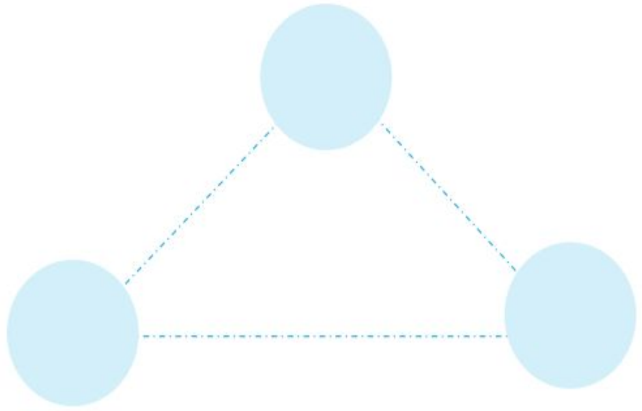
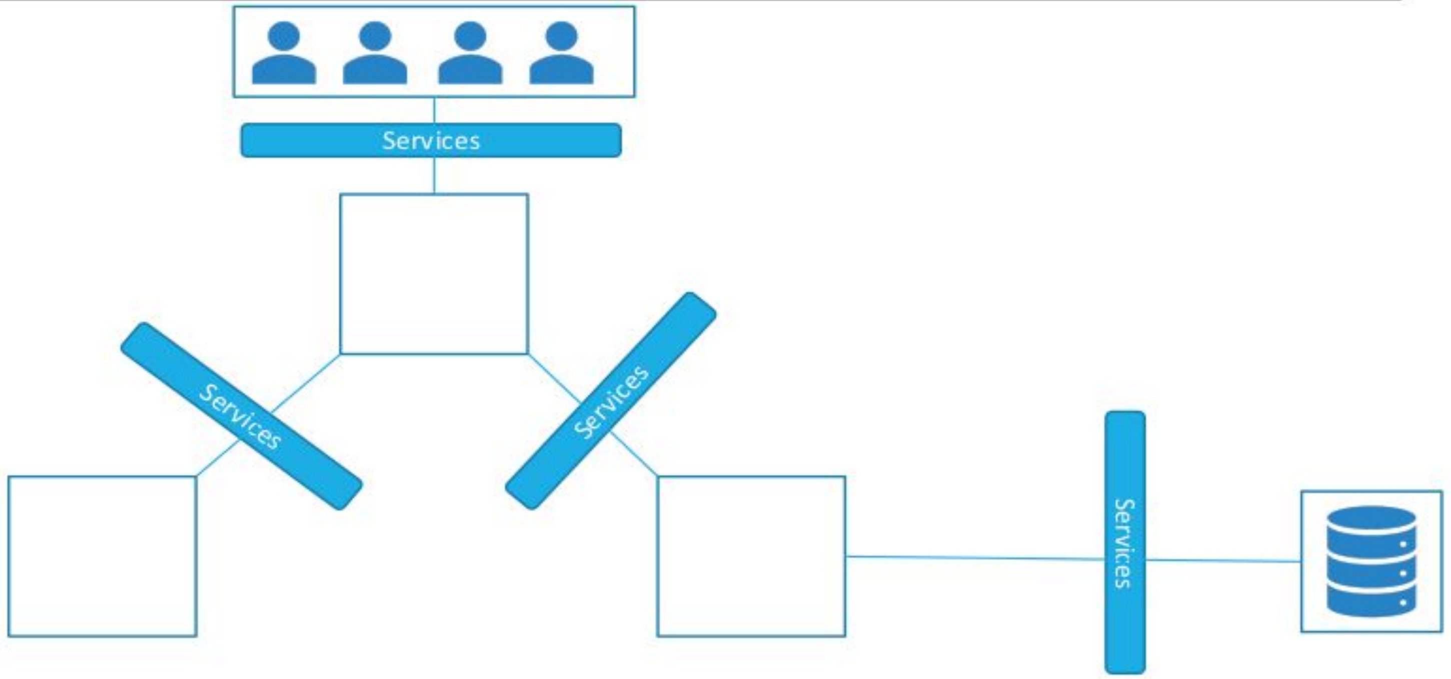
Agenda

- Services
- StartUp & Readiness & Liveness
- Static pods
- DaemonSets
- Init containers
- Multi containers

