

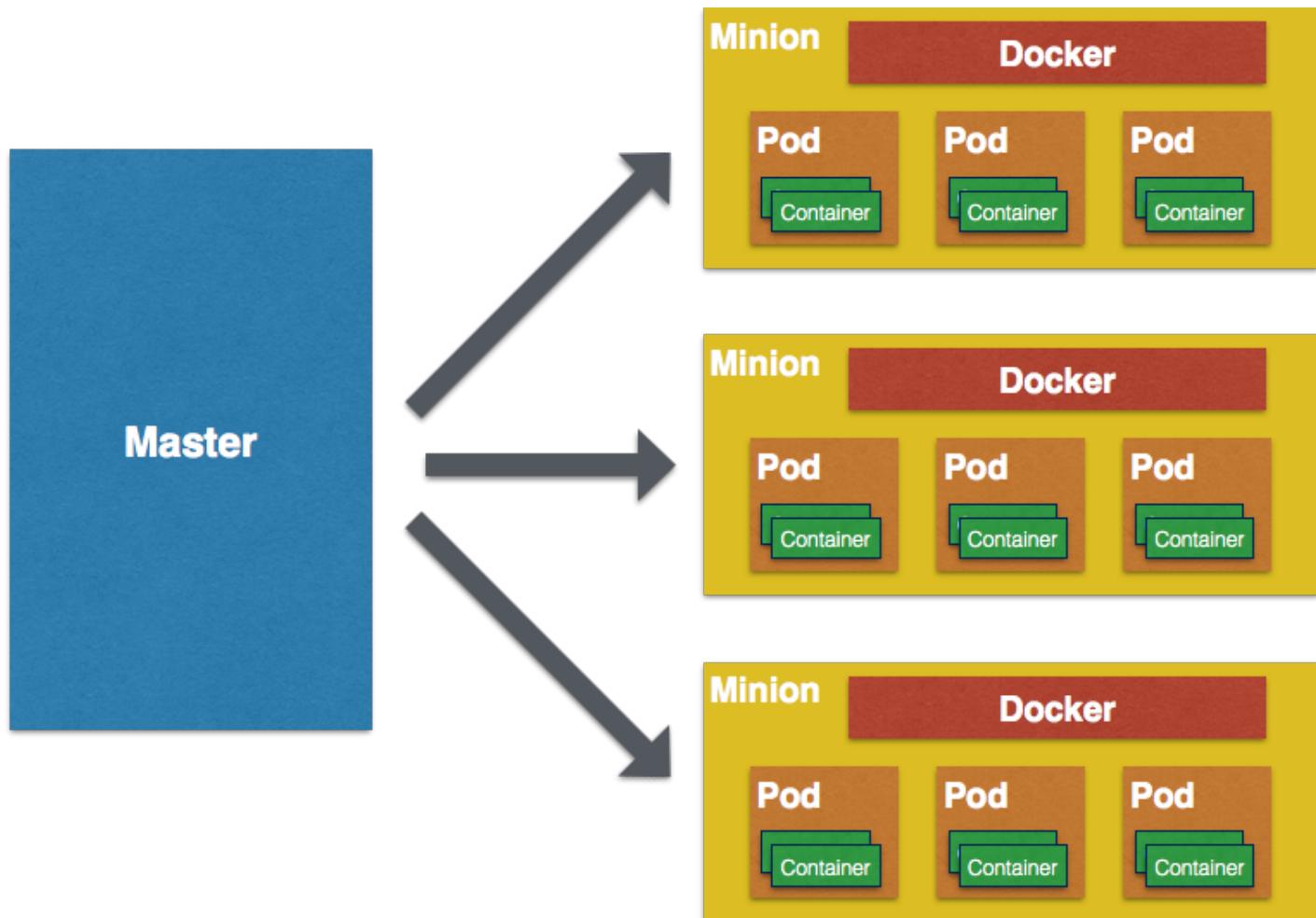
KUBERNETES



WHAT IS KUBERNETES

- abbreviated as k8s
- open-source system for automating deployment, scaling, and management of containerized applications
- originally designed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF)



