

DEVELOPER INTRO TO



kubernetes



@dpokusa

O CZYM BĘDZIE?

O CZYM BĘDZIE?

- Czym jest i jak działa kubernetes?

O CZYM BĘDZIE?

- Czym jest i jak działa kubernetes?
- Jak zacząć pracować z kubernetesem?

O CZYM BĘDZIE?

- Czym jest i jak działa kubernetes?
- Jak zacząć pracować z kubernetesem?
- Dlaczego i kiedy warto rozważyć kubernetesa?

O CZYM BĘDZIE?

- Czym jest i jak działa kubernetes?
- Jak zacząć pracować z kubernetesem?
- Dlaczego i kiedy warto rozważyć kubernetesa?
- Codzienna praca z kubernetesem z perspektywy programisty

O CZYM BĘDZIE?

- Czym jest i jak działa kubernetes?
- Jak zacząć pracować z kubernetesem?
- Dlaczego i kiedy warto rozważyć kubernetesa?
- Codzienna praca z kubernetesem z perspektywy programisty
- Garść doświadczeń

O CZYM **NIE** BĘDZIE?

O CZYM **NIE** BĘDZIE?

- Jak zarządzać produkcyjnym środowiskiem*

O CZYM **NIE** BĘDZIE?

- Jak zarządzać produkcyjnym środowiskiem*
- Jak wygląda codzienna praca od strony OPSa*

O CZYM **NIE** BĘDZIE?

- Jak zarządzać produkcyjnym środowiskiem*
- Jak wygląda codzienna praca od strony OPSa*
- Dyskusji o tym czy dockeryzować bazy danych czy nie :)

O CZYM **NIE** BĘDZIE?

- Jak zarządzać produkcyjnym środowiskiem*
- Jak wygląda codzienna praca od strony OPSa*
- Dyskusji o tym czy dockeryzować bazy danych czy nie :)
- Dyskusji o tym dlaczego **nie** dokeryzować Kafki

O CZYM **NIE** BĘDZIE?

- Jak zarządzać produkcyjnym środowiskiem*
- Jak wygląda codzienna praca od strony OPSa*
- Dyskusji o tym czy dockeryzować bazy danych czy nie :)
- Dyskusji o tym dlaczego **nie** dokeryzować Kafki
- Szczegółów Ingressa i LB*



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)



The Illustrated Children's Guide to Kubernetes

80 891 wyświetleń

👍 2 TYS. 💬 18 ➦ UDOSTĘPNIJ ≡ ⋮



The Illustrated Children's Guide to Kubernetes

80 891 wyświetleń

👍 2 TYS. 💬 18 ➦ UDOSTĘPNIJ 📺 ...

DESIRED STATE MANAGEMENT



Aisle 4
Ethnic Foods Cereal
Flour / Sugar Breakfast On-The-Go
Baking Mixes / Oils Pancake Mix / Syrup
Pie Crusts Spices / Nuts

PICK ONE UP!
It's easy to bake
some "bunny happy"
cakes.

