

SSID: Guest

Password: BrokenWires@@2019

Getting Started with Kubernetes on AWS

Brought to you by the AWS Cape Town Cloud Support Team

Welcome and thank you!

Instructors



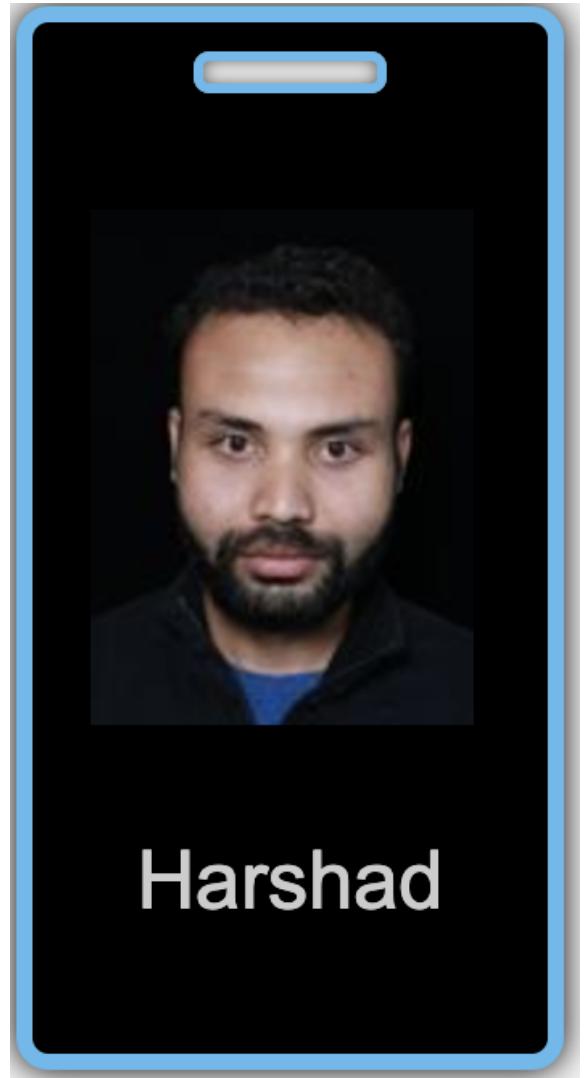
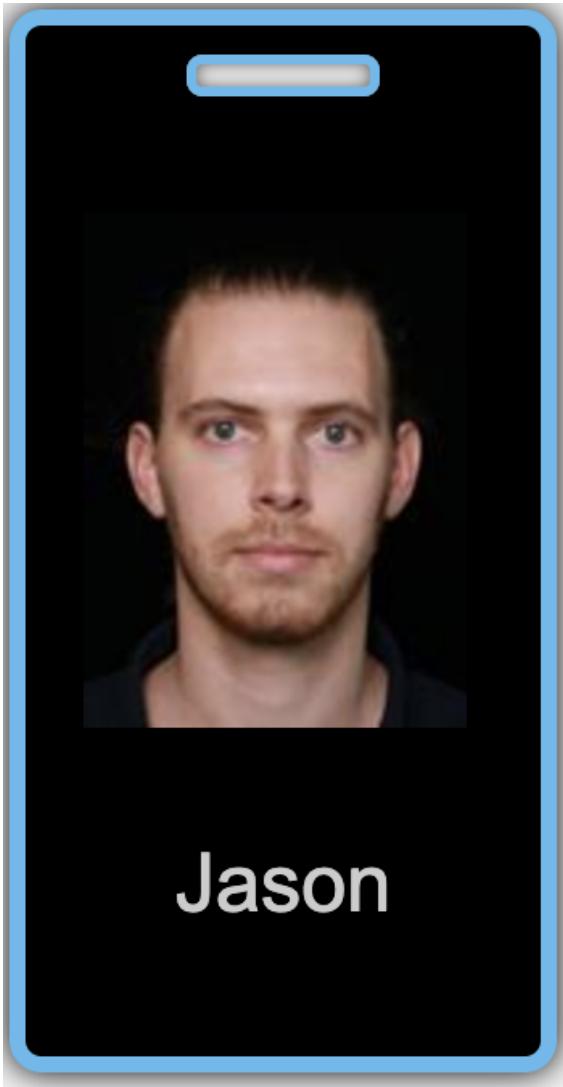
Nataizya



[View Custom Photo](#)

Junaid

Instructors



Helpers around the room



Chidiebube



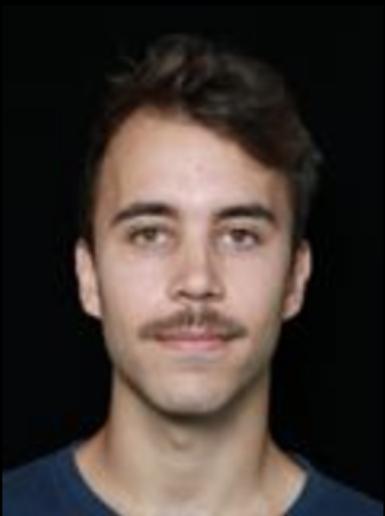
Mohammad
Iqbal



Patrick



Anderson



Felipe

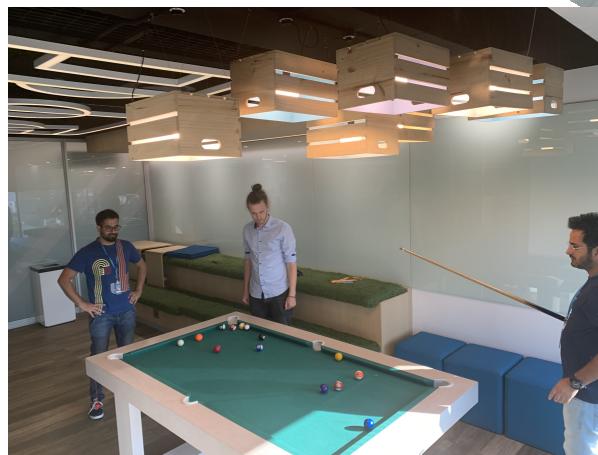


Arun

What is Cloud Support Engineering?