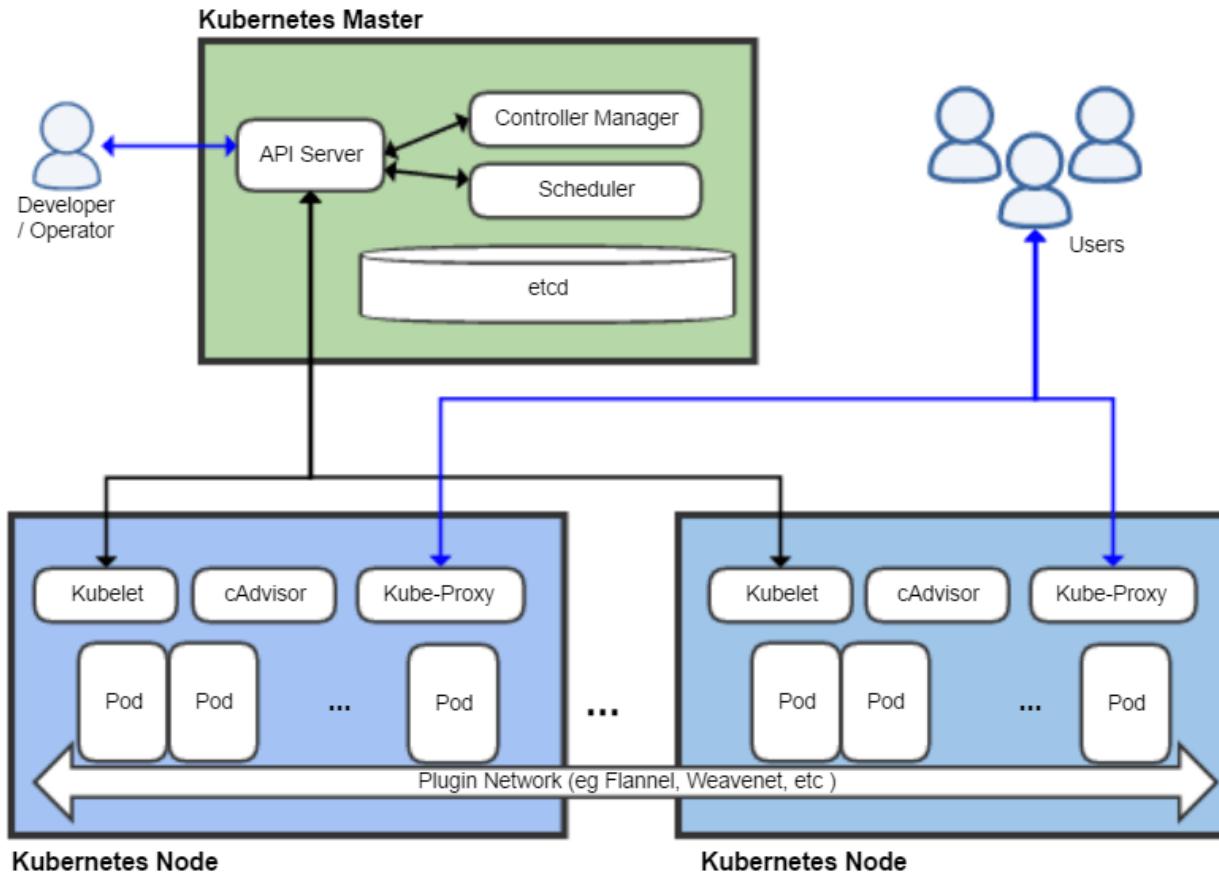


High Level Diagram of Kubernetes Cluster

NOTE:

- Kubernetes officially calls minion a Kubernetes node

ARCHITECTURE

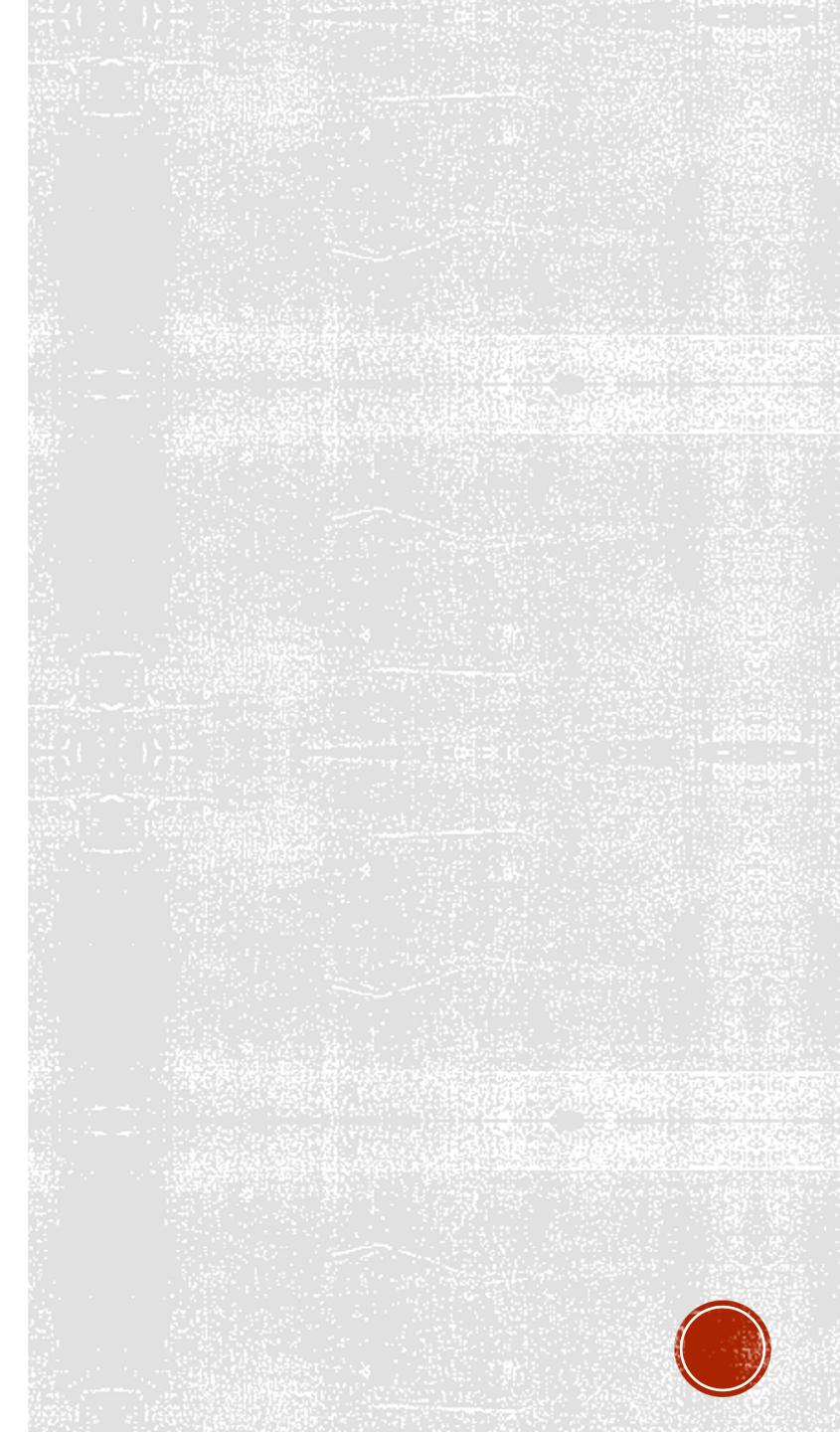


CLUSTER MEMBERS

- **Master** : makes up the Control Plane and manages the cluster
- **Nodes (slaves)** : makes up the Data Plane and runs the actual container images (via pods)

SOME K8S OBJECTS

- **Deployments**
 - High-level construct that defines an application
- **Pods**
 - Instance of a container in a deployment
 - Deployments can have any number of pods required to get the job done
- **Services**
 - Expose an application running on a set of pods as a network service
- **Namespace**
 - Allows to separate cluster into smaller logical clusters
- **ReplicaSets**
 - Maintains a stable set of replica Pods running at any given time



HOW TO GET STARTED

1. Create a k8s cluster
 - Master
 - Node
2. Setup kubectl
 - a CLI for running commands against k8s cluster
3. Create a deployment template which can be in JSON or YAML format
4. Deploy the template

\$ kubectl create -f deployment.yml

5. Check the deployment

\$ kubectl get deployments

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	2	2	2	2	30s

SAMPLE DEPLOYMENT YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # deploy 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

KUBECTL

Allows to run commands against Kubernetes clusters

Some basic kubectl commands:

- Get pods

```
$ kubectl get pods
```

- Describe pods/deployments

```
$ kubectl describe deployment <deployment-name>
```

```
$ kubectl describe pod <pod-name>
```

- Check logs on the pod

```
$ kubectl logs -f <pod-name>
```

- Execute command on a pod

```
$ kubectl exec -it <pod-name> -- <command>
```

- Create a service of type load balancer

```
$ kubectl expose deployment <deployment-name> --type=LoadBalancer
```

```
$ kubectl get service
```

- Delete deployment

```
$ kubectl delete deployment <deployment-name>
```

HEALTH CHECKS

- Readiness Probes
 - To determine when a pod is ready to respond to requests
 - If the readiness probe fails, the endpoints controller removes the Pod's IP address from the endpoints of all Services that match the Pod.
- Liveness Probes
 - To determine when a pod is healthy or unhealthy after it has become ready
 - If liveness probe fails, kubelet kills the container and the container is subjected to its restart policy

