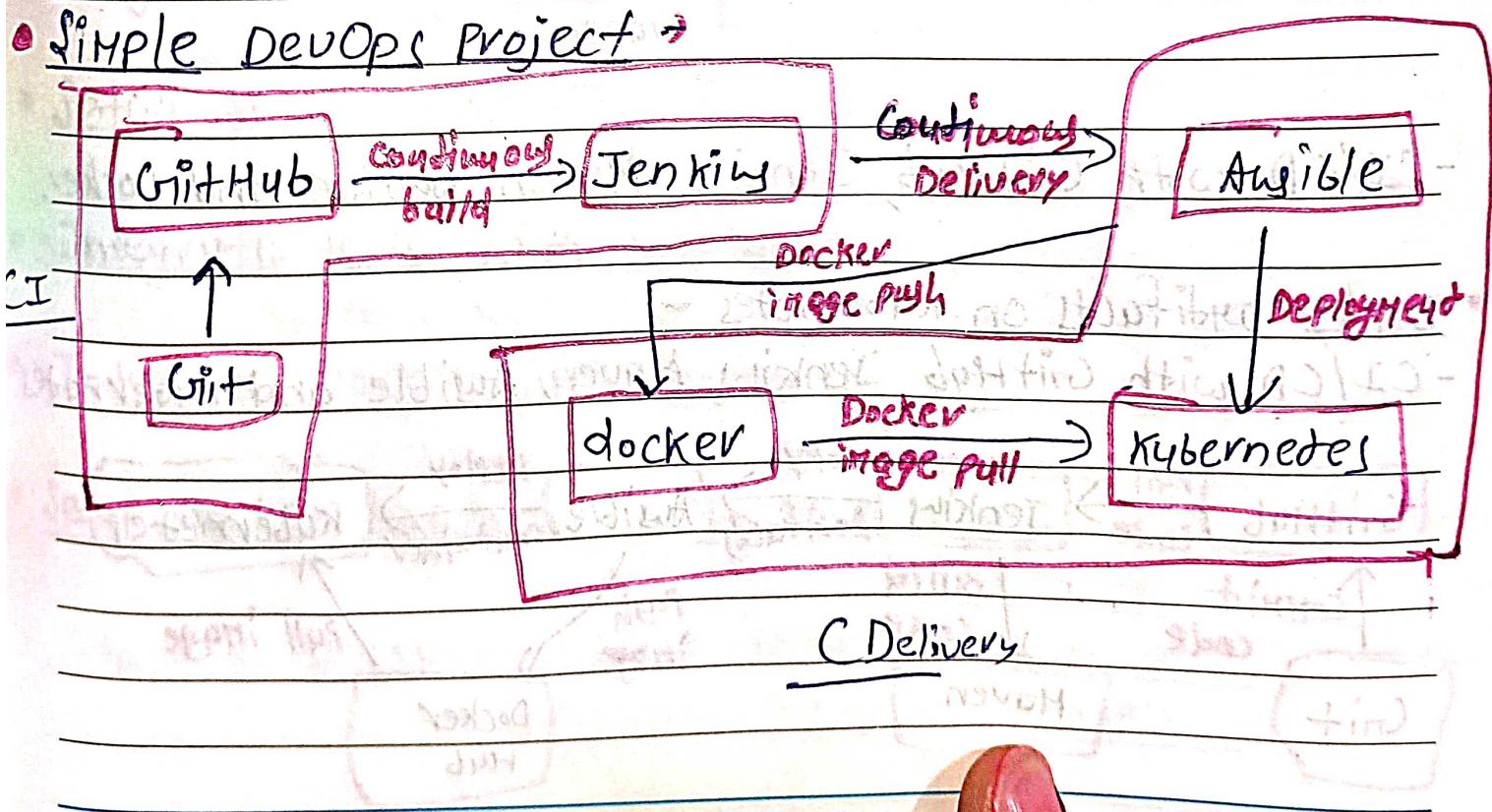


Java DevOps / CICD

* CI and CD →

- Continuous Integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day.
- Continuous Delivery (CD) is a software approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, without doing so manually. It mainly involves building, testing and releasing software with greater speed and frequency.
- Continuous Deployment (CD) is a software release process that uses automated testing to validate if changes to a codebase are correct and stable for immediate autonomous deployment to a production environment.

