Kubernetes: Its a open source container orchestration platform.

By default Kubernetes is installed as a cluster. I.e group of nodes.
Kube depends on yaml file.

Worker node:
1. Kube proxy: generating cluster IP address, gives IP address, expose to outside, load balancer .
2. Kubelet. Creation of pods. This will deal with Api . This ensures pod is always running else inform to control manager.
3. Container runtime. Running container.

MasterNode:
1. Api server: gives input and output.
2. kube Scheduler: just scheduling pods and resources based on api server information.
3. ETCD: backup or restore and info of all cluster. Its a key value store.
4. Controller Manager: To ensure all the pods are running based on mentioned replica sets.
5. Cloud controller manager:

Deployment.yaml: which will deploy and run our services same as container.

Service.yml. name space and label should be matched so this will expose to outside world.

ingress.yml : this will give many load balancers while sending traffic but instead (install and create a single controller which sends traffic to necessary domains and expose visa service and it will have 1 load balancer).

IngressController.yml: This will have only one load balancer but can send traffic to all ingress files mentioned and sends traffic to desired domain sites.

Horizontal Pod Autoscaling. (Scales when pods exceed Cpu usage)

Horizontal Pod Autohealing (this will be created if the pod is down or getting deleted before completely down)

Kubernetes Distributions you are using in production:

We are using many distributions like:

1. Kubernetes
2. Openshift
3. Rancher
4. Tanzo
5. EKS.

Difference between Kubernetes and EKS:

Kubernetes will have to manage clusters on its own.
EKS: Amazon will provide support for eks incase of any problem.

1. Why Kubernetes is deployed as pod and not as docker container??

A: Instead of docker cmd line like docker run -it images - P -v .... we are just putting all the specification in pod.yml file.

Kubectl get pods -o wide
Kubectl create -f pod.yml
Kubectl describe pod nginx
Kubectl get pods -v=7 this will give info what it is doing
Kubectl logs nginx
Kubectl delete pod pod id

Kubeshark:8899 port

Create Docker file take it from abhishek
Create Deployment.yml : we can change labels and selectors and templates name. Keep it same. Give docker image name which u have created.
Create service.yml : Get the app name from deployment file only inside template. app: sample-python-app and change node port same given in deployment file .
We can give type: Loadbalancer here itself.

Kubectl apply -f Deployment.yml and service.yml.

Kubectl get svc : To We can see of its a load bar or cluster based on our requirements.

Docker image public : cmilanf/docker-snake. 3000 port

Kubectl get ing : ngress.yml

Install Kubeshark for visualization of servers load balancing

/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"' >>
~/.bashrc
source ~/.bashrc

Ingress with Ingress Controller:

certbot certonly --manual --preferred-challenges=dns --key-type rsa --email \
tvsravya95@gmail.com \
--server https://acme-v02.api.letsencrypt.org/directory --agree-tos \
-d *.kuttysravya.shop -d kuttysravya.shop

cd /etc/letsencrypt/live/kuttysravya.shop/
cat fullchain.pem > tls.crt
cat privkey.pem > tls.key
Kubectl apply -f ingress-tls.yml

To change password for any username in ubuntu:
sudo passwd username

kubectl create secret tls awscloudops --cert=tls.crt --key=tls.key

sudo chmod 666 /var/run/docker.sock. To give docker permissions

https://www.youtube.com/watch?v=hBGVwa8MY4A&list=PLIMuaFvdF6-g4EFO0MA2TWaQn9NuVJdTl&index=1

RBAC: User management and also managing access for services.

To manage:
1. Service accounts /users.
2. Roles, cluster roles.
3. Role binding, Cluster role binding.
4.

Kubernetes doesn't deal directly with user management like user add etc. It offloads with identity providers. Eg : AWS EKS IAM Roles to access Kube.

For service account/user —> create user and create role. Using role binding we attach role to user.

Note: Creating role within namespace is called role. Within cluster is cluster role.