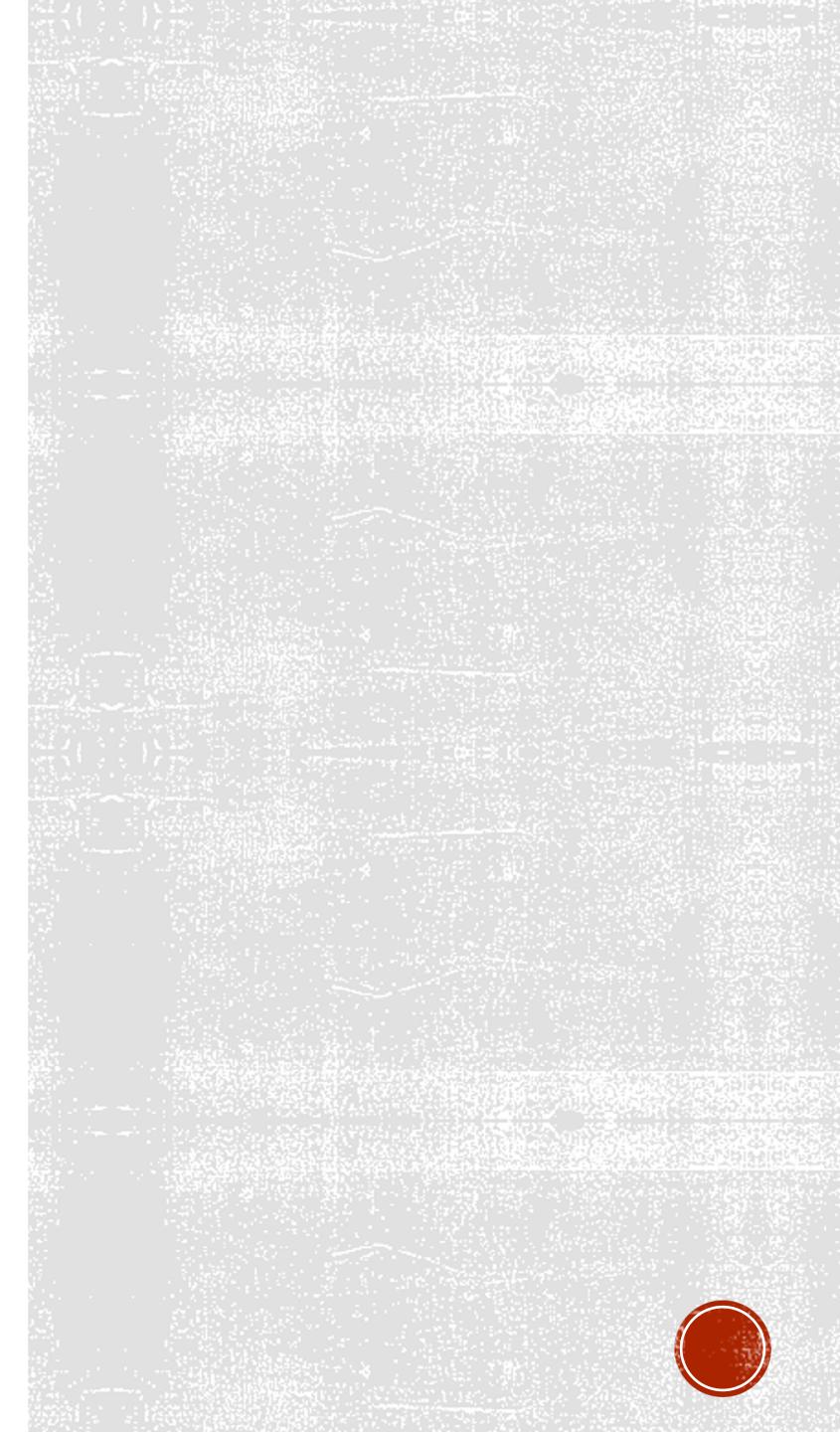
Probes are defined on the container in a Deployment or Pod specification

PARTIAL DEPLOYMENT YAML WITH HEALTH CHECKS

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hc-deployment
---
spec:
  ---
  ports:
    - containerPort: 80
  livenessProbe:
    httpGet:
      path: /test
      port: 80
    initialDelaySeconds: 30
    periodSeconds: 30
  readinessProbe:
    httpGet:
      path: /
      port: 80
    initialDelaySeconds: 15
    periodSeconds: 3
```

MULTI-SERVICE DEPLOYMENT

- Every service in Kubernetes cluster gets a DNS name
- Nomenclature for deciding what DNS name,
`<service-name>.<namespace>.svc.cluster.local`
- An application pod can connect to another deployment pod(s) by using the DNS name given to its service



IN CLOUD

- AWS – Elastic Kubernetes Service (EKS)
- GCP – Google Kubernetes Engine (GKE)
- Azure – Azure Kubernetes Service (AKS)



GET STARTED ON KUBERNETES

- Homepage
 - <https://kubernetes.io>
- Tutorial with Interactive Hands-on lab with minikube
 - <https://kubernetes.io/docs/tutorials/kubernetes-basics/>



REFERENCES

- <https://kubernetes.io>
- <https://en.wikipedia.org/wiki/Kubernetes>

