Managing Objects with Labels, Annotations, and Namespaces

Anthony E. Nocentino
ENTERPRISE ARCHITECT @ CENTINO SYSTEMS

@nocentino www.centinosystems.com

Course Overview

Using the Kubernetes API

Managing Objects with Labels, Annotations, and Namespaces

Running and Managing Pods

Overview

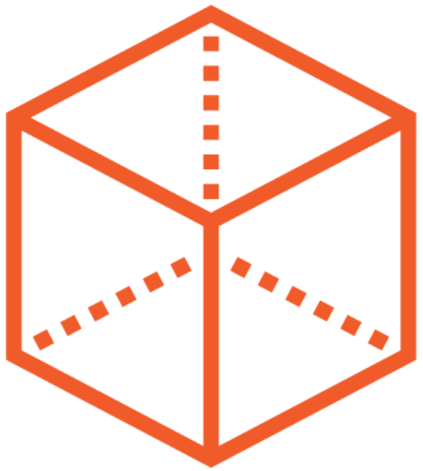
Organizing Objects in Kubernetes

- Namespaces
- Labels
- Annotations

How Kubernetes uses Labels

- Services
- Deployments
- Scheduling

Organizing Objects in Kubernetes

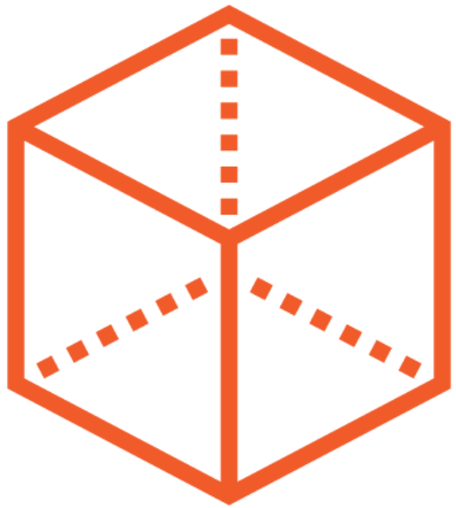


Namespaces

Labels

Annotations

Namespaces



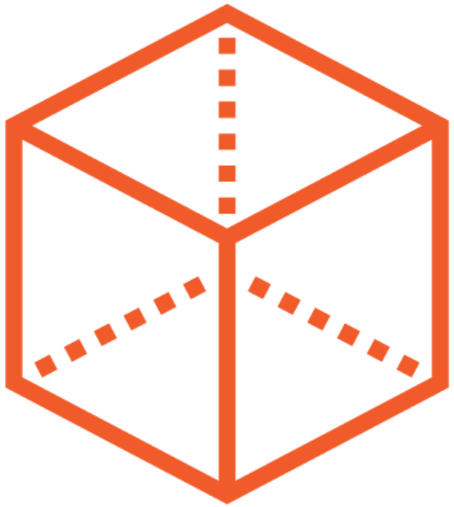
Ability to subdivide a cluster and its resources

Conceptually a “virtual cluster”

Deploy objects into a Namespace

Resource isolation/organization

Namespaces (con’t)