* Simple DevOps Project →

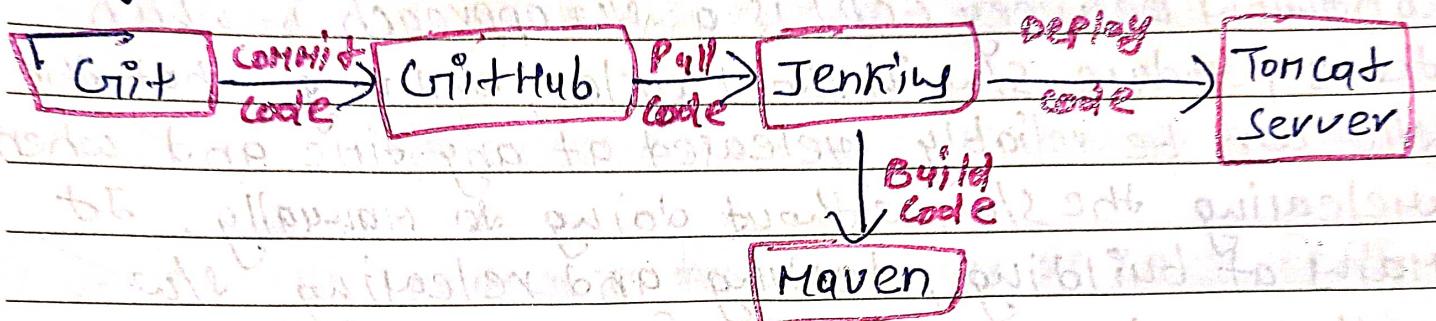


C Delivery

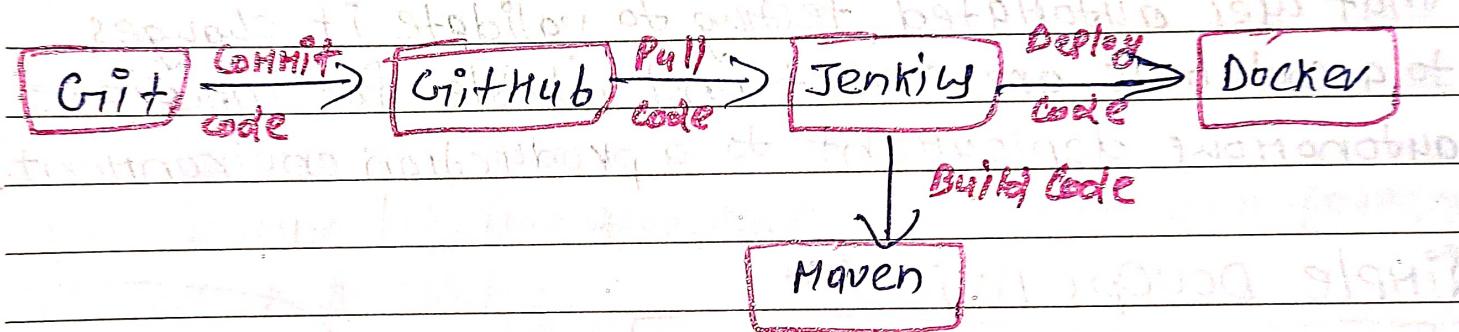
• Build and deploy on Tomcat server -

- Setup CI/CD with GitHub, Jenkins, Maven and Tomcat