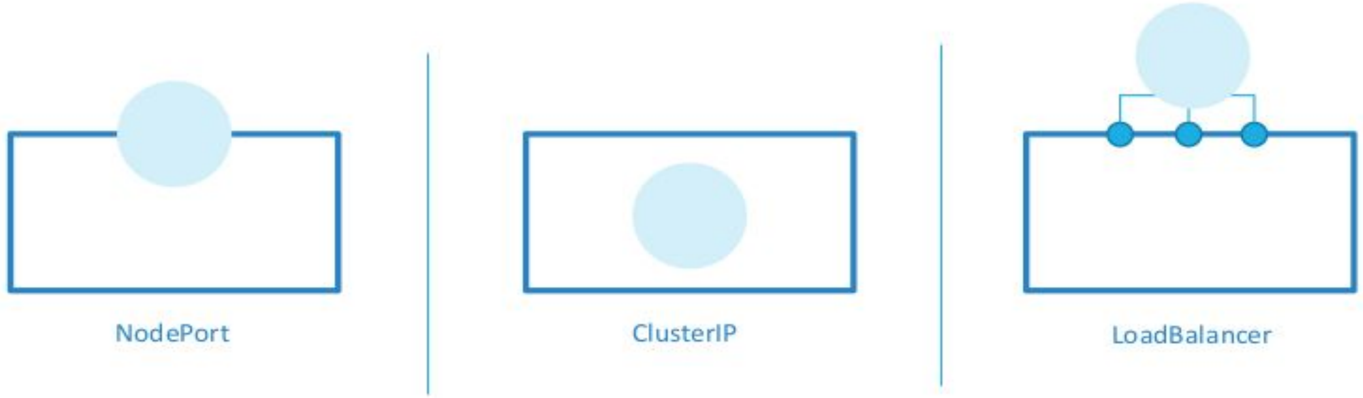
Services



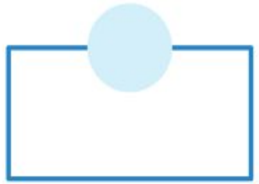
Services



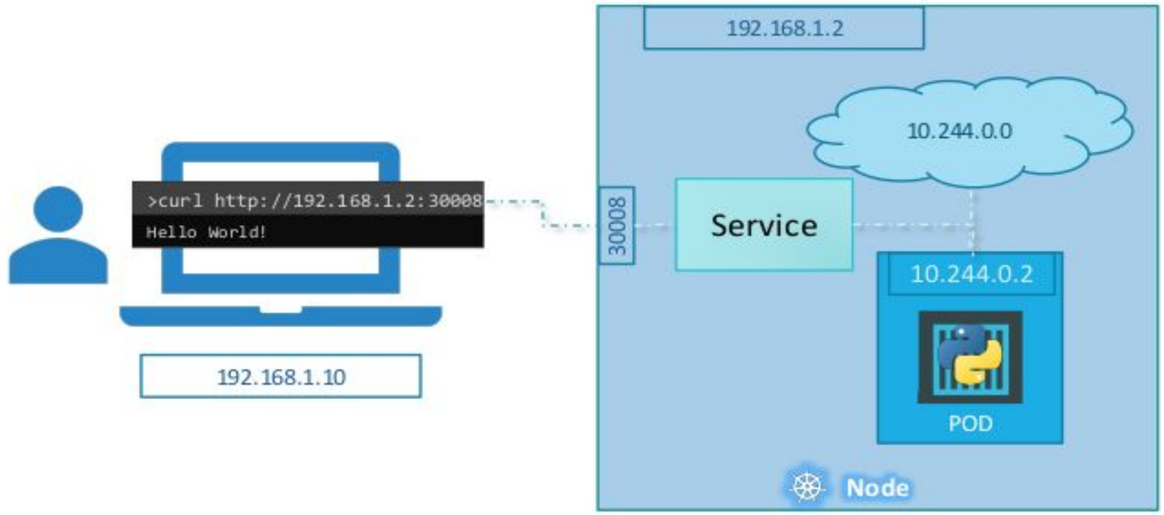
Services Types



NodePort

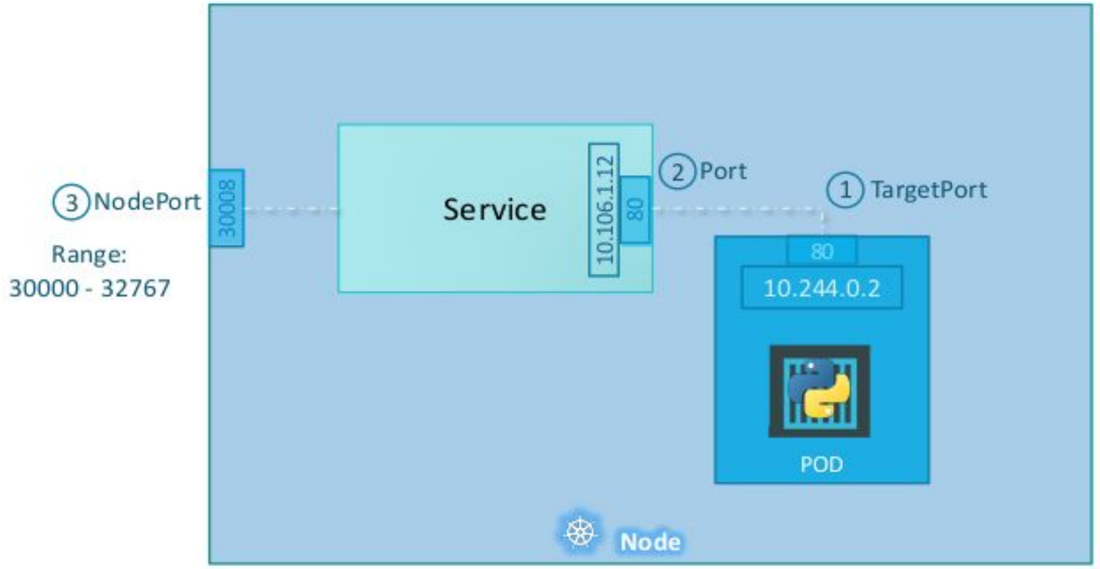


Service - NodePort

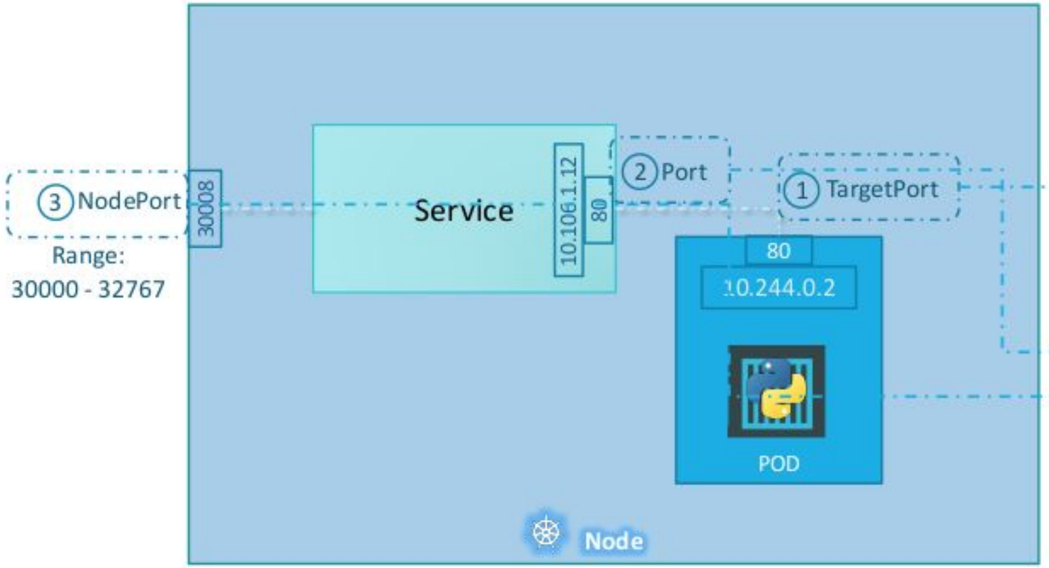


```
>curl http://192.168.1.2:30008
Hello World!
```

Service - NodePort



Service - NodePort



service-definition.yml

```yaml
apiVersion: v1
kind: Service
metadata:
    name: myapp-service

spec:
    type: NodePort
    ports:
     - targetPort: 80
       port: 80
       nodePort: 30008
```

Service - NodePort

```
service-definition.yml
apiVersion: v1
kind: Service
metadata:
    name: myapp-service

spec:
    type: NodePort
    ports:
     - targetPort: 80
       port: 80
       nodePort: 30008
    selector:
```

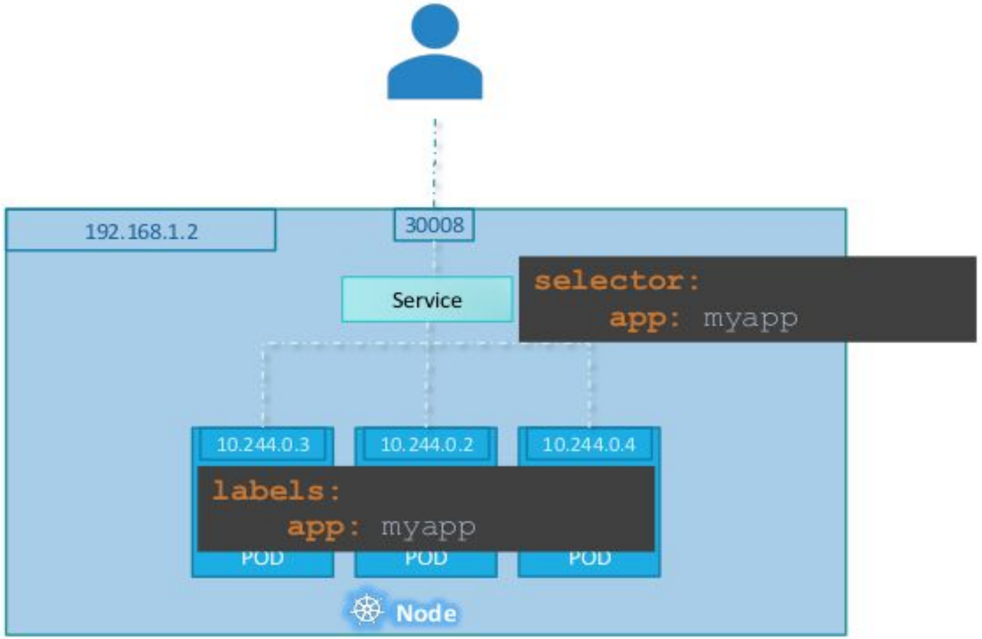
```
pod-definition.yml
        app: myapp
```

```
> kubectl create –f service-definition.yml
service "myapp-service" created
```

```
> kubectl get services
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP        16d
myapp-service   NodePort    10.106.127.123   <none>        80:30008/TCP   5m
```

```
> curl http://192.168.1.2:30008
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
```

Service - NodePort

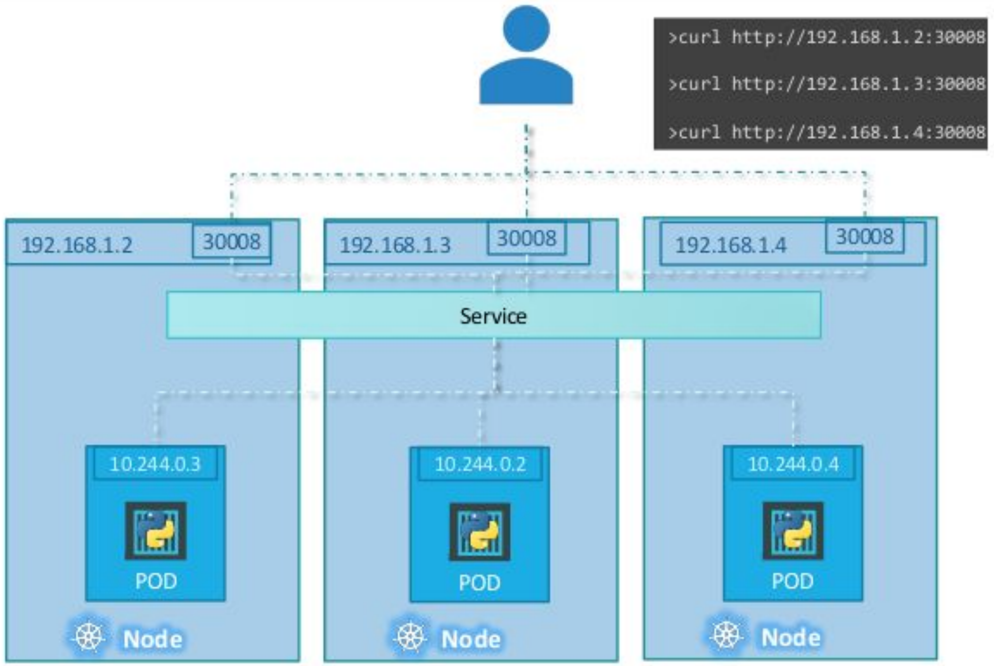


```
selector:
    app: myapp
```

```
labels:
    app: myapp
```

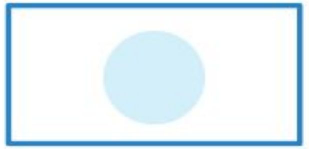
Algorithm: Random
SessionAffinity: Yes

Service - NodePort

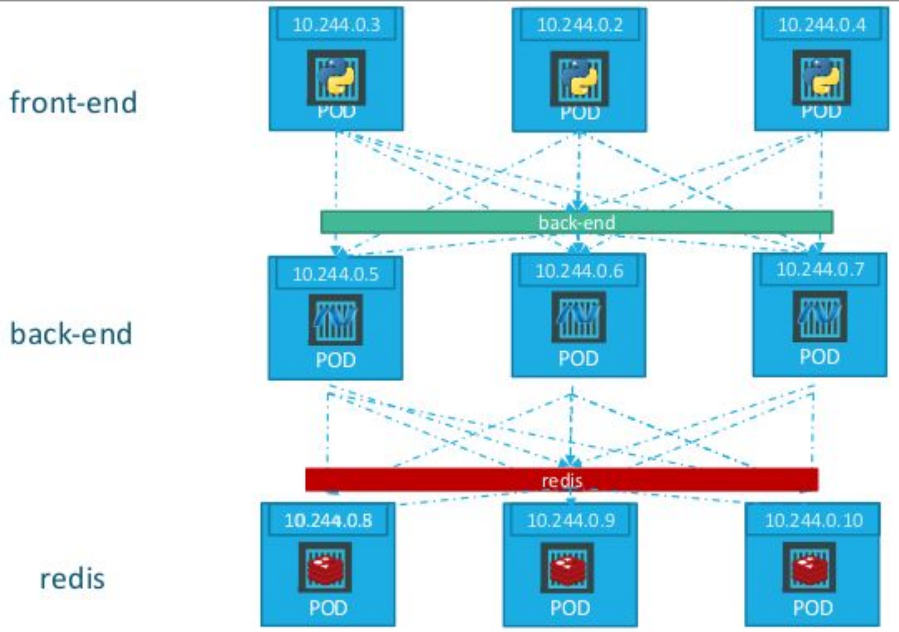


```
>curl http://192.168.1.2:30008

>curl http://192.168.1.3:30008

>curl http://192.168.1.4:30008
```

ClusterIP



ClusterIP


front-end
back-end
redis
10.244.0.3
10.244.0.2
10.244.0.4
POD
back-end
10.244.0.5
10.244.0.6
10.244.0.7
redis
10.244.0.8
10.244.0.9
10.244.0.10

service-definition.yml

```yaml
apiVersion: v1
kind: Service
metadata:
    name: back-end

spec:
    type: ClusterIP
    ports:
     - targetPort: 80
       port: 80

    selector:
```

