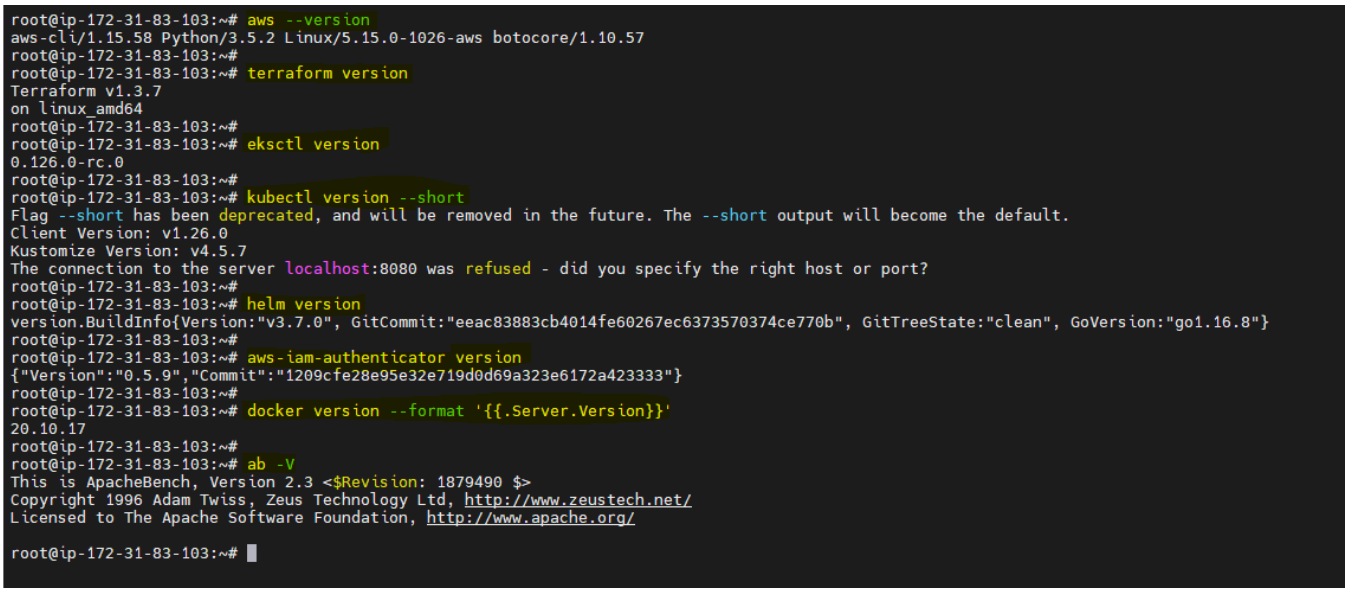
Capstone Project Submission Documentation

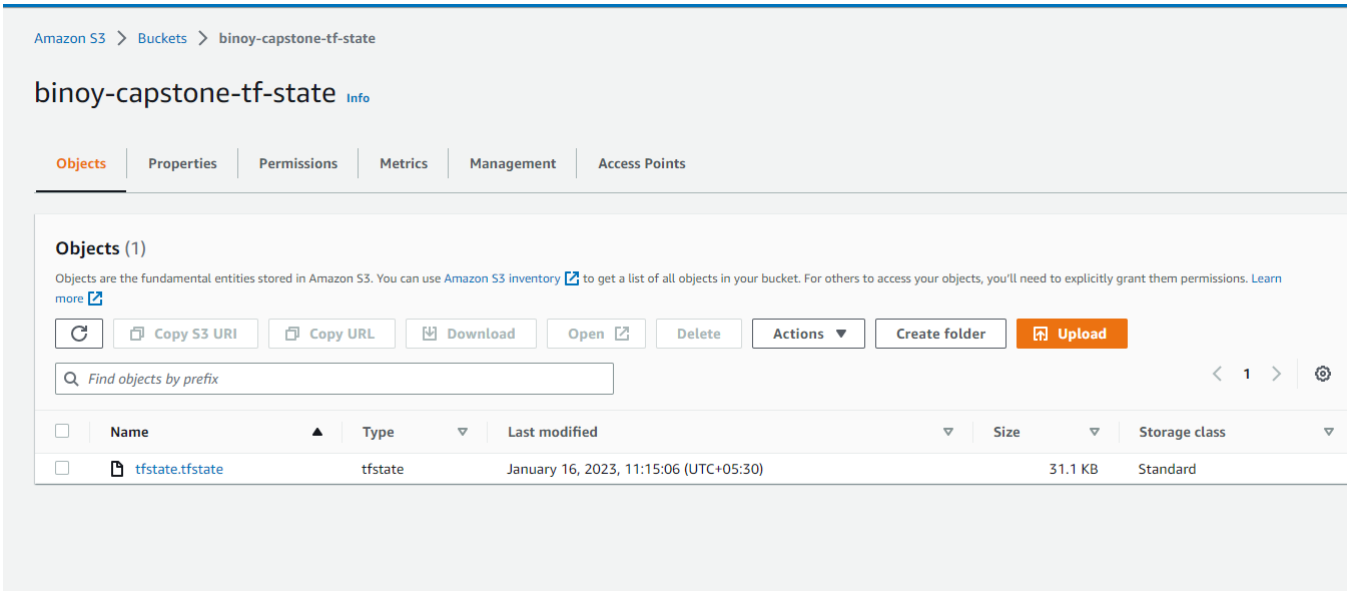
Task 0: Environment Setup

Installed all the requested tools on a linux machine



Task 1: Setup EKS Cluster

SubTask 1
S3 Bucket created for terraform backend store


Amazon S3 > Buckets > binoy-capstone-tf-state
binoy-capstone-tf-state Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
tfstate.tfstate
tfstate
January 16, 2023, 11:15:06 (UTC+05:30)
31.1 KB
Standard

