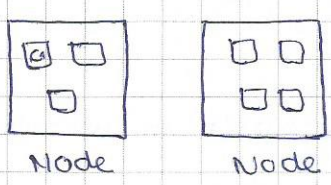
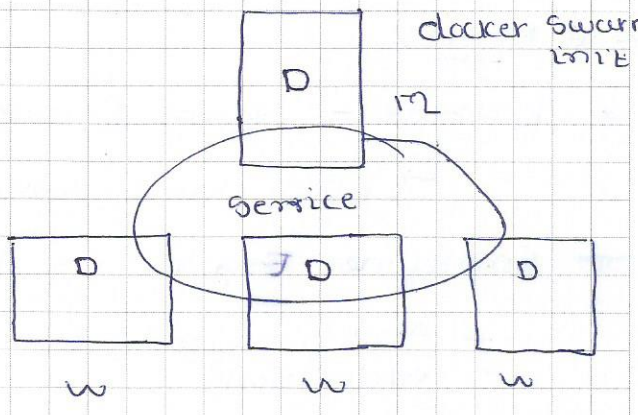
[day-12] [kubernetes]

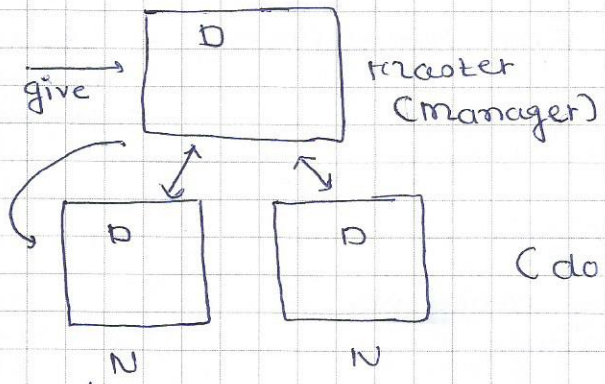
→ What is kubernetes ? - manage containers (help to do that)

↳ one machine [one container]





→

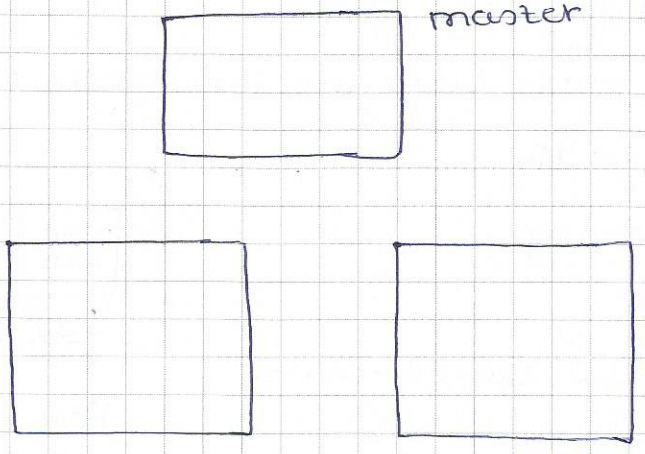


→ master doesn't take work.

→ Node (cluster) takes the load.

↳ kubernetes cluster

→



→ API-Server (heart comp)
- any request (kubct apply whatever command) -f .yml

→ Cluster Store
- storage space

→ k Scheduler
- Schedule container on nodes.

→ Controller-mager

* API-Server
→ compose-yml (in docker). manifest file (.yml or JSON)

* Cluster-Store
- kubernetes saves the date - how many machine
- etcd (software) key-value pair database.

Rakesh	x	x	x	x
Ravi	✓	✓	✓	x

(Normal DB-) (Ravi has a value)

(key-pair)

Ravi — vinay — Ashal

{text} — { } — { }

→ you don't update data for all.

→ how many Nodes
- how many container

* Scheduler

* Controll manager - env. config. (desired states)
  → Bring one.

API-Server → {comm. with all}

Nodes.

