Kubernetes for Developers: Core Concepts

KUBERNETES FROM A DEVELOPER PERSPECTIVE

Dan Wahlin
WAHLIN CONSULTING

@danwahlin www.codewithdan.com

Course Overview

- Kubernetes from a Developer Perspective
- Understanding Storage Options
- Creating Pods
- Creating ConfigMaps and Secrets
- Creating Deployments
- Putting It All Together
- Creating Services
- Course Summary

Target Audience

Developers looking to understand Kubernetes core concepts

Course Prereqs

Comfortable using command-line tools and virtual machines

General familiarity with software development

Understanding of Docker containers and how they work

Introduction

Module Overview

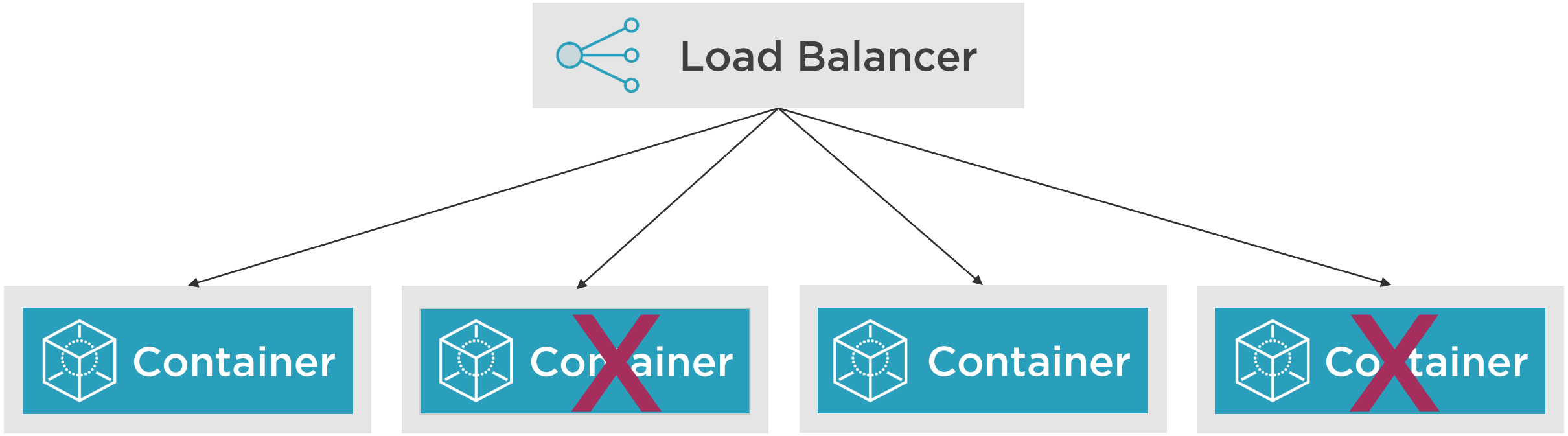
- Kubernetes Overview
- The Big Picture
- Benefits and Use Cases
- Running Kubernetes Locally
- Getting Started with kubectl
- Web UI Dashboard

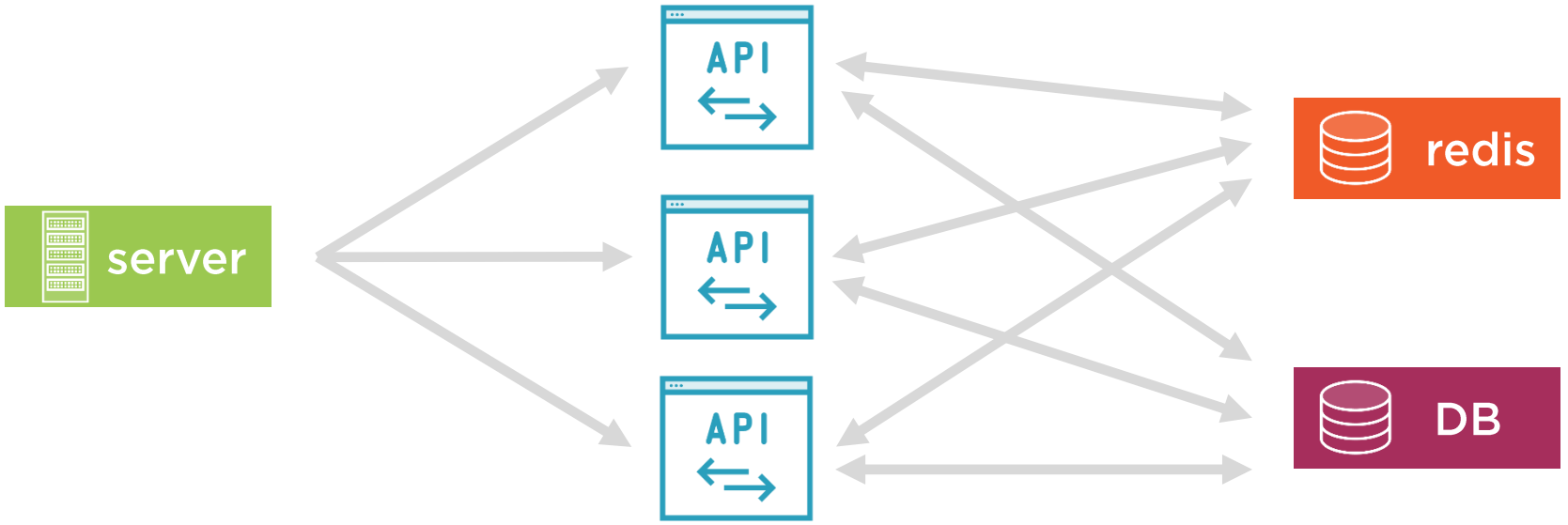
Kubernetes Overview

"Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications."

https://kubernetes.io

How Are You Managing Containers?





How do you manage all of these containers?

It would be

nice if we could...

Package up an app and let something else manage it for us

Not worry about the management of containers

Eliminate single points of failure

Scale containers

Update containers without bringing down the application

Have robust networking and persistent storage options

Container
Container
Container
Kubernetes
(conductor)

Kubernetes is the conductor of a container orchestra.

