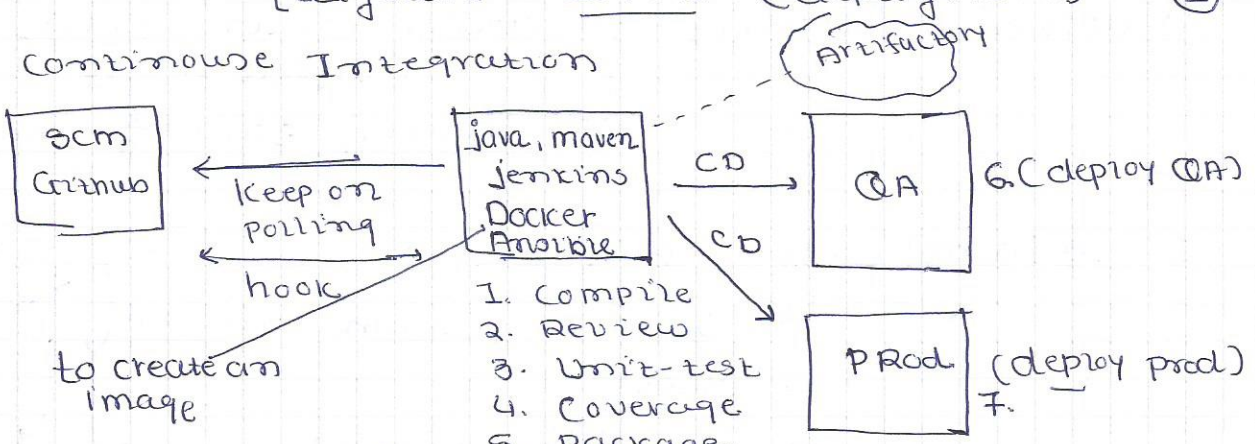
{day-20} CI/CD (deployment) (1)

→ CI → continouse Integration



→ [.war/.jar] → Artifactory (Shared drive or gogle drive)

Artifect → [From artifactory]

└→ package to docker Image. [.war/.jar image]

Push → docker hub

CI (continouse Integration) (DTR - docker trusted Reg.) - own data center

→ New docker image should be created

→ New Version should be coming every new commit

CD → docker hub → can trigger → deployment

6. Deploy QA

7. deploy prod

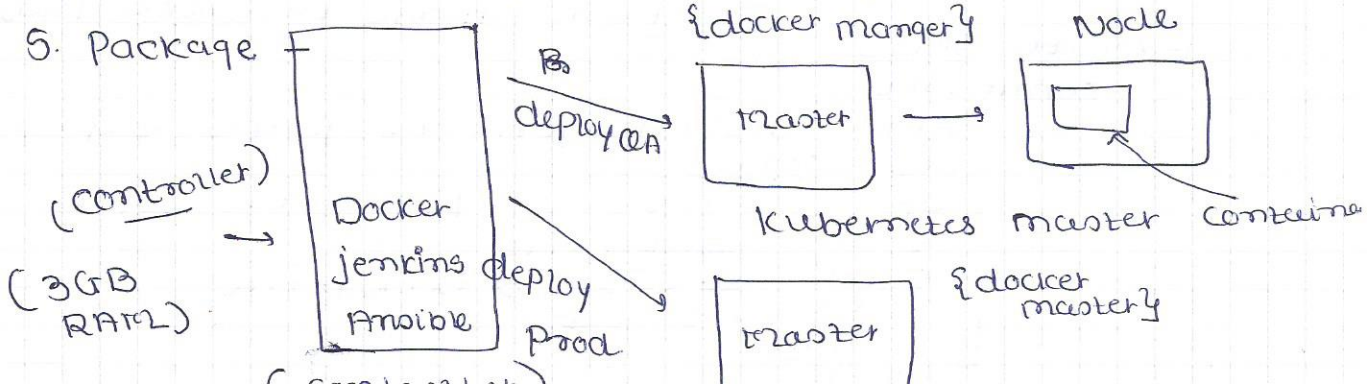
* Configuration management

jenkins → trigger → Production machine

Ansible

→ create Inf./ in Ansible jenkins

5. Package

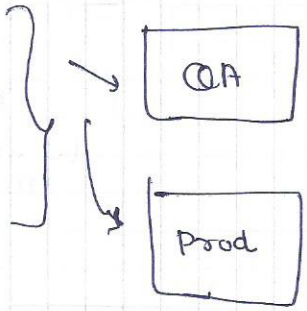


↳ docker → Security → docker content trust (signed image
↳ Security scanning from docker

→

I. @ /home/ awsl playbook.
↳ service jenkins status
↳ docker
↳ Ansible -version

jenkins machine



* Ansible ⟶ install docker

/opt/deploy/ → ansible qa -m ping
inv > ansible prod -m ping.
[qa]
3.12.8.7
[prod]

* ansible qa -m script "/tmp/lab/docker/install/install Docker.sh"
ansible qa -a /tmp/labs/docker/install
↑ it's already in Remote machine
" prod -a /tmp/labs/docker/install/ install Docker.sh -b

→ {Docker Hub DIRY} →

1 Compile job → poll SCM * * * * * (1 minute)
2. Review → Sonar Qube (
3. 5. package → Build war file → create docker image
→ Push image to docker hub.

Execute Shell
command :
sudo chown jenkins:jenkis deploy
cd /opt/
cp /var/lib/... /petclinic.war .
touch Dockerfile
→ [sudo cat << EOT >> Dockerfile
FROM tomcat
ADD ...
EOT

sudo docker build -t leaddevops/petclinic:$ ③
$Build_Number.

↑ Building an image tag. (jenkins #50 (Build No.) job No. then it will be version no.

" Login -u

>— sudo docker login -u leaddevops -p) (Each image own version)

docker push LeadDev/petclinic:$Build_Number ↑ pod]

docker run -d --name = deployapp -P

↑ image.

② ~~sudo do~~ sudo docker login DTR URL -u leaddevops -p

* Post deploy prod. yml

key version
value $Pno.

* {Build pipeline plugin}

> docker run -d -P 3000:3000

* Prometheus → collects the data from docker, kubernetes.

execute: container_cpu_user_second

↳ Grafana (better visualization

Add datasource

↳ where Prometeus Running data source working

→ create a dash board

→ each & every Query (container_cpu_

* Kubernetes dashboard for grafana (4018)

↳ manage → import 4018 (prometheus data source)

Pod

→ all deployment → No. nodes,
→ how many containers

* kubectl run ravipodex --image httpd --gen

↳ pod

→ Add Query : copy premeutes query

↳ Add views (graph)

← 4028 → manage → save dashboard

— manage → 4028 → kubernetes (premeteus data sources)

↙

Preme→ kubernetes cluster data source (add matrics)

* JFrog with Jenkins

↳ logs → {SPLUNK is best.}

↳