Security boundary for Role-based Access Controls

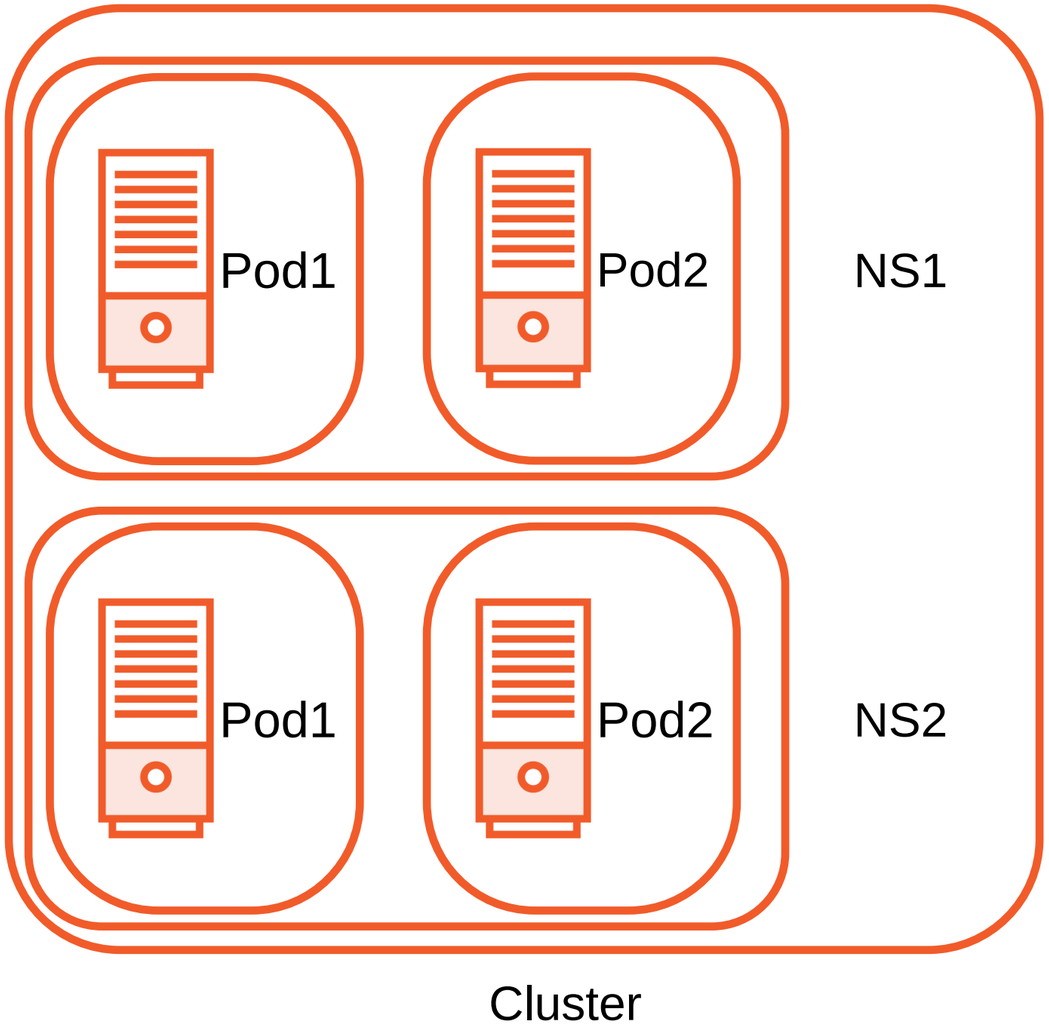
Naming boundary

A resource can be in only one namespace

Has nothing to do with the concept of a Linux namespace

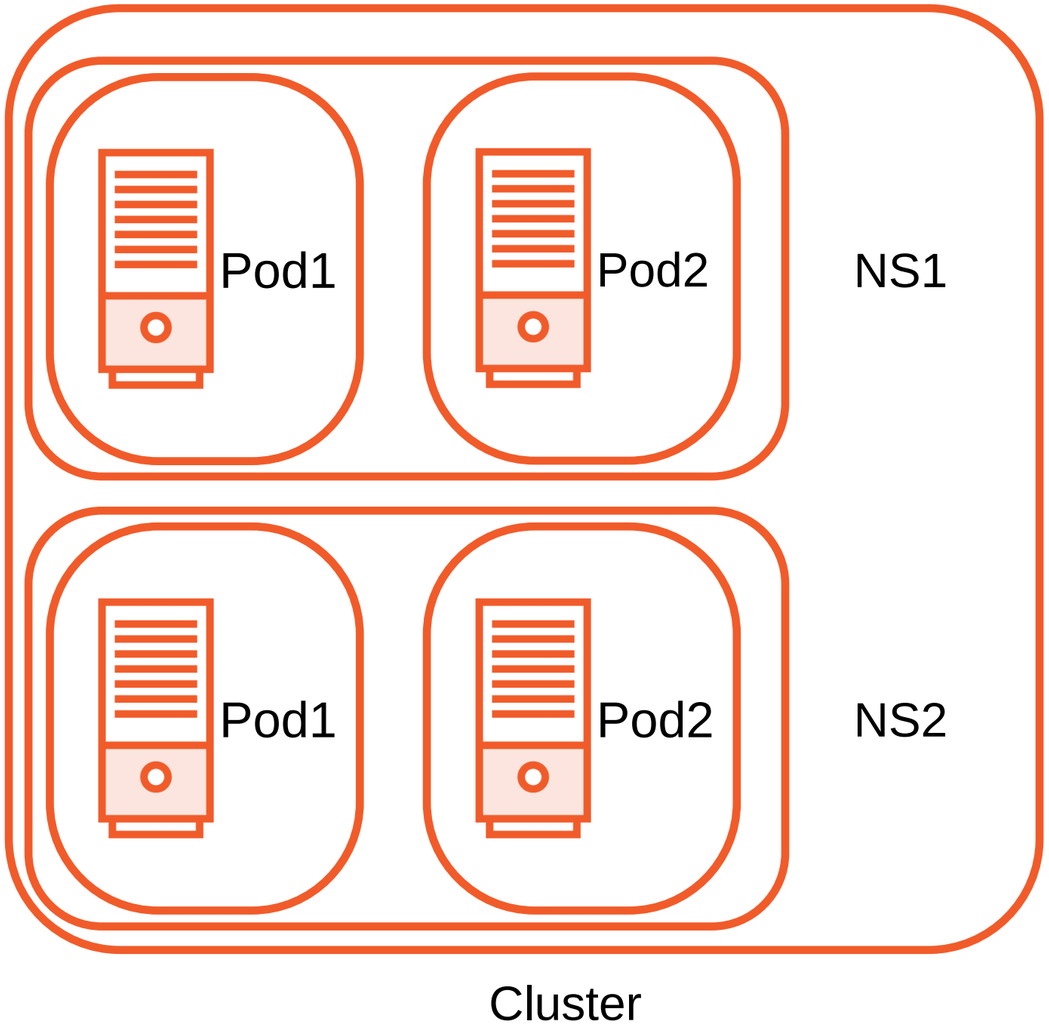
Working with Namespaces

- Create/Query/Delete
- Operate on objects in a Namespace
- Some objects are Namespaced…some aren’t
  - Resources are Namespaced…`Pods`, `Controllers`, `Services`
  - Physical things are not…`PersistentVolumes`, `Nodes`



Working with Namespaces (con’t)

- `default`
- `kube-public` sharable namespace
- `kube-system`
- User Defined
  - Imperatively with `kubectl`
  - Declaratively in a Manifest in YAML



```
apiVersion: v1
kind: Namespace
metadata:
  name: playgroundinyaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: playgroundinyaml
```

Creating Namespaces and Creating Objects in Namespaces

```
kubectl create namespace playground1

kubectl run nginx --image=nginx --namespace playground1
```

Demo

Creating a Namespace

Adding resources to a Namespace

Querying a Namespace

Interacting with resources in a Namespace

Deleting all resources in a Namespace

Labels

Used to organize resources - Pods, Nodes and more

Label Selectors are used to select/query Objects

Return collections of Objects that satisfy search conditions

Enables you to perform operations on a collection of resources…like Pods

Influence internal operations of Kubernetes

Labels (con’t)

Non-hierarchical, key/value pair

Have more than one label per resource

Enables more complex representations of state and ability to query

Keys can be 63 characters or less

Values can be 253 characters or less

Using Labels

- Creating resources with Labels
  - Imperatively with `kubectl`
  - Declaratively in a Manifest in YAML
- Editing existing resources' Labels
  - Assign (add) a new Label
  - Overwriting an existing Label