Subtask 2

Create AWS resources using Terraform

```
root@ip-172-31-83-103:~/terraform# terraform init
Initializing modules...
Downloading registry.terraform.io/terraform-aws-modules/vpc/aws 3.19.0 for vpc...
- vpc in .terraform/modules/vpc

Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Finding hashicorp/aws versions matching ">= 3.73.0"...
- Installing hashicorp/aws v4.50.0...
- Installed hashicorp/aws v4.50.0 (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.
root@ip-172-31-83-103:~/terraform#
```

```
root@ip-172-31-83-103:~/terraform# terraform apply
module.vpc.aws_eip.nat[0]: Refreshing state... [id=eipalloc-0675c959933a44c0d]
module.vpc.aws_eip.nat[1]: Refreshing state... [id=eipalloc-08d0a5846d1018811]
module.vpc.aws_vpc.this[0]: Refreshing state... [id=vpc-03aa2627fa0ec797f]
module.vpc.aws_subnet.private[0]: Refreshing state... [id=subnet-0359345a9ee7d967b]
module.vpc.aws_subnet.public[0]: Refreshing state... [id=subnet-0e5451c312dadb934]
module.vpc.aws_route_table.private[1]: Refreshing state... [id=rtb-0f95fbcb1aaf026cc]
module.vpc.aws_internet_gateway.this[0]: Refreshing state... [id=igw-06e3e52aaa3925f1d]
module.vpc.aws_route_table.public[0]: Refreshing state... [id=rtb-0159e5063695aead3]
module.vpc.aws_subnet.public[1]: Refreshing state... [id=subnet-03ccf0c210fa2b318]
module.vpc.aws_subnet.private[1]: Refreshing state... [id=subnet-0e774adbe78126b72]
module.vpc.aws_route_table.private[0]: Refreshing state... [id=rtb-082d50a5615d7a07a]
module.vpc.aws_route.public_internet_gateway[0]: Refreshing state... [id=r-rtb-0159e5063695aead31080289494]
module.vpc.aws_nat_gateway.this[1]: Refreshing state... [id=nat-0d9a3c6337613b70e]
module.vpc.aws_route_table_association.public[1]: Refreshing state... [id=rtbassoc-074c19f1b2f51b273]
module.vpc.aws_route_table_association.public[0]: Refreshing state... [id=rtbassoc-07bdb76bc74fb47b4]
module.vpc.aws_nat_gateway.this[0]: Refreshing state... [id=nat-0d3677544de2402a1]
module.vpc.aws_route_table_association.private[1]: Refreshing state... [id=rtbassoc-0e69813cbbd2ad85c]
module.vpc.aws_route_table_association.private[0]: Refreshing state... [id=rtbassoc-0dc233ce8f09c42fb]
module.vpc.aws_route.private_nat_gateway[0]: Refreshing state... [id=r-rtb-082d50a5615d7a07a1080289494]
module.vpc.aws_route.private_nat_gateway[1]: Refreshing state... [id=r-rtb-0f95fbcb1aaf026cc1080289494]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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

Outputs:

private_subnets = [
  "subnet-0359345a9ee7d967b",
  "subnet-0e774adbe78126b72",
]
public_subnets = [
  "subnet-0e5451c312dadb934",
  "subnet-03ccf0c210fa2b318",
]
vpc_id = "vpc-03aa2627fa0ec797f"
root@ip-172-31-83-103:~/terraform#
```

Subtask 3
Creation of EKS Cluster

```
root@ip-172-31-83-103:~/eksctl# kubectl get no -n kube-system
NAME                             STATUS   ROLES    AGE     VERSION
ip-10-0-1-106.ec2.internal   Ready    <none>   3h13m   v1.24.7-eks-fb459a0
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# eksctl get cluster
NAME            REGION          EKSCTL CREATED
my-eks-201      us-east-1       True
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# eksctl get nodegroup --cluster my-eks-201
CLUSTER         NODEGROUP       STATUS          CREATED                 MIN SIZE        MAX SIZE        DESIRED CAPACITY        INSTANCE TYPE   IMAGE ID                ASG NAME                                                        TYPE
my-eks-201      pub-201-a-1     CREATE_COMPLETE 2023-01-14T08:10:08Z    0               1               0                       t2.medium       ami-06c9b6a12f5bd0a96   eksctl-my-eks-201-nodegroup-pub-201-a-1-NodeGroup-15IE74RH2ZR4M unmanaged
my-eks-201      pvt-201-a-1     CREATE_COMPLETE 2023-01-14T08:10:08Z    0               3               1                       t2.medium       ami-06c9b6a12f5bd0a96   eksctl-my-eks-201-nodegroup-pvt-201-a-1-NodeGroup-22ZR6AY4OAJ5  unmanaged
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# eksctl get iamidentitymapping --cluster my-eks-201
ARN                                                                             USERNAME                                GROUPS                                  ACCOUNT
arn:aws:iam::223316186016:role/eksctl-my-eks-201-nodegroup-pub-2-NodeInstanceRole-1HNNS0LNLUWH5 system:node:{{EC2PrivateDNSName}}      system:bootstrappers,system:nodes
arn:aws:iam::223316186016:role/eksctl-my-eks-201-nodegroup-pvt-2-NodeInstanceRole-10JXFJFTIWONN system:node:{{EC2PrivateDNSName}}      system:bootstrappers,system:nodes
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl#
```

