Performing Blue-Green Deployments

Dan Wahlin
WAHLIN CONSULTING

@danwahlin www.codewithdan.com

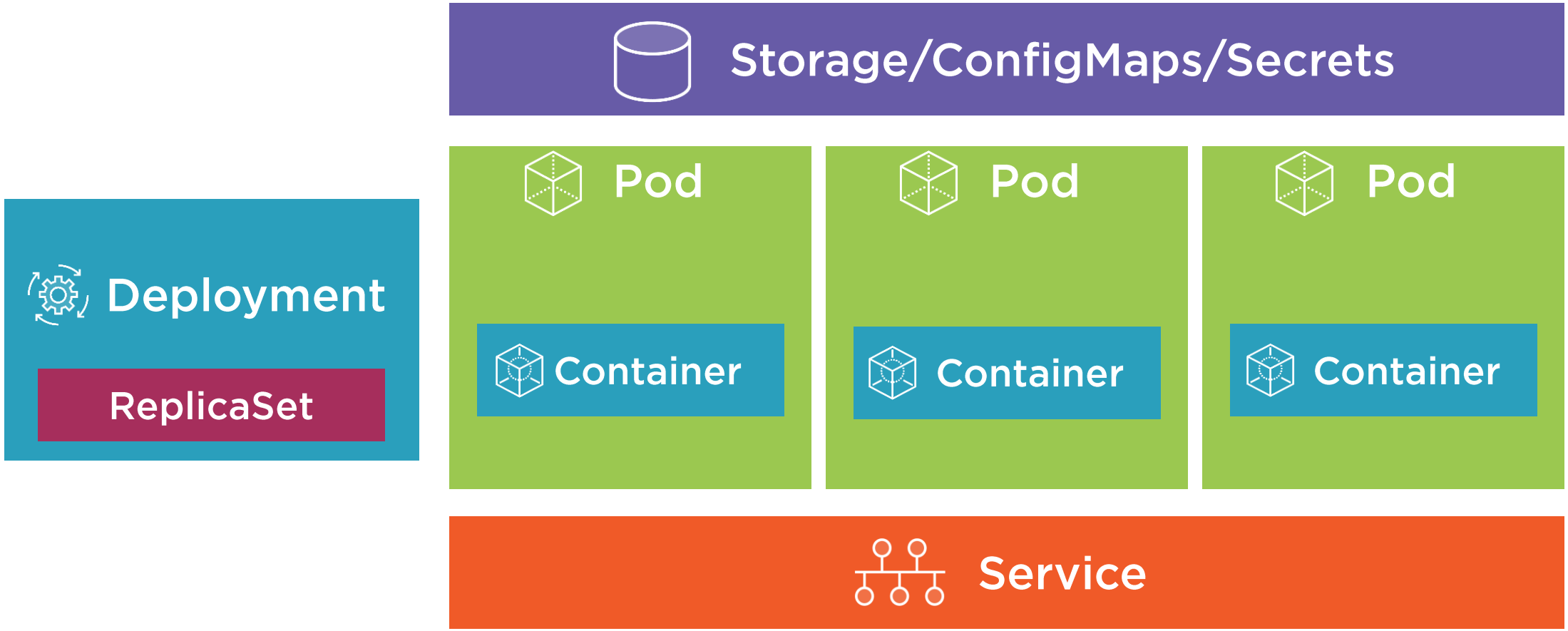
Module Overview

Understanding Blue-Green Deployments

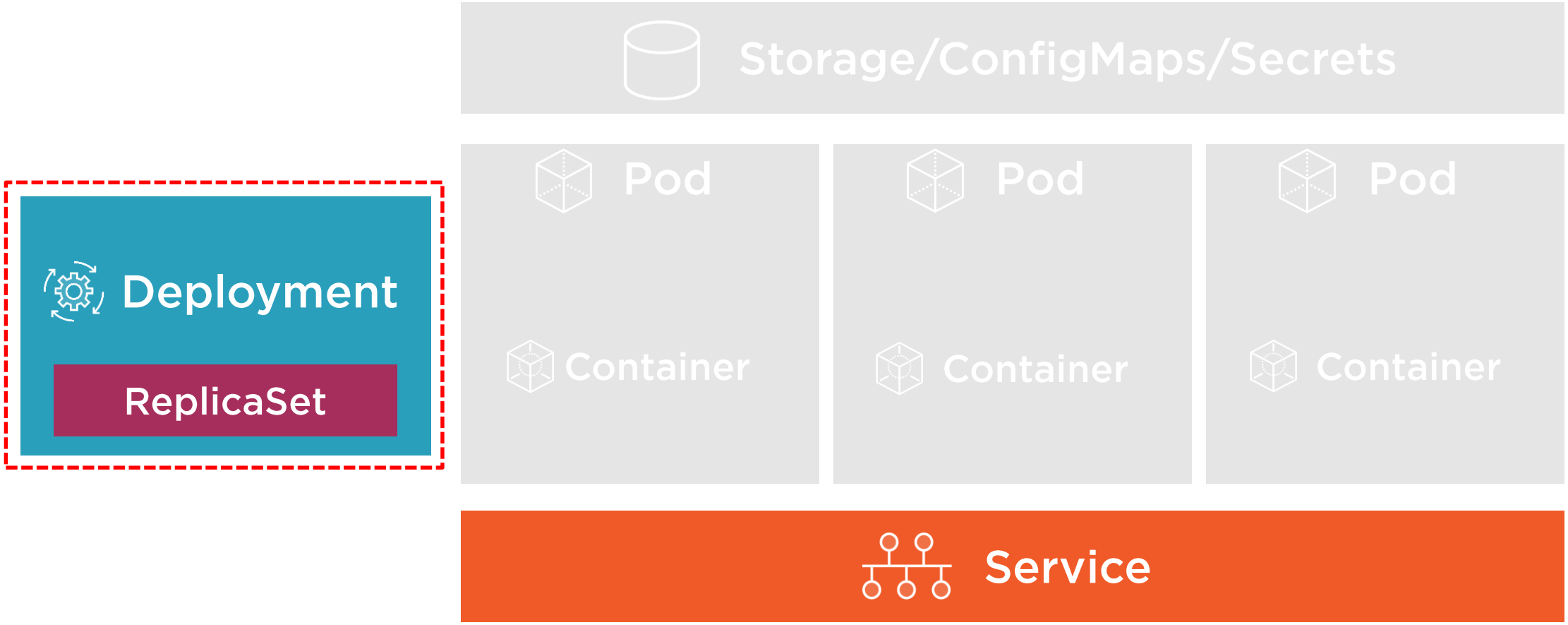
Creating a Blue-Green Deployment

Blue-Green Deployments in Action

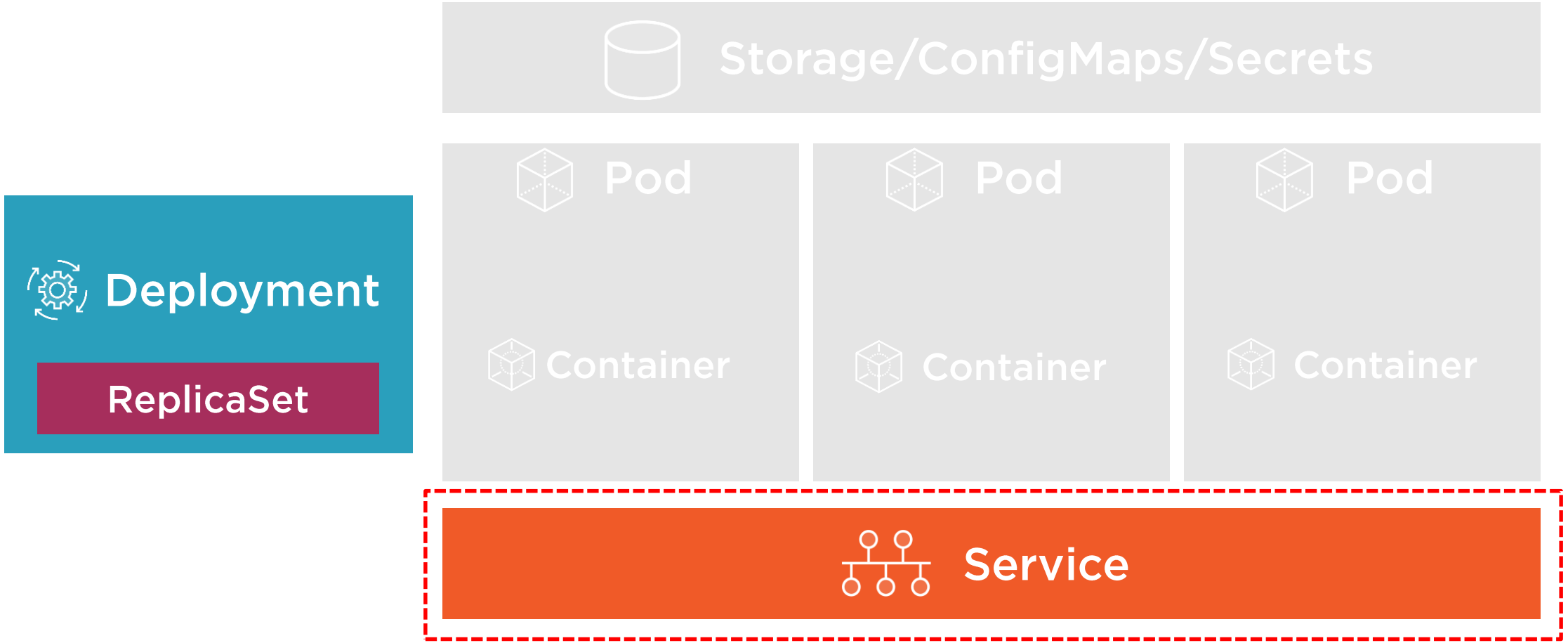