isco Pie Crusts

Nabisco Pie Crusts

EXTRA SAVINGS!
\$2.57

EXTRA SAVINGS!
\$2.55

EXTRA SAVINGS!
\$2.17

EXTRA SAVINGS!
\$2.57

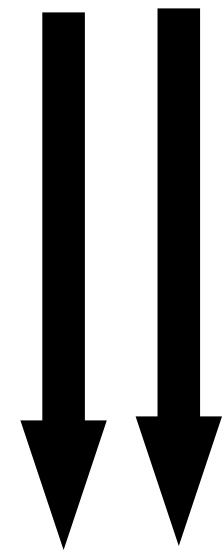


TOBACZ
KUCHA

SKLEP NIECZYNNY
Z POWODU, ŻE
JEST ZAMKNIĘTY

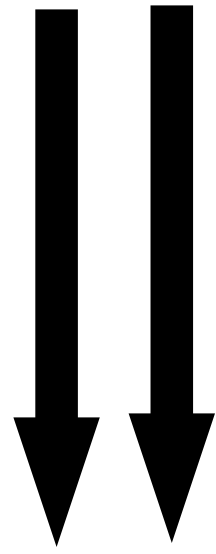
TOBACZ
KUCHA

PRZE SWE TLAMY
1974
NA CYZERNIE
KUMINTA



API

MASTER
K8s Cluster Services



API

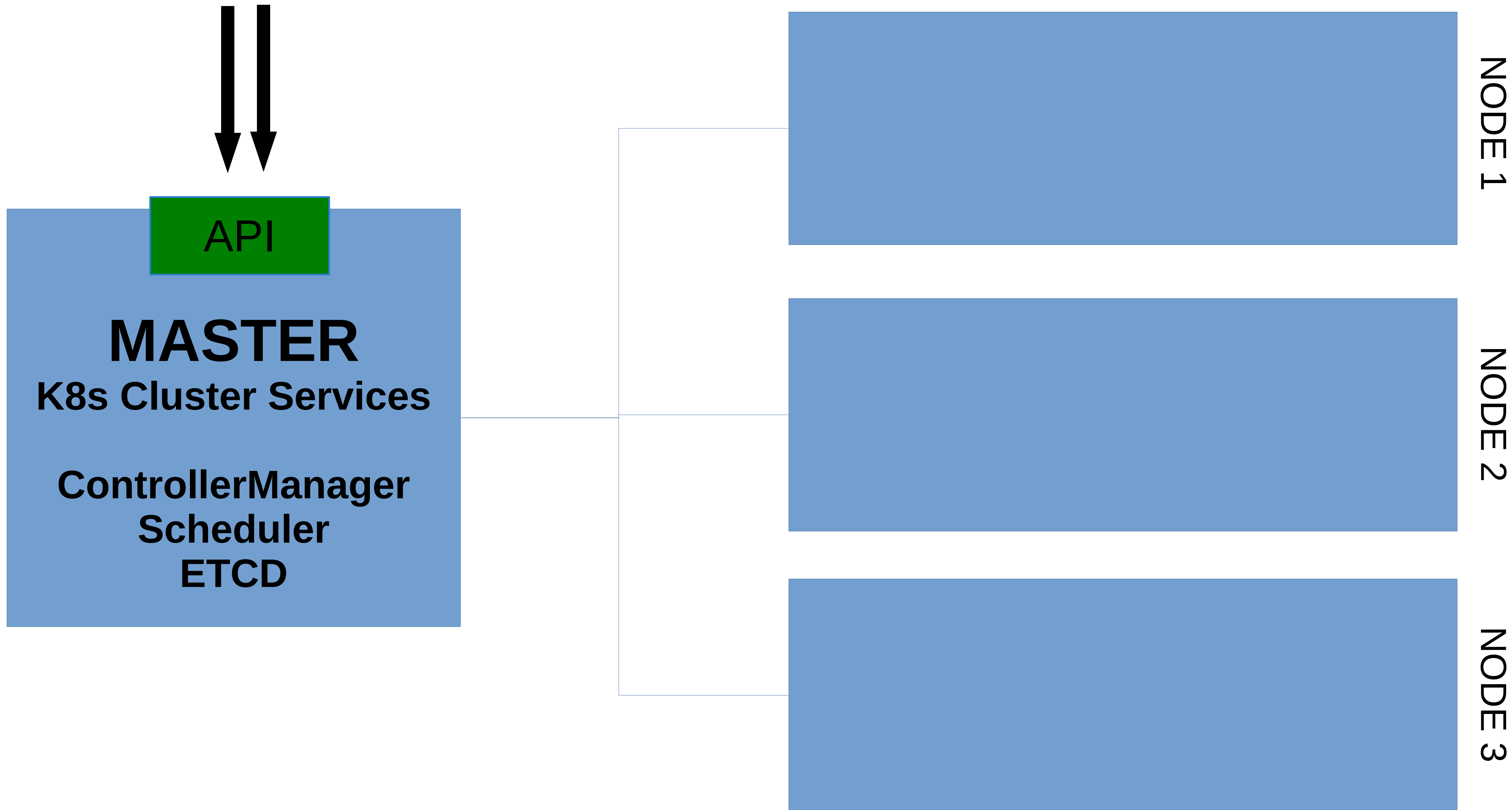
MASTER

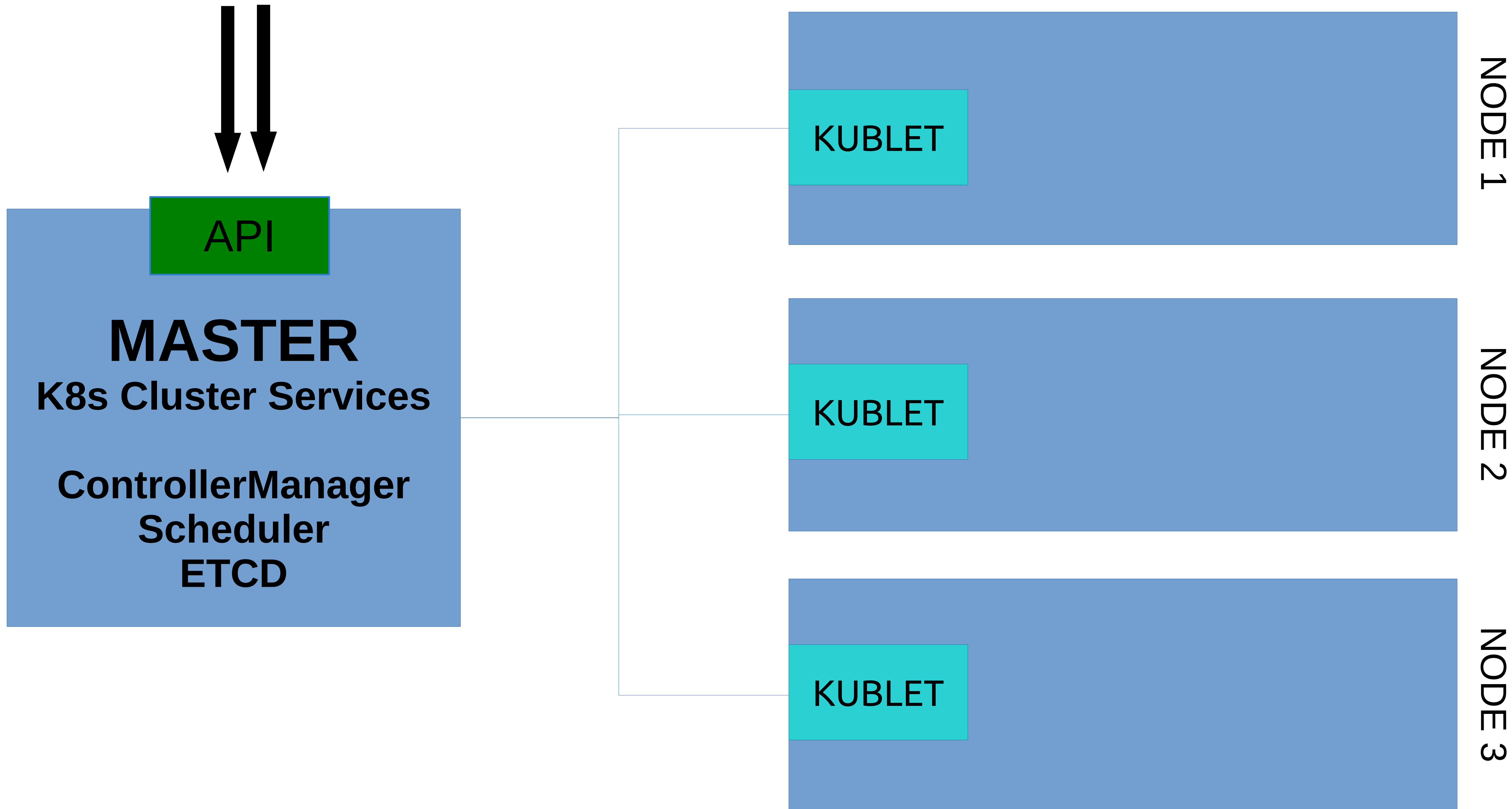