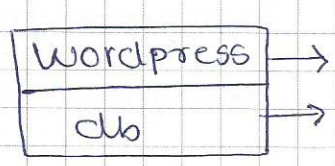
* Kublet - Agent (one master → 10 Nodes) (every Node)
  ↳ every Node contain (Api Server for request)
* Container Engine (- to Run containers
  - Pull images
  - Start/Stop
  - LXC
  - LXD
  - Rocket
  -
* Kube-Proxy
  - Networking
  - Route Req. to many container
  - IP Address to container.

*

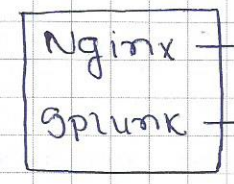
Pods ≈ Container
↳ extra Layer on Containers

IP → IP
↑ Pod
pod IP doesint change.

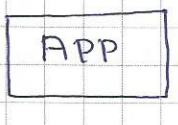
Wordpress



Single deployment.

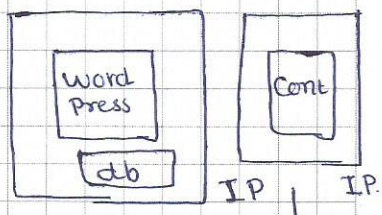


Log collecting



→ one Image
→ Couple of Image.

→ two Service in Seperate container



→ (one Pod → 1 Container)

→ Scaling is possible for pods.

→ Containerized Pod.

→

* AWS Instant. (4GB RAM) (T2. medium)
  - ~~gp~~
* Google Cloud Platform. (credits)
  - Hard way (API-Server, etcd)

VM2 Instance (machine)
↳ master
↳ machine ty (2 vp. 7.5 GB RAM)
↳ Ubuntu 18.04 LTS (10 GB)
↳ Allow default access
→ Create

→ node-1 – (3.75 GB) → Ubuntu (10 GB).
→ node-2 – Standard (3.75 GB) → " "

→ ssh promt direct on browser.
sudo su
root@master:~#

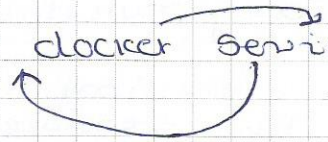
> Create users. (AWS → setup-user.sh)
vi s.sh
chmod 755 s.sh
./sh
* create user and connect from ⊙ Console.
putty, gitbash,
[Chmod 755 s.sh]

* ssh devops@34.70.10.2
password: _

devops@master:~#

* Install kubernetes [kubeadm – every thing is a container]
– docker installation

> docker service status.



→ kube ctl
– kubeadm.

CNM2 (default Network) { → Kubernetes
– part of docker. – config. Net. explicitly.

→ CNI [Network Plugin]
– weave
– Calico
– Flannel (

→ kubernetes Calico.
* master
sudo kubeadm init --pod-network-cdir=
10.244.0.0/16
IP address writing that

docker,
> kubectl _

Kubectl get pods --all-namespaces.

root@master: → . (means hidden)

sudo mkdir -p $HOME/.kube.

sudo chown $(id-u): $(id-g) $HOME/.kube/config.

> kubectly get pods --all-namespaces.

- etcd-master
- kube-apiserver.
-

> @ kubectly apply -f (file)

- core dns- 66b .. 1/1 } Networking plugin
  core dns- 66bf1 1/1 }

CNI (container Networking interface)

> kubectl get nodes.

Name Status Roles Age

* → same Networking apply to Nodes also.

Namespace → kubernetes separation

Pod -

RS -

Deployment -

Service -

Dashboard -

> kube get component status.

> dump - for seing more data

> Services } → everything
> deployment } kub. resource/ Object.

> Service [kubelet] status

API server

[master] [Node]

→ Generate token → Create gen kubeadm join.sh

* Imparative with docker.
* kubernetes declarative (manifest.yml)

* .yml (File)
  apiVersion: v1 → api server comm.
  kind: Pod (what kind of Resource)
  metadata: (data about object -

```
metadata:
    name: firstpod        } Pod info
    labels:
        app: myApp  ——————  labeling as.

Spec:          - Specification
  containers:
  - name: tomcat   (Name of container)
    image: nginx
    Ports:
      - containerPort: 8080.
```

> kubectl create -f pod.yml

> kubectl get Pods -o wide
↳ output.