

First Steps on Kubernetes

...

MindSwap - Mindera, 28 Feb

Who Am I?

- Full Cycle Developer
- Tech Lead@Project A
 - Around Mindera for 5 years
- @pontoporpono
 - LinkedIn
 - Twitter
 - GitHub
- 20 years of backend development (mostly!)

Mindera



We use technology to build products we are proud of,
with people we love.

[Handbook](#)

What is Kubernetes? (k8s)

...

kubernetes.io

**Container
Orchestration
Tool**

**Control Plane
+
Nodes**

**Declarative
Config**

**Robust &
Resilient**

Extensible

**Managed
vs
Serviced**

Topics Covered

- Pod
- Deployment
 - Probes
 - Replicas
- Service
- Volumes & Secrets
- Related Topics

Setup

- Install kubectl (Homebrew, chocolatey, apt-get, download)
- Configure kubectl autocomplete - Google it, please!
- <https://github.com/pontoporpono/k8s-workshop>
- `./setup.sh workshop-[NUMBER]`
- `kubectl get nodes`

Pod

1 Pod == 1 Container

```
kubectl apply -f simple-container-pod.yaml
```

```
kubectl get pods
```

```
kubectl logs simple-container-workshop
```


Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: simple-container-workshop
spec:
  containers:
  - name: simple-container
    image: gcr.io/pontoporponeto/simple-container:latest
    resources:
      limits:
        cpu: 250m
      requests:
        memory: 250Mi
```

Deployment

Stateless Pods Management

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

```
kubectl port-forward simple-service-workshop-XXXXXX 8080
```

<http://localhost:8080/workshop/XX/simple>

```
kubectl describe deployment simple-service-workshop
```

Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simple-service-workshop
spec:
  selector:
    matchLabels:
      app: simple-service
  template:
    metadata:
      labels:
        app: simple-service
    spec:
      containers:
        - name: simple-container
          image: gcr.io/pontoporponeto/simple-service:latest
          resources:
            limits:
              memory: 50M
```

Liveness Probe

Readiness Probe

Exec

TCP

HTTP

periodSeconds

timeoutSeconds

failureThreshold

successThreshold

Probes

```
livenessProbe:  
  httpGet:  
    path: /workshop/simple  
    port: 8080  
  periodSeconds: 10  
  timeoutSeconds: 5  
  failureThreshold: 2
```

