**Step wise approach for deploying application on multi master cluster**

Here we gonna see the stepwise approach of creating multi master cluster and later deploying the application on it for high availability of the application. For this we can gone a use kops where kops makes it very easy creating the multi master cluster for us.

1. Firstly, we create cluster for that we gonna have a dns-zone and s3 bucket for storage. So we will create one dns-zone and s3 bucket using aws console itself.
2. Now that we have the dns-zone and s3 bucket. We will make the cluster.

**Command-**

**kops create cluster \**

**--name=production.fulllstack.info \**

**--state=s3://multim-bucket \**

**--zones=us-east-1d,us-east-1f \**

**--master-count=3 \**

**--master-size=t2.medium \**

**--node-count=2 \**

**--node-size=t2.medium \**

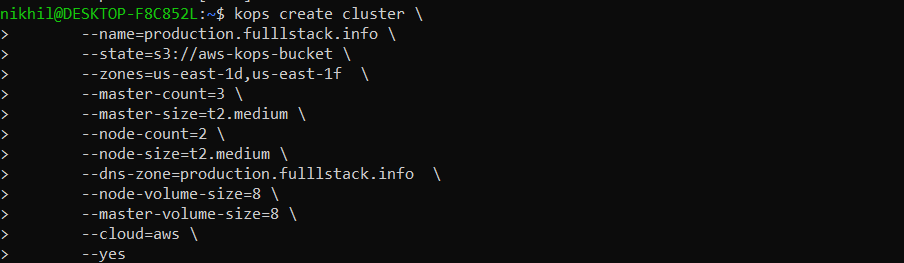
**--dns-zone=production.fulllstack.info \**

**--node-volume-size=8 \**

**--master-volume-size=8 \**

**--cloud=aws \**

**--yes**

****

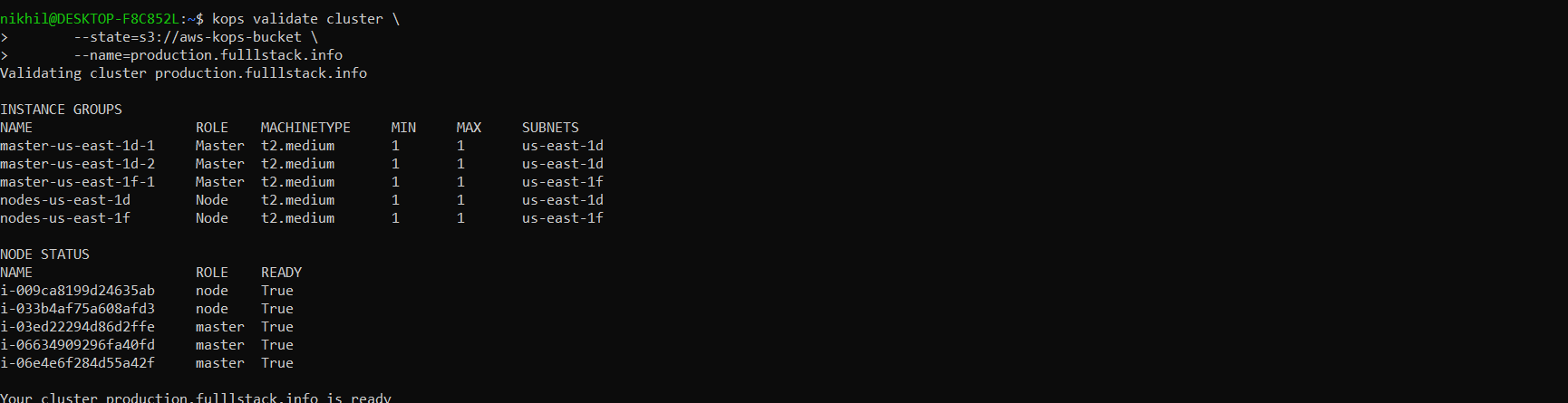
1. After this, it will take a while to steer up the cluster (10 min approx). Now, we need check whether the cluster is up or its still spinning. So, for that reason we will validate the cluster.

