SSID: Guest

Password: BrokenWires@@2019

Getting Started with Kubernetes on AWS

Brought to you by the AWS Cape Town Cloud Support Team

Day 2

Agenda

- Basic Kubernetes primatives
- Exploring the cluster with kubectl
- Deploying an application with Kubernetes
- Designing your application for high availability
- Architecting your application using services
- Exposing your application to the world

Firstly...

Do you have a cluster?

Using a terminal in Cloud9, verify there are Worker Nodes in your cluster

Do you have a cluster?

Using a terminal in Cloud9, verify there are Worker Nodes in your cluster

Ooops, no, I don't have a cluster!

https://github.com/aws-els-cpt/eks

```
eksctl create cluster --ssh-access --version 1.14 --node-type t3.medium --name eks
```

Lets pull in the latest changes

```
Admin:~/environment $
```

Review

Questions:

Review

Questions:

- What are some advantages of using containers?

Review

Questions:

- What are some advantages of using containers?
- What's the most common runtime environment for containers?

Review

Questions:

- What are some advantages of using containers?
- What's the most common runtime environment for containers?
- What are some of the Linux features/tools used by containers?

Review

Questions:

- What are some advantages of using containers?
- What's the most common runtime environment for containers?
- What are some of the Linux features/tools used by containers?
- If we have Docker why we need something like Kubernetes?

Review

Questions:

- What are some advantages of using containers?
- What's the most common runtime environment for containers?
- What are some of the Linux features/tools used by containers?
- If we have Docker why we need something like Kubernetes?
- True or False - Is Kubernetes open source?

Review

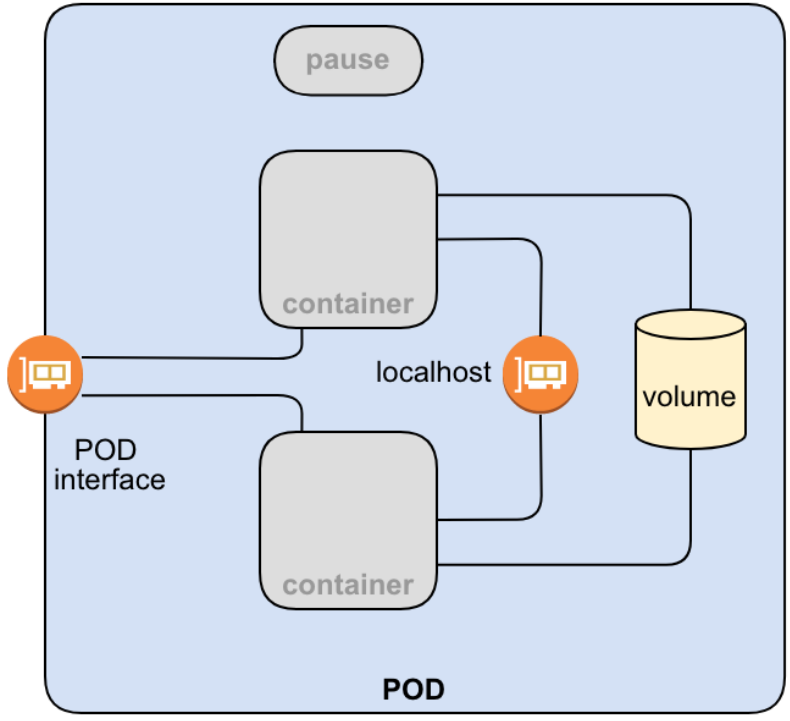
Questions:

- What are some advantages of using containers?
- What's the most common runtime environment for containers?
- What are some of the Linux features/tools used by containers?
- If we have Docker why we need something like Kubernetes?
- True or False - Is Kubernetes open source?
- Multiple Choice - Which are components of a Kubernetes Master?
  - API Server
  - Scheduler
  - Kubelet
  - Cloud Controller Manager
  - Garbage Collector