- Cloud Support Engineers are the front line of AWS' technical support
- Multiple teams, each with specific focus areas
- What do we do? A surprising amount:
 - Customer interactions
 - Training
 - Learning is part of the job
 - Hiring

The Team



Day 1

Agenda

- Introduction to Containers
- Docker Overview
- Kubernetes Explained
- EKS - Exploring and Deploying
- Environment Setup
- Launching a Kubernetes Cluster

Before we get started

Adding credits to your AWS Account and Setting up Environment



Adding credits to your AWS Account

1. Open the Credits page of the Billing and Cost Management console.
2. In the Promo Code box, type the promotional code.
3. In the box labeled "**Please type the characters as shown above**", type the code.
4. Choose Redeem.

Adding credits to your AWS Account

1. Open the Credits page of the Billing and Cost Management console.
2. In the Promo Code box, type the promotional code.
3. In the box labeled "**Please type the characters as shown above**", type the code.
4. Choose Redeem.

Note

The credit amount will cover running the resources we need for the sessions, other resources you may wish to use in your AWS account may not be covered by this.

You are responsible for deleting the resources on Monday. We'll remind you again next week.

Setting up the environment

Launching your Lab Environment

1. Cloud9 - this will be where you'll be performing the labs throughout the sessions.

Follow the steps provided in the README.md to setup your environment

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



Containers and Docker

Container Overview

- What is a container?

Container Overview

- What is a container?
- What is Docker? what benefits does it provide?

Container Overview

- What is a container?
- What is Docker? what benefits does it provide?
 - Consistent environment

Container Overview

- What is a container?
- What is Docker? what benefits does it provide?
 - Consistent environment
 - Portability of code, runtime, system tools, system libraries, etc.

Container Overview

- What is a container?
- What is Docker? what benefits does it provide?
 - Consistent environment
 - Portability of code, runtime, system tools, system libraries, etc.
 - Version control

Container Overview

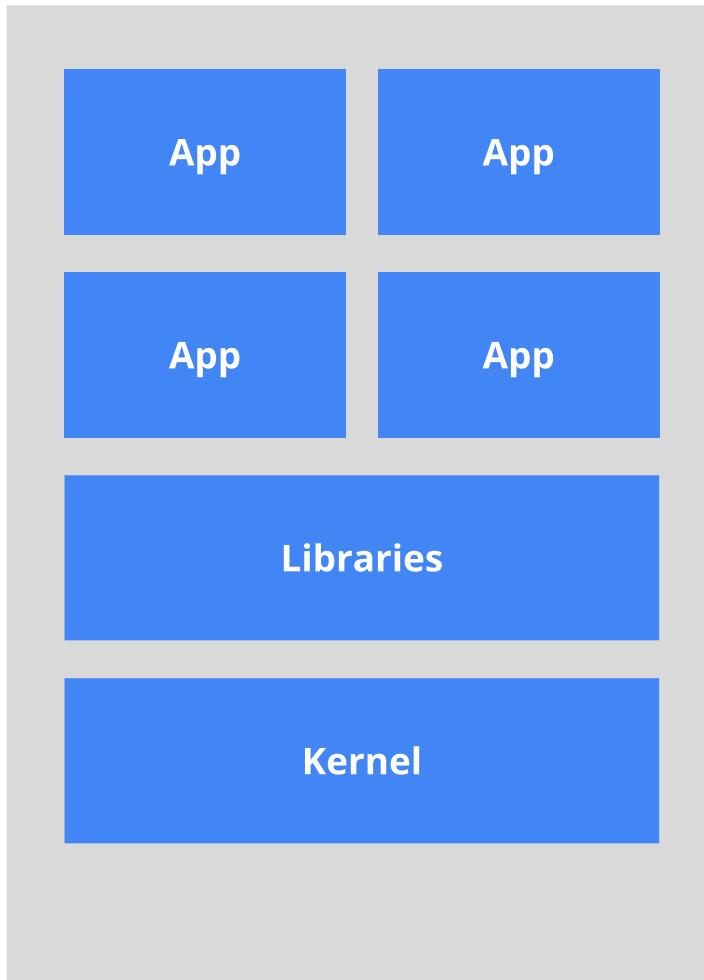
- What is a container?
- What is Docker? what benefits does it provide?
 - Consistent environment
 - Portability of code, runtime, system tools, system libraries, etc.
 - Version control
 - Lightweight

Container Overview

- What is a container?
- What is Docker? what benefits does it provide?
 - Consistent environment
 - Portability of code, runtime, system tools, system libraries, etc.
 - Version control
 - Lightweight
 - Microservices

