Managing Kubernetes Controllers and Deployments

INTRODUCTION AND USING CONTROLLERS TO DEPLOY APPLICATIONS

Kien Bui
DevOps & Platform Engineer

Course Overview

Using Controllers to Deploy Applications and Deployment Basics

Maintaining Applications with Deployments

Deploying and Maintaining Applications with DaemonSets and Jobs

Overview

Controllers in Kubernetes

How Controllers Work Types of Controllers

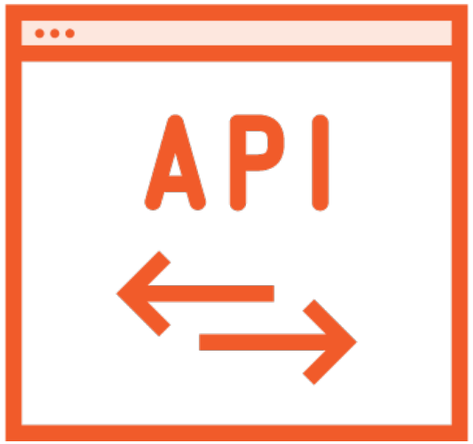
`Deployment` Controller Basics Underst

`ReplicaSets`

Kubernetes Principles

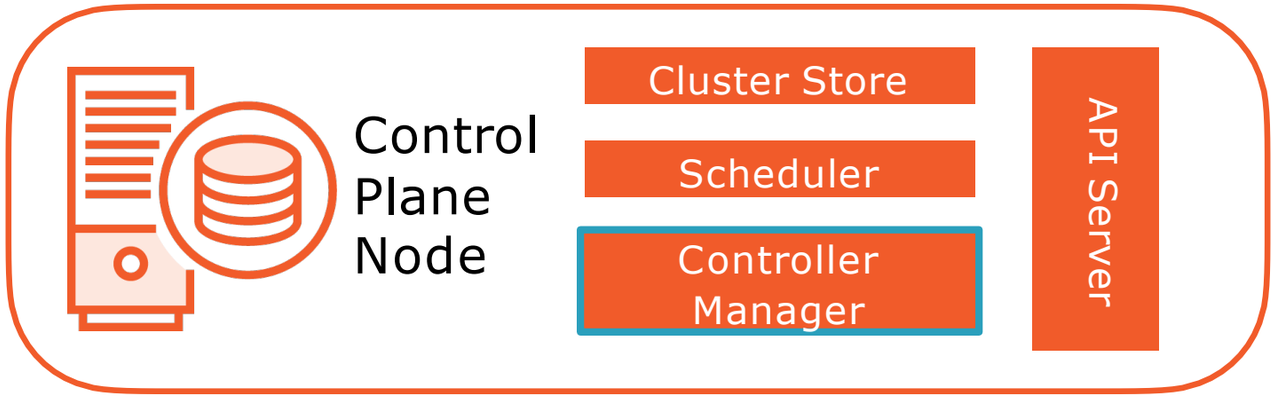
Desired State
Declarative
Configuration

Controllers
Control Loops



The API Server

Control Plane Components



