PIN

MundosE-PIN 2301

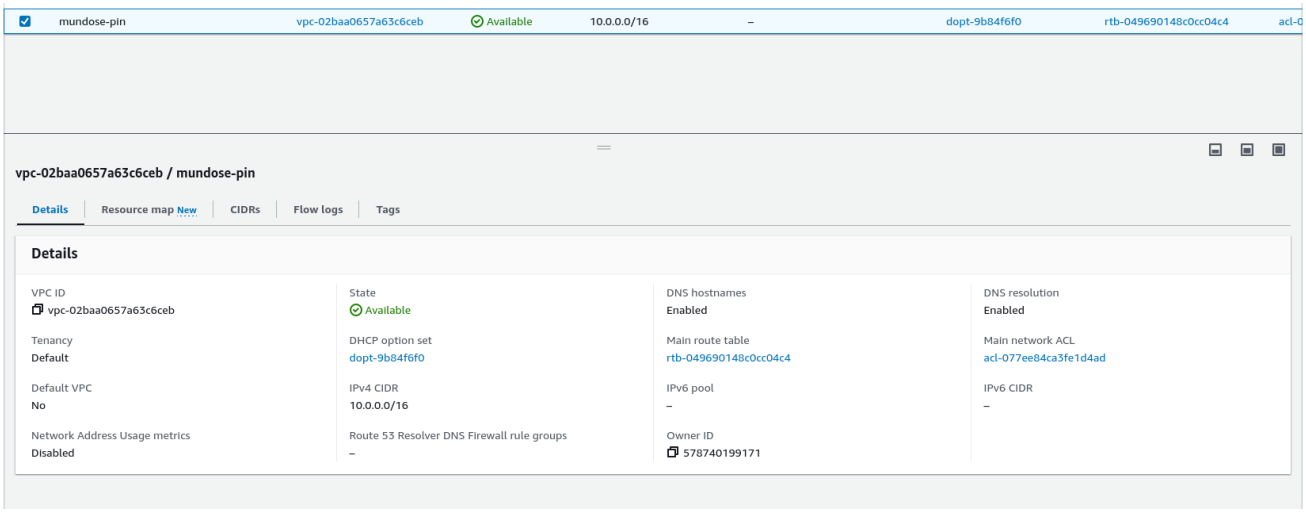
(This example show the steps to expose a simple app using eks on aws.)

Creating the RDS PostgreSQL + ECR repository

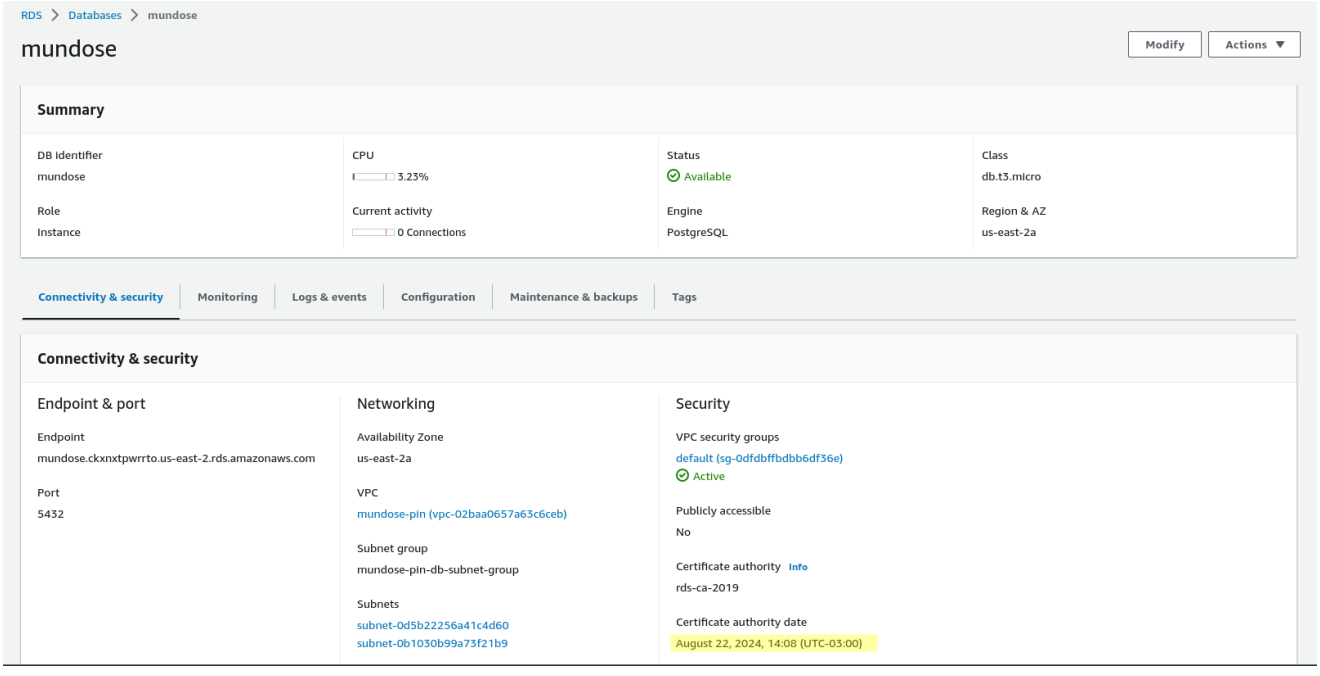
Terraform create vpc + rds postgresql db + ecr repo

```
make rds-ecr-create
```