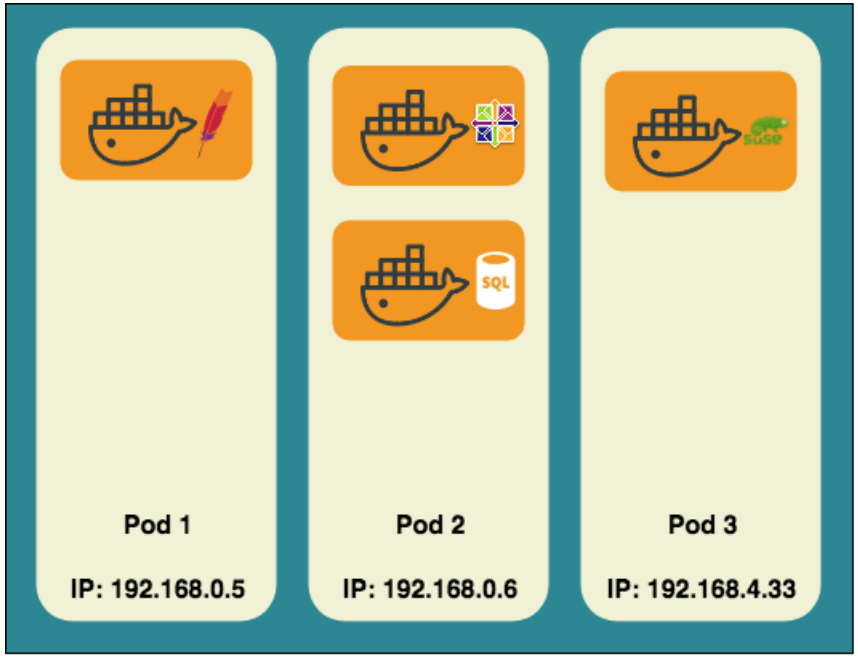
Pod

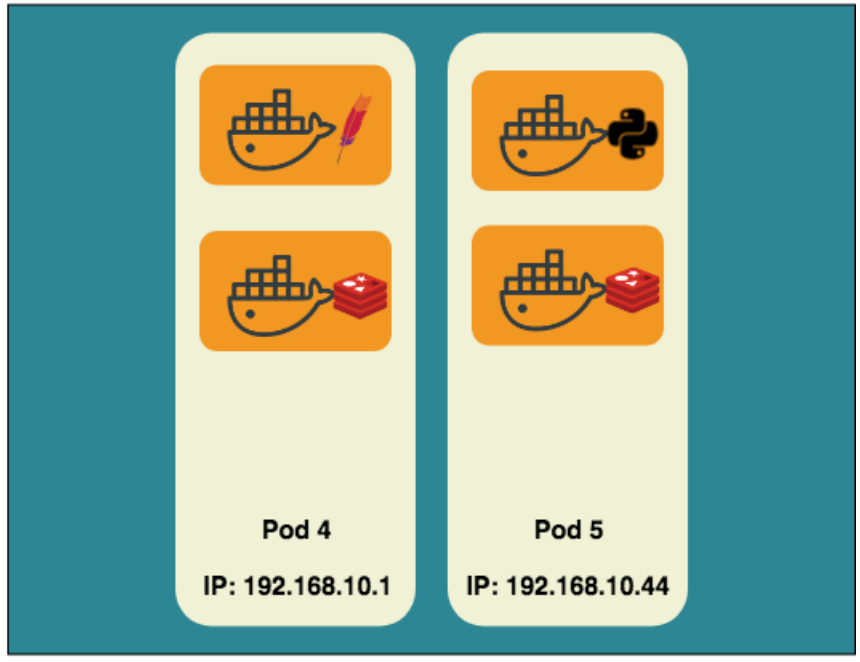
What is a Pod?

- The smallest building block of Kubernetes
- A Pod encapsulates the container(s) and resources needed to run the application
- A unit of deployment



What is a Pod?





