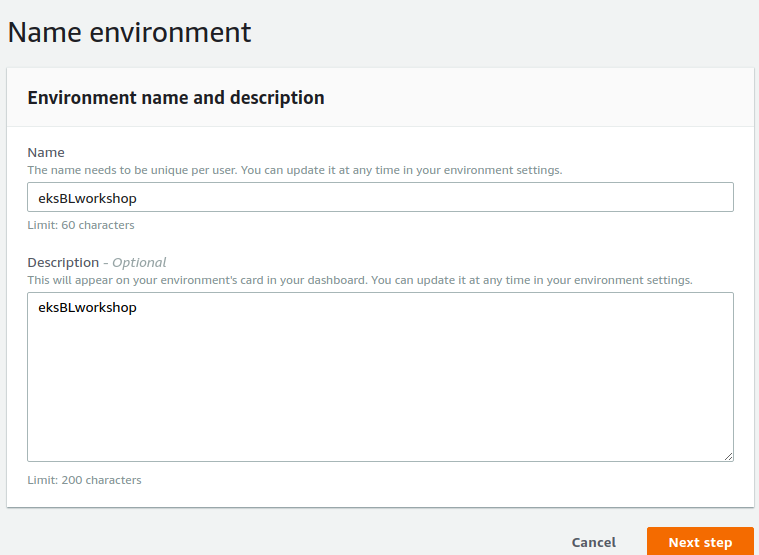
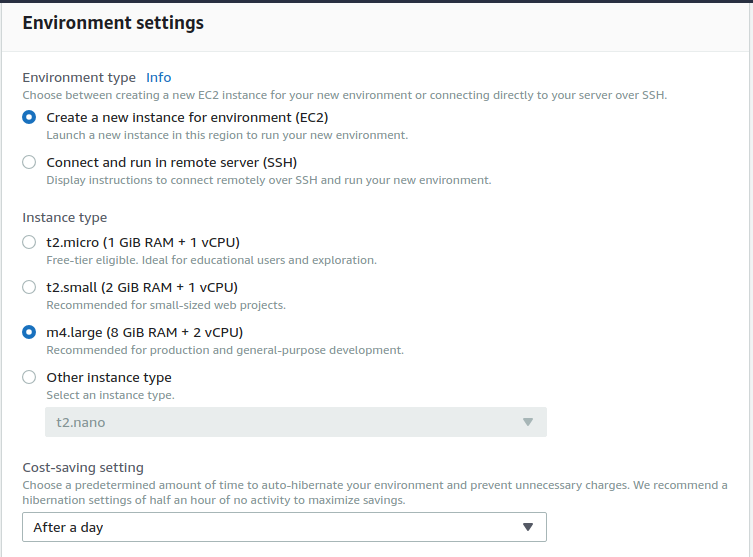
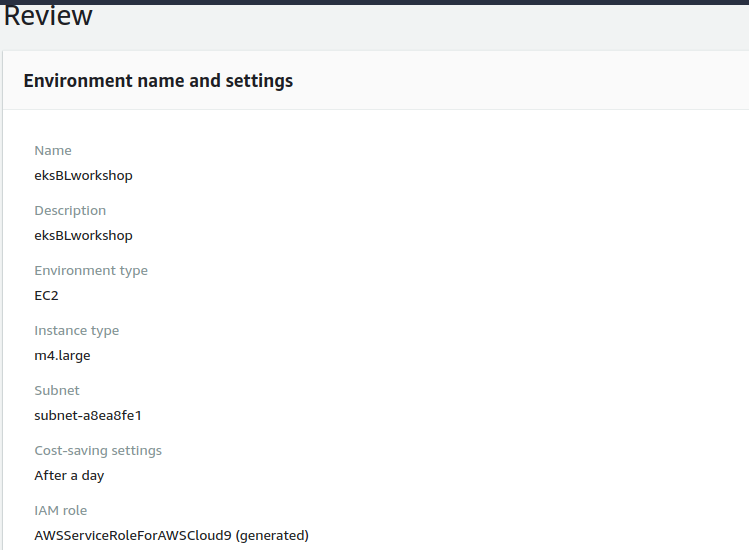
**Introduction Amazon EKS by Cloud9**