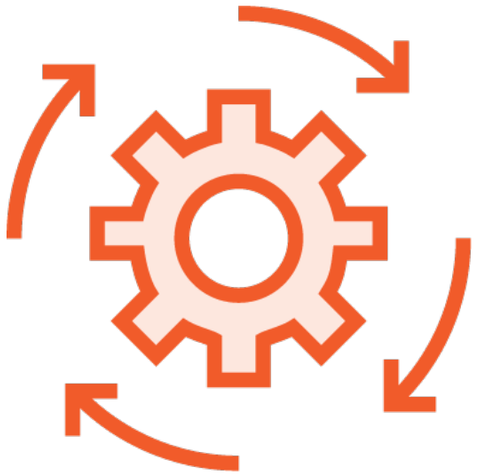
Controller Manager

kube-controller-manager

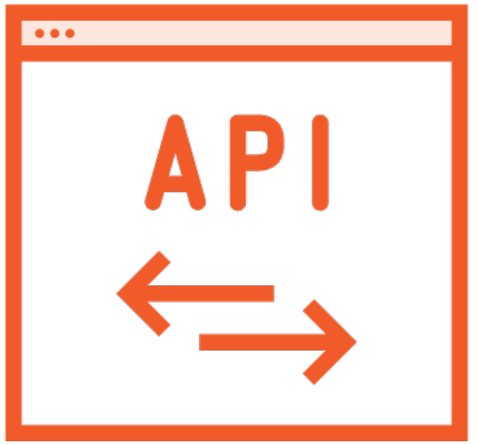
cloud-controller-manager

Controller Operations

Watch State



Operations



API Server

Controllers in Kubernetes

Pod Controllers

Other Controllers

Pod Controllers

- ReplicaSet
- Deployment
- DaemonSet
- StatefulSet
- Job
- CronJob

Other Controllers

`Node`

`Service`

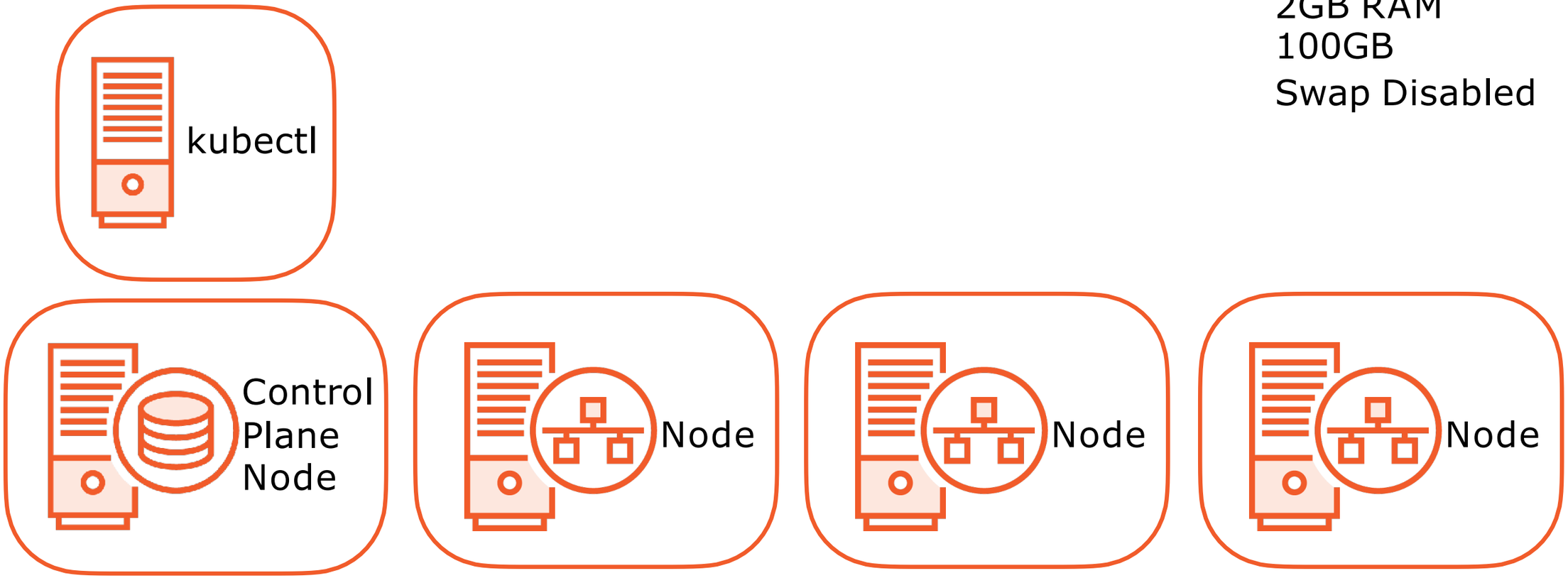
`Endpoint`

Many more…

Hostnames set
Host file on each

Lab Environment

Ubuntu 18.0.4
VMware Fusion VMs
2vCPU
2GB RAM
100GB
Swap Disabled



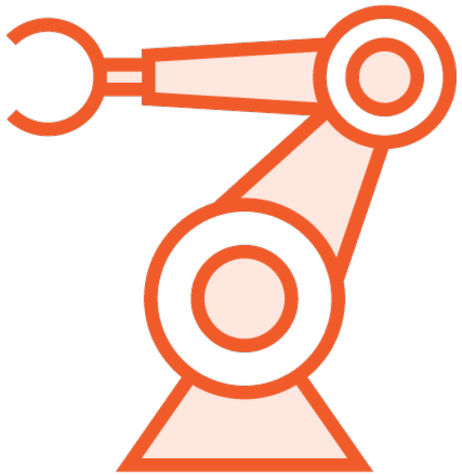
c1-cp1	c1-node1	c1-node2	c1-node3
172.16.94.10	172.16.94.11	172.16.94.12	172.16.94.13

