kubectl

minikube

master/ : : worker node

minikube-cluster

linux

mac --driver vmware
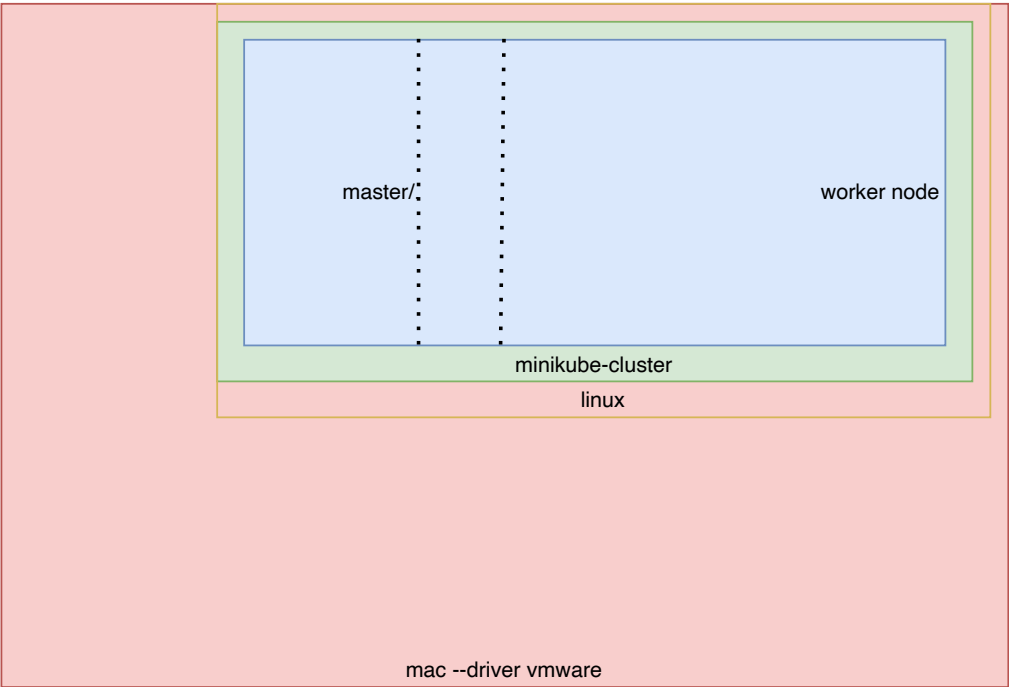
kubernetes objects

imperatively

docker run --name -p -it -e

Declarative

config yaml files

Pods

Deployments

services

configmap

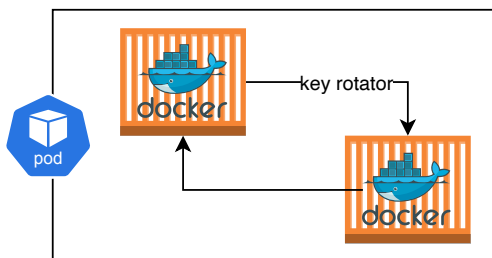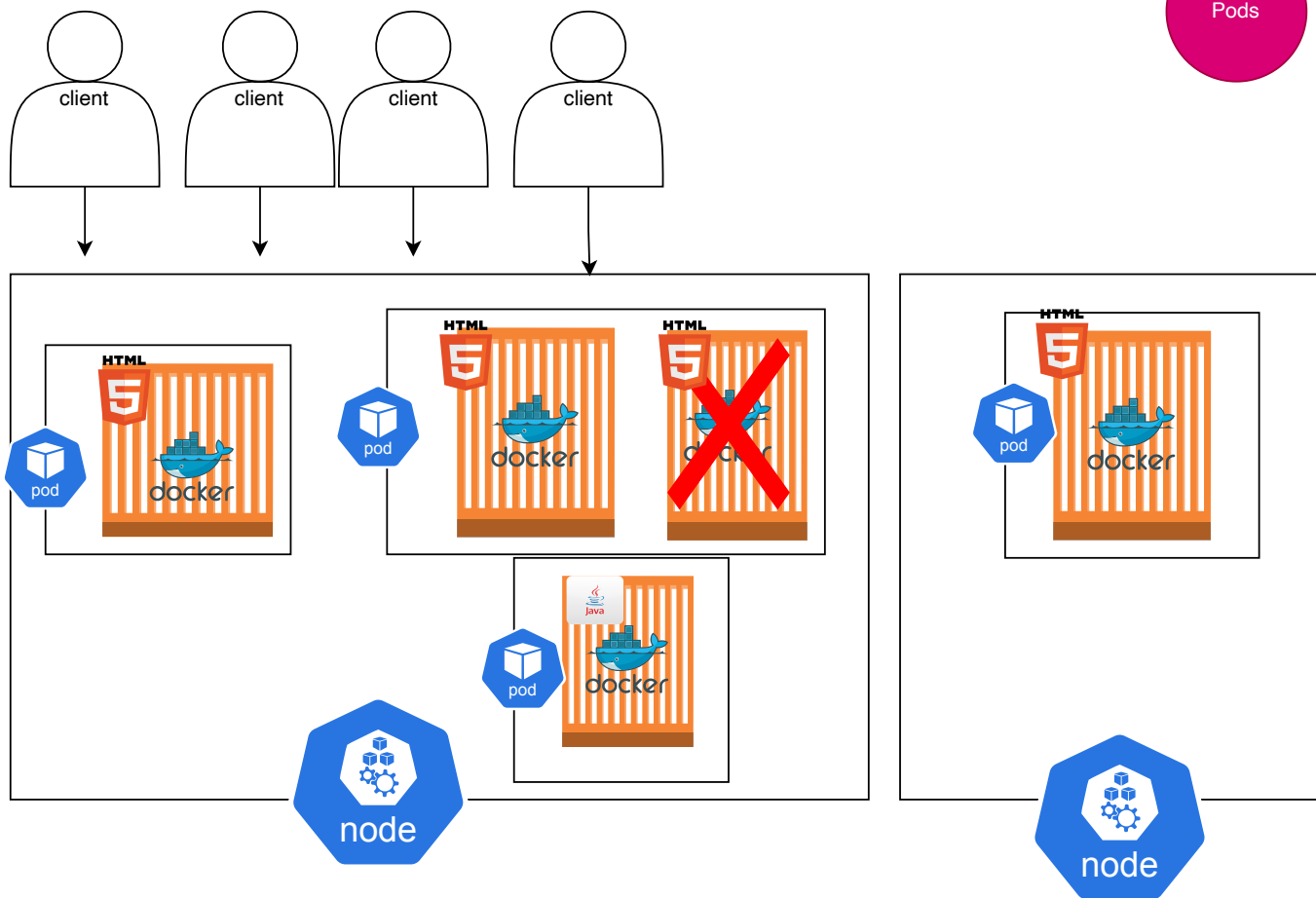statfulset

deamonset

secrets

volume

replicaset

Pods

client client client client

pod

HTML
5
docker

pod

HTML
5
docker

HTML
5
docker

pod

java
docker

node

pod

HTML
5
docker

node

pod

docker key rotator

docker

how to create a pod ?

kubectl run nginx --image nginx --port 80 --name blah blah

to create the pod use

kubectl apply -f file.yml

list all pods

kubectl get pods

describe the pod

kubectl describe pod/<podname>

```
pod.yml

apiVersion: v1      String
kind: Pod           String
metadata:           dict
  name: nginx-pod
  labels:           dict
    app: app1
    type: front-end
spec:               dict
  containers:       list
    - name: nginx-container
      image: nginx
```

replicaset

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-replica-set
  labels:
    app: nginx-rs
spec:
  replicas: 2          ← scaling
  template:
    metadata:
      name: nginx-pod
      labels:
        app: app1
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  selector:
    matchLabels:
      type: front-end
```

← pod.yml

label matching

to create the pod use

kubectl apply -f file.yml

list all rs

kubectl get rs

describe the rs

kubectl describe rs/<podname>

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx-rs
spec:
  replicas: 2
  template:
    metadata:
      name: nginx-pod
      labels:
        app: app1
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  selector:
    matchLabels:
      type: front-end
```

deployment

to create the pod use

kubectl apply -f file.yml

list all deployments

kubectl get deployments

describe the pod

kubectl describe deployments/<podname>

services

service — ingress

service

1
1
1

service

2
2
2

3
3
3

service

db

192.168.1.2

10.244.0.0

service
nodePort: 30008
targetPort: 80
port: 3080

nodePort
30008

80
10.244.0.3
app: app1
pod
docker

80
10.244.0.2
app: app1
pod
docker

node

curl http://192.168.1.2:30008/hello