Creating Pods in Kubernetes

Lab 2: Define a Pod

Pod definition

```
apiVersion: v1
kind: Pod
metadata:
  name: web-server
spec:
  containers:
  - name: container1
    image: nginx
```

Lab 2: Define a Pod

Pod definition

```
apiVersion: v1
kind: Pod
metadata:
  name: web-server
spec:
  containers:
  - name: container1
    image: nginx
```

File: `labs/02-pods/pod.yaml`

Lab 2: Creating a Pod

Send the definition to the cluster:

```
$ kubectl apply -f pod.yaml
pod/web-server created
```

Lab 2: Check the Pod

List and describe the Pod

```
# View the deployed pod
$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
web-server                        1/1     Running   0          10s

# See details of the pod
$ kubectl describe pod web-server
Name:               web-server
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               ip-10-0-100-30.eu-north-1.compute.internal/10.0.100.30
Start Time:         Fri, 27 Sep 2019 15:55:35 +0200
Labels:             <none>
[...]
```

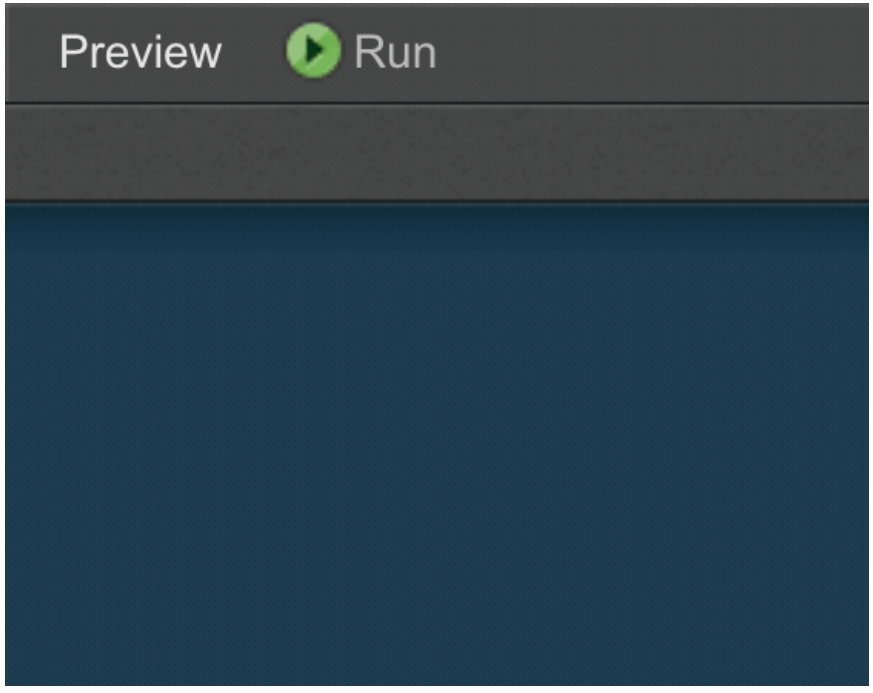
Lab 2: Check the Pod

Create a tunnel and connect to your Pod

```
kubectl port-forward pod/web-server 8080:80 &

curl localhost:8080
```

Or, connect using a browser and Cloud9



Lab 2: Clean up

We ran the port-forward in the background, lets clean it up before we move on

In the Cloud9 terminal to bring it back to the foreground:

```
fg

# Hit Ctrl + C to kill the port-forward
```

Your Turn

Or visit the link on GitHub:

https://github.com/aws-els-cpt/eks

Follow the steps in the `labs/02-pods/README.md` file.

Working with Pods

Lab 3: Working with Pods

Check the logs

```
$ kubectl logs web-server
```

Lab 3: Working with Pods

Check the logs

```
$ kubectl logs web-server
```

Connect

```
# run one command
kubectl exec web-server cat /etc/hostname

# run with console connected to pod
kubectl exec -it web-server -- bash
```

Lab 3: Working with Pods

Check the logs

```
$ kubectl logs web-server
```

Connect

```
# run one command
kubectl exec web-server cat /etc/hostname

# run with console connected to pod
kubectl exec -it web-server -- bash
```

You can delete it: *But keep it for now!*

```
kubectl delete pod web-server

# OR

kubectl delete -f labs/01_pod.yaml
```

Your Turn

Or visit the link on GitHub:

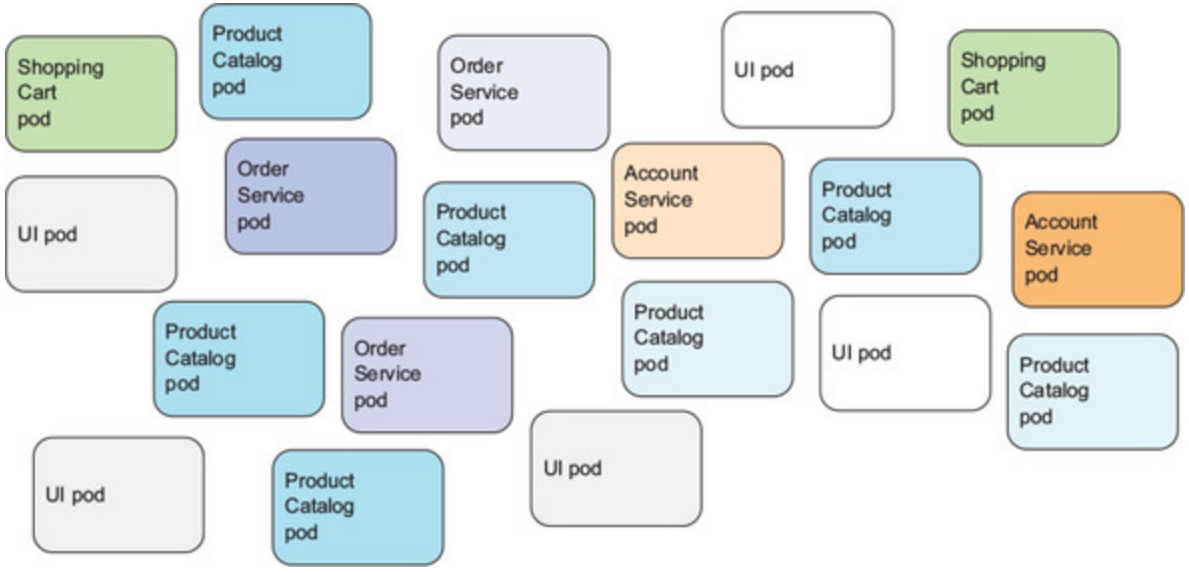
https://github.com/aws-els-cpt/eks

Follow the steps in the `labs/03-more-pods/README.md` file.

Labels

Labels

What if we are running a lot of pods?

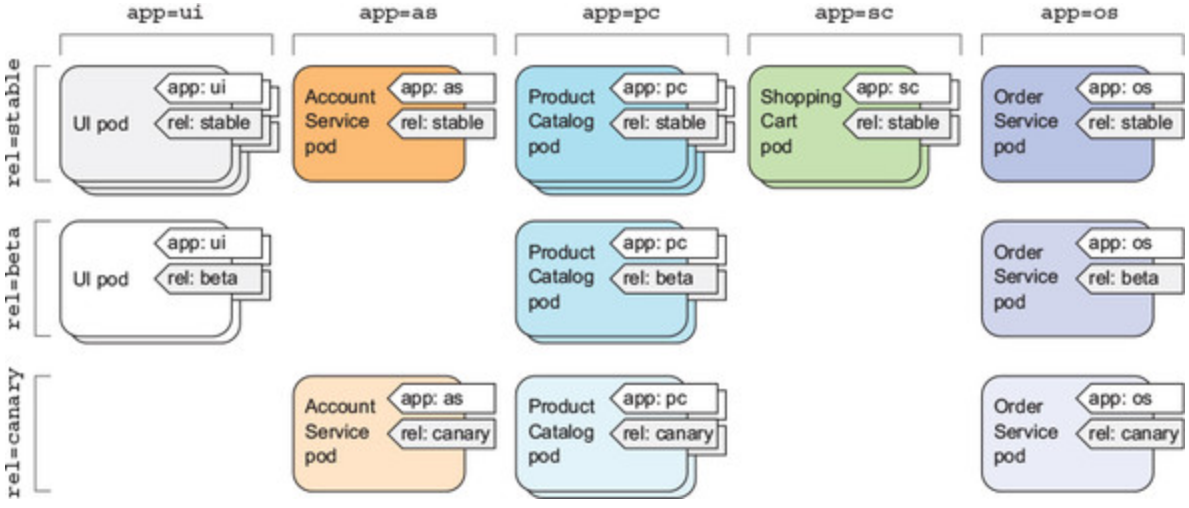


Picture from Kubernetes in action

Labels

Labels are key/value pair tags (like tags in AWS)