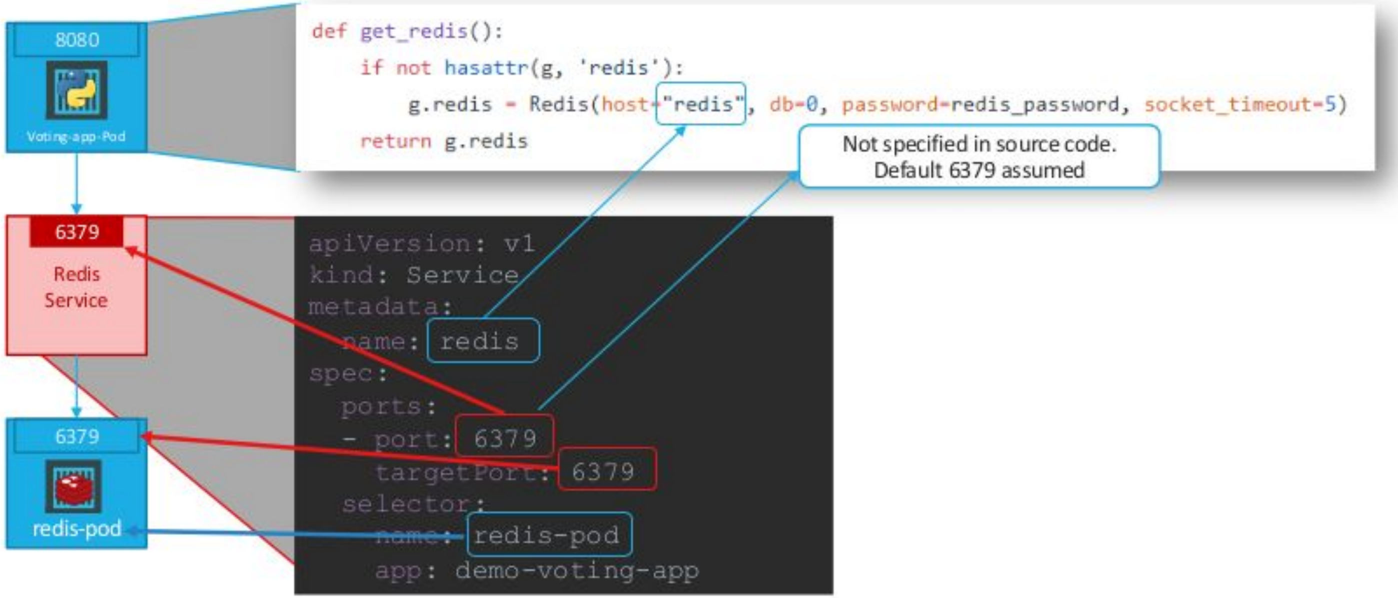
pod-definition.yml

```
> kubectl create -f service-definition.yml
service "back-end" created
```

```
> kubectl get services
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP    16d
back-end      ClusterIP   10.106.127.123  <none>        80/TCP     2m
```

```yaml
        app: myapp
        type: back-end
spec:
  containers:
  - name: nginx-container
    image: nginx
```

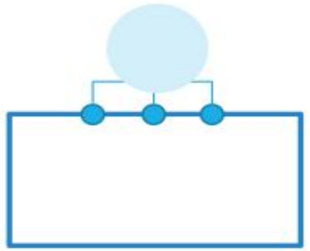
Service



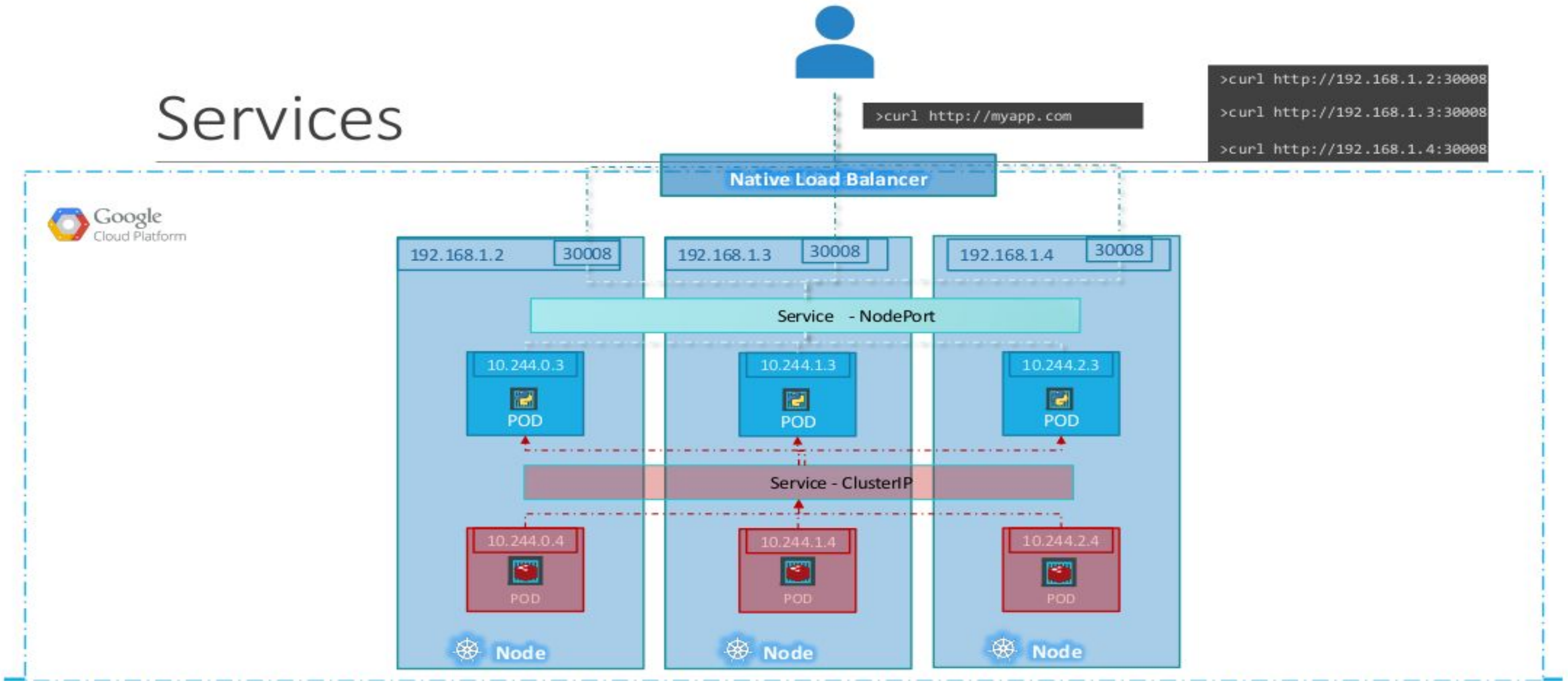
```
def get_redis():
    if not hasattr(g, 'redis'):
        g.redis = Redis(host="redis", db=0, password=redis_password, socket_timeout=5)
    return g.redis
```

```
apiVersion: v1
kind: Service
metadata:
  name: redis
spec:
  ports:
  - port: 6379
    targetPort: 6379
  selector:
    name: redis-pod
    app: demo-voting-app
```

Service - LoadBalancer

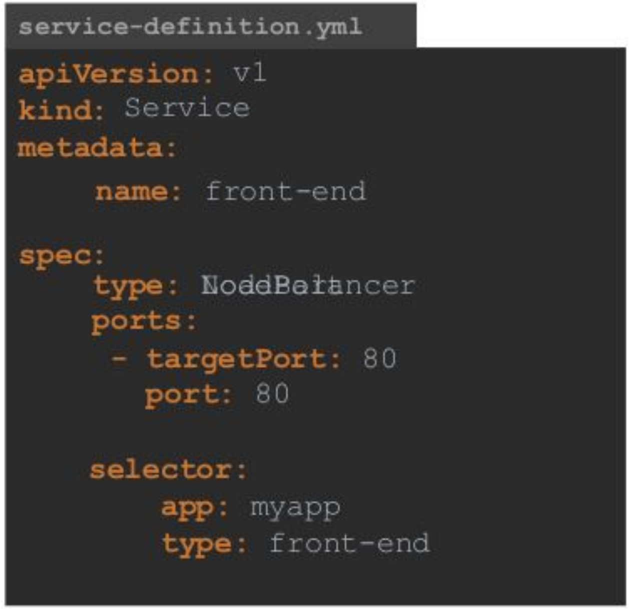


Services

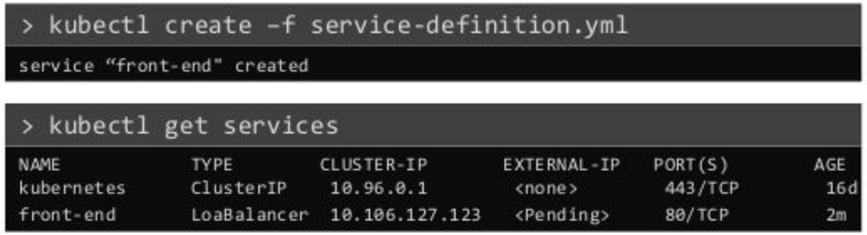


```
>curl http://myapp.com
```

```
>curl http://192.168.1.2:30008
>curl http://192.168.1.3:30008
>curl http://192.168.1.4:30008
```

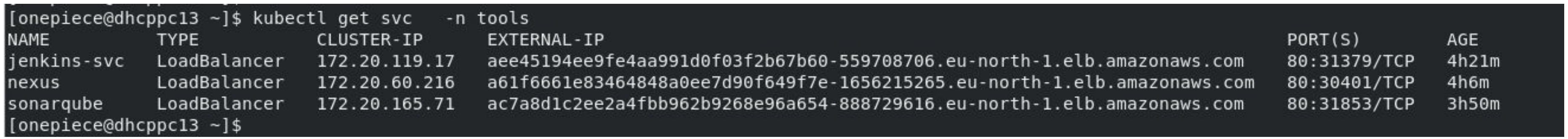


```
service-definition.yml
apiVersion: v1
kind: Service
metadata:
     name: front-end

spec:
     type: LoadBalancer
     ports:
      - targetPort: 80
        port: 80

     selector:
         app: myapp
         type: front-end
```



```
> kubectl create -f service-definition.yml
service "front-end" created
```

```
> kubectl get services
NAME            TYPE         CLUSTER-IP       EXTERNAL-IP   PORT(S)     AGE
kubernetes      ClusterIP     10.96.0.1        <none>        443/TCP     16d
front-end       LoaBalancer   10.106.127.123   <Pending>     80/TCP       2m
```

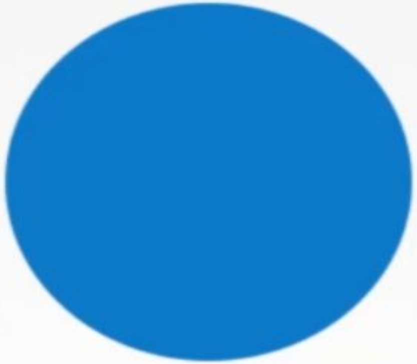


```
[onepiece@dhcppc13 ~]$ kubectl get svc   -n tools
NAME          TYPE           CLUSTER-IP      EXTERNAL-IP                                                               PORT(S)        AGE
jenkins-svc   LoadBalancer   172.20.119.17   aee45194ee9fe4aa991d0f03f2b67b60-559708706.eu-north-1.elb.amazonaws.com    80:31379/TCP   4h21m
nexus         LoadBalancer   172.20.60.216   a61f6661e83464848a0ee7d90f649f7e-1656215265.eu-north-1.elb.amazonaws.com   80:30401/TCP   4h6m
sonarqube     LoadBalancer   172.20.165.71   ac7a8d1c2ee2a4fbb962b9268e96a654-888729616.eu-north-1.elb.amazonaws.com    80:31853/TCP   3h50m
[onepiece@dhcppc13 ~]$
```