Subtask 4: Addons to EKS Cluster

AWS Load Balancer Controller v2

```
root@ip-172-31-83-103:~/eksctl# eksctl utils associate-iam-oidc-provider \
    --region us-east-1 \
    --cluster my-eks-201 \
    --approve
2023-01-14 08:57:05 [ℹ]  IAM Open ID Connect provider is already associated with cluster "my-eks-201" in "us-east-1"
root@ip-172-31-83-103:~/eksctl# curl -o iam-policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.2.1/docs/install/iam_policy.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7273  100  7273    0     0  39219      0 --:--:-- --:--:-- --:--:-- 39527
root@ip-172-31-83-103:~/eksctl# aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://iam-policy.json
{
    "Policy": {
        "AttachmentCount": 0,
        "DefaultVersionId": "v1",
        "UpdateDate": "2023-01-14T08:57:37Z",
        "PolicyId": "ANPATH7VOSOQJRBD34CXT",
        "PolicyName": "AWSLoadBalancerControllerIAMPolicy",
        "Arn": "arn:aws:iam::223316186016:policy/AWSLoadBalancerControllerIAMPolicy",
        "Path": "/",
        "IsAttachable": true,
        "PermissionsBoundaryUsageCount": 0,
        "CreateDate": "2023-01-14T08:57:37Z"
    }
}
root@ip-172-31-83-103:~/eksctl# eksctl create iamserviceaccount \
--cluster=my-eks-201 \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--attach-policy-arn=arn:aws:iam::223316186016:policy/AWSLoadBalancerControllerIAMPolicy \
--override-existing-serviceaccounts \
--approve
2023-01-14 08:58:30 [ℹ]  7 existing iamserviceaccount(s) (cert-manager/cert-manager,kube-system/aws-load-balancer-controller,kube-system/aws-node,kube-system/cluster-autoscaler,kube-s
oller-sa,kube-system/external-dns) will be excluded
2023-01-14 08:58:30 [ℹ]  1 iamserviceaccount (kube-system/aws-load-balancer-controller) was excluded (based on the include/exclude rules)
2023-01-14 08:58:30 [!]  metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2023-01-14 08:58:30 [ℹ]  no tasks
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories
root@ip-172-31-83-103:~/eksctl# kubectl apply -k "github.com/aws/eks-charts/stable/aws-load-balancer-controller//crds?ref=master"
customresourcedefinition.apiextensions.k8s.io/ingressclassparams.elbv2.k8s.aws created
customresourcedefinition.apiextensions.k8s.io/targetgroupbindings.elbv2.k8s.aws created
root@ip-172-31-83-103:~/eksctl# helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system --set clusterName=<cluster-name> --set serviceAccount.create=
false --set serviceAccount.name=aws-load-balancer-controller
-bash: cluster-name: No such file or directory
root@ip-172-31-83-103:~/eksctl# helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system --set clusterName=my-eks-201 --set serviceAccount.create=fals
e --set serviceAccount.name=aws-load-balancer-controller
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Jan 14 09:00:17 2023
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
```

```
root@ip-172-31-83-103:~/eksctl# kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           99s
root@ip-172-31-83-103:~/eksctl#
```

Kubernetes-metric-server

```
root@ip-172-31-83-103:~/eksctl# 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
root@ip-172-31-83-103:~/eksctl#
```

```
root@ip-172-31-83-103:~/eksctl# kubectl get deployment -n kube-system metrics-server
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           40s
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# kubectl get svc -n kube-system metrics-server
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
metrics-server   ClusterIP   172.20.80.186    <none>        443/TCP   64s
root@ip-172-31-83-103:~/eksctl#
```

Cluster Auto-scaler

```
root@ip-172-31-83-103:~/eksctl# kubectl -n kube-system logs --tail=10 deployment.apps/cluster-autoscaler
I0116 07:01:22.631758       1 filter_out_schedulable.go:87] No schedulable pods
I0116 07:01:22.631785       1 static_autoscaler.go:432] No unschedulable pods
I0116 07:01:22.631810       1 static_autoscaler.go:479] Calculating unneeded nodes
I0116 07:01:22.631843       1 scale_down.go:448] Node ip-10-0-1-228.ec2.internal - cpu utilization 0.064767
I0116 07:01:22.631888       1 scale_down.go:509] Scale-down calculation: ignoring 1 nodes unremovable in the last 5m0s
I0116 07:01:22.631941       1 static_autoscaler.go:522] ip-10-0-1-228.ec2.internal is unneeded since 2023-01-16 06:56:31.077245838 +0000 UTC m=+32.332062898 duration
 4m51.391170805s
I0116 07:01:22.631970       1 static_autoscaler.go:533] Scale down status: unneededOnly=false lastScaleUpTime=2023-01-16 05:56:21.075494173 +0000 UTC m=-3577.6696887
13 lastScaleDownDeleteTime=2023-01-16 05:56:21.075494173 +0000 UTC m=-3577.669688713 lastScaleDownFailTime=2023-01-16 05:56:21.075494173 +0000 UTC m=-3577.669688713
scaleDownForbidden=false isDeleteInProgress=false scaleDownInCooldown=false
I0116 07:01:22.632003       1 static_autoscaler.go:546] Starting scale down
I0116 07:01:22.632034       1 scale_down.go:828] ip-10-0-1-228.ec2.internal was unneeded for 4m51.391170805s
I0116 07:01:22.632063       1 scale_down.go:917] No candidates for scale down
root@ip-172-31-83-103:~/eksctl#
```

```
root@ip-172-31-83-103:~/eksctl# kubectl apply -f cluster-autoscaler-autodiscover.yaml
Warning: resource serviceaccounts/cluster-autoscaler is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resou
rces created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
serviceaccount/cluster-autoscaler configured
clusterrole.rbac.authorization.k8s.io/cluster-autoscaler created
role.rbac.authorization.k8s.io/cluster-autoscaler created
clusterrolebinding.rbac.authorization.k8s.io/cluster-autoscaler created
rolebinding.rbac.authorization.k8s.io/cluster-autoscaler created
deployment.apps/cluster-autoscaler created
root@ip-172-31-83-103:~/eksctl#
root@ip-172-31-83-103:~/eksctl# kubectl get deployment -n kube-system cluster-autoscaler
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
cluster-autoscaler   1/1     1            1           23s
root@ip-172-31-83-103:~/eksctl#
```

```
root@ip-172-31-83-103:~/eksctl# kubectl get deployment -n kube-system cluster-autoscaler -o wide
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS           IMAGES                                                SELECTOR
cluster-autoscaler   1/1     1            1           14m   cluster-autoscaler   k8s.gcr.io/autoscaling/cluster-autoscaler:v1.24.0   app=cluster-autoscaler
root@ip-172-31-83-103:~/eksctl#
```

```
root@ip-172-31-83-103:~/eksctl# eksctl create iamserviceaccount \
  --cluster=my-eks-201 \
  --namespace=kube-system \
  --name=cluster-autoscaler \
  --attach-policy-arn=arn:aws:iam::223316186016:policy/AmazonEKSClusterAutoscalerPolicy \
  --override-existing-serviceaccounts \
  --approve
2023-01-16 06:43:12 [ℹ]  7 existing iamserviceaccount(s) (cert-manager/cert-manager,kube-system/aws-load-balancer-controller,kube-system/aws-node,kube-system/cluster-autoscaler,kube-system/ebs-csi-controlle
r-sa,kube-system/efs-csi-controller-sa,kube-system/external-dns) will be excluded
2023-01-16 06:43:12 [ℹ]  1 iamserviceaccount (kube-system/cluster-autoscaler) was excluded (based on the include/exclude rules)
2023-01-16 06:43:12 [!]  metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2023-01-16 06:43:12 [ℹ]  no tasks
root@ip-172-31-83-103:~/eksctl#
```