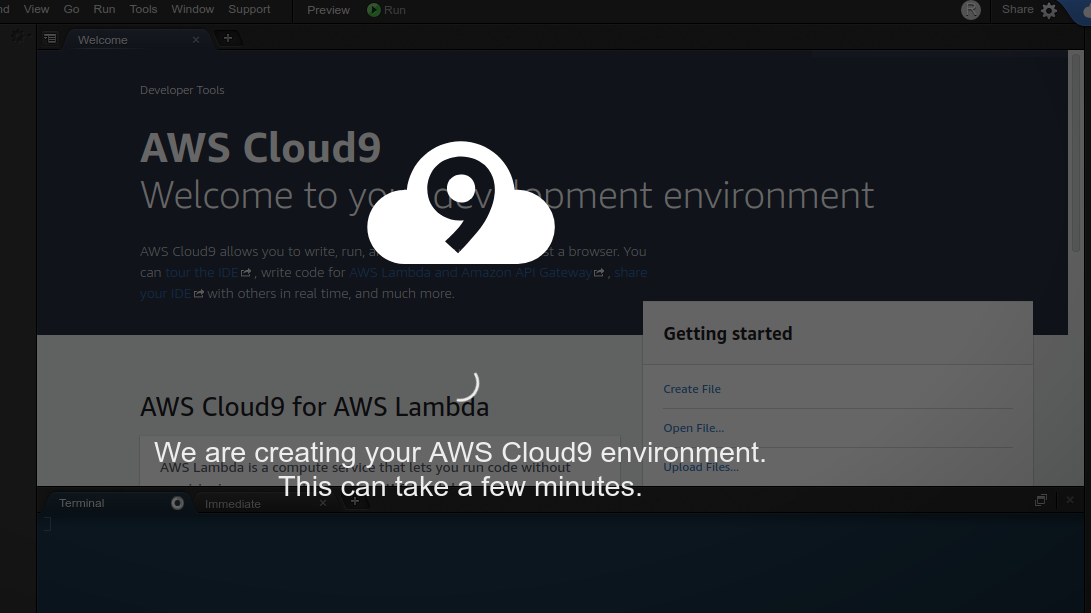
The Cloud9 workspace should be built by an IAM user with Administrator privileges, not the root account user. Please ensure you are logged in as an IAM user, not the root account user.

* Create a [Cloud9 Environment](https://us-west-2.console.aws.amazon.com/cloud9/home?region=us-west-2)
  + Select **Create environment**
* Name it **eksBLworkshop**, and take all other defaults

