DEVELOPER INTRO TO kubernetes

Daniel Pokusa Paweł Młynarczyk

https://github.com/kubernetes/minikube

```
$ minikube start
$ minikube start --memory=6000
```

VT-x enabled in BIOS

Installed kubectl

Virtualization system installed

(virtualbox, vmwarefusion, KVM, xhyve, Hyper-V)

KUBERNETES

- Google

KUBERNETES

- Google
- Projekt został uruchomiony w 2014

KUBERNETES

- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)

KUBERNETES

- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)
- Pierwszy “release” w Lipcu 2015

KUBERNETES

- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)
- Pierwszy “release” w Lipcu 2015
- 100% Open Source

KUBERNETES

- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)
- Pierwszy “release” w Lipcu 2015
- 100% Open Source
- Go

KUBERNETES

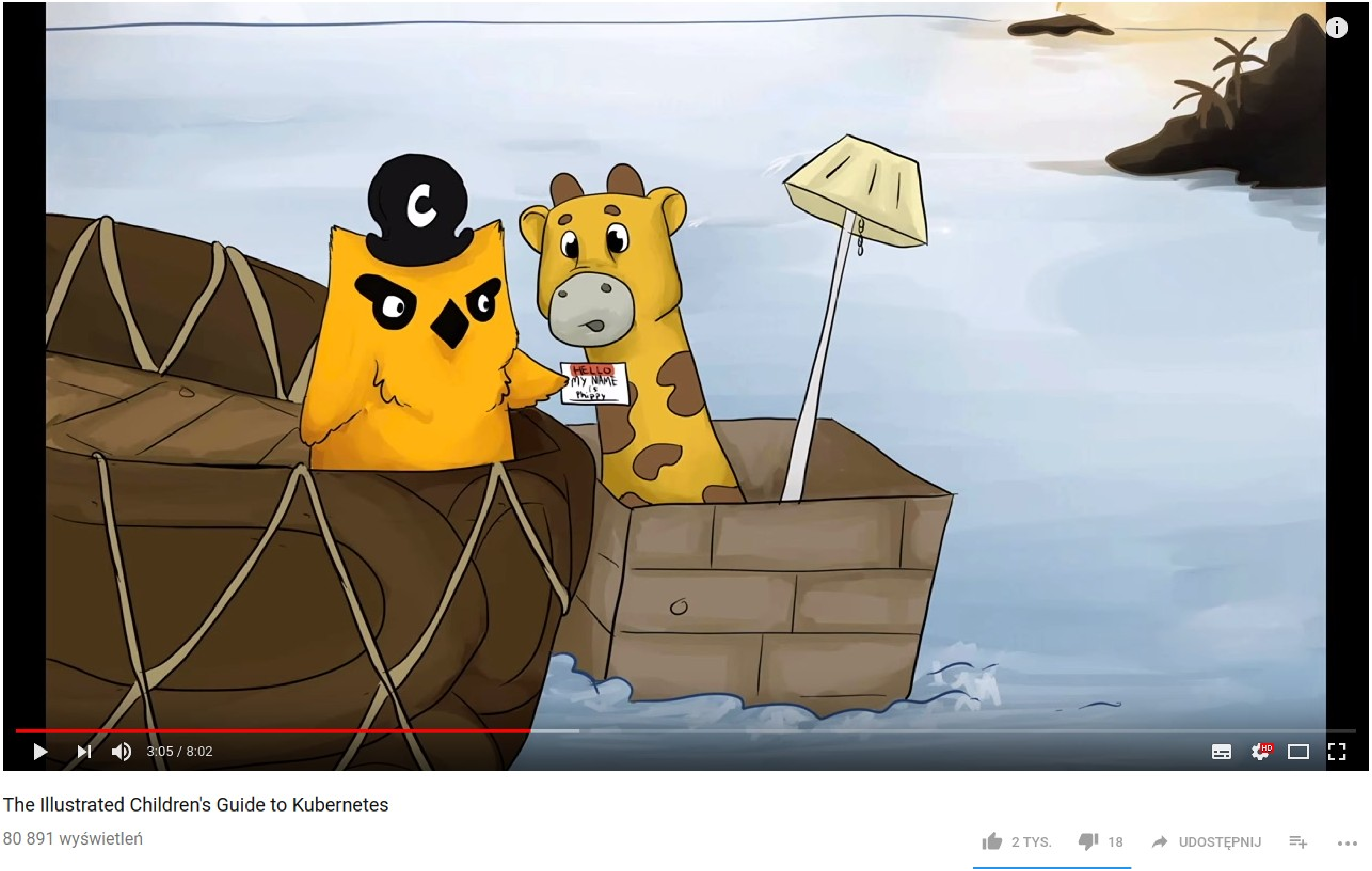
- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)
- Pierwszy “release” w Lipcu 2015
- 100% Open Source
- Go
- Z greckiego “*Sternik*” (helmsman or pilot)

KUBERNETES

- Google
- Projekt został uruchomiony w 2014
- Oparty o ponad 10 letnie doświadczenie (“Borg”)
- Pierwszy “release” w Lipcu 2015
- 100% Open Source
- Go
- Z greckiego “*Sternik*” (helmsman or pilot)
- Każda większa chmura aktualnie wspiera k8s!