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels
    app: v1
    tier: PROD
spec:
   ...
```

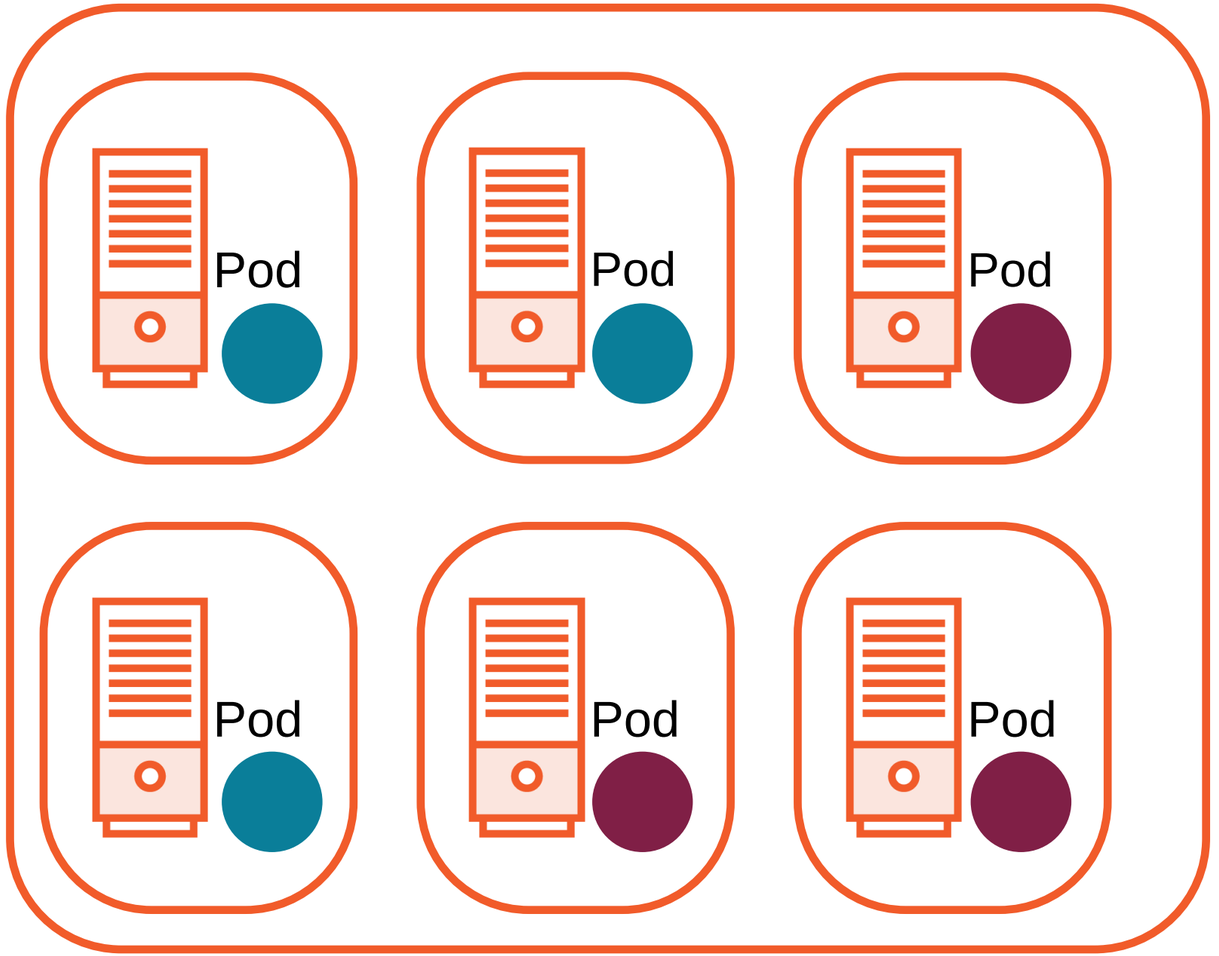
Adding and Editing Labels