**Command-**

**kops validate cluster \**

**--state=s3://multim-bucket \**

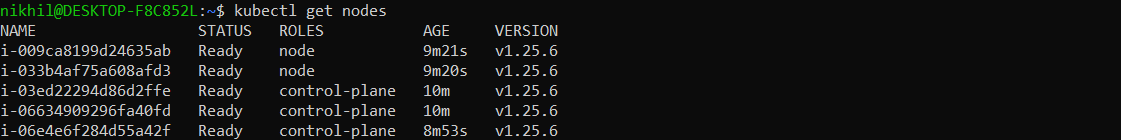
**--name=production.fulllstack.info**

****

1. Now that we have validated the cluster, for the last check we check whether the nodes are up and running. Also we check whether the master nodes and worker nodes are there corresponding to the command that we specified.

**Command-**

**kubectl get nodes**



1. Now we will apply the deployment file. One thing to remember is that you need to have the deployment file in the same directory.

**Command-**

**kubectl apply -f deployment.yml**

****

deployment.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: server-demo

spec:

replicas: 1

selector:

matchLabels:

app: web

template:

metadata:

labels:

app: web

spec:

containers:

- name: back-end

image: 655682474236.dkr.ecr.us-east-1.amazonaws.com/wordpress:latest

ports:

- containerPort: 80

---

kind: Service

apiVersion: v1

metadata:

name: backend-service

annotations:

service.beta.kubernetes.io/aws-load-balancer-type: "nlb"

spec:

type: LoadBalancer

selector:

app: web

ports:

- name: http

port: 80

targetPort: 80

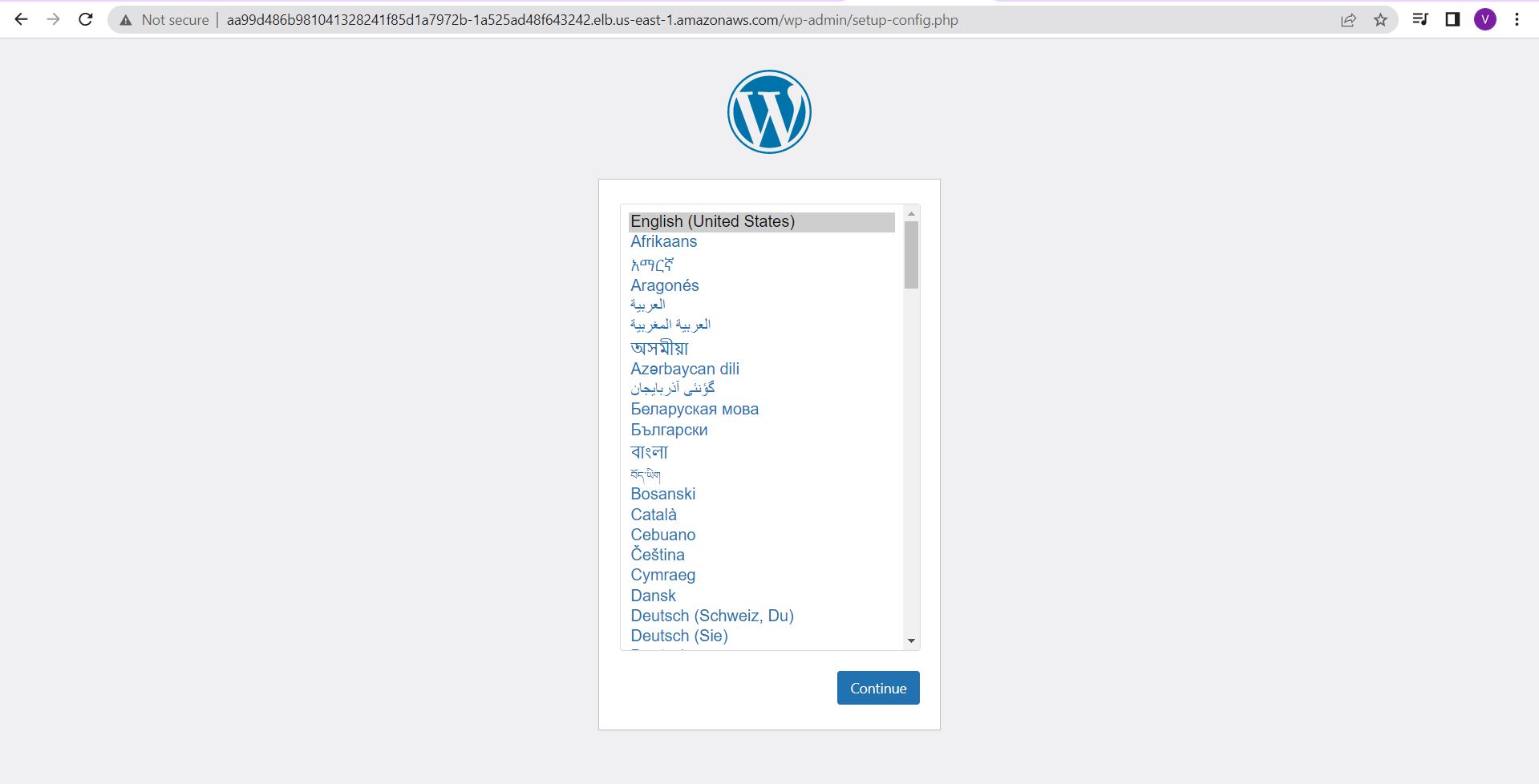
1. As we have used load balancer in the deployment file and for that reason we can use the ingress to connect to the application.

