TEL4VN

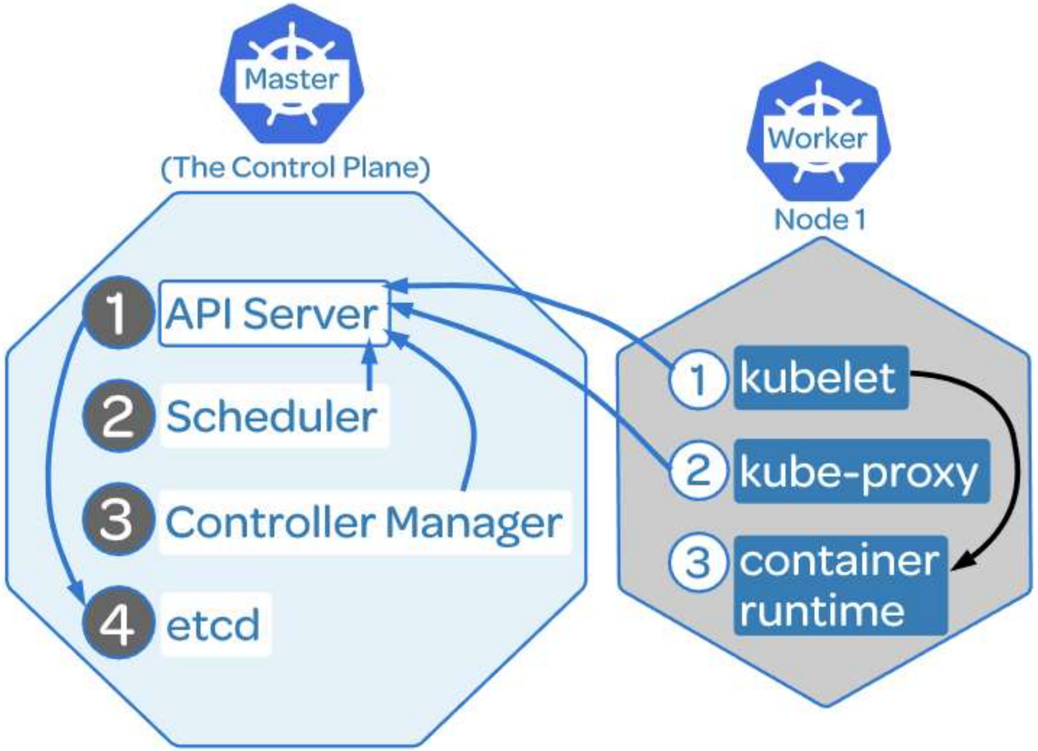
K8S Architecture

THAO LUONG
03/2022

Content

- ❑ Kubernetes Cluster Architecture
- ❑ Kubernetes Core Service
- ❑ Kubernetes API Primitives
- ❑ Kubernetes Services and Network Primitives

Cluster Architecture



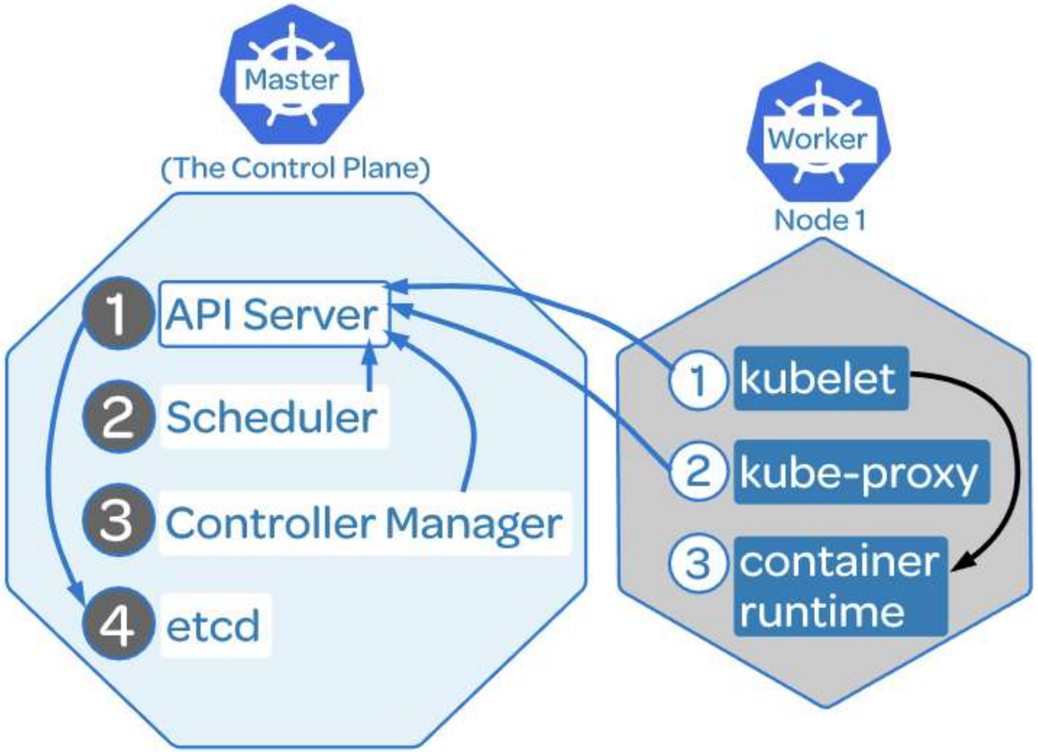
- **API Server**: The communication hub for all cluster components. It exposes the k8s API

- **Scheduler**: Assigns workloads to a worker node based on resource requirements, hardware constraints, …

- **Controller Manager**: Maintains the cluster, handles node failures, replicating components, maintaining the amount of pods…