Demo

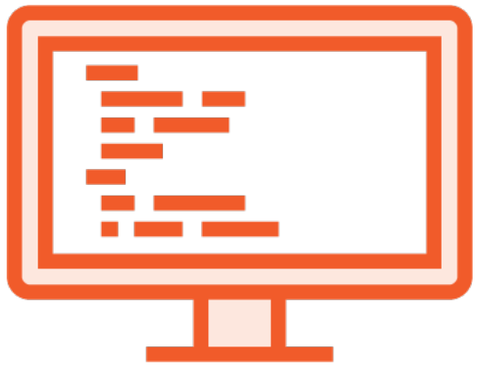
Examining System Pods and their Controllers

Deployment Controller

Declarative
Updates



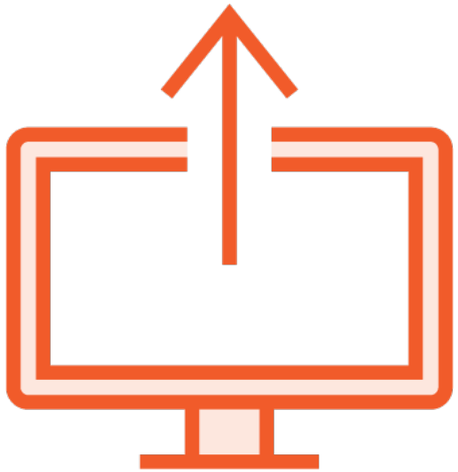
Orchestration



Managing
Application State

Managing Application State with Deployments

Creating



Updating

Scaling

Creating Deployments

Declaratively

Writing a `Deployment` Spec in code (YAML)

`Selector`

`Replicas`

`Pod Template`

Imperatively

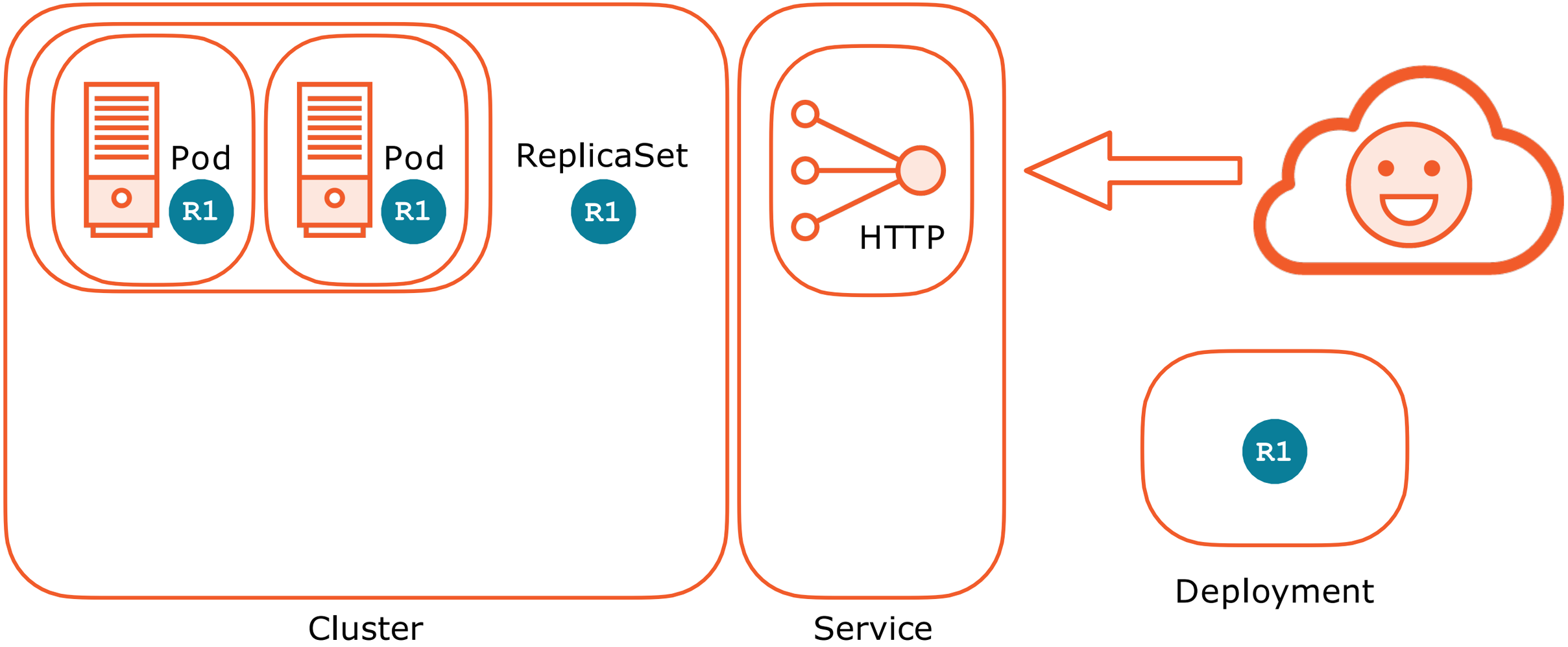
```
kubectl create deployment hello-world --image=gcr.io/google-samples/hello-app:1.0

kubectl scale deployment hello-world --replicas=5
```

