

Trigger when build is ?

Stable

Trigger build without parameters ?

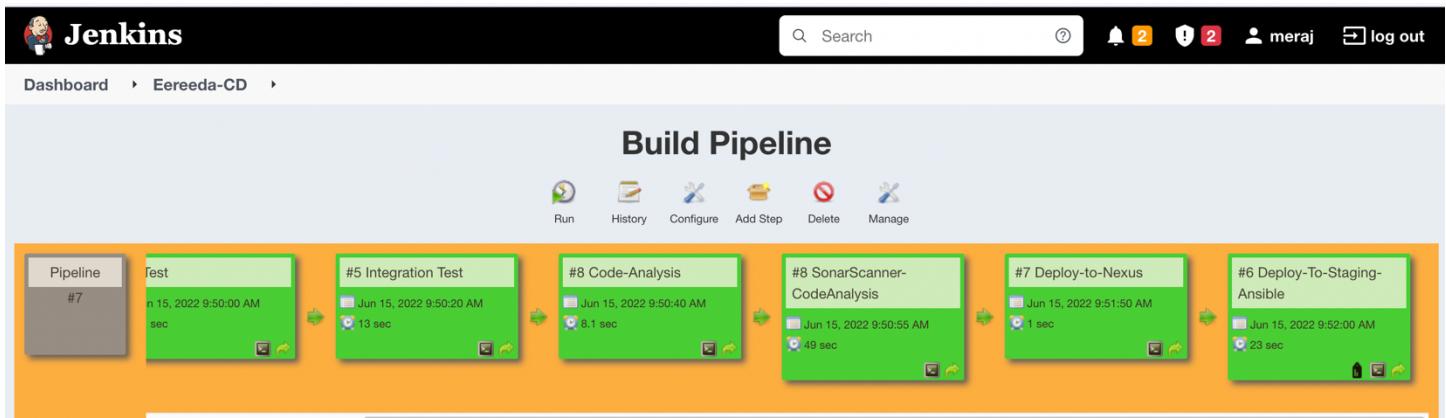
Predefined parameters

X

Parameters ?