- **etcd**: Data store that store the cluster configuration

Cluster Architecture

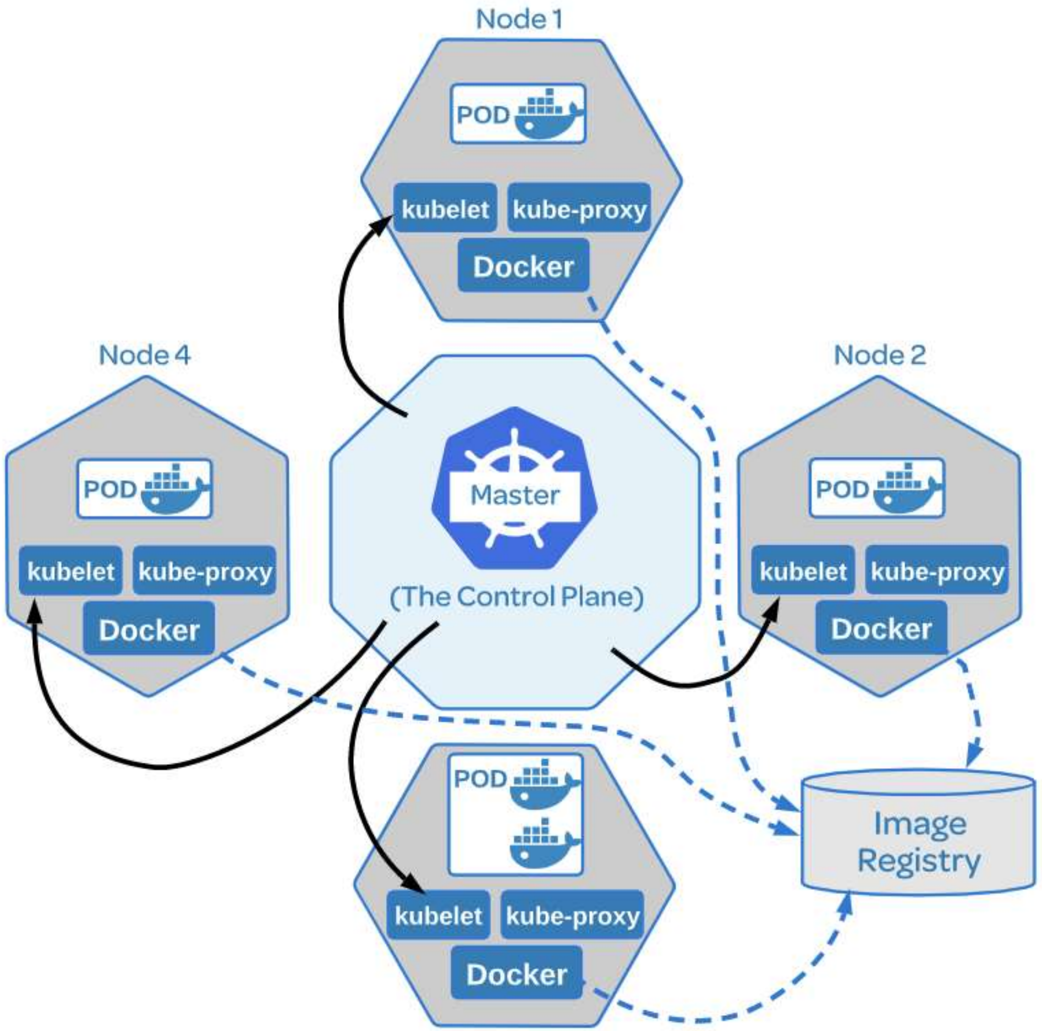


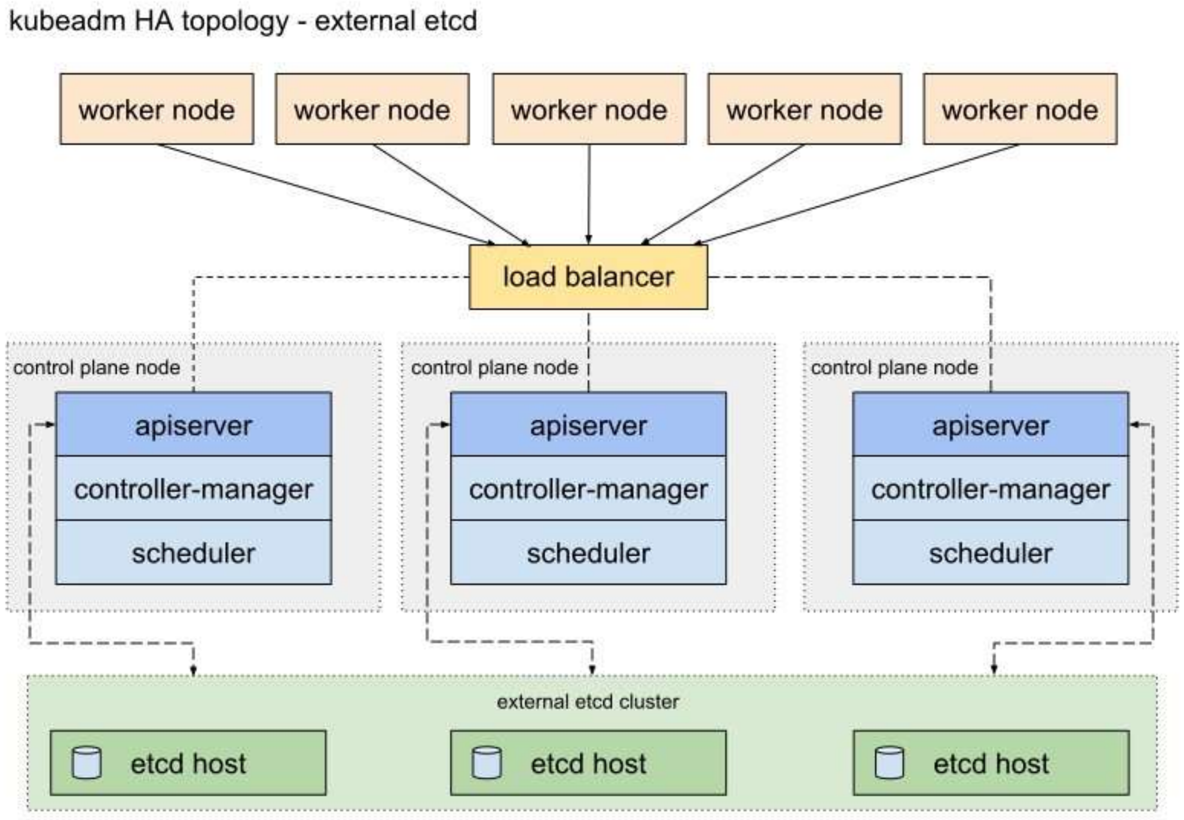
- **kubelet**: Runs and manages the containers on the node and talks to the API server

- **kube-proxy**: Load balancing traffic between application

- **container runtime**: The program that runs your containers

Multiple workers - multiple masters

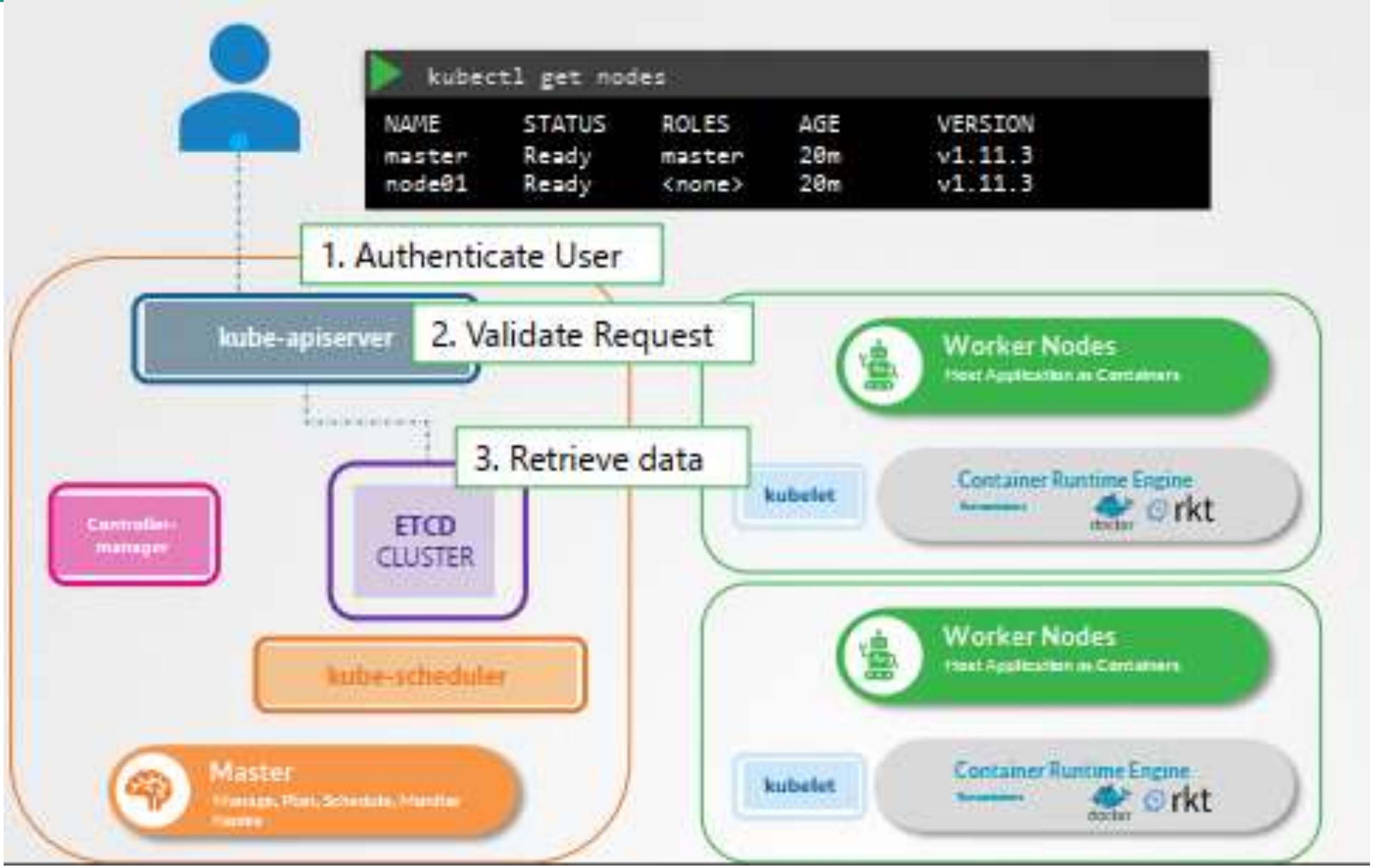




Core Concept

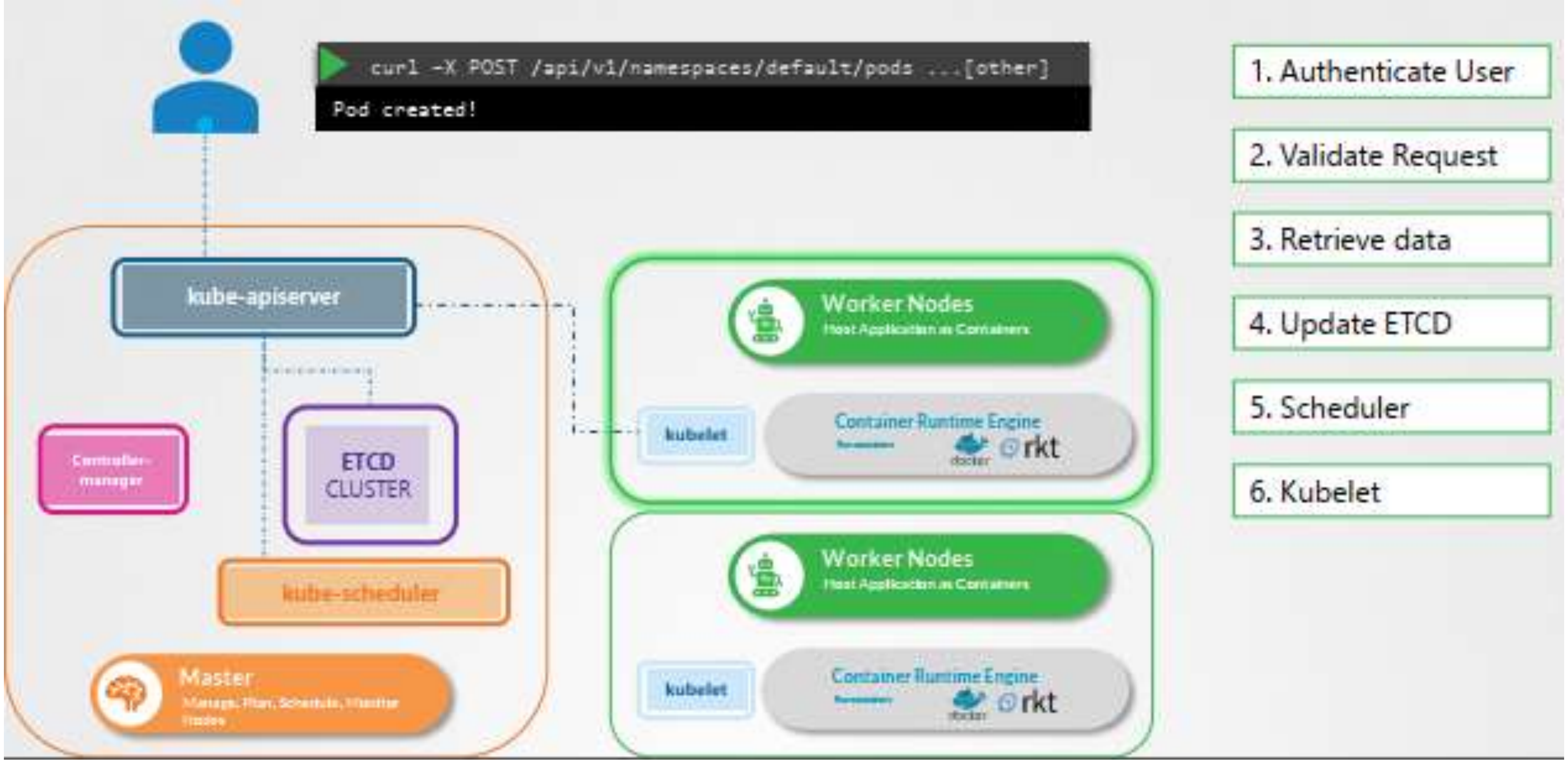
- ❑ Kube API Server
- ❑ ETCD
- ❑ Controller manager
- ❑ Scheduler
- ❑ Kubelet
- ❑ Kube Proxy