**Command-**

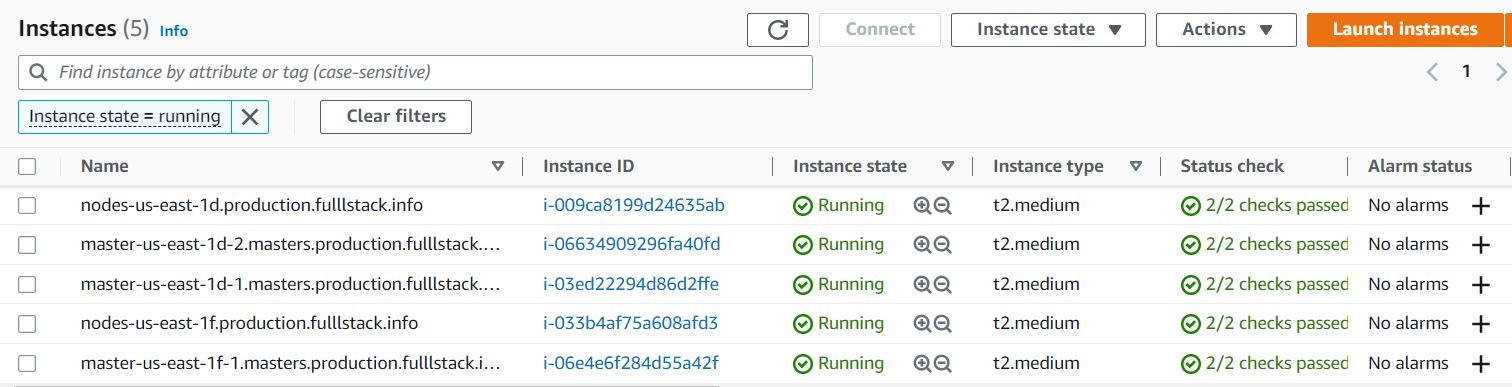
**kubectl describe svc backend-service**

****

1. Use the ingress, to connect to the application. It would be up in (2 min).



1. When done now you can check the aws console whether multi master are deployed or not.



1. When done, now its time to delete the cluster.

**Command-**

**kops delete cluster \**

**--state=s3://multim-bucket \**

**--name=production.fulllstack.info \**

**--yes**

