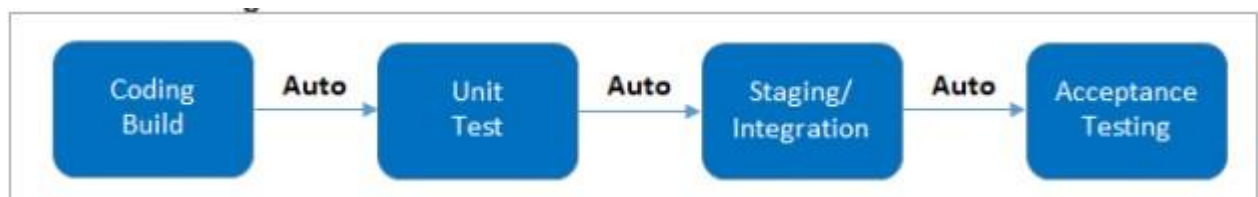


# JENKINS

## What is Jenkins

1. Jenkins is an open-source automation server
2. Jenkins is used for continuous integration (CI) and continuous deployment (CD) in software development
3. It helps to automate the process of building, testing and deployment.
4. It can consist of plugins
5. It was developed by sun micro systems but Oracle buy from them and they named as Jenkins

## What is continuous integration (CI)



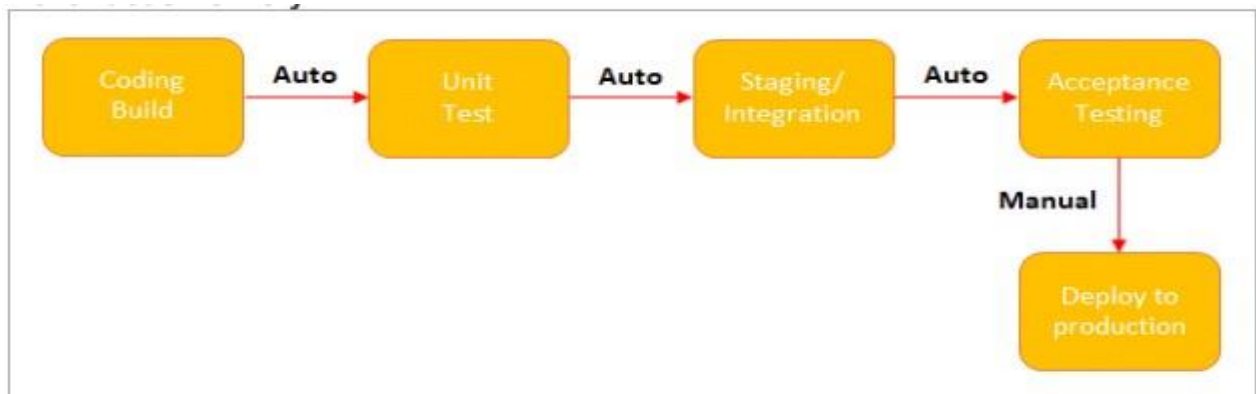
Continuous integration means when developer can release the code for build and test, if the code can fail to build then that code can be return to the developer and after modifying the code it will again perform the build, if code failed in any stage, it can repeat the process until code get success

