DEVOPS

Training Material

DEVOPS TOOLS AND AUTOMATION

TABLE OF CONTENTS

1. JENKINS TOOL:

- Jenkins is an open-source automation server that is commonly used in DevOps for ***Continuous Integration (CI) and Continuous Deployment (CD).*** Jenkins allows developers to ***build, test, and deploy their applications automatically***, therebyreducing the time and effort needed for these tasks.
- Jenkins is highly extensible, and there are thousands of plugins available that can be used to customize the build and deployment process.
- Jenkins also supports integration with various tools used in the DevOps toolchain, such as ***Git, GitHub, Jira, Docker, Kubernetes***, and many others.
- Jenkins works by executing a ***series of jobs*** that are defined in the Jenkins file.
- The jobs can be configured to perform tasks such as compiling code, running unit tests, packaging the application, deploying it to a server, and sending notifications.

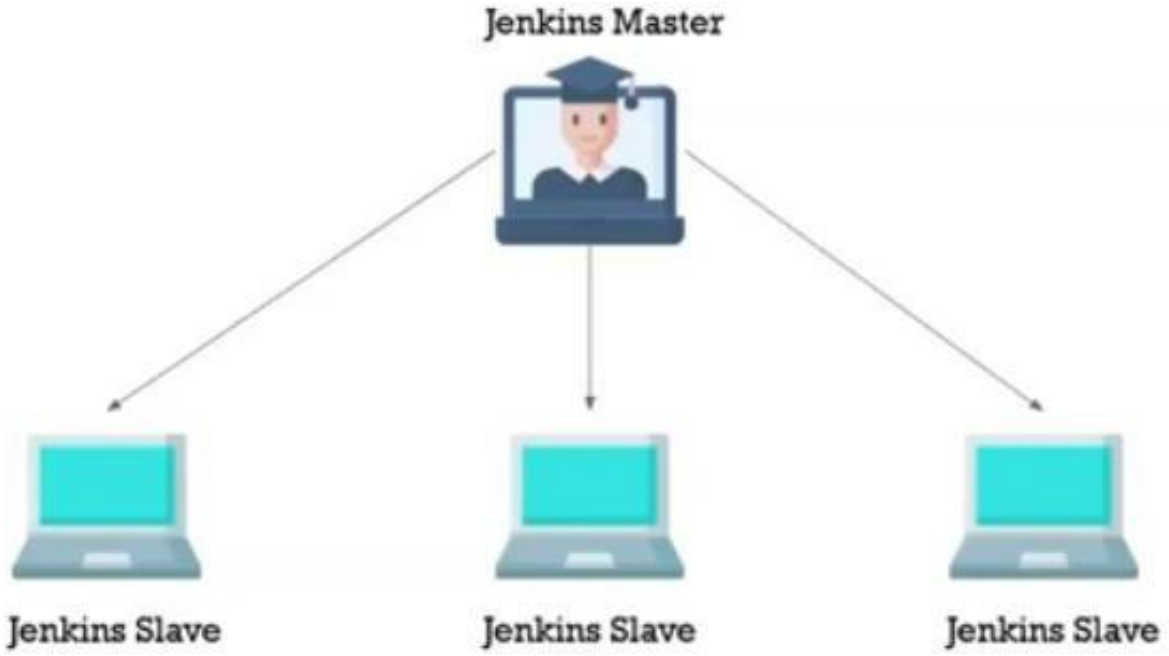
WORKING IN JENKINS:

1. **Install and set up Jenkins:**

Once installed, you can configure Jenkins by creating a new job, setting up source code management, and defining build triggers.

2. **Create a new job:**
   - In Jenkins, a ***job represents a task*** or a step in the build process.
   - To create a new job, go to the Jenkins dashboard, click on "***New Item,"*** enter a name for the job, and select the type of job you want to create.
3. **Configure the job:**
   - Once you create a job, you can configure it by defining the ***build steps, build triggers***, and other parameters.
   - For example, you can define the source code repository, build tool, and target platform for your job.
4. **Run the job:**
   - After configuring the job, you can run it manually or set up ***automatic triggers*** to run it based on ***certain conditions***.
   - Jenkins will execute the build steps defined in the job and report the results.
5. **Monitor the job**:
   - Jenkins provides ***real-time feedback*** on the ***status*** of the build process.
   - We can monitor the job by checking the ***console output,*** build history, and build artifacts.

JENKINS ARCHITECTURE



COMPONENTS IN JENKINS

- **Master Server:**
  - ✓ Controls Pipeline
  - ✓ Schedules Build
- **Agents and Minions:**
  - ✓ Performs the Build

AGENTS:

- Commit Triggers Pipeline
- Agent selected based on Configured Labels
- Agent runs Build

Types of Agents:

1. **Pernament Agents**

Dedicated servers for running Jobs.

