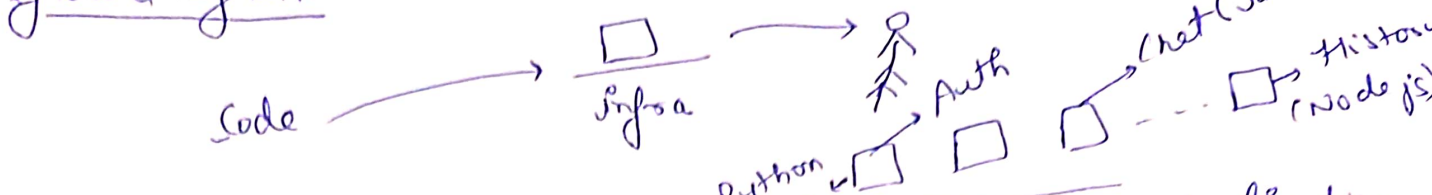
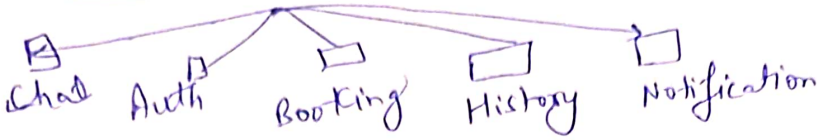
Docker Compose & Practical — Day 13

Flow Diagram.



. Micro Services Architecture.

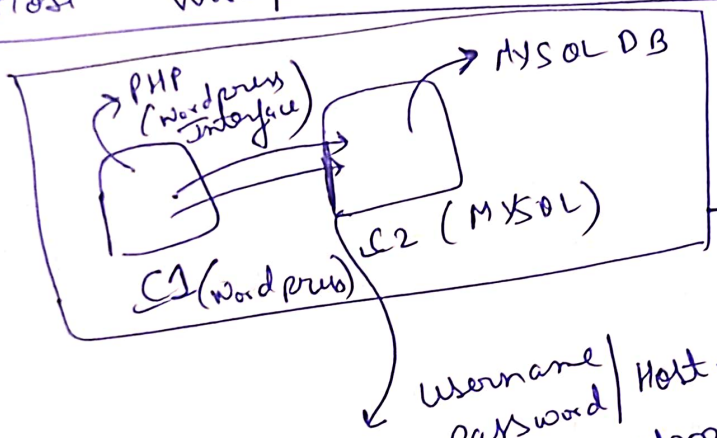


Launch New Instance, t2.medium

Wordpress Image Pull

Custom N/W, Custom Volume

Host Wordpress Server in EC2 Instance.



⇒ Need to create a volume and Mount that volume to MySQL container. (So that No Data lost) (/var/lib/mysql).

⇒ docker volume create.

⇒ docker network create --driver bridge --subnet=CIDR.

gfgnet.

Tedious and can cause error.

⇒ docker run -dit --name db -e ~~ -e ~~ = ~~.
-e ~~ = ~~ -e ~~ = ~~
-e — --network gfgnet
-v abhivol : /var/lib/mysql

Basic Syntax of compose yml file.

```
name: myproject
services:

  wordpress:
    image: wordpress
    restart: always
    networks:
      - gfgnet
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: gfg
      WORDPRESS_DB_PASSWORD: gfg
      WORDPRESS_DB_NAME: gfgdb
    volumes:
      - wordpress: /var/www/html

  db:
    image: mysql:8.0
    restart: always
    networks:
      - gfgnet
    environment
      MYSQL_DATABASE: gfgdb.
      MYSQL_USER: gfg
      MYSQL_PASSWORD: redhat
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db: /var/lib/mysql

Volumes:
  wordpress:
  db:

networks:
  gfgnet
    driver: bridge
```

One tab Space

Indentation we can check in a systematic way by just pressing tab button. (Sequentially).

Port No of Container. ← Port No of Base System

⇒ This Database is only accessible from EC2 instance. If need to then add port Mapping as well and connect to then

⇒ In docker compose file we are creating 2 containers wordpress & db (MySQL) with volumes and networks also created for both of them.