Defining a Basic Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 5
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
 ...
```


```
kubectl apply -f deployment.yaml
```

Controller Operations - Deployment

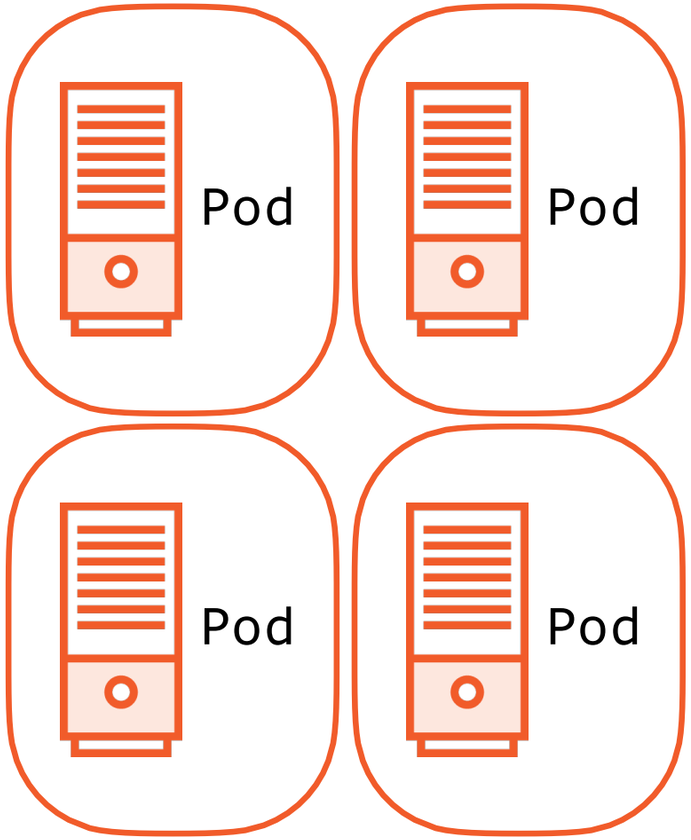


Demo

Creating a `Deployment`

- Imperatively
- Declaratively