Kubernetes Resources



Kubernetes Resources



Kubernetes Resources



Understanding Blue-Green Deployments

Have you ever deployed an application to production and experienced problems? (don't laugh too much ☺)

"A blue/green deployment is a change management strategy for releasing software code."

~ TechTarget.com

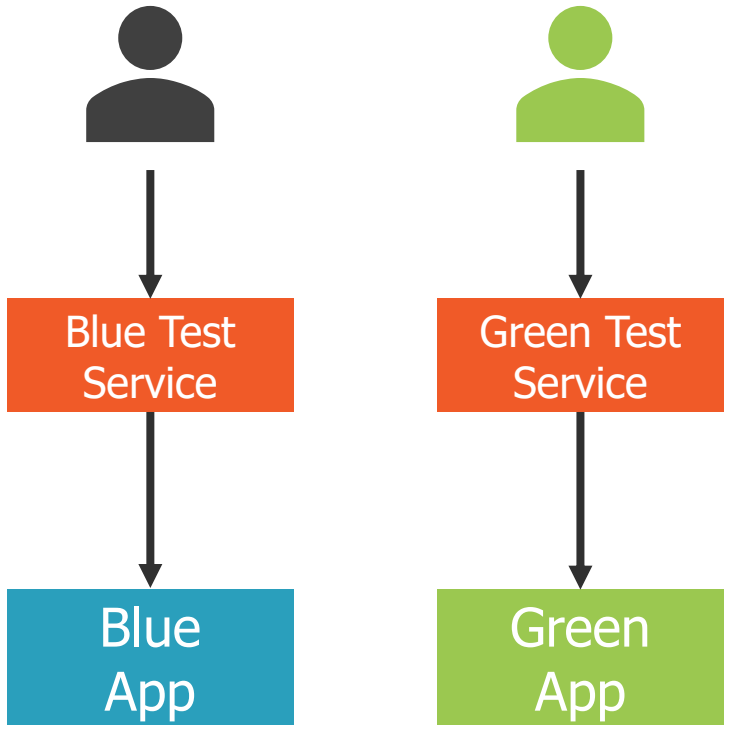
"A blue/green deployment is a change management strategy for releasing software code. Blue/green deployments, which may also be referred to as A/B deployments require two identical hardware environments that are configured exactly the same way. While one environment is active and serving end users, the other environment remains idle."