```
kubectl label pod nginx tier=PROD app=v1

kubectl label pod nginx tier=DEBUG app=v1 --overwrite

kubectl label pod nginx app-
```

Using Labels in Kubernetes



Querying Using Labels and Selectors

`kubectl`	`get`	`pods`	`--show-labels`	
`kubectl`	`get`	`pods`	`--selector`	`tier=prod`
`kubectl`	`get`	`pods`	`-l`	`'tier in (prod,qa)'`
`kubectl`	`get`	`pods`	`-l`	`'tier notin (prod,qa)'`
`kubectl`	`get`	`nodes`	`--show-labels`	

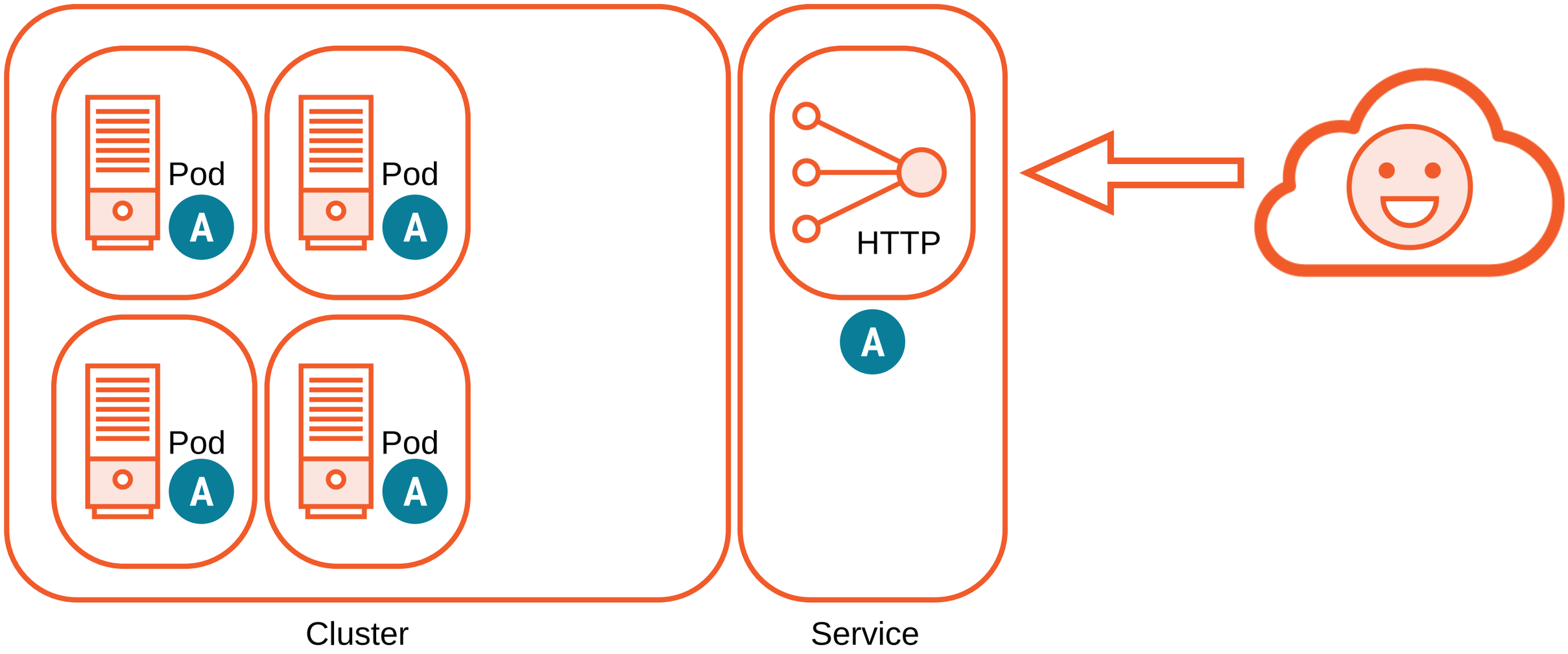