Understanding ReplicaSets



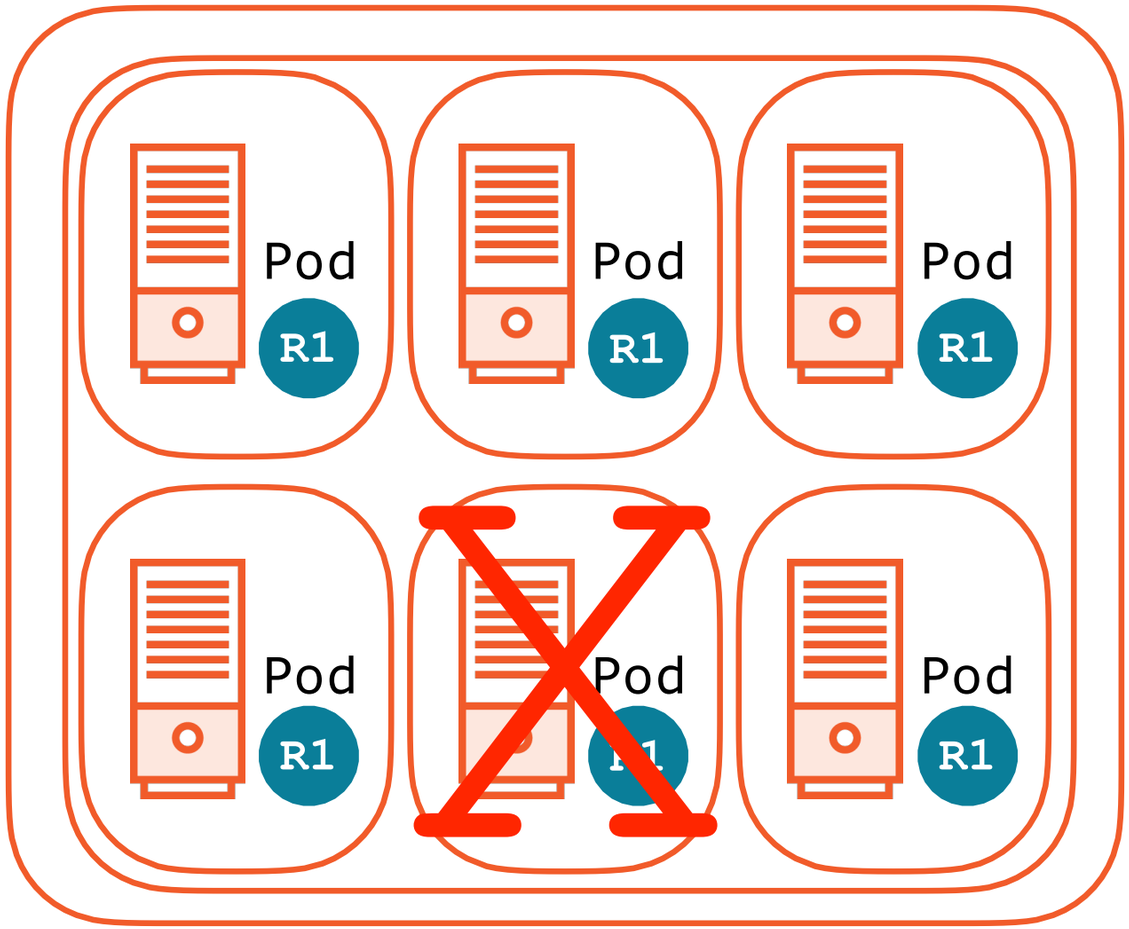
Deploys a defined number of Pods

Consists of a Selector, Number of Replicas (Pods) and a Pod Template

Generally speaking you don't create `ReplicaSets` directly

You create `Deployments`

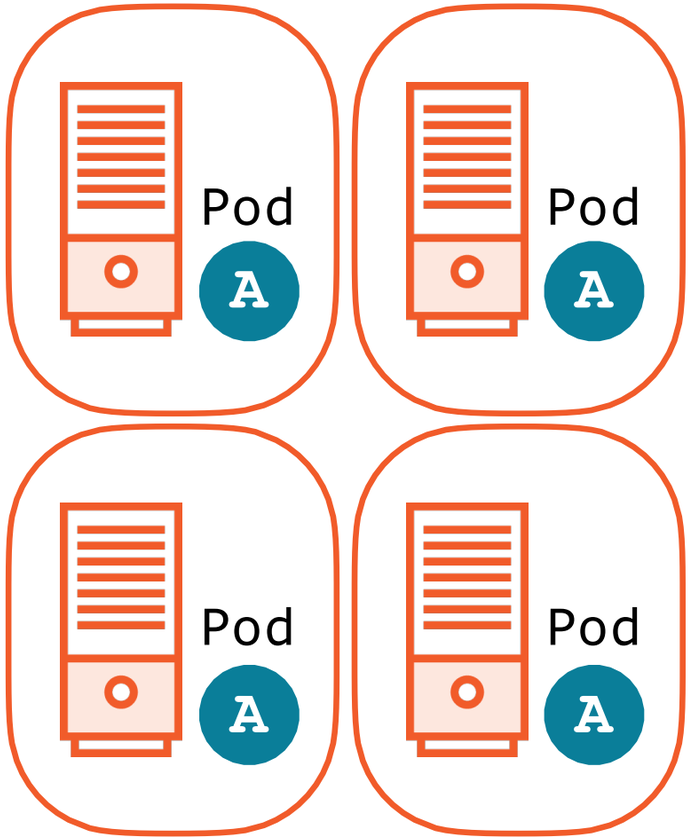
ReplicaSets Pod Operations



ReplicaSet

