

Jenkins

CI and CD → continuous integration and continuous delivery.

Jenkins is a open source automation tool written in java.

Continuous Integration:-

The practise of automatically building a code periodically is called continuous integration.

continuous delivery)

releasing software at any time. It aims at building, testing and releasing software faster and more frequently.

- Reduce Risk.
- Increased confidence.
- Better quality code.
- Ready to ship code.
- Systematic variation.

SDHc and Devops culture

Manual integration



Jenkins feature and architecture



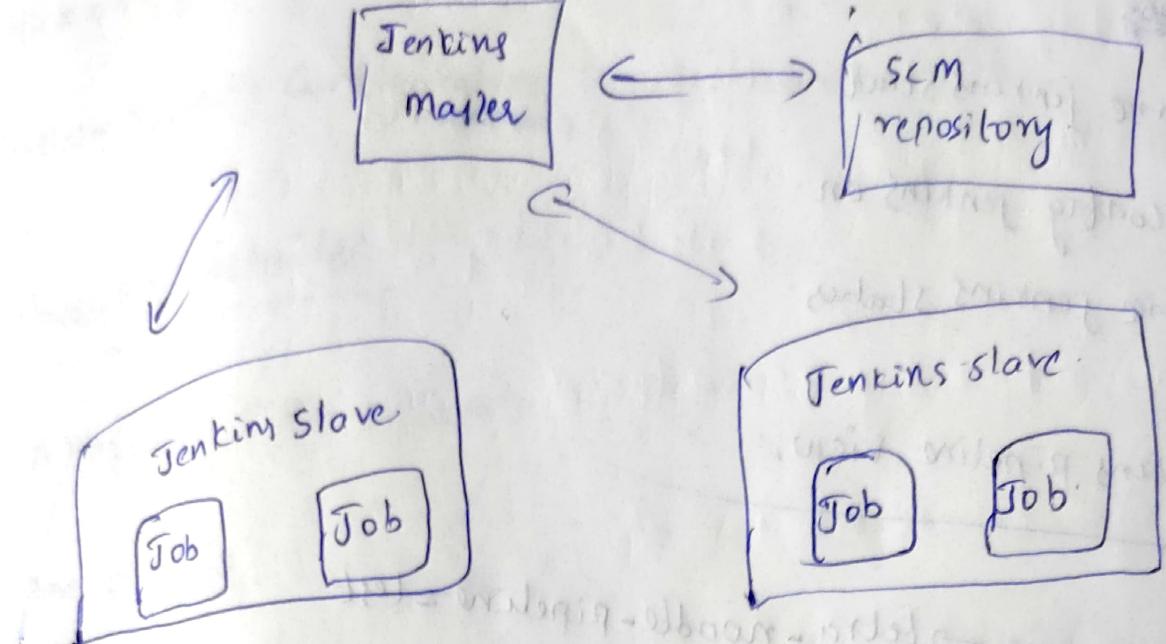
Implement CI and CD

Features

- ① Increased productivity
- ② open source platform for implementing Devops pipeline
- ③ cross platform
- ④ Extensible with Plugins
- ⑤ tried and Tested

How does Jenkins works?

Distributed architecture



Jenkins use master-slave distributed architecture

In this architecture master and slave communicate with TCP and IP protocol.

The main jenkins server in distributed architecture is master server

The master job is to pull SCM repository from git.

→ DevOps practices Bring agility in the SDLC

→ Reduce Risk and Increases Quality

CI and CI pipeline with Jenkins, Gradle.