K8s Cluster Services

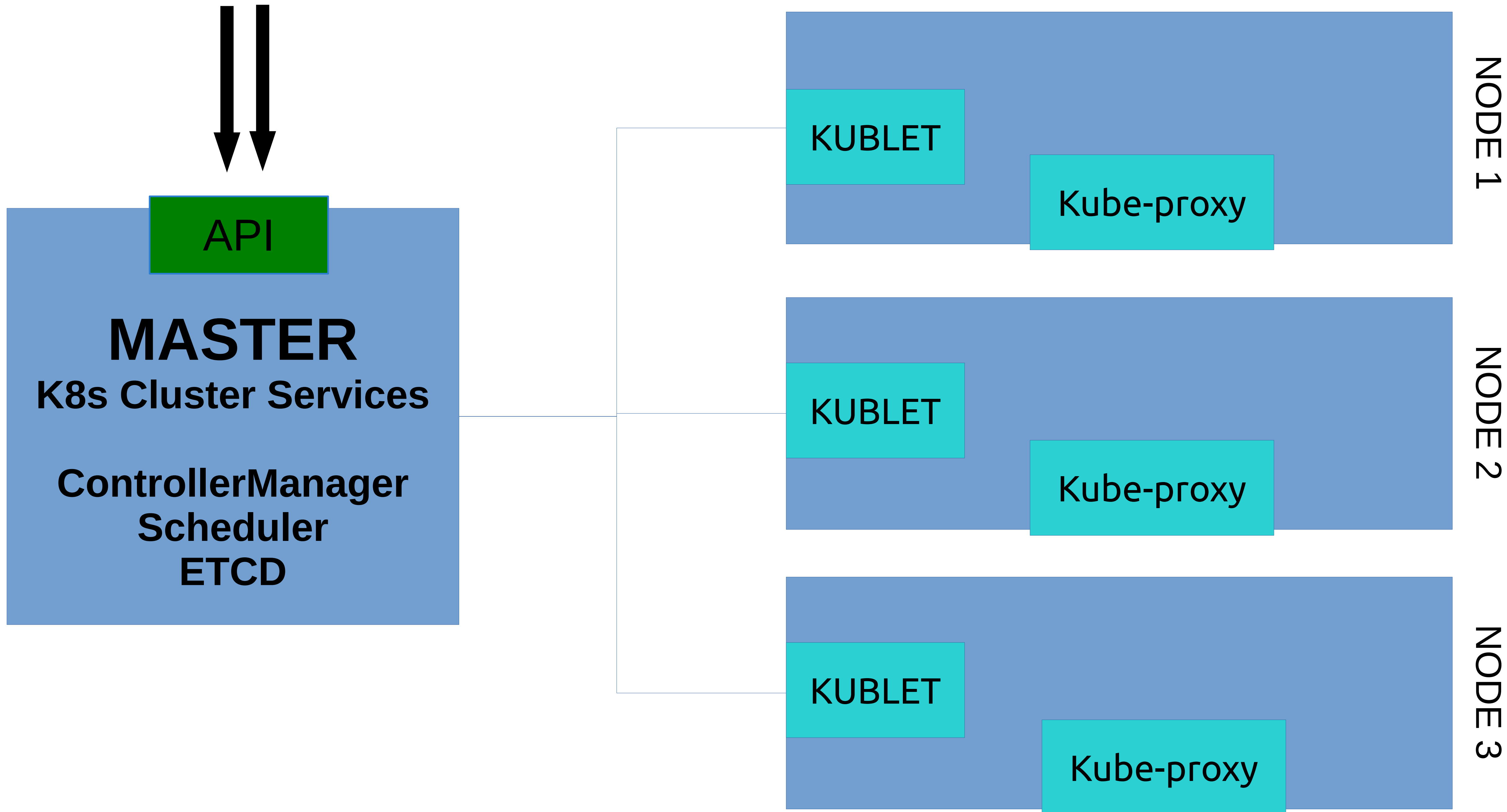
ControllerManager

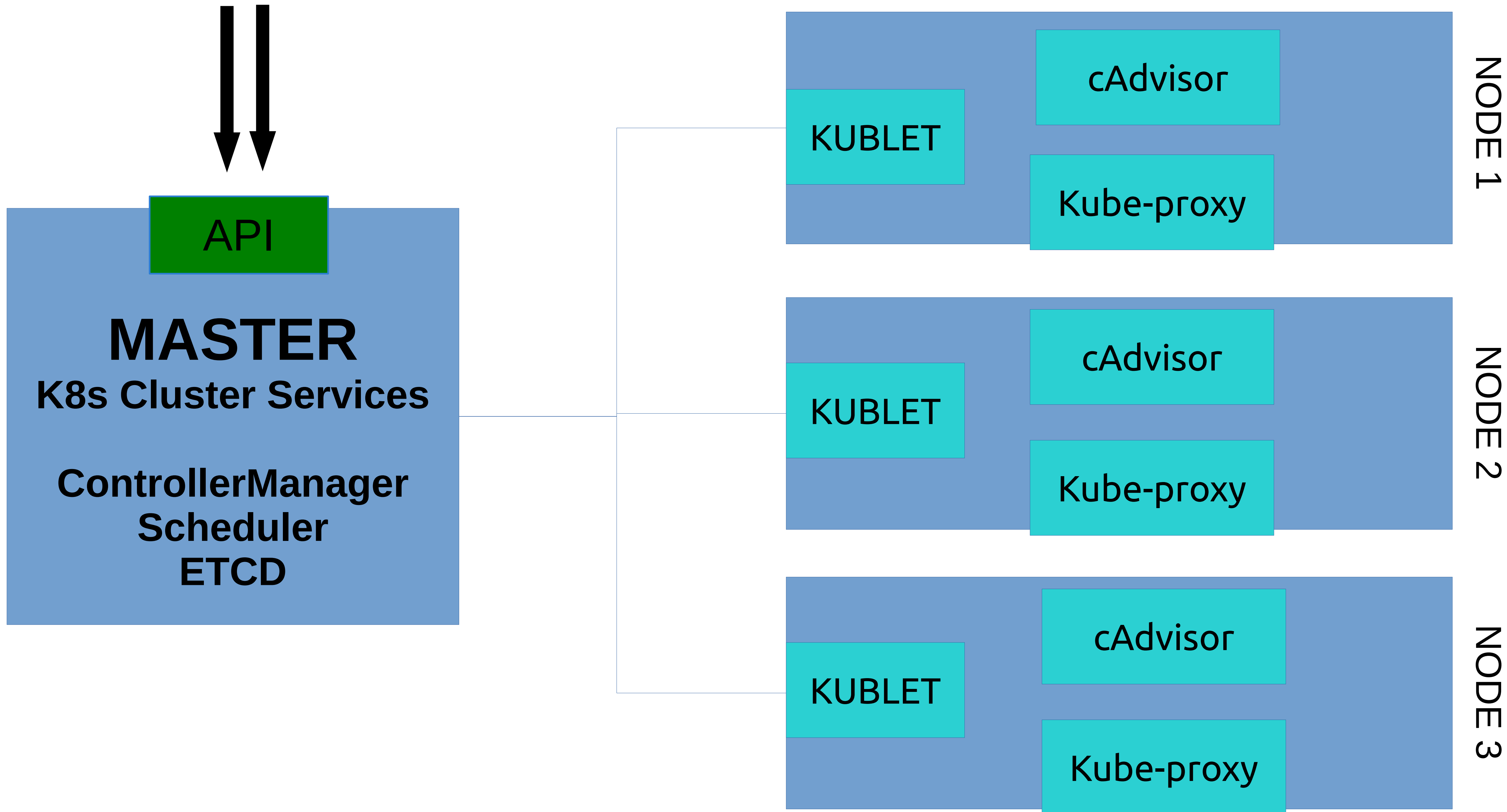
Scheduler

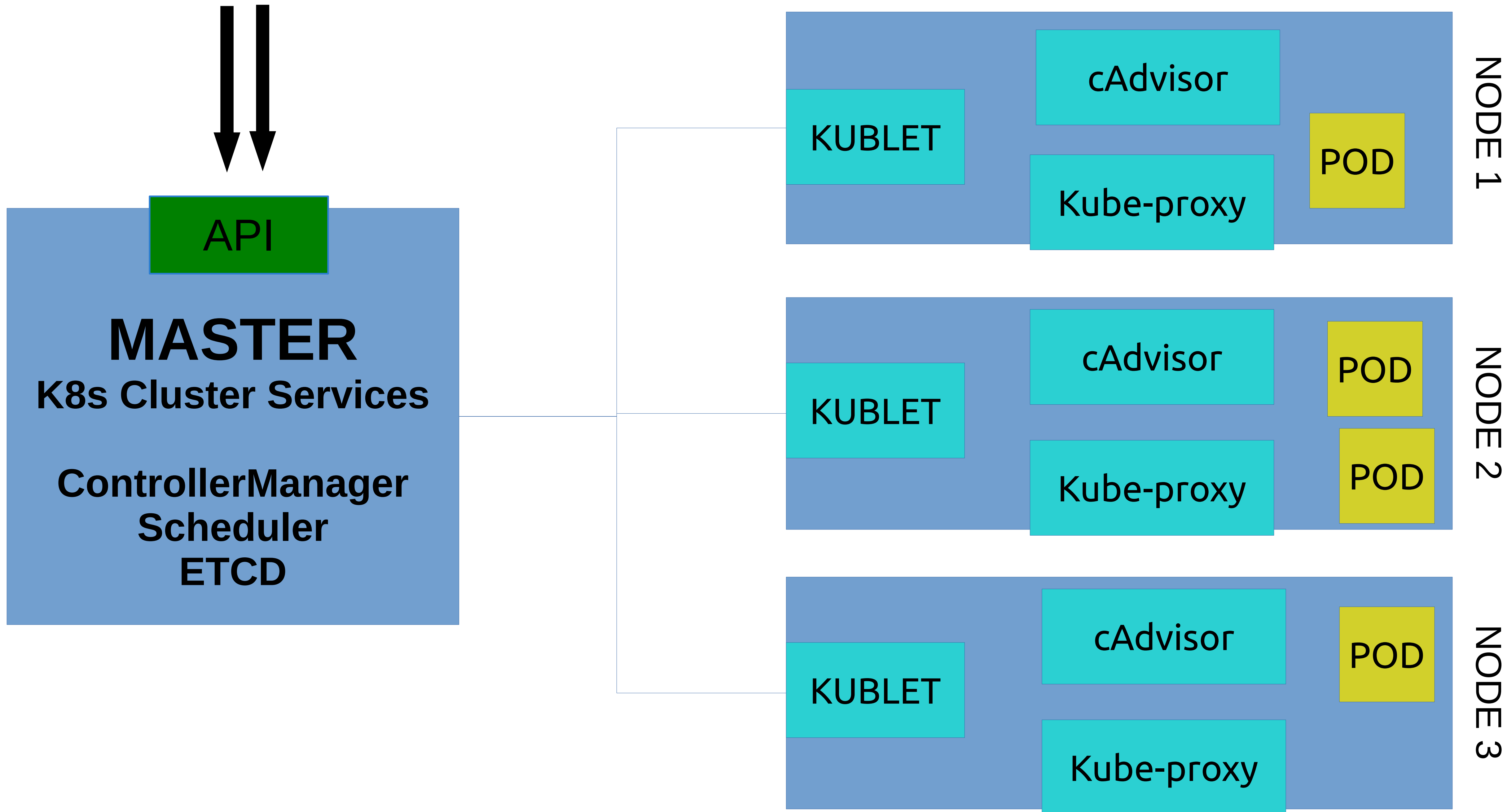
ETCD



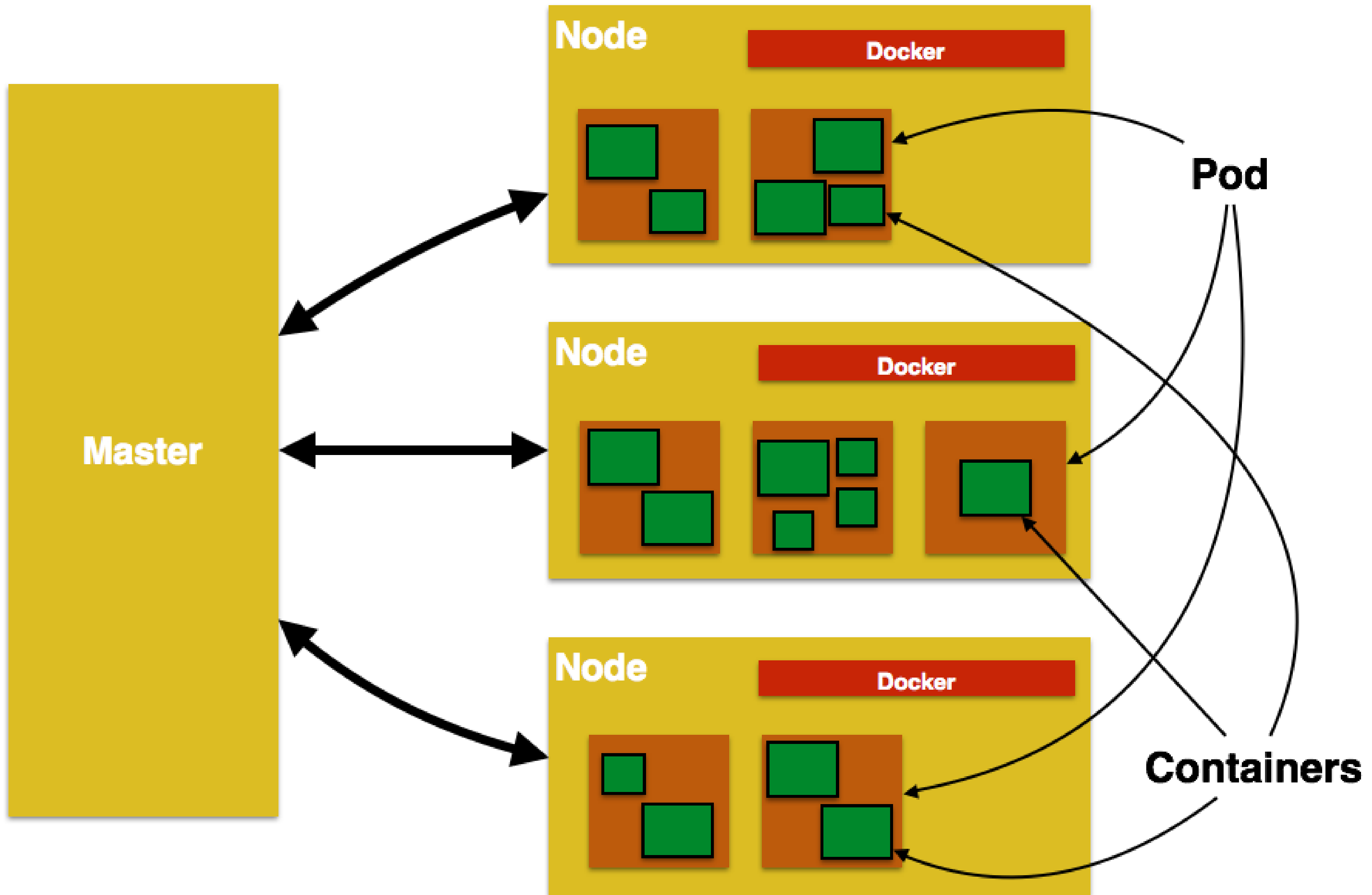








