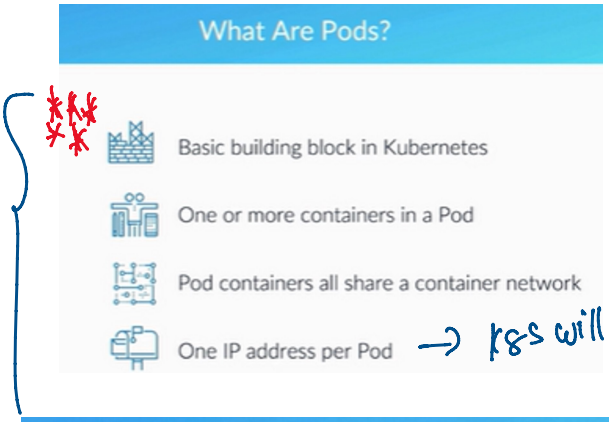
2. Pods

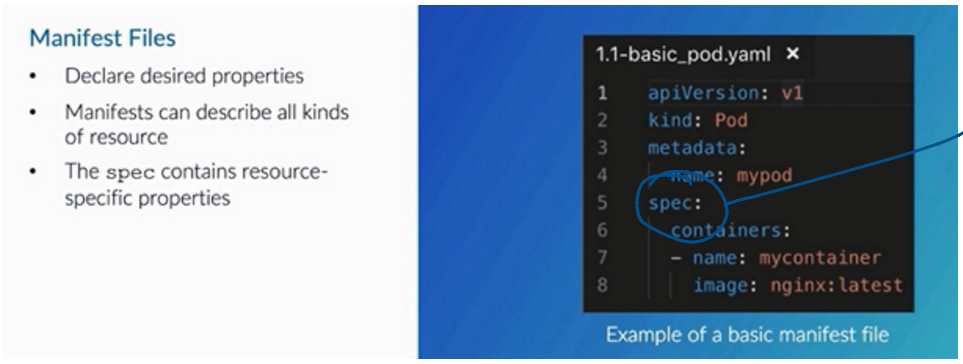
27 February 2022 14:42



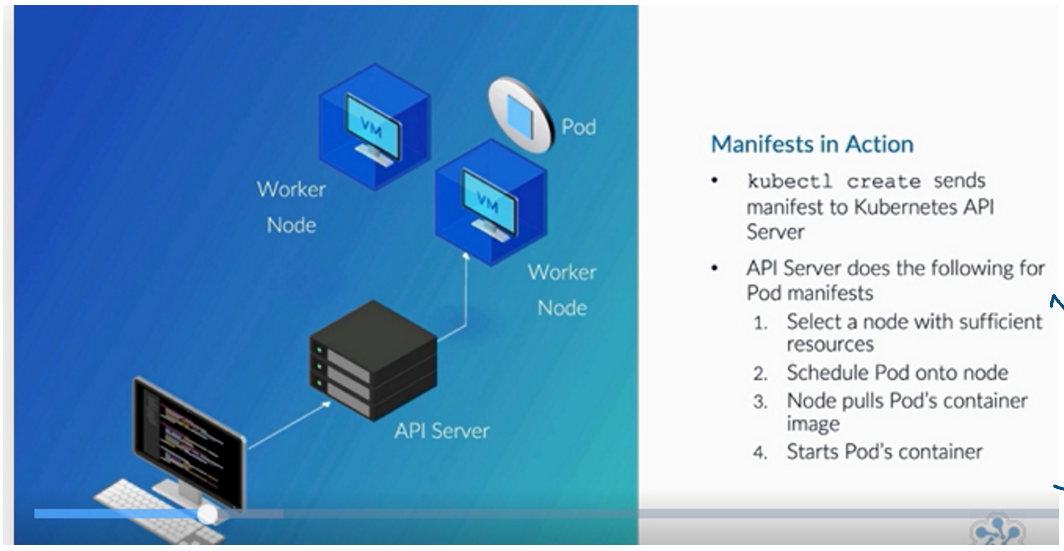
→ rss will take care of it.

when to restart if fails.