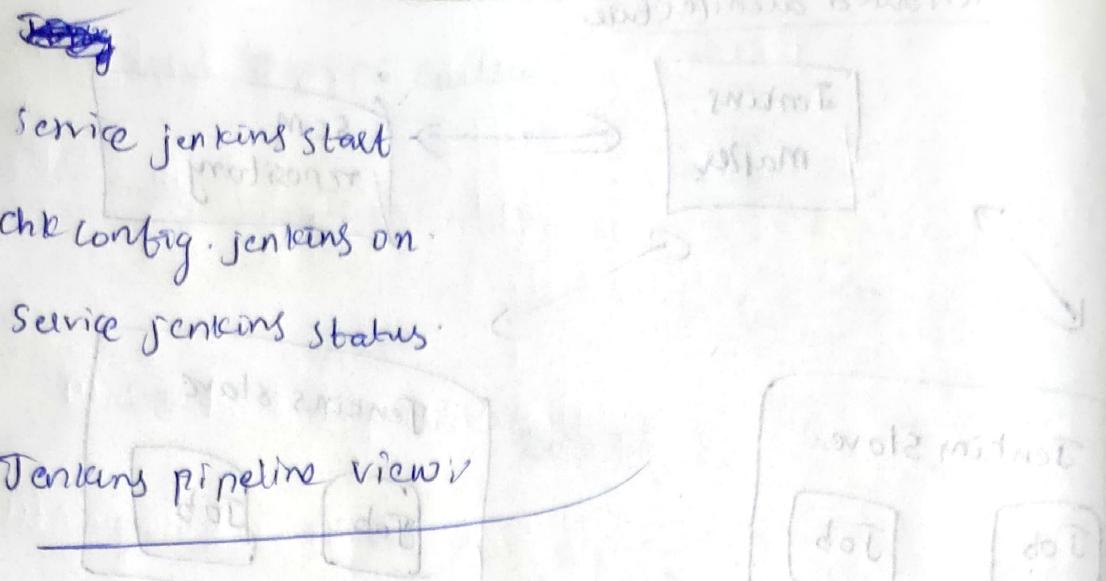
and Artifactory

Java Rest API

Git SCM - BitBucket

Repository manager - Artifactory

↓ slow tools



View name → tetra-noodle-pipeline-test

• Build pipeline view:

Configuring artifactory

cd /nirivesh/etc/

Installed.

Artifactory plugin

1. Connect.

Configure more.

Gradle.

↓
Gradle-Artifactory Integration

Artifactory deployment server: <http://art.noodle.tetra>

: 8081/artifactory

click on Refine

Custom staging configuration: ext-release-local

Resolution details:

Artifactory resolve server

more details

capture and publish build info.

publish artifacts to Artifactory

Build

Add Build Step

↳ flow of build



Invoke Gradle Script

Gradle
executable for
bin directory

Jenkins Home page

↓
manage Jenkins

↓
global tool config

↓
Gradle

Install automatically

CI/CD with Jenkins and maven

Build system - maven

- Git SCM - Bitbucket
et. ↳ address
1. how SW is build.
2. its dependency

Maven hosted by apache tomcat.

Deployment - Apache Tomcat.

Jenkins



New items



Free style project

Discard old builds

Source code management

git

Repository URL: https://bitnami-node@bitnami-test.org

Build Environment

Delete workspace before build starts

Build

```
graph TD; A[Add Build Step] --> B[ ];
```

Invoice ~~noten~~ Artifacts noten?

provided by ~~antibody~~
~~antibody~~ plugin

Invoice top-level marsh targets -

↓
provided by part of standard
surfing distribution.

mvn package

mv n - version.

Jenkins → manage Jenkins → Global Tool Configuration

Maven → Add maven
Install automatically
Name :

post-build actions to change our actions in pipeline

Post-build actions

Achieve the artifacts

Files to archive | mult03/target/war

Create peptide

View name: Wal-fetra-noodle-deployment

Show pipeline project headers Yes

Show pipeline parameters in project Yes

Show pipeline parameters in revision box Yes

Jenkins

Manage Jenkins

Manage Plugins

Copy artifact plugin

Install without restraint

Configure

Post-build actions

Deploy war/ear to a container (path of war/ear file)

war/ear files **/*

Content path : http://tomcat-node-tetra-8080/

File : tetra-node-app-\$BUILD_NUMBER.DEB

Container 'Tomcat 7'

chain projects in pipeline:-

Implement Java web Application.