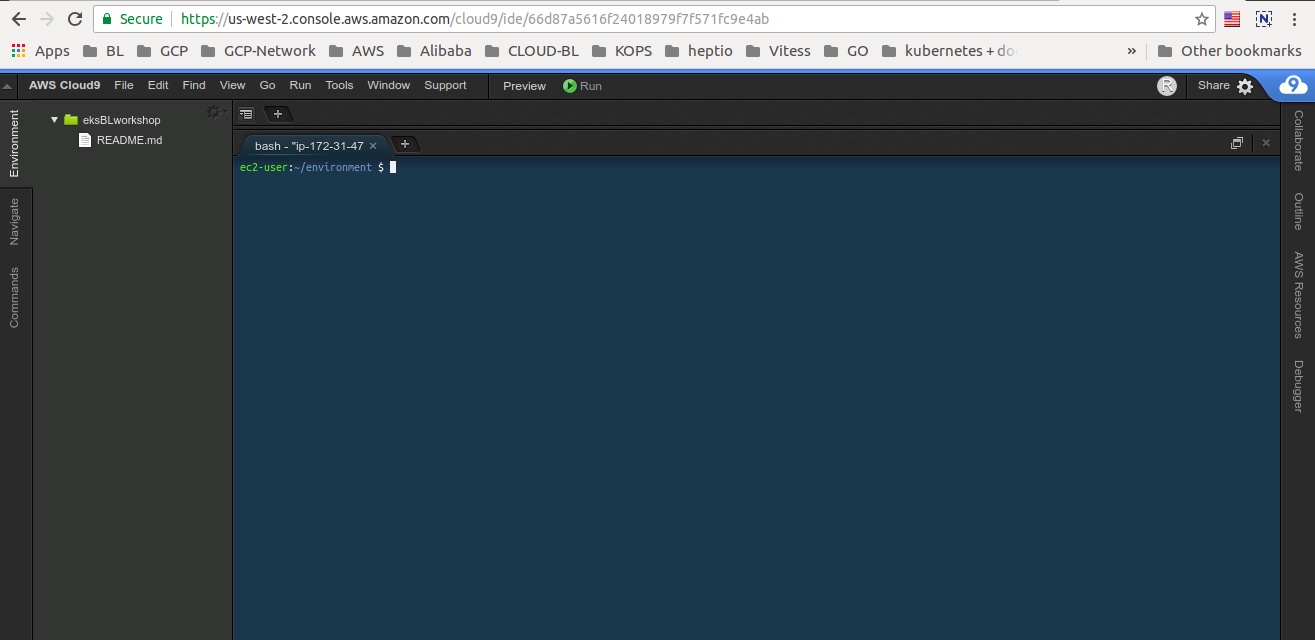








* When it comes up, customize the environment by closing the **welcome tab** and **lower work area**, and opening a new **terminal** tab in the main work area:



#### **Install AWS IAM Authenticator**

ec2-user:~/environment $ go get -u -v github.com/kubernetes-sigs/aws-iam-authenticator/cmd/aws-iam-authenticator

ec2-user:~/environment $ sudo mv ~/go/bin/aws-iam-authenticator /usr/local/bin/aws-iam-authenticator

#### **Verify the binaries**

ec2-user:~/environment $ kubectl version --short --client

Client Version: v1.10.3

ec2-user:~/environment $ aws-iam-authenticator help

A tool to authenticate to Kubernetes using AWS IAM credentials

Usage:

aws-iam-authenticator [command]

Available Commands:

help Help about any command

init Pre-generate certificate, private key, and kubeconfig files for the server.

server Run a webhook validation server suitable that validates tokens using AWS IAM

token Authenticate using AWS IAM and get token for Kubernetes

verify Verify a token for debugging purpose

version Version will output the current build information

Flags:

-i, --cluster-id ID Specify the cluster ID, a unique-per-cluster identifier for your aws-iam-authenticator installation.

-c, --config filename Load configuration from filename

-h, --help help for aws-iam-authenticator

-l, --log-format string Specify log format to use when logging to stderr [text or json] (default "text")

Use "aws-iam-authenticator [command] --help" for more information about a command.

#### **Install JQ (Dependancies)**

ec2-user:~/environment $ sudo yum -y install jq

# **CLONE THE SERVICE REPOS**

ec2-user:~/environment $ cd ~/environment

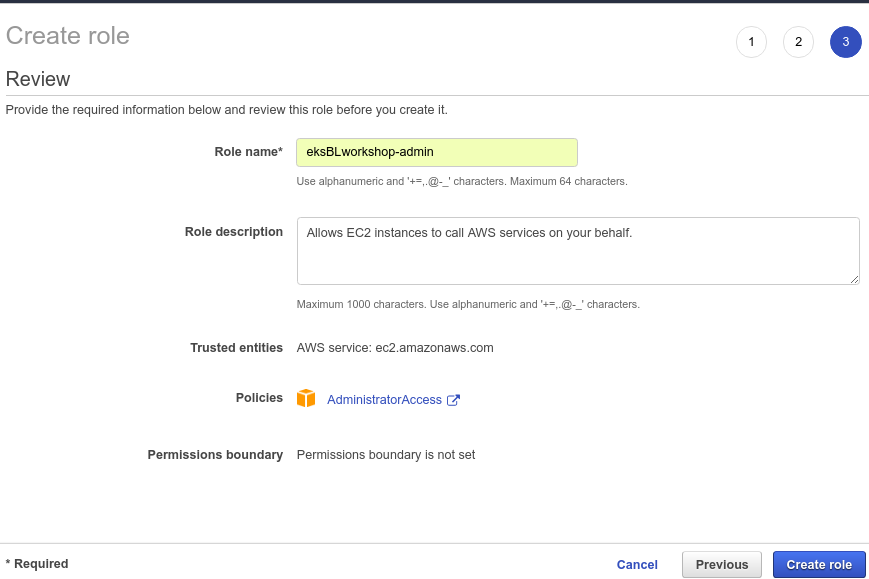
ec2-user:~/environment $ git clone <https://github.com/ahmaddanni/ecsdemo-frontend.git>