~ TechTarget.com

"A blue/green deployment is a change management strategy for releasing software code. Blue/green deployments, which may also be referred to as A/B deployments require two identical hardware environments that are configured exactly the same way. While one environment is active and serving end users, the other environment remains idle."

~ TechTarget.com

Blue-Green Deployments



Strategy for checking the viability of a deployment before it's publicly available

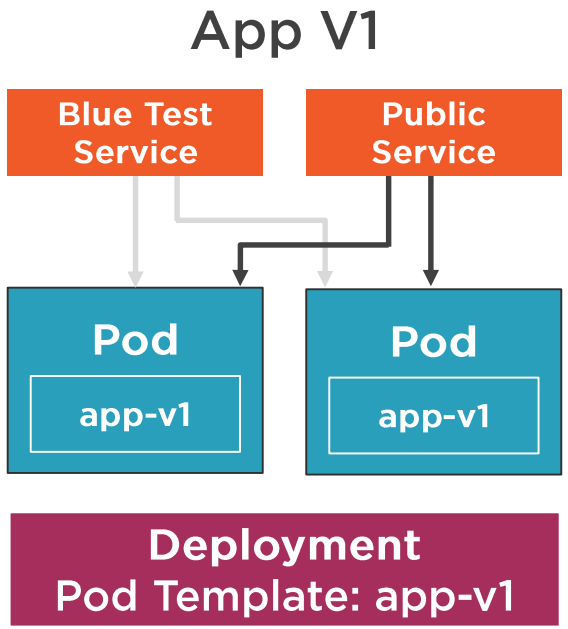
Run two identical production environments at the same time

New application (green) is deployed alongside the old application (blue)

