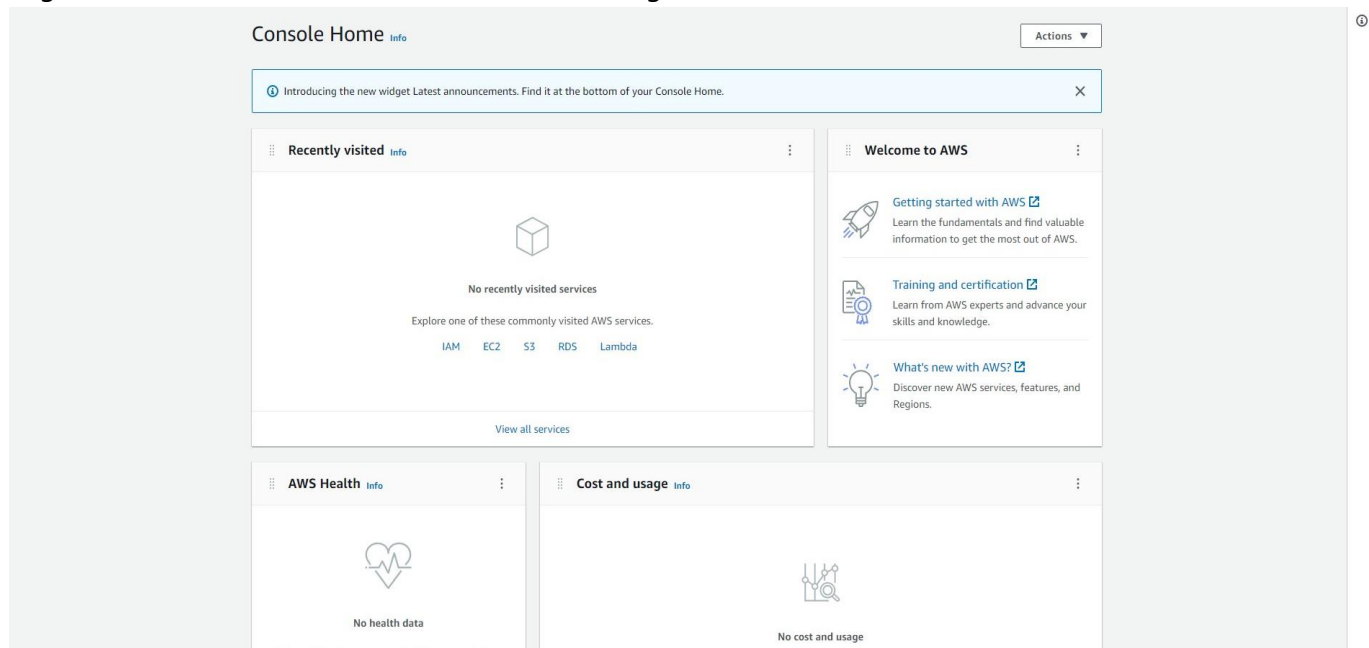

DevOps Infra Optimization Submission

All source files are found at <https://github.com/rani-salman/infra-optimization>

Log into the AWS Console. We will use this to manage and monitor our resources (EC2, EKS, etc)



Create EC2 Instance using Terraform

From the RDP session:

- ♦ Verify that terraform is installed

```
rani1salmanoutl@ip-172-31-26-39:~$ terraform version
```

```
Terraform v1.1.6  
on linux_amd64
```

```
Your version of Terraform is out of date! The latest version  
is 1.2.3. You can update by downloading from  
https://www.terraform.io/downloads.html
```

Next we export our AWS Key and ID for terraform to use

```
export AWS_ACCESS_KEY_ID="ASIAWDPX7QMXF5HTAS4B"  
export AWS_SESSION_TOKEN="  
NzrUXcqrIPODmHv+hH02P23AzCv6ThXrX3BVLxQu "  
export AWS_SECRET_ACCESS_KEY="NzrUXcqrIPODmHv+hH02P23AzCv6  
ThXrX3BVLxQuNzrUXcqrIPODmHv+hH02P23AzCv6ThXrX3BVLxQuFwo  
GZXIvYXdzEFcaDAavQuVR5n/1AFQExCK4AeYJkGcKaFZtzZ0ZWzzvGf
```