⇒ docker-compose up -d (for detach mode). (Similar like docker run command)

docker-compose PS (Containers created using docker compose file).

Note:
Now if I simply create a container using Docker run command then it will not show under containers list that we get from docker-compose command.

docker-compose ps

* Images and text files cannot be stored in MySQL DB. whenever we create a web page. Therefore one volume is also mounted to Wordpress container (a shown in yml file)

* If I made changes in the existing compose yml file and add one more container, then existing containers will not be recreated unless I did not make any configuration changes in them.

Note: If forcefully need to recreate/update existing once, can use below command

docker-compose up -d --force-recreate

Docker Compose yml file (Provide all the information) (It will do everything for us)

⇒ Tool for defining & running multiple containers.

docker compose install -o → Store ./usr/local/bin/docker-compose.

↓

docker-compose.yml ⇒ add code in the file.

⇒ docker-compose --help

⇒ docker-compose up -d (> /dev/null) → Redirect console output to this file.
Start My Services.

docker-compose ps → Command shows the status of services defined in Docker Compose file. It displays which containers are running, their names, Ports & state.

⇒ To db Container

mysql -u gdg -p redhat -I (dbviewer).

Show Databases; (To show list of all Databases)

⇒

⇒ USE gdgdb. (Will now use gdgdb as Default Database now).

Dbvisuer ⇒ Open source tool to visualize DB tables| Schema is a better way.

Wordpress ⇒ Web content Management tool, Drag and Drop templates, images etc.

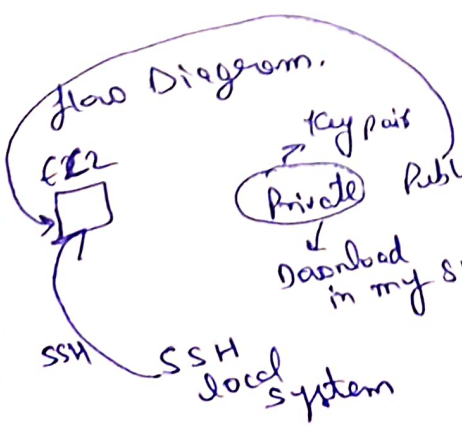
Docker-Compose yml file (Start, stop and rebuild services).

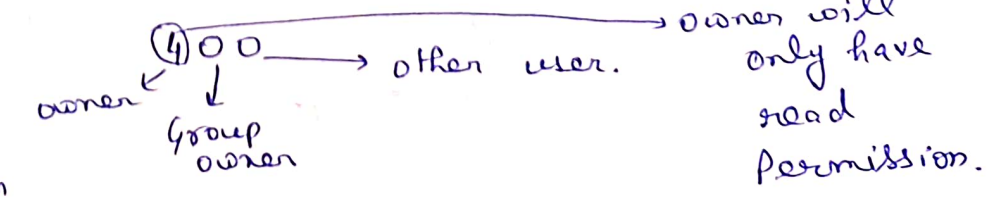
This file defines services, networks and volumes, used in a multi-container Docker application. (In yml file indentation spacing is imp)

Query: SSH connection issue while connecting to EC2 instance.

Answer: Whenever we create new EC2 instance with new key pair. AWS behind the scene will create EC2 instance and AWS will used RSA algorithm to create a private and public key. AWS will store the public key in EC2 itself and will download the private key in the Downloads folder. While doin SSH. Have to authenticate first, password login will not happen. Authentication will happen by the key pairs that we have created.

EC2 with public key need to encrypt the data and will ask decrypt it and then will match with the private key which is in my system. If matches then authentication will be done else not. (Permission should be 400).





* Instead of running Docker command Manually on command line, will create a Docker compose yml will will do the same thing for us, but in a better and efficient way.

⇒ will be launching 2 containers (one for Wordpress and second for MySQL DB) using docker compose yml file.