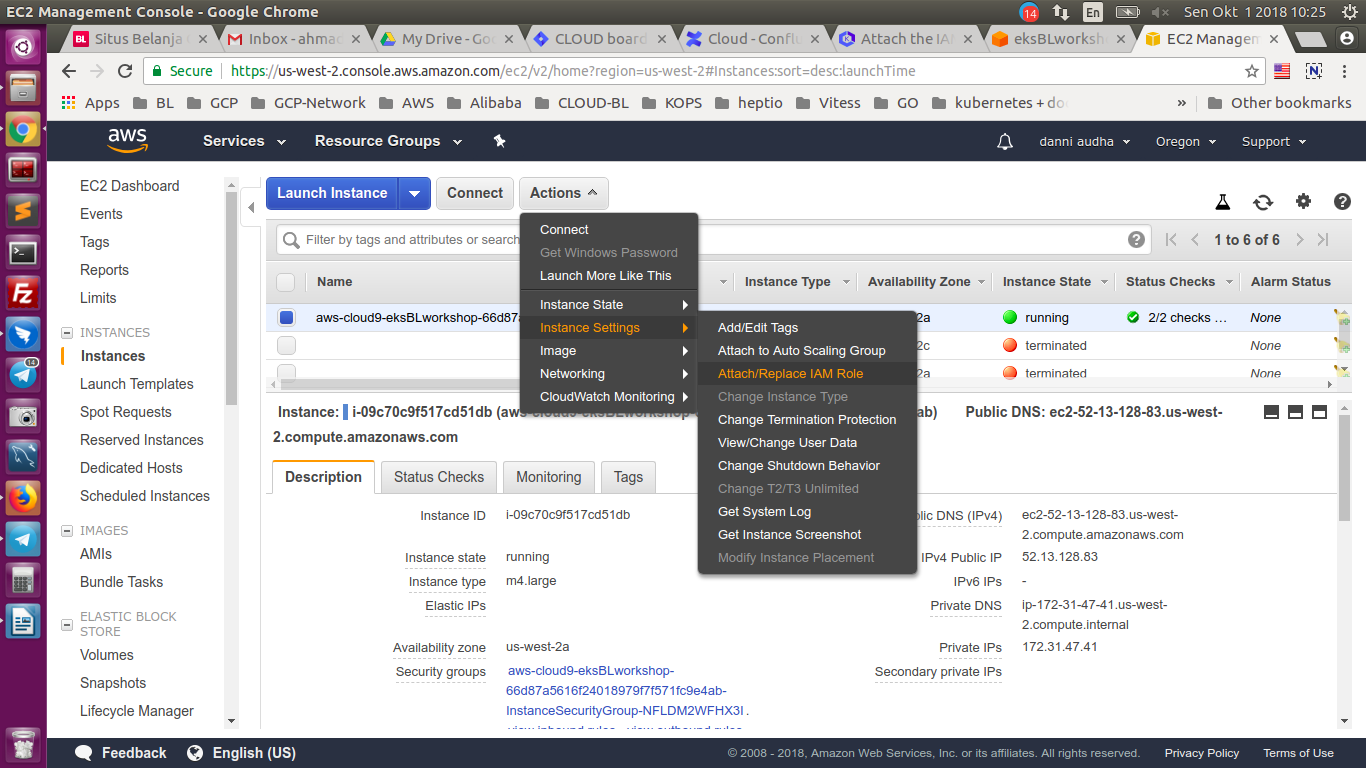
ec2-user:~/environment $ git clone https://github.com/ahmaddanni/ecsdemo-nodejs.git