Key Kubernetes Features

Service Discovery/ Load Balancing

Storage Orchestration

Automate Rollouts/Rollbacks

Self-Healing

Secret and Configuration Management

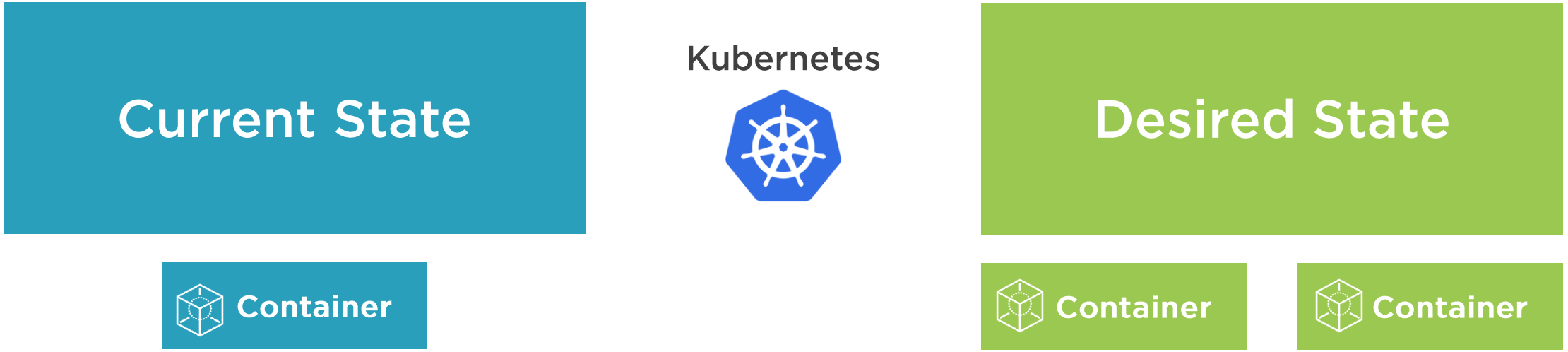
Horizontal Scaling

The Big Picture

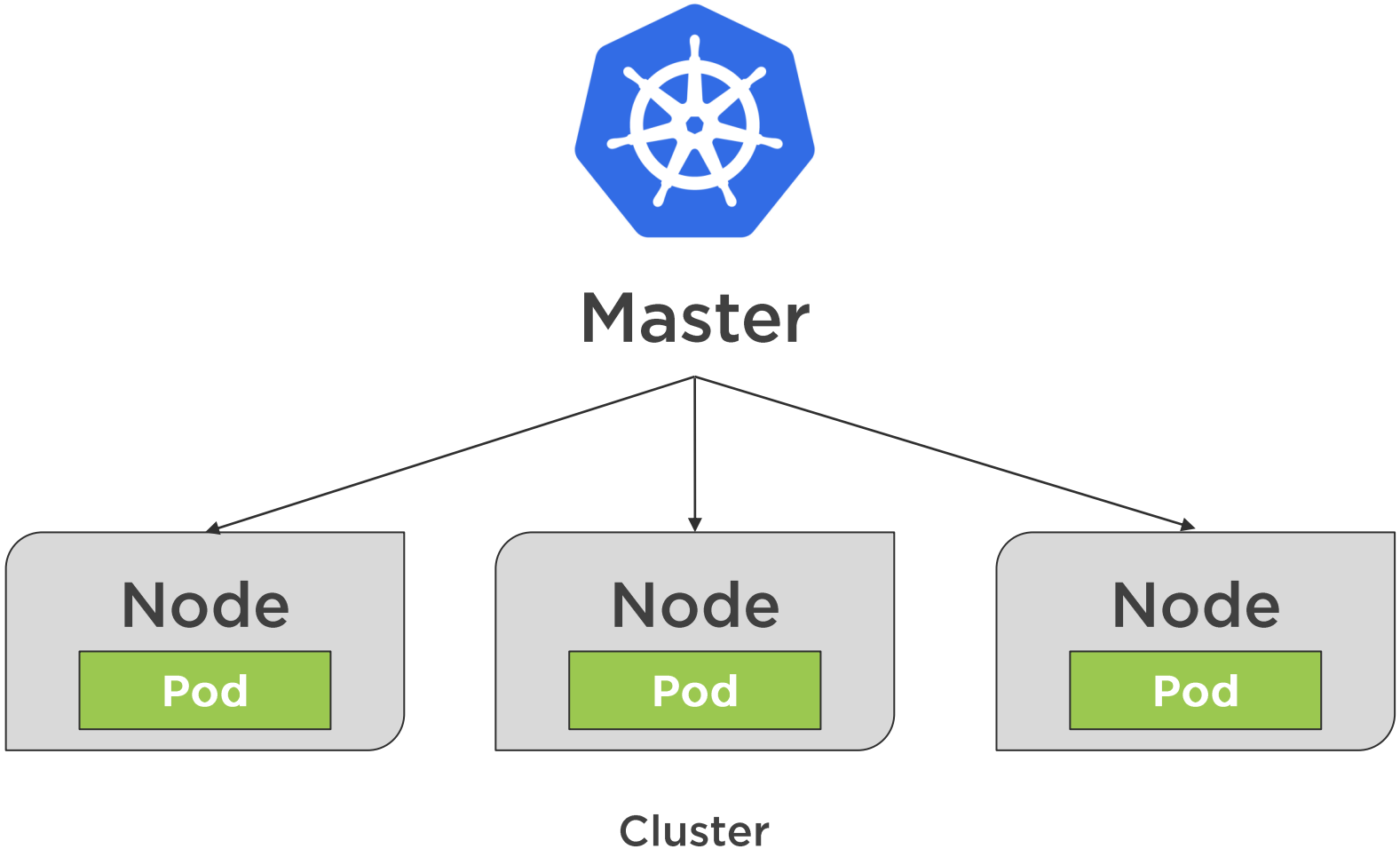
Kubernetes

- Container and cluster management
- Open source project
- Used internally by Google for 15+ years and donated to the Cloud Native Computing Foundation
- Supported by all major cloud platforms
- Provides a "declarative" way to define a cluster's state