Cluster Architecture



```
kubectl get nodes
NAME      STATUS   ROLES    AGE   VERSION
master    Ready    master   20m   v1.11.3
node01    Ready    <none>   20m   v1.11.3
```

Cluster Architecture

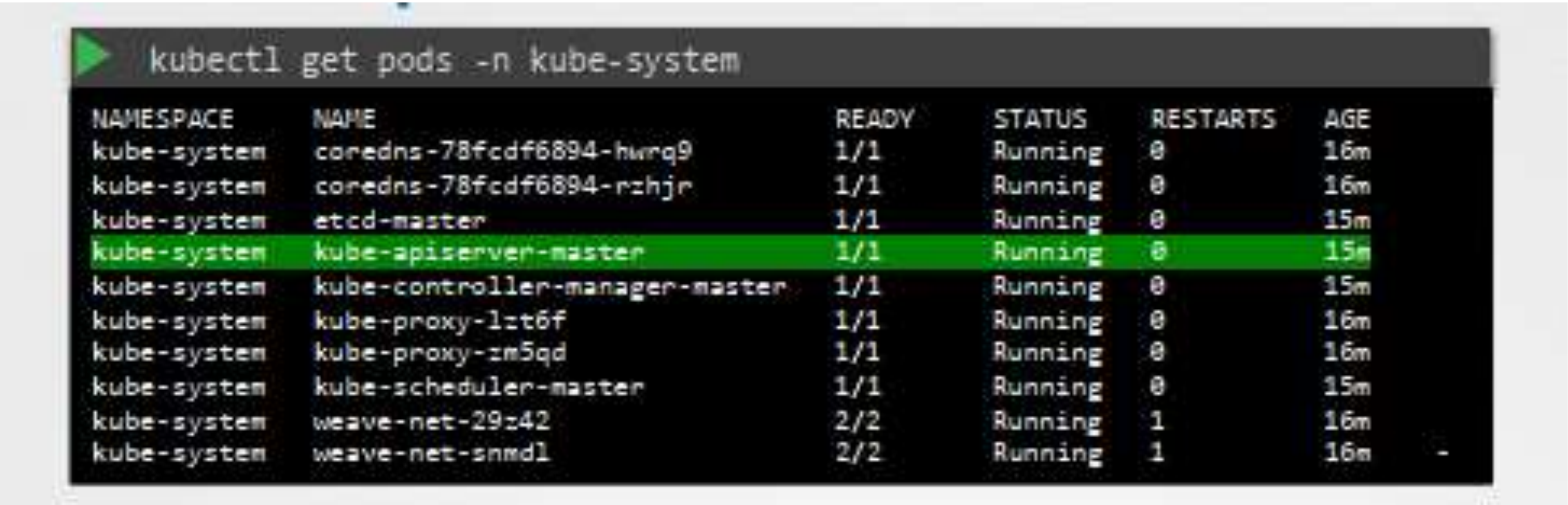


```
curl -X POST /api/v1/namespaces/default/pods ...[other]
Pod created!
```

1. Authenticate User
2. Validate Request
3. Retrieve data
4. Update ETCD
5. Scheduler
6. Kubelet

Kube API Server

“The Kubernetes API server validates and configures data for the api objects which include pods, services, replication controllers, and others. The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact”

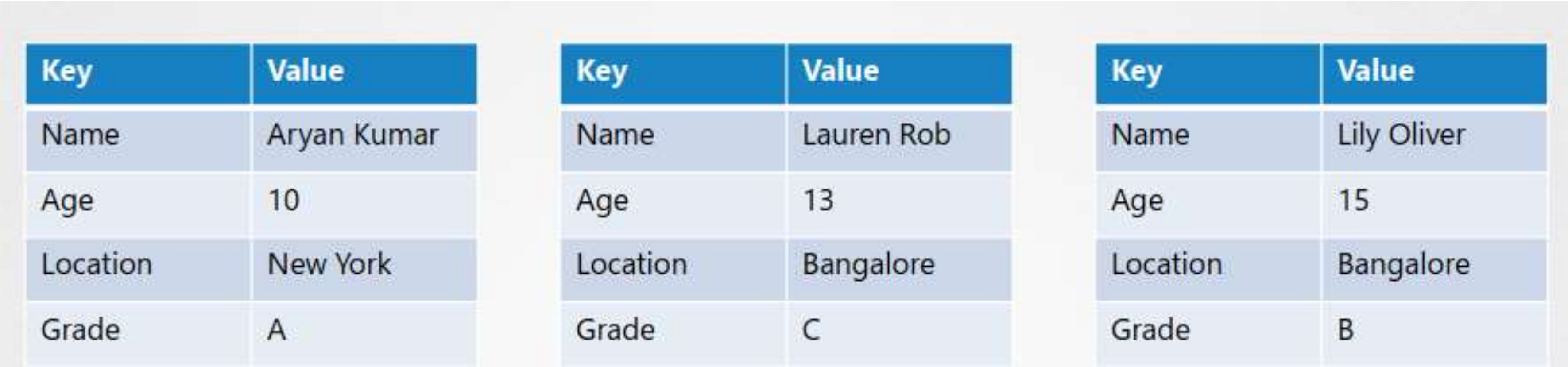


```
kubectl get pods -n kube-system

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
```

ETCD

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data



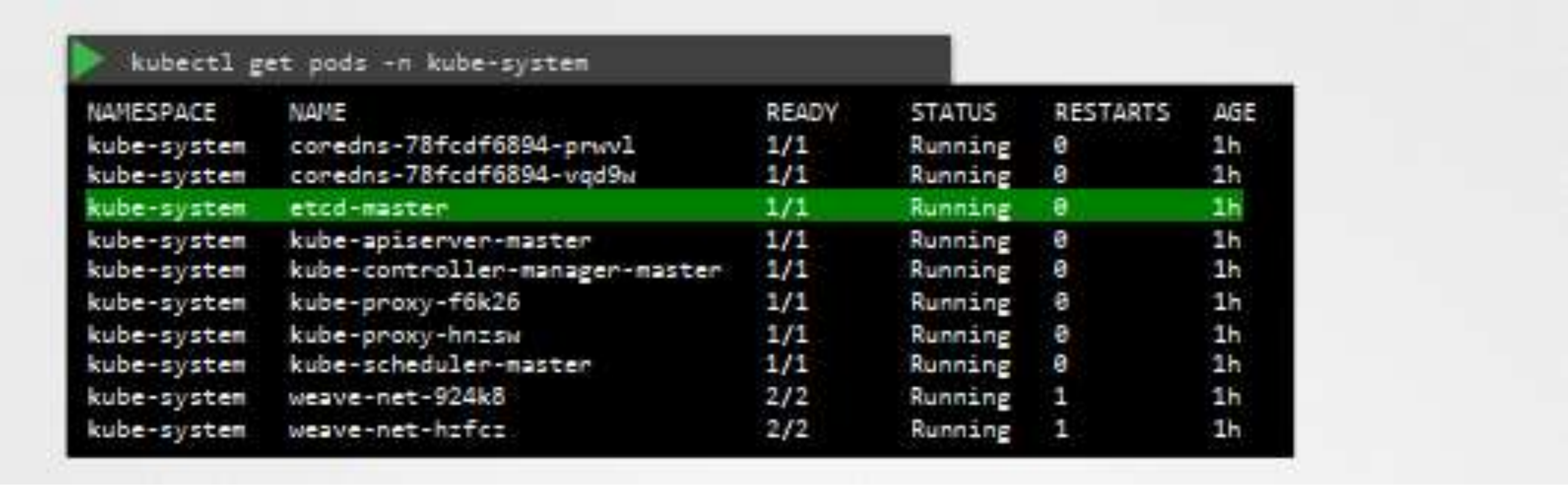
Key	Value
Name	Aryan Kumar
Age	10
Location	New York
Grade	A