Traffic routed from blue to green when checks pass

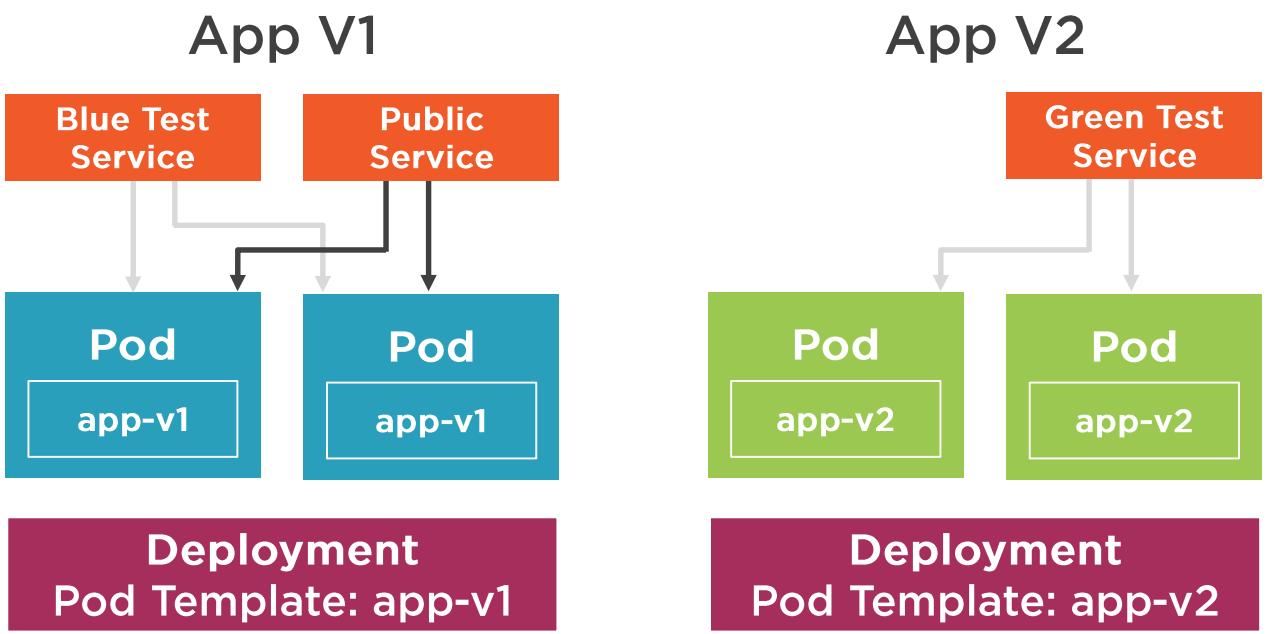
Blue-Green Deployments

1 **Create BLUE Deployment and Services**



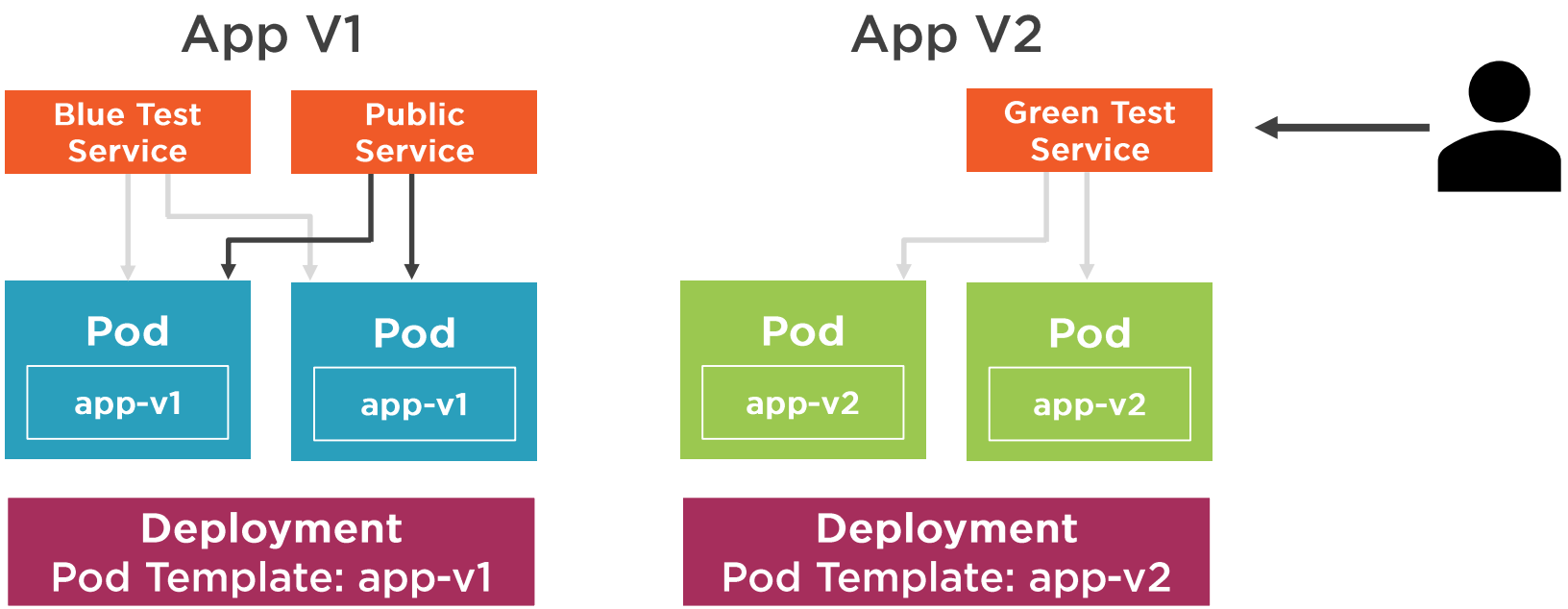
Blue-Green Deployments

2 Create GREEN Deployment and Service



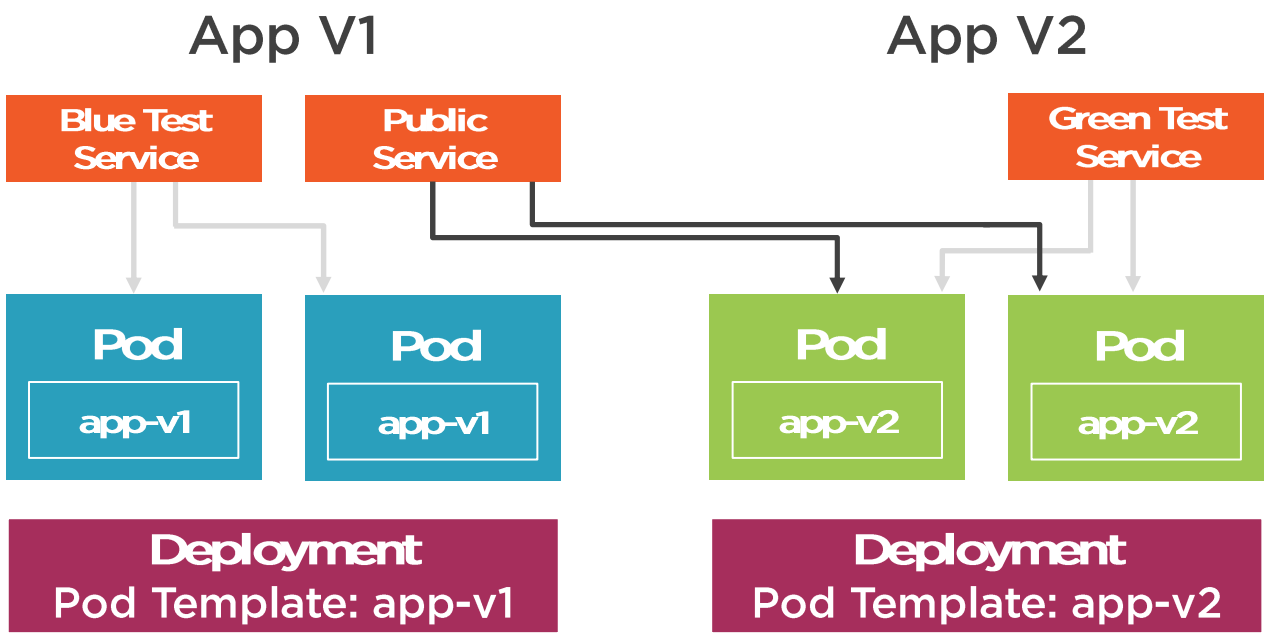
Blue-Green Deployments

3 Test GREEN Pods