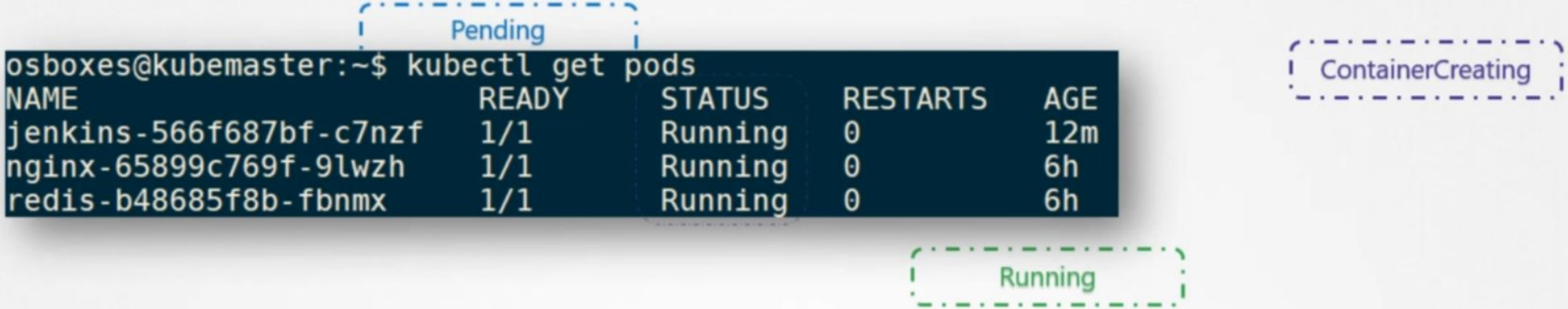
Readiness Probes

POD Status

POD Conditions

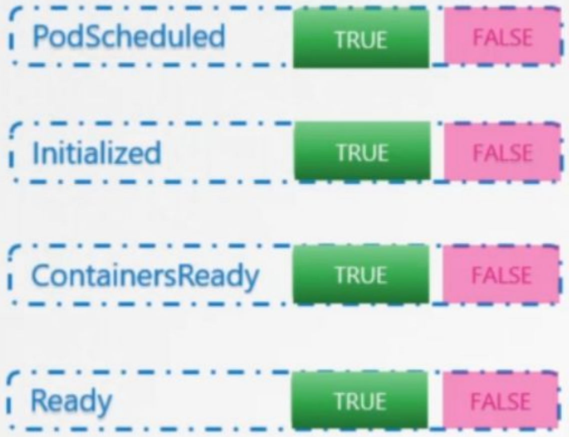


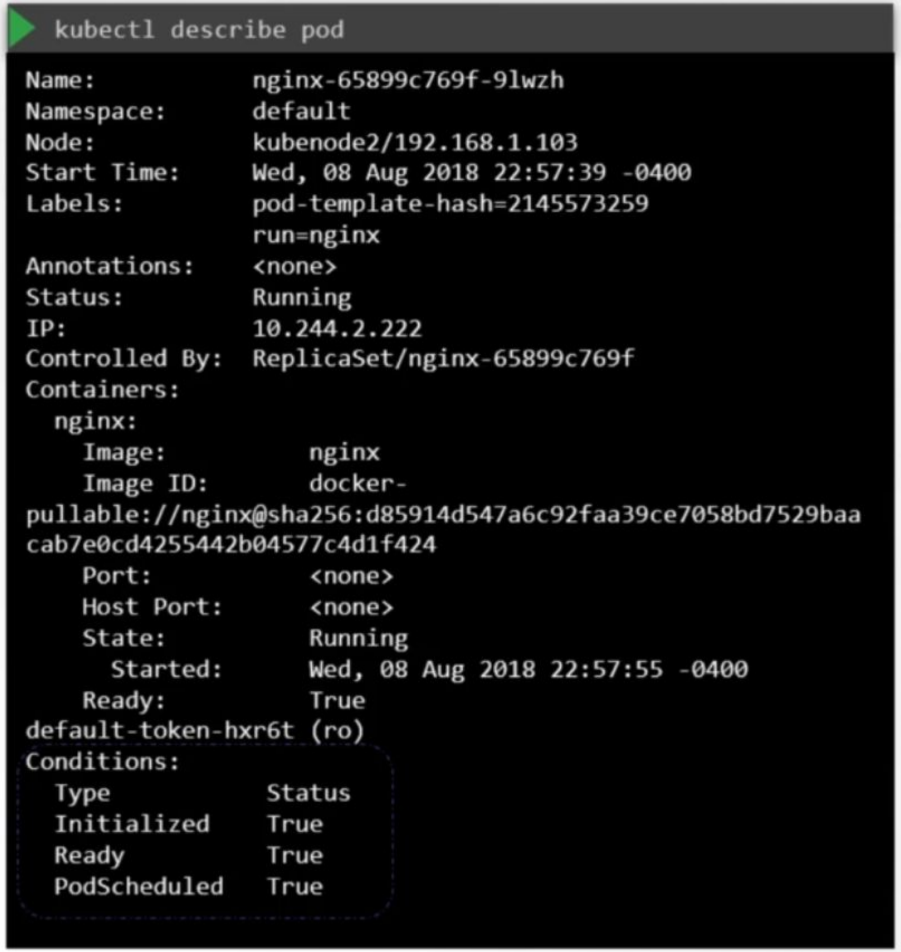
POD Status



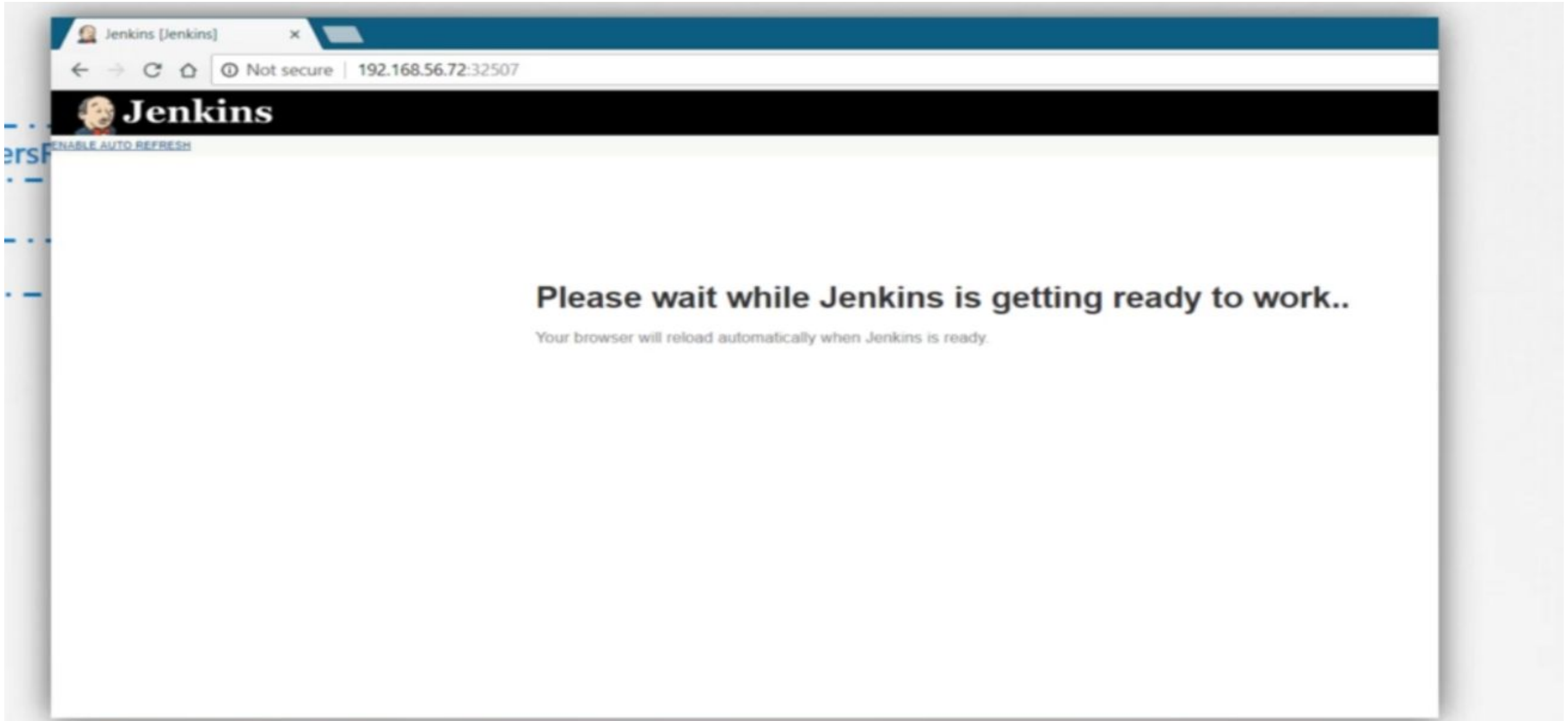
```
osboxes@kubemaster:~$ kubectl get pods
NAME                        READY     STATUS     RESTARTS   AGE
jenkins-566f687bf-c7nzf     1/1       Running    0          12m
nginx-65899c769f-9lwzh      1/1       Running    0          6h
redis-b48685f8b-fbnmx       1/1       Running    0          6h
```

POD Conditions





```
▶ kubectl describe pod
Name:           nginx-65899c769f-9lwzh
Namespace:      default
Node:           kubenode2/192.168.1.103
Start Time:     Wed, 08 Aug 2018 22:57:39 -0400
Labels:         pod-template-hash=2145573259
                run=nginx
Annotations:    <none>
Status:         Running
IP:             10.244.2.222
Controlled By:  ReplicaSet/nginx-65899c769f
Containers:
  nginx:
    Image:          nginx
    Image ID:       docker-
pullable://nginx@sha256:d85914d547a6c92faa39ce7058bd7529baa
cab7e0cd4255442b04577c4d1f424
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 08 Aug 2018 22:57:55 -0400
    Ready:          True
default-token-hxr6t (ro)
Conditions:
  Type           Status
  Initialized    True
  Ready          True
  PodScheduled   True
```

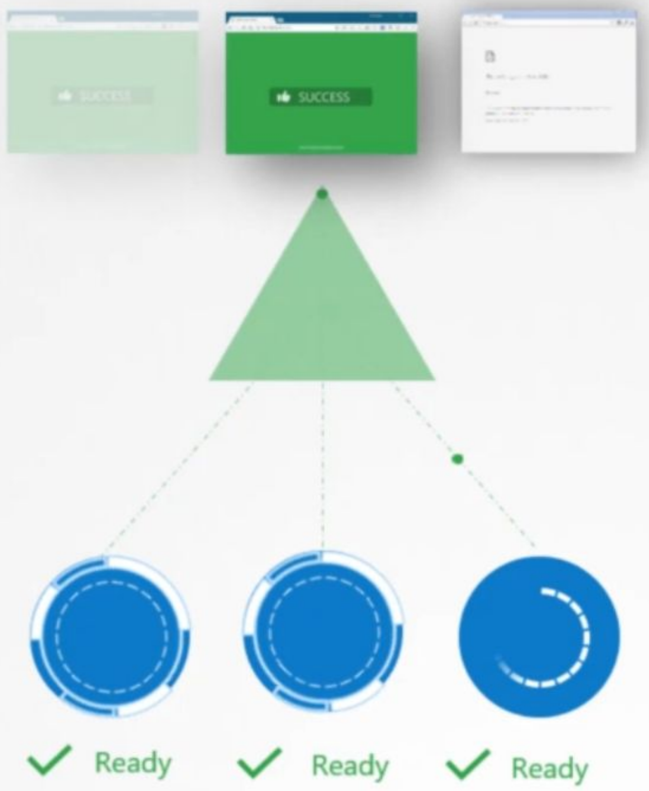


Please wait while Jenkins is getting ready to work..