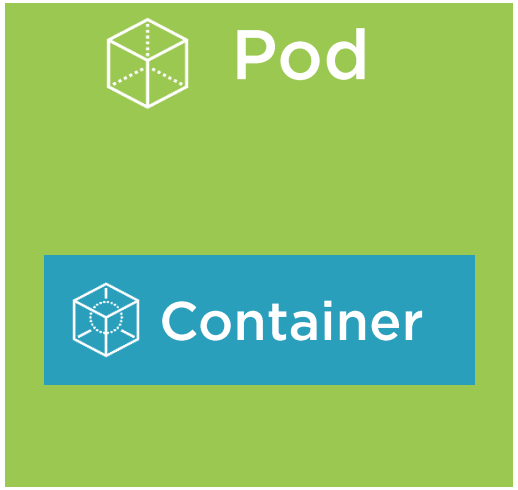
Kubernetes Moves You to a Desired State



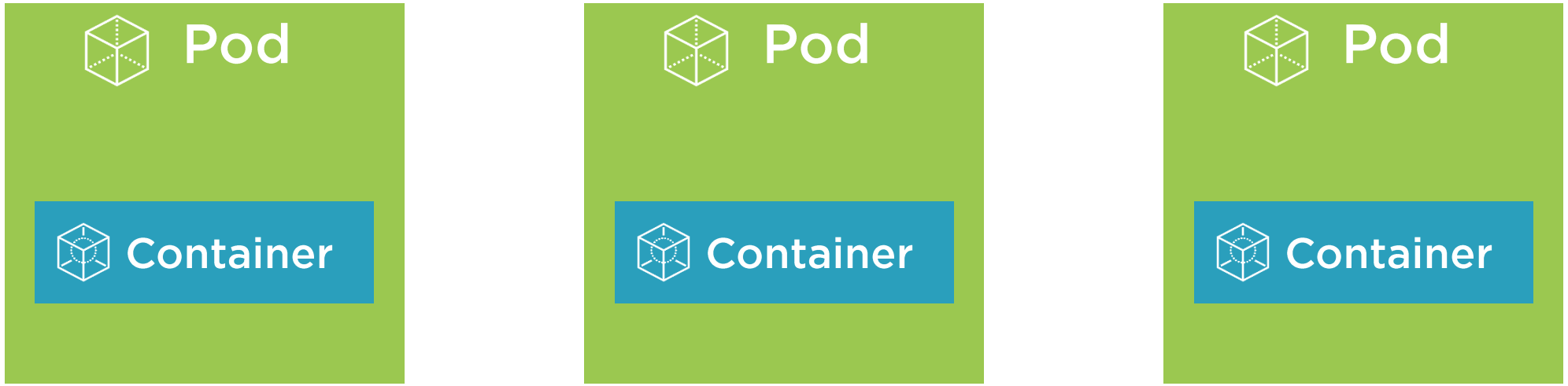
The Big Picture

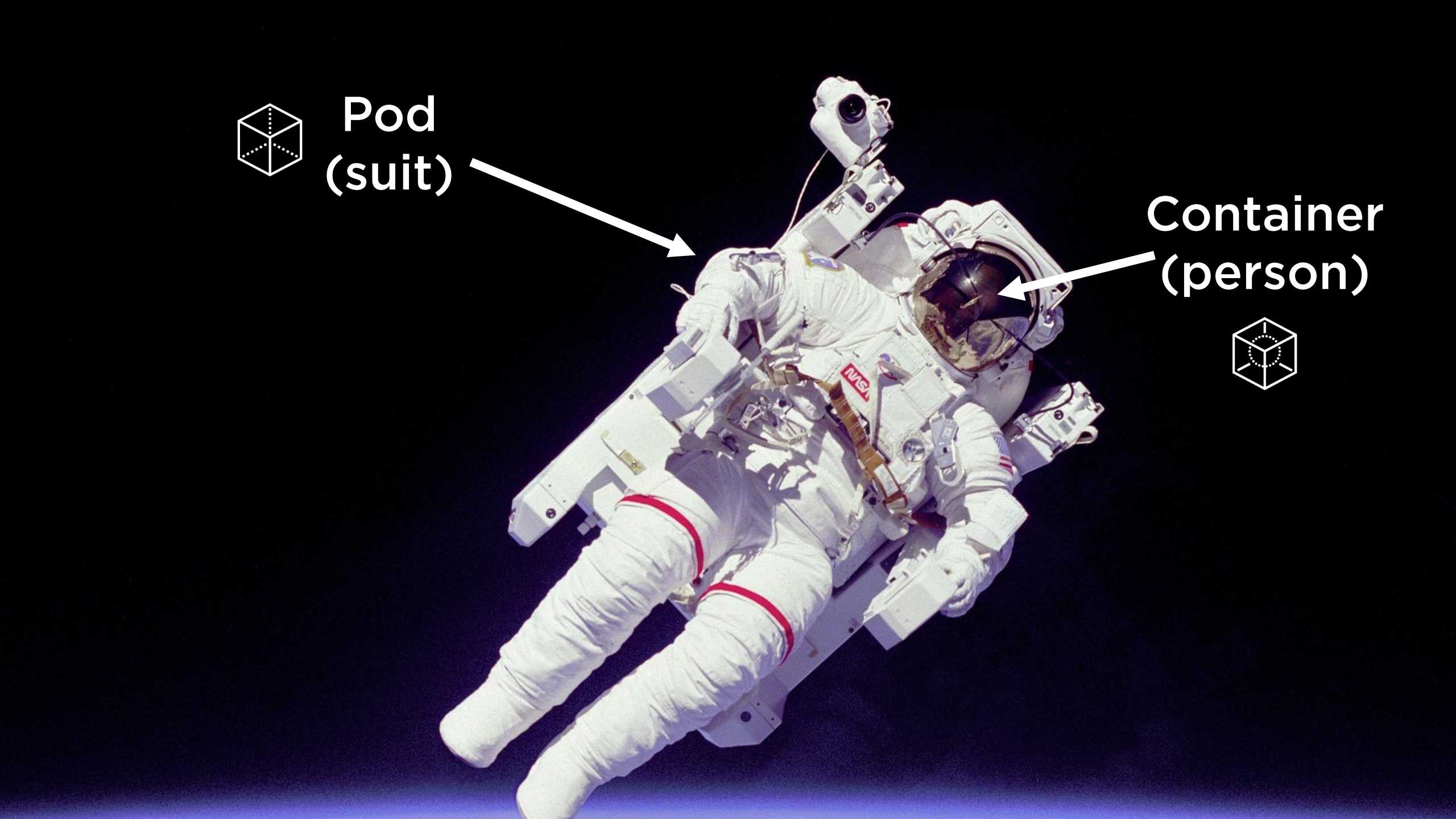


