Kubernetes for Developers: Moving from Docker Compose to Kubernetes

COMPARING DOCKER COMPOSE AND KUBERNETES

Dan Wahlin
WAHLIN CONSULTING

@danwahlin www.codewithdan.com

Course Overview

Comparing Docker Compose and Kubernetes

Moving from Docker Compose to Kubernetes Using Docker Stack

Moving from Docker Compose to Kubernetes Using Kompose

Moving from Docker Compose to Kubernetes Using Skaffold

Putting It All Together

Target Audience, Pre-Requisites, Software, and Code Samples

Target Audience

Developers looking to switch from running containers using Docker Compose to running containers using Kubernetes.

Course Pre-Reqs

Comfortable using command-line tools and virtual machines

General understanding of Docker containers and how they work - **Docker for Web Developers course**

Understand Kubernetes core concepts - **Kubernetes for Developers: Core Concepts** course

Understand Kubernetes deployment options – **Kubernetes for Developers: Deploying Your Code** course

Required Software

Docker Desktop

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

Minikube

https://github.com/kubernetes/minikube

kind

https://kind.sigs.k8s.io

kubeadm

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

Code Samples

https://github.com/DanWahlin/Angular-JumpStart

https://github.com/DanWahlin/DockerAndKubernetesCourseCode

https://github.com/DanWahlin/CodeWithDanDockerServices

Introduction

Module Overview

Docker Compose Review

Kubernetes Review

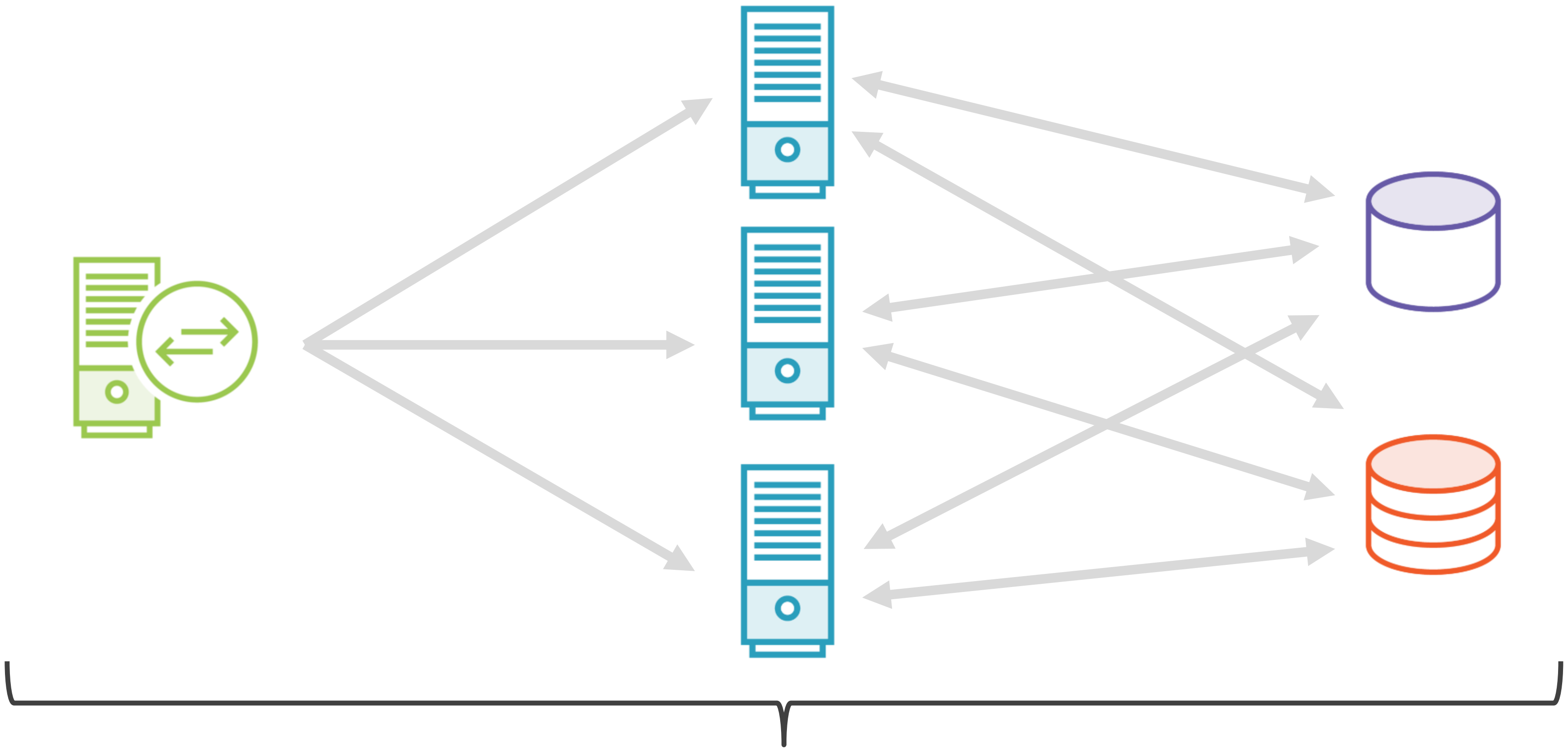
Mapping Docker Compose Services to Kubernetes Resources

Docker Compose Review

> Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

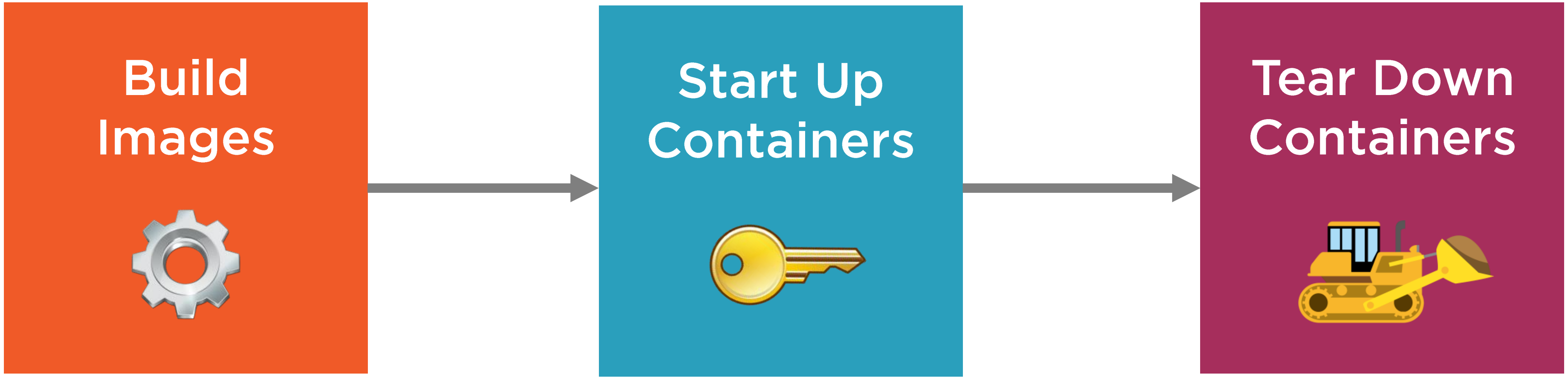
Docker Documentation

Using Docker Compose



Docker Compose
(docker-compose.yml)

Docker Compose Overview

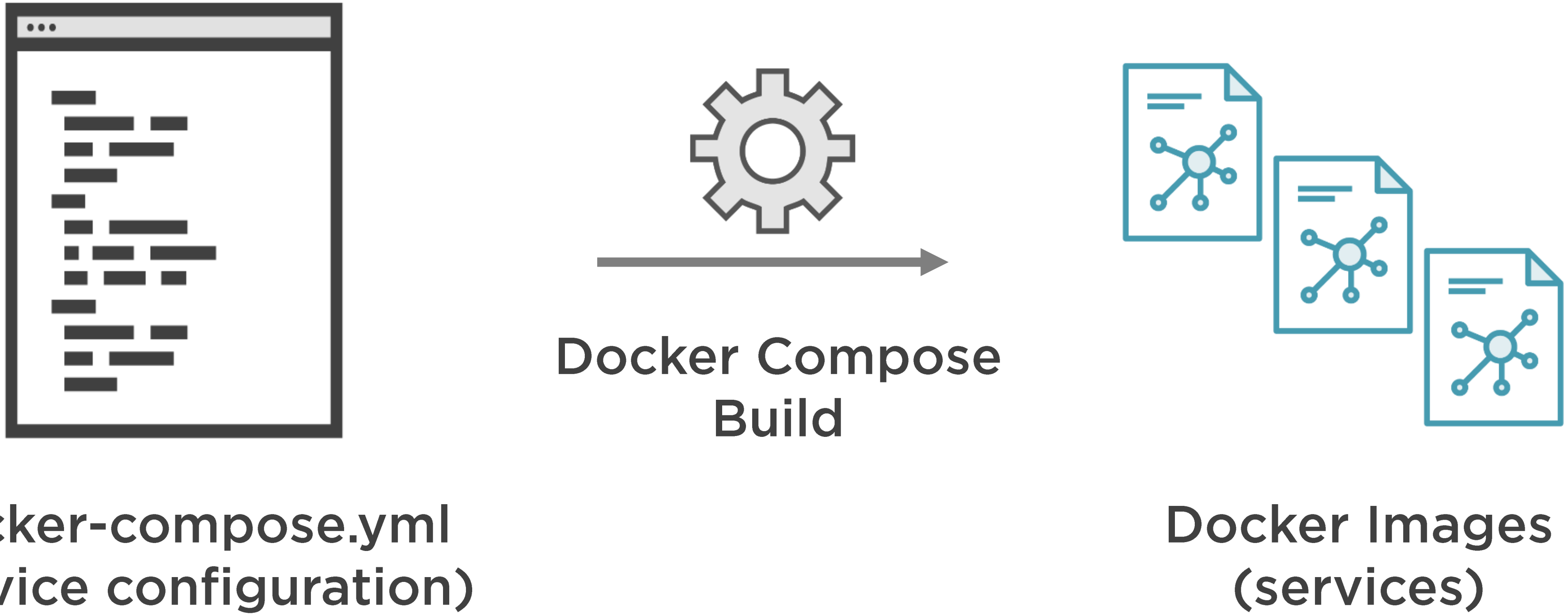


Build Images

Start Up Containers

Tear Down Containers

The Role of Docker Compose



docker-compose.yml
(service configuration)

Docker Compose
Build

Docker Images
(services)

docker-compose.yml Example

```yaml
version: '3.x'
services:

  aspnetcore:
    build:
      context: .
      dockerfile: aspnetcore.dockerfile
    networks:
      - aspnetcoreapp-network

  postgresql:
    image: postgres
    networks:
      - aspnetcoreapp-network

networks:
  aspnetcoreapp-network
    driver: bridge
```

◀ Docker Compose version

◀ ASP.NET Core service with build information and network

◀ PostgreSQL service with image to use and network

◀ Bridge network used for container communication

Key Docker Compose Commands

```
docker-compose build
docker-compose up
docker-compose down
docker-compose logs
```

Question:

Given all the functionality provided by Docker Compose, why move your containers to Kubernetes?

Answer:

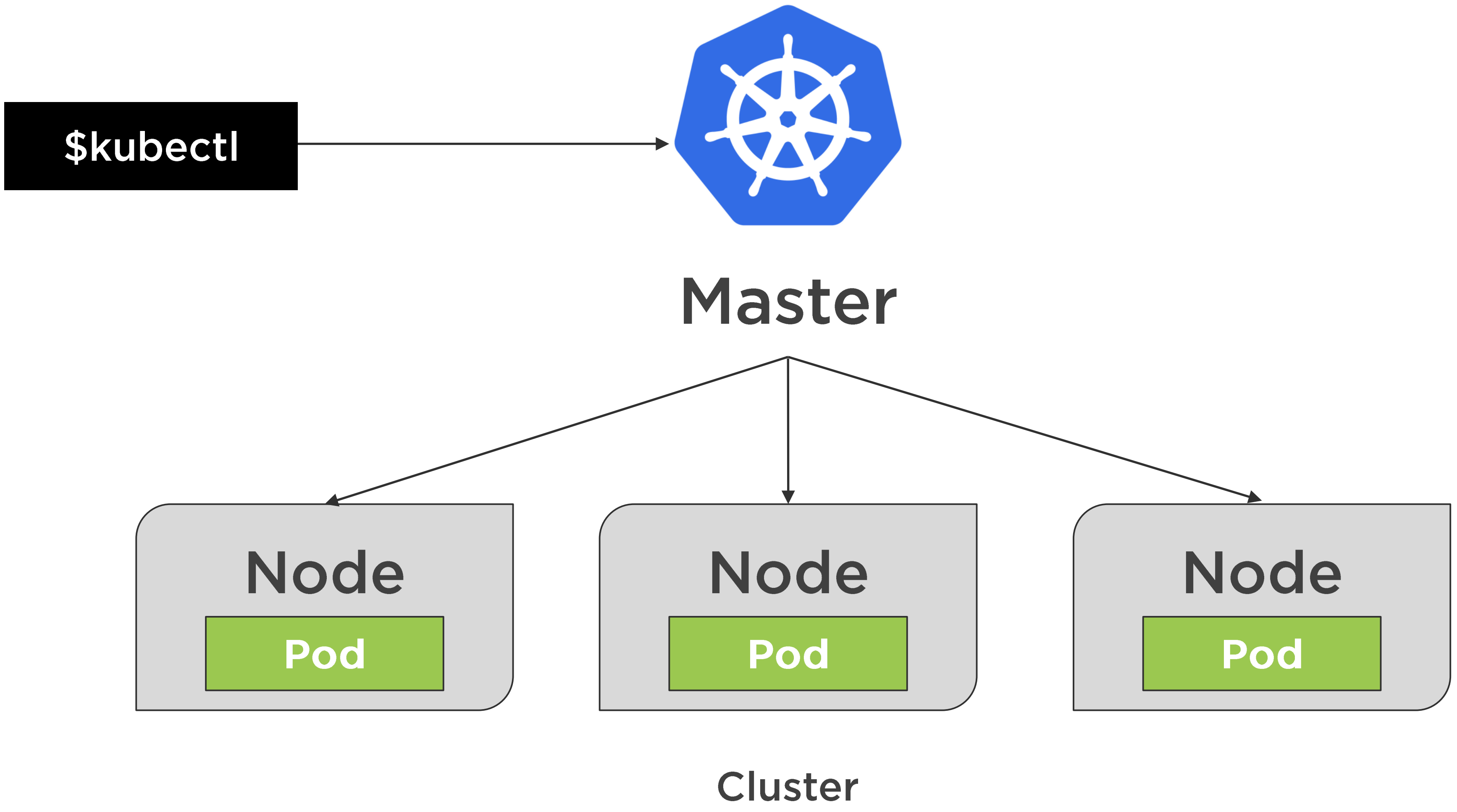
Kubernetes can scale containers across multiple nodes, provide container monitoring and healing functionality, offer robust deployment options, and much more.

Kubernetes Review

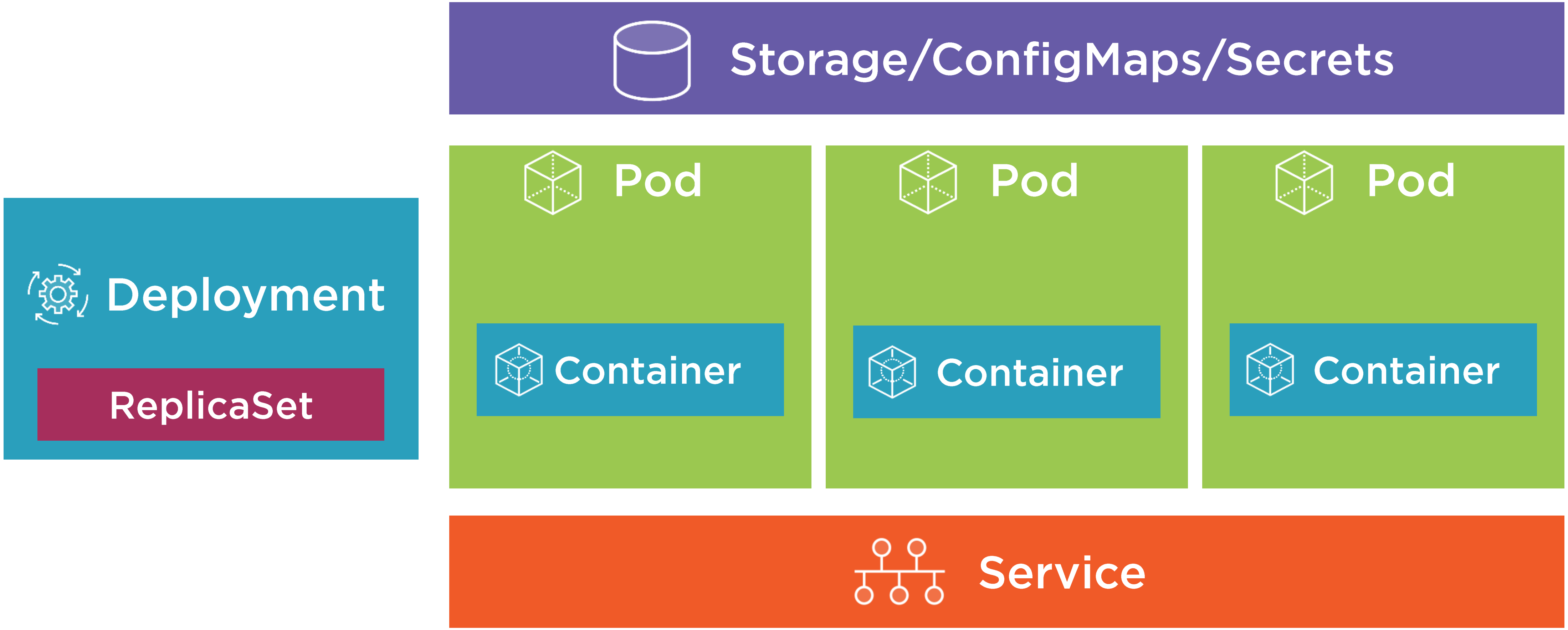
Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications.

Kubernetes Documentation

The Role of Kubernetes



Kubernetes Resources


Storage/ConfigMaps/Secrets

Deployment

ReplicaSet

Pod

Container

Pod

Container

Pod

Container

Service

Defining a Kubernetes Deployment with YAML

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: my-nginx
    tier: frontend
spec:
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: my-nginx
        image: nginx:alpine
```

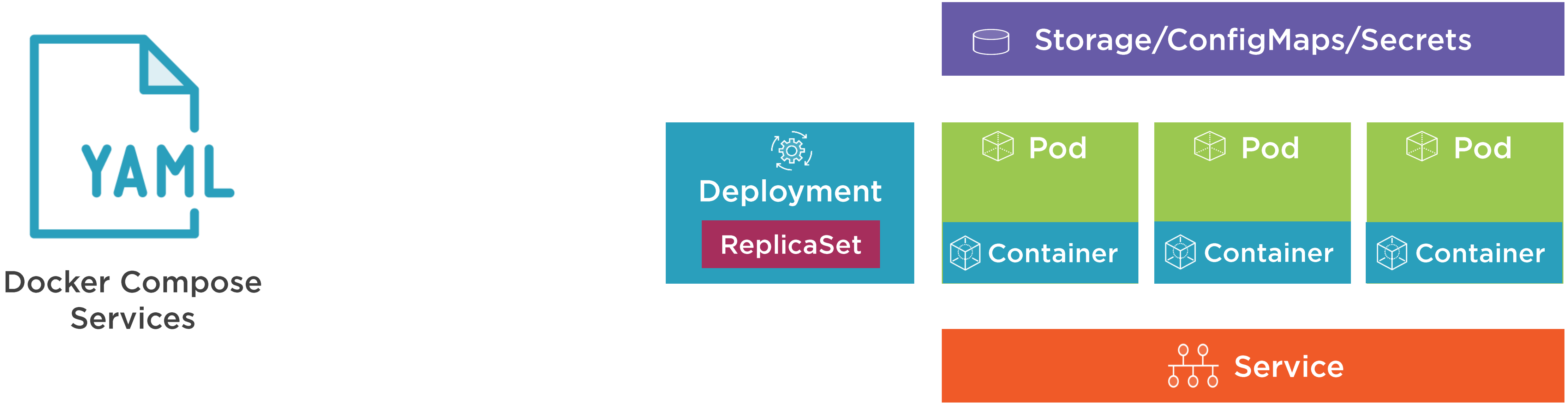
- ◀ Kubernetes API version and resource type (Deployment)
- ◀ Metadata about the Deployment
- ◀ The selector is used to "select" the template to use (based on labels)
- ◀ Template to use to create the Pod/Containers (note that the selector matches the label)

Key kubectl Commands

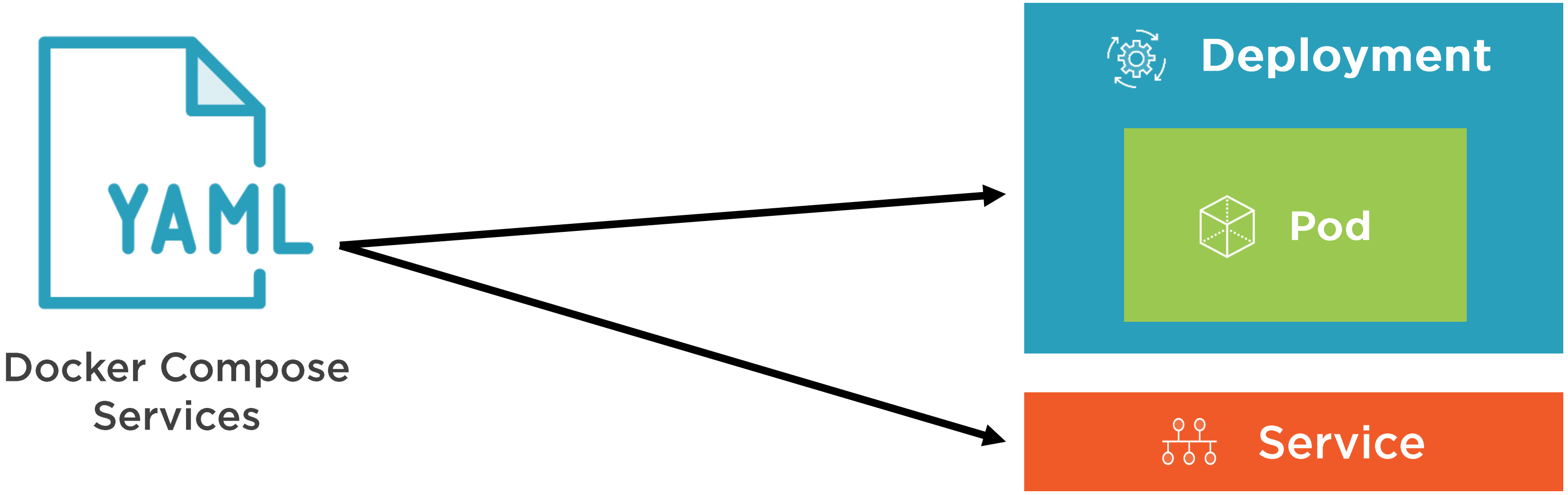
```
kubectl get all