Installing info again for Docker Compose :-

⇒ Copy curl (client url) command from GFG repo inside Docker folder and this will install docker compose in the EC2 instance. (executable file, not Downloaded at (/usr/local/bin/docker-compose)

⇒ Make Docker Compose executable (Take command from same file as taken from above).

⇒ docker-compose --version (installed info).

⇒ Create a file vi docker-compose.yml file. (Take code or write yourself from Wordpress Image).

(Note) Don't store any hard coded password in yml files (Better to store or take from any other config files, one example can take from Vault).

CI-CD (Pipeline) Day 13 continue

Now once I entered below URL in browser, then will get Pages for Wordpress initial Setup.

URL: 13.126.196.113:8080

Public IP of EC2 instance. Port Mapping done in yml file.

Now will do configurations in Wordpress Dashboard. and can check all that info in MySQL DB.

Now go into the db container using below steps:-

1> docker exec -it db-container_name bash.

2.> mysql -u username -p

3> Enter password (and will be connect to MySQL).

4> SHOW DATABASES;

5> USE Database_name;] This will now refer to the mentioned DB.

6> SHOW TABLES;

Basic MSQL queries to visualize the data.

Select * from wp-posts limit 1.] All colums from table

Select comment_count from wp-posts] Will select only specific column.

→ Select AVG(comment_count) from wp_posts.] Average comment count

→ Select * from wp_users; (Users table).

To view result in vertical format (DB viewer)

Select * from wp_users\G ⇒ vertical.

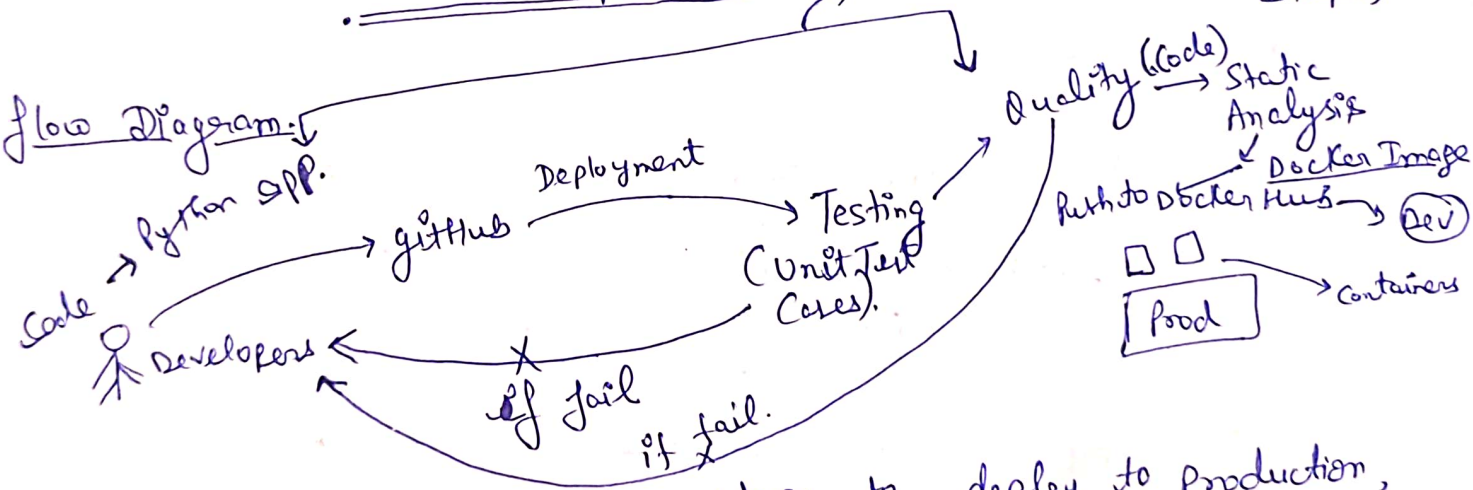
To connect MYSQL on DB viewer from outside world need to do port Mapping in yml file and also add one (Inbound) rule to Securing Groups (3306) for MYSQL connection.