Pods and Containers

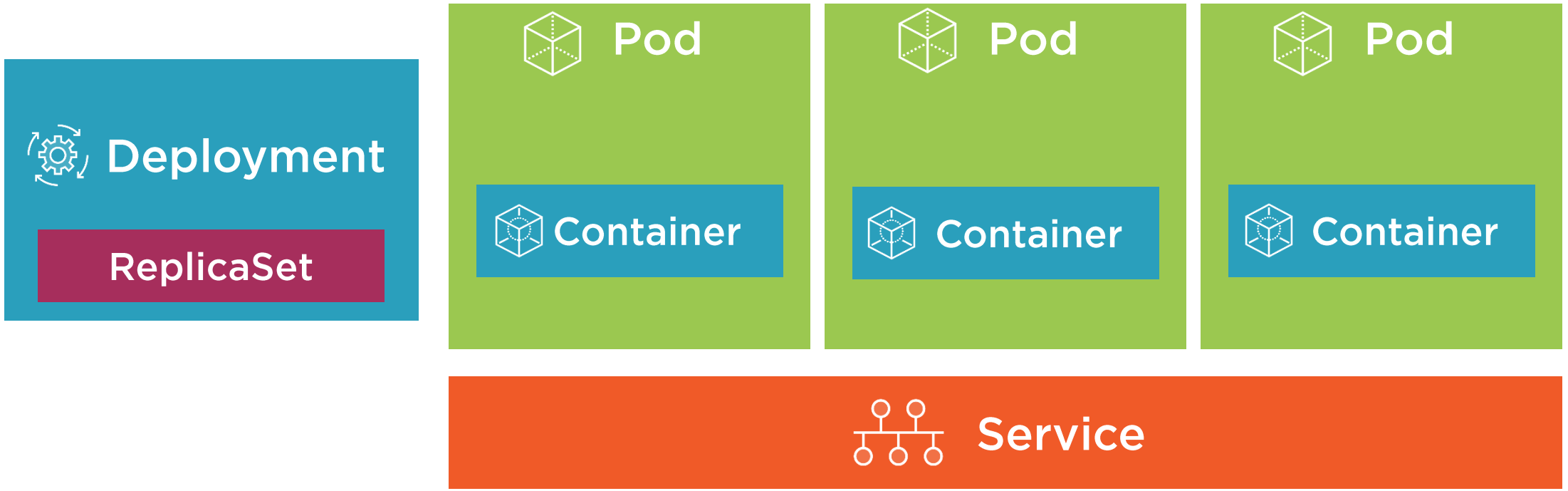


Pods and Containers

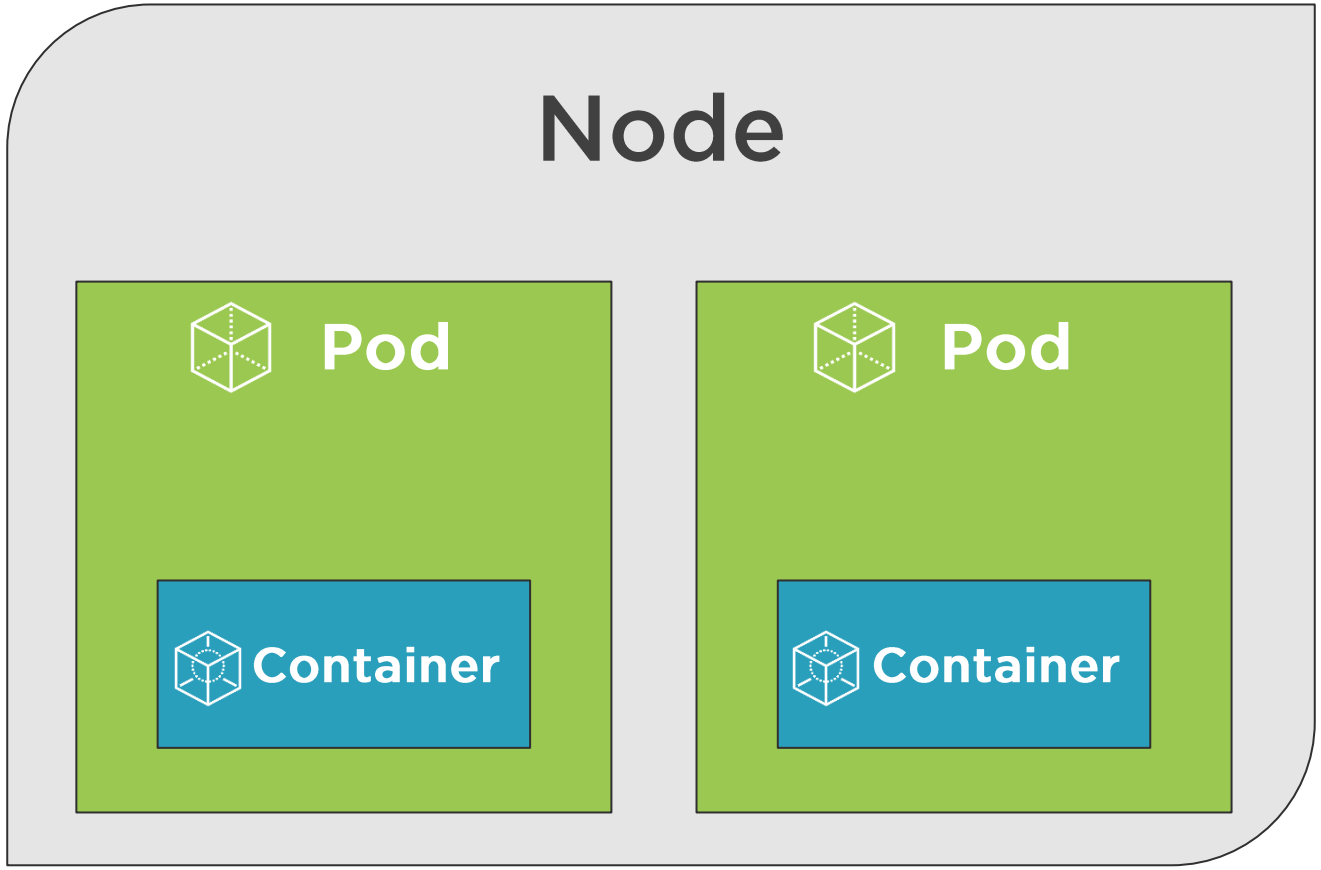



Pod
(suit)
Container
(person)