POD



```
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

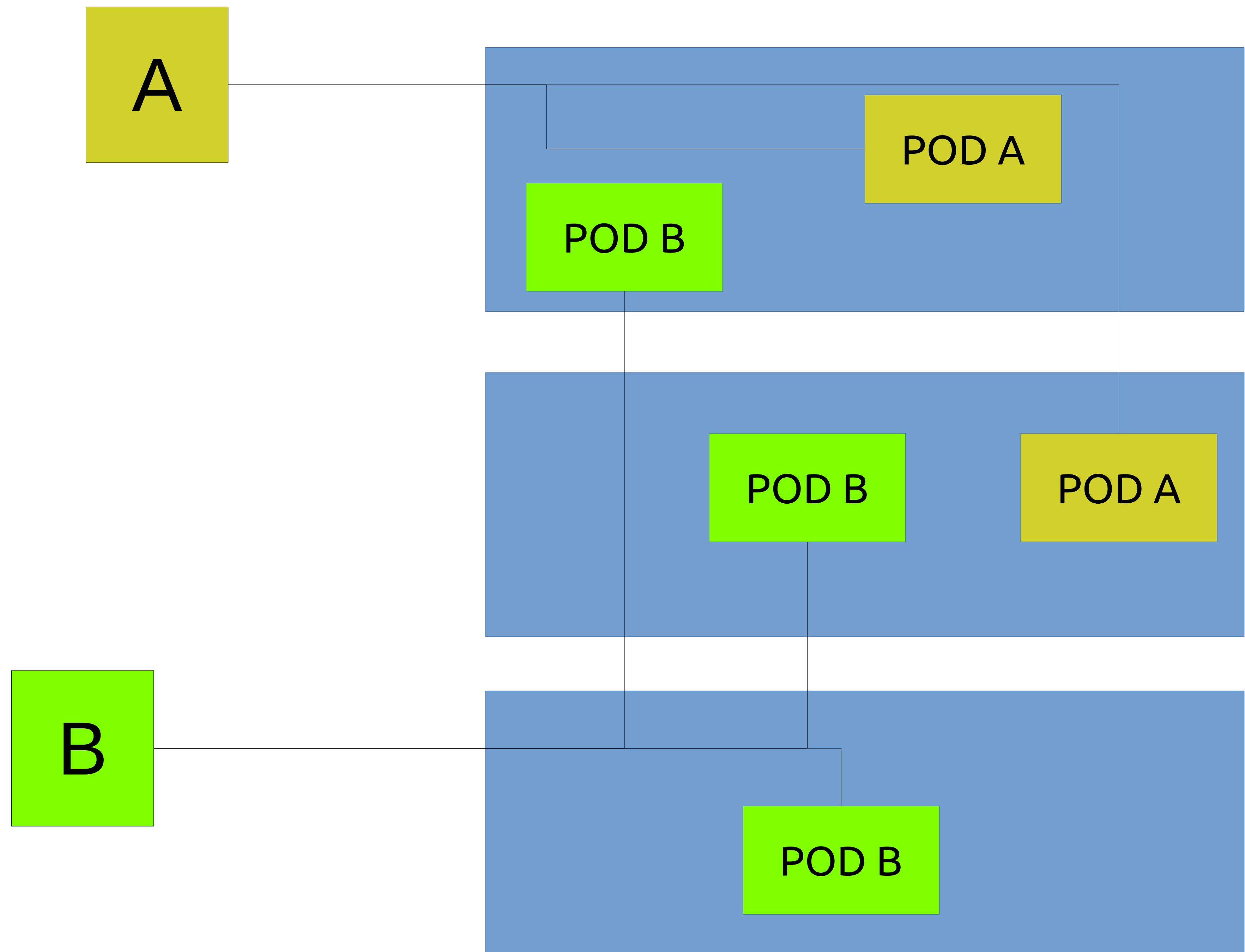


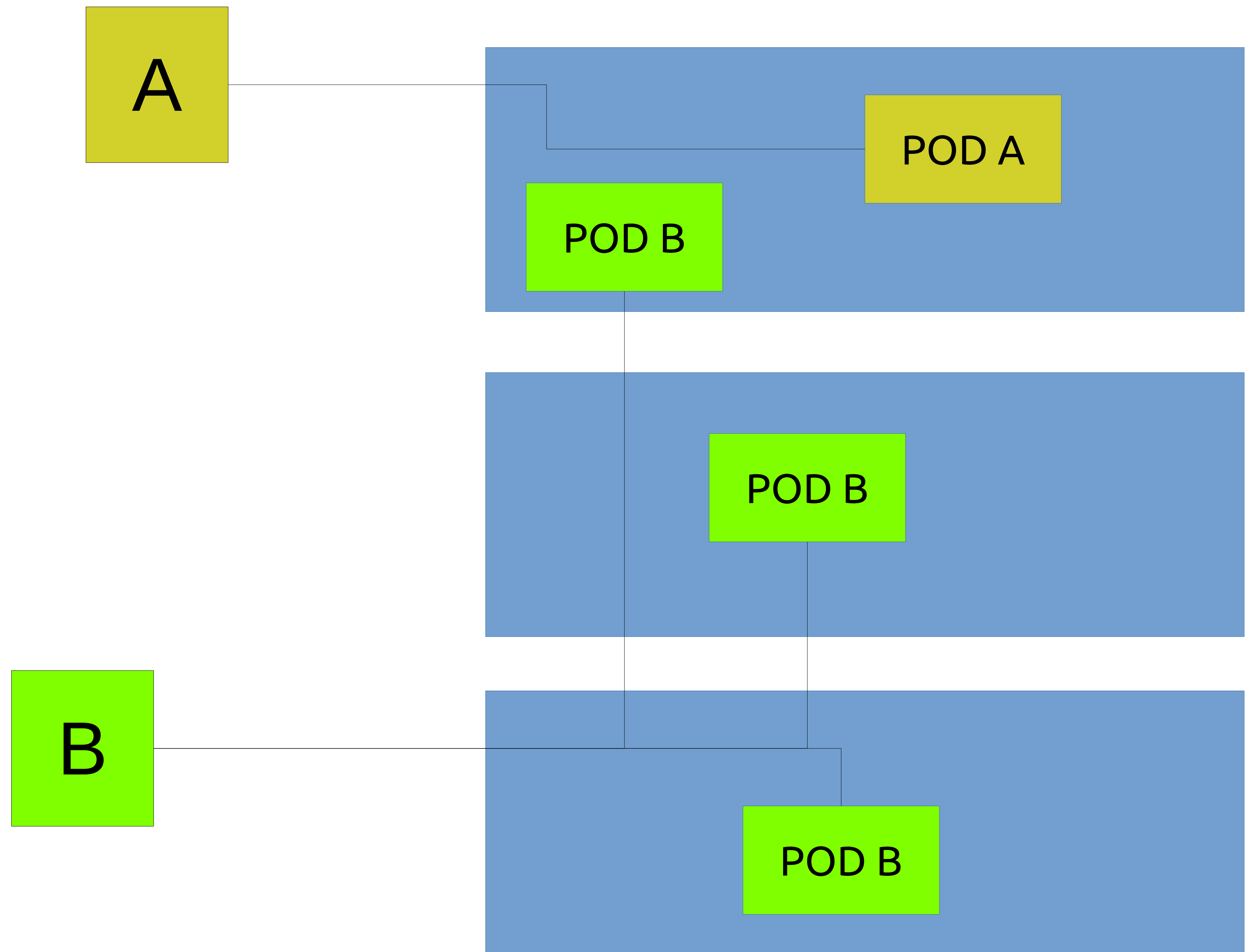


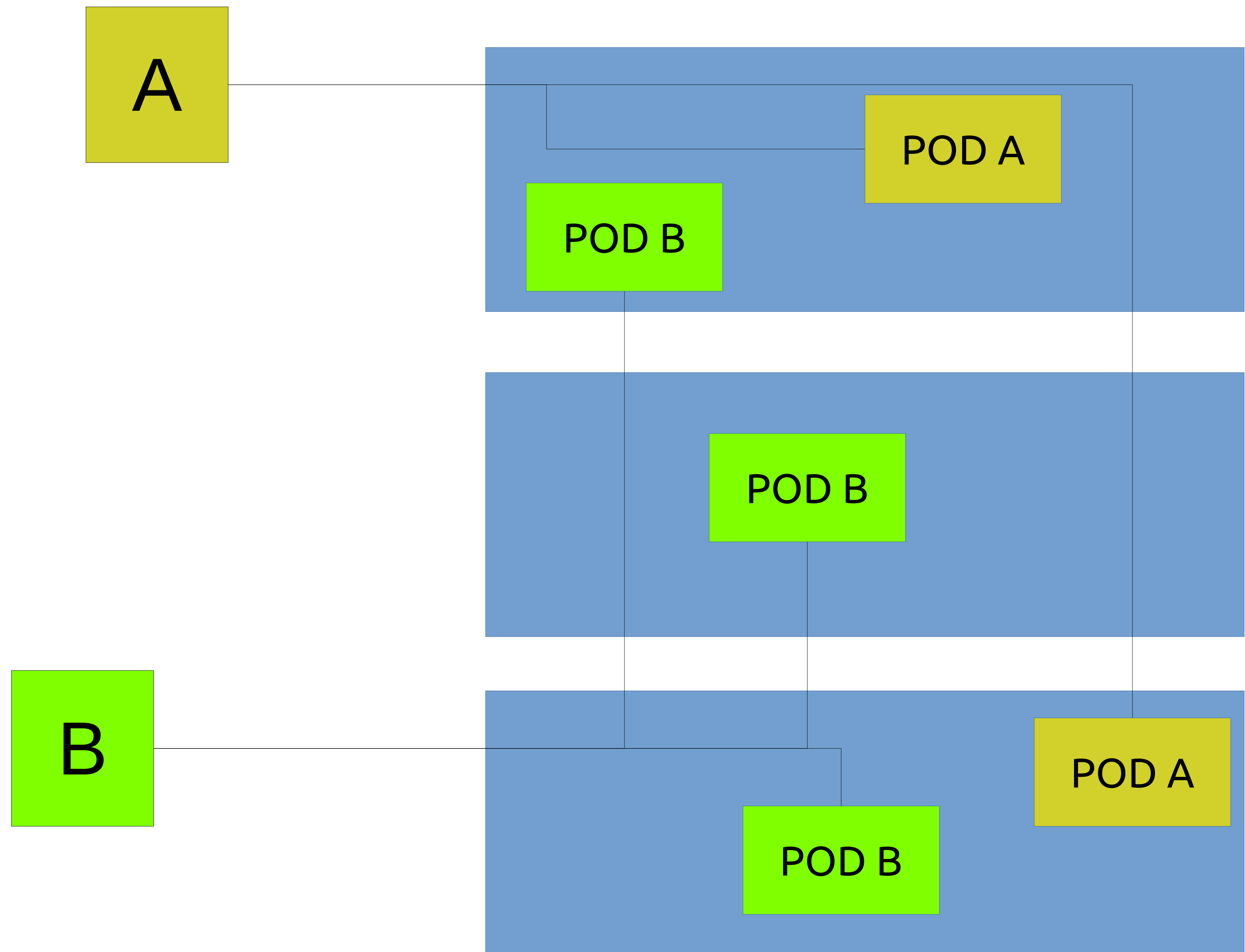
SERVICE