## What is continuous deployment (CD)

There are two types of deployments

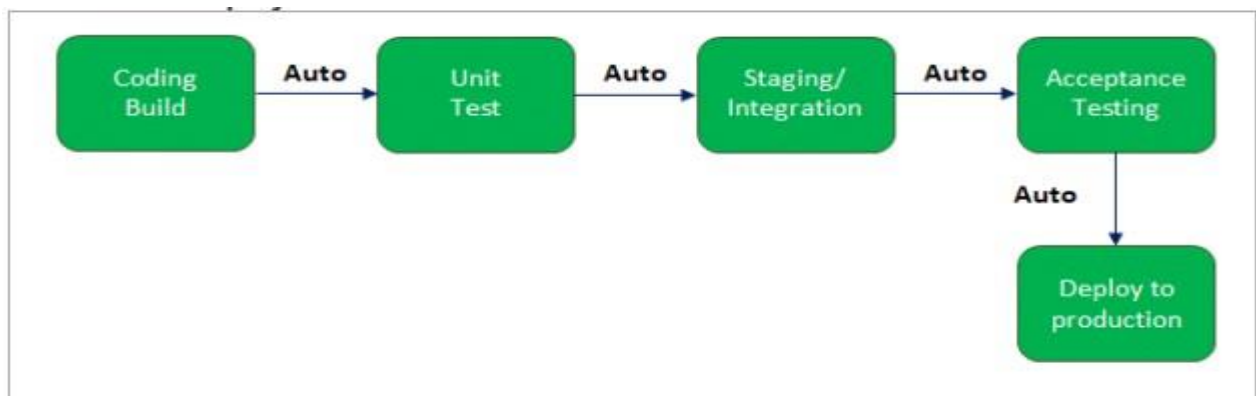
1. Continuous delivery
2. Continuous deployment

## Continuous delivery:



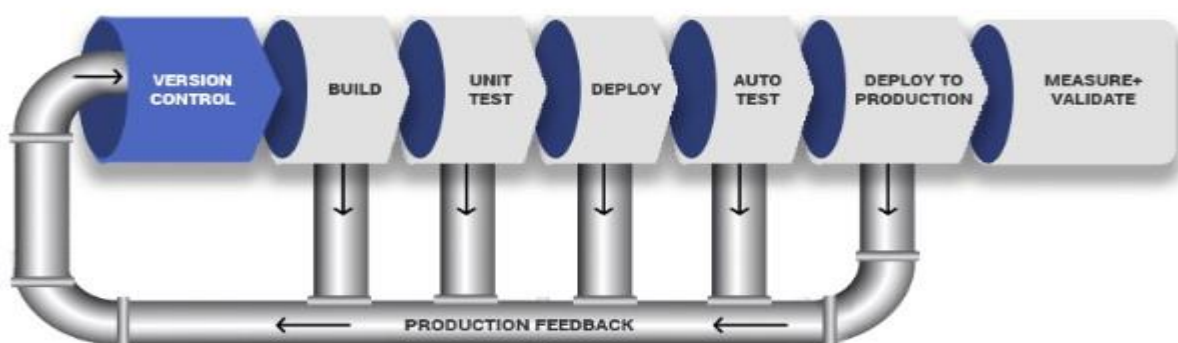
In continuous delivery the production deployment is done by manually