Subtask Bonus:

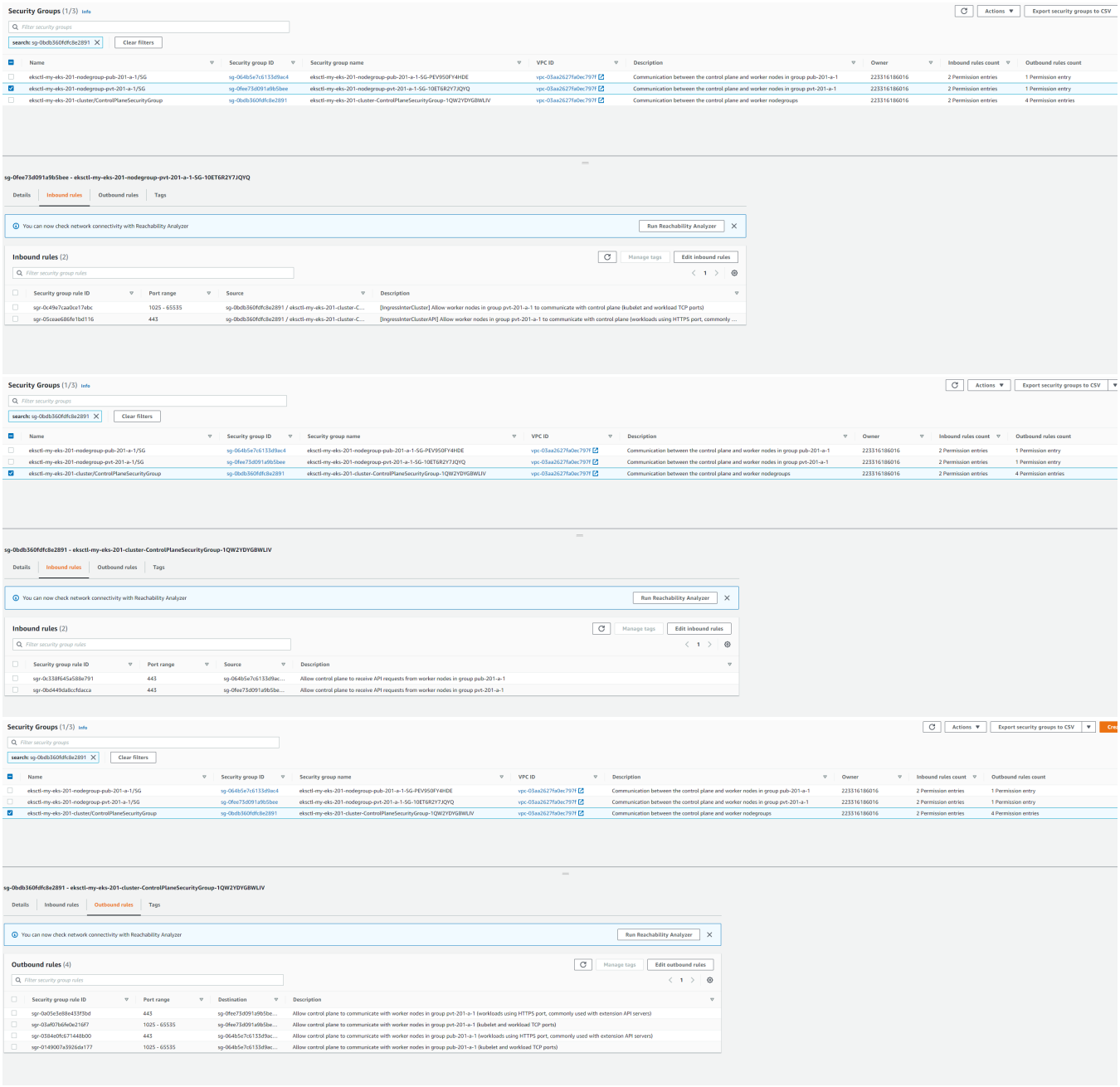
Security Groups Created

Security Group : **sg-0bdb360fdfc8e2891** was created
Name : **eksctl-my-eks-201-cluster/ControlPlaneSecurityGroup**
Role : **Communication between the control plane and worker nodegroups**

Incoming rules
Allow control plane to receive API requests from worker nodes in group pub-201-a-1 on 443
Allow control plane to receive API requests from worker nodes in group pvt-201-a-1 on 443

Outgoing Rules
Allow control plane to communicate with worker nodes in group pvt-201-a-1 on 443
Allow control plane to communicate with worker nodes in group pvt-201-a-1 on 1025 - 65535
Allow control plane to communicate with worker nodes in group pub-201-a-1 on 443
Allow control plane to communicate with worker nodes in group pub-201-a-1 on 1025 - 65535

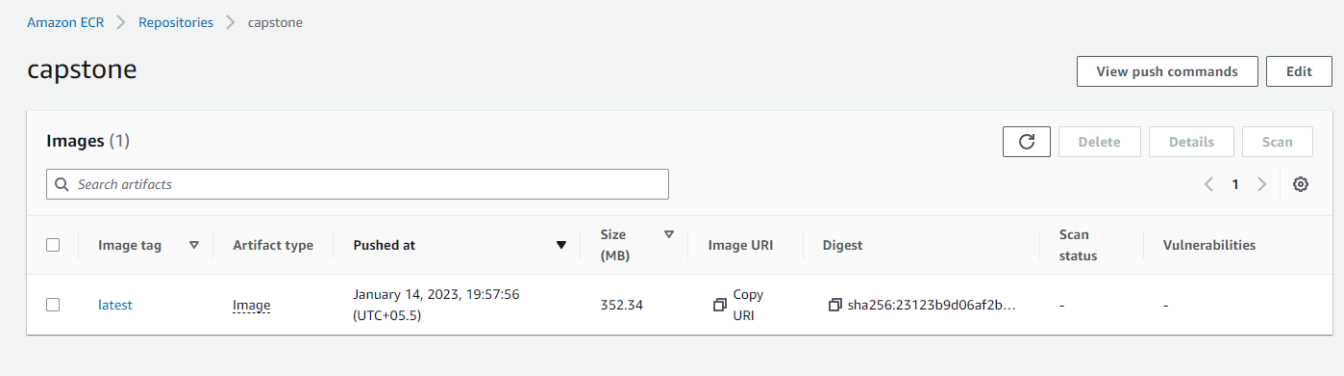
2 more security group (eksctl-my-eks-201-nodegroup-pub-201-a-1/SG and eksctl-my-eks-201-nodegroup-pvt-201-a-1/SG) for node groups
Incoming rules
[IngressInterCluster] Allow worker nodes in group pvt-201-a-1 to communicate with control plane (kubelet and workload TCP ports)
[IngressInterClusterAPI] Allow worker nodes in group pvt-201-a-1 to communicate with control plane (workloads using HTTPS port, commonly used)