• Deploy artifacts on Tomcat server -



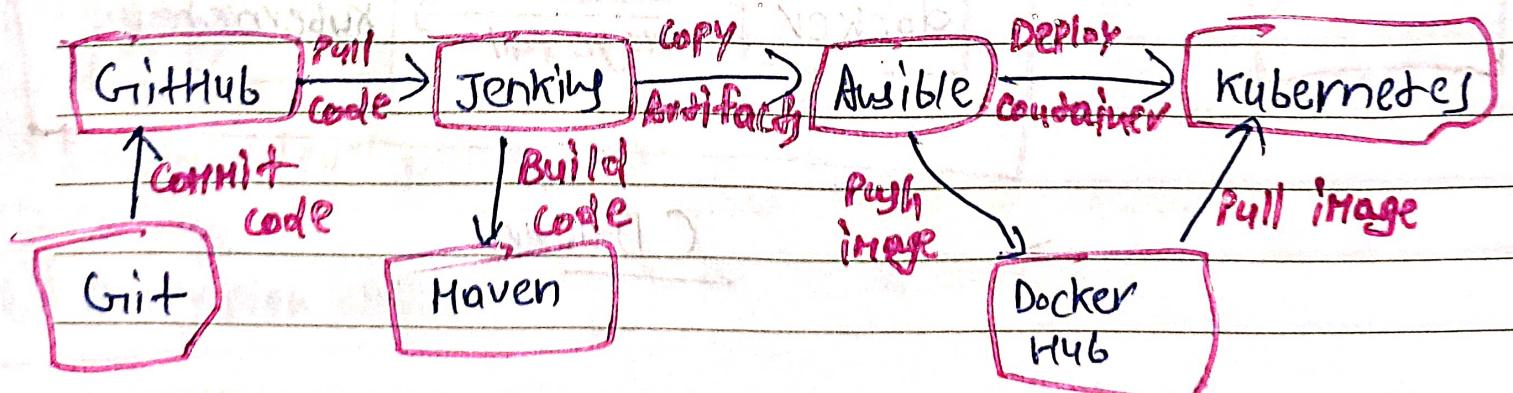
• Deploy artifacts on a container -



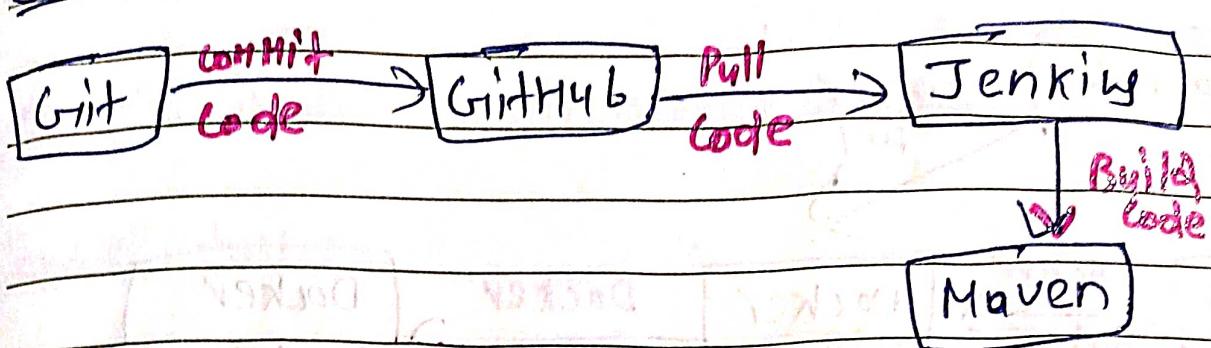
- CI/CD with GitHub, Jenkins, Maven, Ansible and Docker.