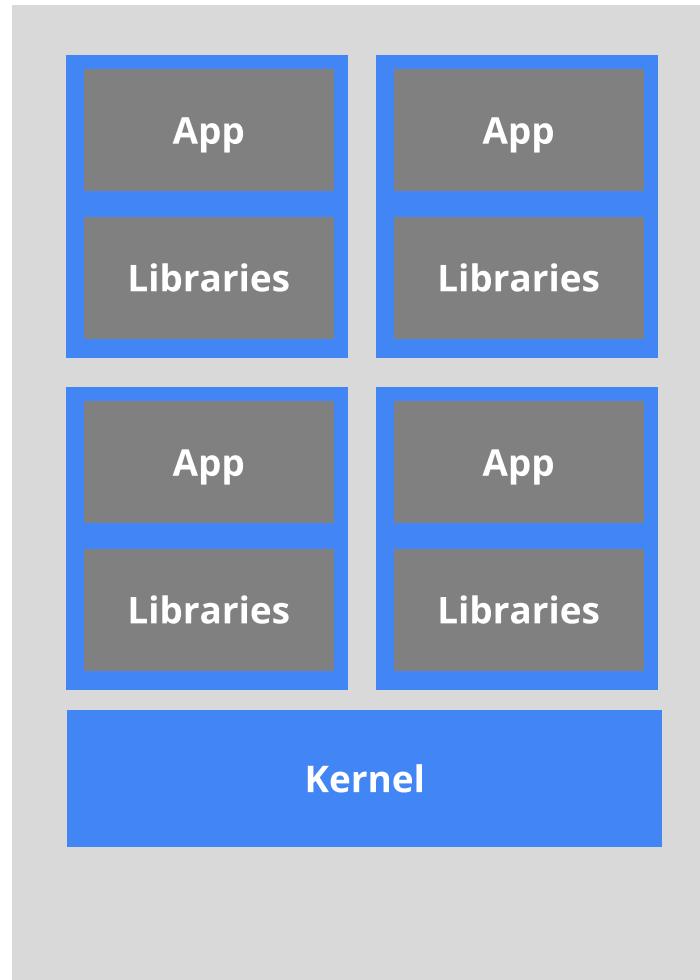
Why containers

The old way: Applications on host



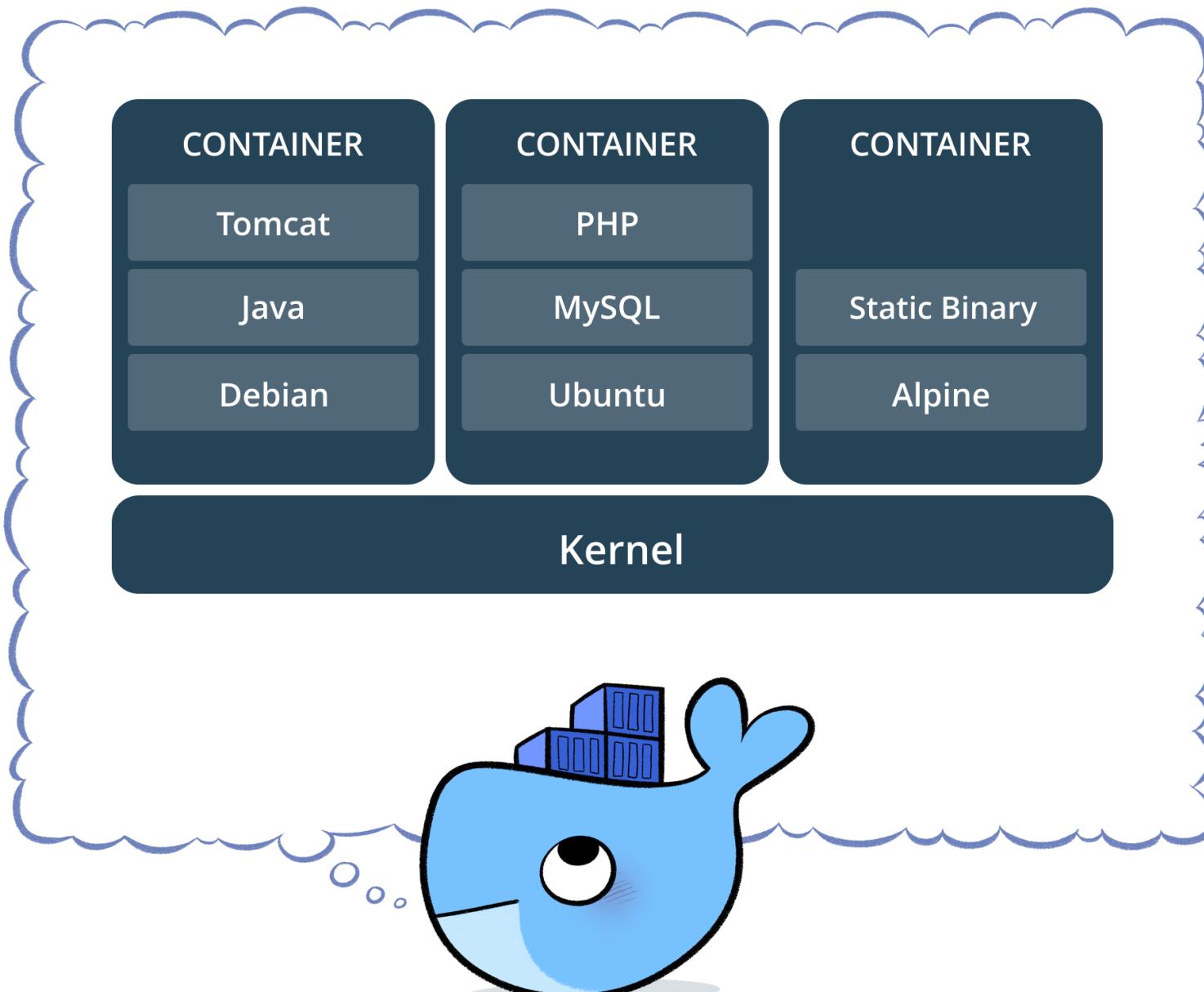
*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers

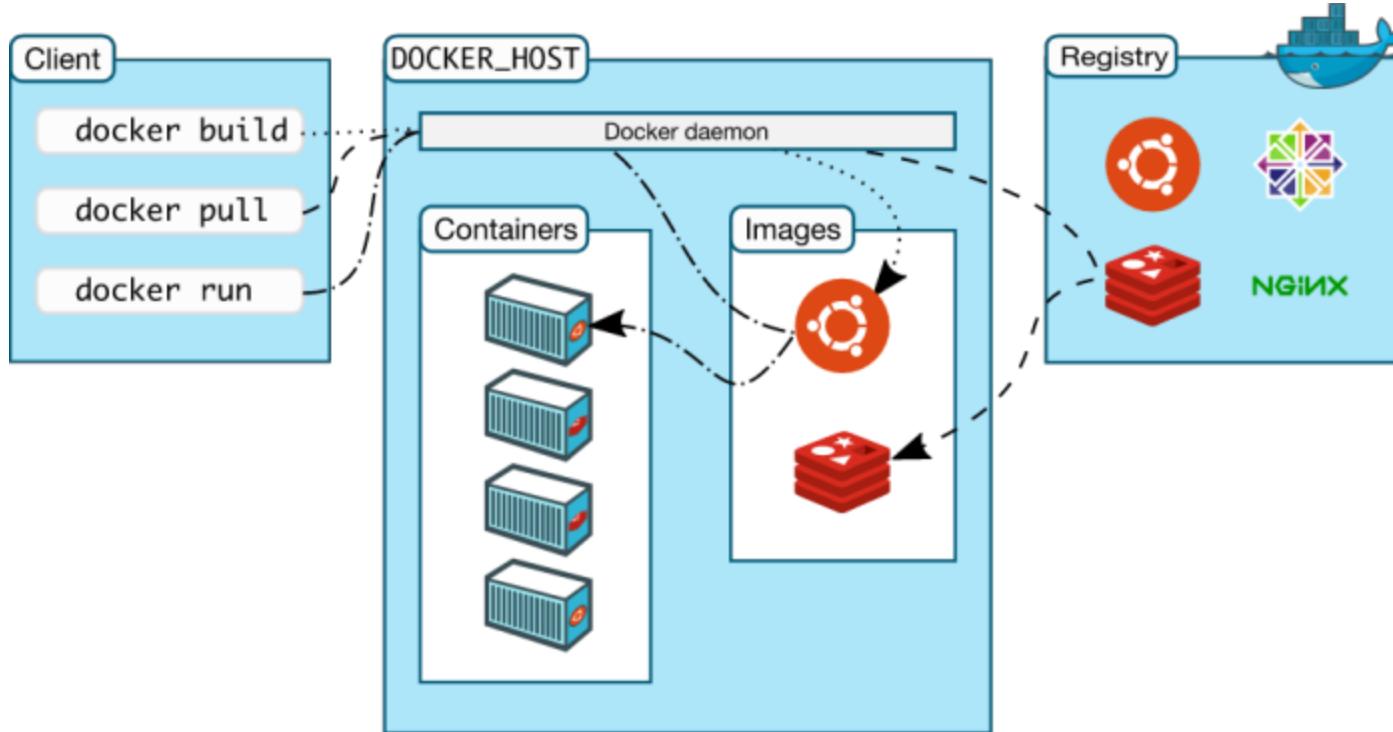


*Small and fast, portable
Uses OS-level virtualization*

Container Overview



Container Overview



Docker commands - pull

Pull an image from an image registry

```
Admin:~/environment $ █
```

Docker commands - run

Running a container from an image

```
Admin:~/environment $ █
```

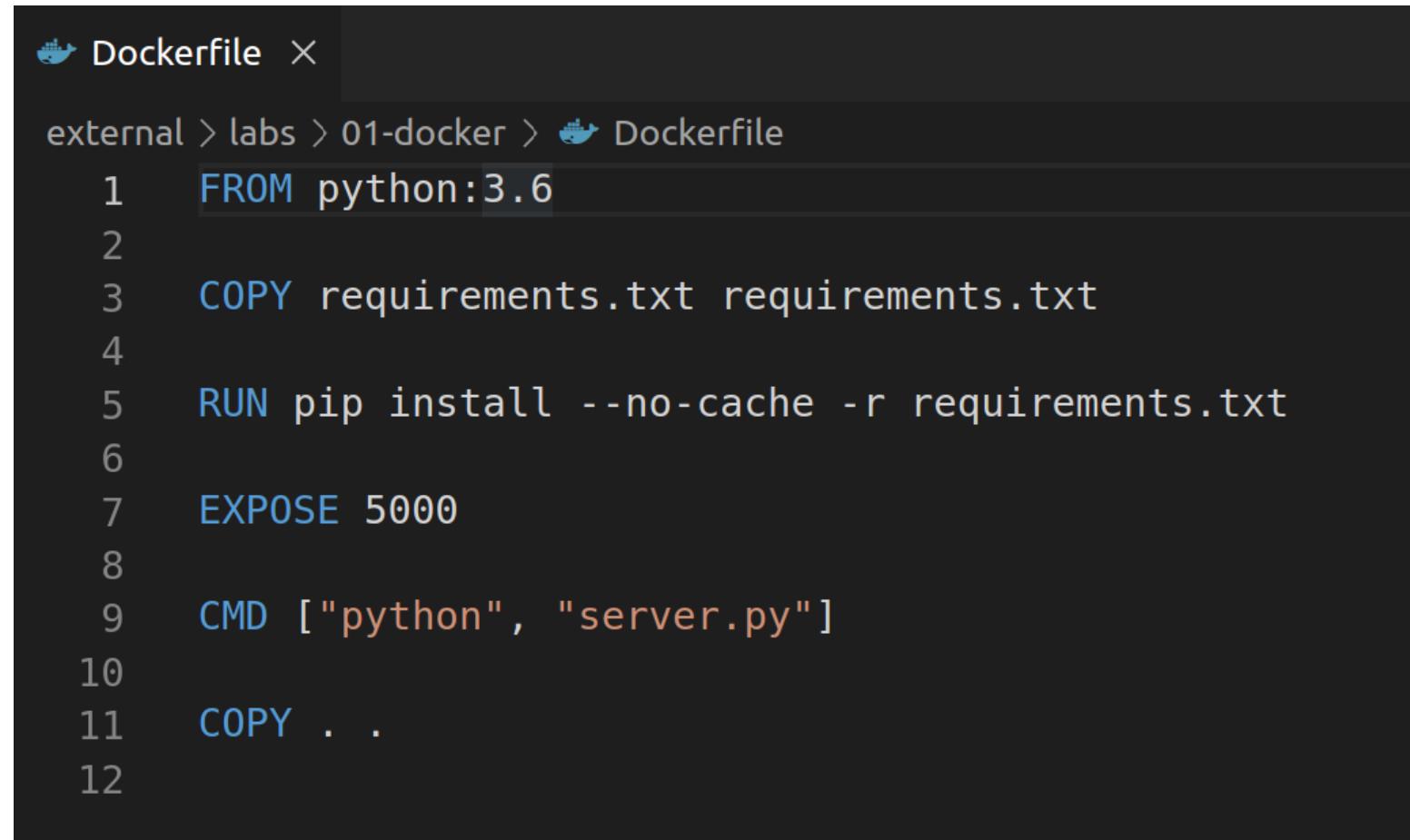
The `-p` flag maps a port from the host to the container