## Continuous deployment:



In continuous deployment the production deployment is done by automatically

## **what is pipe line in (CI/CD)**

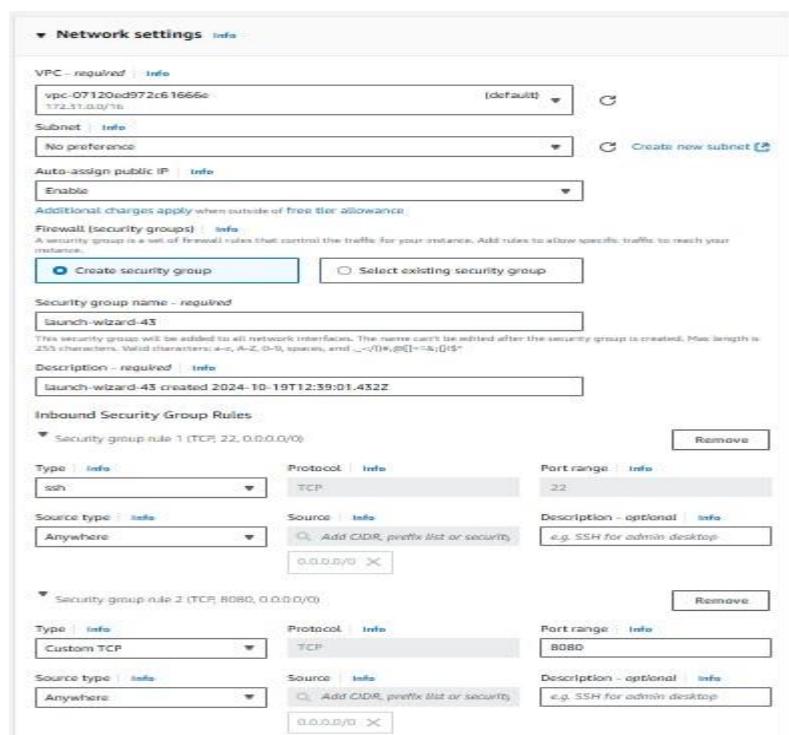


In (CI/CD) pipeline after developing the code from developers it can perform the build stage, if build is success, then the code will move to unit test, if unit test is success, then the code will move to deploy stage, if deployment get success, then the code will move to auto test if auto test is done then code will deploy to the production if the code is fail in any stage then the code can return to the developer and again starts from first step and after that we will monitor the code.

## Steps to create Jenkins server

### Step 1:

1. Launch the instance with ubuntu
2. Click on launch instance
  - Give the name for instance
  - Select the AMI as ubuntu
  - Select the created key pair
  - Click on edit for network settings in that click on add security group rules with port number 8080



The screenshot displays the 'Network settings' configuration page for an AWS instance. It shows the VPC (vpc-07120ed972c61666e), Subnet (No preference), and Auto-assign public IP (Enable) settings. The Firewall (security groups) section is active, showing a security group named 'launch-wizard-43'. Under 'Inbound Security Group Rules', two rules are defined: Rule 1 for SSH (port 22) and Rule 2 for Custom TCP (port 8080). Both rules allow traffic from 'Anywhere' (0.0.0.0/0).

Type	Protocol	Port range	Source type	Source	Description
ssh	TCP	22	Anywhere	0.0.0.0/0	e.g. SSH for admin desktop
Custom TCP	TCP	8080	Anywhere	0.0.0.0/0	e.g. SSH for admin desktop

- Finally click on launch instance

## Step 2:

1. Select the created instance and click on connect and connect directly to the terminal
2. After connecting to the Jenkins terminal go through with some command

Commands	
<code>sudo -i</code>	To switch root user
<code>apt update -y</code>	To update the server
<code>apt install openjdk-17-jre -y</code>	To install java packages
<code>java --version</code>	To check the version of java
<code>apt install maven -y</code>	To install maven packages
<code>mvn -v</code>	To see the version of maven
<code>mvn clean package</code>	To get build packages

3. Install Jenkins package in terminal
4. Type Jenkins.io in new web page
  - Select the installing Jenkins
  - Select the Linux option
  - Copy the path `sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debianstable/jenkins.io-2023.key` and paste in terminal
  - Copy the second path `echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null` and paste in terminal
  - Type `apt-get update -y` in terminal to update server
  - Type `apt-get install Jenkins` in terminal to install jenkins

- Type `systemctl start Jenkins` in terminal to start the Jenkins server
- Type `systemctl status Jenkins` in terminal to check the status of Jenkins

### Step 3:

1. Copy the Jenkins public IP and paste the public IP with port number :8080 in new web page
2. It will connect Jenkins server as shown in below figure