Can be used to query and organize resources



Picture from Kubernetes in action

Lab 4: Applying labels

Labels in the object definition

```
apiVersion: v1
kind: Pod
metadata:
  name: echo-server
  labels:
   env: training
   type: single_pod
spec:
  containers:
  - name: echo
    image: k8s.gcr.io/echoserver:1.4
```

Checking the labels

```
kubectl apply -f labs/03_labels.yaml

kubectl get pods --show-labels
```

File: labs/03_labels.yaml

Your Turn

Or visit the link on GitHub:

https://github.com/aws-els-cpt/eks

Follow the steps in the `labs/04-labels/README.md` file.

Links

Pods

- https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/
- https://kubernetes.io/docs/concepts/workloads/pods/pod/

Kubernetes in Action

- https://www.safaribooksonline.com/library/view/kubernetes-in-action/9781617293726/

Demos

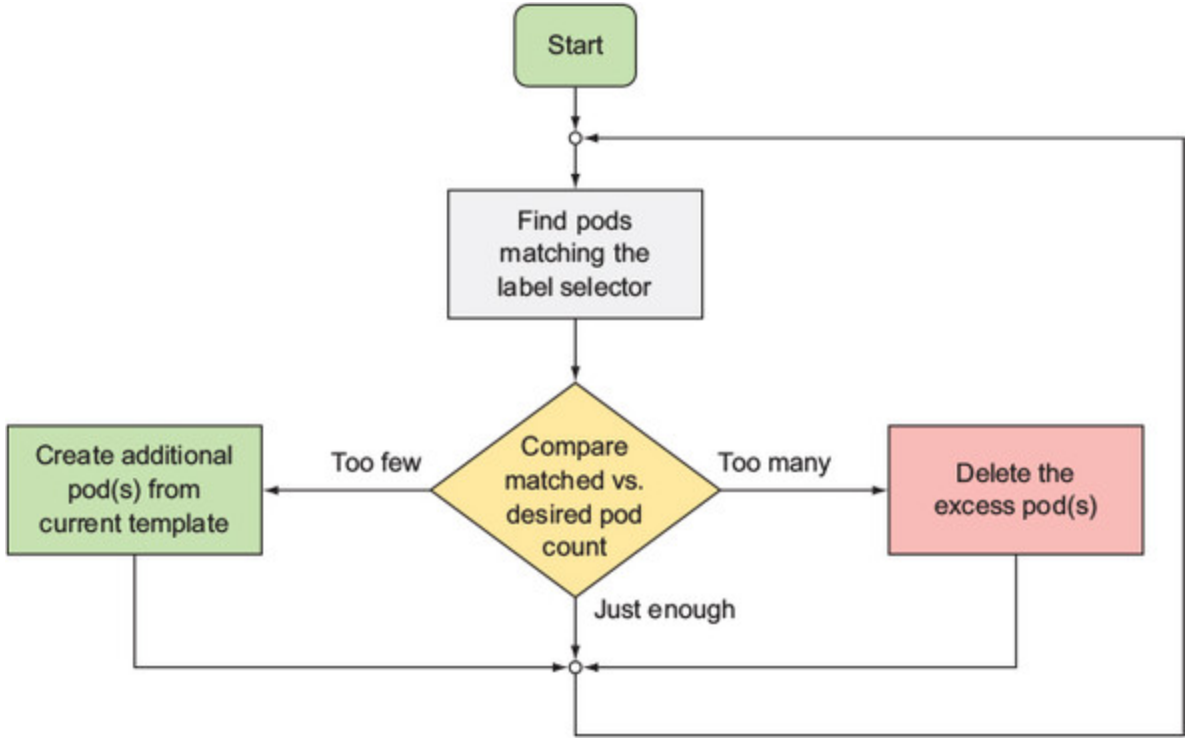
- https://eksworkshop.com/
- https://github.com/kubernetes/contrib/blob/master/micro-demos/

Controllers

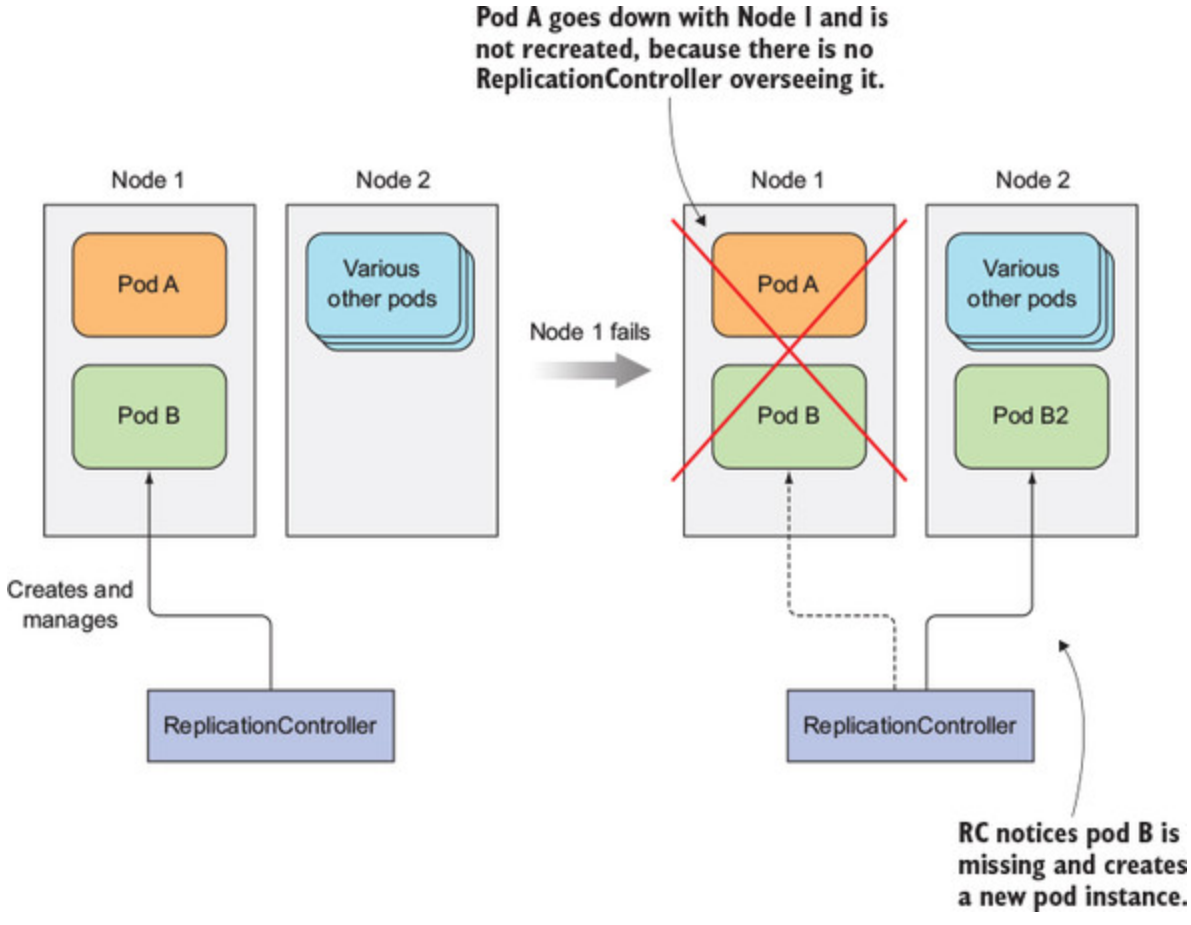
What controllers are and what they do?

- Resource responsible for managing pods
- Ensure pods are always running
- Replace missing and unhealthy pods
- Delete 'extra' pods
- Provide an easy way to scale the application
- Rely on Labels to account for the pods

What controllers are and what they do?



What controllers are and what they do?