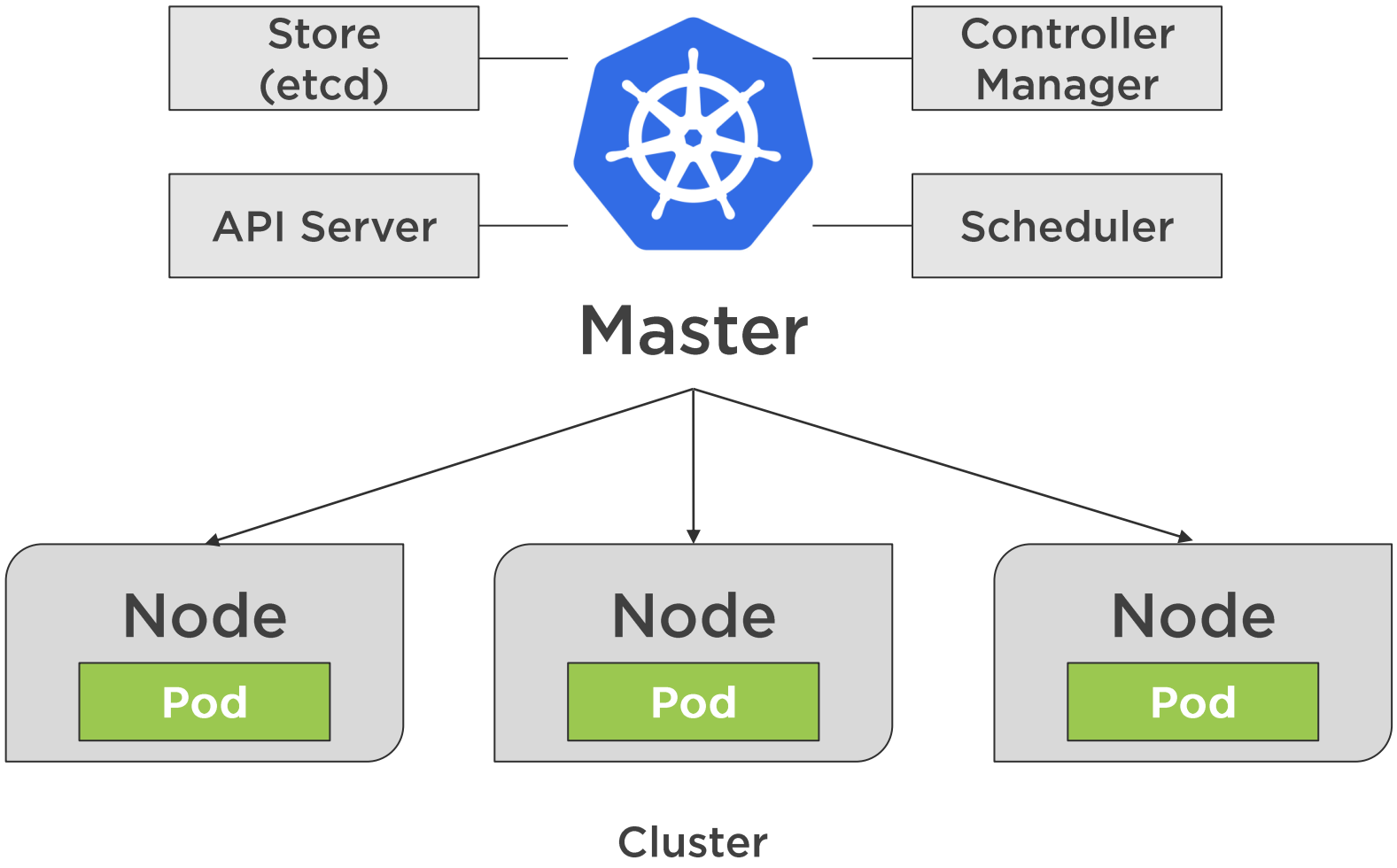
Kubernetes Building Blocks



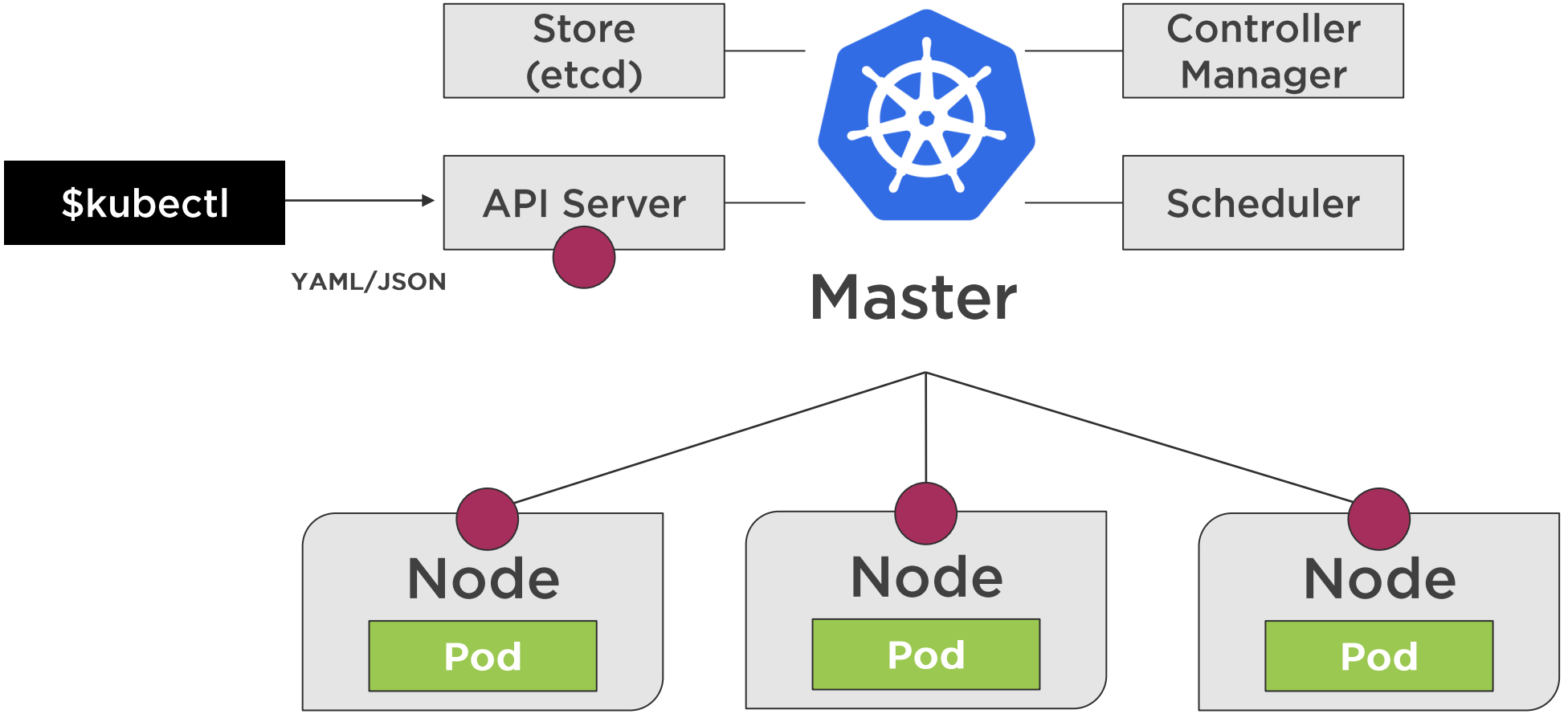
Pods and Nodes



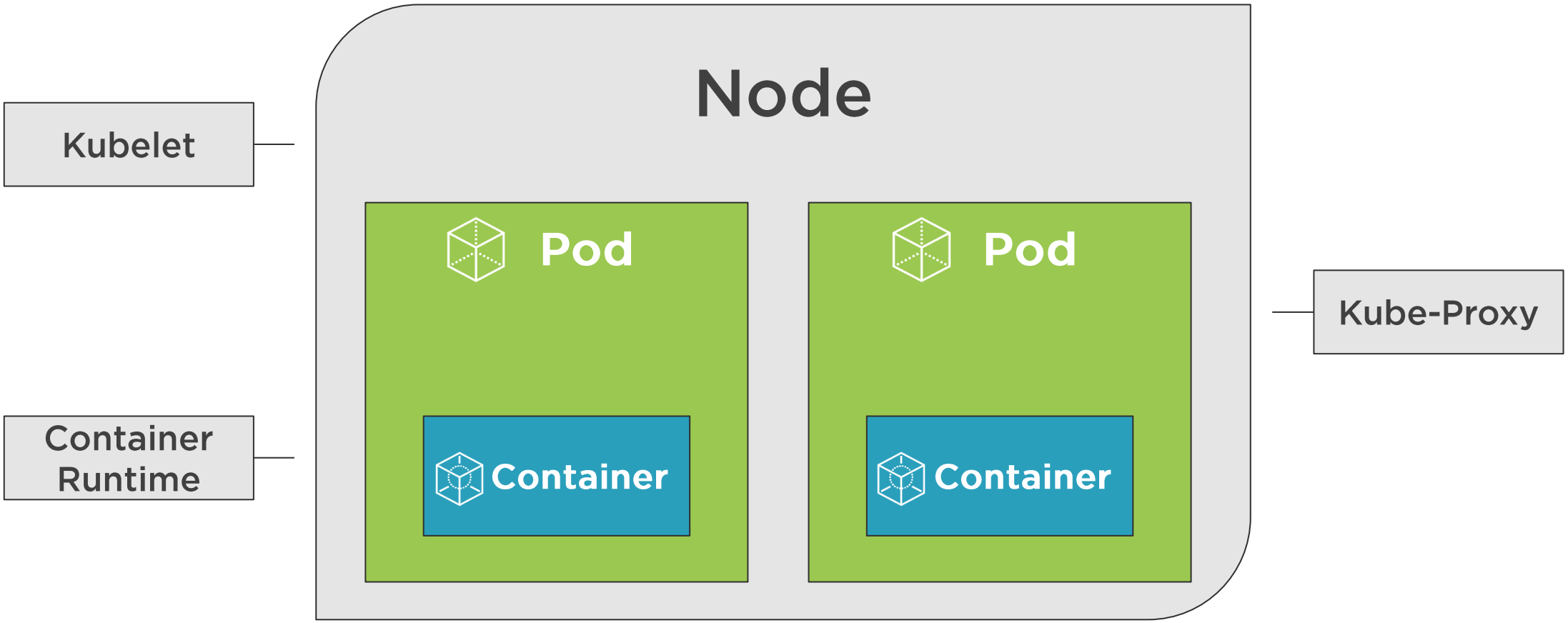
The Master Node



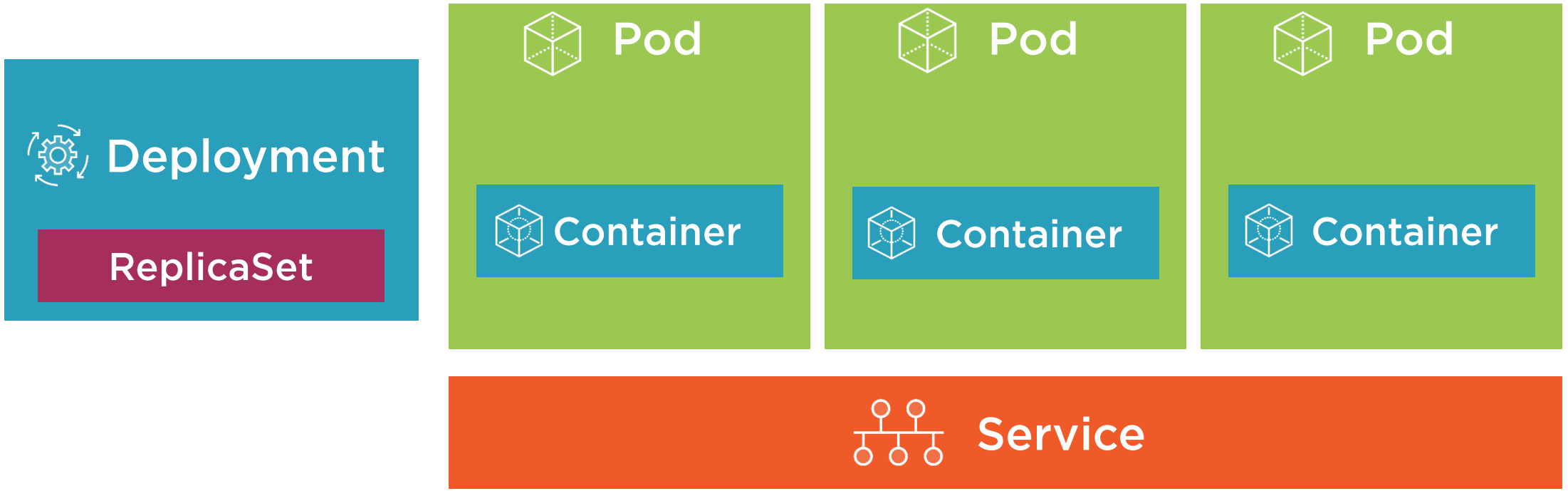
Communicating with kubectl



Kubernetes Nodes



The Big Picture

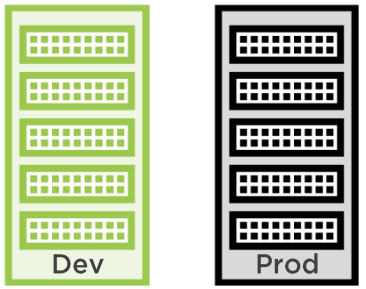


Benefits and Use Cases

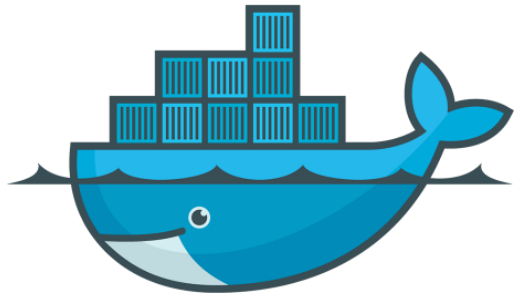
Key Container Benefits

Accelerate Developer Onboarding

Eliminate App Conflicts

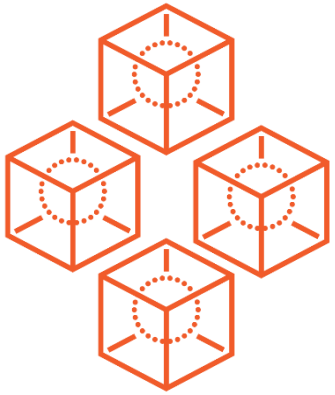


Environment Consistency

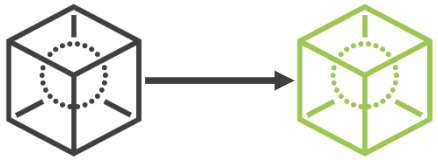


Ship Software Faster

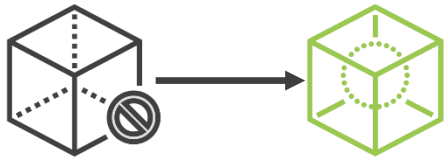
Key Kubernetes Benefits



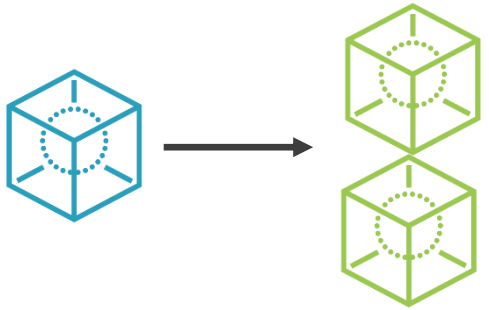
Orchestrate Containers



Zero-Downtime Deployments



Self Healing (superpowers)



Scale Containers

Developer Use Cases

- Emulate production locally
- Move from Docker Compose to Kubernetes
- Create an end-to-end testing environment
- Ensure application scales properly
- Ensure secrets/config are working properly
- Performance testing scenarios
- Workload scenarios (CI/CD and more)
- Learn how to leverage deployment options
- Help DevOps create resources and solve problems

Running Kubernetes Locally

Installing and Running Kubernetes

Minikube

https://github.com/kubernetes/minikube

Docker Desktop

https://www.docker.com/products/docker-desktop

Installing and Running Kubernetes

kind

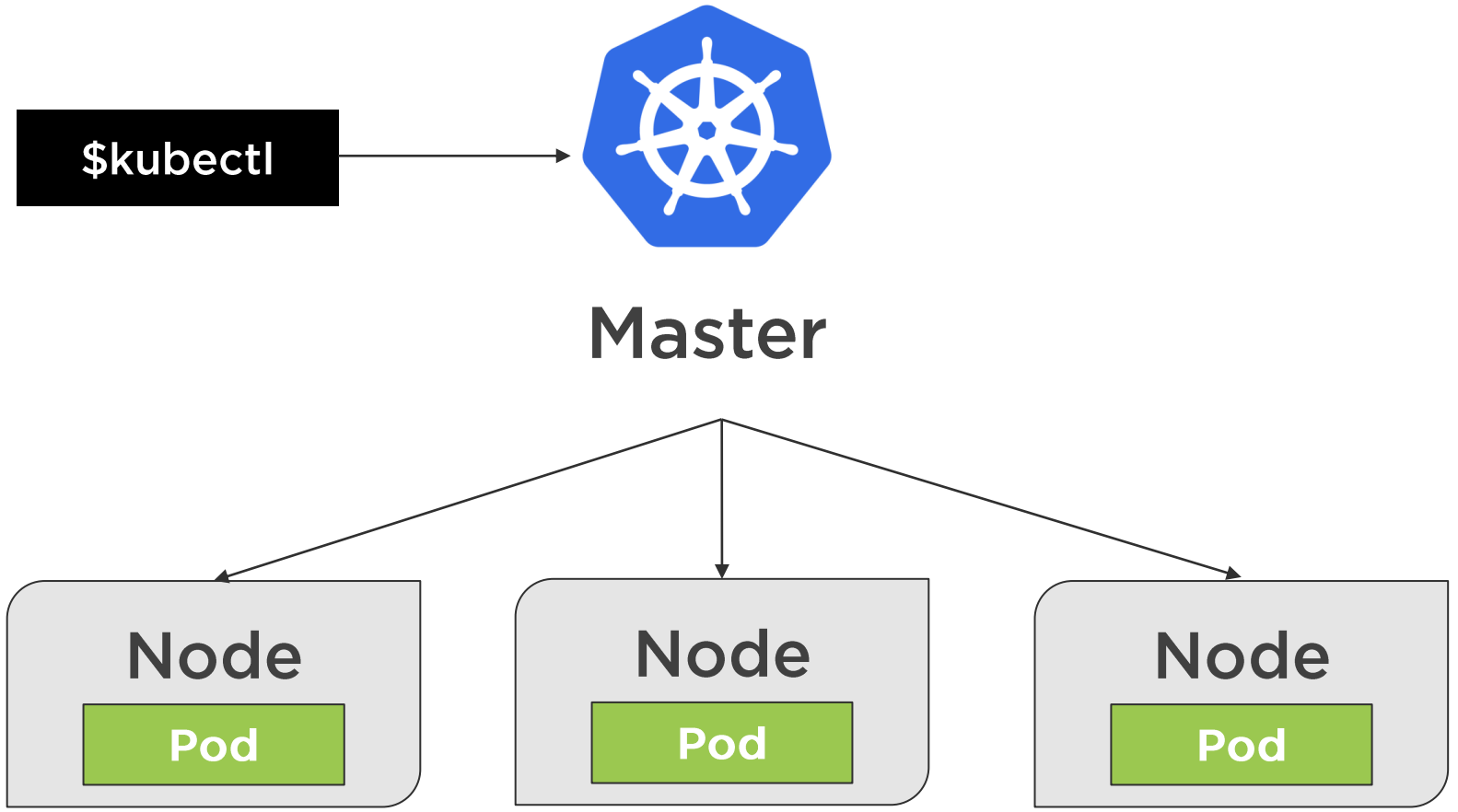
https://kind.sigs.k8s.io

kubeadm

https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm

Getting Started with kubectl

Using kubectl



Getting Started with kubectl Commands

```
kubectl version

kubectl cluster-info

kubectl get all

kubectl run [container-name]
  --image=[image-name]

kubectl port-forward [pod] [ports]

kubectl expose …

kubectl create [resource]

kubectl apply [resource]
```

- Check Kubernetes version
- View cluster information
- Retrieve information about Kubernetes Pods, Deployments, Services, and more
- Simple way to create a Deployment for a Pod
- Forward a port to allow external access
- Expose a port for a Deployment/Pod
- Create a resource
- Create or modify a resource

Aliasing kubectl (to save on typing)

```
# PowerShell

Set-Alias -Name k -Value kubectl
```

◀ Create alias for PowerShell

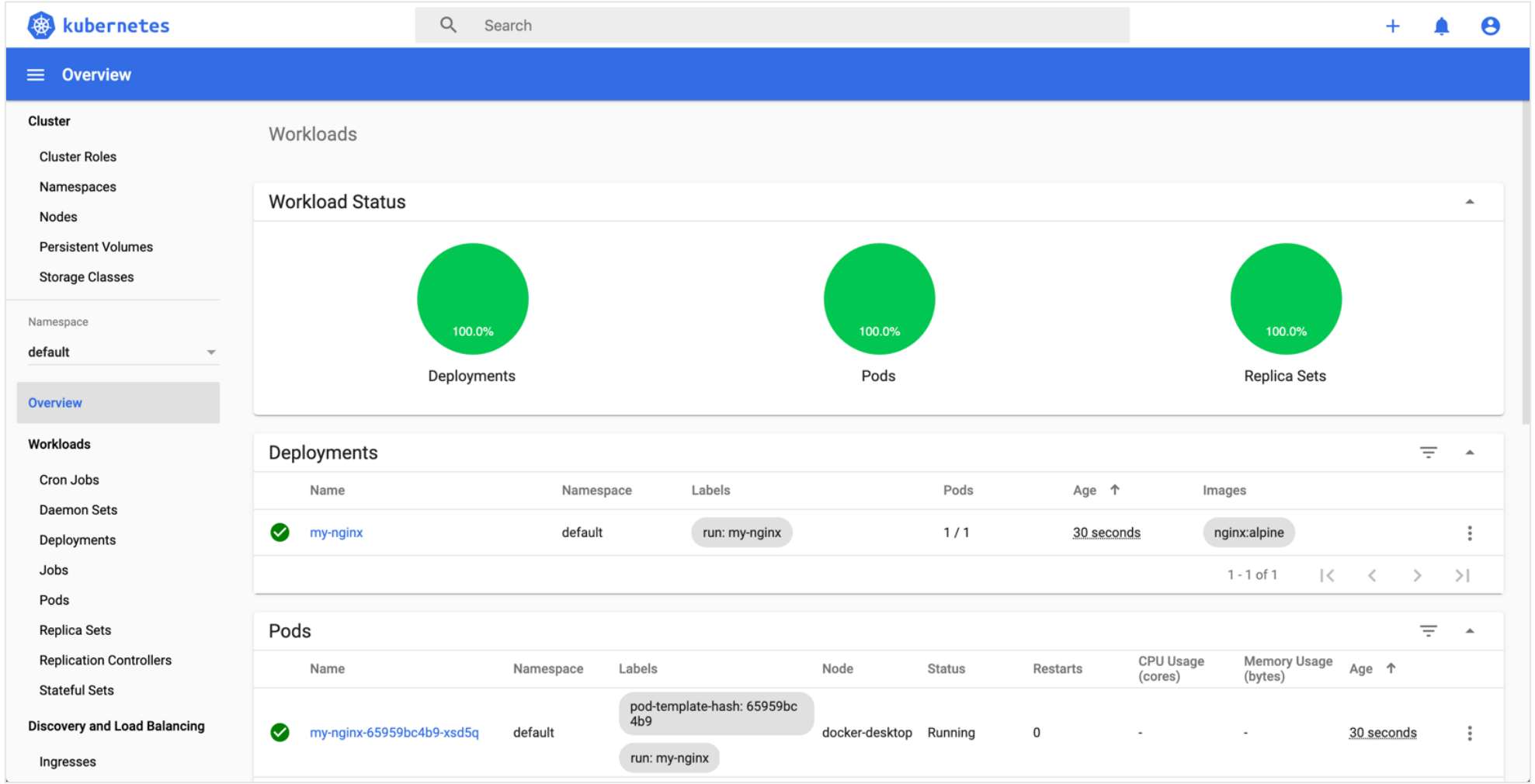
```
# Mac/Linux

alias k="kubectl"
```

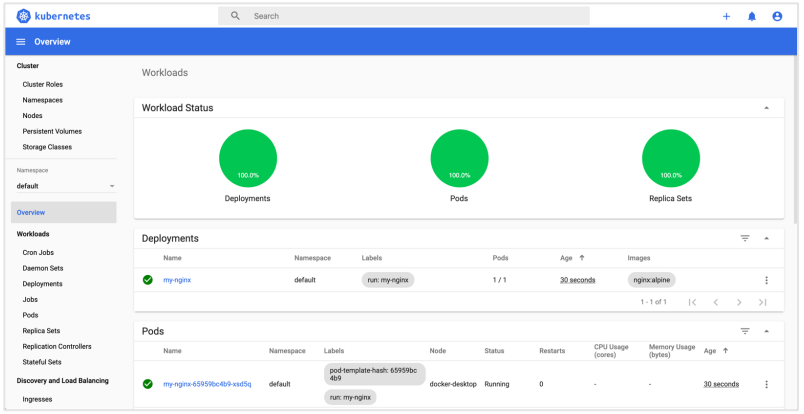
◀ Create alias for Mac/Linux shell

Web UI Dashboard

The Web UI Dashboard



Enabling the Web UI Dashboard



Web UI dashboard provides a user interface to view Kubernetes resources

Steps to enable the UI Dashboard:

- kubectl apply [dashboard-yaml-url]
- kubectl describe secret -n kube-system
- Locate the kubernetes.io/service-account-token and copy the token
- kubectl proxy
- Visit the dashboard URL and login using the token

Summary

Kubernetes provides container orchestration capabilities

Use for production, emulating production, testing, and more

Several options are available to run Kubernetes locally

Interact with Kubernetes using kubectl