Data Route Wordpress : /var/lib/docker/volumes.

CI-CD Pipelines.

CI-CD Pipeline (All these steps).

Flow Diagram:



Now from GitHub to further steps to deploy to production, then need to follow a process, don't want Developers to manually trigger / ask OPs team to test this code / Quality of the code. So need an automated tool here for this process. Whenever new code pushed to GitHub, it should be tested first.

Jenkins:- The leading open source automation server. Jenkins provides hundreds of plugins to support building, deploying and automating any project.

⇒ first will creating Jenkins Server, then will create CI-CD pipelines. (written in Java. → So should Have Java in System.)

⇒ Within Docker container can also install Jenkins.

⇒ Copy & paste Jenkins create (Container) commands in the t2 medium EC2 Instance.

(By default jenkins stores all its data at ~~jenkins-home: /var/jenkins/jenkins-home~~ /var/jenkins-home.

⇒ Jenkins by default run on. 8080 (Port).

Command

EC2 port (Base system).

docker run -p 80:8080 -p 5000:5000 --restart=on-failure

-v jenkins-home:/var/jenkins-home

Volume mounting.

jenkins/jenkins:lts-jdk17

Image Name.

Now Jenkins setup will be done.

. Plugins are main part/power of Jenkins. Have 100's/1000's of Plugins available for different kind of jobs.

* for Jenkins need to keep checking the git repository. and any changes then it should work accordingly.

⇒ Jenkin Plugins will give some extra intelligence so that Jenkins can communicate with other programs/ servers/ tools.

⇒ Want to see Docker logs :-

⇒ docker logs — container-id.

⇒ Jenkins server can be run on local System/ inside a Container / on Server.

· Job creation Creat a job.

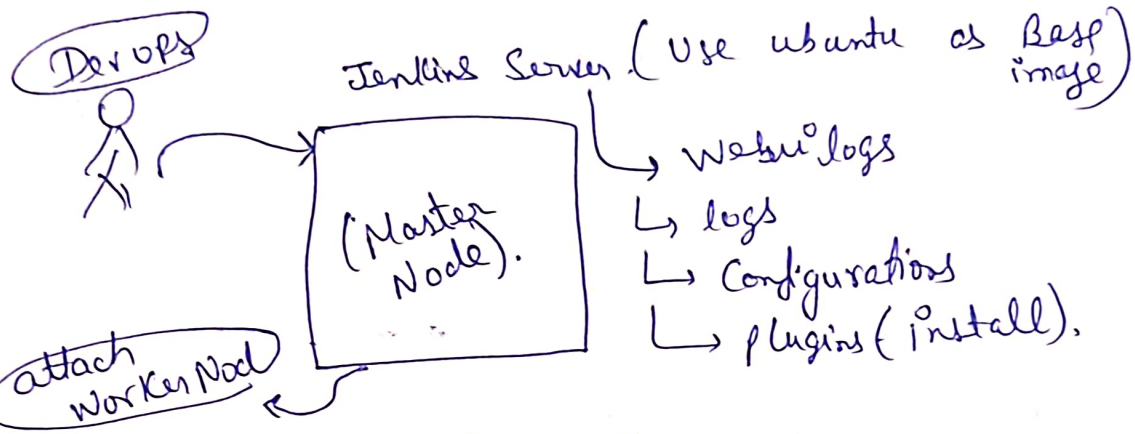
Jobs :- Jobs are simply the whole steps that we include in the flow Diagram.

Each of those all steps is like one job.

Will be creating Jenkins Pipeline And Deploying on Jenkins

Add Git Hub repo code url in Source Code Management.

Build Now, will start the build and once build is success, we can see the data under (Workspace) Section.



Note → Always recommended when using Jenkins server, don't perform such activities like. installing a Package etc inside Jenkins server.

Imp

→ (Worker Node) If need to perform any job / script which need to be execute, create a seperate system for that, known as (Worker Node)

Worker / Slave Node.