Controllers

- Most commonly used controllers in Kubernetes:
  - ReplicationController
  - ReplicaSet the next generation of Replication Controllers
  - Deployments - preferred way to manage Replica Sets
  - DaemonSet
  - Jobs
  - CronJobs
  - StatefulSets

Deployments

Deployments

What is it?

- A Deployment controller provides declarative updates at controlled rate for Pods
- An easy way to deploy updates for existing applications
- Allows you to pause/resume deployments

Deployments

Comparing: Pods vs Deployment:

Single Pod:

```
apiVersion: v1
kind: Pod
metadata:
  name: web-server
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.7.9
```

Deployment:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-server-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
     app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
```

File: labs/04_deployment.yaml

Deployments

Lab 5: Create a Deployment

- Creating a Deployment

```
kubectl apply -f labs/04_deployment.yaml
```

- Checking the results

```
kubectl get deployments
kubectl get pod
kubectl get pod -l app=nginx
```

- Scale out the Deployment

```
kubectl scale deployment web-server-deployment --replicas=5
```

- Check the nginx server version

```
kubectl port-forward web-server-deployment-XXXXXX-YYYYY 8080:80
```

Deployments

Lab 5: Scale the Deployment

- Scale in the deployment. Let's be frugal

```
kubectl edit deployment web-server-deployment

# set spec.replicas to 3
```

- Checking the results

```
kubectl get pod

# OR

kubectl get pod -l app=nginx -L app
```

Deployments

Lab 5: Update the Deployment

- Update the deployment

```
kubectl set image deployment web-server-deployment nginx=httpd

# OR

kubectl edit deployment web-server-deployment
# change spec.template.spec.containers.image to httpd
```

- Checking the results

```
kubectl rollout status deployment web-server-deployment
# OR
kubectl get pod
# OR
kubectl get pod -l app=nginx -L app
```

- Check the Nginx server version now

```
kubectl port-forward web-server-deployment-XXXXXX-YYYYY 8080:80
```

Deployments

Your Turn

Or visit the link on GitHub:

https://github.com/aws-els-cpt/eks

Follow the steps in the `labs/05-deployments/README.md` file.

Services

Services

What's a service?

- Service is another layer on top of the pods
- Instead of connecting to the pods directly we connect to the service instead
- Very similar to a load balancer

Services

What's a service?

- Service is another layer on top of the pods
- Instead of connecting to the pods directly we connect to the service instead
- Very similar to a load balancer

But why?

- Pods are ephemeral
- Pods' IPs are dynamic
- A single application might contain several Pods

Services

What's a service?

- Service is another layer on top of the pods
- Instead of connecting to the pods directly we connect to the service instead
- Very similar to a load balancer

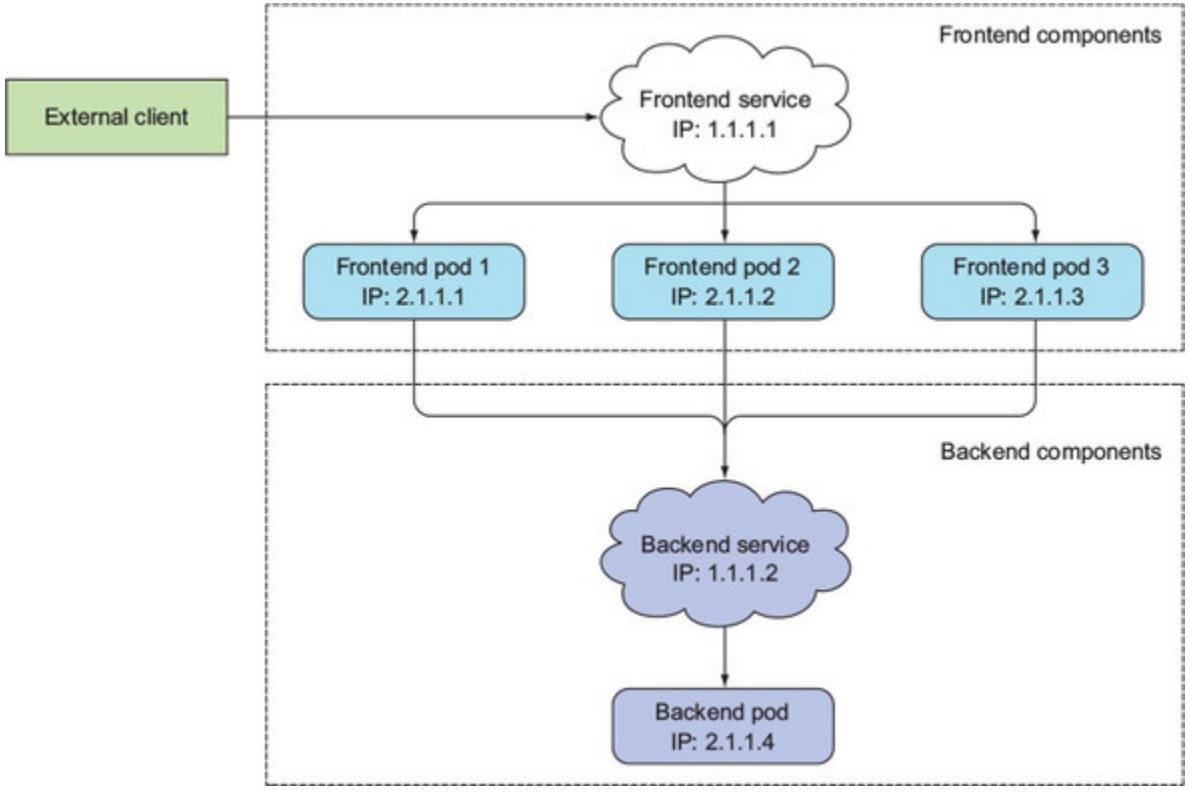
But why?

- Pods are ephemeral
- Pods' IPs are dynamic
- A single application might contain several Pods

So, how to reach an application?

Services

A Service is an abstraction layer which enables external traffic exposure, load balancing and service discovery



* from https://kubernetes.io/

Lab 6: Create a Service