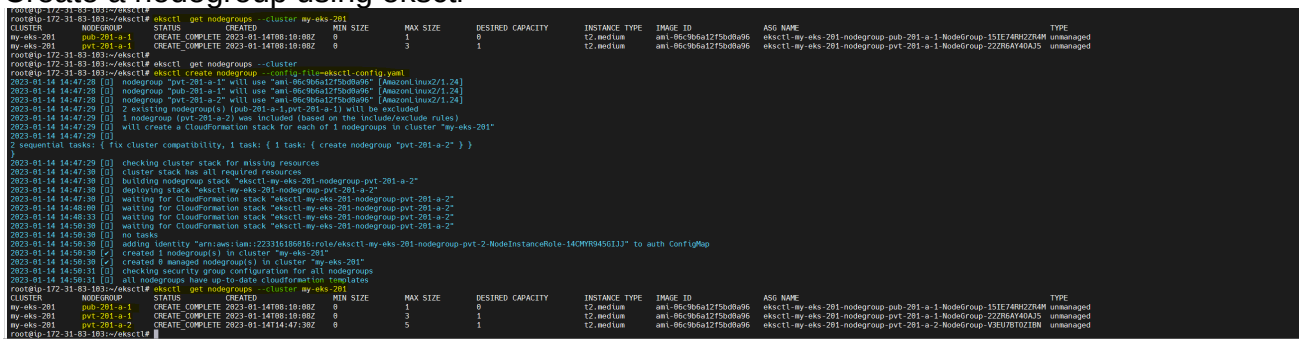
Task 2: Deployment of sample application

Subtask 1:
ECR Repository

Created the repo and pushed the image. Dockerfile in the source git repo



Subtask 2:
Create a nodegroup using eksctl



Subtask 3:
App Deployment

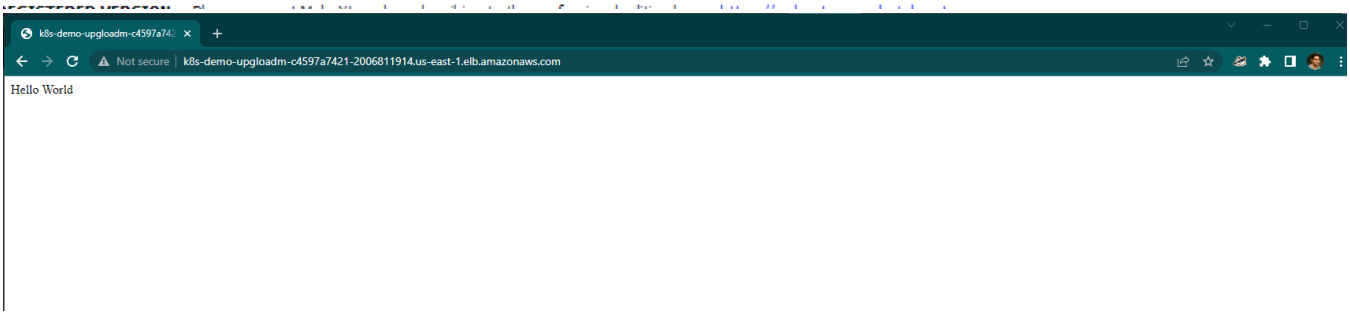
```
root@ip-172-31-83-103:~# kubectl get ns
NAME              STATUS   AGE
cert-manager      Active   8h
default           Active   8h
demo              Active   70m
kube-node-lease   Active   8h
kube-public       Active   8h
kube-system       Active   8h
root@ip-172-31-83-103:~#
```



```
root@ip-172-31-83-103:~# kubectl get deploy -n demo -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                                                          SELECTOR
upg-loadme    1/1     1            1           25m   upg-loadme   223316186016.dkr.ecr.us-east-1.amazonaws.com/capstone:latest   app=upg-loadme
root@ip-172-31-83-103:~#
```

```
root@ip-172-31-83-103:~# kubectl get service -n demo -o wide
NAME          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE   SELECTOR
upg-loadme    ClusterIP   172.20.240.191   <none>        80/TCP    26m   app=upg-loadme
root@ip-172-31-83-103:~#
```

```
root@ip-172-31-83-103:~# kubectl describe service -n demo upg-loadme
Name:              upg-loadme
Namespace:         demo
Labels:            <none>
Annotations:       <none>
Selector:          app=upg-loadme
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                172.20.240.191
IPs:               172.20.240.191
Port:              app-port  80/TCP
TargetPort:        8081/TCP
Endpoints:         10.0.1.212:8081
Session Affinity:  None
Events:            <none>
root@ip-172-31-83-103:~#
```




```
14/01/2023 21:35.30 /home/mobaxterm curl -w "\n" k8s-demo-upgloadm-c4597a7421-2006811914.us-east-1.elb.amazonaws.com:80
Hello World

14/01/2023 21:35.34 /home/mobaxterm
```

Task 3: Deploy Redis server on Kubernetes

SubTask1 :
Deploy redis server

ConfigMap

```
root@ip-172-31-83-103:~/k8s# kubectl get cm -n demo -o wide
NAME                       DATA   AGE
kube-root-ca.crt           1      4h6m
redis-ss-configuration     1      4h2m
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s# kubectl describe cm -n demo redis-ss-configuration
Name:         redis-ss-configuration
Namespace:    demo
Labels:       app=redis
Annotations:  <none>

Data
====
master.conf:
----
timeout 300
appendonly yes
protected-mode no
dbfilename dump.rdb
dir /var/lib/redis


BinaryData
====

Events:  <none>
root@ip-172-31-83-103:~/k8s#
```