Key	Value
Name	Lauren Rob
Age	13
Location	Bangalore
Grade	C

Key	Value
Name	Lily Oliver
Age	15
Location	Bangalore
Grade	B

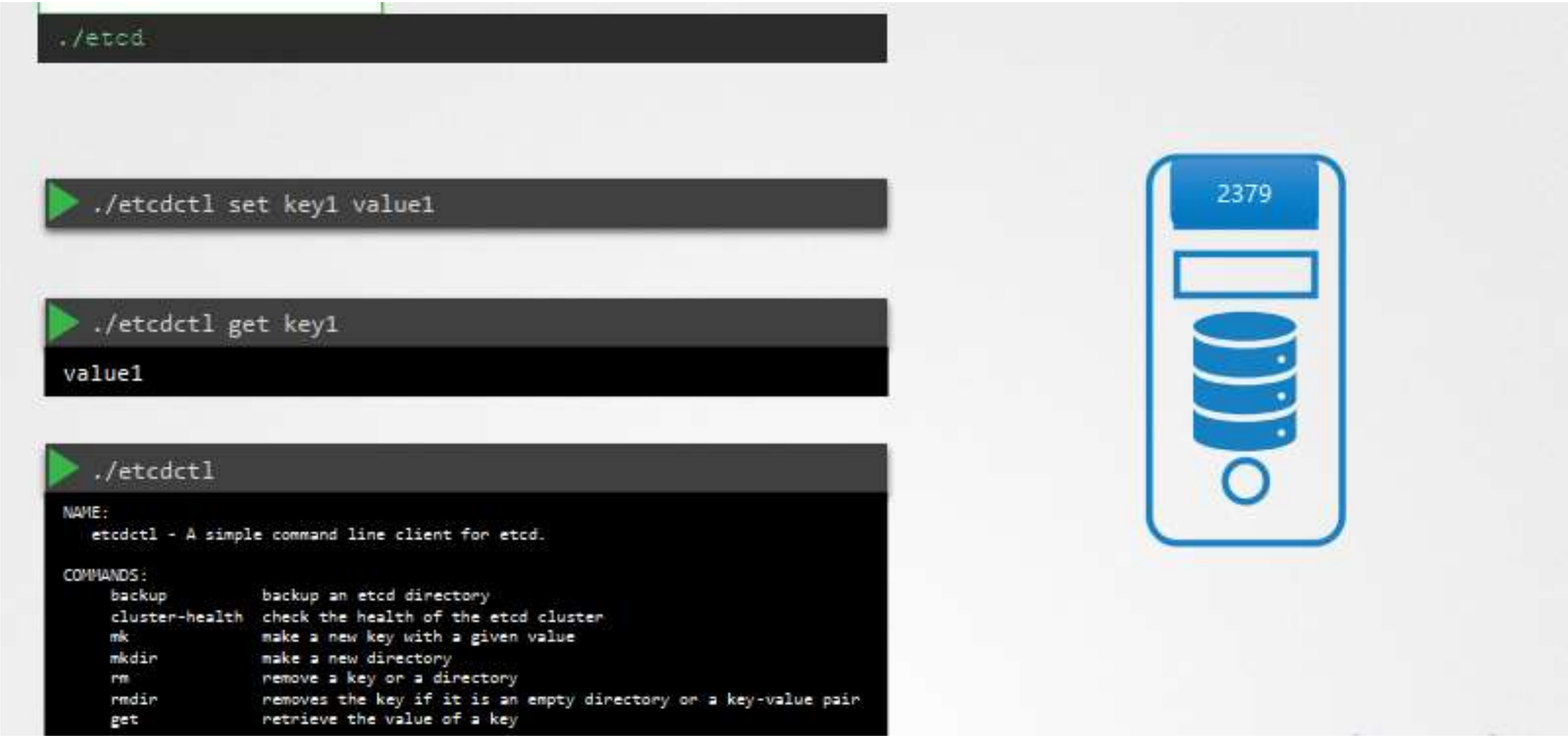
ETCD

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data



```
kubectl get pods -n kube-system
NAMESPACE     NAME                               READY     STATUS    RESTARTS   AGE
kube-system   coredns-78fcdf6894-prwvl           1/1       Running   0          1h
kube-system   coredns-78fcdf6894-vqd9w           1/1       Running   0          1h
kube-system   etcd-master                        1/1       Running   0          1h
kube-system   kube-apiserver-master              1/1       Running   0          1h
kube-system   kube-controller-manager-master     1/1       Running   0          1h
kube-system   kube-proxy-f6k26                   1/1       Running   0          1h
kube-system   kube-proxy-hnzsw                   1/1       Running   0          1h
kube-system   kube-scheduler-master              1/1       Running   0          1h
kube-system   weave-net-924k8                    2/2       Running   1          1h
kube-system   weave-net-hzfcz                    2/2       Running   1          1h
```

ETCD



```
./etcd
```

```
./etcdctl set key1 value1
```

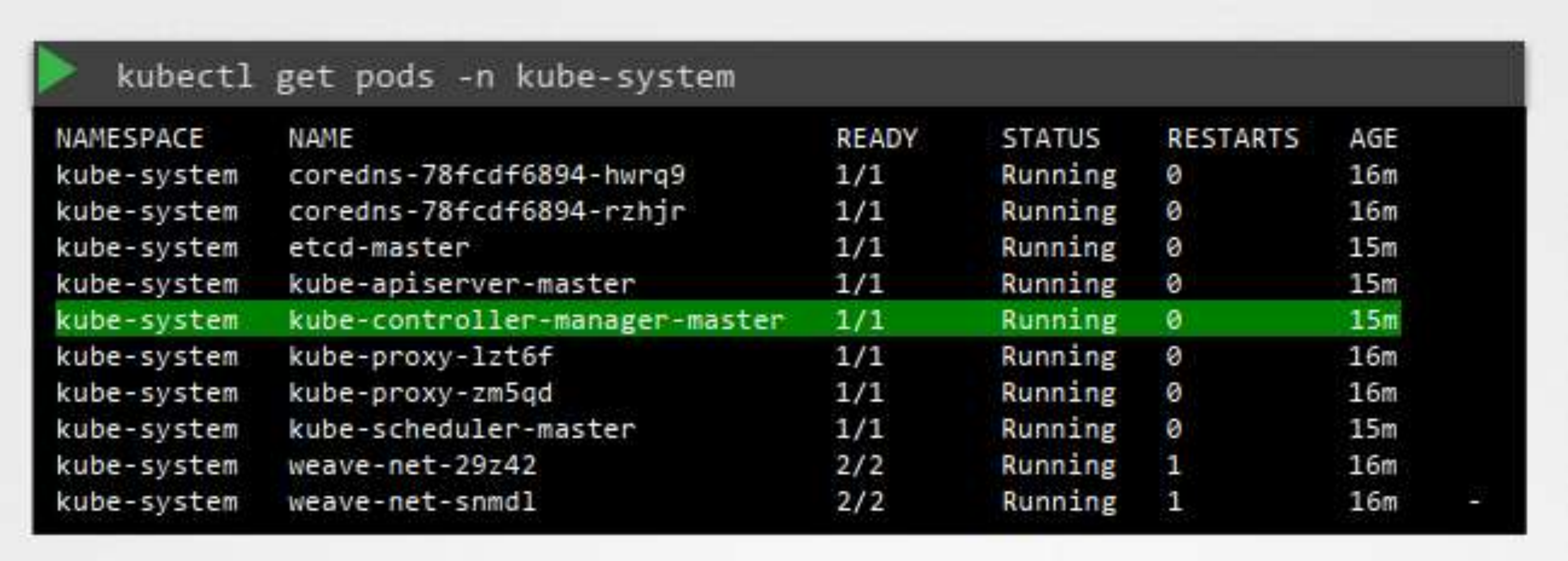
```
./etcdctl get key1
value1
```

```
./etcdctl
NAME:
   etcdctl - A simple command line client for etcd.

COMMANDS:
     backup          backup an etcd directory
     cluster-health  check the health of the etcd cluster
     mk              make a new key with a given value
     mkdir           make a new directory
     rm              remove a key or a directory
     rmdir           removes the key if it is an empty directory or a key-value pair
     get             retrieve the value of a key
```

Kube Controller Manager

a controller is a control loop that watches the shared state of the cluster through the apiserver and makes changes attempting to move the current state towards the desired state

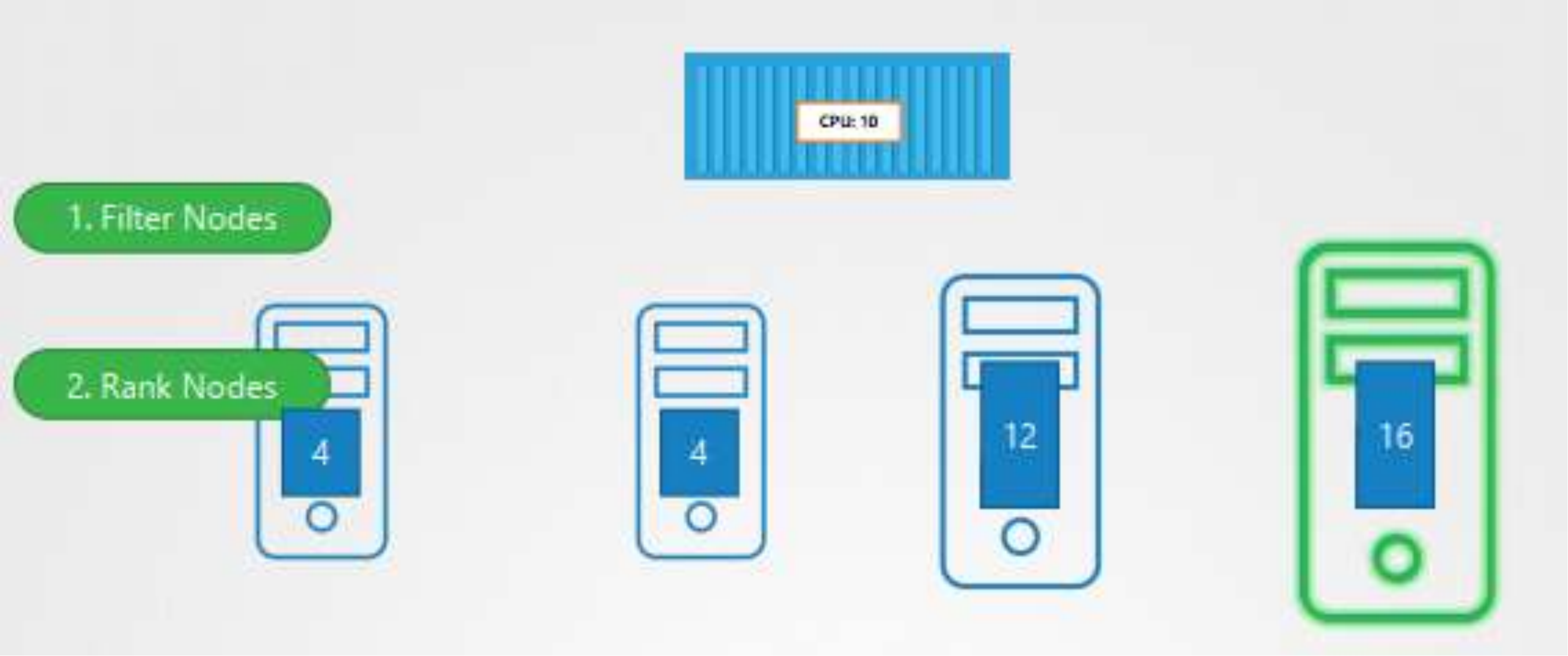


```
kubectl get pods -n kube-system

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
```