The `-d` flag runs the container in the background - detached

Dockerfile

Dockerfile contains a list of instructions to build a container image

```
labs/01-docker/Dockerfile
```



The screenshot shows a code editor window with a dark theme. The title bar says "Dockerfile X". The file path is "external > labs > 01-docker > Dockerfile". The code itself is a Dockerfile with the following content:

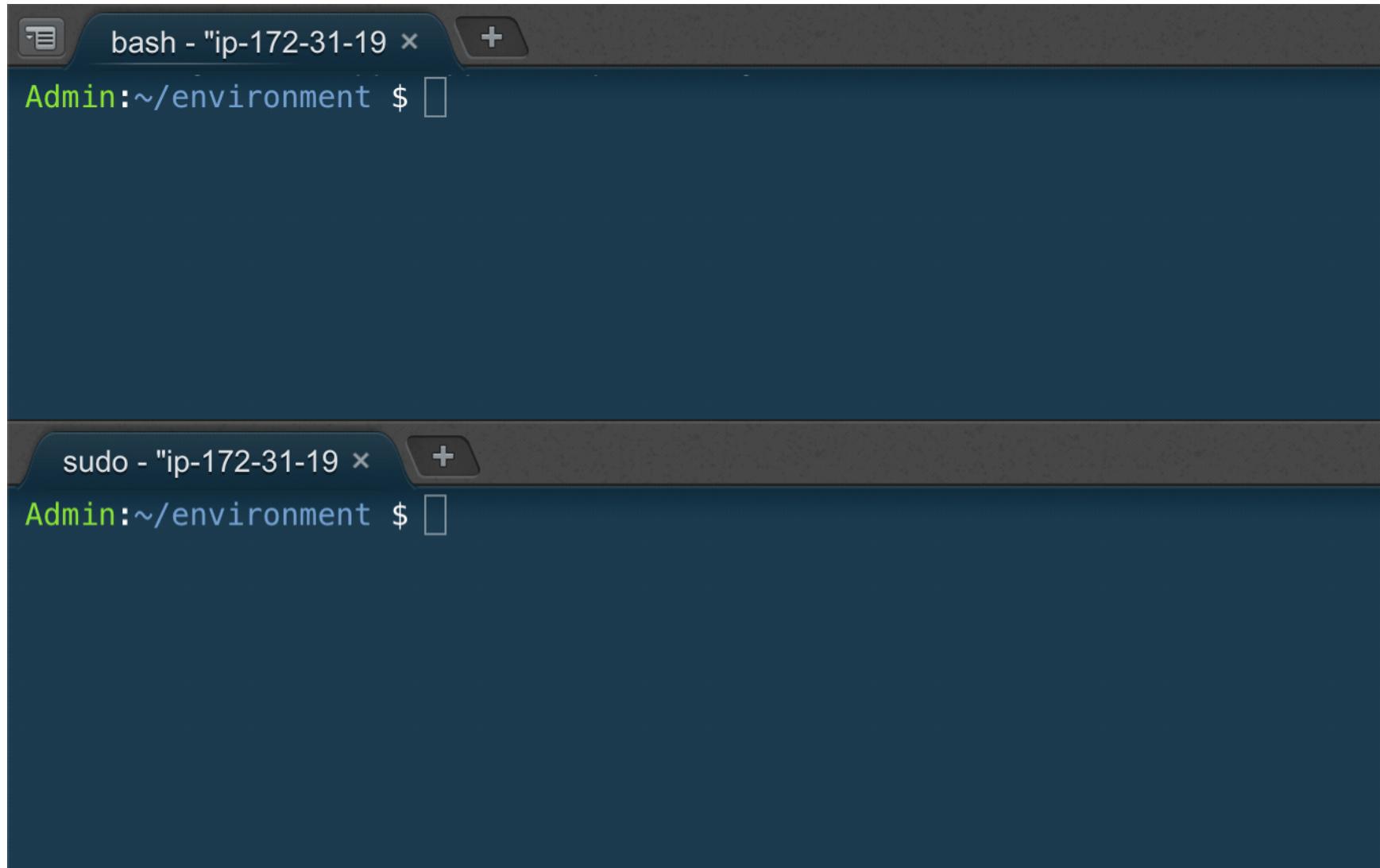
```
1 FROM python:3.6
2
3 COPY requirements.txt requirements.txt
4
5 RUN pip install --no-cache -r requirements.txt
6
7 EXPOSE 5000
8
9 CMD ["python", "server.py"]
10
11 COPY . .
12
```

Docker build

```
Admin:~/environment/python-app $ █
```

Docker run

Finally we can run a container from the image



The image shows a terminal window with two tabs. The top tab is titled 'bash - "ip-172-31-19 x"' and the bottom tab is titled 'sudo - "ip-172-31-19 x"'. Both tabs show the prompt 'Admin:~/environment \$' followed by a cursor. The background of the terminal is dark blue.

Before we move on, any questions?

What is



Kubernetes?

What is Kubernetes?

What is Kubernetes?

Kubernetes is a portable, extensible, open-source container orchestrator for managing highly available and fault tolerant containerized workloads and services.

Kubernetes name originates from Greek which means "governor", "helmsman" or "captain.

Originally designed by Google, Its development and design are heavily influenced by Google's Borg system. Google open-sourced the Kubernetes project in 2014 and is now maintained by the Cloud Native Computing Foundation.

Kubernetes Clusters abstract their underlying computing resources, allowing users to deploy workloads to the entire cluster as opposed to a particular server.

Why Kubernetes?

Containers are a good way to bundle and run your applications. Containers paved the way to build cloud native systems, in which services are implemented using small clouds of containers.

However, in a production environment, you need to manage the thousands of containers that run the applications and ensure that there is no downtime.

Key capabilities were missing !

Resource Utilisation

Using multiple containers with shared resources

Monitoring running containers

Handling dead containers

Autoscaling container instances to handle load

Making the container services easily accessible
Connecting containers to a variety of external data sources

Kubernetes architecture enables

A single administrator to manage thousands of containers running simultaneously without the necessity to actually login to the servers where the application is run.

We don't need to care where applications runs.