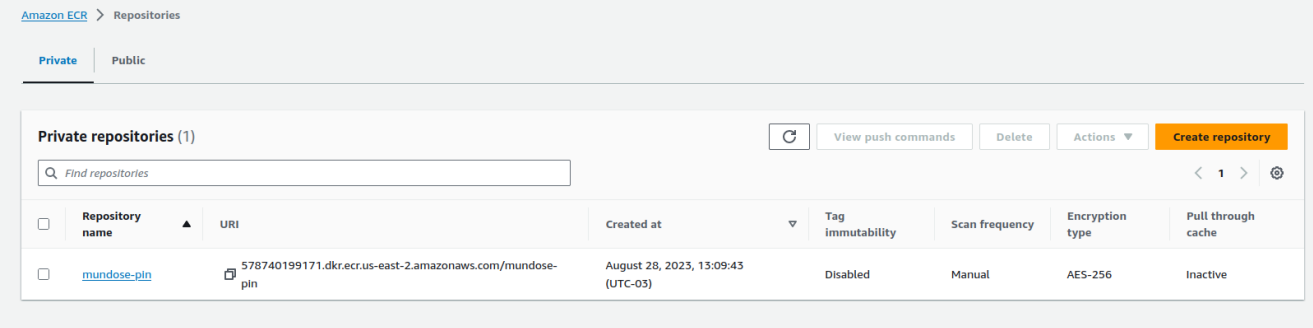
VPC



RDS



ECR

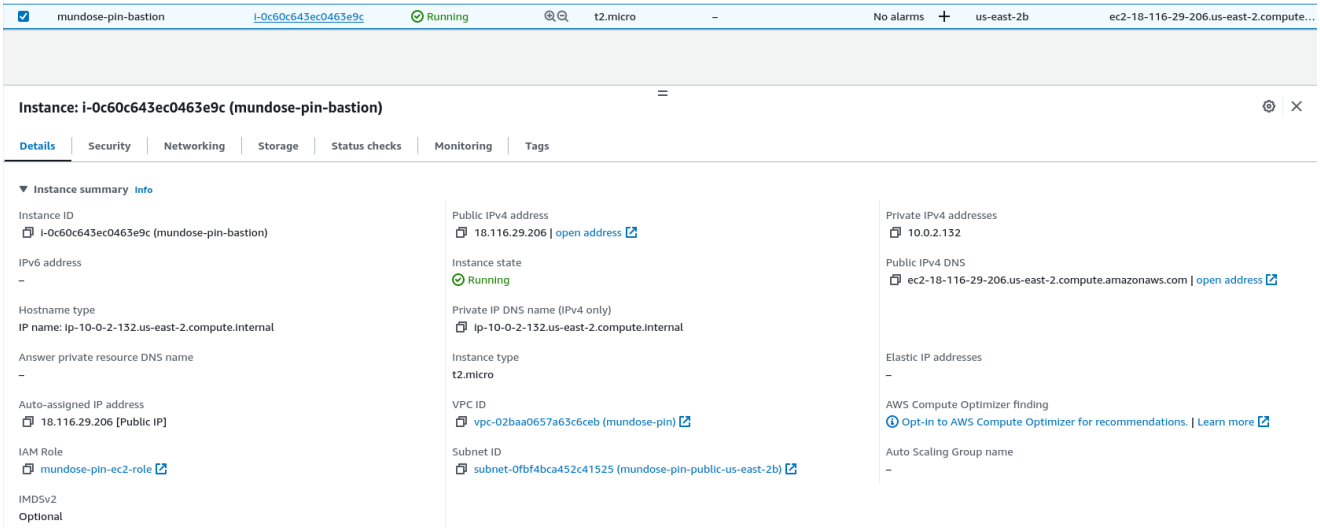


Output

```
Apply complete! Resources: 30 added, 0 changed, 0 destroyed.

Outputs:

account_id = "578740199171"
db_address = "mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com"
ecr_repository_url = "578740199171.dkr.ecr.us-east-2.amazonaws.com/mundose-pin"
project_name = "mundose-pin"
region = "us-east-2"
END  2023-28-08 13:13:56
DURATION  269 seconds
```

