EXCELSOFT

# CICD PIPELINE DOCUMENT

**Submitted by Excelsoft Technology**

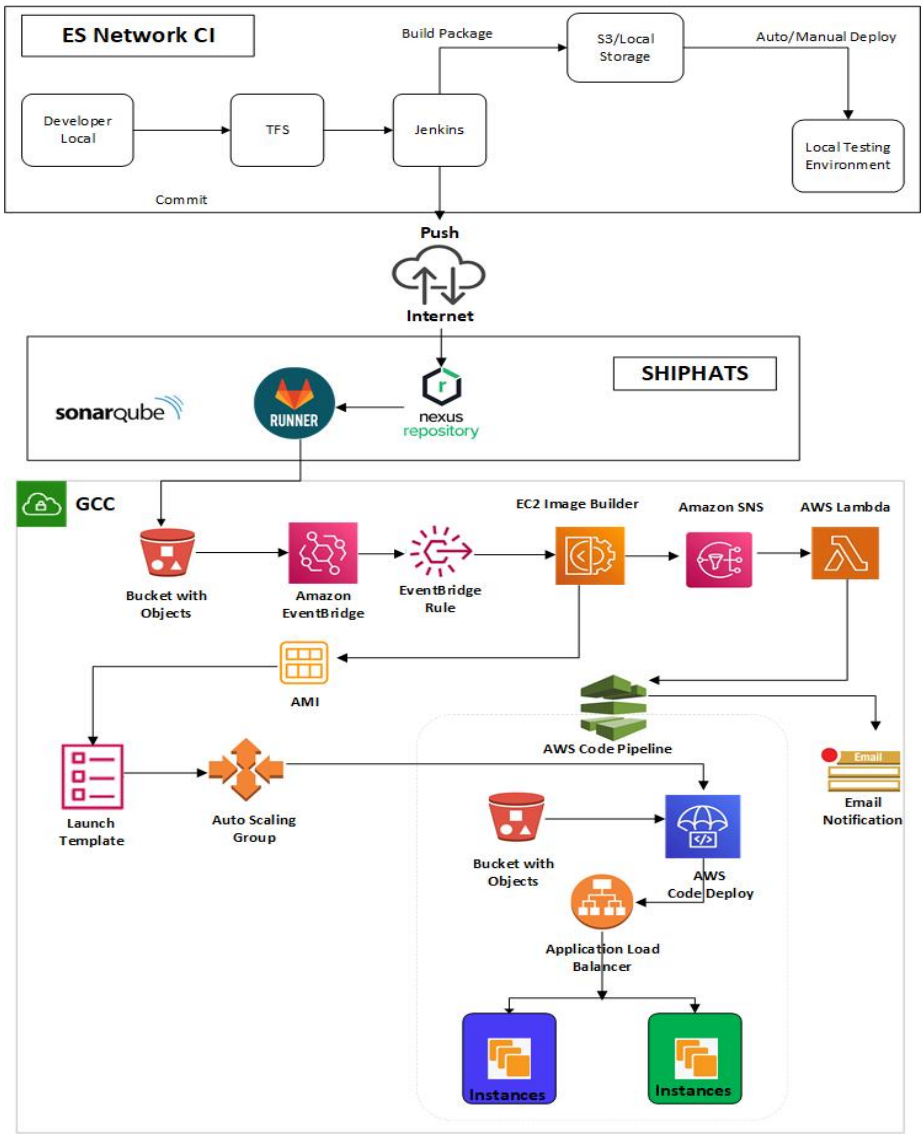
| Version | Date | Author |
|---|---|---|
| 0.7 | 02-Aug-2023 | Sandesh Gowda P |

**CONTENT**

# 1. An Overview of CI/CD Pipeline

## 1. Introduction

This document provides an overview of CI/CD Pipeline configuration to Build, Setup, Test, Release and Deploy the Package/Application for all the modules. This document represents a 'point in time' setup, or snapshot of all the environments such as Testing, SIT, UAT, CAT and Production.