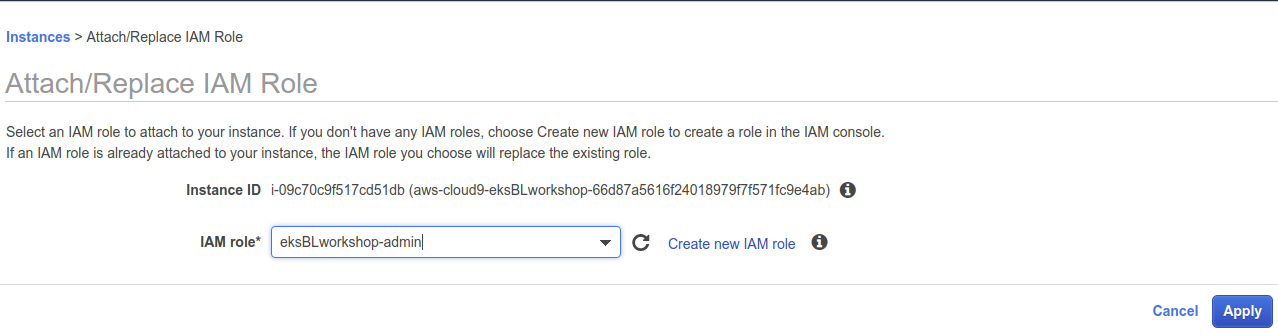
ec2-user:~/environment $ git clone <https://github.com/ahmaddanni/ecsdemo-crystal.git>

# **CREATE AN IAM ROLE FOR YOUR WORKSPACE**

* Follow [this deep link to create an IAM role with Administrator access.](https://console.aws.amazon.com/iam/home" \l "/roles$new?step=review&commonUseCase=EC2%2BEC2&selectedUseCase=EC2&policies=arn:aws:iam::aws:policy%2FAdministratorAccess)
* Confirm that **AWS service** and **EC2** are selected, then click **Next** to view permissions.
* Confirm that **AdministratorAccess** is checked, then click **Next** to review.
* Enter **eksBLworkshop-admin** for the Name, and select **Create Role**