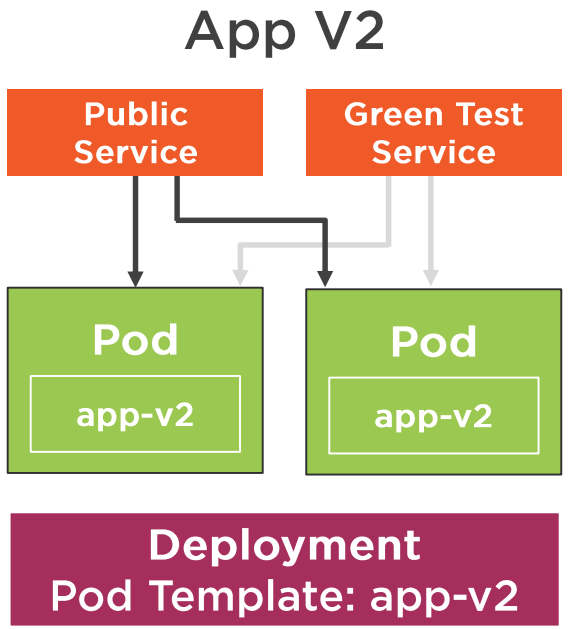
Blue-Green Deployments

4 Change public Service from BLUE to GREEN

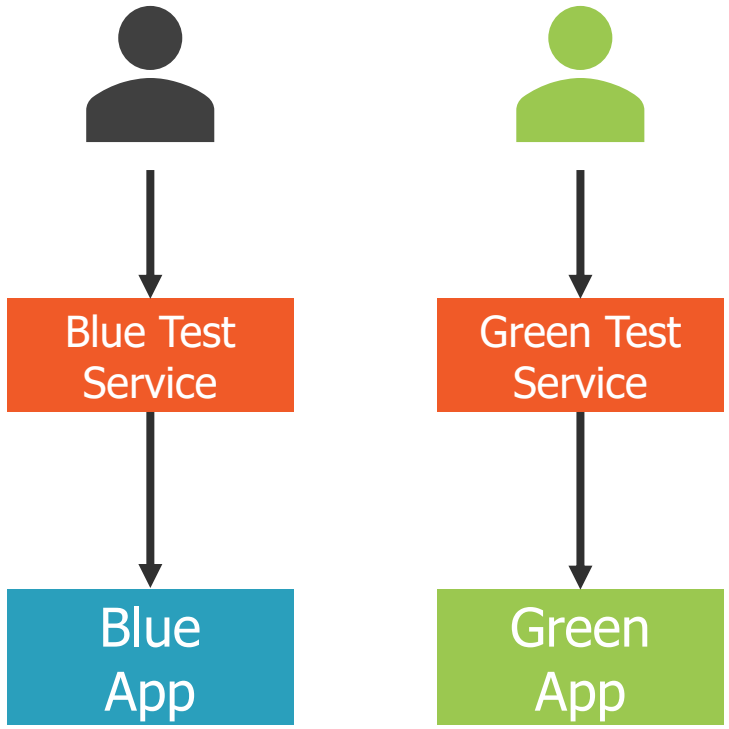


Blue-Green Deployments

5 Remove BLUE Deployment and Service



Blue-Green Deployment Considerations

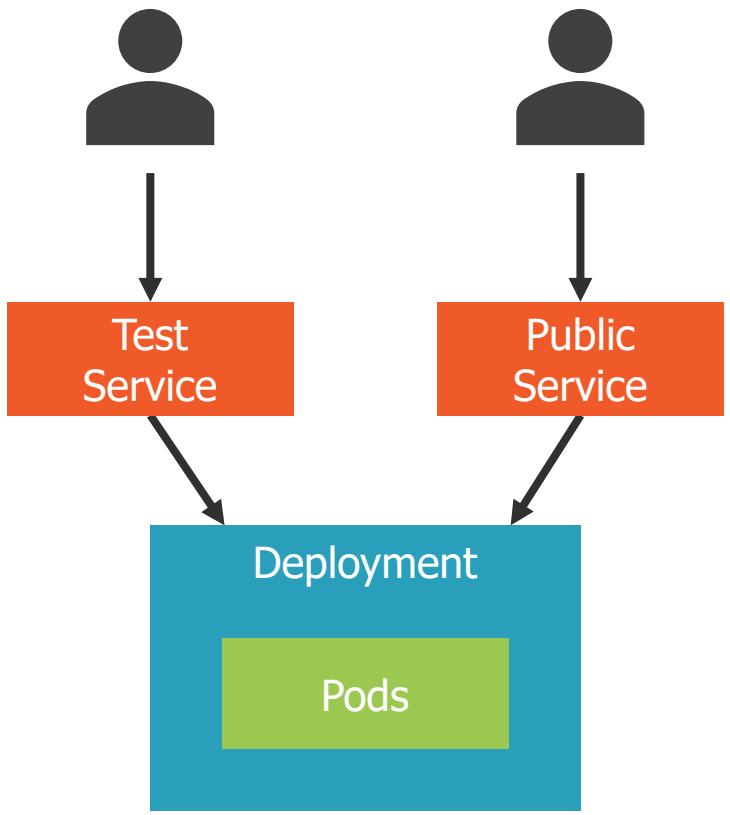


Key considerations:

- How many Pods are being deployed to each environment?
- How much memory is required to run the Pods?
- What are the CPU requirements?
- Other considerations (volumes, sessions, node affinity, etc.)

Creating a Blue-Green Deployment

Blue-Green Deployment Resources

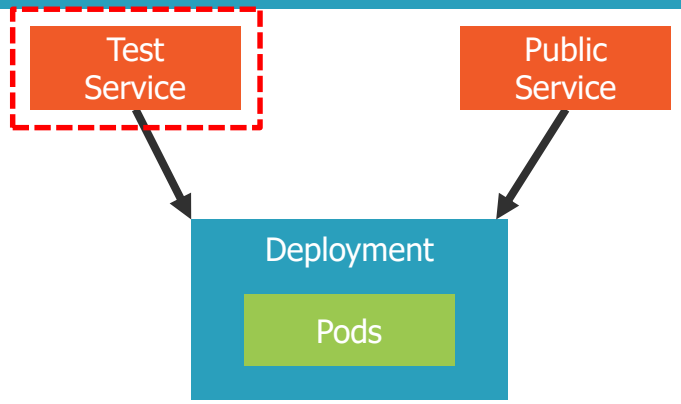


A Blue-Green Deployment involves 3 main Kubernetes resources:

- Test Service
- Public Service
- Deployment

Defining a Test Service

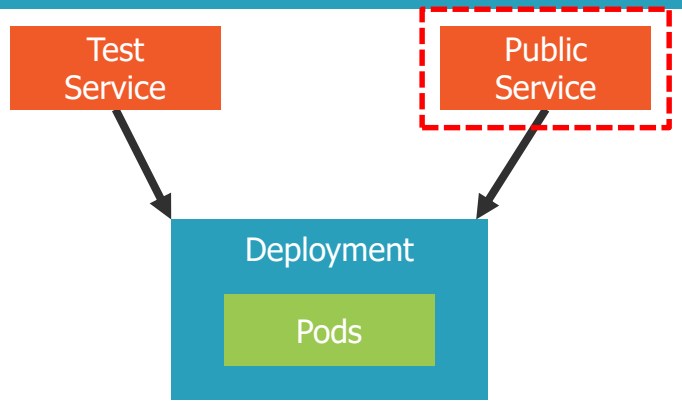
```yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-blue-test
  labels:
    app: nginx
    role: blue-test
    env: test
spec:
  type: LoadBalancer
  selector:
    app: nginx
    role: blue
  ports:
    - port: 9000
      targetPort: 80
```



- Role that the service plays (blue-test for test environment)
- Service will apply to Pods with these labels (note the role: blue)
- Expose port 9000 (test port)

Defining a Public Service

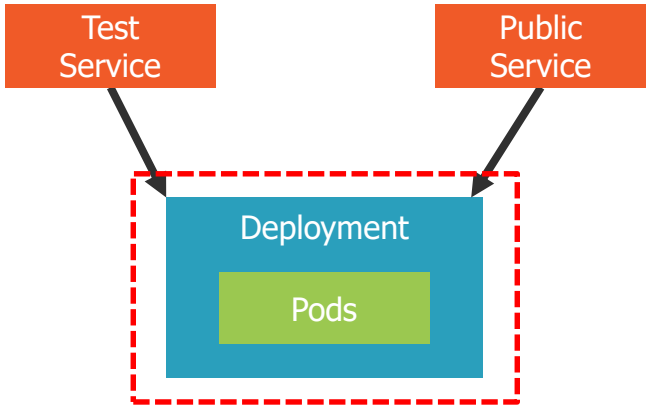
```yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  labels:
    app: nginx
    role: blue
    env: prod
spec:
  type: LoadBalancer
  selector:
    app: nginx
    role: blue
  ports:
    - port: 80
      targetPort: 80
```



- Role that the service plays (blue for production environment)
- Service will apply to Pods with these labels (note the role: blue)
- Expose port 80 (public port)

Defining a Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-blue
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
      role: blue
  template:
    metadata:
      labels:
        app: nginx
        role: blue
    spec:
      containers:
      - name: nginx-blue
        image: nginx:1.x.x-alpine
        ports:
        - containerPort: 80
```



- Apply to Pods with these labels (note the role: blue)
- Pod labels (note role: blue)
- Container image

Changing From Blue to Green

Once a GREEN deployment has been successfully rolled out and tested, change the public service's selector to "green"

```
selector:
  app: nginx
  role: green
```

Change role to green in public service's selector

```
# Apply changes made to service's YAML (declarative)
kubectl apply -f file.service.yml

# Change Service's selector to green (imperative)
kubectl set selector svc [service-name] 'role=green'
```

Blue-Green Deployments in Action – The Blue Deployment

Blue-Green Deployments in Action – The Green Deployment

Summary

Blue-Green Deployments allow two environments to be deployed at the same time

Provides a way to test a new version of an application before switching over to it

Works by changing the "blue" Service's selector to point to the "green" Deployment