Your browser will reload automatically when Jenkins is ready.

```
osboxes@kubemaster:~$ kubectl get all
NAME                      READY     STATUS     RESTARTS   AGE
pod/jenkins               1/1       Running    0          11s
```

POD Conditions



Readiness Probes

HTTP Test - /api/ready

TCP Test - 3306

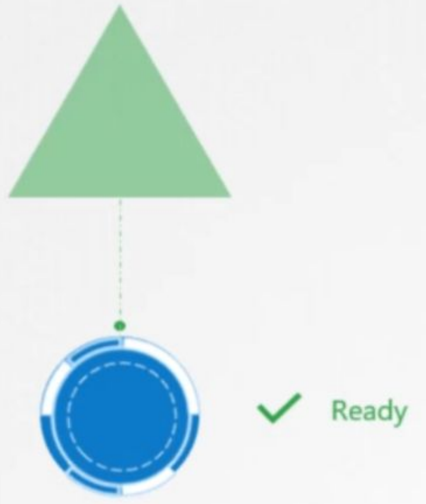
Exec Command

Common Probe Parameters

Each type of probe has common configurable fields:

- **initialDelaySeconds**: Seconds after the container started and before probes start. (default: 0)
- **periodSeconds**: Frequency of the pod. (default: 10)
- **timeoutSeconds**: Timeout for the expected response. (default: 1)
- **successThreshold**: How many success results received to transition from failure to a healthy state. (default: 1)
- **failureThreshold**: How many failed results received to transition from healthy to failure state. (default: 3)

Readiness Probe



HTTP Test - /api/ready

pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp
  labels:
    name: simple-webapp
spec:
  containers:
  - name: simple-webapp
    image: simple-webapp
    ports:
      - containerPort: 8080

    readinessProbe:
      httpGet:
        path: /api/ready
        port: 8080
```

Readiness Probe

```yaml
readinessProbe:
  httpGet:
    path: /api/ready
    port: 8080

  initialDelaySeconds: 10

  periodSeconds: 5

  failureThreshold: 8
```

```yaml
readinessProbe:
  tcpSocket:
    port: 3306
```

```yaml
readinessProbe:
  exec:
    command:
      - cat
      - /app/is_ready
```

HTTP Test - /api/ready

TCP Test - 3306

Exec Command

StartUp Probes

```yaml
ports:
- name: liveness-port
  containerPort: 8080
  hostPort: 8080

livenessProbe:
  httpGet:
    path: /healthz
    port: liveness-port
  failureThreshold: 1
  periodSeconds: 10

startupProbe:
  httpGet:
    path: /healthz
    port: liveness-port
  failureThreshold: 30
  periodSeconds: 10
```

Liveness Probes

This site can't be reached

Liveness Probes

HTTP Test - /api/healthy

TCP Test - 3306

Exec Command

Liveness Probe

HTTP Test - /api/ready

pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp
  labels:
    name: simple-webapp
spec:
  containers:
  - name: simple-webapp
    image: simple-webapp
    ports:
      - containerPort: 8080

    livenessProbe:
      httpGet:
        path: /api/healthy
        port: 8080
```

Liveness Probe

```
readinessProbe:
  httpGet:
    path: /api/ready
    port: 8080

  initialDelaySeconds: 10

  periodSeconds: 5

  failureThreshold: 8
```

```
readinessProbe:
  tcpSocket:
    port: 3306
```

```
readinessProbe:
  exec:
    command:
      - cat
      - /app/is_ready
```