```
apiVersion: apps/v1
kind: ReplicaSet
...
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world-pod
  template:
    metadata:
      labels:
        app: hello-world-pod
    spec:
      containers:
      ...
```

ReplicaSet Selectors



`ReplicaSets` allow for more complex, set based selectors

`matchExpressions` as the selector

`Operators`

`In, NotIn, Exists and DoesNotExist`

`Keys`

`Values`

```
kind: ReplicaSet
...
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world-pod
  template:
    metadata:
      labels:
        app: hello-world-pod
    spec:
      containers:
```

```
...
spec:
  replicas: 1
  selector:
    matchExpressions:
      - key: app
        operator: In
        values:
          - hello-world-pod-me
  template:
    metadata:
      labels:
        app: hello-world-pod-me
    spec:
```

ReplicaSets and Failures

Pod Failures

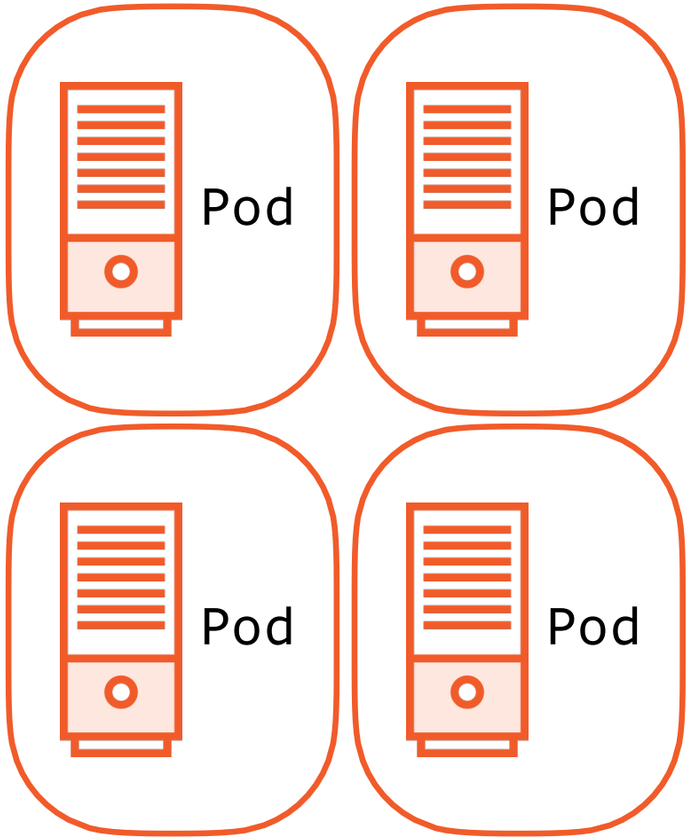
Rescheduled and a new Pod is started in the c

Node Failures Transient failure Permanent failure

kube-controller-manager

`pod-eviction-timeout` - 5 minutes (default)

A Side Note on Replication Controllers



Legacy documentation and code samples

`ReplicationController`

Only a single label (key and value pair)

`ReplicaSets` allow for more expressive representations of state with set based selectors

Demo

Create a `Deployment` (`ReplicaSet`)

Deleting a Pod in a `ReplicaSet`

Isolating a Pod from a `ReplicaSet`

Taking over an existing Pod in a `ReplicaSet`

Node failures and `ReplicaSets`

ReplicaSets or Deployments?

`Deployments` to manage our `ReplicaSets`

`ReplicaSets` are the building blocks of `Deployments`

Review

Controllers in Kubernetes

How Controllers Work Types of Controllers

`Deployment` Controller Basics Underst

`ReplicaSets`

What's Next!

Deploying and Maintaining Applications with Deployments