• Deploy artifacts on Kubernetes -

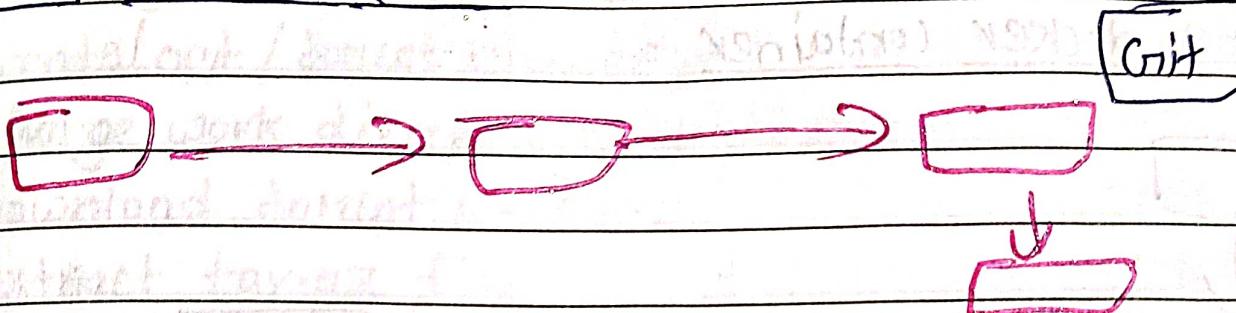
- CI/CD with GitHub, Jenkins, Maven, Ansible and Kubernetes



Build Code -



Setup Jenkins server -



Integrate Maven with Jenkins -

Integrate GitHub with Jenkins -

Setup Tomcat Server -

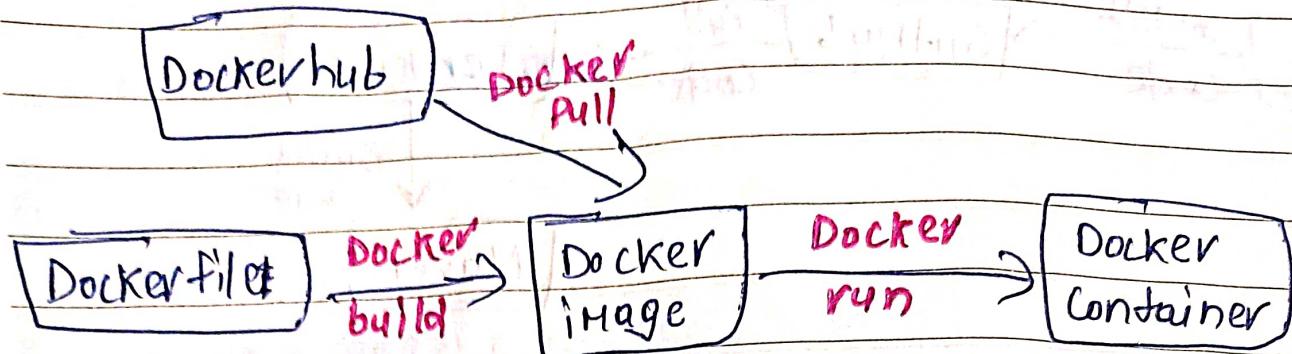
Integrate Tomcat with Jenkins -

Integrate Docker with Jenkins -

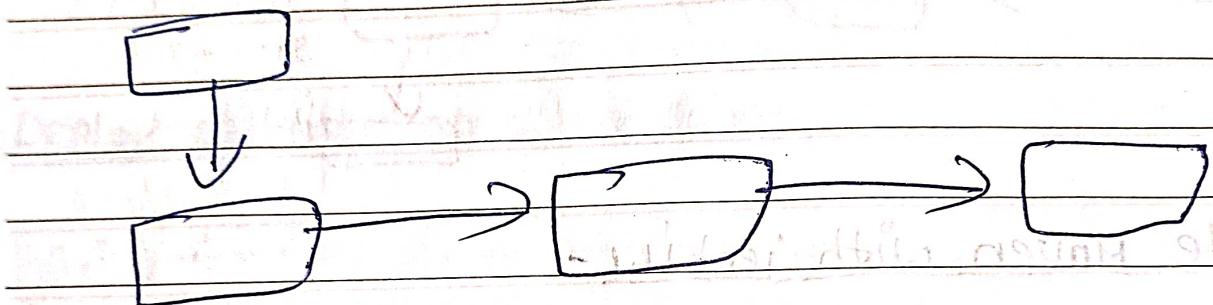
Setup Docker host -

Integrate Docker Host with Jenkins -

- To create Dockerfile -



- To create docker container -



- Write your 1st Docker file -

- **FROM** → To pull the base image
- **RUN** → To execute commands
- **CMD** → To provide defaults for an executing container.
- **ENTRYPOINT** → To configure a container that will run by an executable
- **WORKDIR** → To set the working directory.
- **COPY** → To copy a directory from your local machine to the docker container.
- **ADD** → To copy files and folders from your local machine to docker container's.



DATE 21/03/22

- EXPOSE → Informs docker that the container listens on the specified network ports at runtime.
- ENV → To set environment variables.
- Install tomcat on Centos -

Pull centos from dockerhub

Install java

Create /opt/tomcat directory

Change work directory to /opt/tomcat

Download tomcat packages

Extract tar.gz file

Rename to tomcat directory

Tell to docker that it's listening on port 8080

Start tomcat services

Script - FROM

- RUN

- RUN

- WORKDIR

- ADD / RUN

- RUN

- RUN

- EXPOSE

- CMD

- Manage DockerHost with Ansible -

• On Docker Host

- Create ansible

- Add ansible to sudoers file

- Enable password based login

• On Ansible Node

- Add to hosts file

- Copy ssh keys

- Test the connection.

Ansible

Docker

- Integrate Ansible with Jenkins -

Jenkins

Ansible

• Deploy Ansible playbook -

- Remove existing container
- Remove existing image
- Create new container

• Prepare Ansible Server -

- Setup EC2 instance
- Setup Hostname
- Create awxadmin user
- Add user to sudoers file
- Generate SSH keys
- Enable password based login
- Install ansible

• Setup Kubernetes -

• Deployment Tools

- Bootstrapping clusters with kubeadm
- Juicing Kubernetes with kops
- Juicing Kubernetes with kubespray