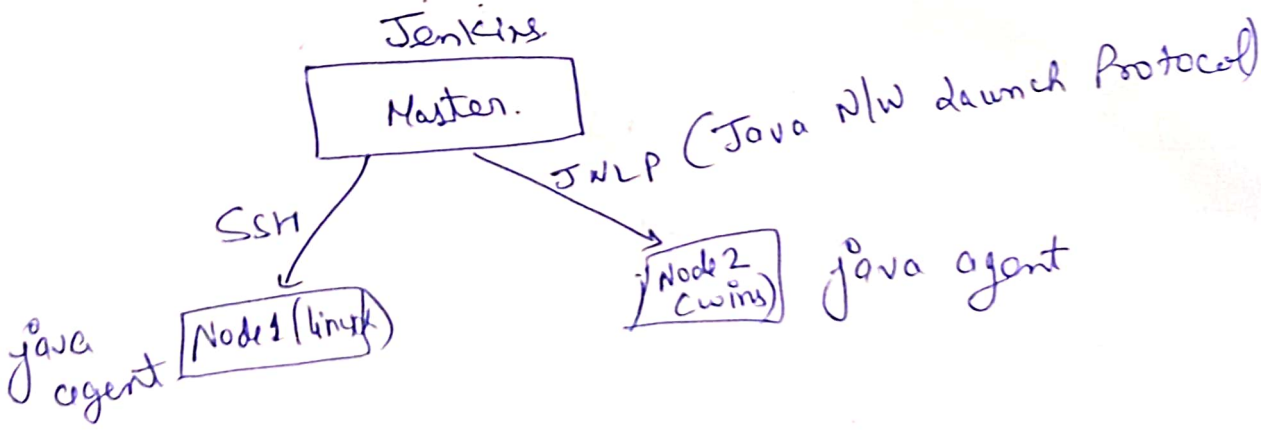
(Multi Node Cluster).

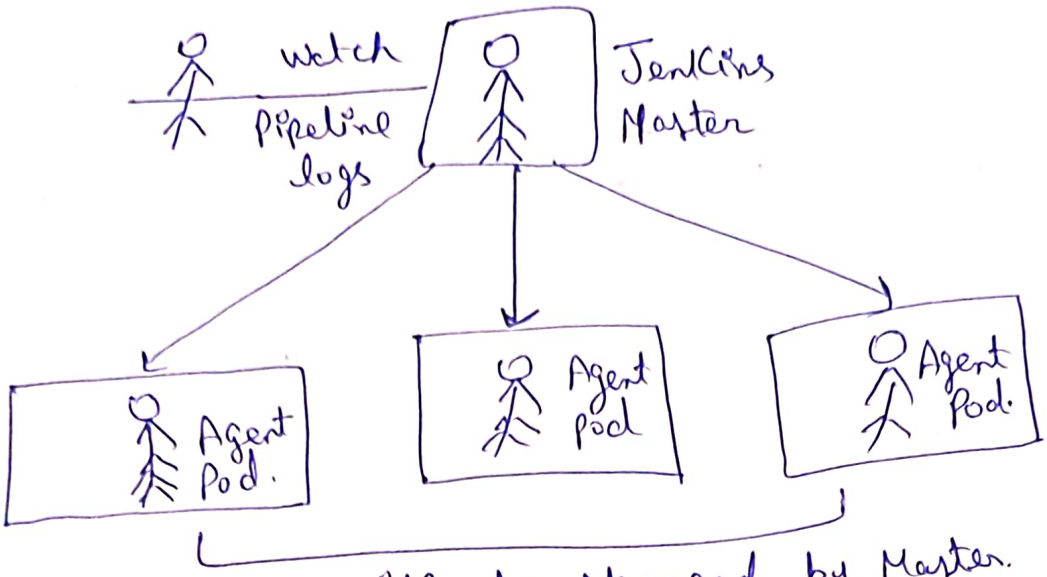
→ Jenkins allows you to distribute work-load across multiple machines, improving efficiency and scalability.

The core idea is to have one Jenkins Master node that Manages the system, and multiple slave or agent node that executes the jobs.