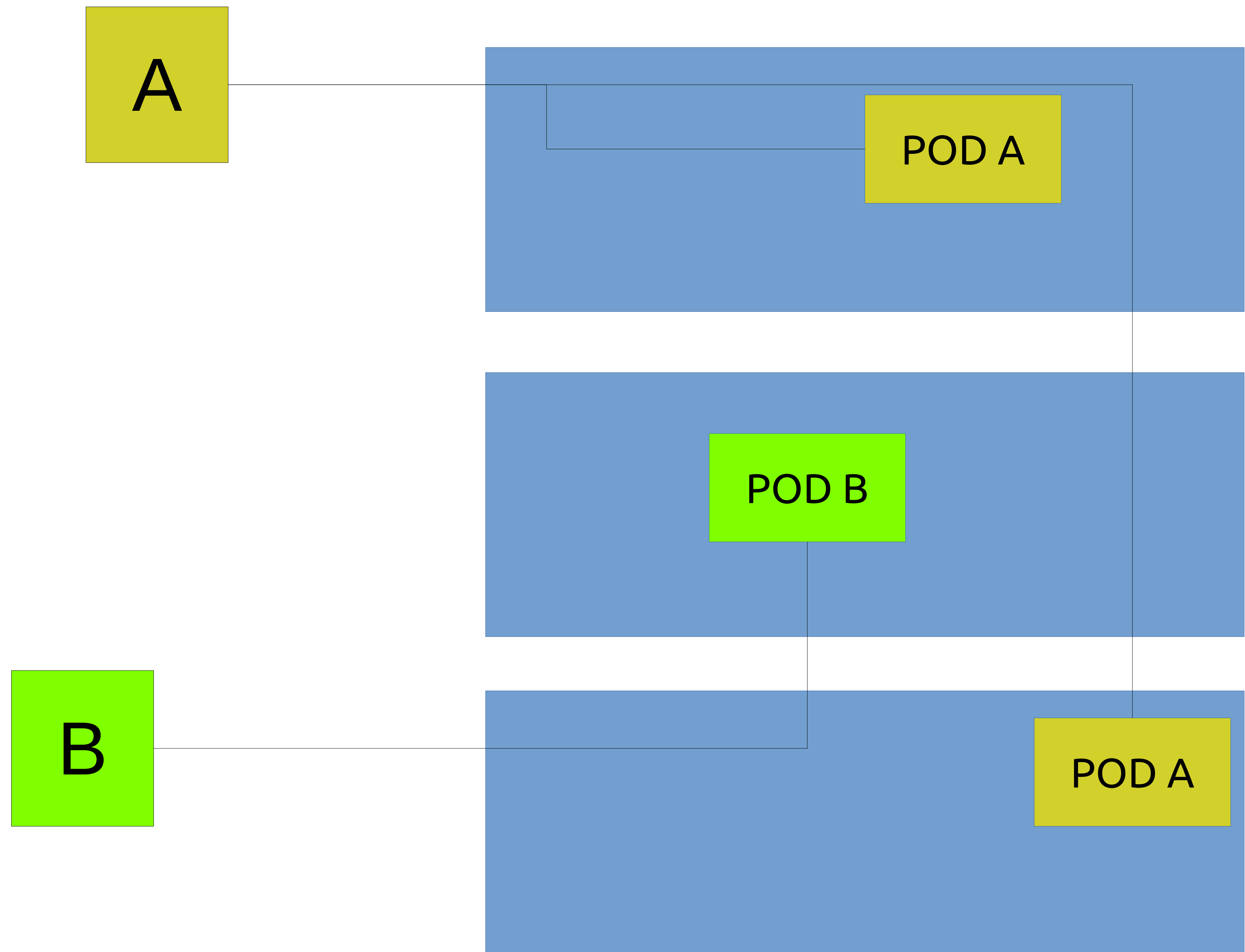

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```



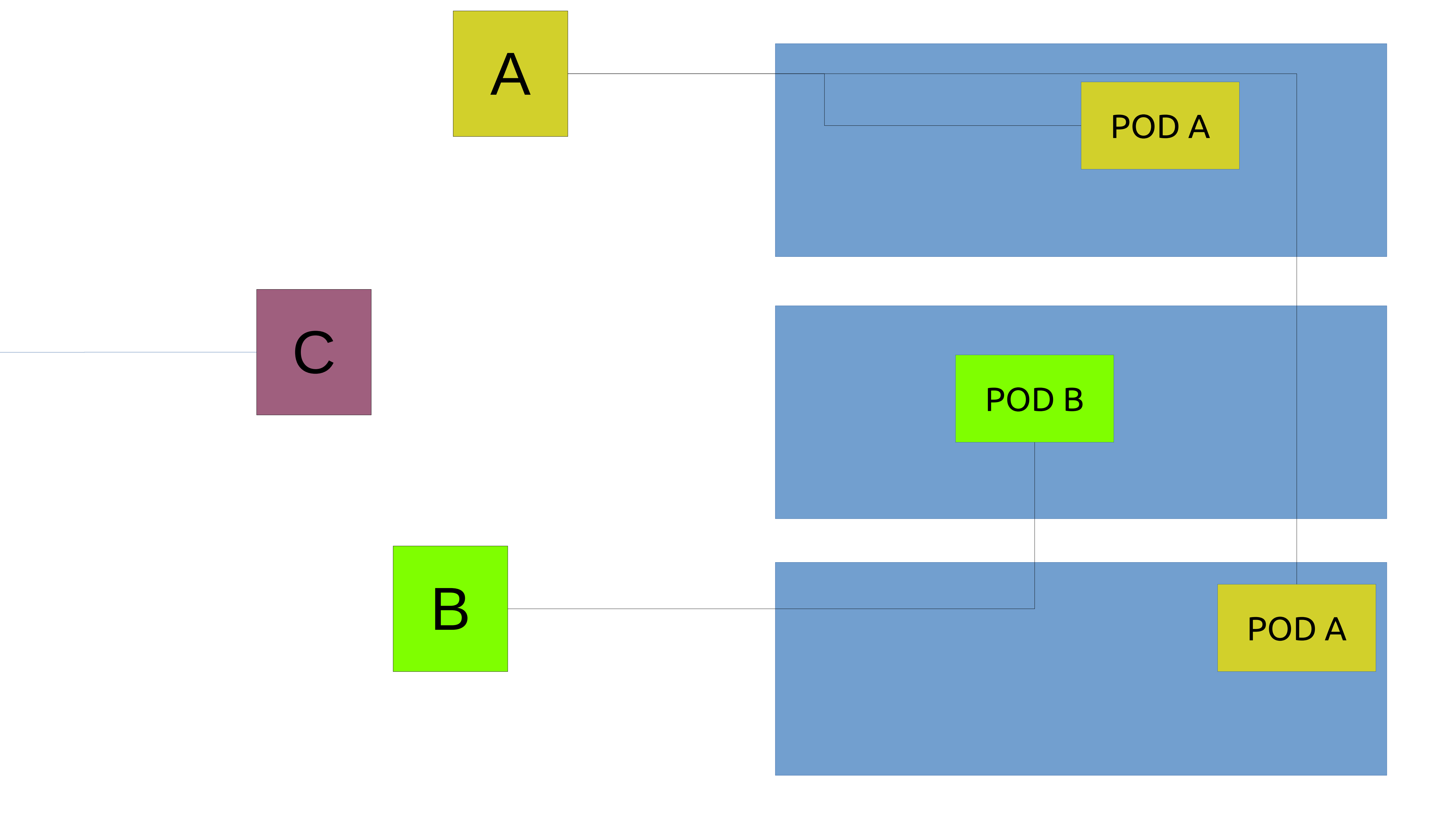


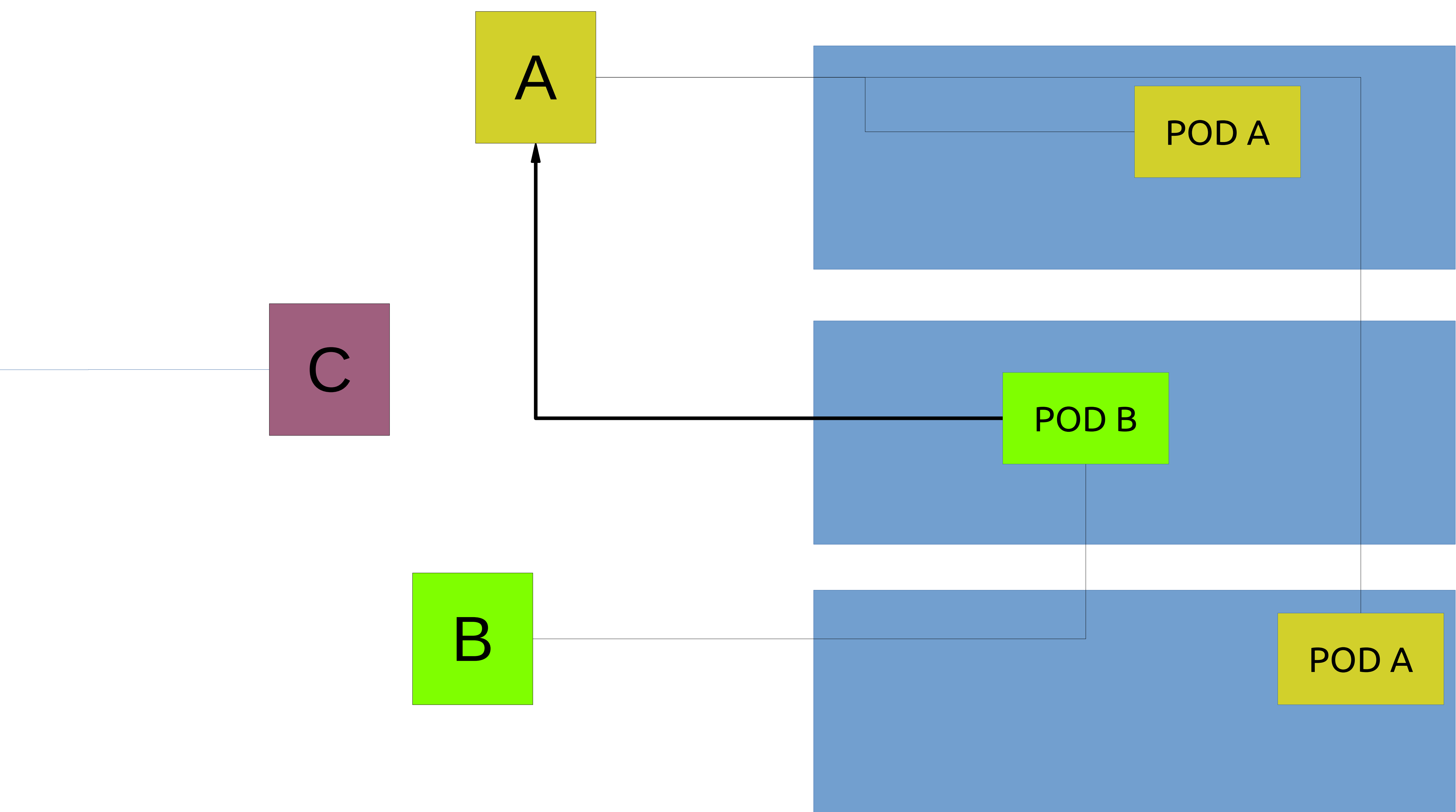


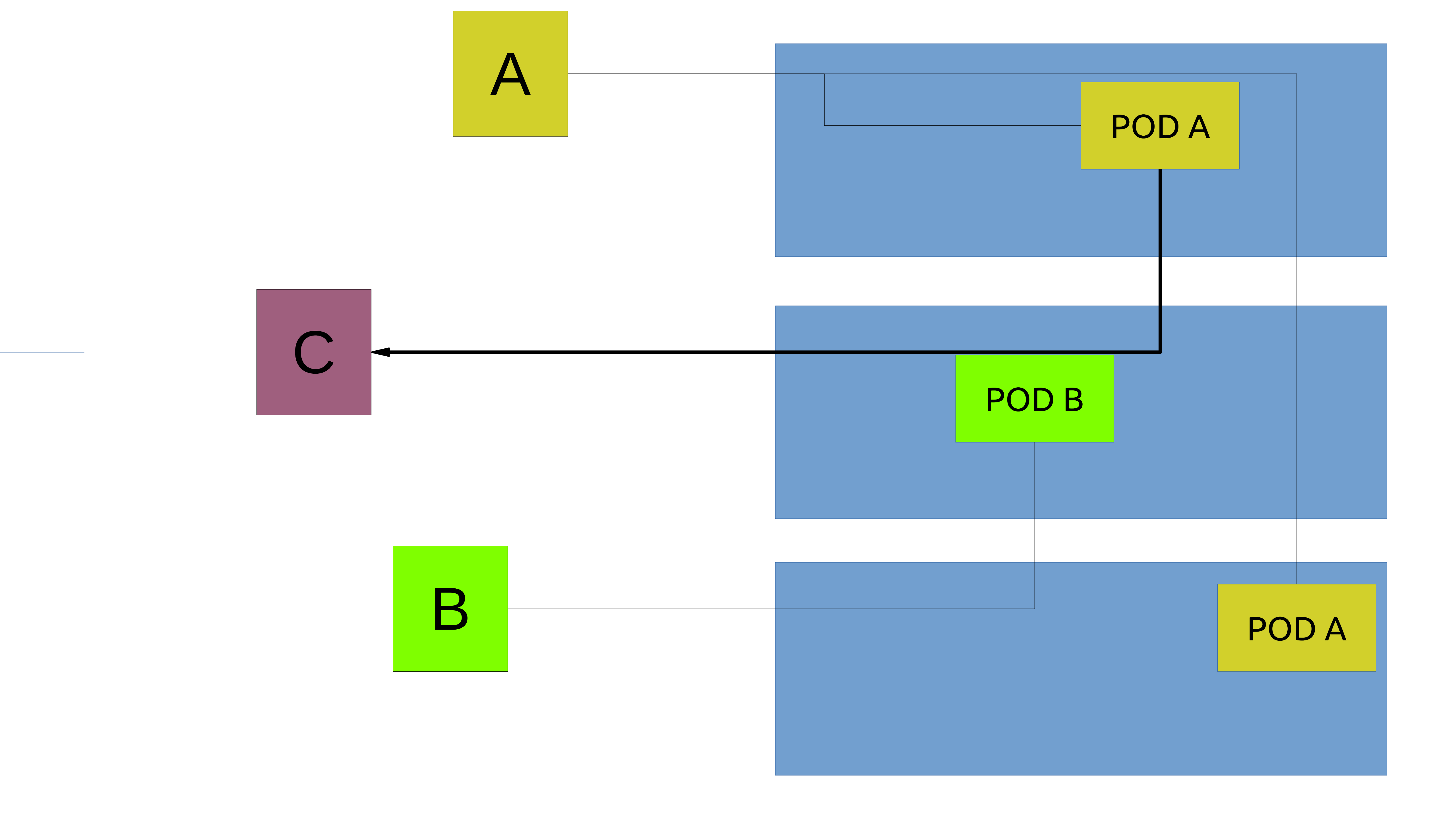