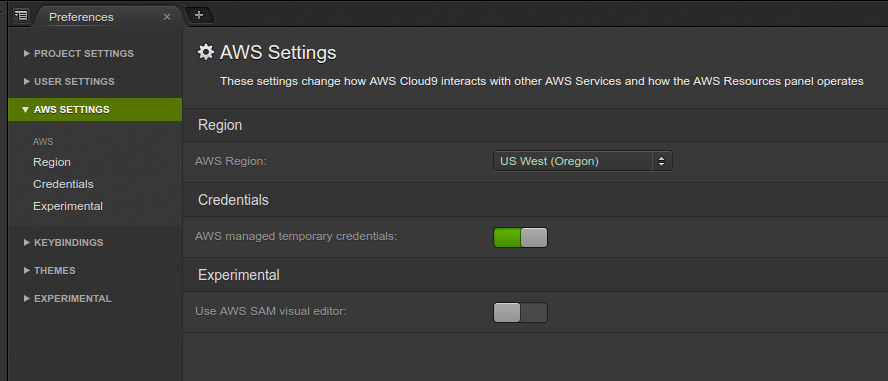




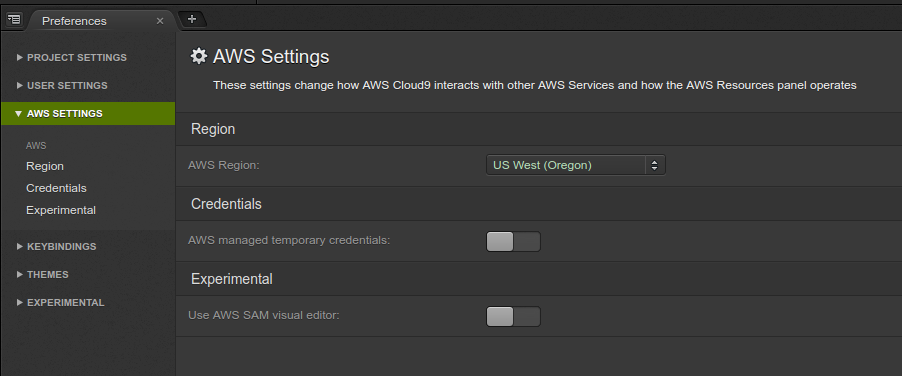
Choose **eksBLworkshop-admin** from the **IAM Role** drop down, and select **Apply**

# **UPDATE IAM SETTINGS FOR YOUR WORKSPACE**

* Return to your workspace and click the sprocket, or launch a new tab to open the Preferences tab
* Select **AWS SETTINGS**



* Turn of **AWS managed temporary credentials**



* Close the Preferences tab
* To ensure temporary credentials aren’t already in place we will also remove any existing credentials file:

ec2-user:~/environment $ rm -vf ${HOME}/.aws/credentials

* We should configure our aws cli with our current region as default:

ec2-user:~/environment $ export AWS\_REGION=$(curl -s 169.254.169.254/latest/dynamic/instance-identity/document | jq -r .region)

ec2-user:~/environment $ echo "export AWS\_REGION=${AWS\_REGION}" >> ~/.bash\_profile

ec2-user:~/environment $ aws configure set default.region ${AWS\_REGION}

ec2-user:~/environment $ aws configure get default.region

us-west-2

# **LAUNCH USING** [**EKSCTL BY WEAVEWORKS**](https://eksctl.io/)

I will highlight a tool contributed by [Weaveworks](https://weave.works/) called [eksctl](https://eksctl.io/), based on the official AWS CloudFormation templates, and will use it to launch and configure our EKS cluster and nodes.

For this module, we need to download the [eksctl](https://eksctl.io/) binary:

ec2-user:~/environment $ curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/latest\_release/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

ec2-user:~/environment $ sudo mv -v /tmp/eksctl /usr/local/bin

‘/tmp/eksctl’ -> ‘/usr/local/bin/eksctl’

**Confirm the eksctl command works:**

ec2-user:~/environment $ eksctl version

2018-10-01T04:05:47Z [ℹ] versionInfo = map[string]string{"gitCommit":"13286de1b06717200a5fbe692878be4937a1b05f", "gitTag":"0.1.4", "builtAt":"2018-09-28T08:29:56Z"}

**To create a basic EKS cluster, run:**

(Launching EKS and all the dependencies will take approximately 15 minutes)

ec2-user:~/environment $ eksctl create cluster --name=eksworkshop-eksctl --nodes=3 --node-ami=auto --region=${AWS\_REGION}

2018-10-01T04:12:14Z [ℹ] setting availability zones to [us-west-2b us-west-2a us-west-2c]

2018-10-01T04:12:15Z [ℹ] using "ami-0a54c984b9f908c81" for nodes

2018-10-01T04:12:15Z [ℹ] creating EKS cluster "eksworkshop-eksctl" in "us-west-2" region

2018-10-01T04:12:15Z [ℹ] will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup

2018-10-01T04:12:15Z [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --name=eksworkshop-eksctl'

2018-10-01T04:12:15Z [ℹ] creating cluster stack "eksctl-eksworkshop-eksctl-cluster"

2018-10-01T04:23:11Z [ℹ] creating nodegroup stack "eksctl-eksworkshop-eksctl-nodegroup-0"

2018-10-01T04:27:02Z [✔] all EKS cluster resource for "eksworkshop-eksctl" had been created

2018-10-01T04:27:02Z [✔] saved kubeconfig as "/home/ec2-user/.kube/config"

2018-10-01T04:27:02Z [ℹ] the cluster has 0 nodes

2018-10-01T04:27:02Z [ℹ] waiting for at least 3 nodes to become ready

2018-10-01T04:27:32Z [ℹ] the cluster has 3 nodes

2018-10-01T04:27:32Z [ℹ] node "ip-192-168-191-161.us-west-2.compute.internal" is ready