AWS DevOps in AWS V2 This Lab will get reset on 20th March 2023, 3:53 PM

Current Lab : AWS Certification - Dedicated Account

Access Information Lab Details Components Log Details Usage Details

Applications

AWS Web Console AWS API Access

AWS API Access

Access Key ASIAWDPK7QM5HTA54B Secret Key FwoGZXivYXdzEFcaDAavQuVR5n/IAFQE Security Token

Session Expires in: 7h 58m 7s Refresh Link

1. Session Duration is for 8 Hours. Post the session duration all the resources will be cleaned up automatically.
2. Auth URL enables Single-Sign-On, so the URL will vary for each session and the same URL will not work next time. Refresh the Access Details page if the session

Powered by CORESTACK

Amazon Web Services (AWS) offers a suite of cloud-computing services that make up an on-demand computing platform. AWS has more than 70 services, spanning a wide range, including compute, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools and tools for the Internet of things.

TERMINATE LAB ACCESS

Terms & Conditions

These values were pulled from the AWS API Access tab in the lab page

Create aws.tf

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ cat aws.tf

provider "aws" {
  region      = "us-east-1"
}
```

initialize Terraform project

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.20.0...
- Installed hashicorp/aws v4.20.0 (signed by HashiCorp)
```

Terraform has created a lock file `.terraform.lock.hcl` to record the provider selections it made above. Include this file `in` your version control repository so that Terraform can guarantee to make the same selections by default when you run `"terraform init"` `in` the future.

Terraform has been successfully initialized!

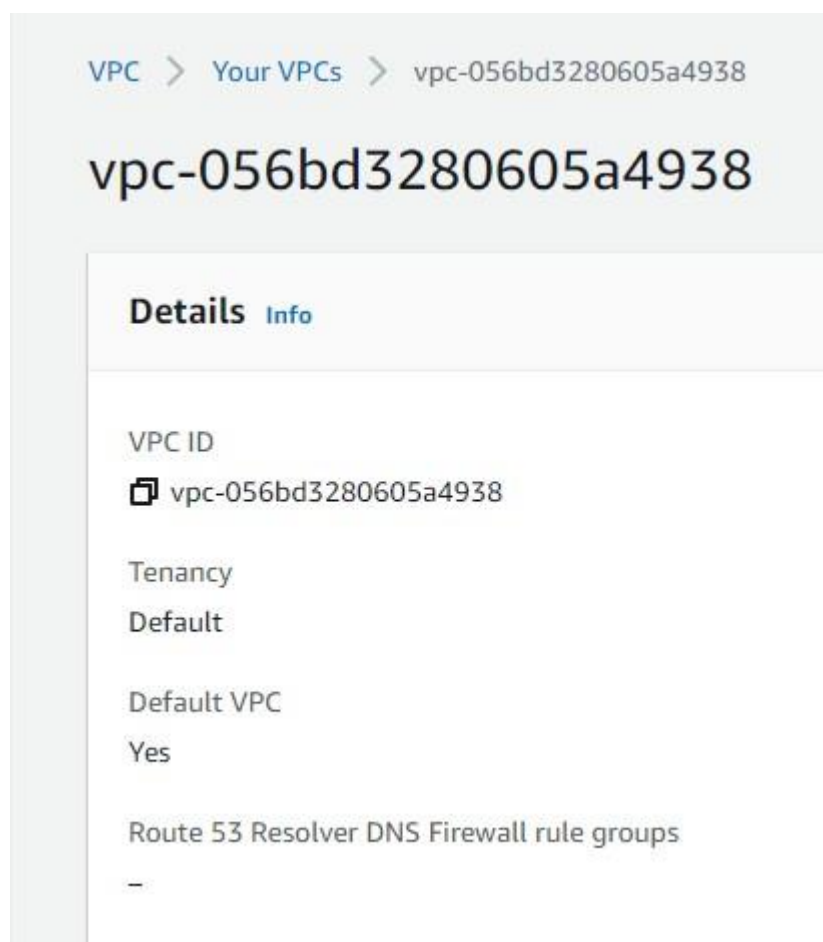
You may now begin working with Terraform. Try running `"terraform plan"` to see any changes that are required `for` your infrastructure. All Terraform commands should now work.

If you ever `set` or change modules or backend configuration `for` Terraform, rerun this `command` to reinitialize your working directory. If you forget, other commands will detect it and remind you to `do` so `if` necessary.

