# Insrtuctions in :

<https://github.com/Gamal-Mohammad/instructions/blob/main/MiniKube%20Ubuntu_Debian_docker_driver>

**Install Kubernetes on Ubuntu/Debian**

#### Install Docker First if Installed Skip these steps

sudo apt update

sudo apt install curl

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker.gpg

sudo add-apt-repository "deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt update

sudo apt -y install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-registry

sudo usermod -aG docker $USER

newgrp docker

**######### Install MiniKube (kubernetes platform)**

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube\_latest\_amd64.deb

sudo dpkg -i minikube\_latest\_amd64.deb

minikube start --driver=docker --nodes=2

sudo snap install kubectl --classic

minikube kubectl -- get pods

kubectl get pods

minikube kubectl -- cluster-info

kubectl cluster-info

minikube addons list

minikube dashboard &

kubectl get nodes

kubectl get pods -A

Kubernates commands :

List pods running and not running in all nodes

**Kubectl get pods** === docker ps -a (in current node only)

List replicasets

**Kubectl get replicasets**

List deployments

**Kubectl get deployment**

List nodes

**kubectl get nodes**

create pod in deployment

**kubectl create deployment senior –image httpd –replicas 3**

**check the status**

**kubectl get pods -o wide**

get pod information

**kubectl describe pod seniordeployment-5dd9fff8bb-6hsvr**

delete running delete (replica ll create another one )

**kubectl delete pod seniordeployment-5dd9fff8bb-6hsvr**

increase no of pods in deployment

**kubectl scale deployment senior replicas 6**

start minikude in each vm start

**minikube start**

start minikube dashboard

**minikube dashboard &**

A screenshot of a computer

Description automatically generated

enable autocomplete in terminal

**kubectl completion -h**

A screenshot of a computer program

Description automatically generated

Create standalone pod (not recommended)

**Kubectl run lab1 –image httpd**

Delete pod

**Kubectl delete pod lab1**list all k8s resources

**Kubectl api-resources**

Get num of resources -> **Kubectl api-resources | wc -l** (word cound – lines) one line table coloms names and num of resources

Note : namespace = true -> created inxide the namespace , false -> created outside the namespace على مستوى الكلاستر نفسها

Note: KIND -configuration file اسم الاوبجكت بالطريقه الى هيتنادي بيها ف ال

Note : appversion appعنى اسم ال

Read information about any resource

**Kubectl explain (resource name)**

Namespaces

Kubernetes object divide the big cluster to mini clusters and I can control each cluster resources

Create namespace:

**Kubectl create namespace dev**

List all namespaces

**Kubectl get namespaces**

Know what inside namespace

**Kubectyl Get pod –namespace dev**

Know which namespace am I in

**Kubectl config get-context**

How to switch to another namespace

**Kubectl config set-context (claster name) --namespace dev**

Delete namespace

**Kubectl delete ns dev**

List namespaces

**Kubectl get ns**

**Scaling :**

Manual automatic

Cmd yaml cmd yaml

Hpa scale pods automatic usind cmd : but we need to determind the limit

Depends on HW resources

Orchestration tools:

Swarm 🡪 docker company docker built in tool, work only with docker container engine tools.

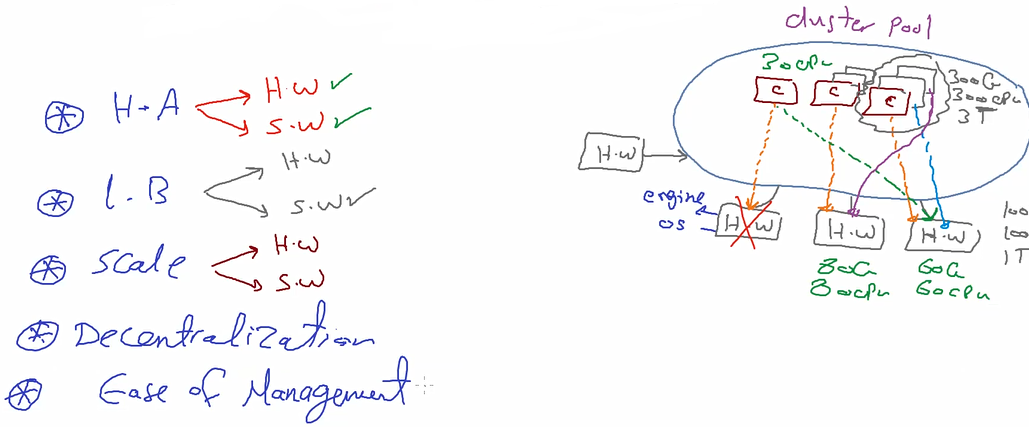
Kubernetes k8s 🡪 google company, work with all container engine tools.