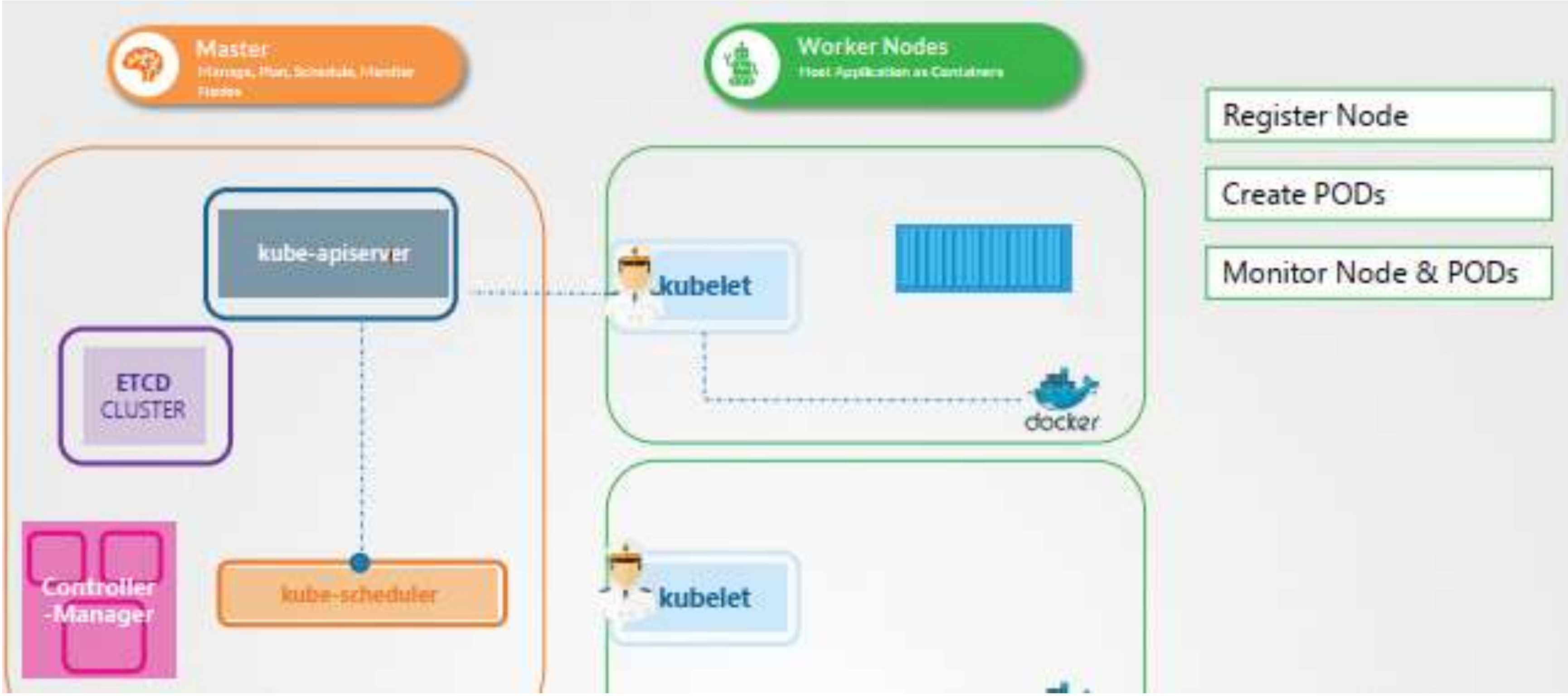
Kube Scheduler

The scheduler needs to take into account individual and collective resource requirements, quality of service requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, deadlines, and so on.



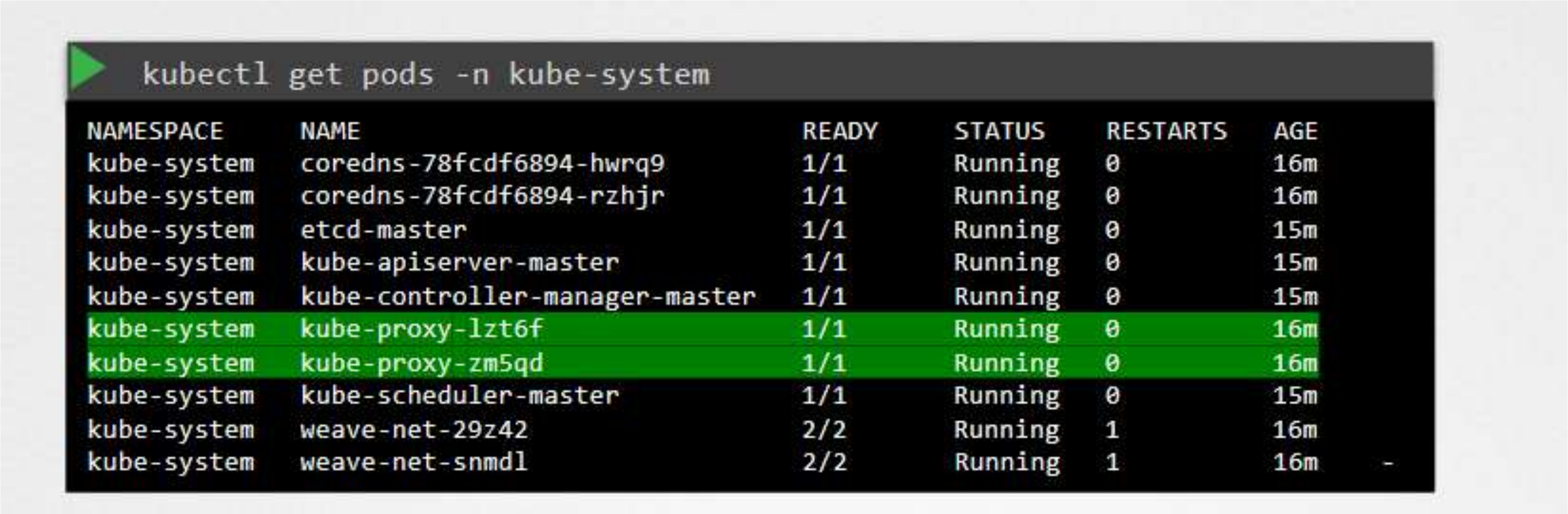
Kubelet

The kubelet is the primary "node agent" that runs on each node. It can register the node with the apiserver using one of: the hostname



Kube Proxy

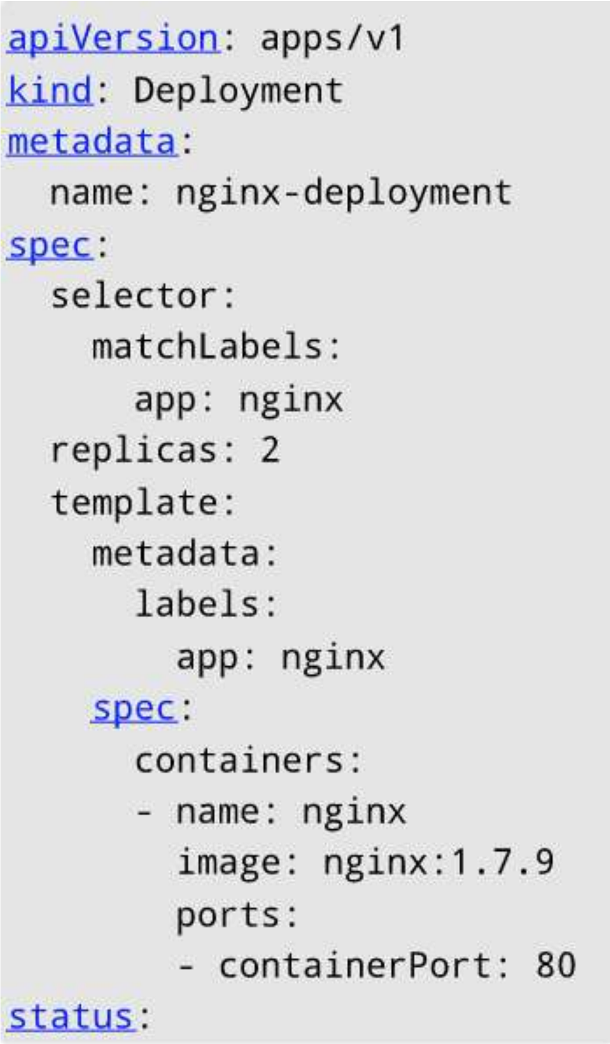
The Kubernetes network proxy runs on each node. This can do simple TCP, UDP, and SCTP stream forwarding or round robin TCP, UDP, and SCTP forwarding across a set of backends



```
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
```

