Kubernetes

open-source container orchestration platform
which is designed to automate the deployment, scaling and manage the containers.
Problem: manage multiple containers between multiple environment and that's problem.
Features:

1. Self-Healing: Auto Restart, replace and reschedule the failed containers.
2. Load Balancing: Distribute traffic and provide high availability.
3. Secret and configuration management: securely managing the data like password, API keys.

Kubernetes Components

1. Master Node: managing the cluster and handles all scheduling, state management and doing updates.
2. Worker Nodes: Actual application workload running here as pods
3. Pod: very smallest deployable unit in k8s, where you can add more than 1 containers.
4. API Server: playing a role of communication hub between users and cluster.
5. Controller Manager: maintain clusters state by managing replication, load balancing like other tasks.
6. Scheduler: assigns pods to nodes based on requirement resources.
7. ETCD: Storage for storing cluster data, in the form of key value pair.
8. Kubectl: agent which is running on each worker node to maintain container state to its desired state.
9. Kube-proxy: manages the networking for all K8s Service.

Kubectl (Kubernetes Command Line tool)

Using this we can interact with kubenetes cluster.
We can manage cluster resources, deploy application and monitor cluster.
It will work with local cluster using minicube as well remote cluster

Minicube:

Its a Tool which is used to set up and run a single node kubernetes cluster locally for development and testing.
It provides and environment to practice kubernates cluster without using any full production cluster.

Download Minikube

Once you download install the same.

Open cmd and check version: minikube version

if you are getting and error like minikube is not found as internal or external command them.

Right click of This PC --> Properties --> Advanced System settings --> Environment Variable --> path --> new and

add minikube path (you can copy path from C:\Program Files\Kubernetes\Minikube set again check

```
C:\Users\NEW>minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty

C:\Users\NEW>minikube start
* minikube v1.35.0 on Microsoft Windows 11 Pro 10.0.26100.3037 Build 26100.3037
* Automatically selected the docker driver
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 3.19 Mi
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 2.05 Mi
```

```
* Creating docker container (CPUs=2, Memory=2200MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/re
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass

! C:\Program Files\Docker\Docker\resources\bin\kubectl.exe is version 1.29.2, which may have incompati
  - Want kubectl v1.32.0? Try 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\NEW>minikube kubectl -- get pods -A
    > kubectl.exe.sha256:  64 B / 64 B [----------------------] 100.00% ? p/s 0s
    > kubectl.exe:  56.13 MiB / 56.13 MiB [----------] 100.00% 3.90 MiB p/s 15s
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
kube-system   coredns-668d6bf9bc-7mbdz           1/1     Running   0             71s
kube-system   etcd-minikube                      1/1     Running   0             75s
kube-system   kube-apiserver-minikube            1/1     Running   0             75s
kube-system   kube-controller-manager-minikube   1/1     Running   0             78s
kube-system   kube-proxy-m4zw4                   1/1     Running   0             71s
```

To check the running nodes

```
C:\Users\NEW>kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   5m21s   v1.32.0
```

to get the pods under some name space

```
C:\Users\NEW>kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS        AGE
coredns-668d6bf9bc-7mbdz           1/1     Running   0               5m51s
etcd-minikube                      1/1     Running   0               5m55s
kube-apiserver-minikube            1/1     Running   0               5m55s
kube-controller-manager-minikube   1/1     Running   0               5m58s
kube-proxy-m4zw4                   1/1     Running   0               5m51s
kube-scheduler-minikube            1/1     Running   0               5m55s
storage-provisioner                1/1     Running   1 (5m39s ago)   5m46s
```

Let's Create Pod directly:

```
C:\Users\NEW>kubectl run my-pod --image=nginx --port=80
pod/my-pod created

C:\Users\NEW>kubectl get pods
NAME     READY   STATUS              RESTARTS   AGE
my-pod   0/1     ContainerCreating   0          13s
```

kubectl run: which create a pod

my-pod: name of my pod

--image=nginx: docker image to use for the container

--port=80 expose the port 80 on the container.

When you verify quicky you can see the pod ready status is 0/1 means its not in Ready state.

Lets execute describe command

```
C:\Users\NEW>kubectl describe pod my-pod
Name:             my-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Sat, 08 Feb 2025 16:54:14 +0530
Labels:           run=my-pod
Annotations:      <none>
Status:           Running
IP:               10.244.0.4
IPs:
  IP:  10.244.0.4
Containers:
  my-pod:
    Container ID:   docker://57740038adfd15e157da27a55811de814b7cda101f31fd6f18c9d98f0cbe55c4
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sat, 08 Feb 2025 16:55:02 +0530
    Ready:          True
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  61s   default-scheduler  Successfully assigned default/my-pod to minikube
  Normal  Pulling    60s   kubelet            Pulling image "nginx"
  Normal  Pulled     14s   kubelet            Successfully pulled image "nginx" in 45.699s (45.699
  Normal  Created    14s   kubelet            Created container: my-pod
  Normal  Started    13s   kubelet            Started container my-pod

C:\Users\NEW>kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          96s
```

Here you can see how pod stated the container.

after some time if you see the pod status its running and 1/1 its Ready state.

To get into this pod you can execute below command

```
C:\Users\NEW>kubectl exec -it my-pod -- /bin/bash
root@my-pod:/# ls
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh
root@my-pod:/# exit
exit
```

This pods are not accessible directly in the browser. If you want to access you need to expose the same using service.

kubectl expose pod my-pod --type=LoadBalancer --port=80 --target-port=8080 --name=my-services

kubectl get svc

you can see the available service, then we can access them in browser.