```
kind: Service
apiVersion: v1
metadata:
  name: my-service
  namespace: prod
spec:
  type: ExternalName
  externalName: my.database.example.com
```







LET'S START!



minikube

<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)

```
→ minikube start
Starting local Kubernetes v1.7.5 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Starting cluster components...
Kubectl is now configured to use the cluster.
```

talks/dev-intro-to-kubernetes/kubernetes on  master **took 1m 2s**

```
eval $(minikube docker-env)
```


DASHBOARD

- Cluster
- Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes
- Namespace
- default
- Overview
- Workloads
- Daemon Sets

Deployments

Jobs

Pods

Replica Sets



Replication Controllers





Stateful Sets
- Discovery and Load Balancing
- Ingresses

Services
- Config and Storage
- Config Maps

Persistent Volume Claims

Secrets

| Services | | | | | | |
|--|---|------------|--|--------------------|----------|---|
| Name | Labels | Cluster IP | Internal endpoints | External endpoints | Age | |
|  kubernetes | <div>component: apiserver</div> <div>provider: kubernetes</div> | 10.0.0.1 | kubernetes:443 TCP kubernetes:0 TCP | - | 13 hours |  |

| Persistent Volume Claims | | | | |
|--|---------------------|------------------------|----------|---|
| Name | Volume | Labels | Age | |
|  graphite-storage-claim | graphite-storage-pv | <div>type: local</div> | 12 hours |  |
|  grafana-storage-claim | grafana-storage-pv | <div>type: local</div> | 12 hours |  |

| Secrets | | |
|-------------------------------------|----------|--|
| Name | Age | |
| default-token-zhtsb | 13 hours | |

SERVICE DISCOVERY

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

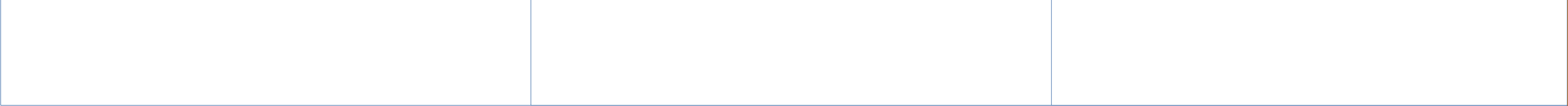
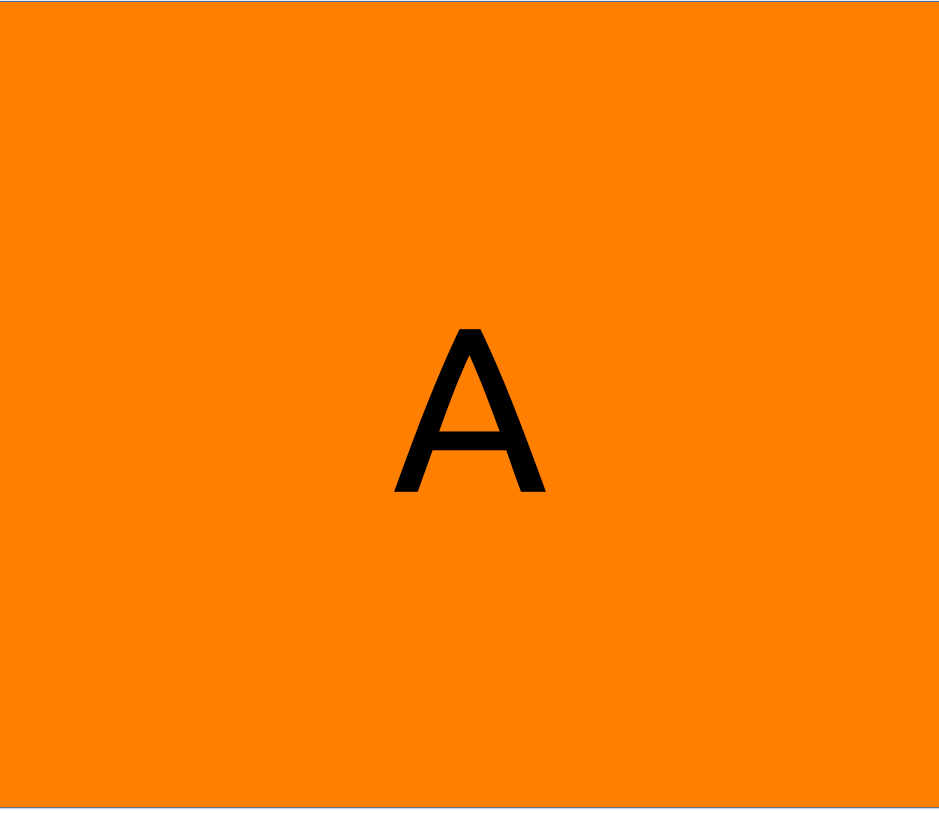
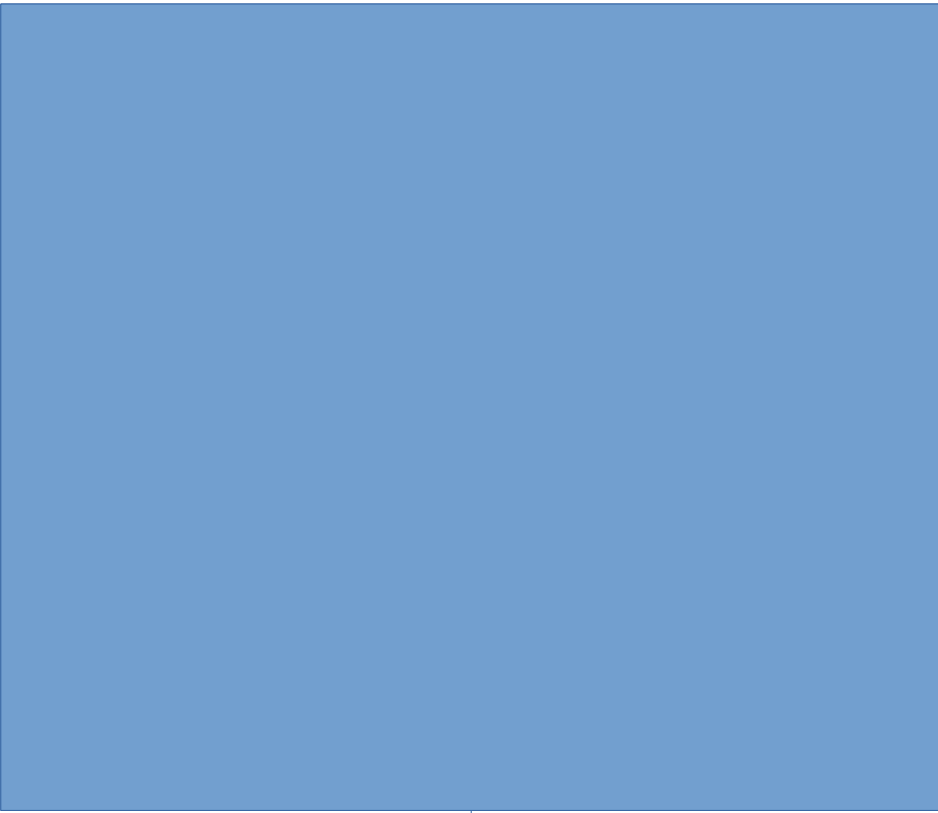
DEPLOYMENT

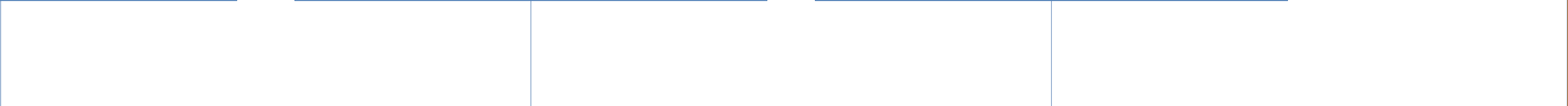
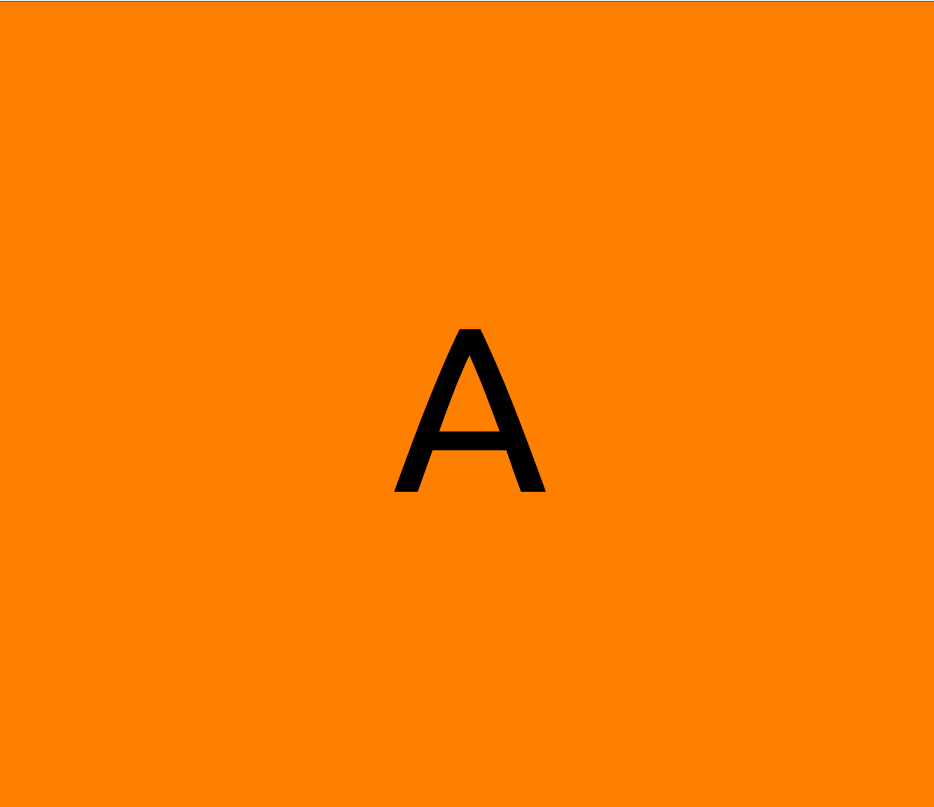
```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