Jenkins Master :- The central control point, responsible for managing the Jenkins system, storing configuration and handling user interactions.

Jenkins Agent | Worker | Slave :- Machines that execute the jobs assigned by the master.





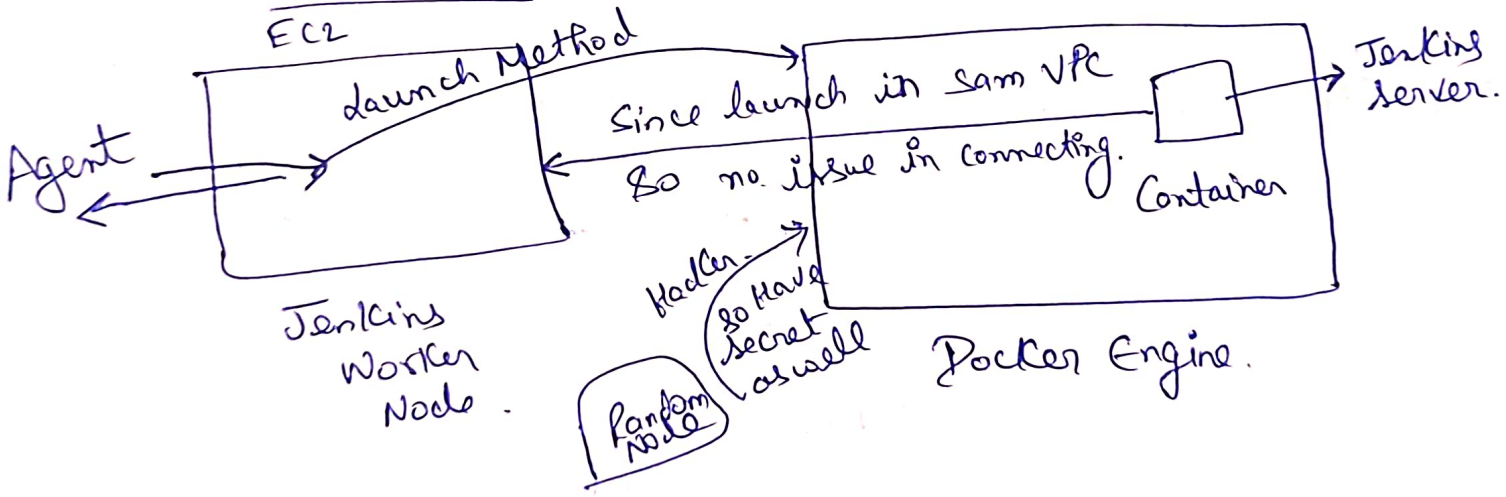
will be Managed by Master.

and for this Have to run **Jenkings Agent** **

↓ will give o/p to Master.

⇒ **No need to Install this Agent**

for this will be creating new EC2 instance which will be Worker Node.



* Go to Jenkins (UI) as have to configure there to work this new EC2 as worker Node.

Steps Jenkins UI → Manage Jenkins → Nodes

(By default Have 1 node (Built in) only where Jenkins and Worker are running in the same system.

Click on Add Node

⇒ Remote root Directory :- In worker node (new EC2 instance) where will kept all files related to Worker Nodes.

folder info ⇒ mkdir data. (/data).

Usage :- Depends on the usage will select Nodes in case of Multiple Nodes. (Use as much as possible).

Launch Method launch agent by connecting it to the Controller.

⇒ while running all the commands suggested (to make the New node Active need to install java as well in the Worker Node instance).

⇒ wget ⇒ command to Download a file.

↳ rpm file for java-jdk was not present in yum repository so have to download.

↳ Once this file is download, do ls in root directory and run below command to install java-jdk.

Yum Install jdk-17.0.12_linux-x64_bin.rpm

to run in background use (&) Symbol at the end of Script/command.

⇒ If need to Kill a program that is running in BackGround follow below steps.

1.) first

fg ↵

foreground ↳ Takes java command into foreground.

2.) Ctrl + C