```
apiVersion: v1
kind: Service
metadata:
  name: web-app
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: web-server
```

File: labs/06-services

Lab 6: Create a Service

```
apiVersion: v1
kind: Service
metadata:
  name: web-app
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: web-server
```

- Creating a service

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

File: labs/06-services

Lab 6: Create a Service

```
apiVersion: v1
kind: Service
metadata:
  name: web-app
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: web-server
```

- Creating a service

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

- Checking the results

```
kubectl get services
kubectl describe svc web-app
```

File: labs/06-services

Services

- There are different types of services:
  - ClusterIP is the default. Used for intra-cluster communication
  - LoadBalancer provisions a Load Balancer for you.

Connecting to your service:

```
kubectl edit svc web-app

# change spec.type from 'ClusterIP' to 'LoadBalancer'
```

- Checking the results

```
kubectl get svc web-app
```

- Now you should get the LB URL from the EC2 console and open it in your browser.

Your Turn

Or visit the link on GitHub:

https://github.com/aws-els-cpt/eks

Follow the steps in the `labs/06-services/README.md` file.

End of Day 2

Questions?

Backup Slides

Lab 5: Playing with Docker commands

- Find the node where your web-server Pod is running and ssh into it using the default ~/.ssh/id_rsa private key of the Cloud9 environment
- Try to play with the following docker commands to explore your pod:
  - `docker ps`
  - `docker exec`
  - `docker logs`
- What will happen if you stop the container with the `docker stop` command? Give it a try... In another terminal watch the pods with: `kubectl get pod web-server --watch`

Dashboard

Install the Dashboard

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recom
```

Install the Dashboard

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recom
```

Start a proxy to the API server

```
kubectl proxy &
```

Preview in Cloud9

Add the following to the end of the URL in the address bar:

/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

Install the Dashboard

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recom
```

Start a proxy to the API server

```
kubectl proxy &
```

Preview in Cloud9

Add the following to the end of the URL in the address bar:

/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

Get the token

```
aws-iam-authenticator token -i XXXXXX
```

{"kind":"ExecCredential","apiVersion":"client.authentication.k8s.io/v1alpha1","spec":{},"status":{"token":"k8s-aws-v1.aHR0cHM6.....IwYWViMGNjY2Nh"}}

Thank you!