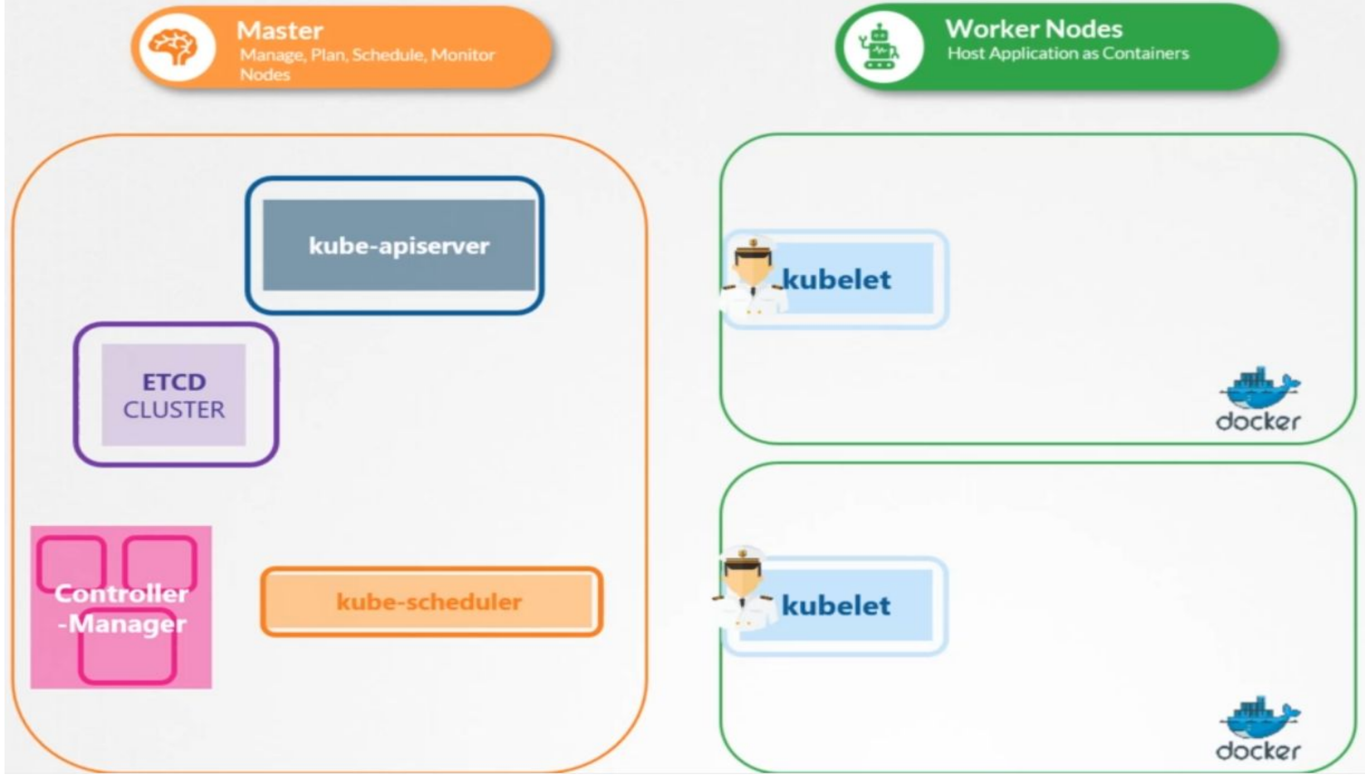
HTTP Test - /api/ready

TCP Test - 3306

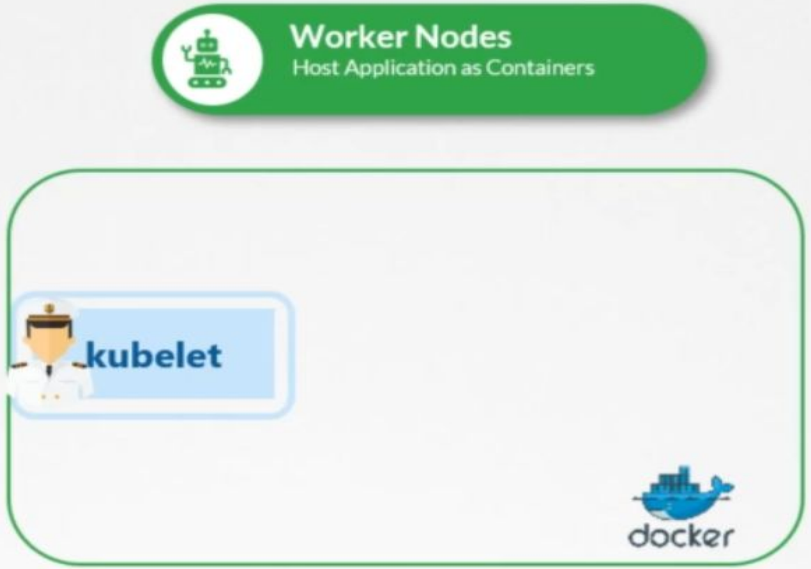
Exec Command

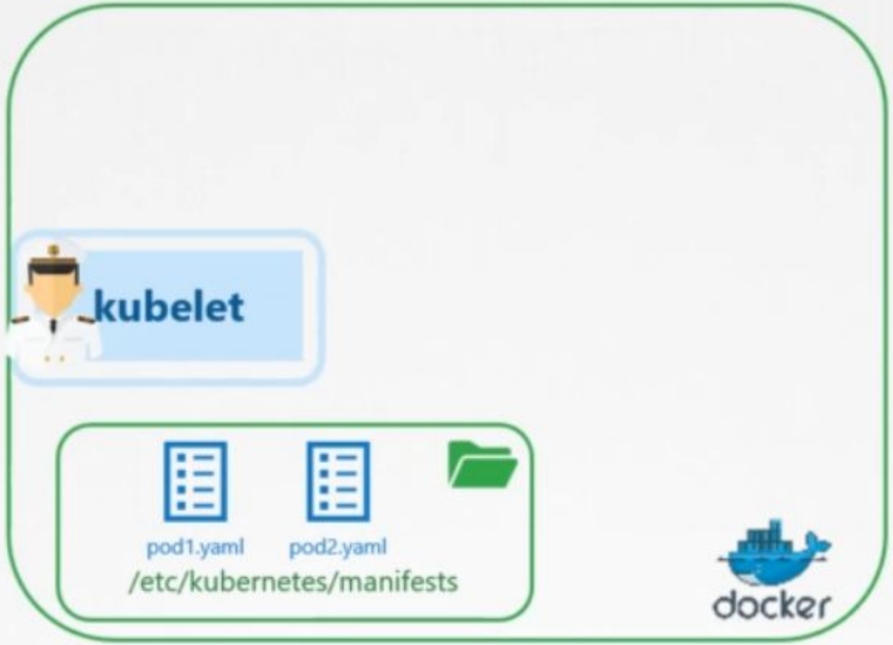
Static PODs

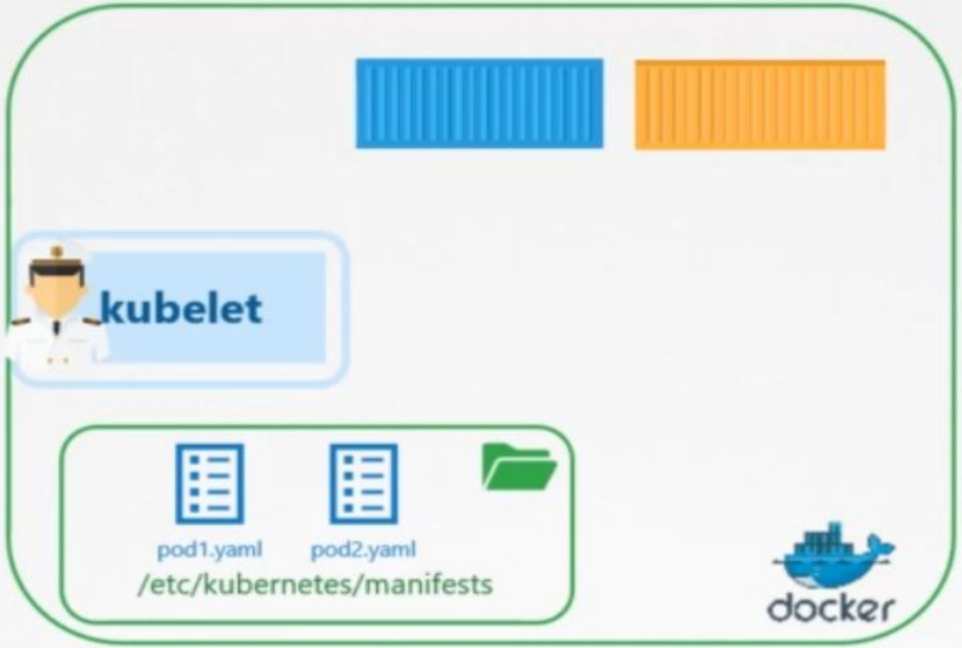
Kubernetes Architecture



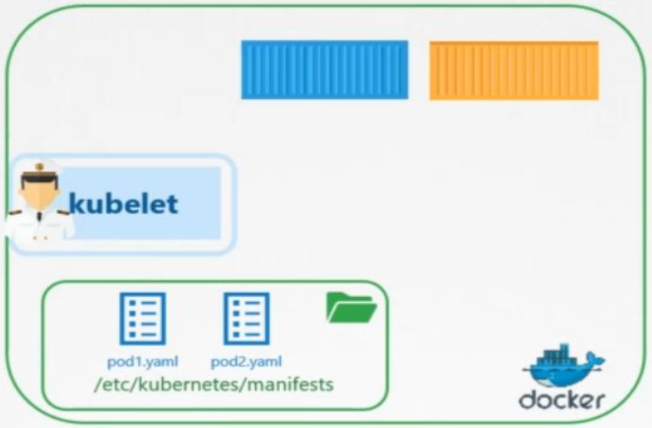
Kubernetes Architecture




kubelet
pod1.yaml
pod2.yaml
/etc/kubernetes/manifests
docker


kubelet
pod1.yaml
pod2.yaml
/etc/kubernetes/manifests
docker

Static PODs



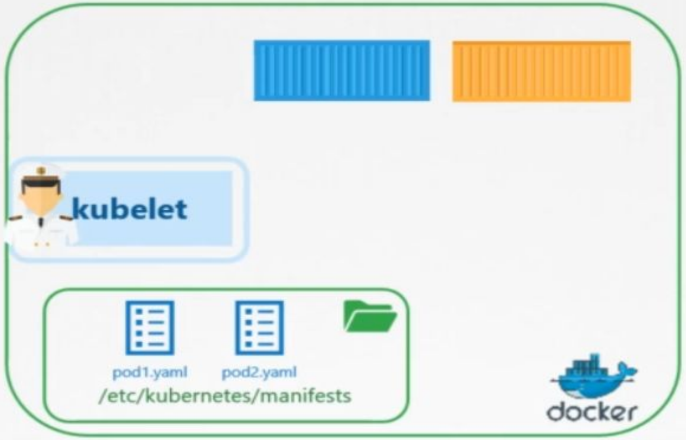
kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\
  --container-runtime=remote \\
  --container-runtime-endpoint=unix:///var/run/containerd/containerd.sock \\
  --config=kubeconfig.yaml \\
  --kubeconfig=/var/lib/kubelet/kubeconfig \\
  --network-plugin=cni \\
  --register-node=true \\
  --v=2
```

kubeconfig.yaml

```
staticPodPath: /etc/kubernetes/manifest
```

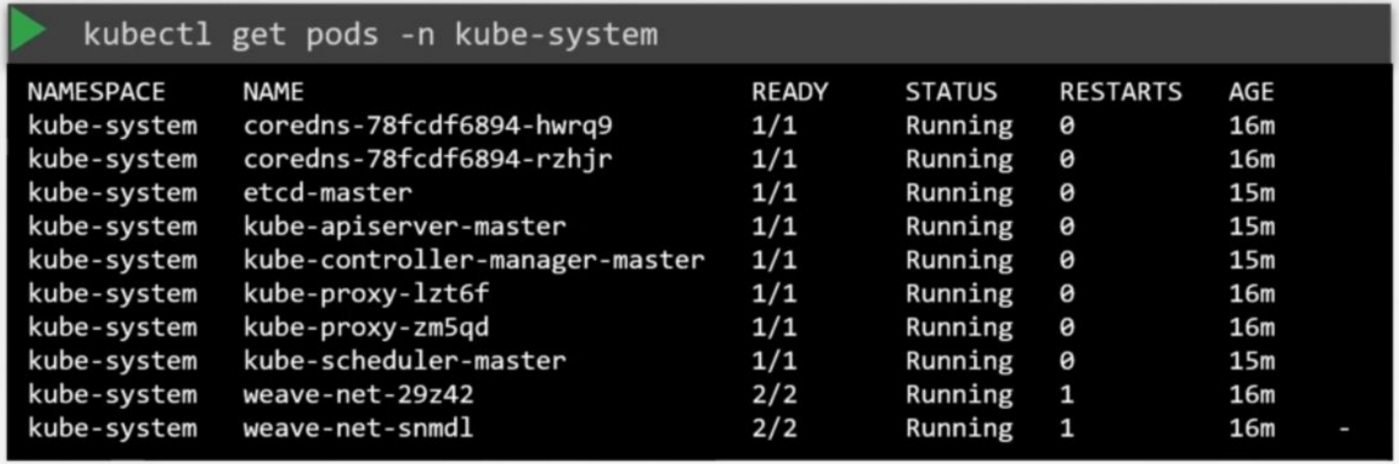
Static PODs



```
docker ps
```

```
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS
PORTS               NAMES
8e5d4c4db7b6        busybox                  "sh -c 'echo Hello K…"   20 seconds ago      Up 20 seconds
k8s_myapp-container_myapp-pod-host01_default_48e37fb432f2e06350e76786bd0bac66_0
f6737e1149cb        k8s.gcr.io/pause:3.1     "/pause"                 24 seconds ago      Up 23 seconds
k8s_POD_myapp-pod-host01_default_48e37fb432f2e06350e76786bd0bac66_0
```

Use Case

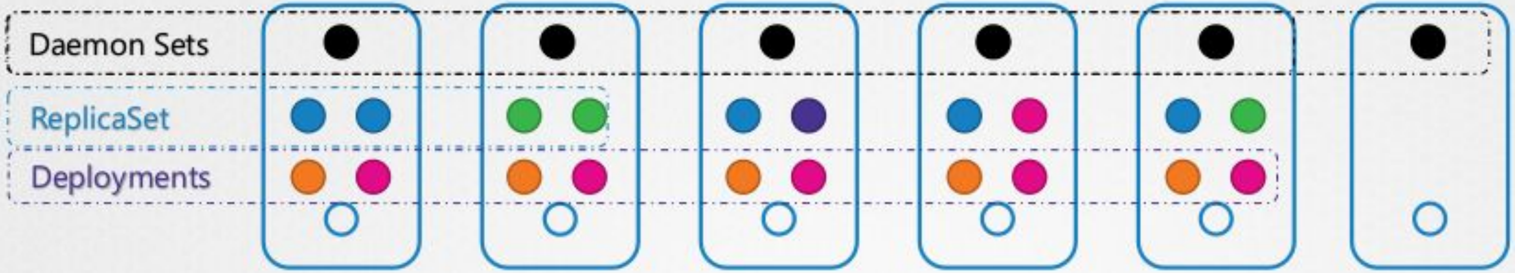