TIME=\$BUILD_TIMESTAMP
ID=\$BUILD_ID

New Pipeline View > Eereeda-CD

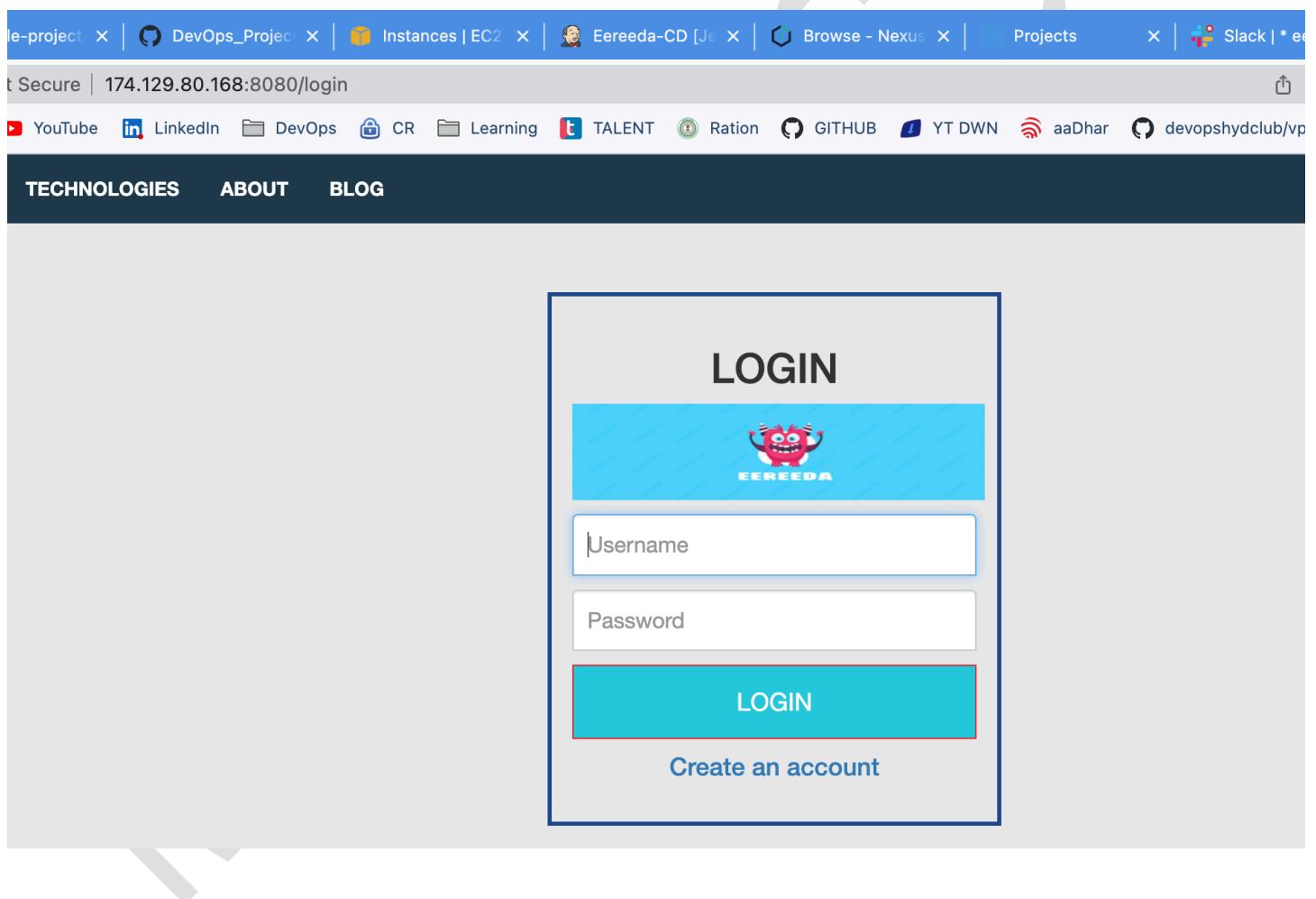


The Jenkins dashboard shows a build pipeline named "Eereeda-CD". The pipeline consists of the following stages:

- Pipeline #7: Test (Jun 15, 2022 9:50:00 AM, 1 sec)
- #5 Integration Test (Jun 15, 2022 9:50:20 AM, 13 sec)
- #8 Code-Analysis (Jun 15, 2022 9:50:40 AM, 0.1 sec)
- #8 SonarScanner-CodeAnalysis (Jun 15, 2022 9:50:55 AM, 49 sec)
- #7 Deploy-to-Nexus (Jun 15, 2022 9:51:50 AM, 1 sec)
- #6 Deploy-To-Staging-Ansible (Jun 15, 2022 9:52:00 AM, 23 sec)

Each stage is represented by a green card with the step name, timestamp, and duration. The pipeline is currently in progress, indicated by the orange background.

Open webpage with public IP of Tomcat Server : **174.129.80.168:8080**



The screenshot shows a browser window with the URL <http://174.129.80.168:8080/login>. The page displays a login form with the following elements:

- LOG IN** at the top center.
- A logo featuring a red cartoon character with the word "EEREEDA" below it.
- Username** input field.
- Password** input field.
- LOGIN** button in a red box.
- Create an account** link at the bottom right.



Jenkins APP 15:19

- Build - #7 Started by user meraj ([Open](#))
- Build - #7 Success after 9 sec ([Open](#))
- Test - #5 Started by upstream project "Build" build number 7 ([Open](#))
- Test - #5 Success after 11 sec ([Open](#))
- Integration Test - #5 Started by upstream project "Test" build number 5 ([Open](#))
- Integration Test - #5 Success after 13 sec ([Open](#))
- Code-Analysis - #8 Started by upstream project "Integration Test" build number 5 ([Open](#))
- Code-Analysis - #8 Success after 8.1 sec ([Open](#))
- SonarScanner-CodeAnalysis - #8 Started by upstream project "Code-Analysis" build number 8 ([Open](#))
- SonarScanner-CodeAnalysis - #8 Success after 49 sec ([Open](#))
- Deploy-to-Nexus - #7 Started by upstream project "SonarScanner-CodeAnalysis" build number 8 ([Open](#))
- Deploy-to-Nexus - #7 Success after 1 sec ([Open](#))
- Deploy-To-Staging-Ansible - #6 Started by upstream project "Deploy-to-Nexus" build number 7 ([Open](#))
- Deploy-To-Staging-Ansible - #6 Success after 23 sec ([Open](#))

B I S ⌂ | ⌂ ⌂ ⌂ | ⌂ ⌂ ⌂ | ⌂ ⌂ ⌂ | ⌂ ⌂ ⌂ | ⌂ ⌂ ⌂

Send a message to #eereeda-jenkins

+ | | Aa

MERAU

1. References

The following table summarizes the documents referenced in this plan.

DOCUMENT NAME	INSTRUCTOR	LOCATION
DevOps Beginners to Advanced	Imran Teli	https://www.udemy.com/course/decodingdevops/learn/lecture/28273912?start=0#overview