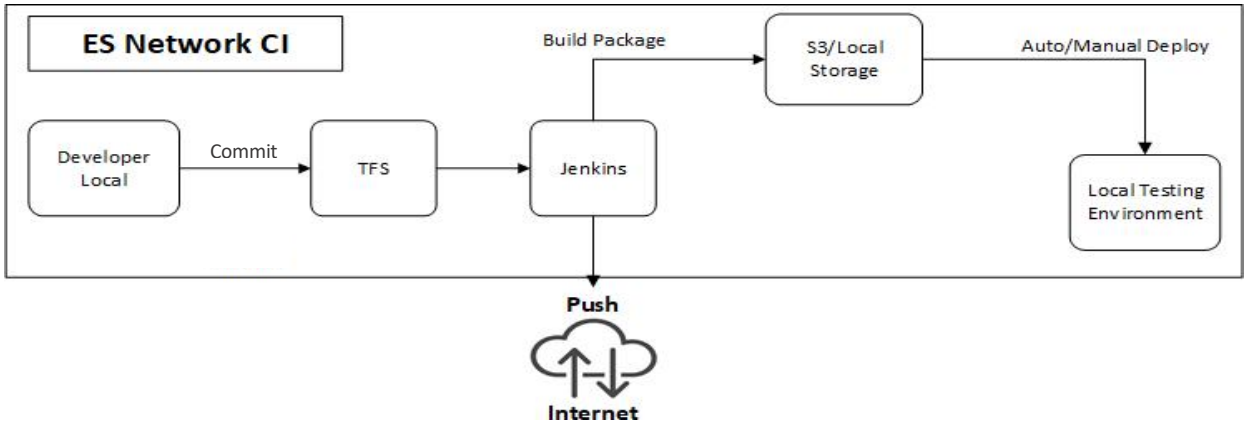



The above diagram explains the complete CI/CD Pipeline flow, and this can be divided into three parts.

- Part1 - Activities that are performed within Excelsoft (ES) using CI/CD tools.
- Part 2 - The tasks that are executed within SHIPHATS.
- Part3 - Activities that are performed within GCC.

-

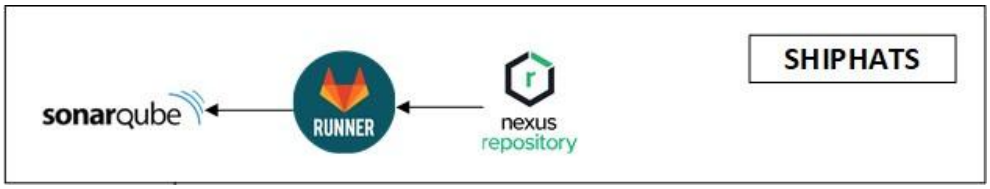
## 2. Activities executed in ES network




**Block Diagram – Part 1**

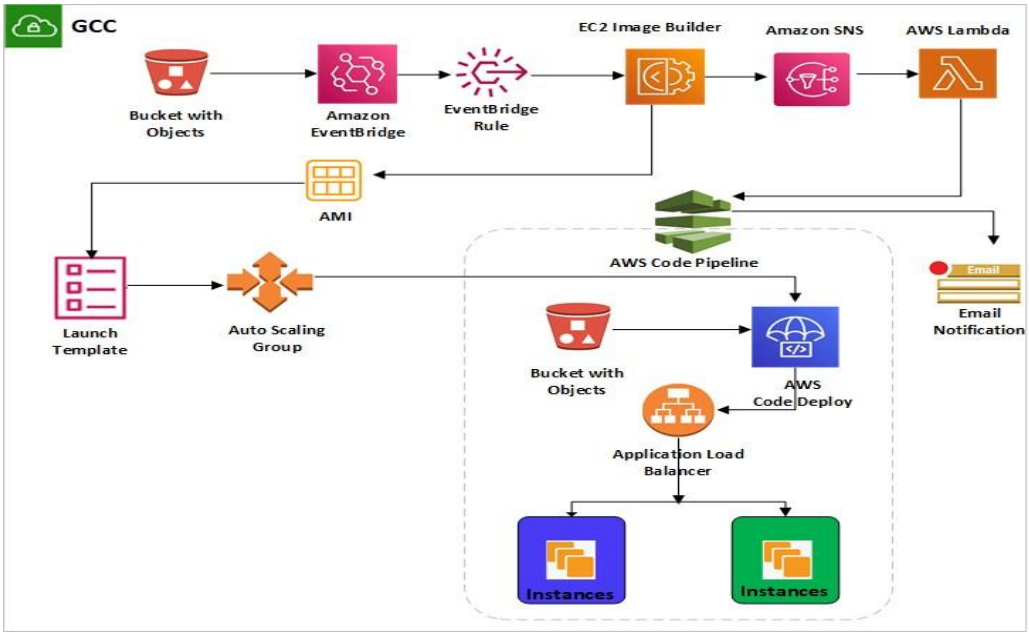
1. The developer performs code changes and checks in them to TFS.
2. CI jobs are configured and executed in Jenkins every 30 minutes to ensure that code changes made by developers have no errors or code breakage.
3. Every night, a scheduled job in Jenkins triggers the nightly build process, collecting the latest source code from TFS to generate the latest package/artifacts.
4. After successful package generation, an automated deployment job will be triggered to deploy the application to the relevant internal servers. In the event of a build failure, an email notification will be sent to the project teams, prompting their intervention the following morning.
5. A copy of the deployed package will be saved in the TFS server with a unique label number. The previously deployed package will be retained as a backup on the respective servers. In the event of any failure, the previous package can be easily deployed from the TFS server or the backup package on the respective server can be used for rollback.
6. The successful build package also triggers the Automation Suite, and the generated report will be emailed to the key team members. Additionally, the QA team executes a smoke test the following day. Upon confirmation, the same build package/artifact will be moved to UAT/CAT based on demand. The build/package with the assigned label number will also be uploaded to the Shiphats nexus repository.

## 3. Activities which are performed within SHIPHATS and CD tools used.




1. Nexus Repo is used to manage the uploaded Artifacts. The latest build package from the ES network will be uploaded to nexus repository manually by using seed laptop.
2. Once the GitLab Runner is implemented, it will be used to upload the package to the S3 bucket which is embedded within in GCC network.
3. SonarQube will be used for Static scan code analysis and will be executed based on requirement.

## 4. Activities which are performed within GCC




1. Once the package reaches from GitLab Runner to S3 bucket, the Event bridge rule will trigger the Ec2ImageBuilder Pipeline.

2. Simultaneously, the output AMI will be updated to the respective Launch Template.

3. The Auto Scaling Group is used to spin up the instances by referring to the latest AMI which is associated in the launch template in order to ensure that the instance are up to date.

4. As soon as the EC2 Image Builder execution completes, the SNS notification will be sent to Lambda function.

5. The Lambda function will trigger the respective AWS Code Pipeline.

6. Code Pipeline will initiate the Blue-Green Deployment process.

7. Switching of the environments will be done via code pipeline.

8. Step1 to Step7 will be repeated for all the modules.

9. The old servers will be terminated after some time (Specified in the Code Pipeline).

10. The same process will be followed in each environment (UAT, CAT, and Prod environments).

# 5. Deployment description

### 4.1 Full Build Deployment

The latest package will be uploaded from ES Network to NexusRepo in Shiphats network via internet, and then the Gitlab runner will be implemented in a such way that it downloads the package from the NexusRepo and push it to the S3 bucket.

Once the above process completes, the pipeline will be triggered in the GCC network to perform the CICD process by using the AWS services.

### 4.2 Incremental Deployment (Manual process)

A new instance will be created from the live instance AMI and incremental patch will be deployed to the new instance.

An incremental patch typically includes updates, bug fixes, security patches, or other changes such as the software running on the instance. This step ensures that the new instance is up to date and includes any necessary improvements.

The application or workload running on the new instance will be be tested and verified.

This testing phase aims to confirm that the patch deployment did not introduce any issues or disruptions and it also confirms that the updated changes are working as expected.

Once the downtime is confirmed, the new instance AMI will be attached to the ASG and deployed to the live environment.