```
kubectl get pods -n kube-system
```

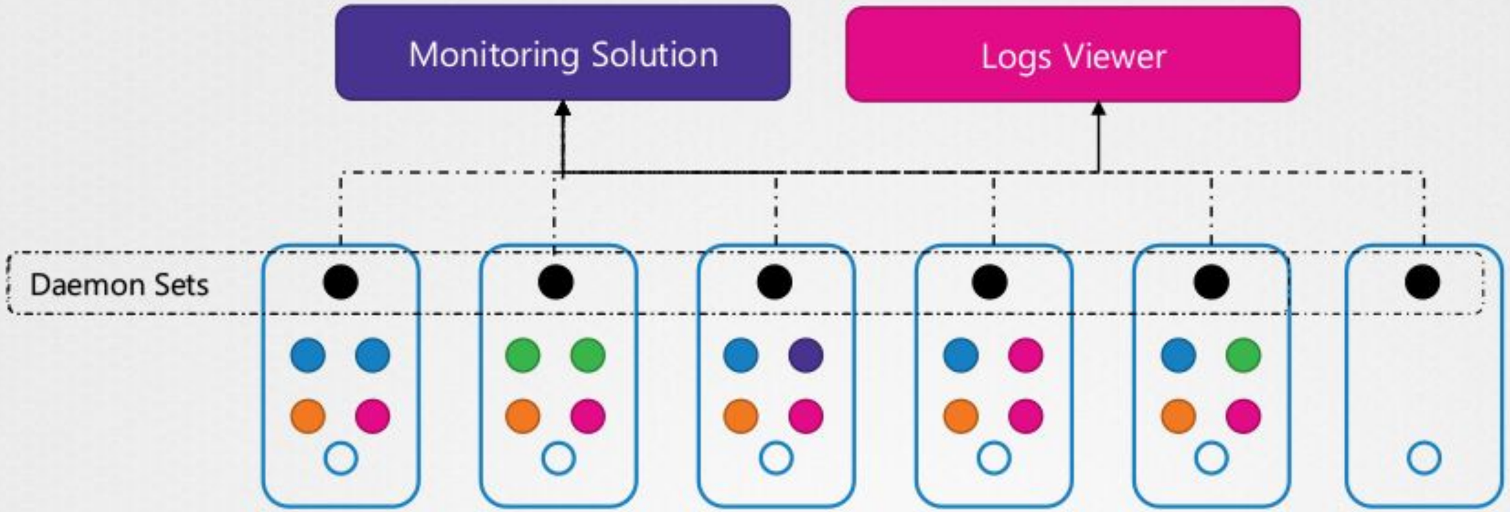
```
NAMESPACE     NAME                              READY    STATUS    RESTARTS   AGE
kube-system   coredns-78fcdf6894-hwrq9          1/1      Running   0          16m
kube-system   coredns-78fcdf6894-rzhjr          1/1      Running   0          16m
kube-system   etcd-master                       1/1      Running   0          15m
kube-system   kube-apiserver-master             1/1      Running   0          15m
kube-system   kube-controller-manager-master    1/1      Running   0          15m
kube-system   kube-proxy-lzt6f                  1/1      Running   0          16m
kube-system   kube-proxy-zm5qd                  1/1      Running   0          16m
kube-system   kube-scheduler-master             1/1      Running   0          15m
kube-system   weave-net-29z42                   2/2      Running   1          16m
kube-system   weave-net-snmdl                   2/2      Running   1          16m    -
```

Daemon Sets

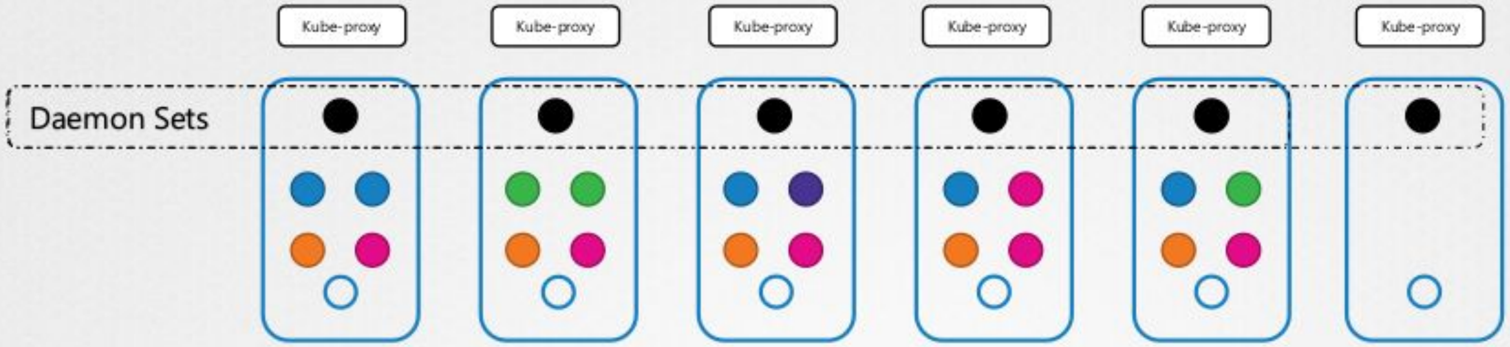
Daemon Sets



Daemon Sets – UseCase



Daemon Sets – UseCase – kube-proxy



DaemonSet Definition

daemon-set-definition.yaml

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
    spec:
      containers:
      - name: monitoring-agent
        image: monitoring-agent
```

replicaset-definition.yaml

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
    spec:
      containers:
      - name: monitoring-agent
        image: monitoring-agent
```

```
kubectl create -f daemon-set-definition.yaml
```

View DaemonSets

```
kubectl get daemonsets
```

```
      NAME         DESIRED    CURRENT    READY     UP-TO-DATE    AVAILABLE   AGE
monitoring-daemon  1          1          1         1             1           41
```

```
kubectl describe daemonsets monitoring-daemon
```

```
Name:            monitoring-daemon
Selector:        name=monitoring-daemon
Node-Selector:   <none>
Labels:          name=monitoring-daemon
Desired Number of Nodes Scheduled: 2
Current Number of Nodes Scheduled: 2
Number of Nodes Scheduled with Up-to-date Pods: 2
Number of Nodes Scheduled with Available Pods: 1
Number of Nodes Misscheduled: 0
Pods Status:  2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:          app=monitoring-agent
  Containers:
```

An **initContainer** is configured in a pod like all other containers, except that it is specified inside a `initContainers` section, like this:

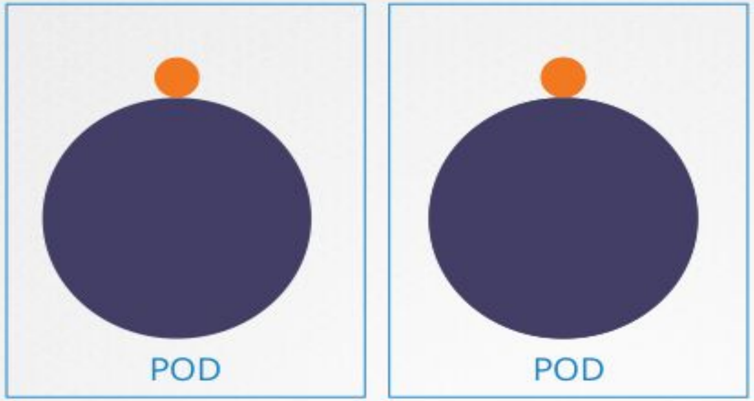
```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox:1.28
    command: ['sh', '-c', 'echo The app is running! && sleep
  initContainers:
  - name: init-myservice
    image: busybox
    command: ['sh', '-c', 'git clone <some-repository-that-wi
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox:1.28
    command: ['sh', '-c', 'echo The app is running! && sleep 3600']
  initContainers:
  - name: init-myservice
    image: busybox:1.28
    command: ['sh', '-c', "echo waiting for myservice; sleep 2;"]
  - name: init-mydb
    image: busybox:1.28
    command: ['sh', '-c', "echo waiting for mydb; sleep 2;"]
```

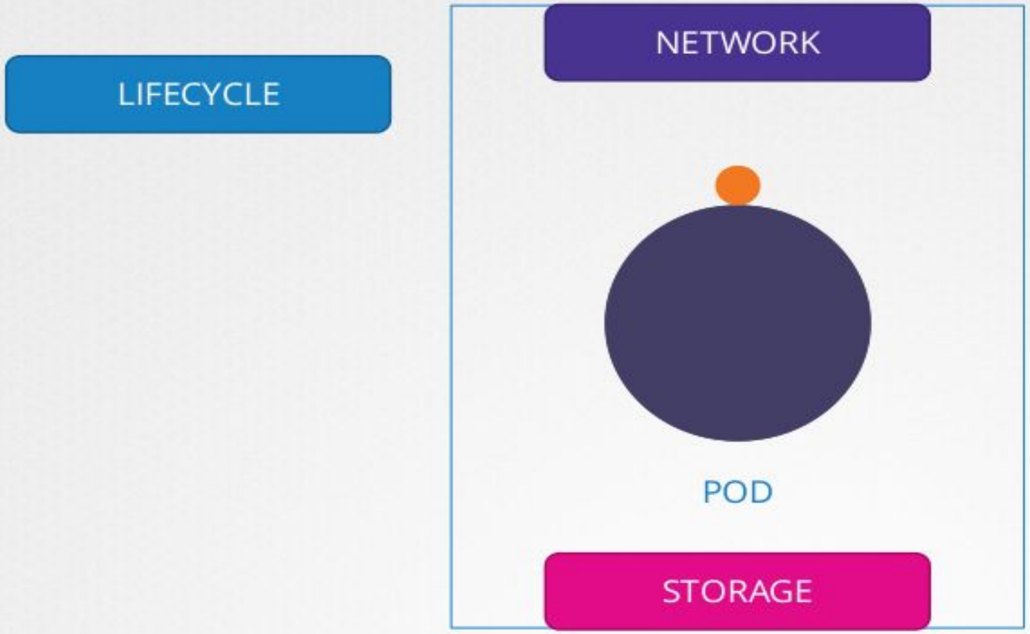
```
kubectl logs myapp-pod -c init-myservice
```

Kubernetes Multi-Container PODs

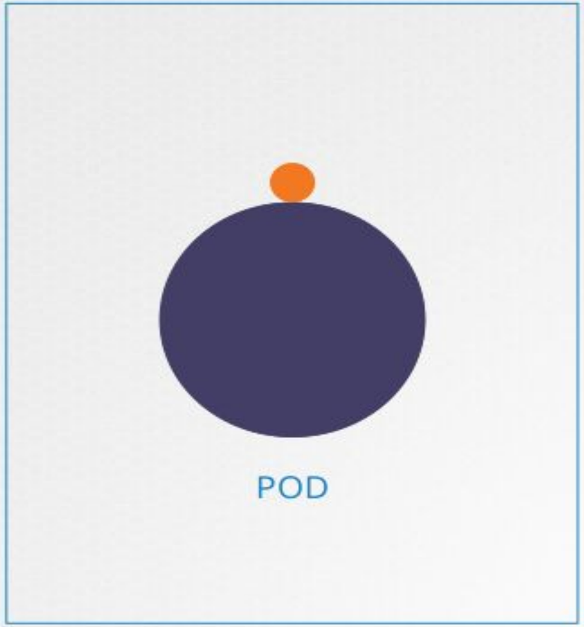
Multi-Container PODs



Multi-Container PODs



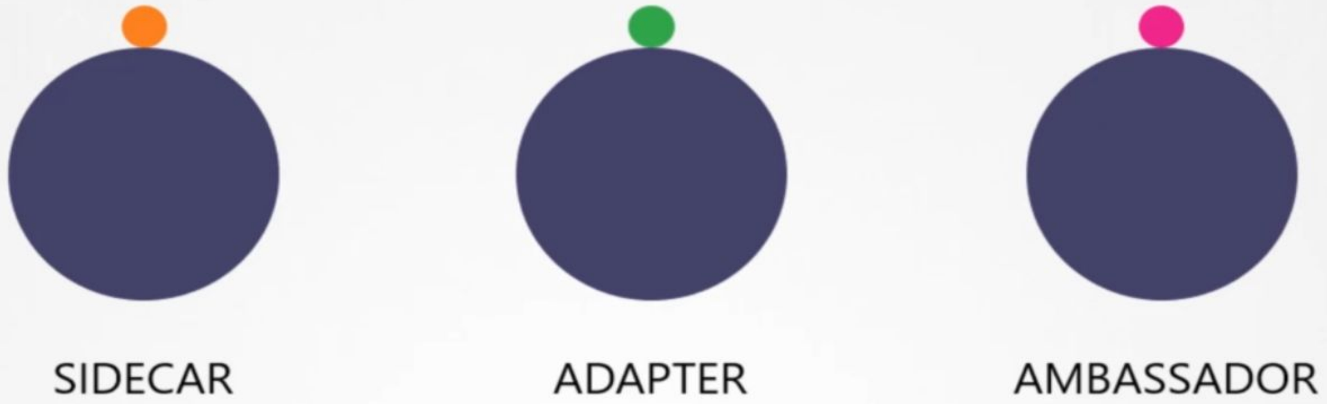
Create



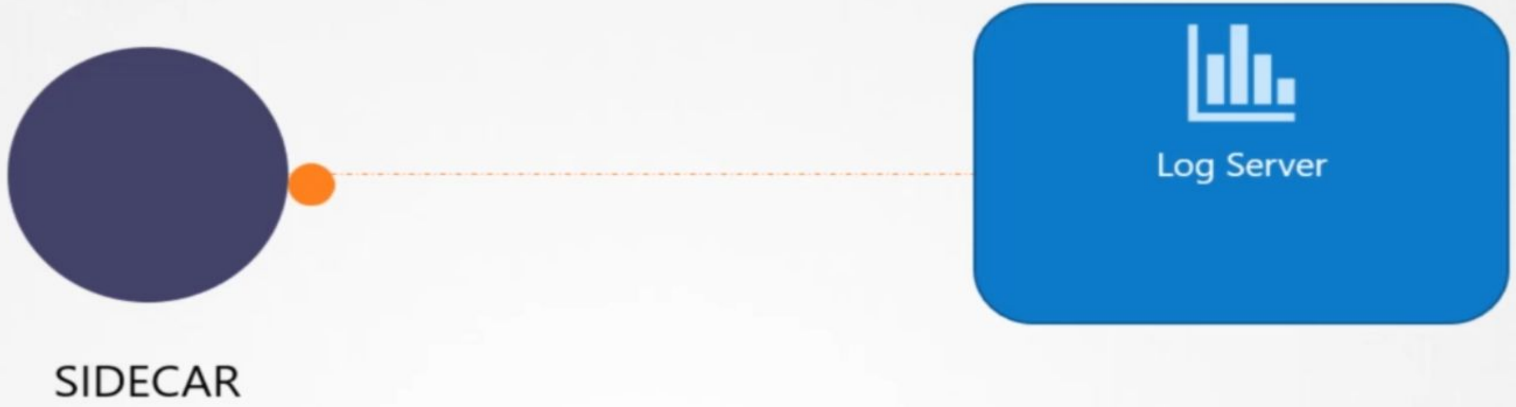
pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp
  labels:
    name: simple-webapp
spec:
  containers:
  - name: simple-webapp
    image: simple-webapp
    ports:
      - containerPort: 8080

  - name: log-agent
    image: log-agent
```