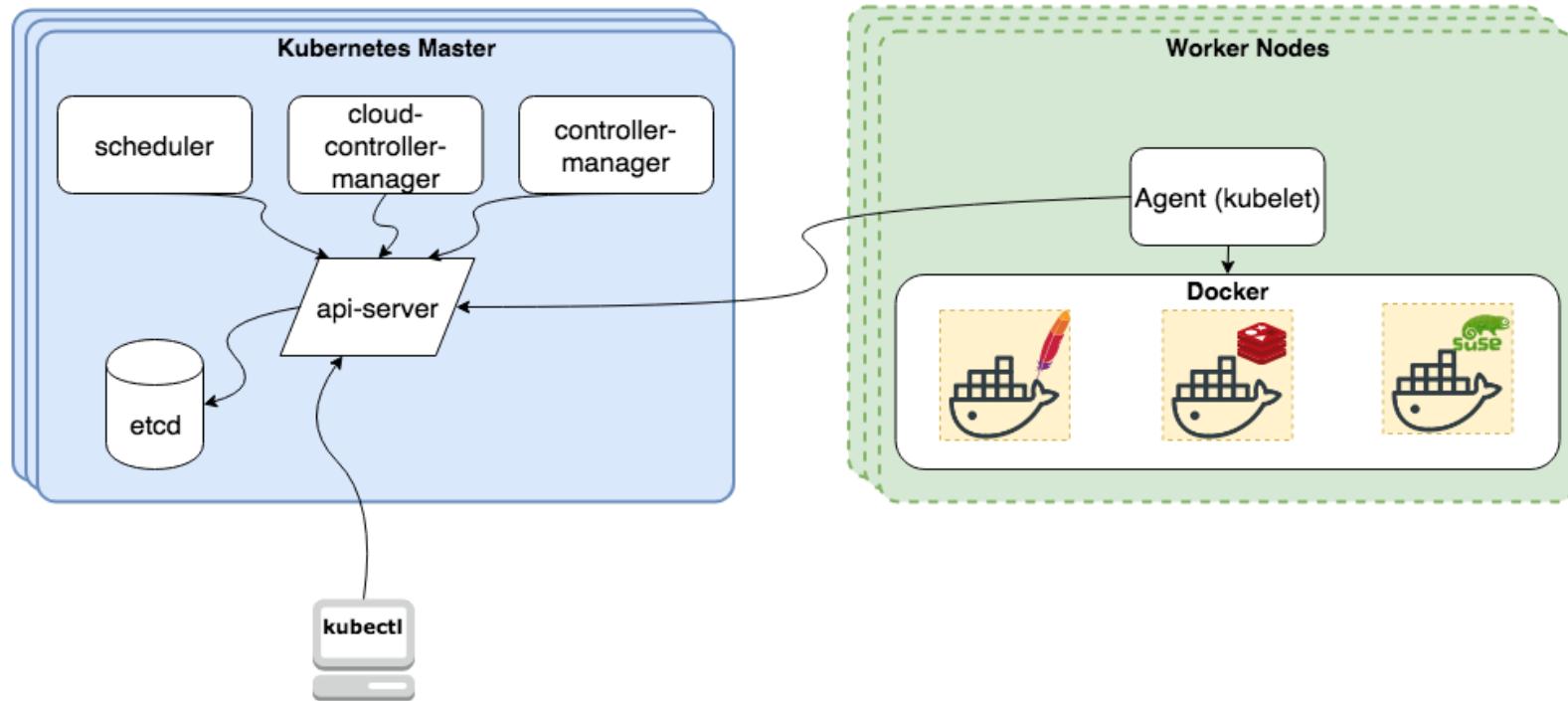
K8s Automates deployment using declarative statement.

Enforces Desired state.

Resource utilisation.

Kubernetes Architecture

Kubernetes Architecture



What is a Pod

A Pod is the basic execution unit of a Kubernetes application—the smallest and simplest unit in the Kubernetes object model that you create or deploy.

A Pod represents processes running on your Cluster.

Pods enable multiple containers to run on a host machine and share resources such as storage, networking, and container runtime information.

Pod spec

`myapp.yaml`

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', 'echo Hello Kubernetes! && sleep 3600']
```