kubectrl port-forward simple-service-workshop-XXXXXXX 8080

<http://localhost:8080/workshop/shutdown>

<http://localhost:8080/workshop/simple>

Replicas

kubectl scale deployment simple-service-workshop --replicas=2

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simple-service-workshop
spec:
  selector:
    matchLabels:
      app: simple-service
  replicas: 2
  template:
    metadata:
      labels:
        app: simple-service
    spec:
      containers:
        - name: simple-container
          image: gcr.io/pontoporpono/simple-service:latest
```

Deployment Strategies

Source: <https://github.com/ContainerSolutions/k8s-deployment-strategies>

DEPLOYMENT STRATEGIES

When it comes to production, a ramped or blue/green deployment is usually a good fit, but proper testing of the new platform is necessary.

Blue/green and shadow strategies have more impact on the budget as it requires double resource capacity. If the application lacks in tests or if there is little confidence about the impact/stability of the software, then a canary, a/b testing or shadow release can be used.

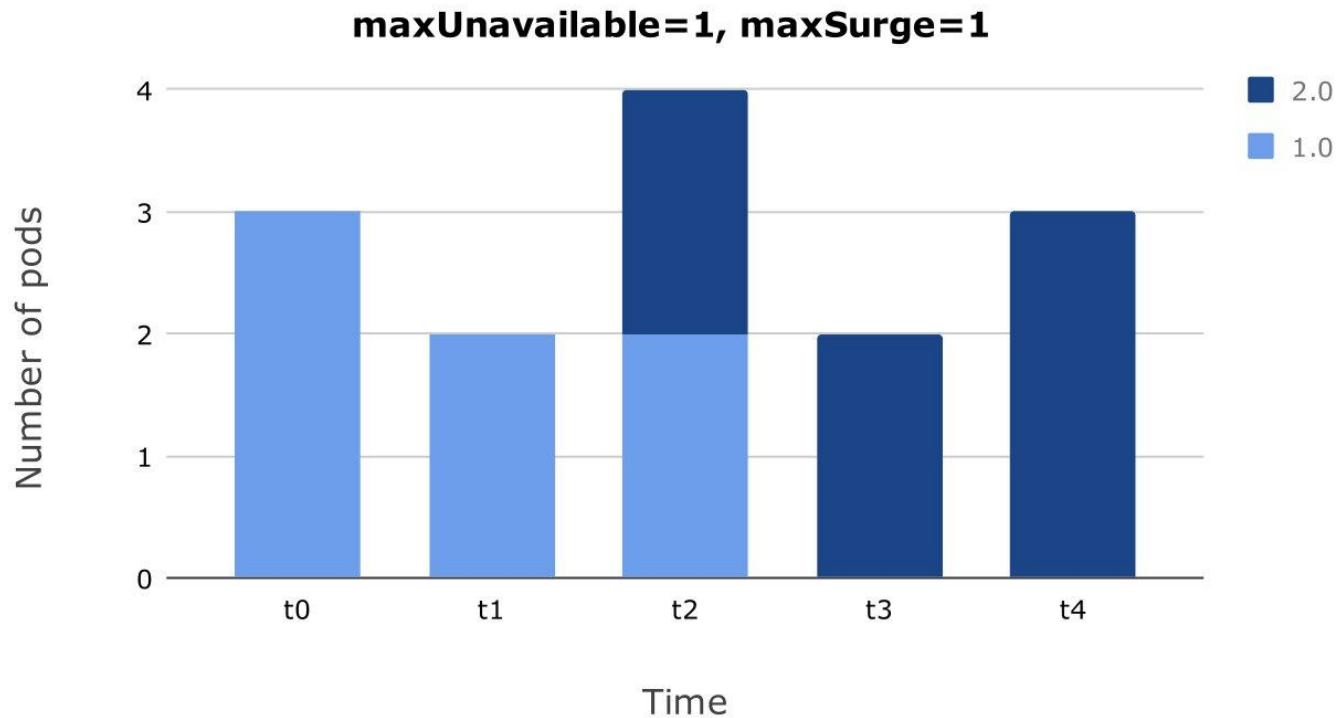
If your business requires testing of a new feature amongst a specific pool of users that can be filtered depending on some parameters like geolocation, language, operating system or browser features, then you may want to use the a/b testing technique.



Strategy	ZERO DOWNTIME	REAL TRAFFIC TESTING	TARGETED USERS	CLOUD COST	ROLLBACK DURATION	NEGATIVE IMPACT ON USER	COMPLEXITY OF SETUP
RECREATE version A is terminated then version B is rolled out	✗	✗	✗	■□□	■■■	■■■	□□□
RAMPED version B is slowly rolled out and replacing version A	✓	✗	✗	■□□	■■■	■□□	■□□
BLUE/GREEN version B is released alongside version A, then the traffic is switched to version B	✓	✗	✗	■■■	□□□	■■■	■■■
CANARY version B is released to a subset of users, then proceed to a full rollout	✓	✓	✗	■□□	■□□	■□□	■■■
A/B TESTING version B is released to a subset of users under specific condition	✓	✓	✓	■□□	■□□	■□□	■■■
SHADOW version B receives real world traffic alongside version A and doesn't impact the response	✓	✓	✗	■■■	□□□	□□□	■■■

Deployment Strategies

Source: <https://www.exoscale.com/syslog/kubernetes-zero-downtime-deployment/>



Service

```
kubectl apply -f simple-service-service.yaml
```

```
kubectl exec -it simple-container-workshop sh
```

```
curl -X GET http://service-endpoint/workshop/XX/simple
```

Service

```
apiVersion: v1
kind: Service
metadata:
  name: service-endpoint
spec:
  type: ClusterIP
  selector:
    app: simple-service
  ports:
    - port: 80
      name: workshop
      targetPort: 8080
```

Service

```
type: NodePort
```

```
#####
```

```
#####
```

```
nodePort: 320XX
```

<http://34.95.72.197/workshop/XX/simple>

Volumes - Secrets

```
kubectl create secret generic custom-configuration  
--from-file=placeholder.txt=placeholder.txt
```

```
kubectl get secrets custom-configuration -o yaml
```

```
kubectl apply -f
```

```
kubectl exec -it simple-service-workshop-XXXX sh  
more config/placeholder.txt
```

Helm

**Service
Mesh**

GitOps

**Developer
Platforms**

Serverless

**Certification
(CKAD)**

Feedback