API Primitives



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
status:
```

- **apiVersion: $GROUP_NAME/$VERSION**
  The API server exposes an HTTP API that lets end users, different parts of your cluster, and external components communicate with one another.

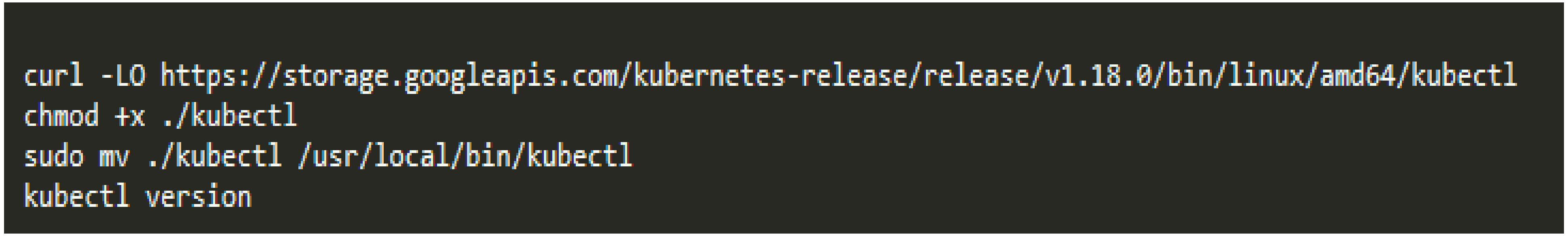
- **kind:**
  Represents the kind of object will be created such as pod, deployment, job,... This field is a required field.

- **metadata:**
  Data that helps uniquely identify the object, including a name, UID and optional namespace.

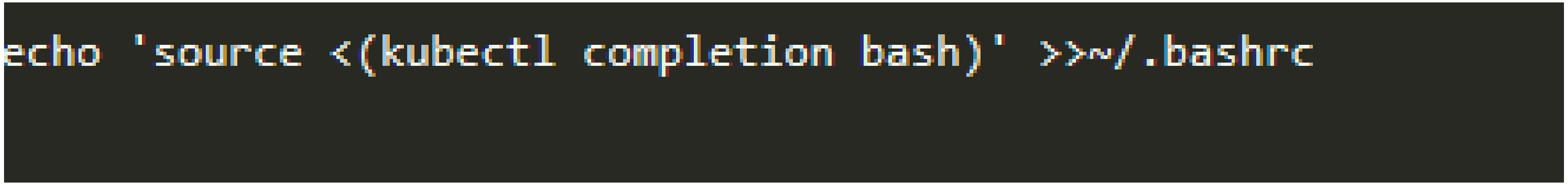
- **spec:**
  Describes the desired state and characteristics of the object. Spec can contains nested specs.

- **status:**
  Describes the current state of the object, supplied and updated by the Kubernetes system and its components.

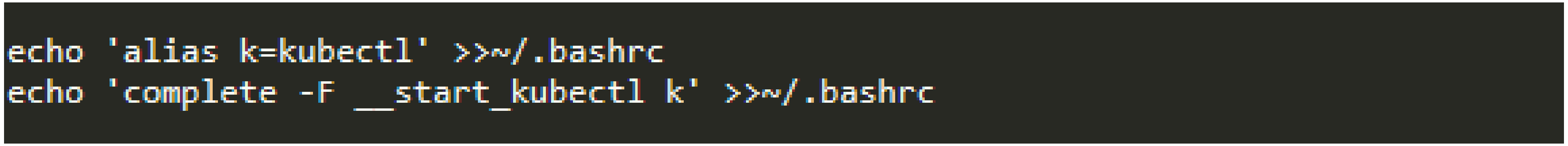
Play with kubectl - Enabling shell autocompletion



```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.18.0/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl
kubectl version
```

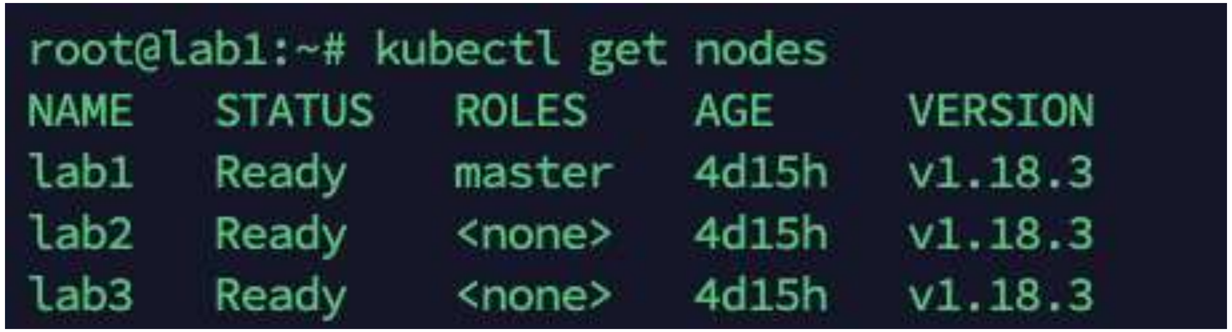


```
echo 'source <(kubectl completion bash)' >>~/.bashrc
```

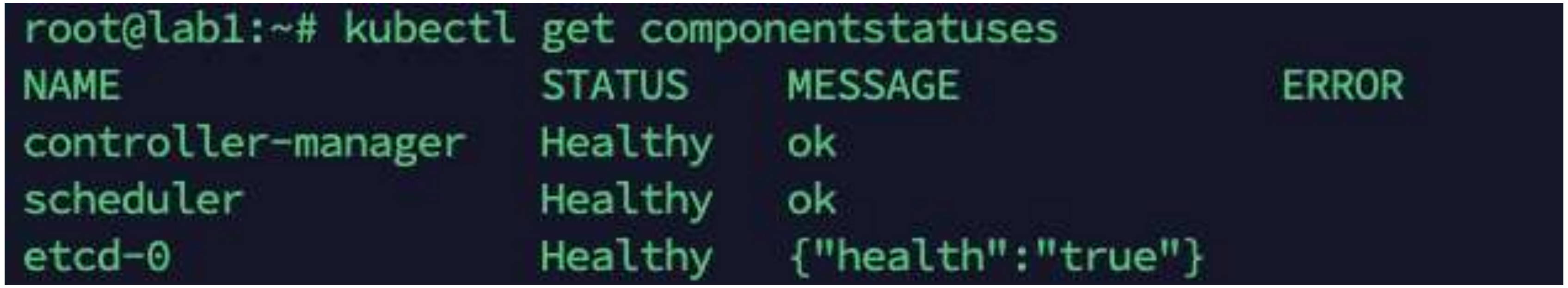


```
echo 'alias k=kubectl' >>~/.bashrc
echo 'complete -F __start_kubectl k' >>~/.bashrc
```

Play with kubectl



```
root@lab1:~# kubectl get nodes
NAME   STATUS   ROLES    AGE     VERSION
lab1   Ready    master   4d15h   v1.18.3
lab2   Ready    <none>   4d15h   v1.18.3
lab3   Ready    <none>   4d15h   v1.18.3
```



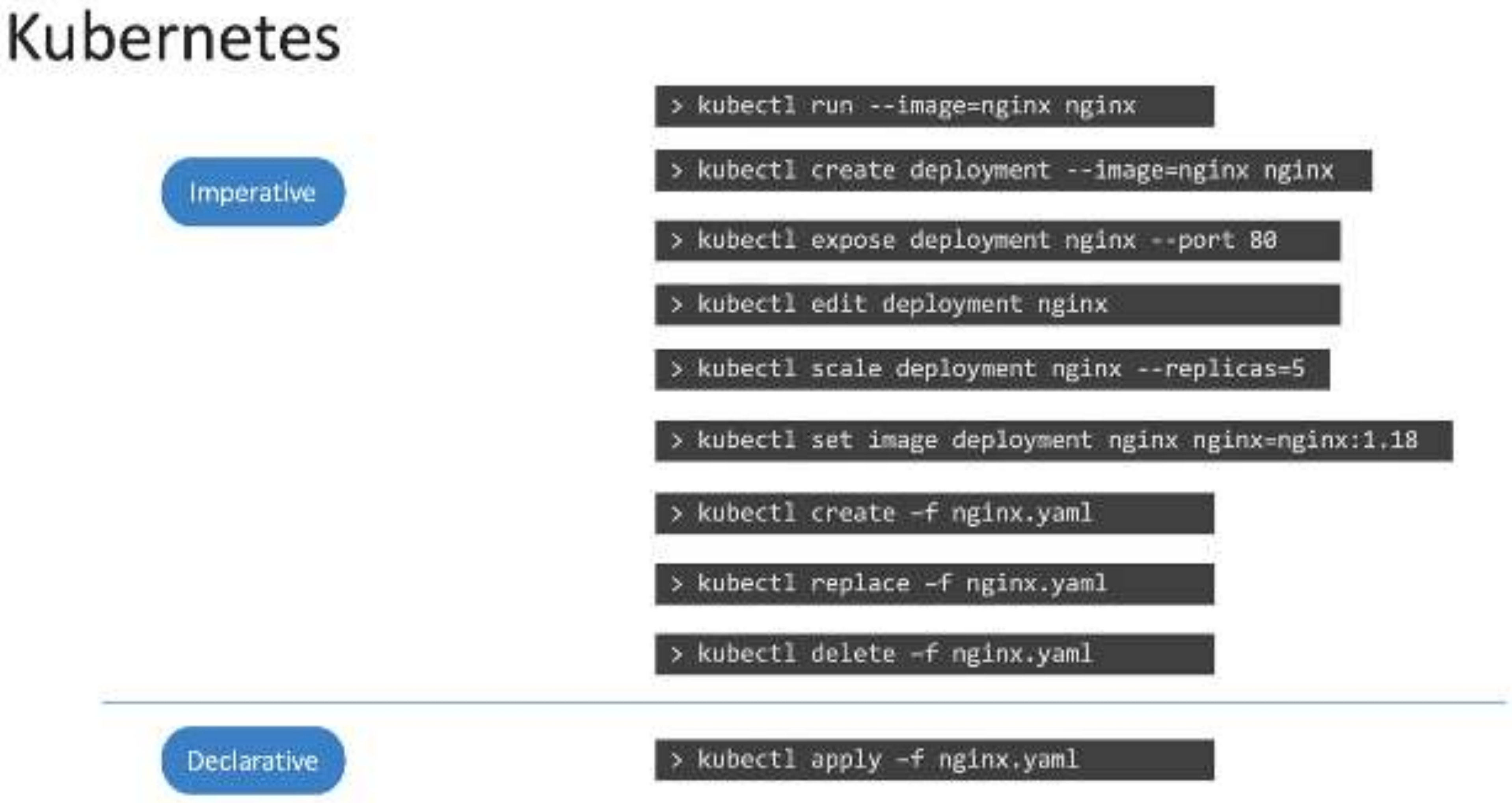
```
root@lab1:~# kubectl get componentstatuses
NAME                 STATUS    MESSAGE             ERROR
controller-manager   Healthy   ok
scheduler            Healthy   ok
etcd-0               Healthy   {"health":"true"}
```

Play with kubectl - create your first k8s object

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

- **File Extension**: both yaml & yml are accepted
- **Indent**: 2 spaces, not Tab (Tab in linux/unix are configurable)
- **Useful free course about yaml**: https://www.udemy.com/course/yaml-essentials/
- **Source Version Control:** should check in a SVC like git
- **Conversion:** kubectl converts yaml object file to JSON as the API request must be made as JSON

Imperative vs Declarative



Kubernetes

Imperative