We can verify that the terraform project initialization files are created

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ls -a
.  ..  .terraform  .terraform.lock.hcl  aws.tf
```

Obtain VPC ID `vpc-056bd3280605a4938`



The screenshot shows the AWS Management Console interface for a VPC. The breadcrumb navigation at the top reads "VPC > Your VPCs > vpc-056bd3280605a4938". Below this, the VPC ID "vpc-056bd3280605a4938" is displayed in a large font. A tabbed interface shows "Details" as the active tab, with "Info" as an alternative. The details section lists the following attributes:

- VPC ID:** vpc-056bd3280605a4938 (with a copy icon)
- Tenancy:** Default
- Default VPC:** Yes
- Route 53 Resolver DNS Firewall rule groups:** -

Create `main.tf` that will define your resources

NOTE: Due to the requirements of kubeadm, the stated t2.micro and t3.micro do not have enough resources to satisfy minimum system requirements. Therefore, t2.medium is being used.

```
cat main.tf

resource "aws_instance" "ubuntu" {
  ami          = "ami-052efd3df9dad4825"
  instance_type = "t2.medium"
  key_name     = "${aws_key_pair.generated_key.key_name}"
  tags = {
    Name = "terraform_instance"
  }
}

output "myEC2IP" {
  value = "${aws_instance.ubuntu.public_ip}"
}

resource "tls_private_key" "example" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

resource "aws_key_pair" "generated_key" {
  key_name     = "mykey2"
  public_key   = tls_private_key.example.public_key_openssh
}

provisioner "local-exec" { # Create "myKey.pem" to your computer!!
  command = "echo '${tls_private_key.example.private_key_pem}' > ./myKey.pem"
}
}
```

Validate and deploy EC2 Instance

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ terraform plan
...

rani1salmanoutl@ip-172-31-26-39:~/capstone$ terraform apply
...

aws_instance.rhel: Creation complete after 42s [id=i-0c95e1e36e5107a67]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

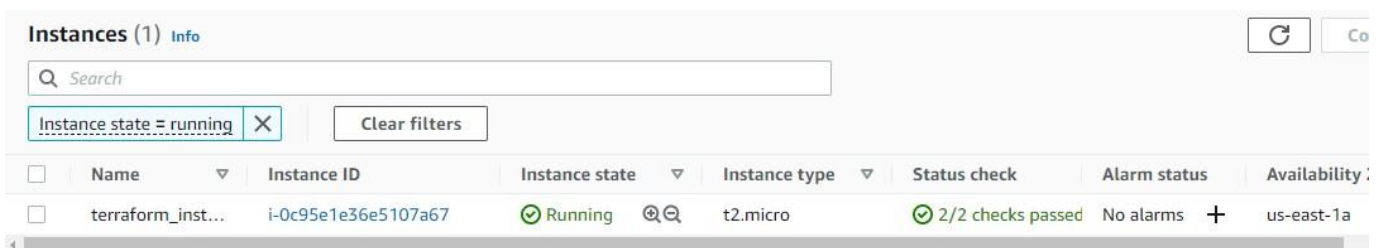
Outputs:
```

```
myEC2IP = "54.83.74.19"
```

We can test this by ssh to the new EC2 instance (make sure to change the permission of the key)

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ chmod 600 myKey.pem
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -i myKey.pem
ubuntu@54.83.74.19 [ec2-user@ip-172-31-26-39~]$ whoami && hostname
ec2-user
ip-172-31-26-39.ec2.internal
```

We can check the status from the AWS console



The screenshot shows the AWS Management Console 'Instances' page. It displays one instance named 'terraform_inst...' with ID 'i-0c95e1e36e5107a67'. The instance is in a 'Running' state, using a 't2.micro' instance type. The status checks show '2/2 checks passed', and there are 'No alarms'. The availability zone is 'us-east-1a'.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	terraform_inst...	i-0c95e1e36e5107a67	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

Now we will scale up by adding these 2 lines to `main.tf`

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ diff

ami          = "ami-052efd3df9dad4825"
+ count = 3
  instance_type = "t2.medium"

...

output "myEC2IP" {
- value = "${aws_instance.ubuntu.public_ip}"
+ value = "${aws_instance.ubuntu.*.public_ip}"
}
```

Now run apply to get the additional instances

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ terraform apply
```

Instances (3) Info								
<input type="text" value="Search"/>								
Instance state (client) != terminated × Clear filters								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	
<input type="checkbox"/>	terraform_inst...	i-02729fac74e3a2fac	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	
<input type="checkbox"/>	terraform_inst...	i-07c5c1d4e313b77ed	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	
<input type="checkbox"/>	terraform_inst...	i-0cab771f33186fc2e	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	

From this point forward, I've added the following to my local `/etc/hosts` file to make hosts more readable.

