Removing Minikube:

```
minikube stop; minikube delete && docker stop $(docker ps -aq) && rm -rf ~/.kube ~/.minikube && sudo rm -rf /usr/local/bin/localkube /usr/local/bin/minikube && launchctl stop '*kubelet*.mount' && launchctl stop localkube.service && launchctl disable localkube.service && sudo rm -rf /etc/kubernetes/ && docker system prune -af --volumes
```

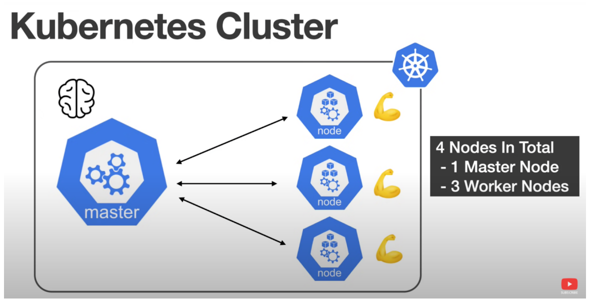
Ref: https://stackoverflow.com/questions/73600518/how-do-i-uninstall-minikube-on-a-mac

Kuberenetes:

- Also known as K8S because of the name Kuberenetes, the underlined characters are 8.
- Its an application orchestrator i.e it orchestrates all of the applications.
- It deploys and manages our applications (containers).
- Scales up & down according to demand
- Zero Downtime Deployments
- Rollbacks
- And more.

Important Terms in K8S Architecture :

1. Cluster:
    - A set of nodes.
    - A node could be a physical machine or a VM
    - These machines can be running on cloud such as AWS, Azure, Google Cloud or even on premises.
    - The cluster consists of master and worker node
    - The master node is brain of the cluster where all decisions are made & worker node is the one like muscles where all the heavy lifting happens such as running our applications.
    - Both the master & worker node communicate with each other using kubelet.
    - Generally there are many worker nodes but only 1 or 2 master node.
    - For this example: we have 4 nodes classified into 1 master node & 3 worker nodes:



    - To create local cluster we are going to use minkube.(Need docker + minkube)

2. Master Node:
    - It contains control plane.
    - In Control plane there are several components such as api server, scheduler, cluster store, controller manager & cloud controller manager (which communicates with cloud api such as google cloud,aws,azure) .
    - All these components communicate through api server

    a. Api Server:
        1. Frontend to K8S control plane
        2. All communication go through API Server whether its External communication & Internal communication
        3. Exposes RestFul API on Port 443.
        4. Authentication & AUthorization checks are also done

    b. Cluster Store | State:
        1. Stores the configuration & state of the entire cluster
        2. Uses etcd(a distributed key value data store) to store,etcd is a single source of truth. Etcd is a cluster brain

    c. Scheduler:
        1. Watches for new workloads/pods and assigns them to a node based on several scheduling factors.
        2. Checks whether node is heathy?
        3. Does node have enough resources.
        4. Is port available?
        5. Affinity & Anti-Affinity rules

    d. Controller Manager:
        1. Daemon that manages the control loop. Controller of controllers
        2. Node Controller.(checks whether current state matches the desired state, if it doesn't it simply kicks in.)
        3.Other controllers are: ReplicaSet,Endpoint, Namespace.Service Accounts
        4. Each controller watches api server for changes

    e. Cloud Controller Manager:
        Responsible for interacting with underlying cloud provider.

3. Worker Node:

- Its a VM / Physical machine running on linux
- Provides runnning env for these applications
- It has 3 components:
    a. Kubelet
        1. Main agent that runs on every node
        2. Receives pod definitions from API Server
        3. Interacts with container runtime to run containers associated with pod.
        4. Report Node & Pod state to master through api server.

b. Container Runtime
    1. Reponsible for pulling image from container registries such docker hub, gcr, ecr,acr.
    2. Responsible for starting,stopping & running containers & abstracts container management for kubernetes
    3. Has container runtime interface - interface for 3rd party container runtime,containerd

c. Kube Proxy
    1. An agent that runs on everyone through Daemon Sets
    2. Responsible for: local cluster networking, each node gets unique IP, Routing n/w traffic to node balanced services.
    3. Eg. if 2 pods wants to talk each other, kube proxy comes into picture.

