Getting Started with Kubernetes on AWS

Brought to you by the AWS Cloud Support Team

Day 3

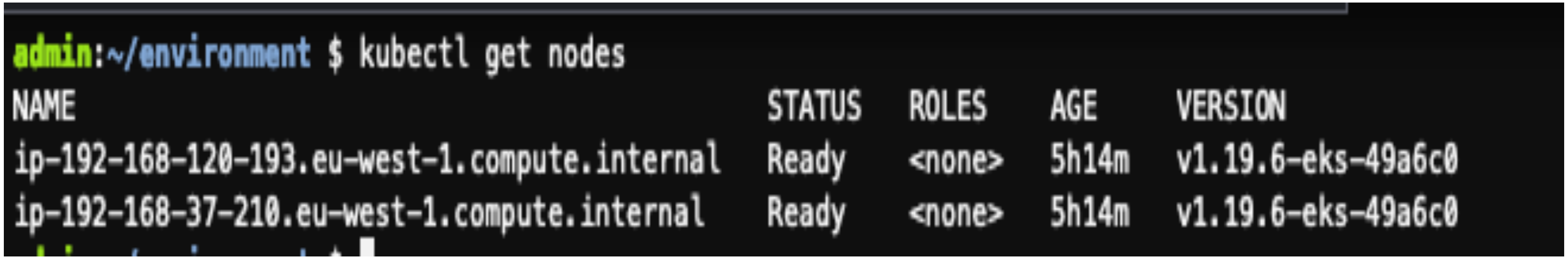
Agenda

- What have we learnt?
- Project
- Write assessment

Firstly...

Do you have a cluster?

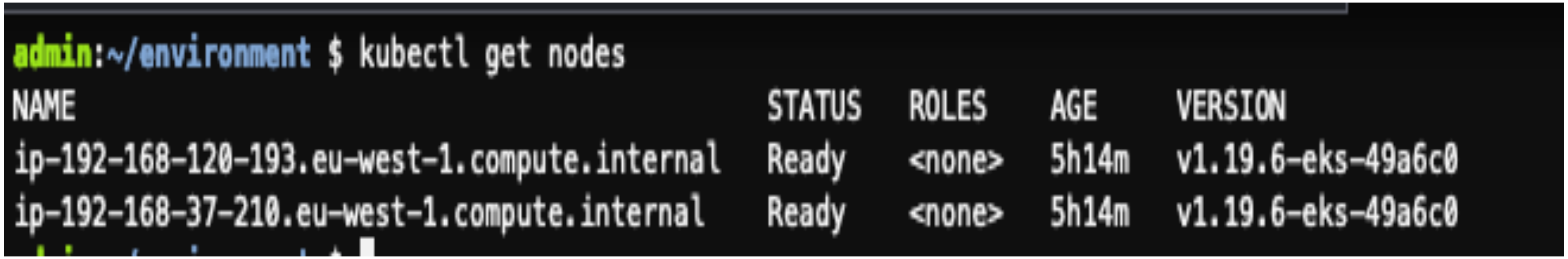
Using a terminal in Cloud9, verify there are Worker Nodes in your cluster



```
admin:~/environment $ kubectl get nodes
NAME                                          STATUS   ROLES    AGE     VERSION
ip-192-168-120-193.eu-west-1.compute.internal Ready    <none>   5h14m   v1.19.6-eks-49a6c0
ip-192-168-37-210.eu-west-1.compute.internal  Ready    <none>   5h14m   v1.19.6-eks-49a6c0
```

Do you have a cluster?

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



```
admin:~/environment $ kubectl get nodes
NAME                                         STATUS   ROLES    AGE     VERSION
ip-192-168-120-193.eu-west-1.compute.internal   Ready    <none>   5h14m   v1.19.6-eks-49a6c0
ip-192-168-37-210.eu-west-1.compute.internal    Ready    <none>   5h14m   v1.19.6-eks-49a6c0
```