The Illustrated Children's Guide to Kubernetes

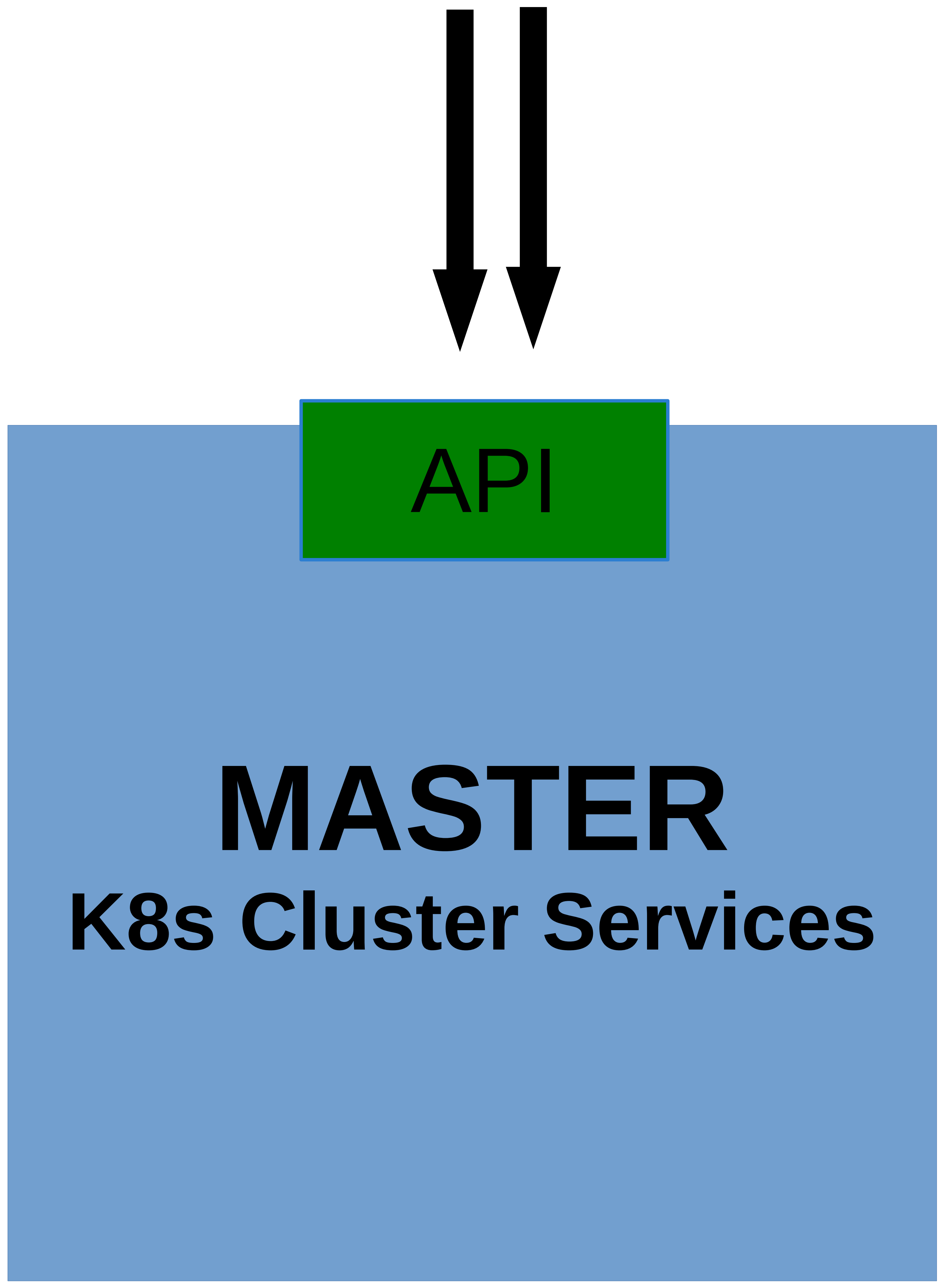
80 891 wyświetleń



The Illustrated Children's Guide to Kubernetes

80 891 wyświetleń

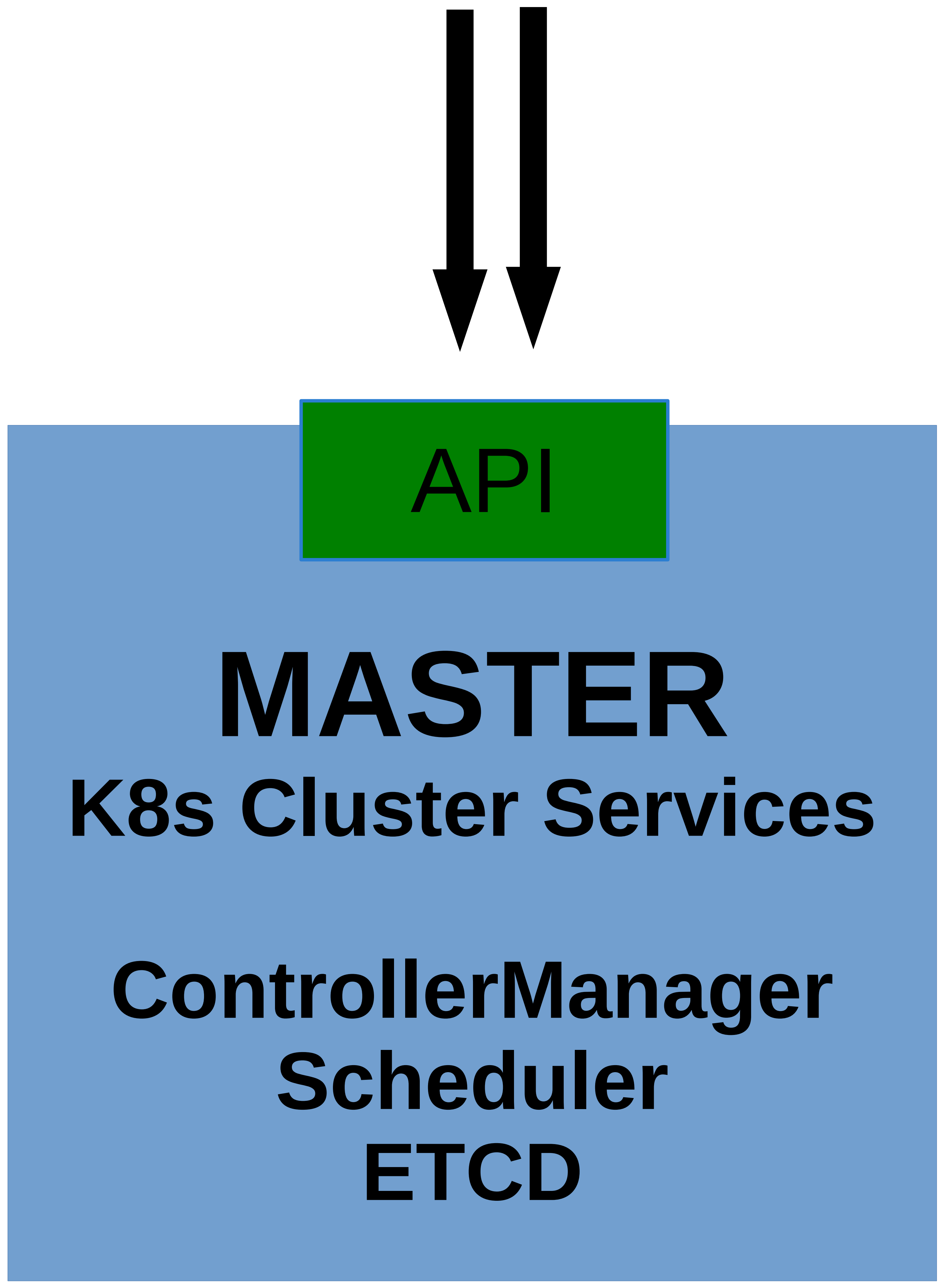
DESIRED STATE MANAGEMENT



API

MASTER

K8s Cluster Services

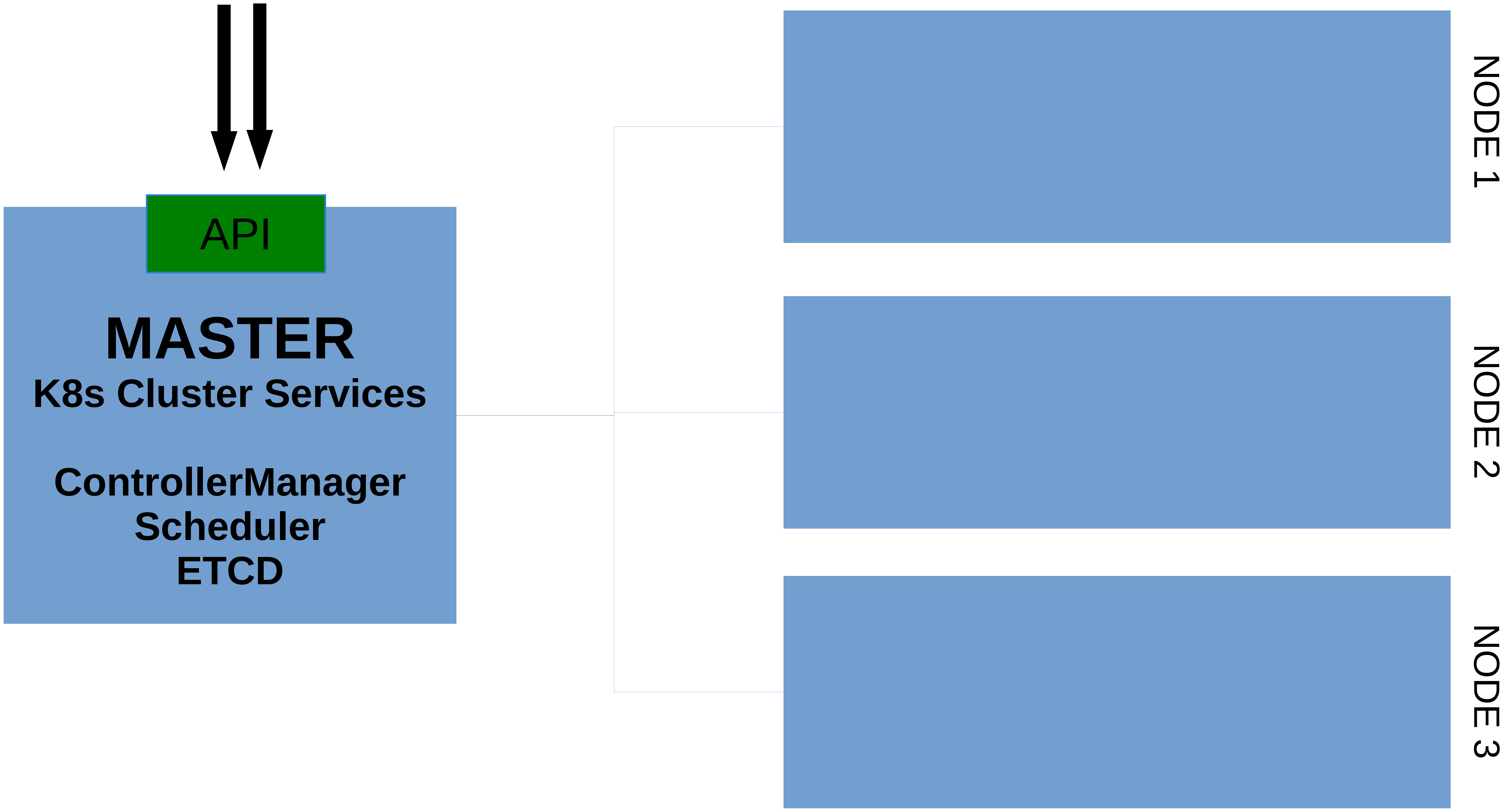


API

MASTER

K8s Cluster Services

ControllerManager
Scheduler
ETCD



API

MASTER

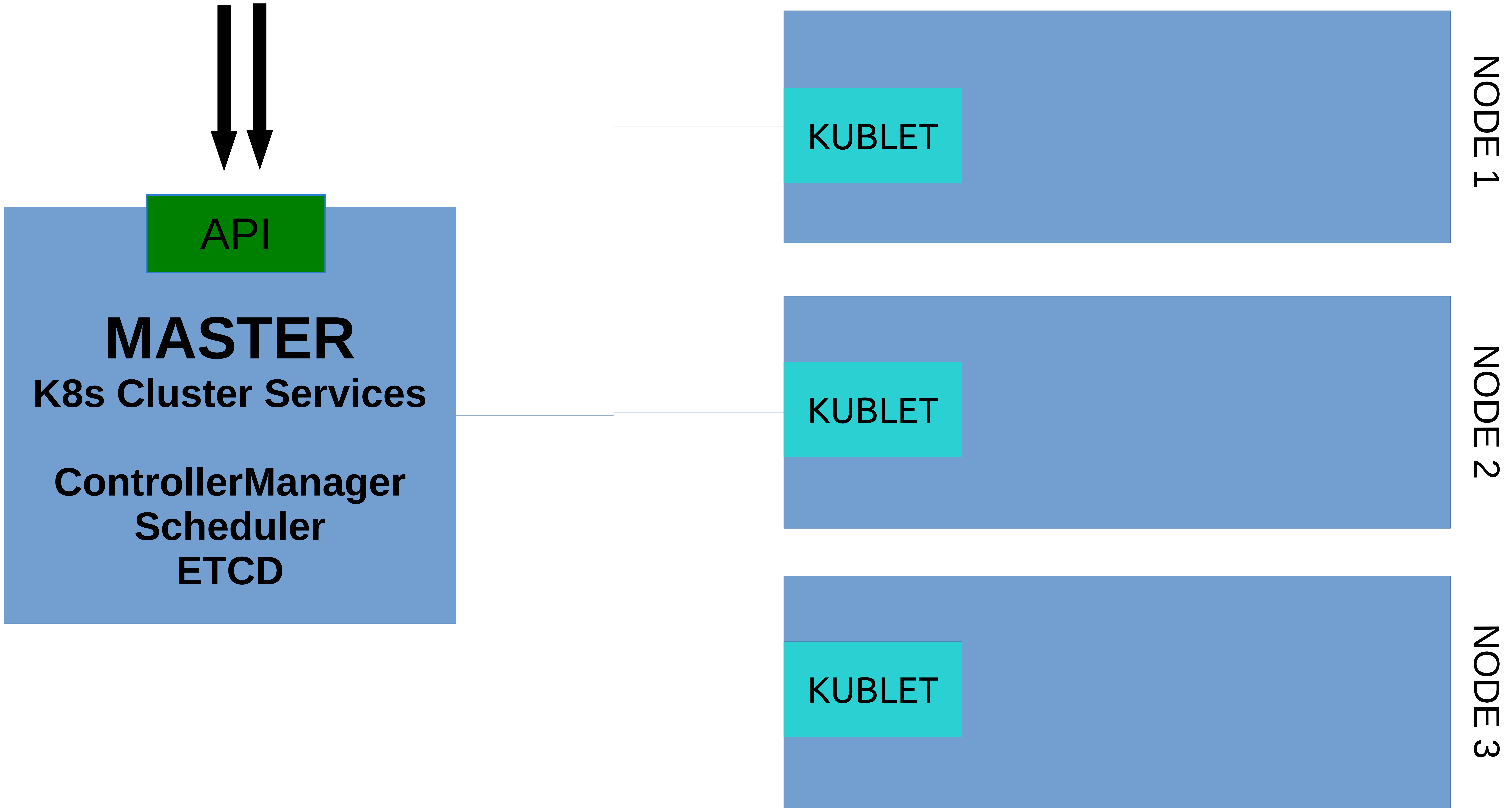
K8s Cluster Services

ControllerManager
Scheduler
ETCD

NODE 1

NODE 2

NODE 3



API

MASTER

K8s Cluster Services

ControllerManager