4. Kubectl:

- Once you've local cluster running up with minikube, then to interact with it, We need kubectl.
- Its a K8S command line tool

- Run commands against our cluster
    1. Deploy applications
    2. Inspect
    3. Edit resources
    4. Debug our cluster
    5. View Logs

5. Pods:

- A pod is the smallest deployable unit in K8S.
- It contains main container which represents your application and may/may not have init container which runs before main container, and may/may not have side container to support main container.
- Also it contains volumes to share data among themselves(containers)
- The containers inside pod,communicate to each other using localhost ports
- The pod also has a unique ip, it means another pod wants to talk to this pod then it uses this unique ip address.

Notes:

1. Deployment is for stateless apps
2. Stateful Set is for stateFul Apps or Databases.
3. Kubectl cli is used for configuring the minikube cluster
4. Minikube cli is used for starting up/deleting the cluster

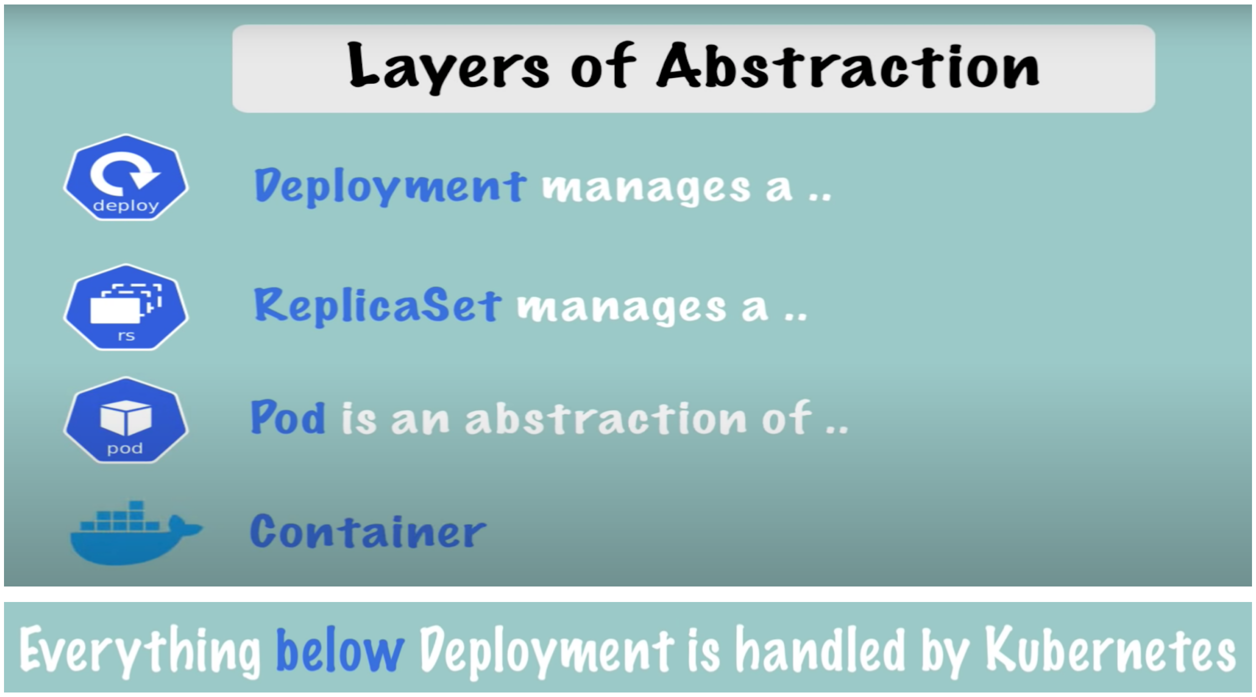
Kubectl commands:

1. kubectl get nodes: Get nodes i,e minikube ,it's roles & version.
2. minikube status: returns all the nodes with detailed description
3. kubectl version: returns versions of kubectl command line
4. kubectl get pod: returns pods which are created, pod-name is usually = deployment(prefix)+replicaset's id+ pod's id
5. kubectl get services: returns services which are created.
6. kubectl create deployment [name] --image = image -> blueprint for creating pods,most basic configuration for deployment.(name & image to use), eg: kubectl create deployment nginx-depl --image=nginx . **Create,Delete& Update happens at deployment level**
7. kubectl get deployment: returns the deployment which is created
8. kubectl get replicaset-> gets replicaset, manages replicas of pod

```
 Shubhams-MacBook-Air:~ shubhamphansekar$ kubectl get pod
        NAME                    READY   STATUS    RESTARTS   AGE
nginx-depl-c88549479-jv5f8   1/1      Running        0        75s
Shubhams-MacBook-Air:~ shubhamphansekar$ kubectl get replicaset
        NAME                DESIRED   CURRENT   READY   AGE
        nginx-depl-c88549479   1          1          1     3m48s
```

NOTE: the replicaset name has prefix & middle part is same as pod name

9. kubectl edit deployment [name] -> gives a auto-generated configuration file with default values eg. kubectl edit deployment nginx-depl
10. kubectl logs [pod_name] -> returns logs inside of the container.
11. kubectl exec -it [pod_name] /bin/bash -> can go into interactive terminal of the pod
12. kubectl delete deployment [deplyoment_name] -> It is used to delete the deployment. eg.kubectl delete deployment nginx-depl
13. kubectl apply -f [filename] -> it is used to create deployment with various option by launching it with help of configuration file
14. kubectl delete -f [filename] -> it is used to delete config file along with deployment inside it.



Layers of Abstraction

Deployment manages a ..

ReplicaSet manages a ..

Pod is an abstraction of ..

Container

Everything below Deployment is handled by Kubernetes

15. kubectl apply -f nginx-deployment.yaml -> used to create/update deployments. If you already created a pod using using this, and then change replica to 2 it will update the pods to 2 rather than creating one.In general we can creat/update various components such as deployment/service/volumes.
16. kubectl describe service service_name -> in general describe gives detailed info, here it's giving for service.
17. minikube service service_name-> returns an external url, which can be typed on browser to access servicer . eg. minikube service mongo-express-service.
18. kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml-> for creating kuberenetes-dashboard
19. minikube addons enable ingress

Configuration file:
Each configuration file has 3 parts:
1. Metadata: contains name
2. Specification: contains specifications such as replicas i.e parameters related to that component
3. Status: this is automatically generated & added by kubernetes. It's stored in etcd.

To run the yaml files: -
Step 1: kubectl apply -f mongo-secret.yaml
Step 2: kubectl apply -f mongo.yaml

Step 3:kubectl apply -f mongo-configmap.yaml
Step 4: kubectl apply -f mongo-express.yaml
Step 5: minikube service mongo-express-service

Namespace:

1. Organizes resources in namespaces
2. Think namespace as a virtual cluster inside a k8s cluster
3. When we create a cluster by default k8s gives you namespaces out of the box.
4. cmd for it -> kubectl get namespace
   4 namespaces per default:
   a. kube-system: do not create or modify in kube-system. The components deployed here are system processes. Master & Kubectl processes
   b. kube-public: has public accessible data,such as config-map which contains cluster info. Cmd: kubectl cluster-info
   c. kube-node-lease: hold info about heartbeats of nodes, determines the availabitilty of nodes
   d. default: resources which we create are located here.