```
> kubectl run --image=nginx nginx
```

```
> kubectl create deployment --image=nginx nginx
```

```
> kubectl expose deployment nginx --port 80
```

```
> kubectl edit deployment nginx
```

```
> kubectl scale deployment nginx --replicas=5
```

```
> kubectl set image deployment nginx nginx=nginx:1.18
```

```
> kubectl create -f nginx.yaml
```

```
> kubectl replace -f nginx.yaml
```

```
> kubectl delete -f nginx.yaml
```

Declarative

```
> kubectl apply -f nginx.yaml
```

Exploring k8s resource detail

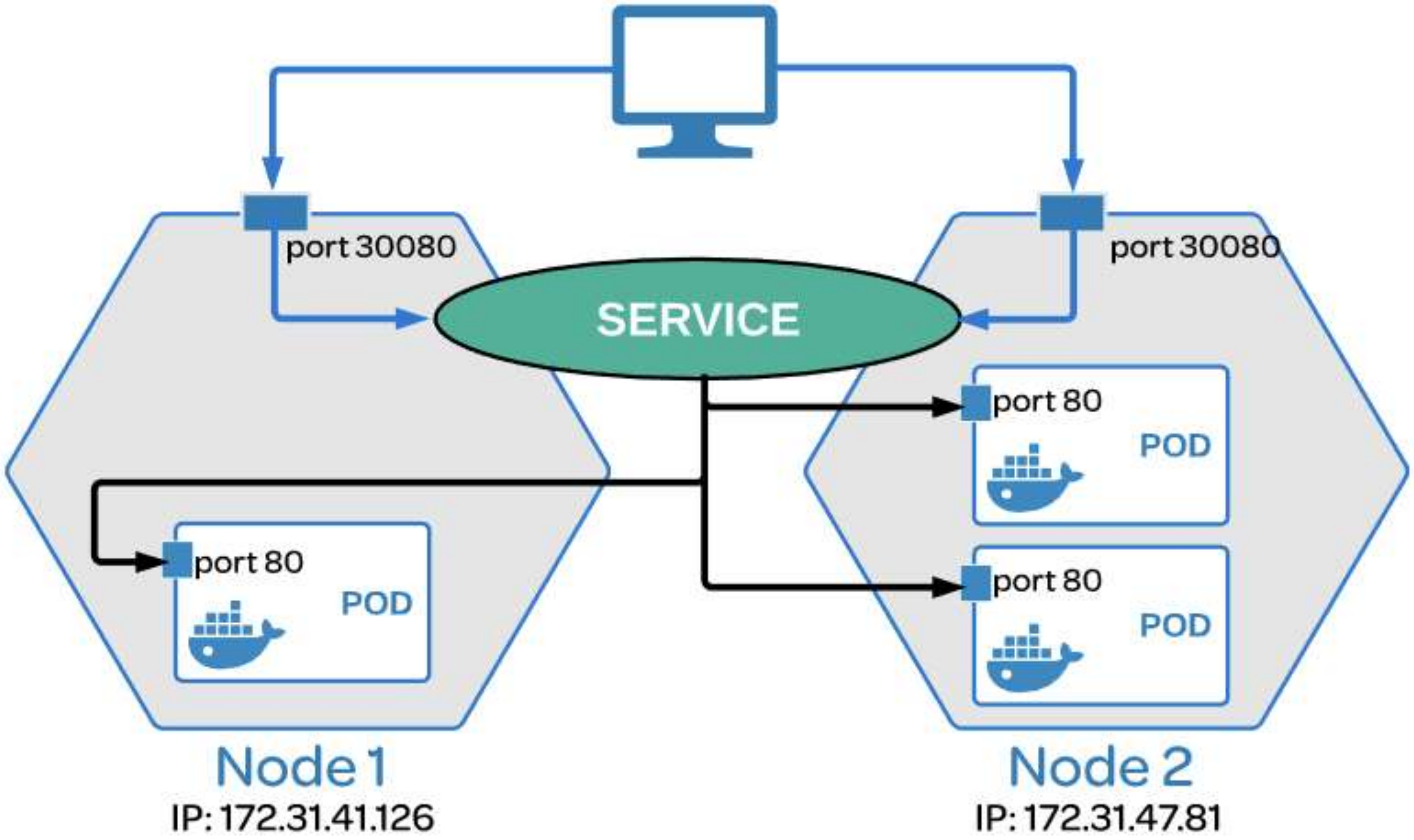
```
kubectl get deployments nginx-deployment -oyaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"name":"nginx-deployment","namespace":"default"},"spec":{"replicas":2,"selector":{"matchLabels":{"app":"nginx"}},"template":{"metadata":{"labels":{"app":"nginx"}},"spec":{"containers":[{"image":"nginx:1.7.9","name":"nginx","ports":[{"containerPort":80}]}]}}}}
  creationTimestamp: "2020-06-28T08:28:14Z"
……………...
```

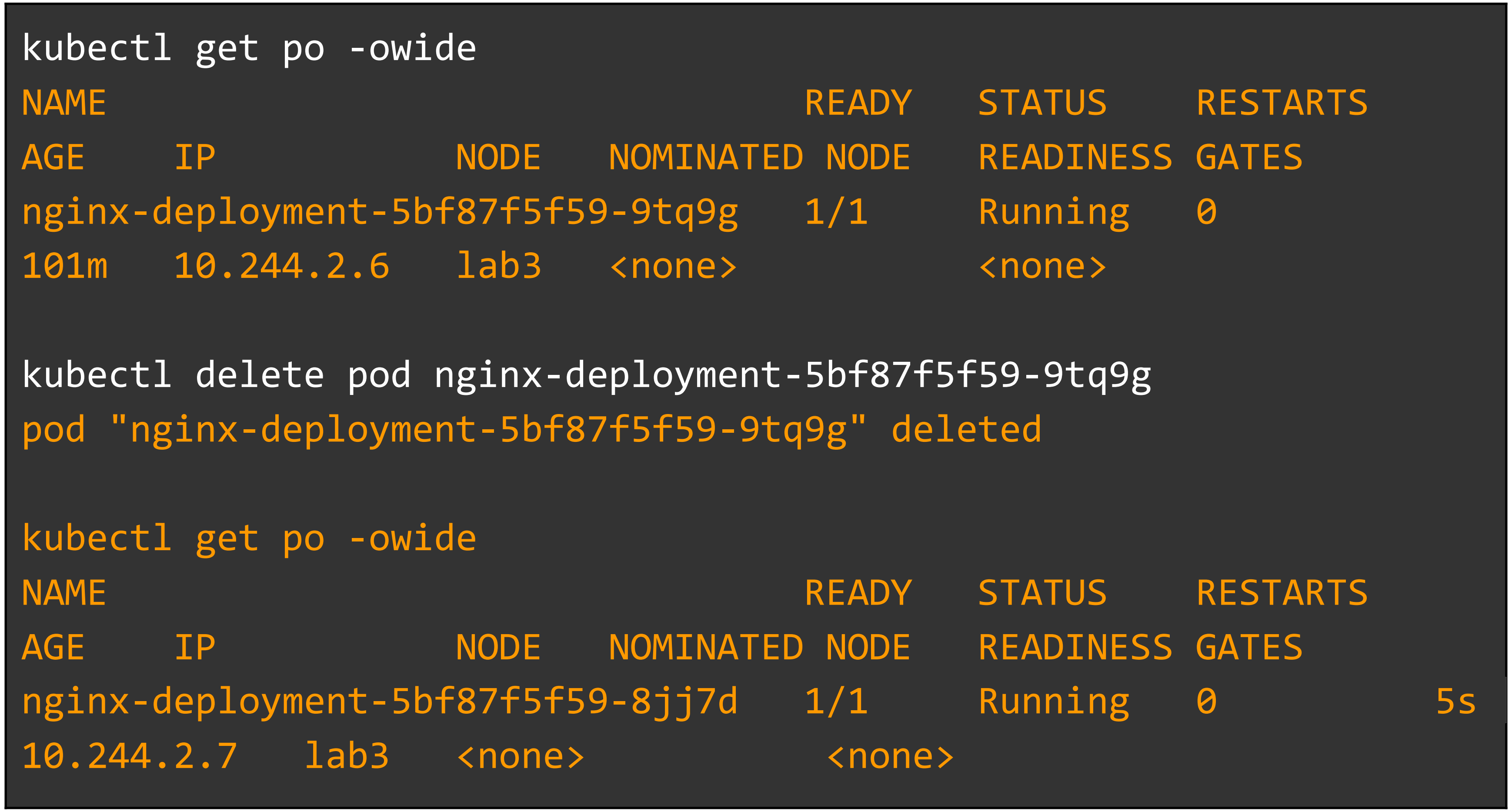
Service & Network Primitives

- Kubernetes services allow you to dynamically access a group of replica pods without having to keep track of which pods are moved, changed, or deleted.

Kubernetes Service



Look into the pods IP addresses