```
...

54.226.15.188 master
54.167.63.97 node1
34.203.223.52 node2

...
```

Install and Configure Kubernetes

All the needed steps to install kubeadm and start our control plane (master) nodes are in the script `install_k8s.sh`

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ cat

install_k8s.sh #!/bin/bash
swapoff -a
curl -fsSL https://get.docker.com -o get-docker.sh
DRY_RUN=1 sudo sh ./get-docker.sh
sudo apt-get install -y apt-transport-https ca-certificates curl
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-cache madison kubeadm
sudo apt-get install -y kubelet=1.23.6-00 kubeadm=1.23.6-00 kubectl=1.23.6-00
sudo hostnamectl set-hostname master.example.com
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

EOF

```
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

sudo kubeadm init
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
kubectl get nodes
```

Copy to master node

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ chmod +x install_k8s.sh
rani1salmanoutl@ip-172-31-26-39:~/capstone$ scp -p -i myKey.pem
install_k8s.sh ubuntu@master:/home/ubuntu
install_k8s.sh
100% 1106      2.2MB/s   00:00
```

Execute the script on the remote system

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -i myKey.pem ubuntu@master
"./install_k8s.sh"

...

kubeadm join 107.21.26.81:6443 --token e8w6ko.4q91edczl30a6ar5 \
--discovery-token-ca-cert-hash
sha256:1d7f217f7c8dc989b8328546559959298e959584d4fab71828c578b90533abdb
NAME                STATUS      ROLES                  AGE    VERSION
master.example.com   NotReady   control-plane,master   3s     v1.23.6
```

Now we must set up our worker nodes First we must allow TCP traffic within subnet

Within **EC2 > Security Groups > sg-0a5402b40de29840d - allow_ssh2 > Edit inbound rules**

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sg-0ace039834c30203e	All TCP ▼	TCP	0 - 65535	Custom ▼	<input type="text" value="Q"/>	<input type="text" value="172.31.0.0/16 X"/>
						Delete

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ scp -p -i myKey.pem node.sh
ubuntu@node2:~
```

```
node.sh                                100%  944    1.9MB/s
00:00

rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -t -i myKey.pem ubuntu@node2
"./node.sh"

...

rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -t -i myKey.pem ubuntu@node2 "sudo
kubeadm join 172.31.26.39:6443 --token e8w6ko.4q91edczl30a6ar5 --discovery-token-
ca-cert-hash
sha256:1d7f217f7c8dc989b8328546559959298e959584d4fab71828c578b90533abdb"

...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ scp -p -i myKey.pem
node.sh
node.sh                                100%  944    1.9MB/s
00:00

rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -t -i myKey.pem ubuntu@node1
"./node.sh"

...

rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -t -i myKey.pem ubuntu@node1 "sudo
kubeadm join 172.31.16.119:6443 --token e8w6ko.4q91edczl30a6ar5 --discovery-token-
ca-cert-hash
sha256:1d7f217f7c8dc989b8328546559959298e959584d4fab71828c578b90533abdb"

...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -i myKey.pem ubuntu@master "kubectl
get nodes"
```


NAME	STATUS	ROLES	AGE	VERSION
master.example.com	NotReady	control-plane,master	110m	v1.23.6
node1.example.com	NotReady	<none>	8m32s	v1.23.6
node2.example.com	NotReady	<none>	74s	v1.23.6

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -i myKey.pem ubuntu@master
ubuntu@node1:~$ kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```

```
rani1salmanoutl@ip-172-31-26-39:~/capstone$ ssh -i myKey.pem ubuntu@master "kubectl
get nodes"
```

NAME	STATUS	ROLES	AGE	VERSION
master.example.com	Ready	control-plane,master	2h21m	v1.23.6
node1.example.com	Ready	<none>	122m	v1.23.6
node2.example.com	Ready	<none>	101m	v1.23.6