How Kubernetes Uses Labels

Controllers and Services match Pods using Selectors

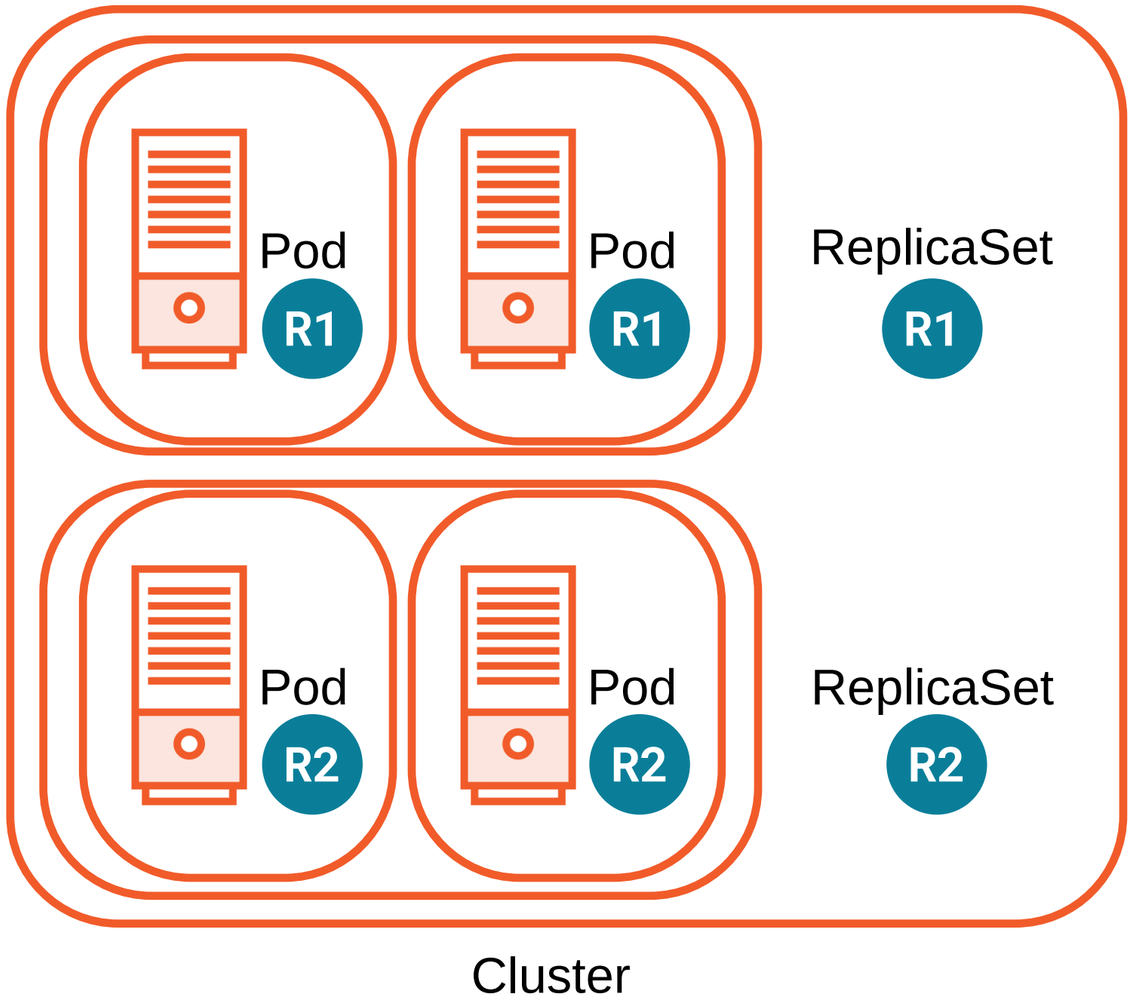
Influencing Pod Scheduling

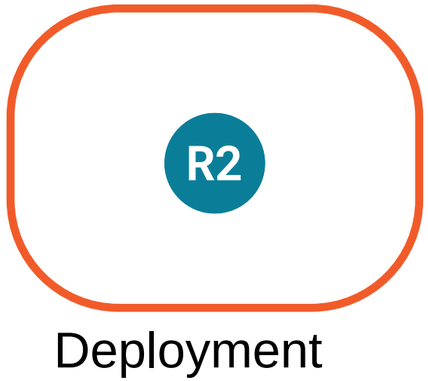
- Scheduling to specific Nodes
- Special hardware (SSD or GPU)
- Using a label selector

Services

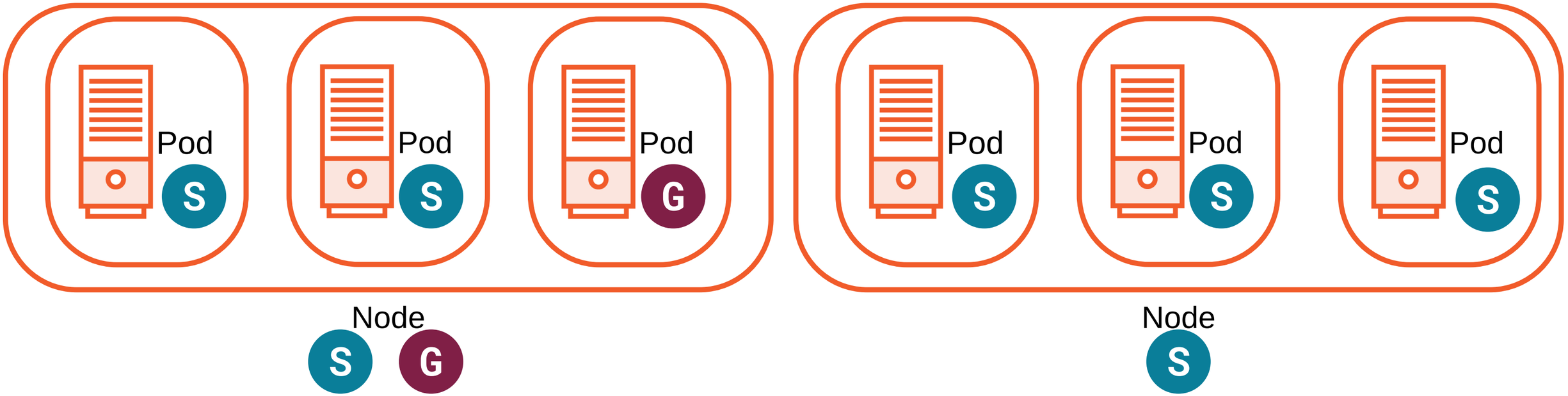


Controller Operations - Deployment



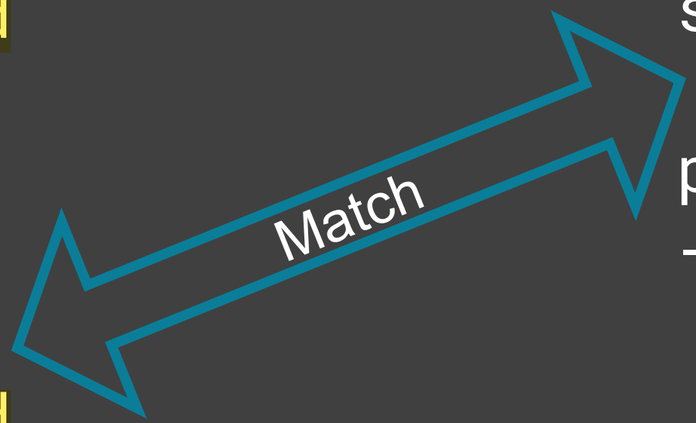


Scheduling Pods to Nodes



Defining Deployments and Services