Redis Server

```
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s# kubectl get all -n demo -o wide
NAME          READY   STATUS    RESTARTS   AGE     IP           NODE                           NOMINATED NODE   READINESS GATES
pod/redis-0   1/1     Running   0          17m     10.0.1.103   ip-10-0-1-222.ec2.internal   <none>           <none>
pod/redis-1   1/1     Running   0          9m17s   10.0.1.252   ip-10-0-1-222.ec2.internal   <none>           <none>

NAME                    TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE   SELECTOR
service/redis-service   ClusterIP   172.20.55.2    <none>        6379/TCP   7s    app=redis

NAME                     READY   AGE   CONTAINERS   IMAGES
statefulset.apps/redis   2/2     17m   redis        redis:6.2.8-alpine
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s#
root@ip-172-31-83-103:~/k8s# kubectl get pvc -n demo -o wide
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS        AGE     VOLUMEMODE
data-redis-0   Bound    pvc-72a5bef1-e851-47fb-9bc1-2fd2080cc11a   1Gi        RWO            storageclassredis   17m     Filesystem
data-redis-1   Bound    pvc-a2b301a8-33b8-4427-a6ea-1202bf1c9a4d   1Gi        RWO            storageclassredis   9m37s   Filesystem
root@ip-172-31-83-103:~/k8s#
```

Redis Server Details

```
root@ip-172-31-83-103:~/k8s# kubectl describe statefulset -n demo redis
Name:               redis
Namespace:          demo
CreationTimestamp:  Mon, 16 Jan 2023 10:48:55 +0000
Selector:           app=redis
Labels:             <none>
Annotations:        <none>
Replicas:           2 desired | 2 total
Update Strategy:    RollingUpdate
  Partition:        0
Pods Status:        2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=redis
  Containers:
   redis:
    Image:        redis:6.2.8-alpine
    Port:         6379/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/lib/redis from data (rw)
  Volumes:
   config:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      redis-ss-configuration
    Optional:  false
Volume Claims:
  Name:          data
  StorageClass:  storageclassredis
  Labels:        <none>
  Annotations:   <none>
  Capacity:      500Mi
  Access Modes:  [ReadWriteOnce]
Events:
  Type    Reason            Age                From                    Message
  ----    ------            ----               ----                    -------
  Normal  SuccessfulCreate  39m                statefulset-controller  create Claim data-redis-0 Pod redis-0 in StatefulSet redis success
  Normal  SuccessfulCreate  31m                statefulset-controller  create Claim data-redis-1 Pod redis-1 in StatefulSet redis success
  Normal  SuccessfulDelete  71s                statefulset-controller  delete Pod redis-1 in StatefulSet redis successful
  Normal  SuccessfulCreate  69s (x2 over 31m)  statefulset-controller  create Pod redis-1 in StatefulSet redis successful
  Normal  SuccessfulDelete  57s                statefulset-controller  delete Pod redis-0 in StatefulSet redis successful
  Normal  SuccessfulCreate  56s (x2 over 39m)  statefulset-controller  create Pod redis-0 in StatefulSet redis successful
root@ip-172-31-83-103:~/k8s#
```

Subtask 2:
Redis-cli

```
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP           NODE                           NOMINATED NODE   READINESS GATES
redis-0                 1/1     Running   0          41m   10.0.1.47    ip-10-0-1-222.ec2.internal   <none>           <none>
redis-1                 1/1     Running   0          41m   10.0.1.41    ip-10-0-1-222.ec2.internal   <none>           <none>
rediscli-fc7494575-bhdw9   1/1     Running   0          74s   10.0.1.252   ip-10-0-1-222.ec2.internal   <none>           <none>
root@ip-172-31-83-103:~/k8s# kubectl describe po -n demo rediscli-fc7494575-bhdw9
Name:             rediscli-fc7494575-bhdw9
Namespace:        demo
Priority:         0
Service Account:  default
Node:             ip-10-0-1-222.ec2.internal/10.0.1.222
Start Time:       Mon, 16 Jan 2023 12:07:24 +0000
Labels:           demo=demo
                  namespace=demo
                  pod-template-hash=fc7494575
Annotations:      kubernetes.io/psp: eks.privileged
Status:           Running
IP:               10.0.1.252
IPs:
  IP:           10.0.1.252
Controlled By:  ReplicaSet/rediscli-fc7494575
Containers:
  rediscli:
    Container ID:  containerd://439d57396bb582e8b0eacc4474a8741748a942678212617ef73801aa2c008784
    Image:         redis
    Image ID:      docker.io/library/redis@sha256:bb474c35022ca2c5618f4c49ca759bd2c0eea1daf5d934c560bd30092b97b498
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
    Args:
      -c
      sleep 100000
    State:          Running
      Started:      Mon, 16 Jan 2023 12:07:27 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-qd7jg (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-qd7jg:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  81s   default-scheduler  Successfully assigned demo/rediscli-fc7494575-bhdw9 to ip-10-0-1-222.ec2.internal
  Normal  Pulling    81s   kubelet            Pulling image "redis"
  Normal  Pulled     78s   kubelet            Successfully pulled image "redis" in 2.868254452s
  Normal  Created    78s   kubelet            Created container rediscli
  Normal  Started    78s   kubelet            Started container rediscli
root@ip-172-31-83-103:~/k8s#
```

Subtask 3:
Testing PVC using redis

Set Key Value Pairs

```
root@ip-172-31-83-103:~/k8s# kubectl exec -n demo -it rediscli-fc7494575-bhdw9 -- bash
root@rediscli-fc7494575-bhdw9:/data# redis-cli -h redis-service
redis-service:6379> SET foo 1
OK
redis-service:6379> GET foo
"1"
redis-service:6379> exit
root@rediscli-fc7494575-bhdw9:/data# exit
exit
root@ip-172-31-83-103:~/k8s#
```

Delete pods and Recreate

```
root@ip-172-31-83-103:~/k8s# kubectl exec -n demo -it rediscli-fc7494575-bhdw9 -- bash
root@rediscli-fc7494575-bhdw9:/data# redis-cli -h redis-service
redis-service:6379> SET foo 1
OK
redis-service:6379> GET foo
"1"
redis-service:6379> exit
root@rediscli-fc7494575-bhdw9:/data# exit
exit
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS    RESTARTS   AGE
redis-0                  1/1     Running   0          47m
redis-1                  1/1     Running   0          48m
rediscli-fc7494575-bhdw9   1/1     Running   0          7m43s
root@ip-172-31-83-103:~/k8s# kubectl delete po -n demo redis-0 redis-1
pod "redis-0" deleted
pod "redis-1" deleted
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS              RESTARTS   AGE
redis-0                  0/1     ContainerCreating   0          1s
rediscli-fc7494575-bhdw9   1/1     Running             0          8m7s
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS              RESTARTS   AGE
redis-0                  0/1     ContainerCreating   0          3s
rediscli-fc7494575-bhdw9   1/1     Running             0          8m9s
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS              RESTARTS   AGE
redis-0                  0/1     ContainerCreating   0          5s
rediscli-fc7494575-bhdw9   1/1     Running             0          8m11s
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS              RESTARTS   AGE
redis-0                  1/1     Running             0          20s
redis-1                  0/1     ContainerCreating   0          12s
rediscli-fc7494575-bhdw9   1/1     Running             0          8m26s
root@ip-172-31-83-103:~/k8s# kubectl get po -n demo
NAME                     READY   STATUS    RESTARTS   AGE
redis-0                  1/1     Running   0          91s
redis-1                  1/1     Running   0          83s
rediscli-fc7494575-bhdw9   1/1     Running   0          9m37s
```

Key Value Verify

```
root@ip-172-31-83-103:~/k8s# kubectl exec -n demo -it rediscli-fc7494575-bhdw9 -- bash
root@rediscli-fc7494575-bhdw9:/data# redis-cli -h redis-service
redis-service:6379> GET foo
"1"
redis-service:6379>
```

Task 4: Test auto scaling of the application.

Subtask 1:
Deploy an HPA

```
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests# kubectl get hpa -n demo
NAME         REFERENCE               TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
upg-loadme   Deployment/upg-loadme   0%/50%    1         5         1          27s
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests#
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests#
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests# kubectl describe hpa -n demo upg-loadme
Warning: autoscaling/v2beta2 HorizontalPodAutoscaler is deprecated in v1.23+, unavailable in v1.26+; use autoscaling/v2 HorizontalPodAutoscaler
Name:                                                  upg-loadme
Namespace:                                             demo
Labels:                                                <none>
Annotations:                                           <none>
CreationTimestamp:                                     Mon, 16 Jan 2023 12:39:13 +0000
Reference:                                             Deployment/upg-loadme
Metrics:                                               ( current / target )
  resource cpu on pods  (as a percentage of request):  0% (1m) / 50%
Min replicas:                                          1
Max replicas:                                          5
Deployment pods:                                       1 current / 1 desired
Conditions:
  Type            Status  Reason               Message
  ----            ------  ------               -------
  AbleToScale     True    ScaleDownStabilized  recent recommendations were higher than current one, applying the highest recent recommendation
  ScalingActive   True    ValidMetricFound     the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of request)
  ScalingLimited  False   DesiredWithinRange   the desired count is within the acceptable range
Events:           <none>
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests#
```

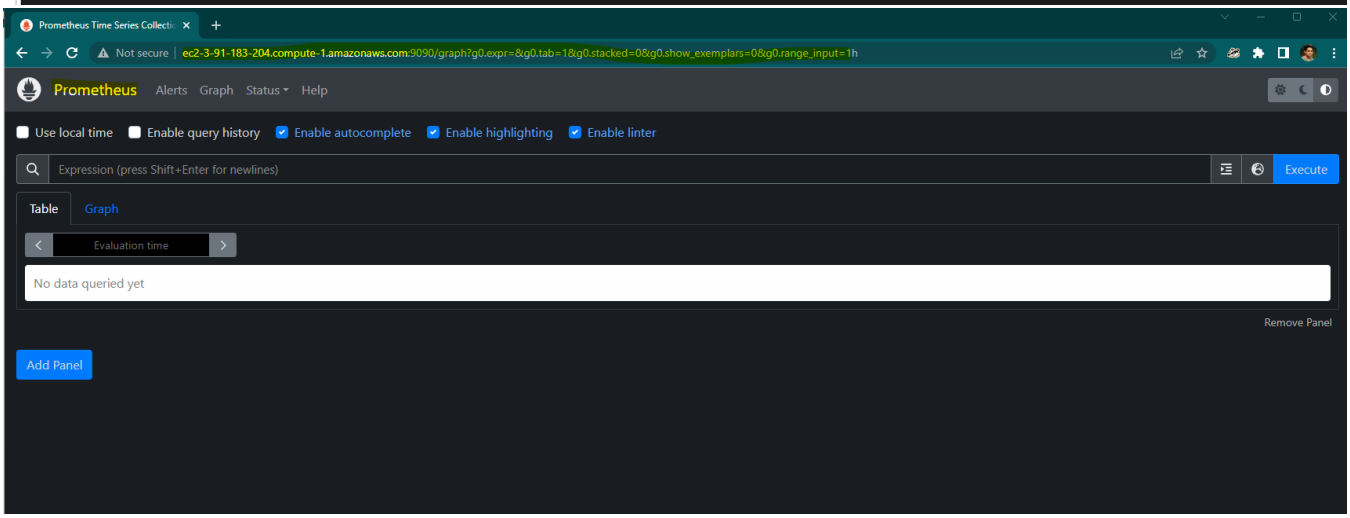
Subtask 2: Prometheus

Install

```
upg-loadme   Deployment/upg-loadme   0%/50%   1         5         1          27s
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests# helm install prometheus prometheus-community/prometheus -n demo --set prometheus-pushgateway.enabled=false --set prometheus-node
rter.enabled=false --set kube-state-metrics.enabled=false --set alertmanager.enabled=false
NAME: prometheus
LAST DEPLOYED: Mon Jan 16 12:55:43 2023
NAMESPACE: demo
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.demo.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace demo -l "app=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace demo port-forward $POD_NAME 9090




For more information on running Prometheus, visit:
https://prometheus.io/
root@ip-172-31-83-103:~/k8s/upg-loadme-manifests#
```

Port forward

```
root@ip-172-31-83-103:~/k8s# kubectl --namespace demo port-forward prometheus-server-f5ffc8766-4jzhk 9090 --address='0.0.0.0'
Forwarding from 0.0.0.0:9090 -> 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
```

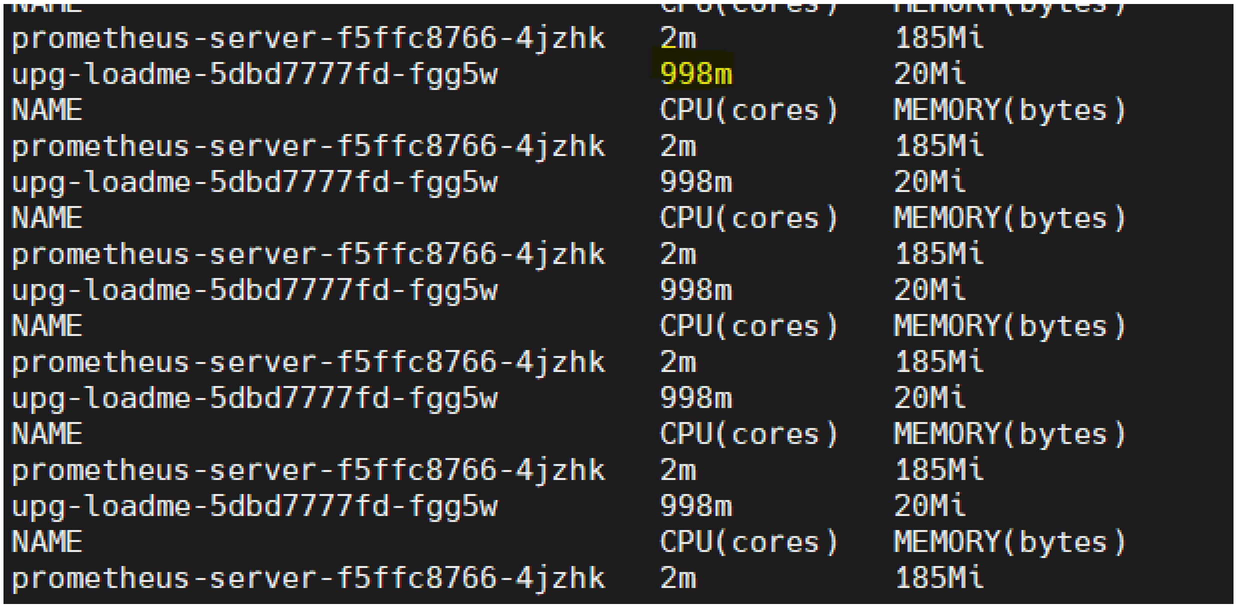


Subtask 3:
Load Generate and HPA testing

```
root@ip-172-31-83-103:~# ab -n300 -c50 'http://k8s-demo-upgloadm-c4597a7421-1106779957.us-east-1.elb.amazonaws.com/load?scale=500'
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking k8s-demo-upgloadm-c4597a7421-1106779957.us-east-1.elb.amazonaws.com (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Finished 300 requests


Server Software:
Server Hostname:        k8s-demo-upgloadm-c4597a7421-1106779957.us-east-1.elb.amazonaws.com
Server Port:            80

Document Path:          /load?scale=500
Document Length:        2 bytes

Concurrency Level:      50
Time taken for tests:   97.256 seconds
Complete requests:      300
Failed requests:        106
   (Connect: 0, Receive: 0, Length: 106, Exceptions: 0)
Non-2xx responses:      106
Total transferred:      67632 bytes
HTML transferred:       13320 bytes
Requests per second:    3.08 [#/sec] (mean)
Time per request:       16209.285 [ms] (mean)
Time per request:       324.186 [ms] (mean, across all concurrent requests)
Transfer rate:          0.68 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        1    2   1.9      1       8
Processing:   507 14799 8610.1  15540   29571
Waiting:      505 14799 8610.2  15540   29571
Total:        508 14801 8610.4  15542   29572

Percentage of the requests served within a certain time (ms)
  50%  15542
  66%  20049
  75%  22553
  80%  23068
  90%  27064
  95%  27570
  98%  28570
  99%  29570
 100%  29572 (longest request)
root@ip-172-31-83-103:~#
```

```
root@ip-172-31-83-103:~/k8s# kubectl get hpa -n demo --watch
NAME          REFERENCE                TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
upg-loadme    Deployment/upg-loadme    0%/50%     1         5         1          69m
upg-loadme    Deployment/upg-loadme    22%/50%    1         5         1          70m
upg-loadme    Deployment/upg-loadme    124%/50%   1         5         1          70m
upg-loadme    Deployment/upg-loadme    125%/50%   1         5         3          71m
upg-loadme    Deployment/upg-loadme    124%/50%   1         5         3          71m
upg-loadme    Deployment/upg-loadme    125%/50%   1         5         3          71m
upg-loadme    Deployment/upg-loadme    109%/50%   1         5         3          72m
upg-loadme    Deployment/upg-loadme    0%/50%     1         5         3          72m
root@ip-172-31-83-103:~/k8s#
```

```
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    5m           192Mi
upg-loadme-5dbd7777fd-fgg5w          1m           18Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    5m           192Mi
upg-loadme-5dbd7777fd-fgg5w          1m           18Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    5m           192Mi
upg-loadme-5dbd7777fd-fgg5w          1m           18Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    16m          185Mi
upg-loadme-5dbd7777fd-fgg5w          178m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk    2m           185Mi
upg-loadme-5dbd7777fd-fgg5w          998m         20Mi
NAME                                 CPU(cores)   MEMORY(bytes)
```



```
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
upg-loadme-5dbd7777fd-fgg5w         998m   20Mi
NAME                                CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
upg-loadme-5dbd7777fd-fgg5w         998m   20Mi
NAME                                CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
upg-loadme-5dbd7777fd-fgg5w         998m   20Mi
NAME                                CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
upg-loadme-5dbd7777fd-fgg5w         998m   20Mi
NAME                                CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
upg-loadme-5dbd7777fd-fgg5w         998m   20Mi
NAME                                CPU(cores)   MEMORY(bytes)
prometheus-server-f5ffc8766-4jzhk   2m     185Mi
```