3. Go to Jenkins terminal
4. Copy the key in terminal as shown in below figure

```

└─3914 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/
Oct 19 14:24:46 ip-172-31-6-140 jenkins[3914]: Fe776147b4974c56ab730c032b845a51
Oct 19 14:24:46 ip-172-31-6-140 jenkins[3914]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 19 14:24:46 ip-172-31-6-140 jenkins[3914]: .....
Oct 19 14:24:46 ip-172-31-6-140 jenkins[3914]: .....
Oct 19 14:24:46 ip-172-31-6-140 jenkins[3914]: .....
Oct 19 14:25:08 ip-172-31-6-140 jenkins[3914]: 2024-10-19 14:25:08.756+0000 [id=45] INFO h.m.DownloadService
Oct 19 14:25:08 ip-172-31-6-140 jenkins[3914]: 2024-10-19 14:25:08.757+0000 [id=45] INFO hudson.util.Retrie
Oct 19 14:25:15 ip-172-31-6-140 jenkins[3914]: 2024-10-19 14:25:15.966+0000 [id=30] INFO jenkins.InitReacto
Oct 19 14:25:15 ip-172-31-6-140 jenkins[3914]: 2024-10-19 14:25:15.992+0000 [id=22] INFO hudson.lifecycle.L
Oct 19 14:25:15 ip-172-31-6-140 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
lines 1-20/20 (END)

```

5. Otherwise, to copy the key we use command

Command	
<code>cd /var/lib/jenkins</code>	It is default path in jenkins
<code>ll</code>	It is used to list the files
<code>cd secrets</code>	To change directory

ll	To list the files
cat initialAdminpassword	To see password

6. After seeing the password just copy the password and paste in jenkins server in administrator password block and click on continue
7. Select the install suggested plugins
8. Provide user name as user and password as jenkins and confirm password as created password and full name as optional and email as any email and click on save continue
9. Click on save and finish
10. Then you see jenkins dashboard

## Step 4:

1. Click on new item in jenkins dashboard to create a new job with freestyle or pipeline it is optional
2. Give name as job and select free style and click on ok option as shown in below figure

### New Item

Enter an item name

Select an item type



#### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

3. Select git in source code management
  - Copy the URL of java code in github
  - No need to provide credentials for git as shown in below figure

**Git** ?

**Repositories** ?

Repository URL ? ✕

`https://github.com/Abeed2024/project.git`

Credentials ?

- none - ▼

+ Add

Advanced ▼

Add Repository

- Provide main in branch specifier block

**Branches to build** ?

Branch Specifier (blank for 'any') ? ✕

`*/main`

Add Branch

**Repository browser** ?

(Auto) ▼

- Select invoke top-level maven targets in add build steps
- Give clean package in goals block as shown in below figure

**Build Steps**

≡ **Invoke top-level Maven targets** ? ✕

Goals

`clean package` ▼

Advanced ▼

- Click on apply and save

- Click on build now option then it will show build is success or failed

## Step 5:

1. Click on dashboard
2. Click on new item and create a job with pipeline and click on ok
3. Select pipeline script and select hello world then sample script is created
4. Click on pipeline syntax
5. Select git in sample step
6. Paste the github URL in repository URL
7. Give main in branch
8. Click on generate pipeline script and copy the URL as shown in below figure

Sample Step

git: Git

git ?

Repository URL ?

`https://github.com/Abeed2024/project.git`

Branch ?

`main`

Credentials ?

`- none -`

+ Add

☒ Include in polling? ?

☒ Include in changelog? ?

**Generate Pipeline Script**

`git branch: 'main', url: 'https://github.com/Abeed2024/project.git'`



9. Paste the copied URL in pipeline script as shown in below figure

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('checkout') {
6       steps {
7         git branch: 'main', url: 'https://github.com/Abeed2024/project.git'
8       }
9     }
10  }
11 }
12
```

Hello World

☒ Use Groovy Sandbox ?

10. Click on apply and save

11. Click on build now option it can show the build is success are failed

12. Create pipeline script for build

13. Click on pipeline script

14. Select the sh: shell script in sample step

15. Give mvn clean package in shell script

16. Click on generate pipe line script then script is generated

17. Copy the generated script as shown in below figure

Steps

Sample Step

sh: Shell Script

sh

Shell Script ?

mvn clean package

Advanced

Generate Pipeline Script

sh 'mvn clean package'

18. Paste the copied URL in pipeline script as shown in below figure



19. Click on apply and save

20. Click on build now option it can show the build is success or fail

21. If build is success or fail that can be seen in build history as shown in below figure