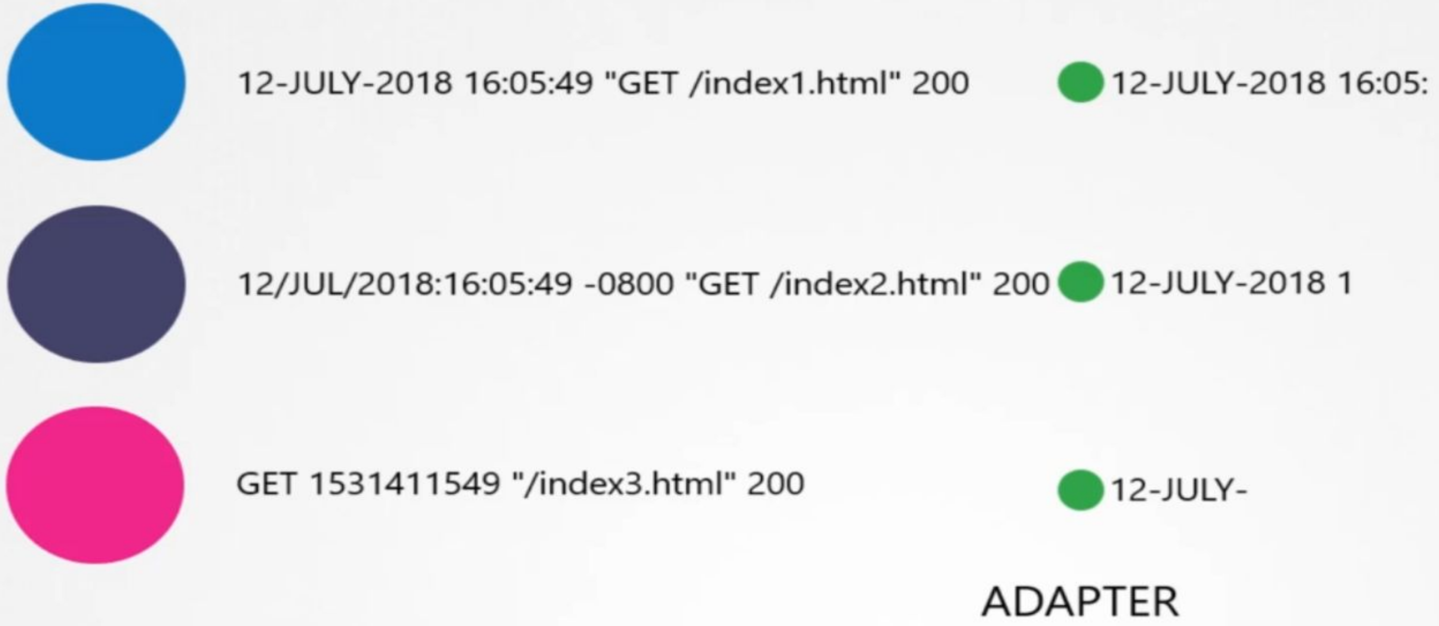
Design Patterns



Design Patterns - Sidecar



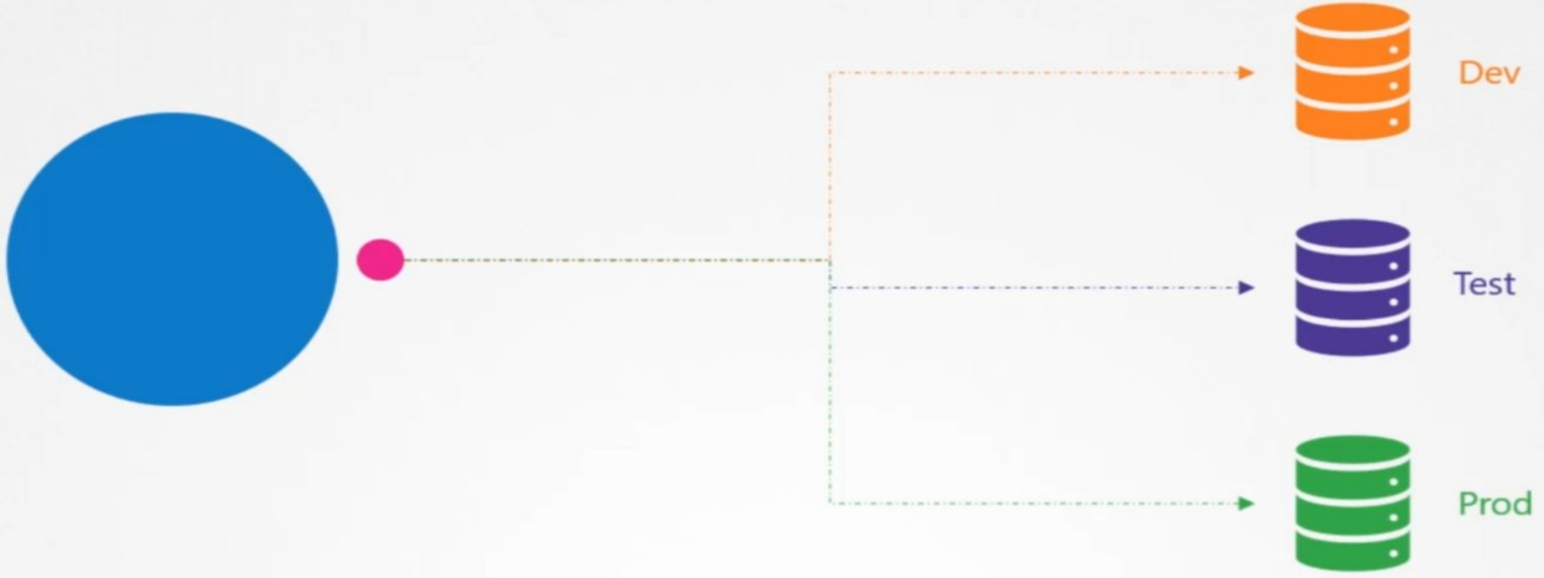
Design Patterns - Adapter



```
12-JULY-2018 16:05:49 "GET /index1.html" 200
12/JUL/2018:16:05:49 -0800 "GET /index2.html" 200
GET 1531411549 "/index3.html" 200
```

```
12-JULY-2018 16:05:
12-JULY-2018 1
12-JULY-
```

Design Patterns - Ambassador



References:

- https://www.udemy.com/course/certified-kubernetes-administrator-with-practice-tests
- https://www.udemy.com/course/certified-kubernetes-application-developer
- https://kubernetes.io/docs