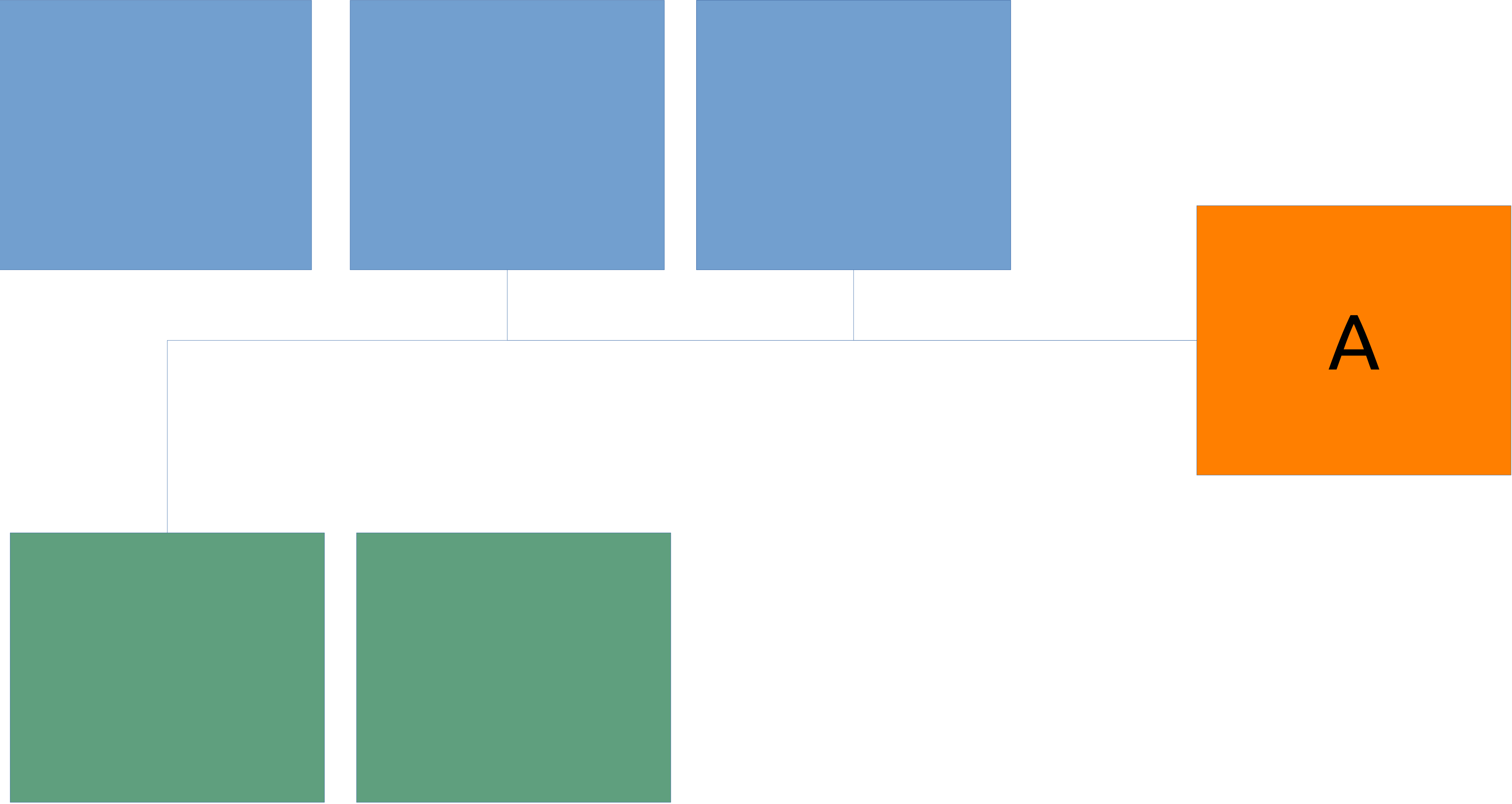
SAY KUBERNETES

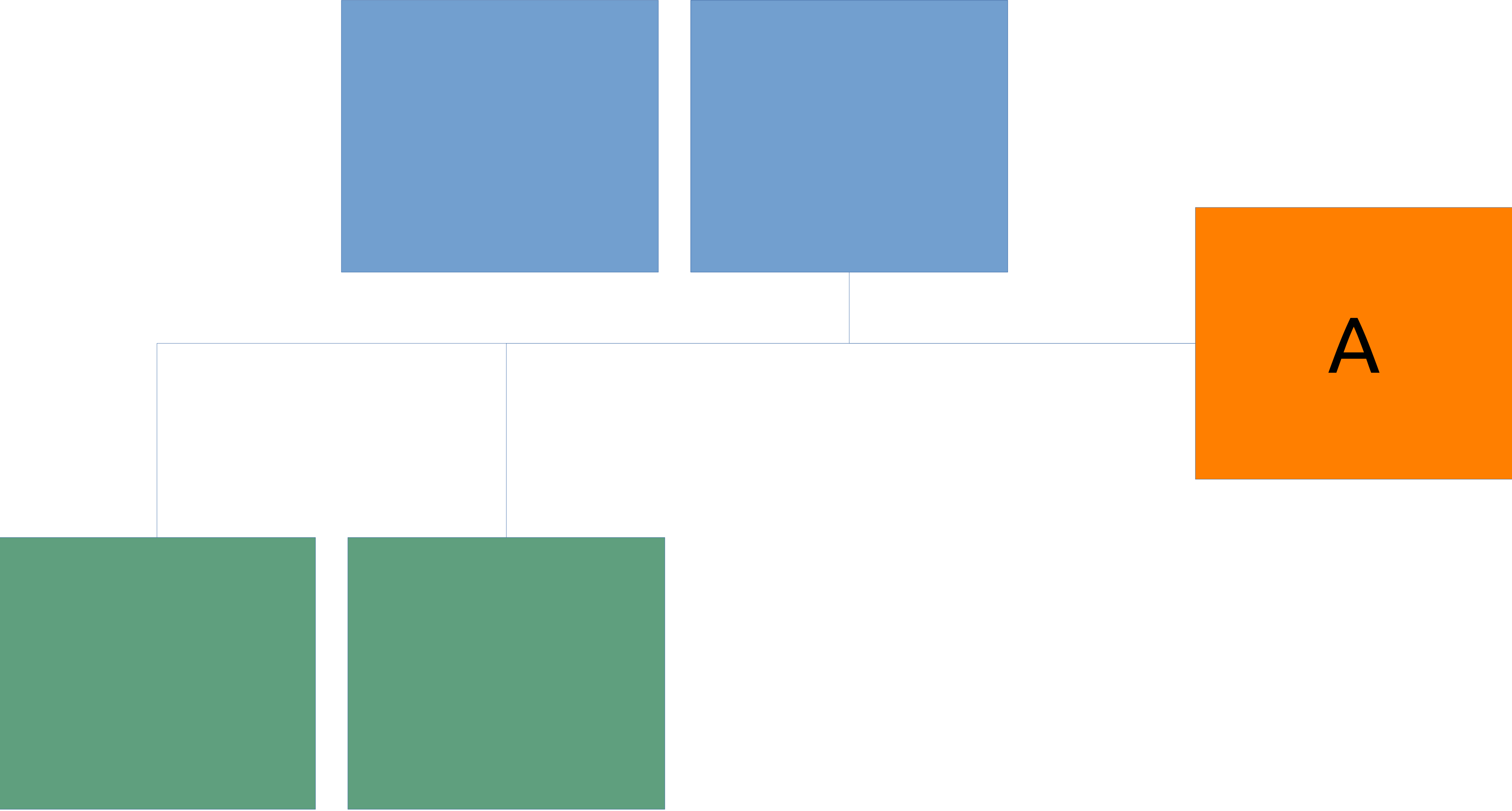
ONE MORE TIME

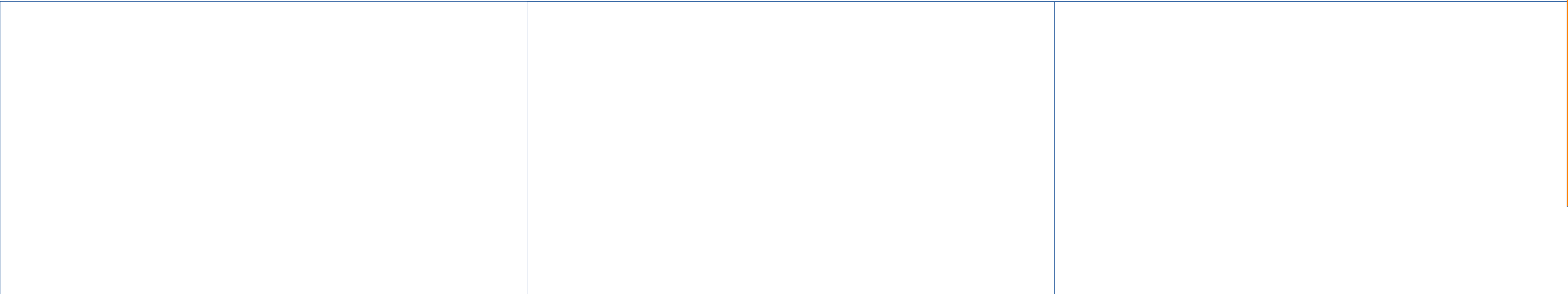
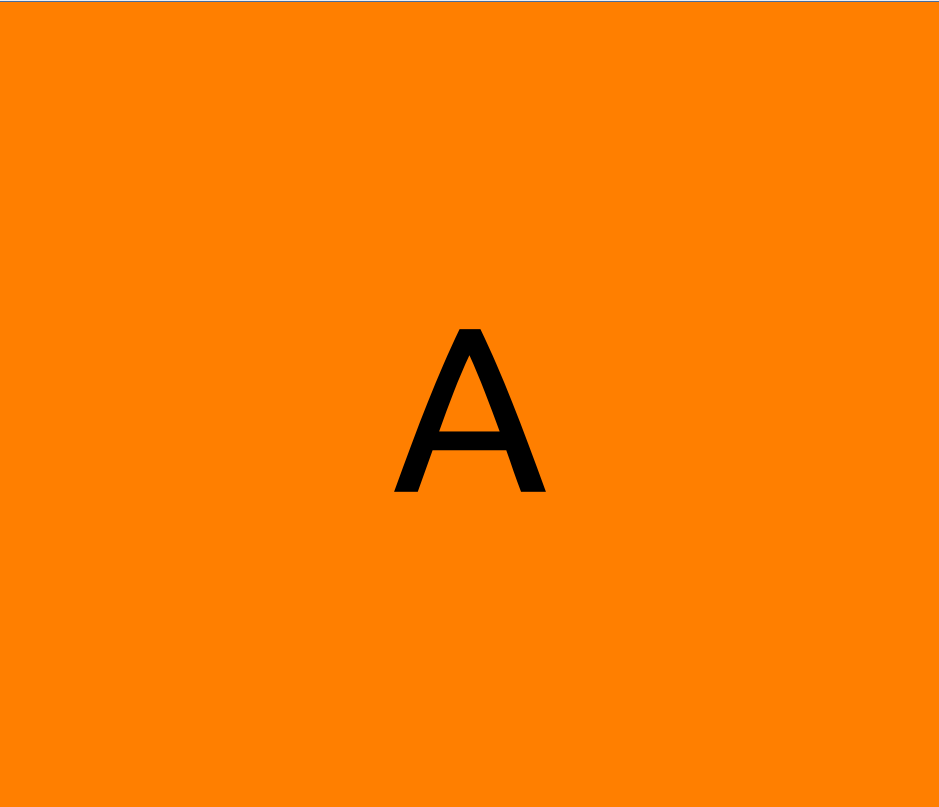
ROLLING UPDATE

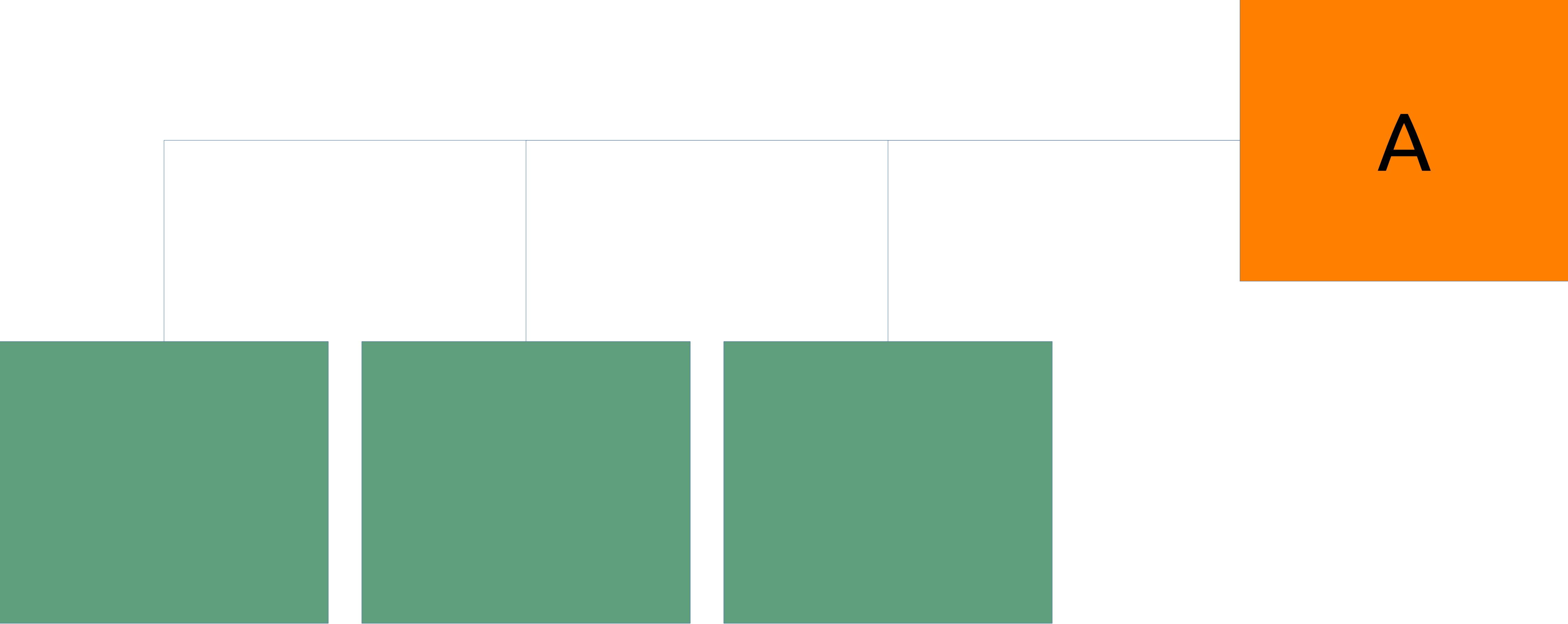












```
$ kubectl set image deployment/nginx-deployment nginx=nginx:1.91
```

```
$ 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
```

SCALING

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

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

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

@Value("\${hrprojects.mongodb.password}")

private String password;

- name: LOG_APPENDER

value: Console

- name: HRPROJECTS_MONGODB_PASSWORD

valueFrom:

secretKeyRef:

name: mongodb-credentials

key: password

YOU ARE READY!

DOCKER EVERYWHERE



KUBERNETES EVERYWHERE

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
```

CONFIG MAPS

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

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]

```
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

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



SOFTWARE-EMPATHY.PL



@dpokusa

ABOUT



SPREADIT.PL

18.11.2017



@dpokusa

ABOUT



@dpokusa

```
$ minikube stop
```

Q&A

```
$ minikube delete
```