```
$ kubectl apply -f myapp.yaml
```



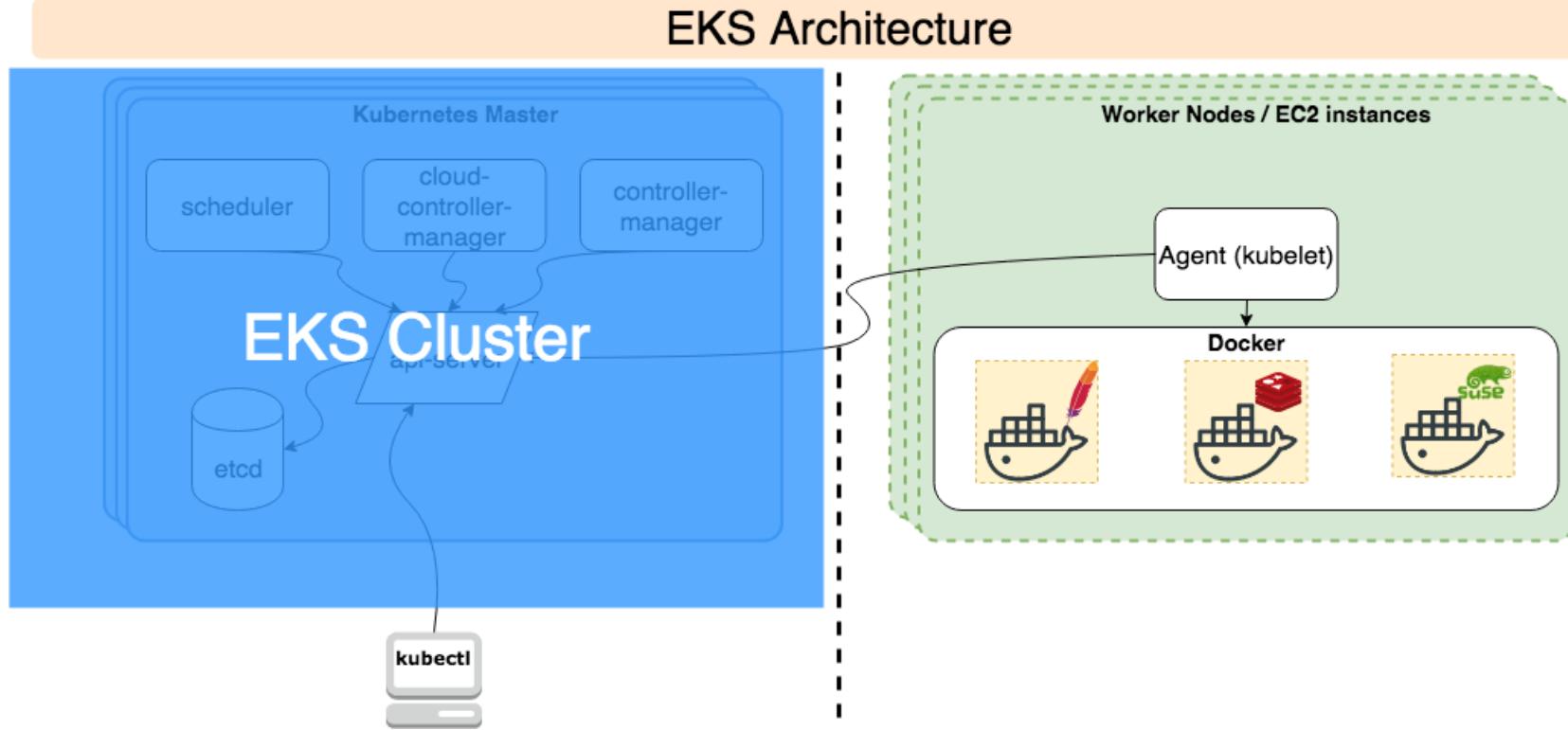
EKS

What is EKS?

What is EKS?

- EKS is the **managed** version of Kubernetes offered by AWS
- EKS launches and maintains the Control Plane for you with **high available** components
- EKS offers integration with other AWS services such as VPC and IAM. These integrations are **open source** projects built with the community
- EKS takes care of upgrades and patching
- EKS is based on vanilla Kubernetes

What is EKS?



Connecting to your K8s/EKS Cluster

What we'll need

- A client, which is a single binary called `kubectl`

What we'll need

- A client, which is a single binary called `kubectl`
- A configuration file for kubectl to be stored under: `~/.kube/config`

What we'll need

- A client, which is a single binary called `kubectl`
- A configuration file for kubectl to be stored under: `~/.kube/config`

The kubectl configuration can be generated automatically.

Launching a Kubernetes Cluster / EKS

Creating a Cluster

For this we'll use `eksctl` to create an EKS cluster and Worker Nodes:

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

This command is also available on the GitHub page

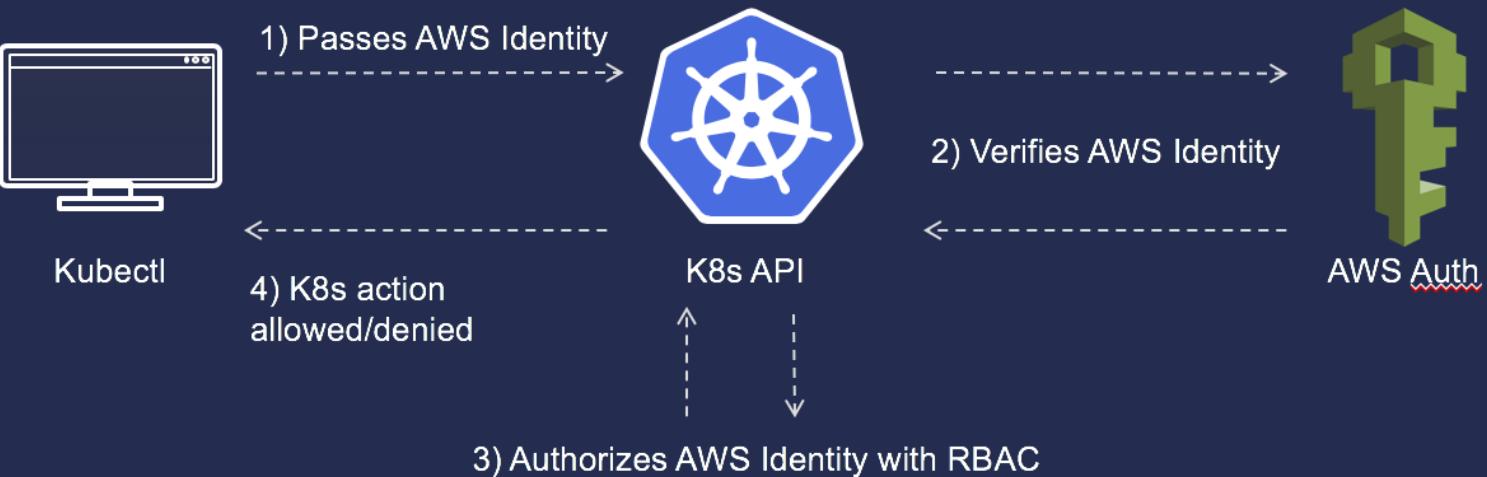
The EKS Cluster will take approximately 15 minutes to create

While that runs...

kubectl configuration - eksctl

```
$ kubectl config view
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: DATA+OMITTED
  server: https://CA124C4EA7486B52460C551DE1787C05.yl4.eu-north-1.eks.amazonaws.com
  name: eks.eu-north-1.eksctl.io
contexts:
- context:
  cluster: eks.eu-north-1.eksctl.io
  user: user@eks.eu-north-1.eksctl.io
  name: user@eks.eu-north-1.eksctl.io
current-context: jason@eks.eu-north-1.eksctl.io
kind: Config
preferences: {}
users:
- name: user@eks.eu-north-1.eksctl.io
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
        - --region
        - eu-north-1
        - eks
        - get-token
        - --cluster-name
        - eks
      command: aws
      env: null
```

kubectl - Authentication



Let's check our new Cluster

Let's check our new Cluster

1 - Head to the EKS service in the AWS Web Console

Let's check our new Cluster

1 - Head to the EKS service in the AWS Web Console

2 - Select the 'eks' cluster and check what options are available

Checking your EKS cluster

The screenshot shows the AWS EKS Cluster configuration page for a cluster named 'eks'. The page is divided into sections: General configuration, Networking, and a bottom navigation bar.

General configuration:

- Kubernetes Version: 1.14
- Platform Version: eks.1
- Status: ACTIVE
- API server endpoint: https://CA124C4EA7486B52460C551DE1787C05.yl4.eu-north-1.eks.amazonaws.com (highlighted with a red box)
- OpenID Connect provider URL: https://oidc.eks.eu-north-1.amazonaws.com/id/CA124C4EA7486B52460C551DE1787C05
- Cluster ARN: arn:aws:eks:eu-north-1:335688126910:cluster/oscar
- Certificate authority: A long string of characters (highlighted with a yellow box): LS0tLS1CRUdjTIBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHvnPNojRYRFRNU1Ea3lNekv3TwPjMU1Wb1hEVek1TURreU1ERXdNakkxTVZvd0ZURVRNQKVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQOFtSYdEIIIVIKS29aSWh2Vn5R1IIIIVC01FBRGd0pVBBRENDD0V/EvO2dnR1IIRSw0CmbhN1R2sTIII3MA01FERD
- Role ARN: arn:aws:iam::335688126910:role/eksctl-eks-cluster-ServiceRole-1X058FD0AGYIO

Networking:

VPC	Subnets	Security groups	API server endpoint access
vpc-04287be37e224d2bc	subnet-09a905b81dc255e44 subnet-0ffae02dd45bc1ed6 subnet-0a8504adb33a2a814 subnet-09d0feb73ac9f0f98 subnet-0d239c4720e4cd59f subnet-019a941b7efc42ce4	sg-0258eba705597213e	Private access Disabled

Bottom navigation:

- eks
- Cluster configuration
- Nodes
- Logs
- Events
- Metrics
- Logs Insights
- Logs Metrics
- Logs Metrics Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights Insights Insights Insights
- Logs Metrics Insights Insights Insights Insights
- Logs Metrics Insights Insights
- Logs Metrics Insights
- Logs Metrics
- Logs

Let's check our new Cluster

Let's check our new Cluster

...using kubectl

Do we have Nodes?

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

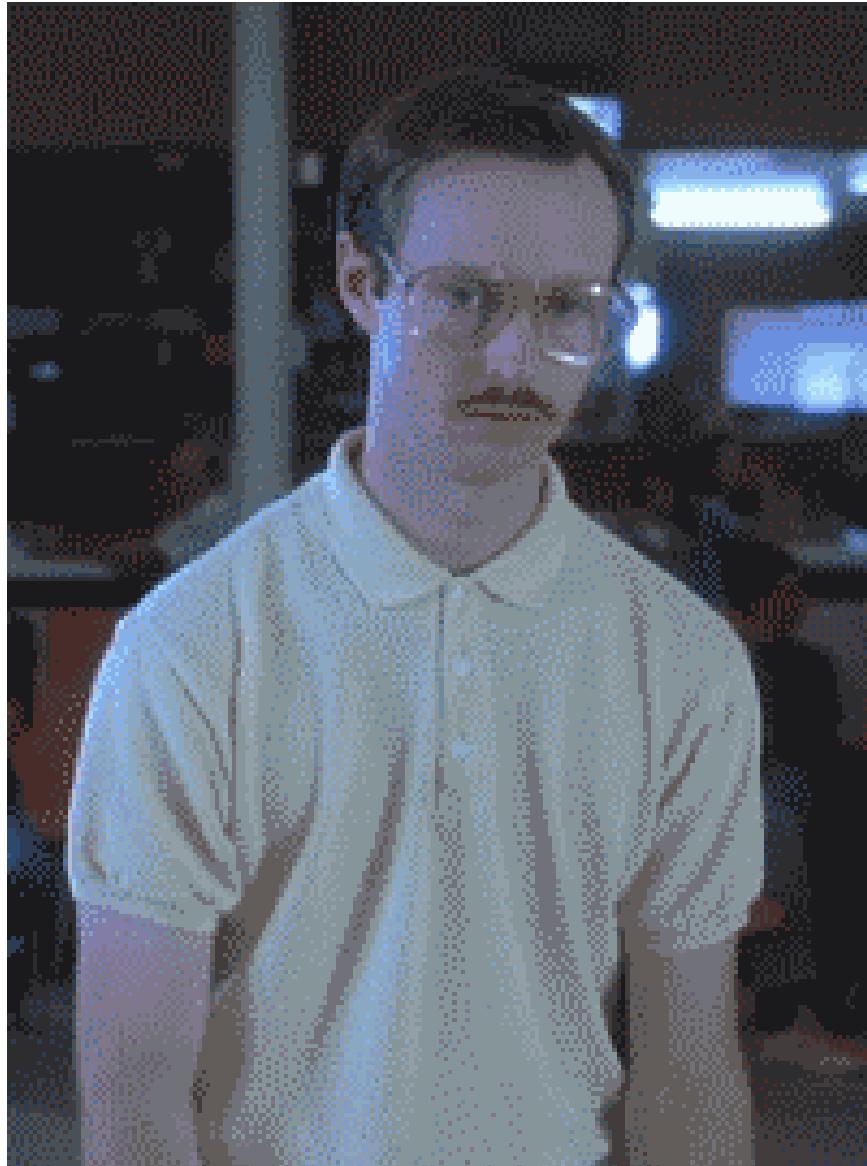
Do we have Nodes?

```
Admin:~/environment $ █
```

No, I don't have a cluster! Run:

```
eksctl create cluster \  
  --region eu-west-1 \  
  --version 1.14 --node-type t3.medium \  
  --name eks
```

Successfully Create and Connect to EKS Cluster



End of Day 1

Questions?

Thank you!