Scheduler
ETCD

KUBLET

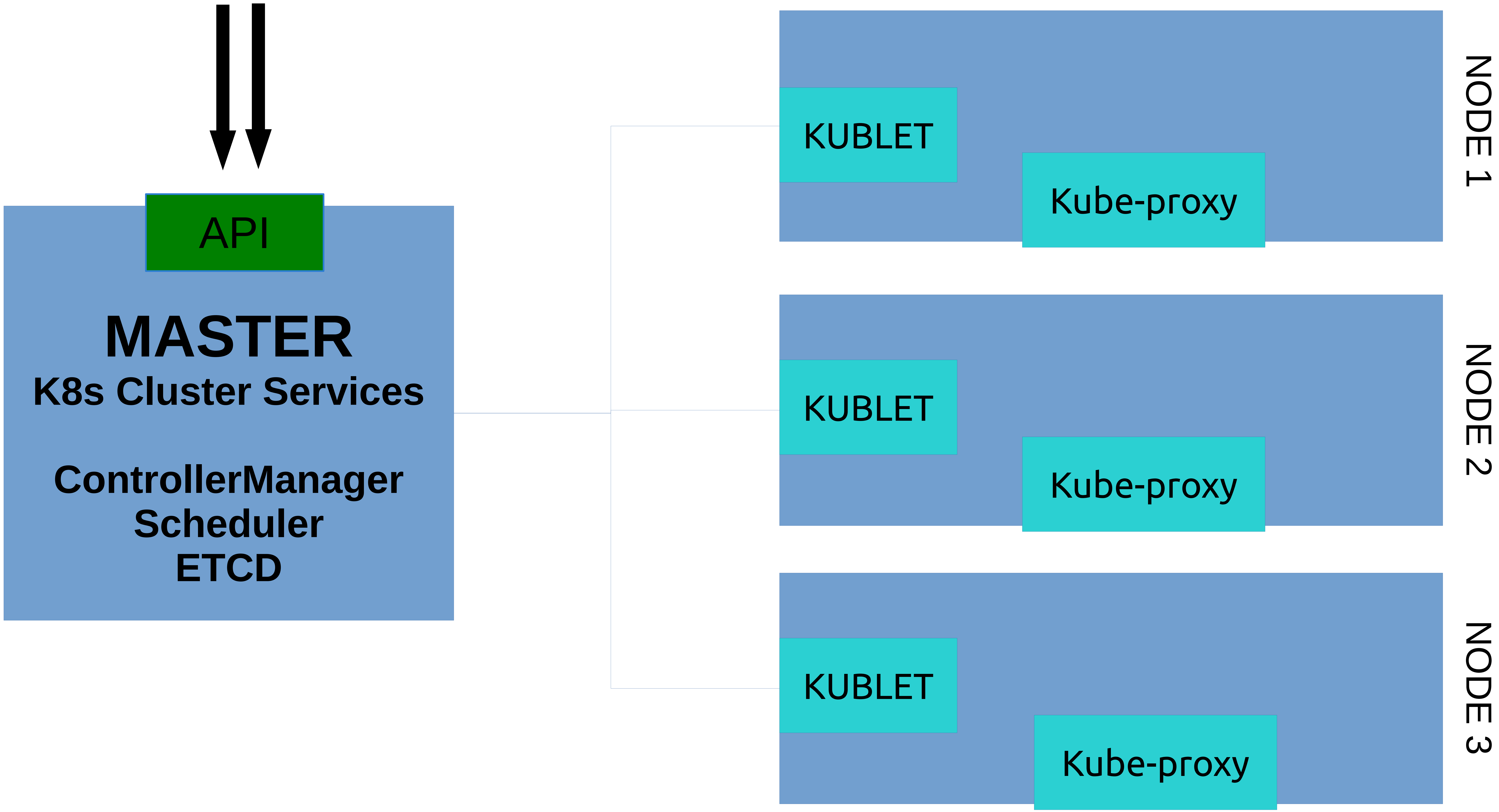
NODE 1

KUBLET

NODE 2

KUBLET

NODE 3



API

MASTER

K8s Cluster Services

ControllerManager
Scheduler
ETCD

KUBLET

Kube-proxy

NODE 1

KUBLET

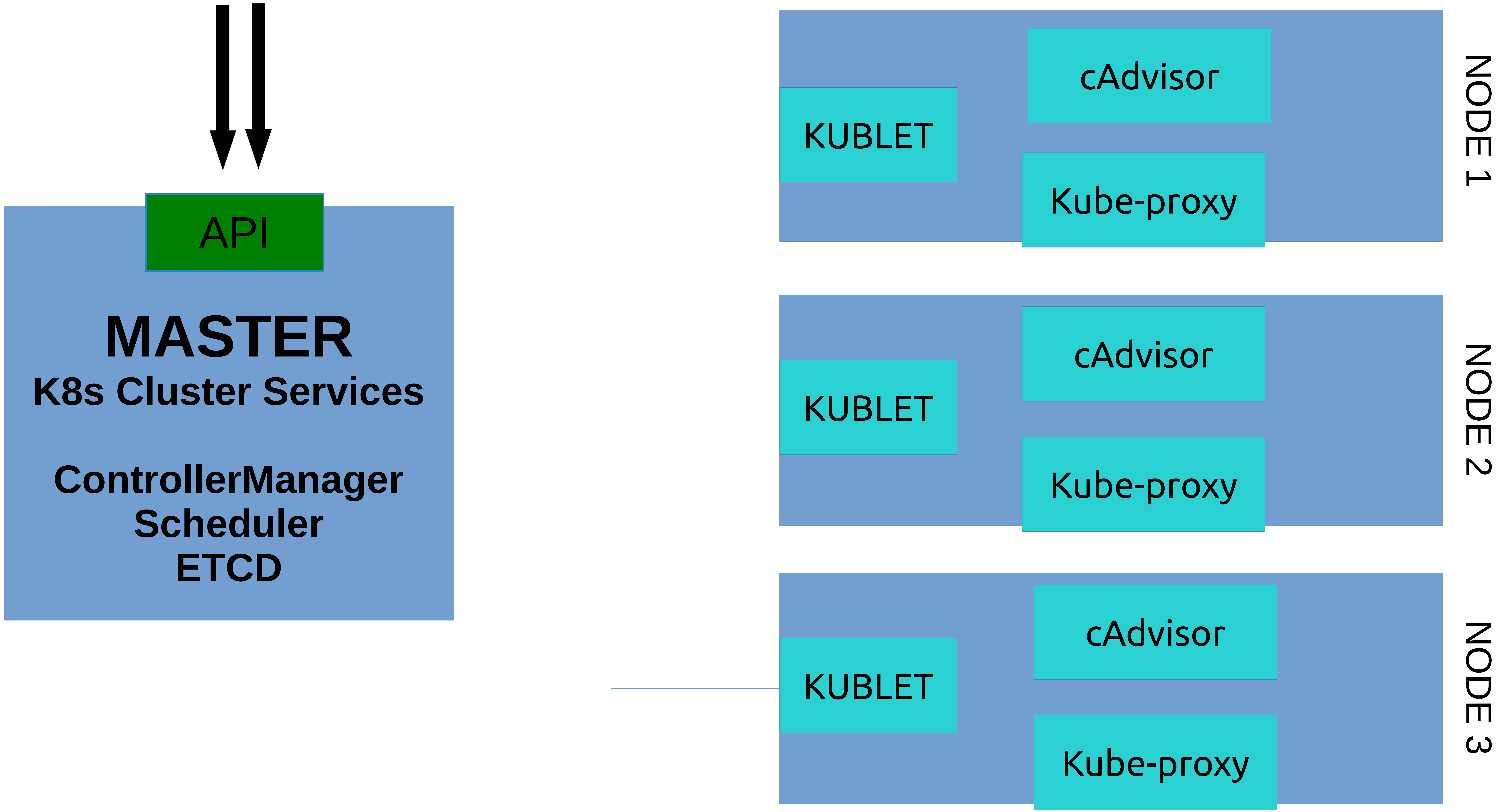
Kube-proxy

NODE 2

KUBLET

Kube-proxy

NODE 3



API

MASTER

K8s Cluster Services

ControllerManager
Scheduler
ETCD

KUBLET

cAdvisor

Kube-proxy

NODE 1

KUBLET

cAdvisor

Kube-proxy

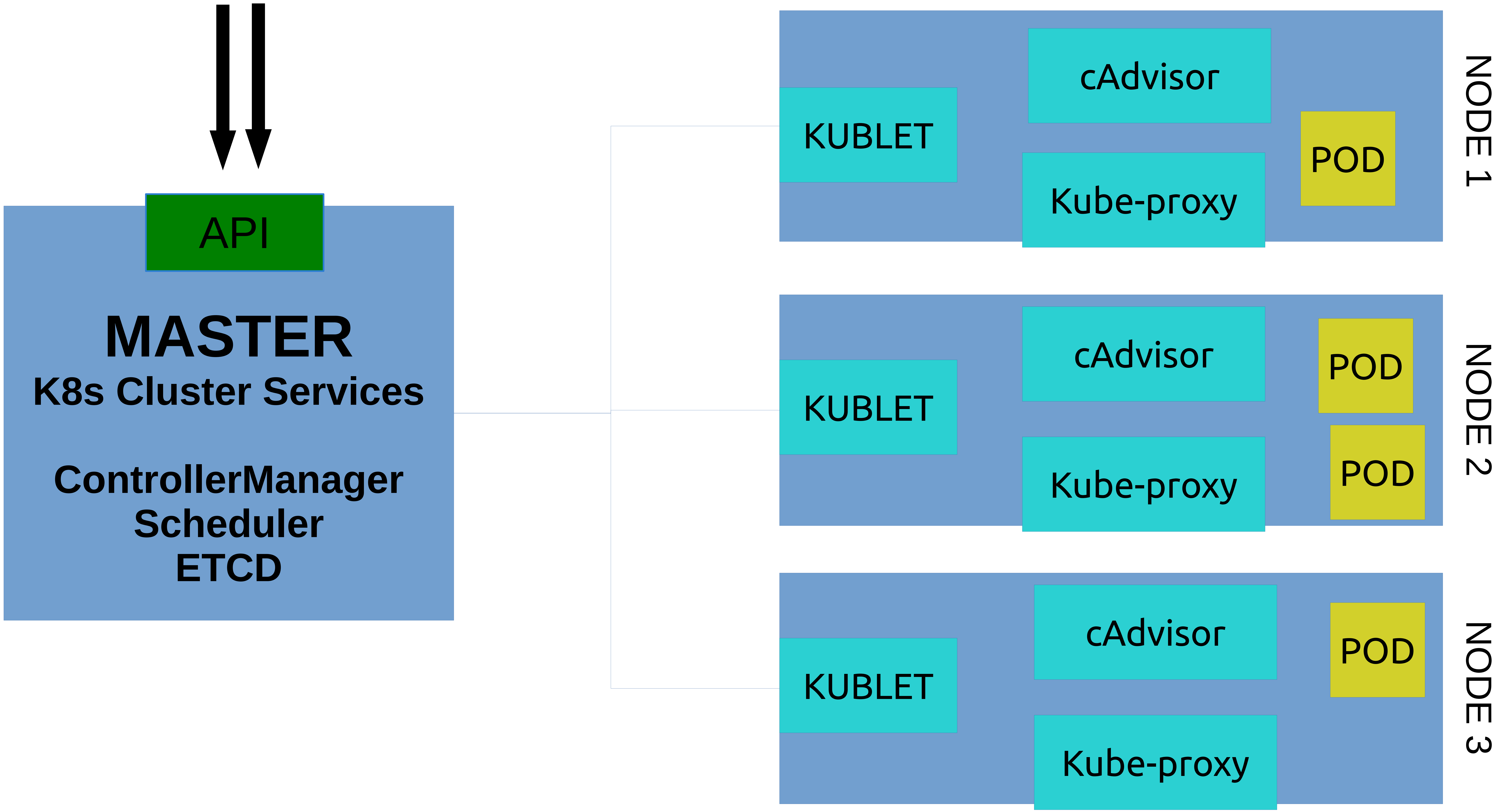
NODE 2

KUBLET

cAdvisor

Kube-proxy

NODE 3



API

MASTER

K8s Cluster Services

ControllerManager

Scheduler

ETCD

KUBLET

cAdvisor

Kube-proxy

POD

NODE 1

KUBLET

cAdvisor

Kube-proxy

POD

POD

NODE 2

KUBLET

cAdvisor

Kube-proxy

POD

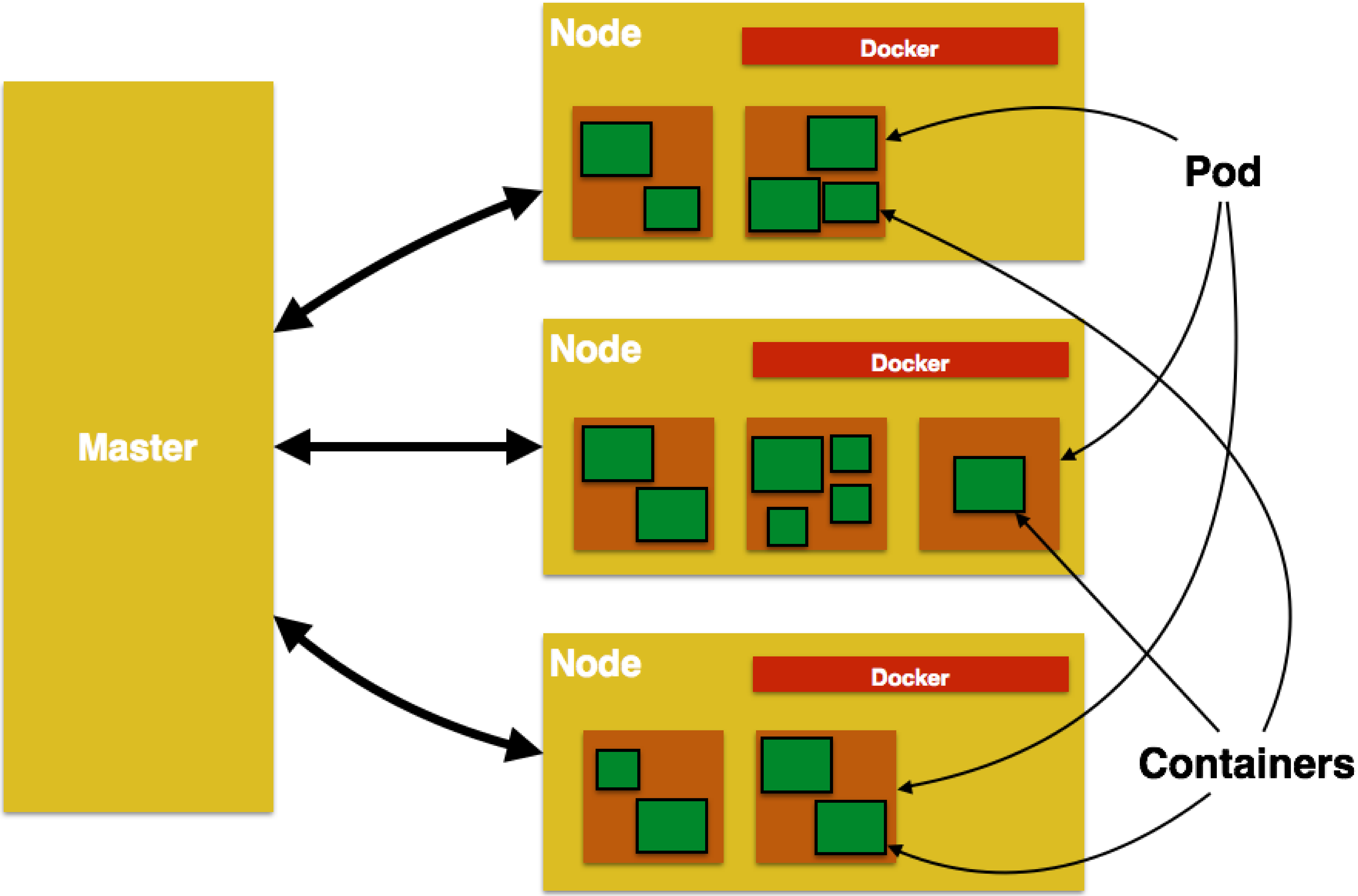
NODE 3

TASK 0

```
 eval $(minikube docker-env)

minikube dashboard
```

TASK 1

POD



http://blog.arungupta.me/wp-content/uploads/2015/07/kubernetes-key-concepts.png

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
    component: my-pod
spec:
  containers:
    - image: some-image:1.0
      name: my-pod
      ports:
        - containerPort: 8080
```

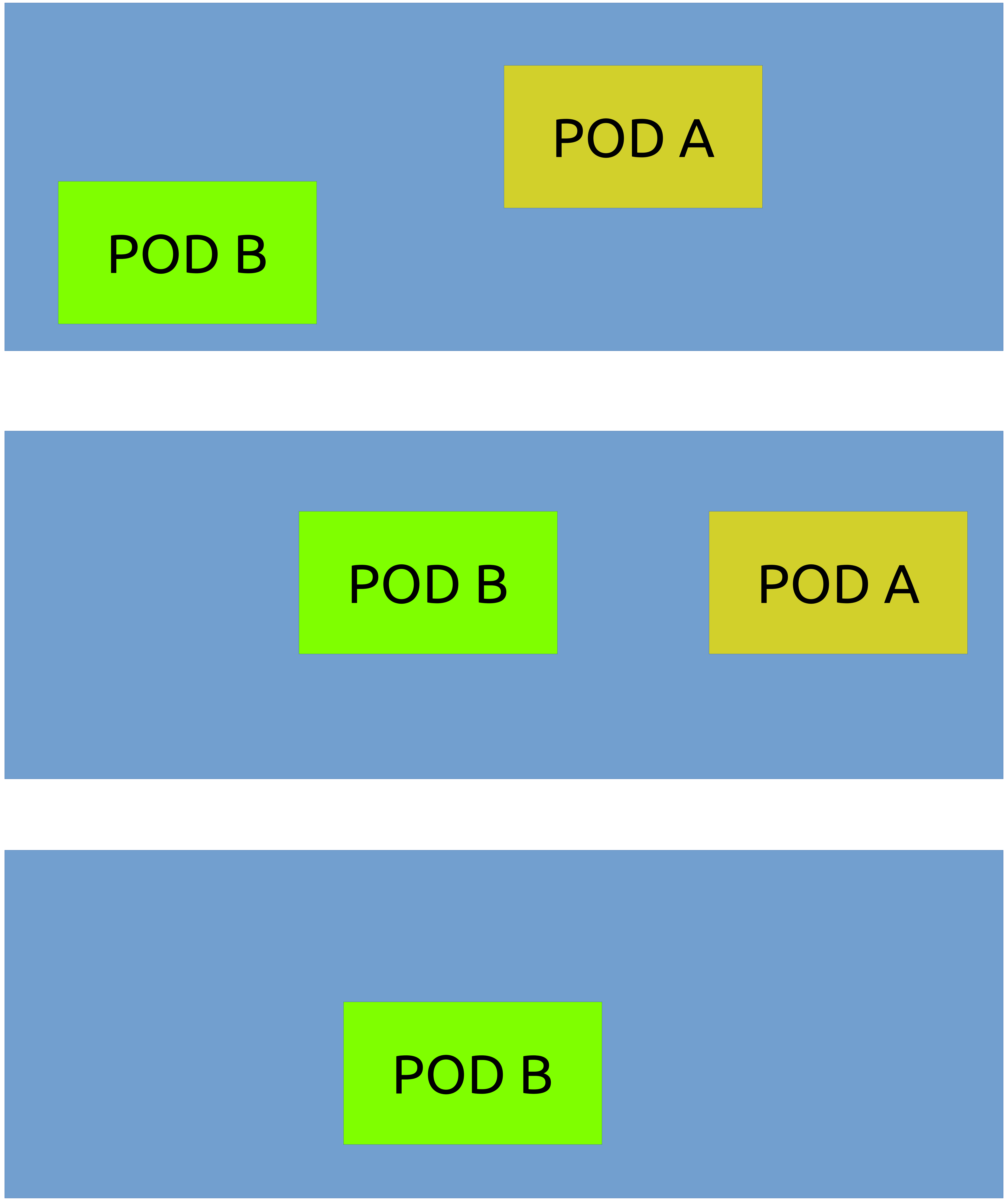
kubectl

```
kubectl create

kubectl delete

kubectl get

kubectl describe

kubectl logs

kubectl exec
```

TASK 2

SERVICE

```yaml
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```



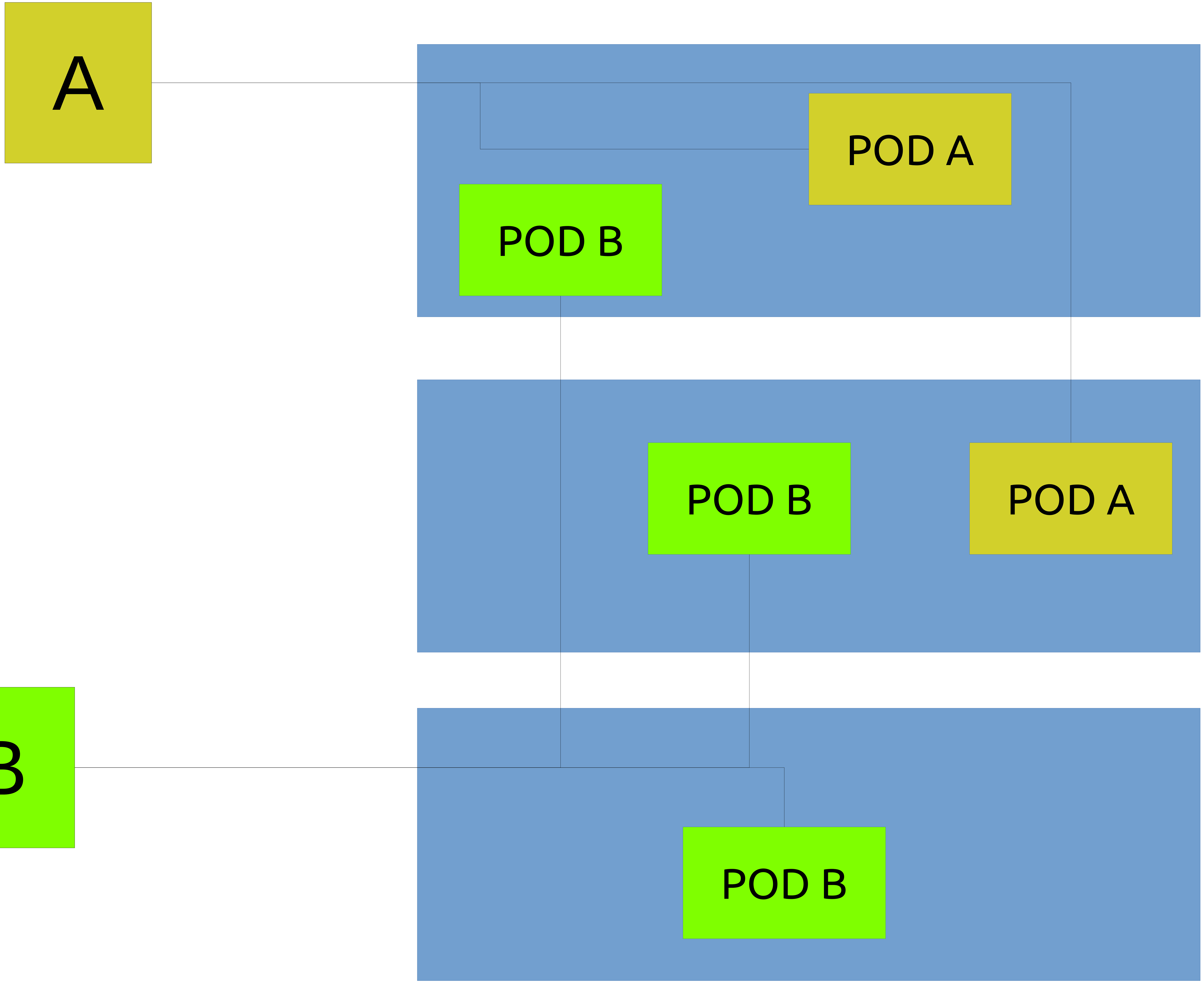
POD A

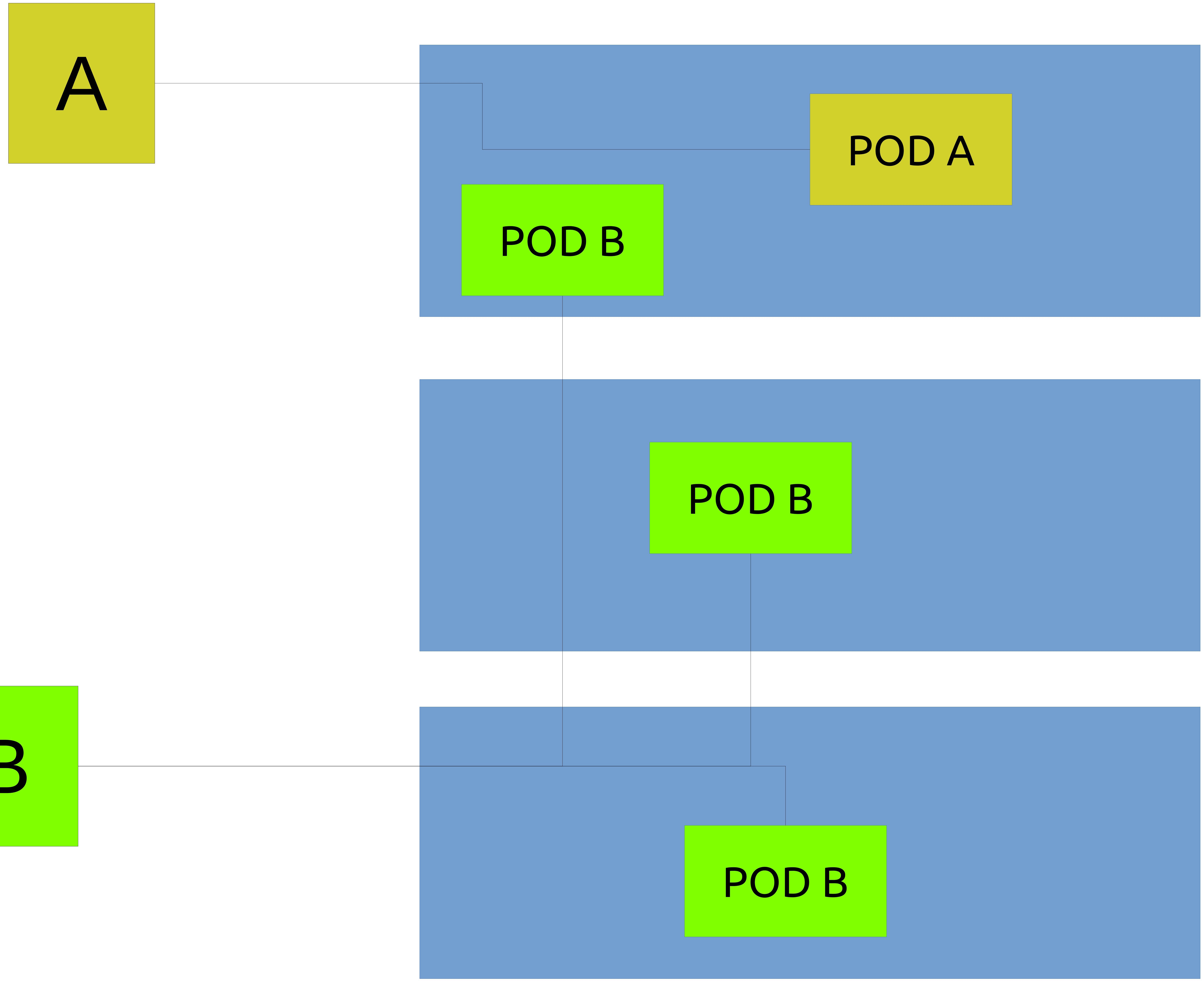
POD B

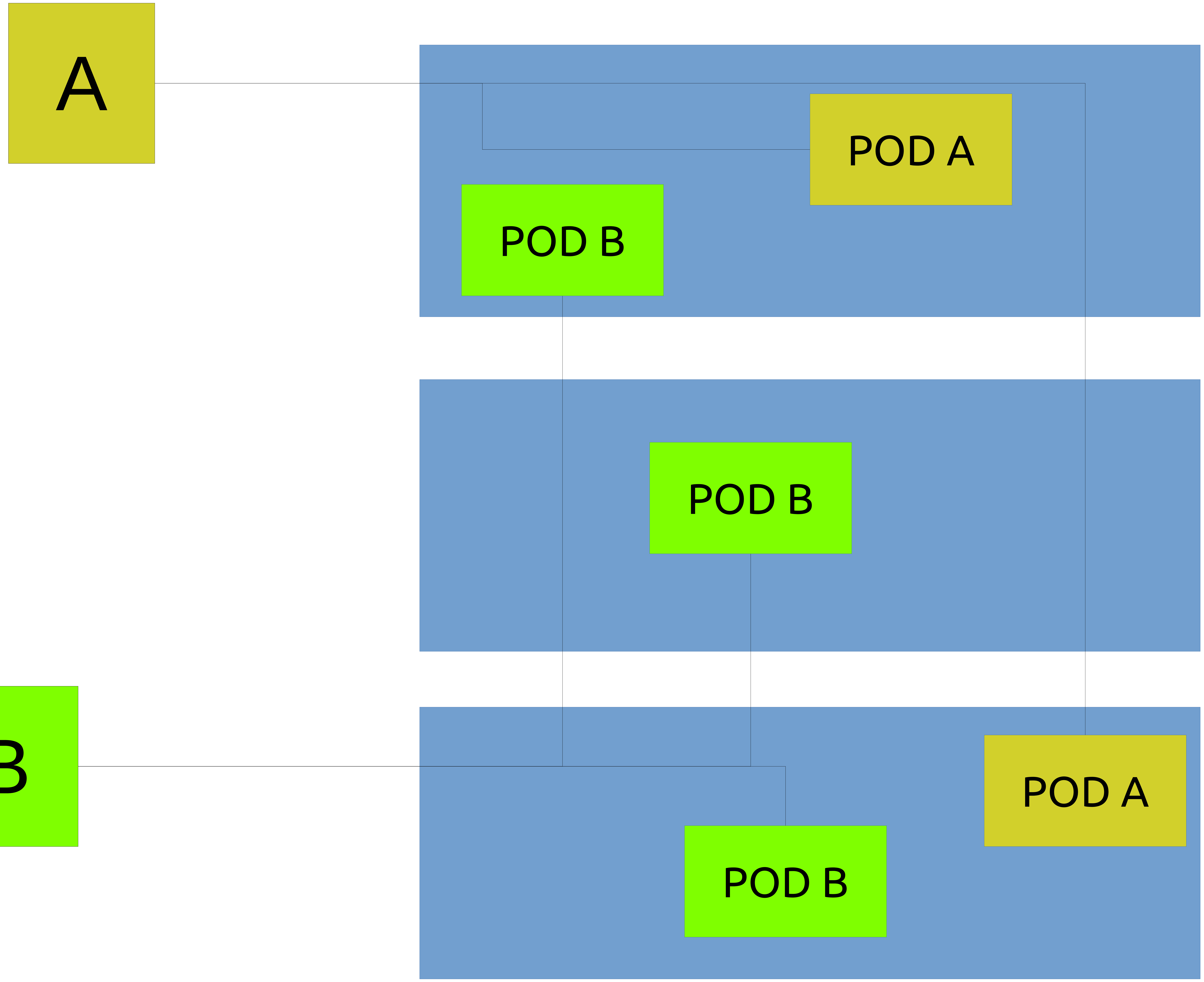
POD B

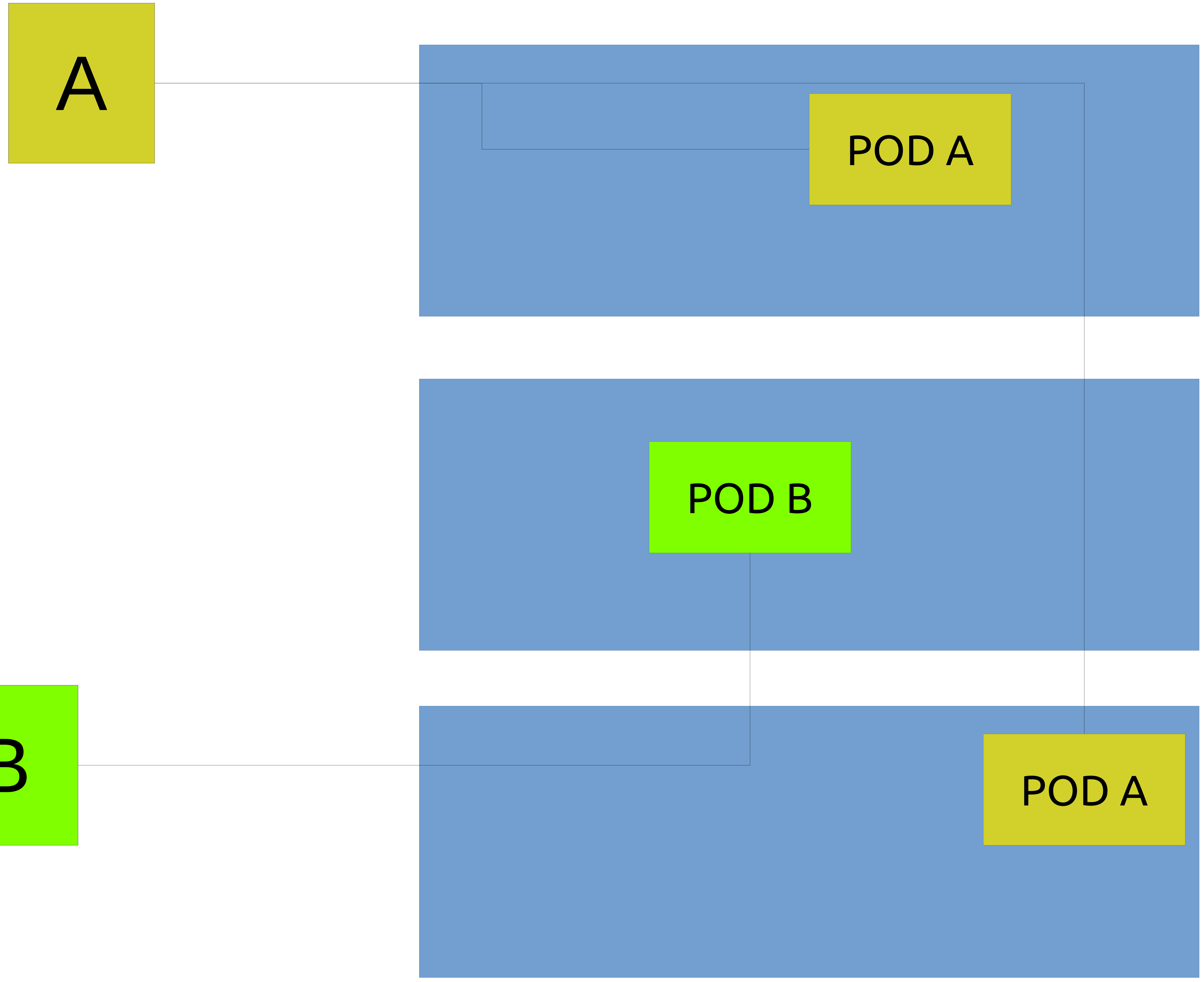
POD A

POD B

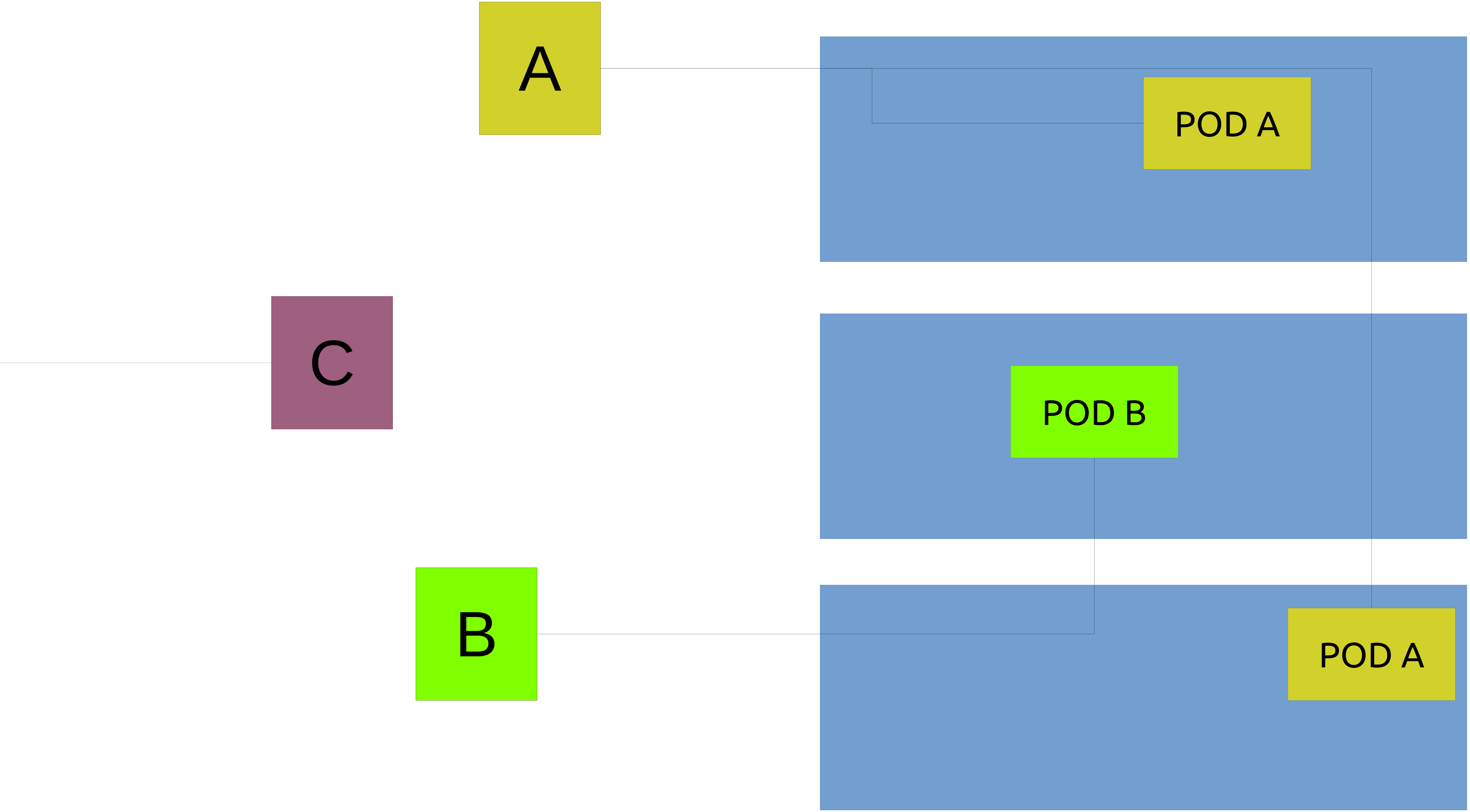

A
POD A
POD B
POD B
POD A
B
POD B

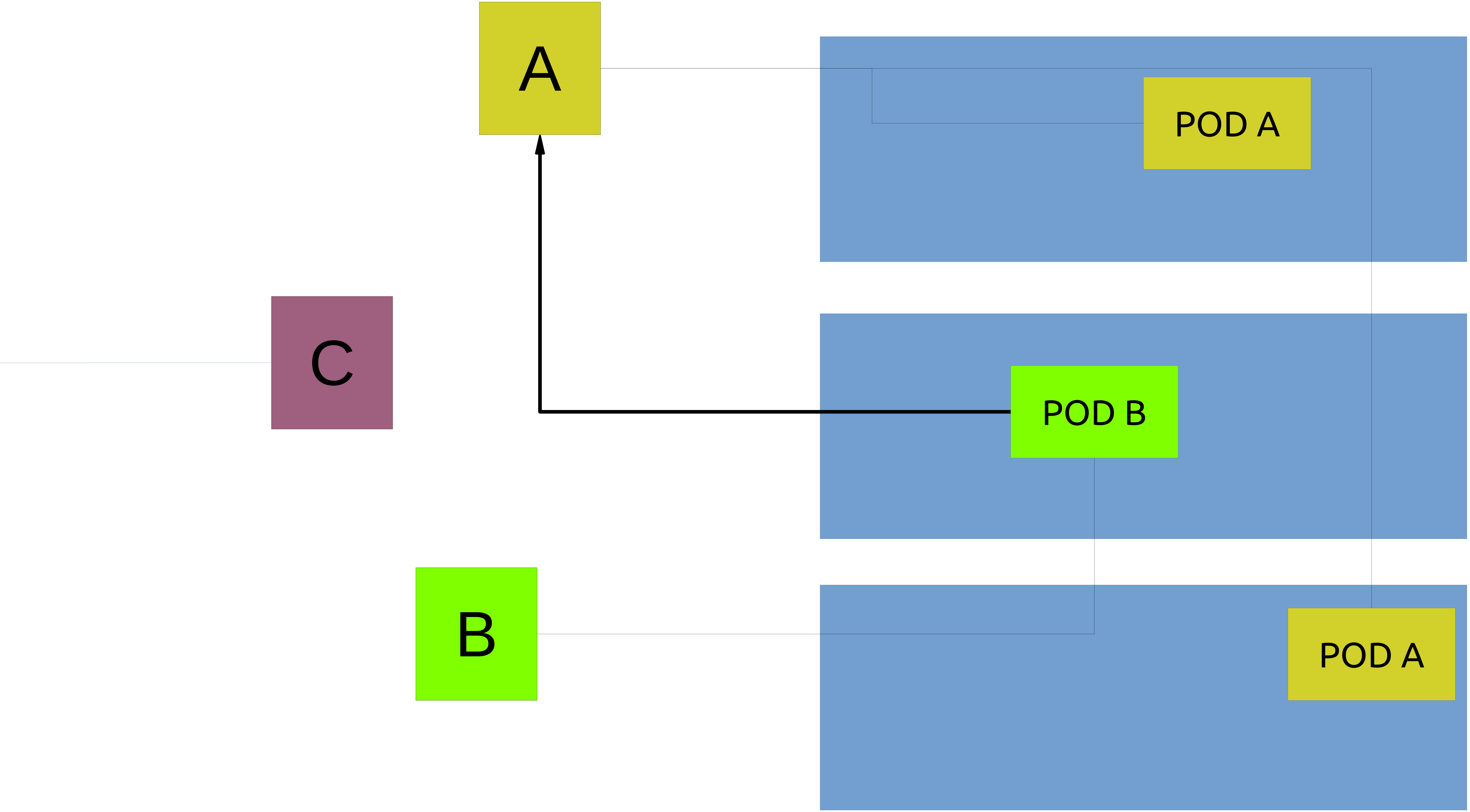

A
POD A
POD B
POD B
B
POD B

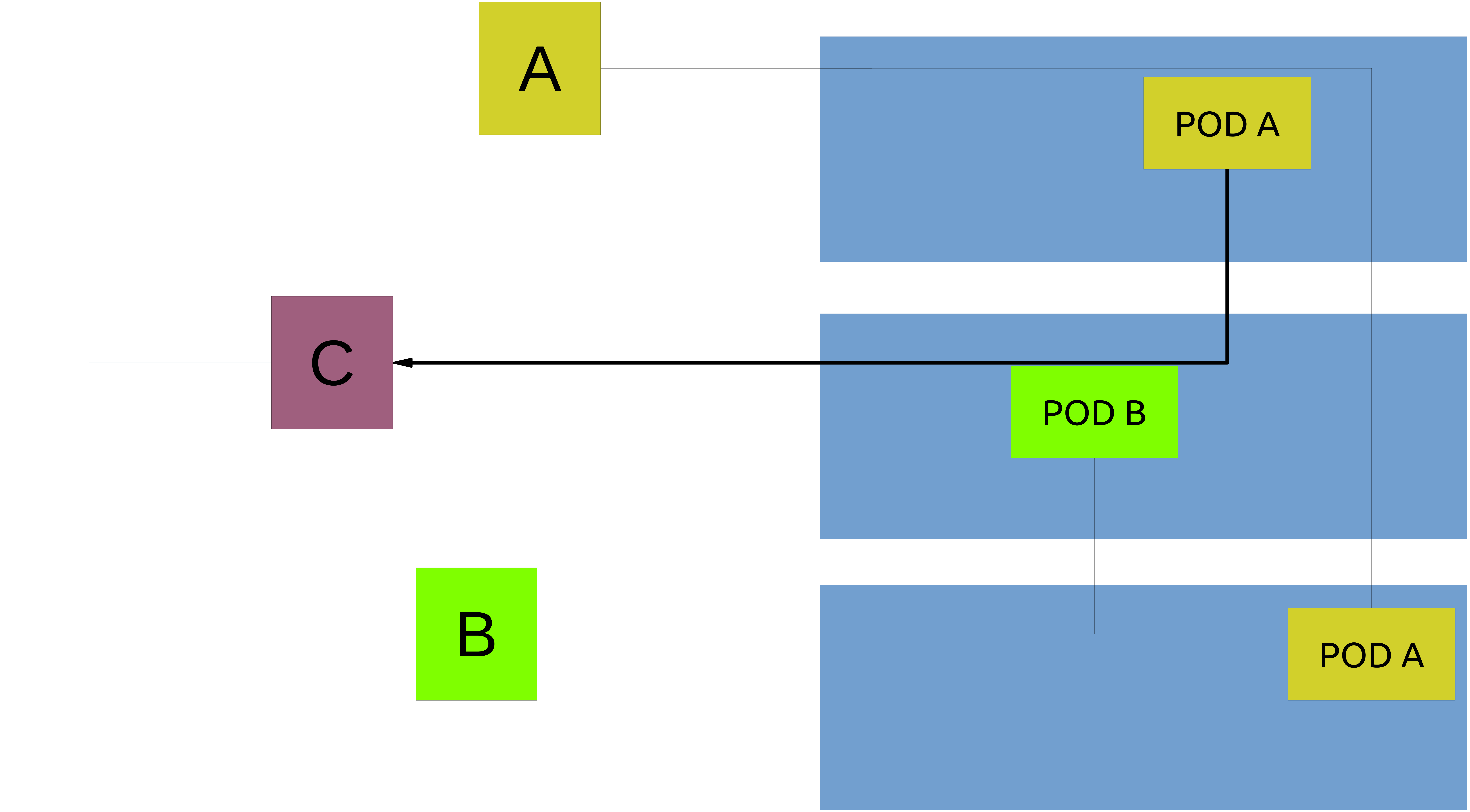

A
POD A
POD B
POD B
B
POD A
POD B


A
POD A
POD B
B
POD A

```yaml
kind: Service
apiVersion: v1
metadata:
  name: my-service
  namespace: prod
spec:
  type: ExternalName
  externalName: my.database.example.com
```


A
POD A
C
POD B
B
POD A


A
POD A
C
POD B
B
POD A


A
POD A
C
POD B
B
POD A

SERVICE DISCOVERY

```
kubectl run curl --image=radial/busyboxplus:curl -i --tty
```

TASK 3

DEPLOYMENT

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

TASK 4

CONFIG MAPS

```
kubectl create configmap spring-app-config
  --from-file=src/main/resources/application.properties
```

TASK 5

HEALTH

TASK 6

SECRETS

```
kubectl create secret generic mongodb-credentials
    --from-literal=username=user -from-literal=password=pass
```

```
spec:
      containers:
        - name: my-app
          image: my-app:0.0.1-SNAPSHOT
          ports:
          - containerPort: 8080
          env:
          - name: LOG_APPENDER
            value: Console
          - name: HRPROJECTS_MONGODB_PASSWORD
            valueFrom:
              secretKeyRef:
                name: mongodb-credentials
                key: password
```

```
kubectl create secret generic mongodb-credentials
    --from-literal=username=user -from-literal=password=pass
```

```
spec:
      containers:
        - name: my-app
          image: my-app:0.0.1-SNAPSHOT
          - name: LOG_APPENDER
            value: Console
          - name: HRPROJECTS_MONGODB_PASSWORD
            valueFrom:
              secretKeyRef:
                name: mongodb-credentials
                key: password
```

```
@Value("${hrprojects.mongodb.password}")
private String password;
```

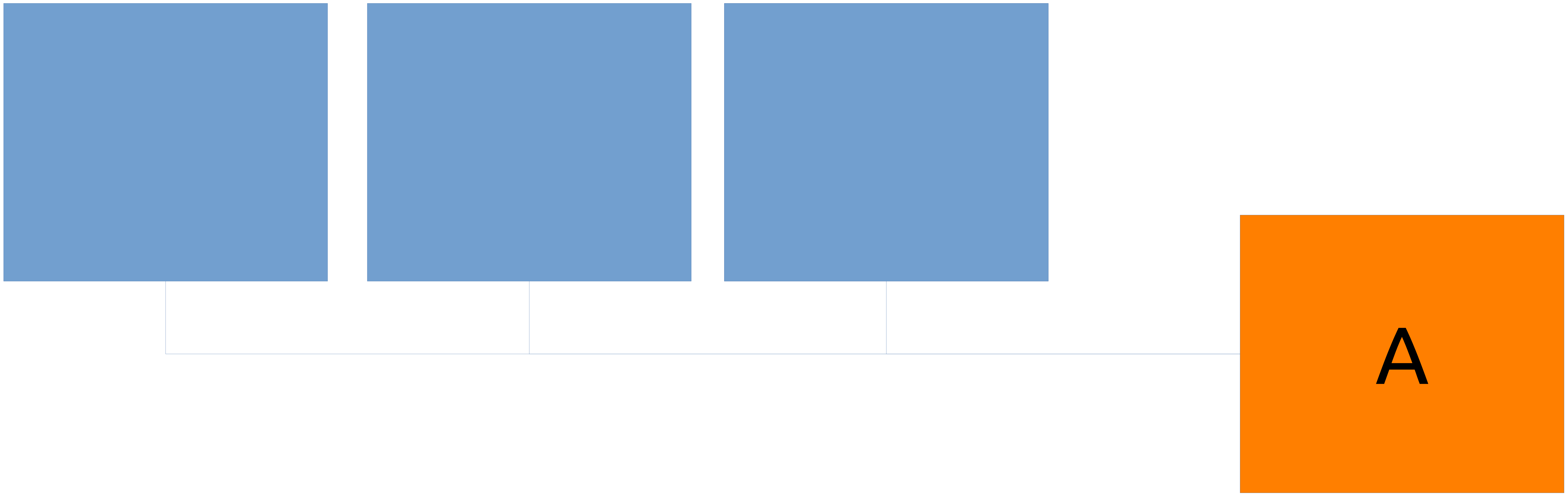
TASK 7

SCALING

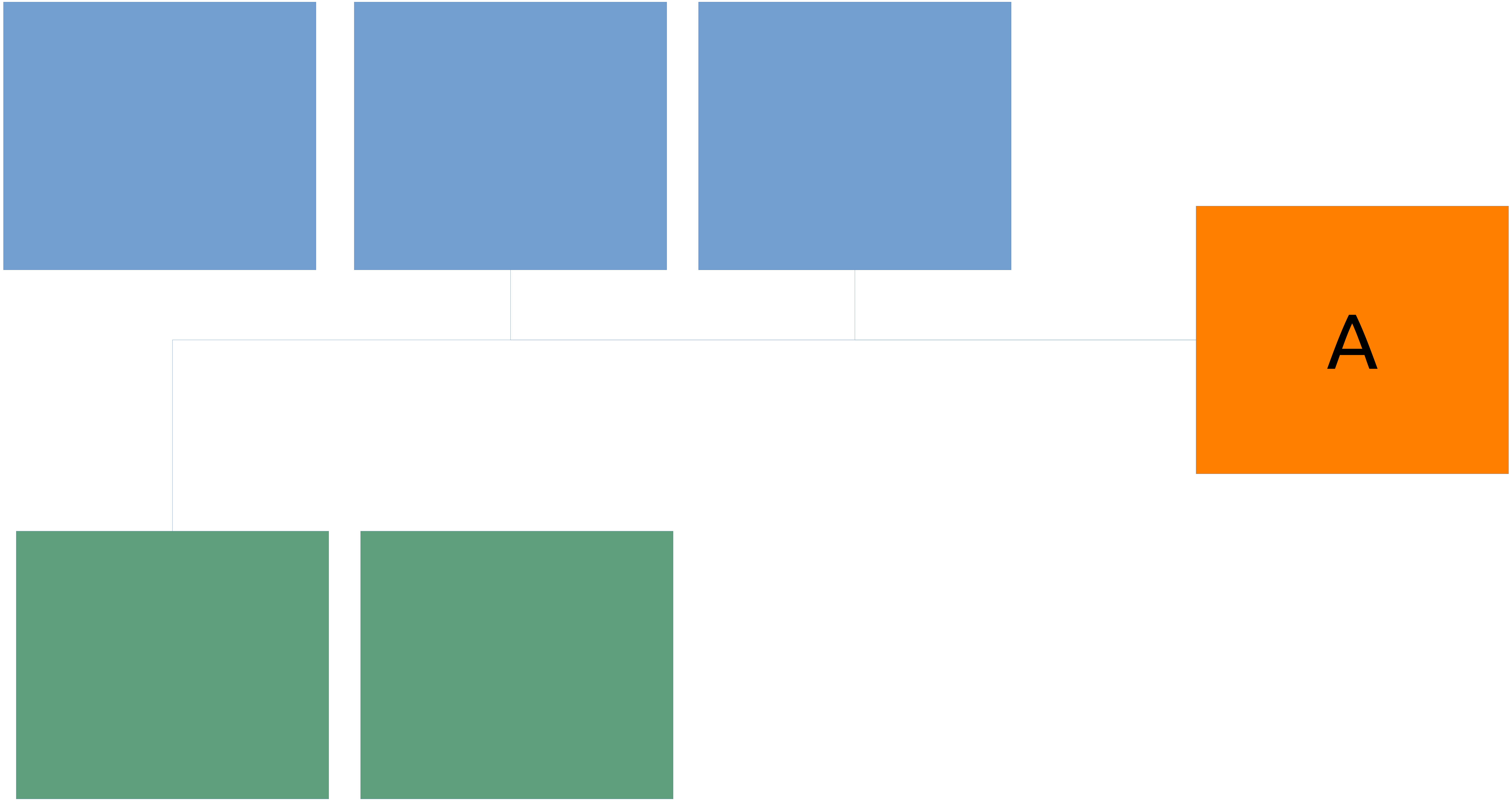
```
$ kubectl scale deployment nginx-deployment --replicas=5

$ kubectl autoscale deployment nginx-deployment --min=10 --max=15 --cpu-percent=80
```

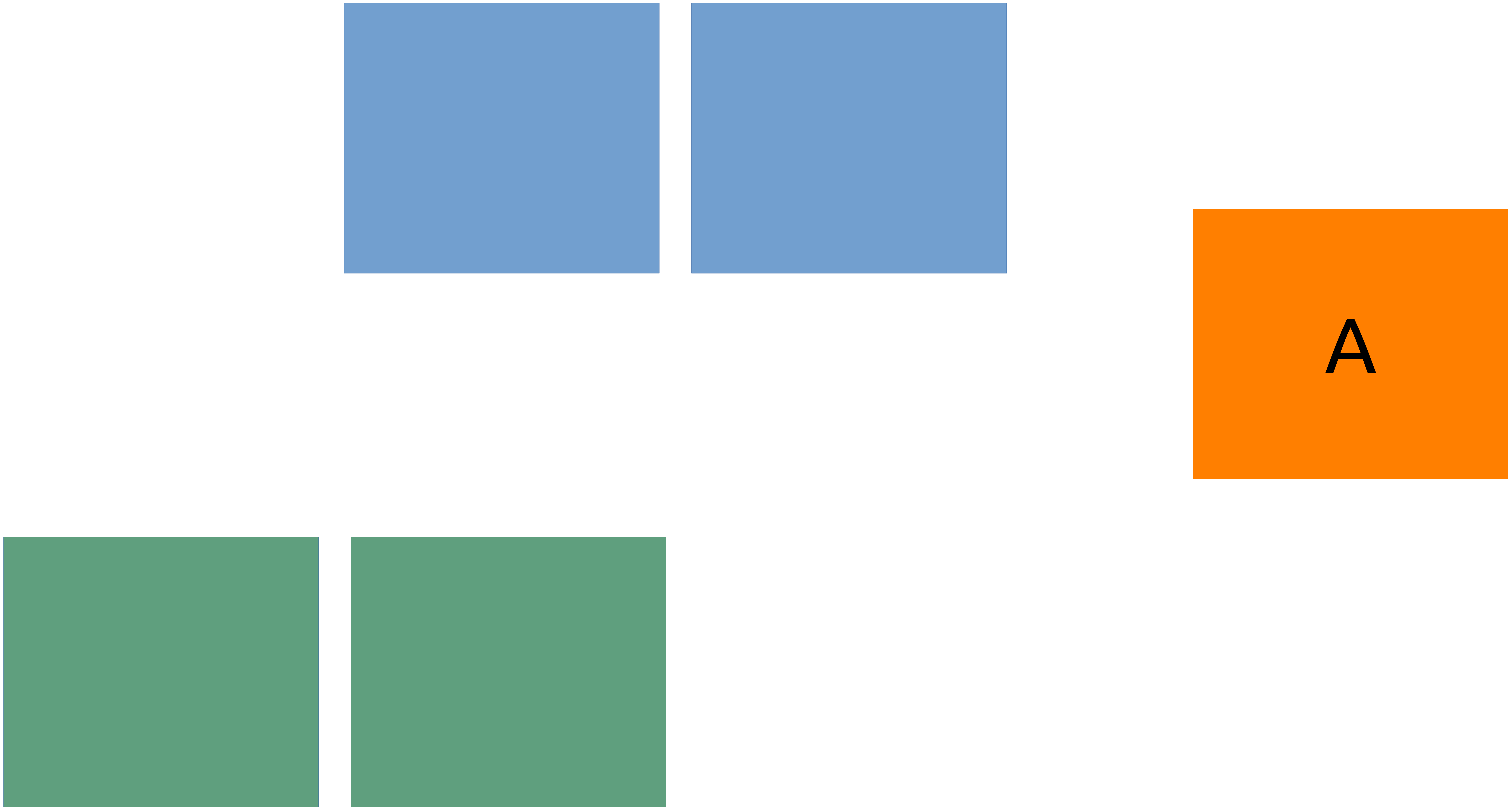
TASK 8

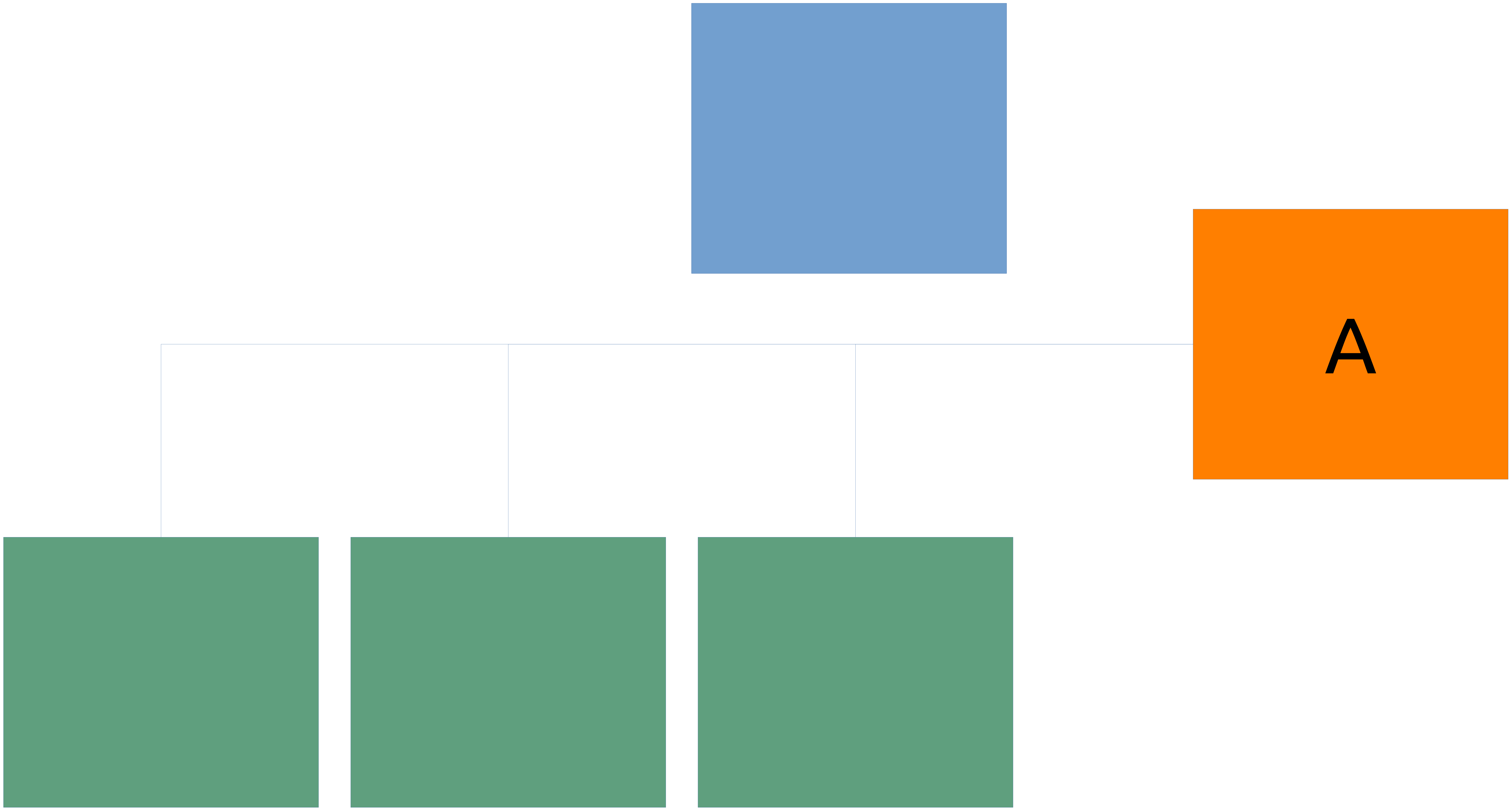
ROLLING UPDATE


A

A


A


A


A

A

```
$ kubectl set image deployment/nginx-deployment nginx=nginx:1.91

$ kubectl rollout status deployments nginx-deployment

$ kubectl rollout history deployment/nginx-deployment

$ kubectl rollout undo deployment/nginx-deployment
```

```
#[Mean    =      907.002, StdDeviation   =      861.077]
#[Max     =     4313.088, Total count    =         7625]
#[Buckets =           27, SubBuckets     =         2048]
----------------------------------------------------------
  7627 requests in 10.01s, 1.58MB read
Requests/sec:    762.30
Transfer/sec:    161.81KB
```

```
#[Mean    =     2866.439, StdDeviation   =     2311.337]
#[Max     =     8552.448, Total count    =         3342]
#[Buckets =           27, SubBuckets     =         2048]
----------------------------------------------------------
  3344 requests in 10.01s, 709.26KB read
  Socket errors: connect 0, read 0, write 4, timeout 111
Requests/sec:    333.96
Transfer/sec:     70.83KB
```

YOU ARE READY!

BUT THERE IS MORE!

VOLUMES

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: graphite-storage-pv
  labels:
    type: local
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 1000Mi
  hostPath:
    path: "/hosthome/dpokusa/tmp/graphite-minikube-storage"
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: graphite-storage-claim
  labels:
    type: local
spec:
  volumeName: graphite-storage-pv
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1000Mi
```

```
spec:
  containers:
    - name: monitoring
      image: xxx/graphite-grafana:0.2.0
      ports:
        - containerPort: 80
          name: grafana
        - containerPort: 81
          name: graphite
        - containerPort: 8125
          name: statsd
        - containerPort: 8126
          name: statsd-admin
        # statsD administrative port: 8126
      volumeMounts:
      - mountPath: /opt/graphite/storage
        name: graphite-storage
      - mountPath: /opt/grafana/storage
        name: grafana-storage
  volumes:
  - name: graphite-storage
    persistentVolumeClaim:
      claimName: graphite-storage-claim
  - name: grafana-storage
    persistentVolumeClaim:
      claimName: grafana-storage-claim
```

JOBS

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
    metadata:
      name: pi
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
      restartPolicy: Never
      backoffLimit: 4
```

PETS
(STATEFUL SETS)

```yaml
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
  spec:
    terminationGracePeriodSeconds: 10
    containers:
    - name: nginx
      image: gcr.io/google_containers/nginx-slim:0.8
      ports:
      - containerPort: 80
        name: web
      volumeMounts:
      - name: www
        mountPath: /usr/share/nginx/html
volumeClaimTemplates:
- metadata:
    name: www
  spec:
    accessModes: [ "ReadWriteOnce" ]
    storageClassName: my-storage-class
    resources:
      requests:
        storage: 1Gi
```

INGRESS

DAEMON SETS

SPRING BOOT K8 INTERGATION

```
<dependency>
      <groupId>io.fabric8</groupId>
      <artifactId>spring-cloud-kubernetes-core</artifactId>
</dependency>
```

RECCOMENDED SOURCES

- kubernetes.io
- *http://blog.arungupta.me*
- *https://github.com/kubernetes/minikube*

ABOUT

ONWELO

WE DELIVER YOUR IDEAS

PAWEŁ MŁYNARCZYK

@psmlynarczyk

ABOUT

ONWELO

WE DELIVER YOUR IDEAS

@dpokusa

DANIEL POKUSA

SOFTWARE-EMPATHY.PL

ABOUT

SPREAD IT

SPREADIT.PL

Wrzesień 2018, Katowice

```
$ minikube stop
$ minikube delete
```