Openshift 🡪 work with all container engine tools.

Mesos🡪 work with all container engine tools.

K8s is orchestration tool that we use to manage docker engines to provide.

all devices compined together to create cluster the orchestration tools use this cluster as one device



1. High availabitiy SW and HW : SW or hardware failure ll not be a big problem orchestration tool can manage and keep the containers (application) available , if one device fail , all containers ll be moved to another device in the cluster

بيعمل كلاستر من كل الاجهزه الى عنده و أي جهاز منه وقع يعرف ينقل الكونتنر على أي جهاز تانى فيهم و لو السوفتوير وقع على واحد فيهم نفس الكلام

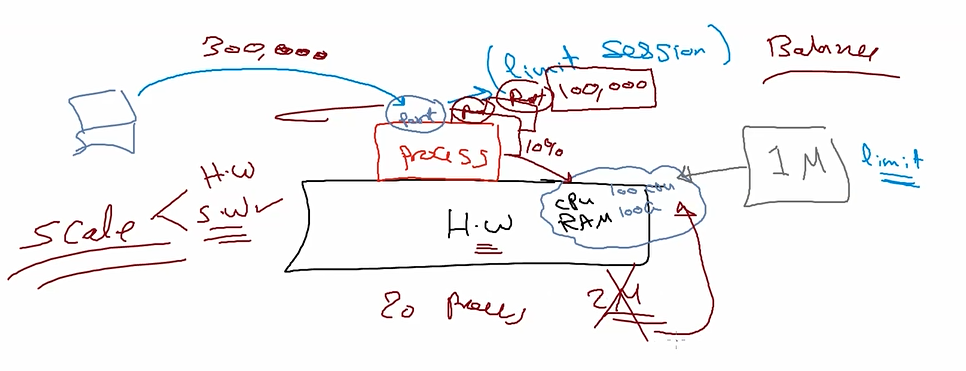
1. Load balance : when we get a request the orchestration tool control all cluster devices and distribute the tasks between all devices

الاجهزه الى ف الكلاستر ده بنقسم الشغل عليهم بالتساوي

1. Scalability SW/HW :increase /decrease pods (SW) , and devices to scale the cluster size

ينفع نزود الاجهزه الى جوه الكلاستر و ينفع نزود الكونتينرز الى شغاله بتستقبل الطلبات بس

لازم حجم الهاردوير يكون متوافق مع احتياج السوفتوير و بناء عليه اقرر ازود و لا لا



لان ده اخر الهاردوير بتاعيContainer يقدر يستقبل 100 الف ده معناه ان أخرى اعمل 10 Container الهاردوير يقدر يستقبل مليون و ال

1. Decentralization مش معتمد لا على هاردوير و لا كونتينر

سهل اتنقل على السيرفر المتاح و سهل انقل لكونتينر شغال

K8s structure

A diagram of a computer system

Description automatically generated with medium confidence

K8s tool creates cluster pool this cluster contains hw nodes , each node installed on it container engine

And k8s orchestrate this container engines

Cluster pool roles

1. Controller (manager)(master)(control-plane) one or more device (recommended more than one)

This manager contains k8s engine (platform) creates the cluster and manage all engines

1. Worker nodes : nodes (servers ) contains ours containers الاااااف الاجهزه

Note:

All this nodes are physical devices and communicate together to create one logical cluster pool

Master :create the cluster , create pods ,assign pods to HW nodes, and if any HW fail master reassign the task to another worker

A drawing of a diagram

Description automatically generated with medium confidence

Request journey :

* Admin or developer send request to master node 🡪 Req go to API-Server

API-Server is the gateway of the master responsible for :

-accept all requests

-communicate with other objects to do the user request

-assign the task to one of the workers

* API-server call etcd : db contains all data of pods (meta data) key, value (unstructured data)

So API server save all pod data in etcd except (hw node , status) because this data not yet ready it save (img , pod name , ports , network ,volume ,env variables,…)

* api-server go to scheduler : scheduler is an object that check