Installation and configuration of Jenkins Plugins

update Jenkins job

Deploy artifacts in to apache tomcat

Relational Database Schema:-

~~saatchi tool~~

Saatchi

↓

Database change management framework

It keeps tracking all these changes

↳ Saatchi init

↳ initializes saatchi project.

<http://saatchi.org/>

Build.

Execute shell

Command : cd \$WORKSPACE

Saatchi deploy db pg deploy:deploy 192.168.0.102:5432

build pipeline

↳ install-build pipeline for saatchi

View name db-fetranode

click on build pipeline

revet command to roll back

Create new job

db-verify



source code management

Build

execute shell

Command (d) Workspace

Sqitch verify db:pg:deploy@192.168.0.103:5432

Configure



Postbuild action



db-verify

click
on

Run to Run - Build pipeline

Norqress communicates only ~~with~~ with TCP/IP.

It does not connect with Jenkins

We need to update the server

with configuration

Jenkins security :-

Securing Jenkins:

manage Jenkins

JNLP

Jar network launch protocol.

Configure Global Security

Select fixed and specify port

Access control

Security realm with user manager

Jenkins own user database

LDAP

Light weight directory Access protocol.

It manages application protocol
and access distributed directory

Information Service

allows to secure jenkins

unix-usergroup database

↓

used authentication of users

Authorization:

1. Any one can do anything
 - ↳ anonymous user → not recommended

2. Legacy mode:
 - ↳ only administrator will can manage and access configuration with Jenkins

3. logged-in users can do anything
 - ↳ not recommended
 - ↓ Jenkins forces the user to log in.

matrix-based Security

is for entire Jenkins infrastructure

5.1 ~~project based~~ matrix authorization

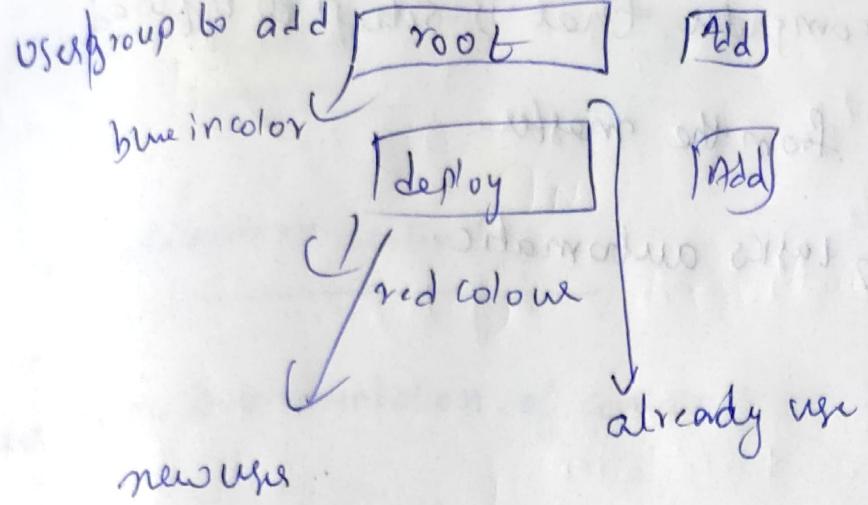
↳ ~~Strategy~~ security → can allow to ~~access~~

↳ security access for specific projects

Both are similar in functionality

For small scaled clusters

matrix-based Security is good option.



Manage Jenkins

↓
manage user

↓

create user

Create User:

Username :

Password :

Confirm password:

Full name :

Email :

Scaling Jenkins:

Scaling the System by using Slave nodes -

Scaling allows single ~~slave~~ Jenkins installations to host to multiple projects.

A master operating by itself is the most basic installation of Jenkins.

The master handles all tasks for build system.