• Managed Services

- Amazon EKS
- Microsoft AKS

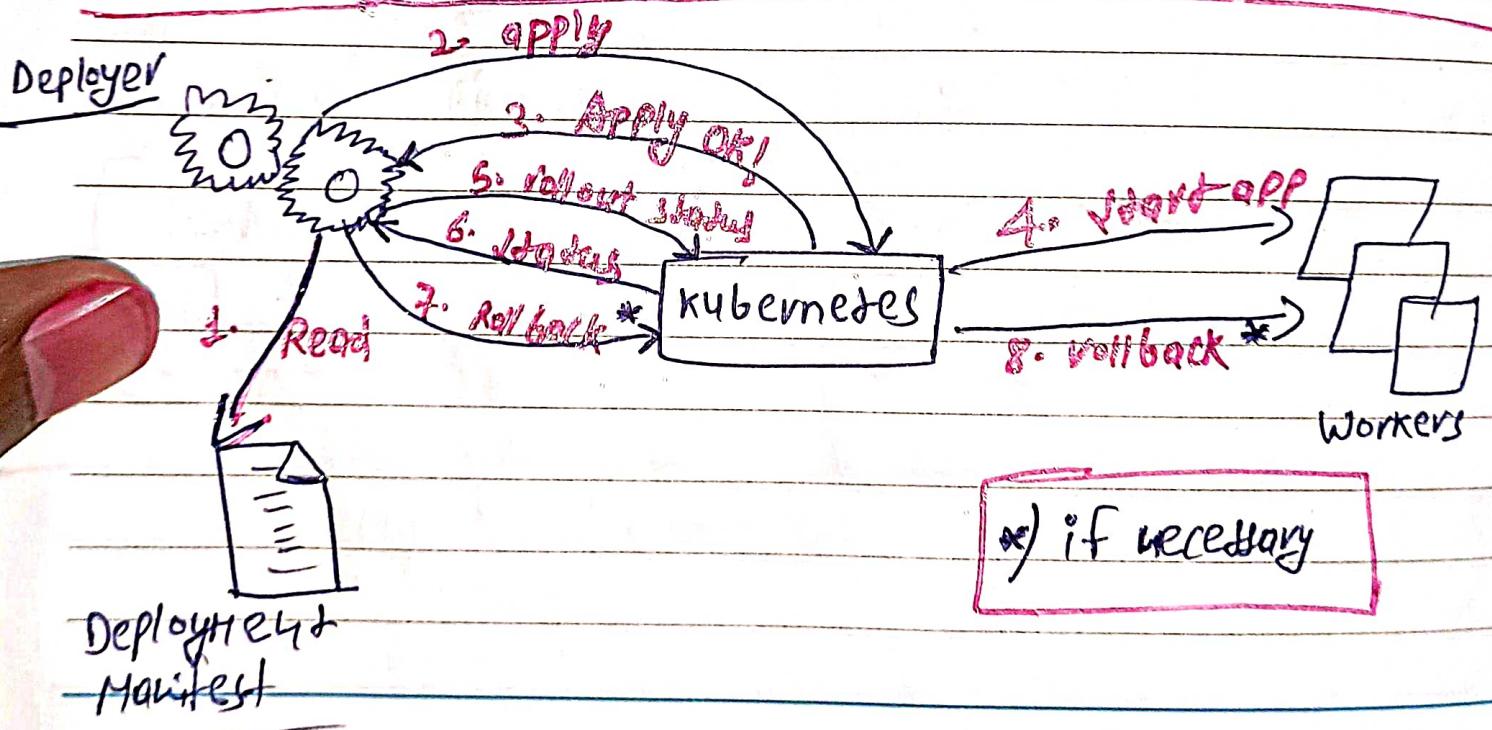
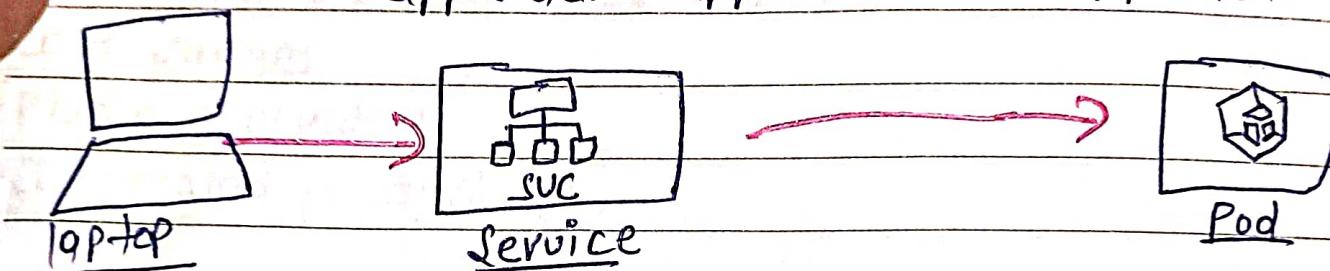
• EKS Setup -

- Launch EC2 instance - Bootstrap
- Latest version of AWSCLI
- Setup kubectl
- Setup eksctl
- Create IAM role
- Create a cluster
- Validate cluster
- Delete cluster

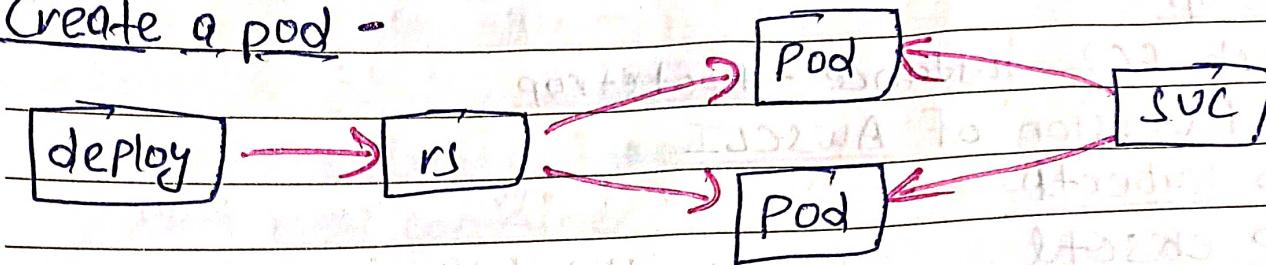
• Setup Pod and Service -

Selector:
app: demo-app

Labels
app: demo-app



Create a pod -



Deploying on a pod -