2. **Cloud Agents:**

Dynamic Agents spun up on demand

BUILD TYPES:

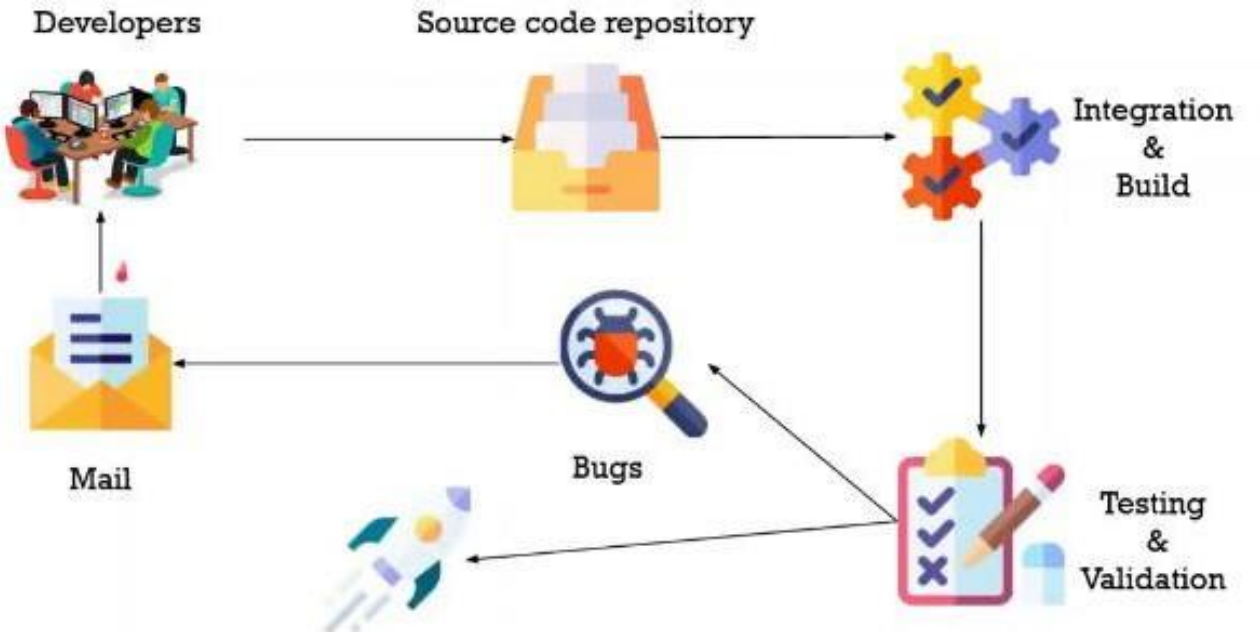
1. Freestyle Build:

- Simplest method to create a build
- "Feels" like Shell Scripting

2.Pipelines:

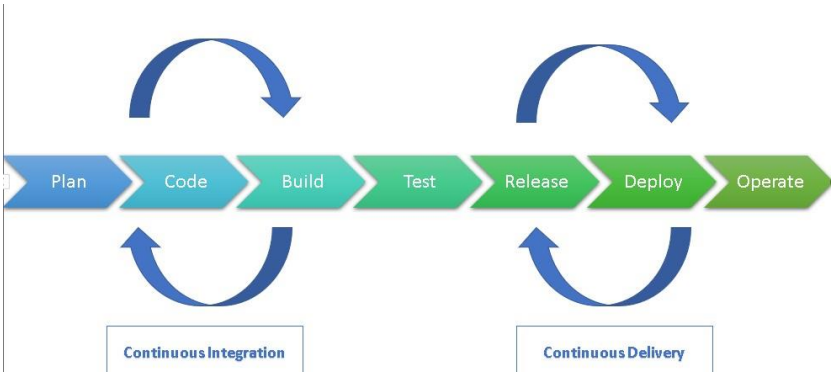
- Use the Groovy Syntax
- Use Stages to break down components

PROCESS BEFORE JENKINS(CI)



PROCESS AFTER JENKINS(CI)

OVERVIEW ABOUT CI/CD Pipeline



CI/CD Pipeline

- A ***Continuous Integration/Continuous Delivery (CI/CD)*** pipeline is a framework that emphasizes iterative, reliable code delivery processes for agile DevOps teams.
- It involves a workflow encompassing continuous integration, testing, delivery, and continuous delivery/deployment practices.
- The pipeline arranges these methods into a unified process for developing high-quality software.
- Test and build automation is key to a CI/CD pipeline, which helps developers identify potential code flaws early in the software development lifecycle (SDLC).
- It is then easier to push code changes to various environments and release the software to production.
- Automated tests can assess crucial aspects ranging from application performance to security.
- In addition to testing and quality control, automation is useful throughout the different phases of a CI/CD pipeline.

It helps produce more reliable software and enables faster, more secure releases.

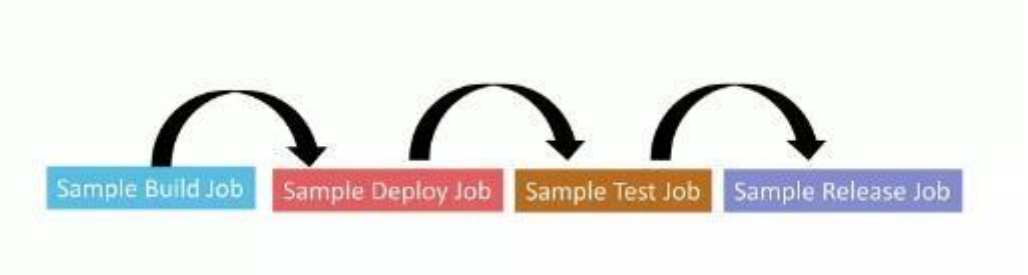
STEPS TO CREATE CI-CD PIPELINE

Step 1: Chain required jobs in sequence Add Upstream/downstream jobs

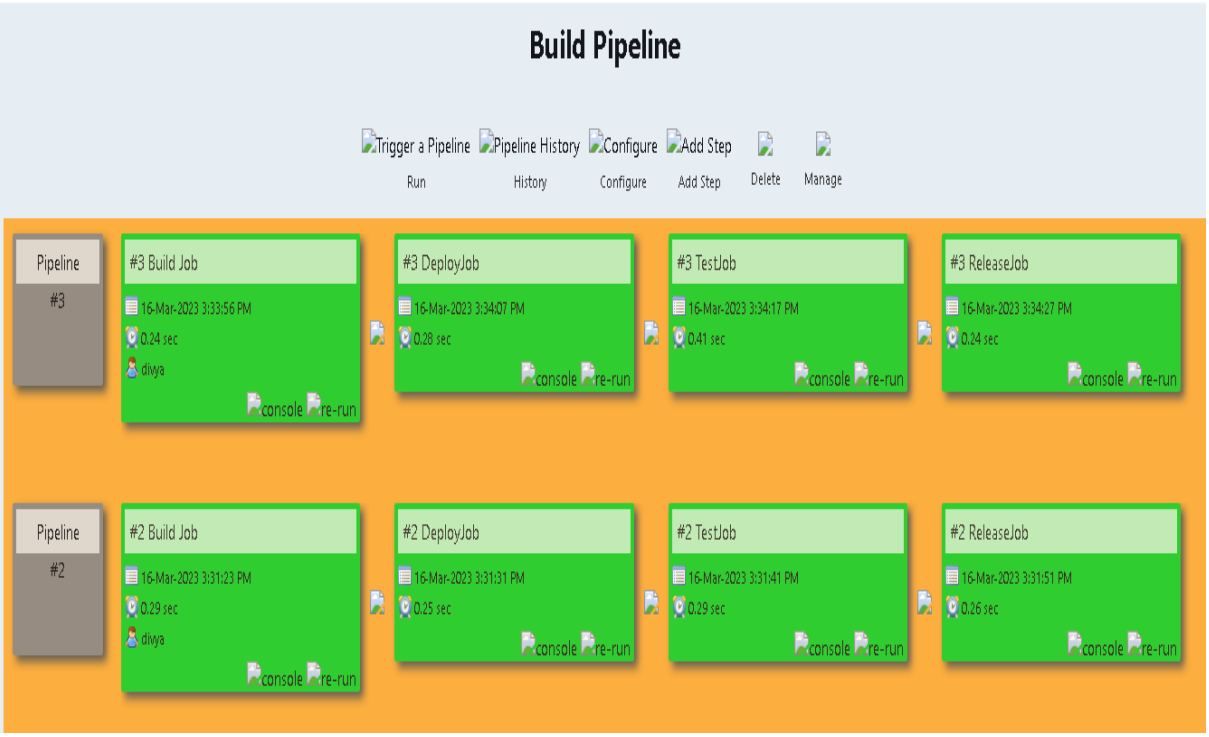
Step 2: Install Build Pipeline and Delivery Pipeline Plugin

Step 3: Add, Build Pipeline View and Configure the view step

Step 4: Run and Validate



BUILD PIPELINE



DELIVERY PIPELINE

STEP BY STEP PROCEDURE:

CREATING JOBS

- Create a New Item-BuildJob

- Create a Freestyle Project -OK
- Build Steps – Execute shell
- Type the commands
- Example: date
    - **echo** “Build process Successfully done”
- Apply and Save
- Proceed for the remaining three Jobs [Deploy, Test and Release]

CREATE CHAIN

- Open each Job
- Configure -> Build Triggers -> Build after other projects is built →----------
- Apply and Save
- Open the Each job in a separate window
- Run (Build Now) the First Job
- Automatically other three jobs are executed

PLUGIN INSTALLATION:

- Manage Plugin ->Install Plugin ->
- Build Pipeline Plugin
- Delivery Pipeline Plugin
- Install without restart

CREATE PIPELINE

- Dashboard -> All -> +
- Name of the Pipeline
- Select Build Pipeline View
- Pipeline Flow -> Initial job -> Build Job
- Display option -> 5
- OK

CREATE PIPELINE DELIVERY

- All ->+
- Name -> Delivery Pipeline View
- Add Component
- Name and Initial Job