2018-10-01T04:27:32Z [ℹ] node "ip-192-168-206-203.us-west-2.compute.internal" is ready

2018-10-01T04:27:32Z [ℹ] node "ip-192-168-70-184.us-west-2.compute.internal" is ready

2018-10-01T04:27:33Z [ℹ] kubectl command should work with "/home/ec2-user/.kube/config", try 'kubectl get nodes'

2018-10-01T04:27:33Z [✔] EKS cluster "eksworkshop-eksctl" in "us-west-2" region is ready

**Confirm your Nodes:**

ec2-user:~/environment $ kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-192-168-191-161.us-west-2.compute.internal Ready <none> 4m v1.10.3

ip-192-168-206-203.us-west-2.compute.internal Ready <none> 4m v1.10.3

ip-192-168-70-184.us-west-2.compute.internal Ready <none> 4m v1.10.3

#### **Congratulations!**

You now have a fully working Amazon EKS Cluster that is ready to use!

# DEPLOY THE OFFICIAL KUBERNETES DASHBOARD

We can deploy the dashboard with the following command:

ec2-user:~/environment $ kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml

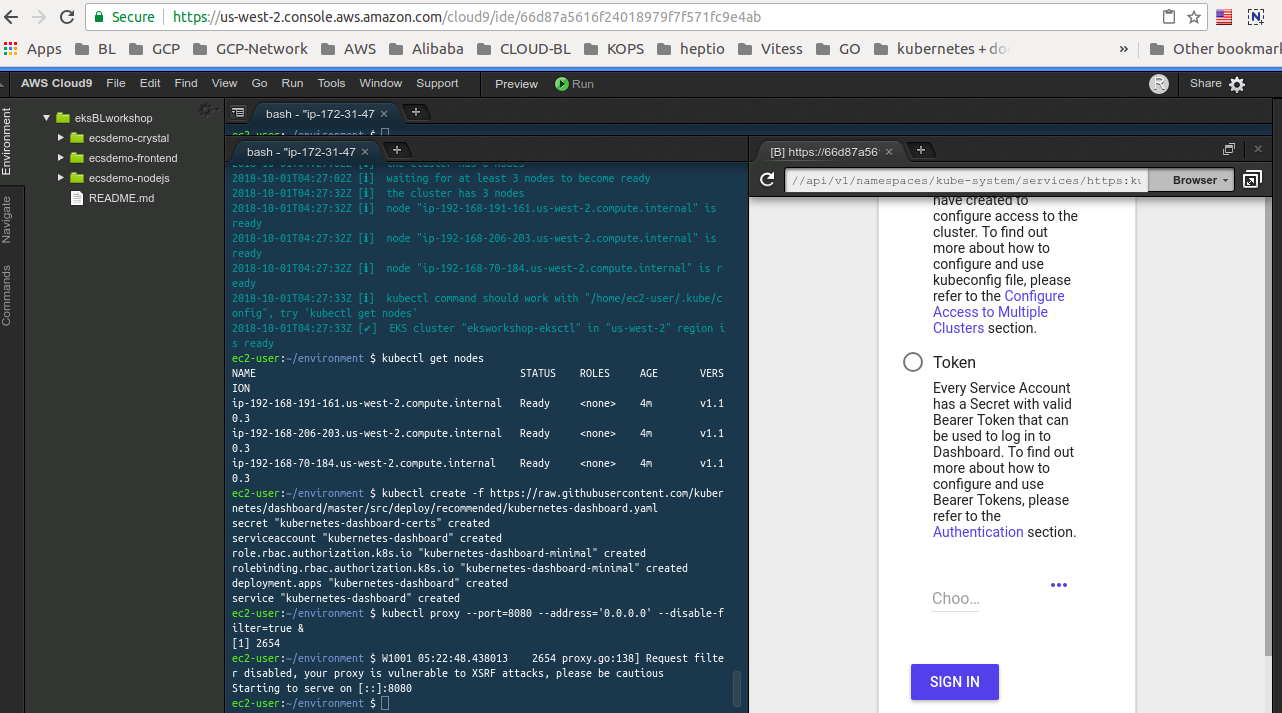
Since this is deployed to our private cluster, we need to access it via a proxy. Kube-proxy is available to proxy our requests to the dashboard service. In your workspace, run the following command:

ec2-user:~/environment $ kubectl proxy --port=8080 --address='0.0.0.0' --disable-filter=true &