CPU/memory



configuration specific to a resource.




```
admin@Jamess-MBP Cloud Academy % minikube start
😄  minikube v1.12.0 on Darwin 10.15.4
✨  Using the hyperkit driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing hyperkit VM for "minikube" ...
🐳  Preparing Kubernetes v1.18.3 on Docker 19.03.12 ...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube"
admin@Jamess-MBP Cloud Academy % kubectl get pods
No resources found in default namespace.
admin@Jamess-MBP Cloud Academy % cd src
admin@Jamess-MBP src % cat 1.1-basic_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mycontainer
    image: nginx:latest
admin@Jamess-MBP src %
```

apiVersion

must include a name

all manifests will have high level some fields

single container with image & name

```
admin@Jamess-MBP src % kubectl create -f 1.1-basic_pod.yaml
pod/mypod created
admin@Jamess-MBP src %
```

create using a file

describe → to describe details.

Just like docker we need to tell the k8s which port we can be accessible.

```
Namespace:    default
Priority:     0
Node:         minikube/192.168.64.2
Start Time:   Fri, 17 Jul 2020 12:09:23 -0400
Labels:       <none>
Annotations:  <none>
Status:       Running
IP:           172.17.0.3
IPs:
  IP:  172.17.0.3
Containers:
  mycontainer:
    Container ID:   docker://dd4aa6999d64bd425d97b09dbca5b93b72ec9c4314c1bad3f593f6579c
707f
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:a93c8a0b0974c967aebe868a186e5c205f4d
cb5423a56559f2f9599074bbcd
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Fri, 17 Jul 2020 12:11:48 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-7gjvg (ro)
Conditions:
:
```

port not provided.

```
admin@Jamess-MBP src % cat 1.2-port_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mycontainer
    image: nginx:latest
    ports:
      - containerPort: 80
admin@Jamess-MBP src %
```

→ Port

K8's cannot update Ports for the running pod.

() we need to delete, update & recreate.

Kubectl delete pod my pod

(or)

kubectl delete -f filename

```
Priority:     0
Node:         minikube/192.168.64.2
Start Time:   Fri, 17 Jul 2020 12:19:34 -0400
Labels:       <none>
Annotations:  <none>
Status:       Running
IP:           172.17.0.3
IPs:
  IP:  172.17.0.3
Containers:
  mycontainer:
    Container ID:   docker://78edf6a17d358ff9cb33e501c74086a9cf56a80170c66b4a0e0
5c05
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:a93c8a0b0974c967aebe868a186e5
cb5423a56559f2f9599074bbcd
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Fri, 17 Jul 2020 12:19:36 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-7gjvg (ro
Conditions:
admin@Jamess-MBP src % curl 172.17.0.3:80
```

→ will not work.

Because pod's ip is in the container network.

```
Name:         mypod
Namespace:    default
Priority:     0
Node:         minikube/192.168.64.2
Start Time:   Fri, 17 Jul 2020 12:19:3
Labels:       <none>
Annotations:  <none>
Status:       Running
IP:           172.17.0.3
IPs:
```

→ Key value pair like region.

() helps to get pods in that region using label.

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: webserver
spec:
  containers:
  - name: mycontainer
    image: nginx:latest
    ports:
      - containerPort: 80
admin@Jamess-MBP src %
```

→ we can have multiple labels.

Its a good idea to set resource requests so that

Its a good idea to set resource requests so that API server knows to which node to schedule based on resource availability.

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: webserver
spec:
  containers:
  - name: mycontainer
    image: nginx:latest
    resources:
      requests:
        memory: "128Mi" # 128Mi = 128 mebibytes
        cpu: "500m"     # 500m = 500 milliCPUs (1/2 CPU)
      limits:
        memory: "128Mi"
        cpu: "500m"
    ports:
      - containerPort: 80
admin@Jamess-MBP src %
```

minimum req resource to schedule pod on node

maximum resource a node can assign to pod.

Pod will be guaranteed this service.

```
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-7gjvg (ro
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-7gjvg:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-7gjvg
    Optional:    false
QoS Class:       Guaranteed
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
```