Deploy application

```
rani1salmanoutl@ip-172-31-26-39 ssh -i myKey.pem
ubuntu@master ubuntu@master:~$ kubectl create -f mydb.yml
ubuntu@master:~$ kubectl expose deployment mydb --port=3306
```

```
ubuntu@master:~$ kubectl create -f wp.yml
ubuntu@master:~$ kubectl expose deployment wp --port=80 --type=NodePort
```

```
kubectl apply -f np.yaml
```

```
ubuntu@master:~$ kubectl create serviceaccount newroleadded
serviceaccount/newroleadded created
```

```
ubuntu@master:~$ kubectl create clusterrole newroleadded --verb=get --verb=list --
verb=create --verb=update --resource=pods
clusterrole.rbac.authorization.k8s.io/newroleadded created
```

```
ubuntu@master:~$ kubectl create clusterrolebinding newroleadded --
serviceaccount=default:newroleadded --clusterrole=newroleadded
clusterrolebinding.rbac.authorization.k8s.io/newroleadded created
```

```
ubuntu@master:~$ TOKEN=$(kubectl describe secrets "$(kubectl describe
serviceaccount newroleadded | grep -i Tokens | awk '{print $2}')" | grep token: |
awk '{print $2}')
```

```
ubuntu@master:~$ kubectl config set-credentials myuser1 --token=$TOKEN
User "myuser1" set.
```

```
ubuntu@master:~$ kubectl config set-context newcontextadded --cluster=kubernetes -
-user=myuser1
Context "newcontextadded" created.
```

```
ubuntu@master:~$ kubectl config use-context newcontextadded
Switched to context "newcontextadded".
```

```
ubuntu@master:~$ kubectl auth can-i get pods --all-namespaces
yes
```

```
ubuntu@master:~$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
mydb-659c7949cd-xrw25	1/1	Running	0	14m
wp-946c66d98-8csh4	1/1	Running	0	13m

```
Error from server (Forbidden): replicationcontrollers is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource
"replicationcontrollers" in API group "" in the namespace "default"
```

```
Error from server (Forbidden): services is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "services" in
API group "" in the namespace "default"
```

```
Error from server (Forbidden): daemonsets.apps is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "daemonsets" in
API group "apps" in the namespace "default"
```

```
Error from server (Forbidden): deployments.apps is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "deployments" in
API group "apps" in the namespace "default"
```

```
Error from server (Forbidden): replicaset.apps is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "replicasets" in
API group "apps" in the namespace "default"
```

```
Error from server (Forbidden): statefulsets.apps is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "statefulsets"
in API group "apps" in the namespace "default"
```

```
Error from server (Forbidden): horizontalpodautoscalers.autoscaling is forbidden:
User "system:serviceaccount:default:newroleadded" cannot list resource
"horizontalpodautoscalers" in API group "autoscaling" in the namespace "default"
```

```

Error from server (Forbidden): cronjobs.batch is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "cronjobs" in
API group "batch" in the namespace "default"
Error from server (Forbidden): jobs.batch is forbidden: User
"system:serviceaccount:default:newroleadded" cannot list resource "jobs" in API
group "batch" in the namespace "default"

ubuntu@master:~$ kubectl auth can-i get pods --all-namespaces
yes

ubuntu@master:~$ kubectl auth can-i get deployment --all-namespaces
no

ubuntu@master:~$ kubectl config use-context kubernetes-admin@kubernetes
Switched to context "kubernetes-admin@kubernetes".

ubuntu@master:~$ kubectl auth can-i get deployment --all-namespaces
yes

```

Database snapshot

```

sudo apt update -y
sudo apt install etcd-client

```

```

ubuntu@master:~$ hostname -I
107.21.26.81 172.17.0.1 10.32.0.1

ubuntu@master:~$ kubectl get nodes -o wide
NAME                                STATUS    ROLES    AGE      VERSION    INTERNAL-IP
IP      EXTERNAL-IP  OS-IMAGE      KERNEL-VERSION  CONTAINER-RUNTIME
master.example.com    Ready     control-plane,master    4h19m    v1.23.6
172.31.16.119        <none>    Ubuntu 22.04 LTS    5.15.0-1011-aws
docker://20.10.17
node1.example.com    Ready     <none>    157m     v1.23.6
172.31.20.65        <none>    Ubuntu 22.04 LTS    5.15.0-1011-aws
docker://20.10.17
node2.example.com    Ready     <none>    150m     v1.23.6
107.21.26.81
<none>              Ubuntu 22.04 LTS    5.15.0-1011-aws    docker://20.10.17

ubuntu@master:~$ export advertise_url="172.31.16.119:2379"

ubuntu@master:~$ echo $advertise_url
107.21.26.81:2379

ubuntu@master:~$ sudo ETCDCTL_API=3 etcdctl --endpoints $advertise_url --cacert
/etc/kubernetes/pki/etcd/ca.crt --key /etc/kubernetes/pki/etcd/server.key --cert

```

```
/etc/kubernetes/pki/etcd/server.crt snapshot save test1.db
```

```
2022-03-05 10:22:50.947059 I | clientv3: opened snapshot stream; downloading
2022-03-05 10:22:51.056969 I | clientv3: completed snapshot read; closing
Snapshot saved at test1.db
```

```
ubuntu@master:~$ du -h test1.db
7.7M    test1.db
```

Set criteria such that if the memory of CPU goes beyond 50%, environments automatically get scaled up and configured

Create Metrics server

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

```
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
```

Needs a patch

```
ubuntu@master:~$ kubectl get pods -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
coredns-64897985d-64mpp	1/1	Running	0	2h26m
coredns-64897985d-v5z2r	1/1	Running	0	2h26m
kube-apiserver-master.example.com	0/1	Pending	0	1s
kube-proxy-n99wn	1/1	Running	0	182m
kube-proxy-p4swz	1/1	Running	1	2h27m
kube-proxy-rp9nb	1/1	Running	0	96m
metrics-server-847dcc659d-qtqq	0/1	Running	0	48s
weave-net-2k5kl	2/2	Running	2 (26m ago)	51m
weave-net-68qm9	0/2	Pending	0	51m
weave-net-pl9j2	2/2	Running	2 (26m ago)	51m

Apply patch

```

ubuntu@master:~$ wget -c
https://gist.githubusercontent.com/initcron/1a2bd25353e1faa22a0ad41ad1c01b62/raw/0
08e23f9fbf4d7e2cf79df1dd008de2f1db62a10/k8s-metrics-server.patch.yaml
--2023-03-05 10:52:30--
https://gist.githubusercontent.com/initcron/1a2bd25353e1faa22a0ad41ad1c01b62/raw/0
08e23f9fbf4d7e2cf79df1dd008de2f1db62a10/k8s-metrics-server.patch.yaml
Resolving gist.githubusercontent.com (gist.githubusercontent.com)...
185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to gist.githubusercontent.com
(gist.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 205 [text/plain]
Saving to: 'k8s-metrics-server.patch.yaml'

```

```

k8s-metrics-server.patch.yaml                                100%
[=====]
=====>]           205  --.-KB/s    in  0s

```

```

2023-03-05 13:22:12 (14.0 MB/s) - 'k8s-metrics-server.patch.yaml' saved [205/205]

```

```

ubuntu@master:~$ kubectl patch deploy metrics-server -p "$(cat k8s-metrics-
server.patch.yaml)" -n kube-system

```

```

deployment.apps/metrics-server patched

```

```

ubuntu@master:~$ kubectl get pods -n kube-system

```

NAME	READY	STATUS	RESTARTS	AGE
coredns-64897985d-64mpp	1/1	Running	0	3h55m
coredns-64897985d-v5z2r	1/1	Running	0	3h55m
etcd-master.example.com	1/1	Running	1 (3h48m ago)	25m
kube-apiserver-master.example.com	1/1	Running	1 (3h48m ago)	25m
kube-controller-manager-master.example.com	1/1	Running	1 (3h48m ago)	25m
kube-proxy-n99wn	1/1	Running	0	123m
kube-proxy-p4swz	1/1	Running	1 (3h48m ago)	3h55m
kube-proxy-rp9nb	1/1	Running	0	152m
kube-scheduler-master.example.com	1/1	Running	1 (3h48m ago)	25m
metrics-server-77b7f4f884-g5jl5	1/1	Running	0	32s
weave-net-2k5kl	2/2	Running	2 (44m ago)	77m
weave-net-68qm9	2/2	Running	0	77m
weave-net-pl9j2	2/2	Running	2 (4m ago)	77m

Now that the metrics server is running we can apply out horizational scaling policy

```
ubuntu@master:~$ kubectl apply -f hpa.yaml
```

```
ubuntu@master:~$ kubectl get horizontalpodautoscaler
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
php-apache	Deployment/wp	0%/50%	1	10	1	23s

Create Load balancer

[Create Load Balancer](#)[Actions](#) ▼**Name****DNS name**

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-056bd3280605a4938
IPv4: 172.31.0.0/16



Mappings [Info](#)

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

☒ **us-east-1a**

Subnet

subnet-00c99ae19bacd8d1d

IPv4 settings

Assigned by AWS

☒ **us-east-1b**

Subnet

subnet-0ca3b7994c3ca3831

IPv4 settings

Assigned by AWS

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

☒ Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-056bd3280605a4938
IPv4: 172.31.0.0/16



Mappings [Info](#)

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

☒ us-east-1a

Subnet

subnet-00c99ae19bacd8d1d

IPv4 settings

Assigned by AWS

Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:31478

Remove

Protocol

HTTP

Port

31478
1-65535

Default action [Info](#)

Forward to

capstone-instances
Target type: Instance, IPv4

HTTP

[Create target group](#)

Add listener

Security groups [Info](#)



A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select up to 5 security groups 



Create new security group [🔗](#)

default sg-0c930c5d873c9b2a0 	allow_ssh2 sg-0a5402b40de29840d 
VPC: vpc-056bd3280605a4938	VPC: vpc-056bd3280605a4938

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type



Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#)  to manage and scale your EC2 capacity.



IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.



Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.



Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Port



:

VPC

Select the VPC with the instances that you want to include in the target group.

-
vpc-056bd3280605a4938
IPv4: 172.31.0.0/16



Protocol version



HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.



HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.



gRPC

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/healthz

Up to 1024 characters allowed.

► Advanced health check settings

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (3/3)

Filter resources by property or value

< 1 > ⌕

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Subnet ID
<input checked="" type="checkbox"/>	i-02a2d718bd3bc4e	terraform_instance	running	allow_ssh2	us-east-1b	subnet-0ca3b7994c3ca3831
<input checked="" type="checkbox"/>	i-0f531cb0547fafbb5	terraform_instance	running	allow_ssh2	us-east-1b	subnet-0ca3b7994c3ca3831
<input checked="" type="checkbox"/>	i-009f0515f0b3468b1	terraform_instance	running	allow_ssh2	us-east-1b	subnet-0ca3b7994c3ca3831

3 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

31406

1-65535 (separate multiple ports with comma)

Include as pending below

Review targets

Targets (0)

Remove all pending

All

Filter resources by property or value

< 1 > ⌕

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
No instances added yet								

Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Cancel

Previous

Create target group

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)

devops-capstone

- Internet-facing
- IPv4

Security groups [Edit](#)

- default
[sg-0c930c5d873c9b2a0](#)
- allow_ssh2
[sg-0a5402b40de29840d](#)

Network mapping [Edit](#)

VPC [vpc-056bd3280605a4938](#)

- us-east-1a
[subnet-00c99ae19bacd8d1d](#)
- us-east-1b
[subnet-0ca3b7994c3ca3831](#)

Listeners and routing [Edit](#)

- HTTP:31478 defaults to
[capstone-instances](#)

Add-on services [Edit](#)

None

Tags [Edit](#)

None

Attributes

ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Cancel

Create load balancer

Create Load Balancer

Actions

search : devops-capstone

Add filter

	Name	DNS name	State	VPC ID	Availability Zones	Type
	devops-capstone	devops-capstone-18736801...	Provisioning	vpc-056bd3280605a4938	us-east-1a, us-east-1b	application

←

→

↺

http://devops-capstone-1873680155.us-east-1.elb.amazonaws.com:32180/wp-admin/install.php

NetBenefits Login P...

CCNA Labs

Simplilearn - DevO...

Presken Family Care

Google Docs

W

English (United States)

Afrikaans

አማርኛ

Aragonés

العربية

العربية المغربية

অসমীয়া

گۆنئی آذربایجان

Azərbaycan dili

Беларуская мова

Български

বাংলা

བོད་ཡིག

Bosanski

Català

Cebuano

Čeština

Cymraeg

Dansk