```
kubectl get po -owide
NAME                                READY   STATUS    RESTARTS
AGE    IP            NODE   NOMINATED NODE   READINESS GATES
nginx-deployment-5bf87f5f59-9tq9g   1/1     Running   0
101m   10.244.2.6    lab3   <none>           <none>

kubectl delete pod nginx-deployment-5bf87f5f59-9tq9g
pod "nginx-deployment-5bf87f5f59-9tq9g" deleted

kubectl get po -owide
NAME                                READY   STATUS    RESTARTS
AGE    IP            NODE   NOMINATED NODE   READINESS GATES
nginx-deployment-5bf87f5f59-8jj7d   1/1     Running   0          5s
10.244.2.7   lab3   <none>           <none>
```

k8s service

```
cat <<'EOF' | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
    nodePort: 30080
  selector:
    app: nginx
EOF

service/nginx-nodeport created
```

Access application through NodePort

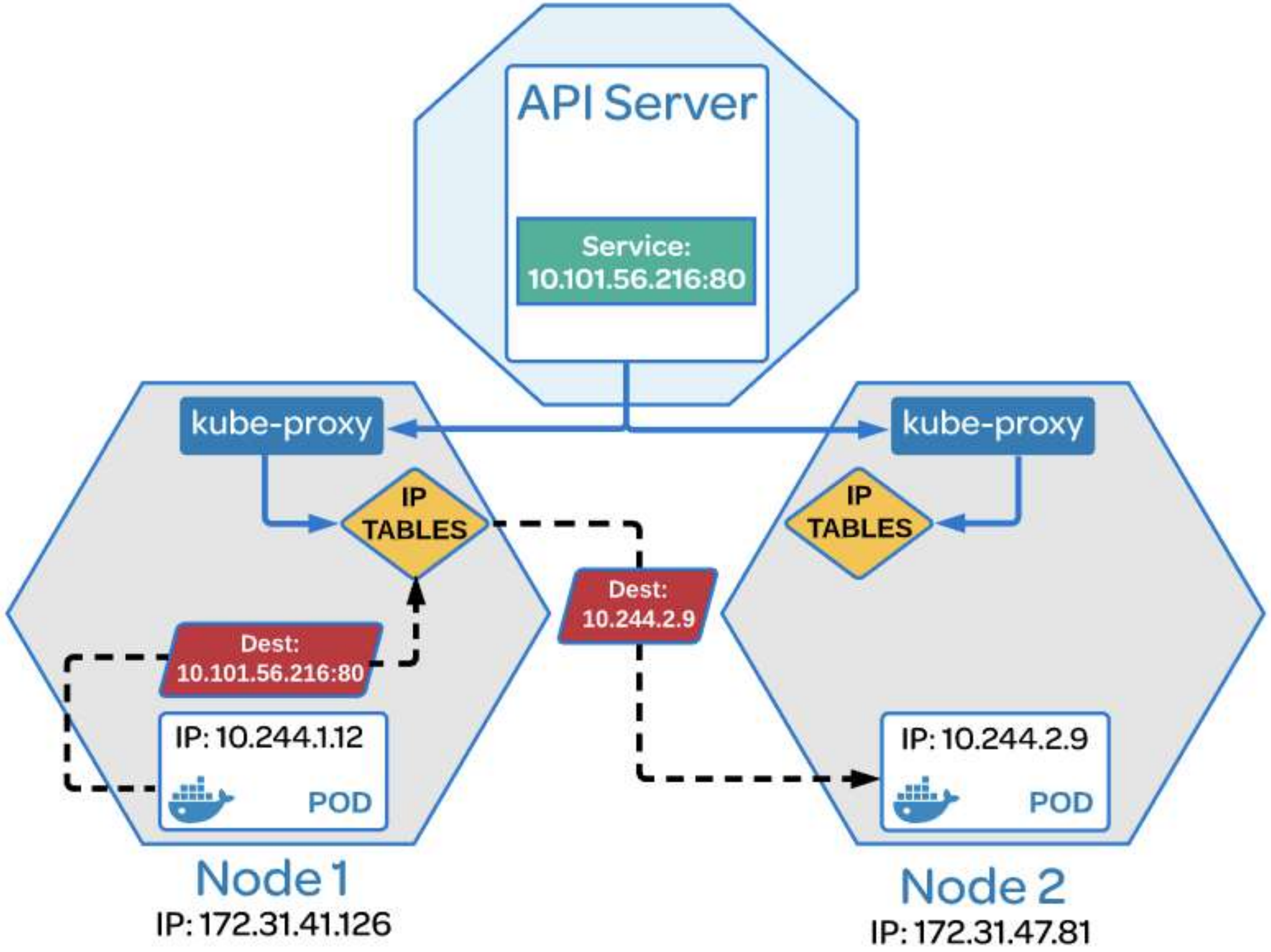
```
curl http://<worker_ip>:<node_port>

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

kube-proxy & iptables

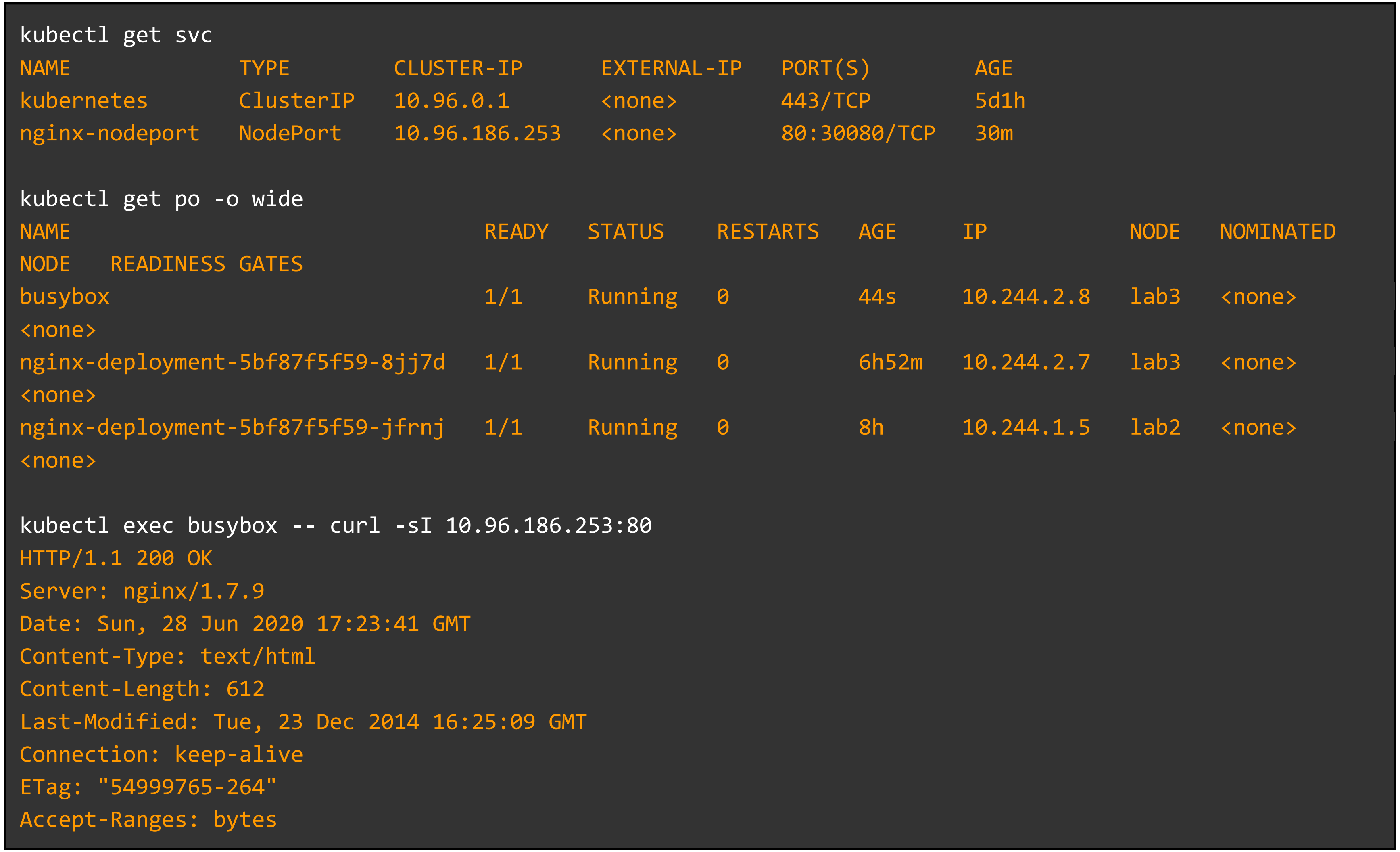


Create a busybox pod

```
cat <<'EOF' | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: busybox
spec:
  containers:
  - name: busybox
    image: radial/busyboxplus:curl
    args:
    - sleep
    - "1000"
EOF

pod/busybox created
```

Inter-cluster communication



```
kubectl get svc
NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1      <none>        443/TCP        5d1h
nginx-nodeport   NodePort    10.96.186.253  <none>        80:30080/TCP   30m

kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE     IP            NODE   NOMINATED
NODE   READINESS GATES
busybox                             1/1     Running   0          44s     10.244.2.8    lab3   <none>
<none>
nginx-deployment-5bf87f5f59-8jj7d   1/1     Running   0          6h52m   10.244.2.7    lab3   <none>
<none>
nginx-deployment-5bf87f5f59-jfrnj   1/1     Running   0          8h      10.244.1.5    lab2   <none>
<none>

kubectl exec busybox -- curl -sI 10.96.186.253:80
HTTP/1.1 200 OK
Server: nginx/1.7.9
Date: Sun, 28 Jun 2020 17:23:41 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 23 Dec 2014 16:25:09 GMT
Connection: keep-alive
ETag: "54999765-264"
Accept-Ranges: bytes
```

Q
A