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

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

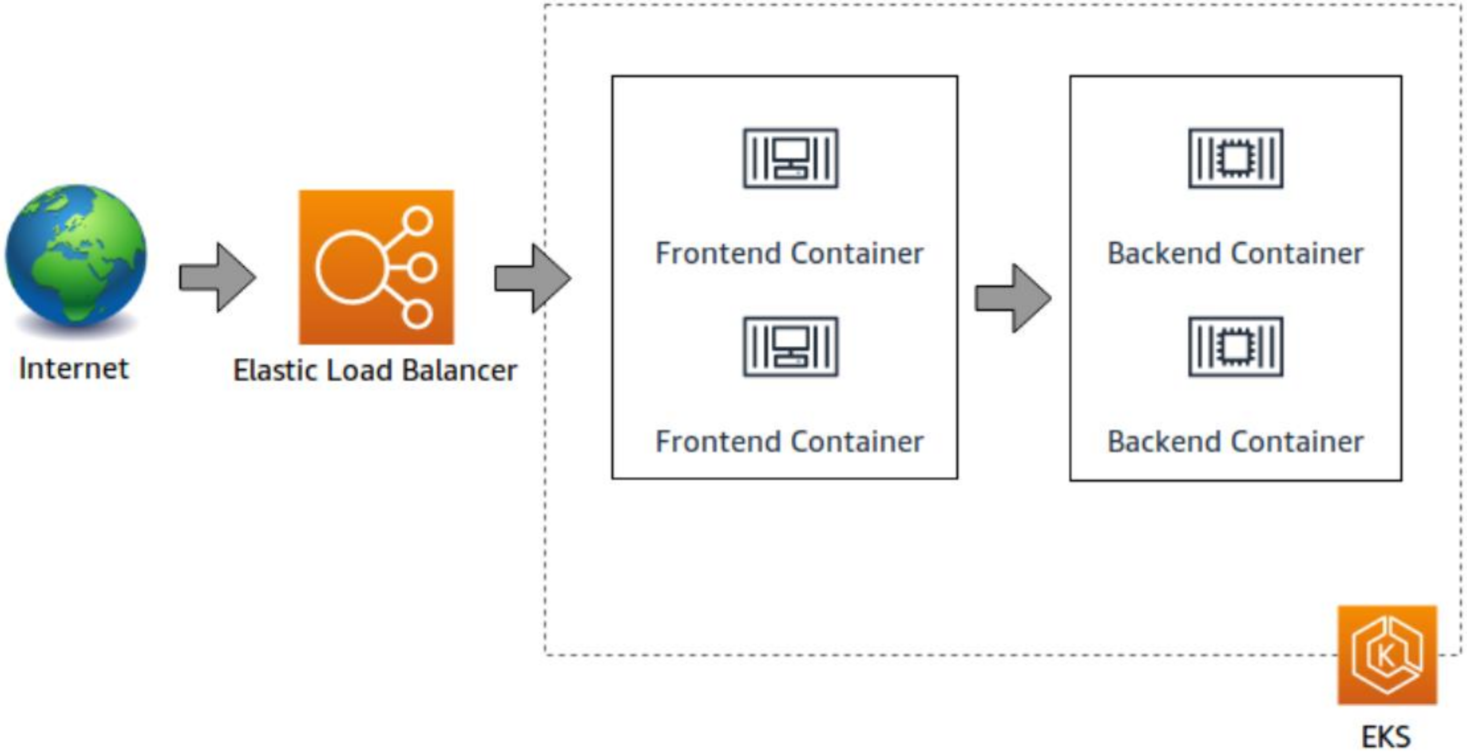
```
eksctl create cluster --version 1.19 --node-type t3.medium --name eks
```

Lets pull in the latest changes

```
$ cd ~/environment/eks
$ git pull origin master
```

Final Project

Two Tier Web Application on Kubernetes



Frontend - Ruby App

1. Download the source code and deploy to your Kubernetes cluster.
    1. You'll need a Docker Hub account, create one if you don't have one yet.
    2. If you need to run Docker commands we recommend doing it from the Cloud9 environment.
    3. Source code is available in the GitHub Repo under `project/frontend`.
    4. Use `awselscpt/frontend-base` (already in Docker Hub) as the base image.
    5. **Important** - the application is configured to listen on port `tcp/4567`
2. Make sure that the frontend is accessible from the internet
3. Test the connection to the frontend
4. Test the connection to the backend from the deployed app

Backend - API

1. Deploy the following image to your cluster `awselscpt/backend` (in Docker Hub already)
2. Configure your frontend to connect to your backend
3. Re-test the connnection to the backend from the frontend app - make corrections as necessary.

Bonus points (in any order)

Once the project is completed, for bonus points work on the below!

- Restrict the access to the frontend to a given IP address or range
- Put your image in Amazon ECR repository and update your K8s objects
- Configure the frontend to automatically scale based on CPU utilization
- Migrate to using an Application Load Balancer for the frontend service
- Configure health checks for the frontend and backend Pods

Good Luck!

Or visit the link on GitHub:

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

Project requirements is in the `project/README.md` file.

Cleaning up

Steps are available at the GitHub repo: https://github.com/aws-els-cpt/eks

Delete the EKS Cluster

```
$ eksctl delete cluster eks
```

Delete the CloudFormation stack

- This can be done in the CloudFormation console, navigate to CloudFormation
- There may be a number of stacks, select the stack named "cloud9", and click the "Delete" button

Thank you!