# ACCESS THE DASHBOARD

Now we can access the Kubernetes Dashboard

* In your Cloud9 environment, click **Preview / Preview Running Application**
* Scroll to **the end of the URL** and append:/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

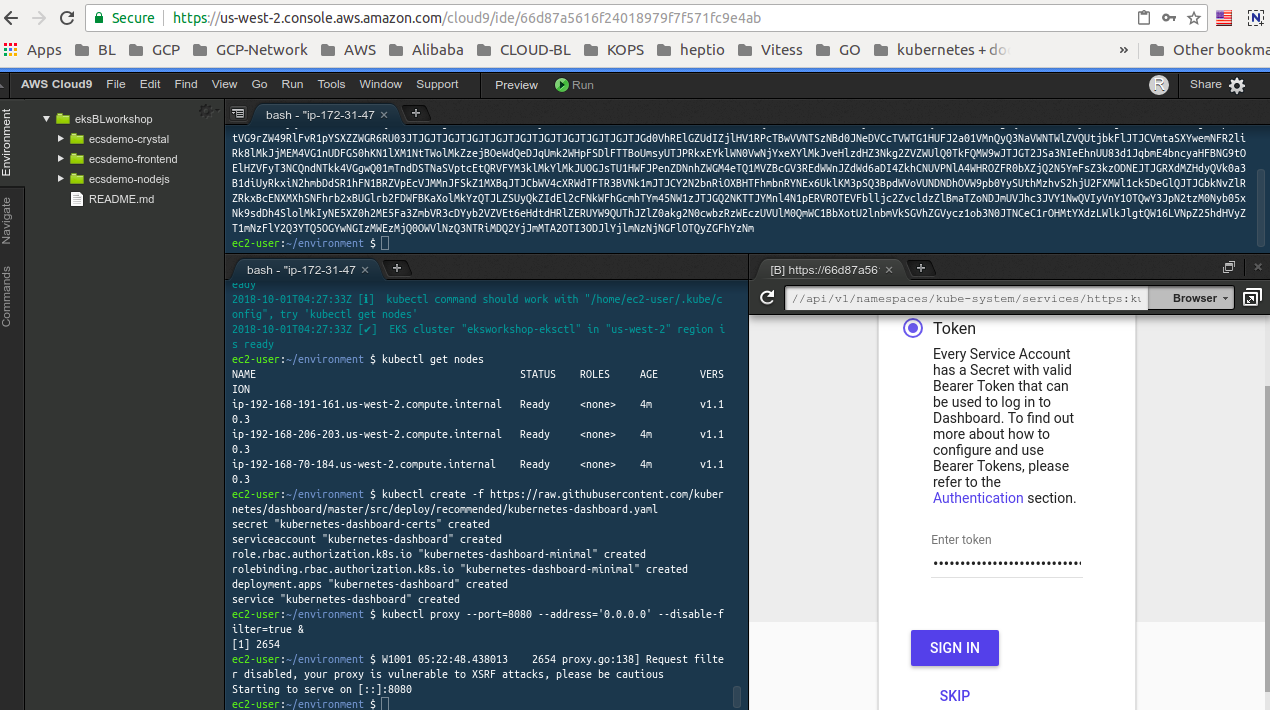


Open a New Terminal Tab and enter

ec2-user:~/environment $ aws-iam-authenticator token -i eksworkshop-eksctl –token-only



Copy the output of this command and then click the radio button next to Token then in the text field below pate the output from the last command.



Then press Sign In.

If you want to see the dashboard in a full tab, click the **Pop Out** button, like below:

