What is CI/CD?

CI/CD automates much or all of the manual human intervention traditionally needed to get new code from a commit into production such as build, test, and deploy, as well as infrastructure provisioning. With a CI/CD pipeline, developers can make changes to code that are then automatically tested and pushed out for delivery and deployment.

What is Jenkins

Jenkins is one of the most popular automation tool used worldwide for continuous integration and continuous delivery.

Why Jenkins?

When working on a project with different teams, developers often face issues with different teams using different CI tools, version management, and other tools. Setting up a CI/CD toolchain for each new project will lead to certain challenges like:

- Slower Releases
- Manual Builds
- Non-repeatable processes
- No Automations

Jenkins is the solution to those challenges. It provides:

- Automated builds
- Automated Tests
- Automated CI/CD pipelines
- Automated deployments
- Ability to install Jenkins locally
- Jenkins support and Plugins

Jenkins is well tested and provide several integrations with 1800+ plugins to support build, deployment and automation for the project.

CI/CD in simple words is a process to take a code, package it up and deploy it to a system that can be serverless, a VM, or a container. CI/CD can be broken down into 3 steps:

- CI – Continuous Integration
- CD – Continuous Delivery
- CD – Continuous Deployment

Key Processes of Continuous Integration

- Package up the code
- Test the code (run unit tests, integration tests, etc)
- Run security checks against the code

Think of the Continuous Integration process like a gift you're wrapping

- The gift comes in pieces
- You put the gift together (maybe a toy chest/box)
- The gift gets wrapped in wrapping paper
- You put it in the car and deliver it to the person.

The basic difference between Continuous Delivery and Continuous Deployment is that in Continuous Delivery to deploy the code after the CI process you have to manually trigger it via some button to deploy on the system whereas in Continuous Deployment this process is automatic.

Key Pieces of CD:

- Ensure you're authenticated to the system or wherever you're deploying
- Ensure that the code that's being deployed is working as expected once it's deployed

Installing Jenkins

https://www.jenkins.io/doc/book/installing/linux/

https://www.jenkins.io/doc/book/installing/linux/#fedora

Jenkins Plugins

Plugins are used in Jenkins to enhance Jenkins functionality and cater to user-specific needs. It allow us to connect one service to other services and work with other products.

Install Plugins

To install a new plugin in Jenkins

1) Go to Manage Jenkins -> Manager Plugins

2) Click Available and search for the desired plugin.

3) Select the desired plugin and Install.

Note: Few plugins may need a restart

To restart Jenkins

$ sudo systemctl restart jenkins

Jenkins Jobs

Different types of jobs that can be created in Jenkins:

Freestyle project

This is a central feature of Jenkins. It will build the project, combine SCM with the build system. It can also be used for things other than building applications.

Pipeline

This is used to create a pipeline

Multi-configuration project

This is great if you need a large number of Jenkins configurations if you need multiple environments like Dev/ UAT.

Folder

This creates containers and stores nested items. It is useful in grouping, creating a namespace, etc.

Organisation folder

Creates a multibranch project for all different subfolders that are available.

Multibranch Pipeline

It sets up pipeline projects for different repositories.

Jenkinsfile

Jenkinsfile is a text file that contains definitions. This could be templates or instructions. It tells pipelines what they should be doing and what services and plugins they should be interacting with.

Components of Jenkinsfile:

Pipeline – The task you are trying to accomplish

Build Agent –The place where you run your pipeline

Stages – Staging/Production/UAT

Steps –Work done in the pipeline

Build Agents

Build Agents are systems that run the processes throughout the pipelines. Build agents help in building codes, deploying, and running automated tests. It is a system that runs the entire workload.

Jenkins Security

Jenkins access control is split into two parts:

1. **Authentication** (users prove who they are) is done using a security realm. The security realm determines user identity and group memberships.
2. **Authorization** (users are permitted to do something) is done by an authorization strategy. This controls whether a user (directly or through group memberships) has a permission