```
kind: Deployment
...
spec:
 selector:
      matchLabels:
        run: hello-world
...
  template:
    metadata:
      labels:
        run: hello-world
    spec:
        containers:
...
```




```
kind: Service
...
spec:
  selector:
    run: hello-world
  ports:
  - port: 80
    protocol: TCP
    targetPort: 8080
```

Demo

Working with Labels

Interacting with Pods by Name and Label

Kubernetes resource management

- Services
- Deployments
- Pod Scheduling

Annotations

Used to add additional information about your cluster resources

Mostly used by people or tooling to make decisions

Build, release, and image information exposed in easily accessible areas

Saves you from having to write integrations to retrieve data from external data sources

Annotations (con’t)

Non-hierarchical, key/value pair

Can’t be used to query/select Pods or other resources

Data is used for “other” purposes

Keys can be up to 63 characters

Values can be up to 256KB

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotation: owner: Anthony
spec:
  containers:
  - name: nginx
    image: nginx
...
```

Adding and Editing Annotations

```
kubectl annotate pod nginx-pod owner=Anthony
```

```
kubectl annotate pod nginx-pod owner=NotAnthony --overwrite
```

Organizing Objects in Kubernetes

Namespaces - When you want to put a boundary around resources, security, or naming

Labels - When you want to act on objects in groups or influence k8s operations

Annotations - When you want to add additional information about a resource

Summary

Organizing Objects in Kubernetes

- Namespaces
- Labels
- Annotations

How Kubernetes uses Labels

- Services
- Deployments
- Scheduling

What’s Next!

Running and Managing Pods