kubectl create –f resource.yml

kubectl apply –f resource.yml

kubectl port-forward [pod-name] [ports]
```

Mapping Docker Compose Services to Kubernetes Resources

Docker Compose and Kubernetes

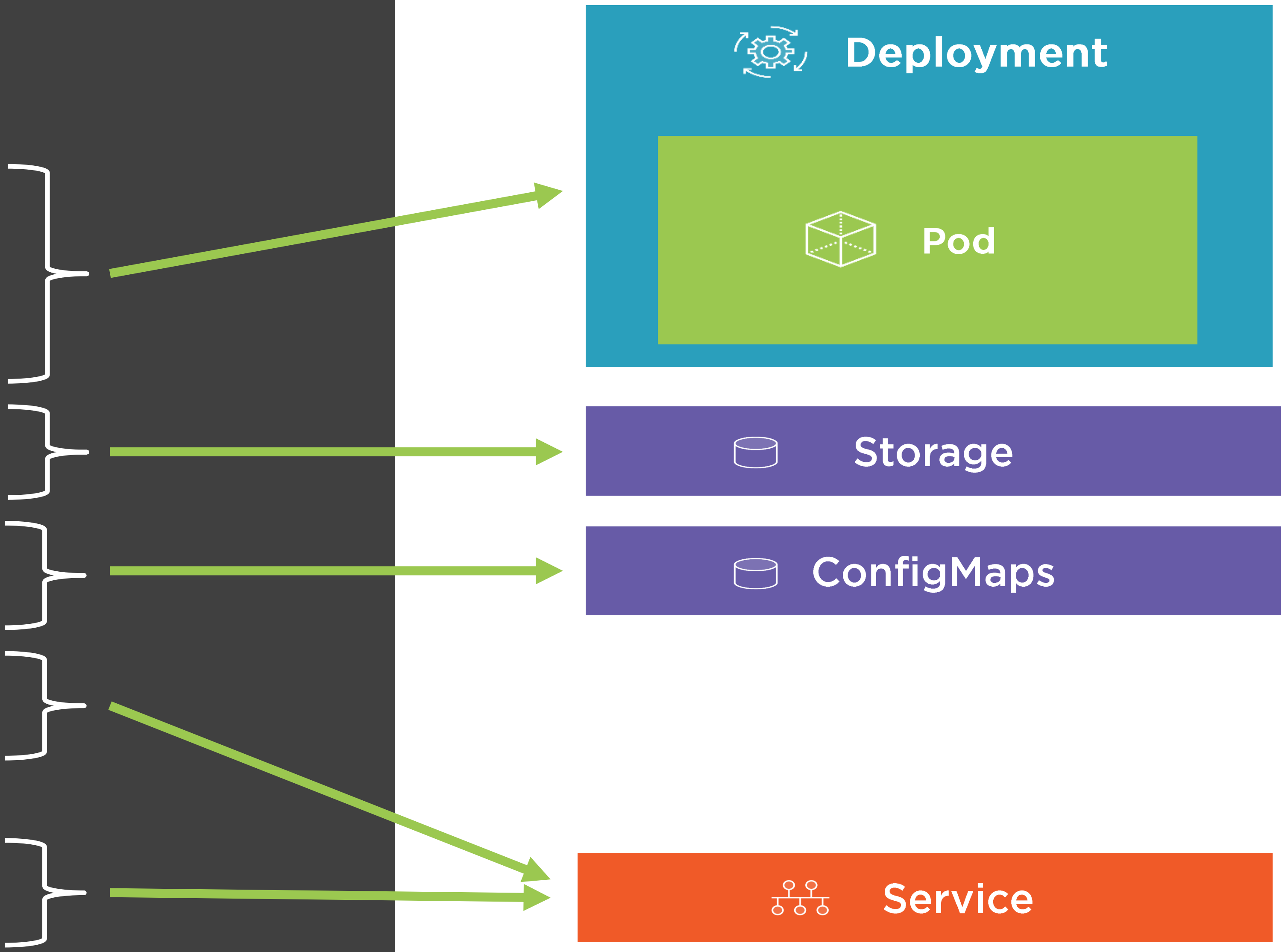


Mapping Docker Compose to Kubernetes



Mapping Docker Compose Services to Kubernetes Resources

```
version: "3.x"
services:
  node:
   container_name: nodeapp
   image: yourAccount/nodeapp
   ports:
    - "80:80"
   volumes:
    - .:/var/www/logs
   env_file:
    - ./nodeapp.env
   networks:
    - nodeapp-network

networks:
 nodeapp-network
  driver: bridge
```



Summary

Docker Compose can be used to:

- Build images
- Orchestrate running multiple containers
- Tear down multiple containers
- View container logs
- More…

Kubernetes can be used to:

- Run multiple containers using Pods
- Monitor and scale containers
- Heal or replace unhealthy Pods/containers
- Much more…

Where Do We Go From Here?

Docker Compose to Kubernetes Options

Use Docker Compose

Use a Tool to Convert

Perform a Custom Conversion