Terraform create ec2 bastion for ssh tunnel

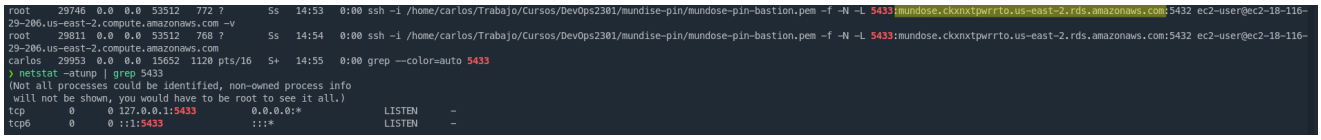
```
make bastion-ssh-create
```



Outputs

```
Outputs:

aws_default_security_group_id = "sg-0dfdbffbdbb6df36e"
aws_subnets_private_ids = tolist([
  "subnet-0b1030b99a73f21b9",
  "subnet-0d5b22256a41c4d60",
])
aws_subnets_public_ids = tolist([
  "subnet-0fbf4bca452c41525",
  "subnet-07166368315c5c7f6",
])
aws_vpc_id = "vpc-02baa0657a63c6ceb"
bastion_public_dns = "ec2-18-116-29-206.us-east-2.compute.amazonaws.com"
bastion_security_group_id = "sg-01c8e4d10039a73ce"
END  2023-28-08 14:29:51
DURATION  57 seconds
```

Create ssh tunnel to RDS




```
[root@ip-10-0-2-132 ec2-user]# psql --version
psql (PostgreSQL) 9.2.24
```

Update psql client on bastion ec2 to avoid security problems

```
[root@ip-10-0-2-132 ec2-user]# sudo yum install -y amazon-linux-extras
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package amazon-linux-extras-2.0.1-1.amzn2.noarch already installed and latest version
Nothing to do
[root@ip-10-0-2-132 ec2-user]# sudo amazon-linux-extras install postgresql10
Installing postgresql
```

```
[root@ip-10-0-2-132 ec2-user]# psql --version
psql (PostgreSQL) 10.21
[root@ip-10-0-2-132 ec2-user]# psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose
Password for user master:
psql (10.21, server 15.3)
WARNING: psql major version 10, server major version 15.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

mundose=>
```

Create data base structure

```
[root@ip-10-0-2-132 ec2-user]# vi create.sql
[root@ip-10-0-2-132 ec2-user]# cat create.sql
drop table if exists vote;

--
-- Create table `vote`
--
CREATE TABLE vote (
  name varchar(255),
  value integer
);

--
-- Insert values into `vote`
--
INSERT INTO vote VALUES
('up', '0'),
('down', '0');
```

from local

psql "postgresql://master:masterpass@127.0.0.1:5433/mundose?sslmode=require" < sql/create.sql

```
❯ psql "postgresql://master:masterpass@127.0.0.1:5433/mundose?sslmode=require"
psql (10.23 (Ubuntu 10.23-0ubuntu0.18.04.2), server 15.3)
WARNING: psql major version 10, server major version 15.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

mundose=>
```

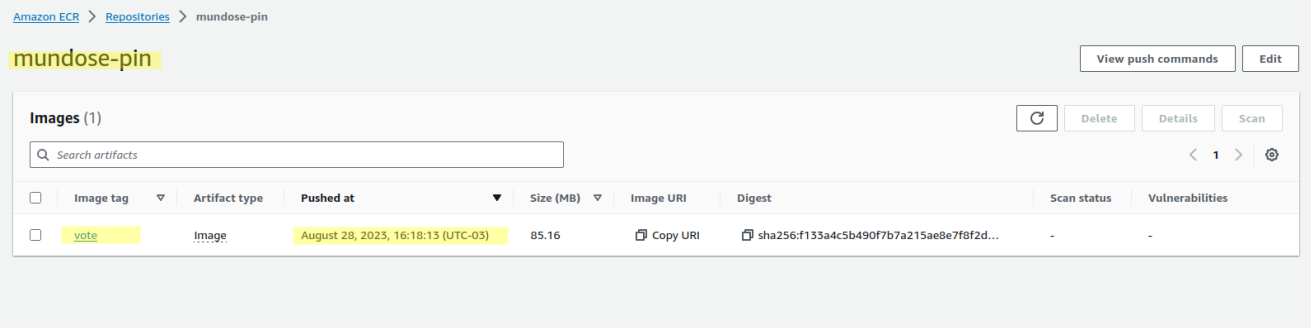
from EC2 Bastion

```
[root@ip-10-0-2-132 ec2-user]# psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose < create.sql
Password for user master:
NOTICE:  table "vote" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 2
[root@ip-10-0-2-132 ec2-user]#
```

Push vote app image to ECR

```
make ecr-push
```

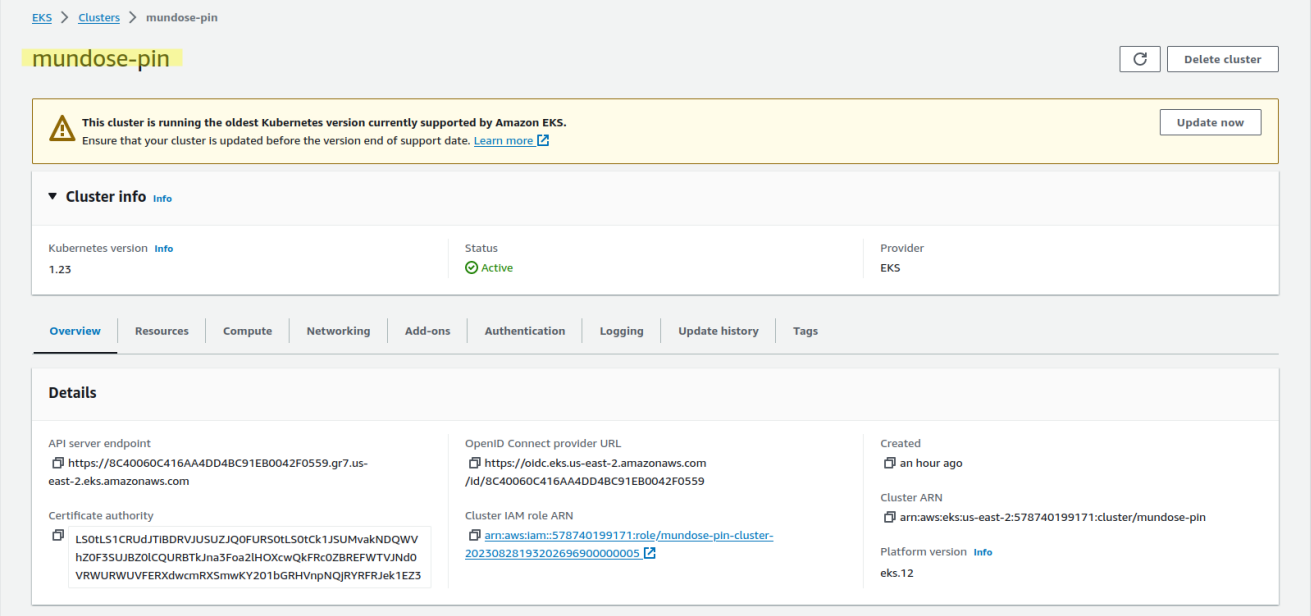
```
Step 21/21 : CMD ["node", "index.js"]
 ---> Running in ccbfd889534c
Removing intermediate container ccbfd889534c
 ---> 800e1f224eac
Successfully built 800e1f224eac
Successfully tagged vote:latest
 AWS_ACCOUNT_ID  578740199171
 REPOSITORY_URL  578740199171.dkr.ecr.us-east-2.amazonaws.com/mundose-pin
WARNING! Your password will be stored unencrypted in /home/carlos/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
The push refers to repository [578740199171.dkr.ecr.us-east-2.amazonaws.com/mundose-pin]
bfd251d74980: Pushed
aa7922b82478: Pushed
a39d67fb040d: Pushed
efacba6c0e9c: Pushed
77b897ea207d: Pushed
ea1c1a315bbc: Pushed
63cb23df0560: Pushed
08249ce7456a: Pushed
vote: digest: sha256:f133a4c5b490f7b7a215ae8e7f8f2d3a0b9a27c7a16d758f4052efff3906a0b5 size: 1995
```



Terraform create EKS cluster

```
make eks-create
```

```
eks_cluster_endpoint = "https://8C40060C416AA4DD4BC91EB0042F0559.gr7.us-east-2.eks.amazonaws.com"
eks_cluster_id = "mundose-pin"
eks_nodegroup_rolearn = "arn:aws:iam::578740199171:role/green-eks-node-group-20230828193202693900000004"
project_name = "mundose-pin"
region = "us-east-2"
vpc_id = "vpc-02baa0657a63c6ceb"
vpc_private_subnets = tolist([
  "subnet-0b1030b99a73f21b9",
  "subnet-0d5b22256a41c4d60",
])
 END  2023-28-08 16:41:31
 DURATION  590 seconds
```

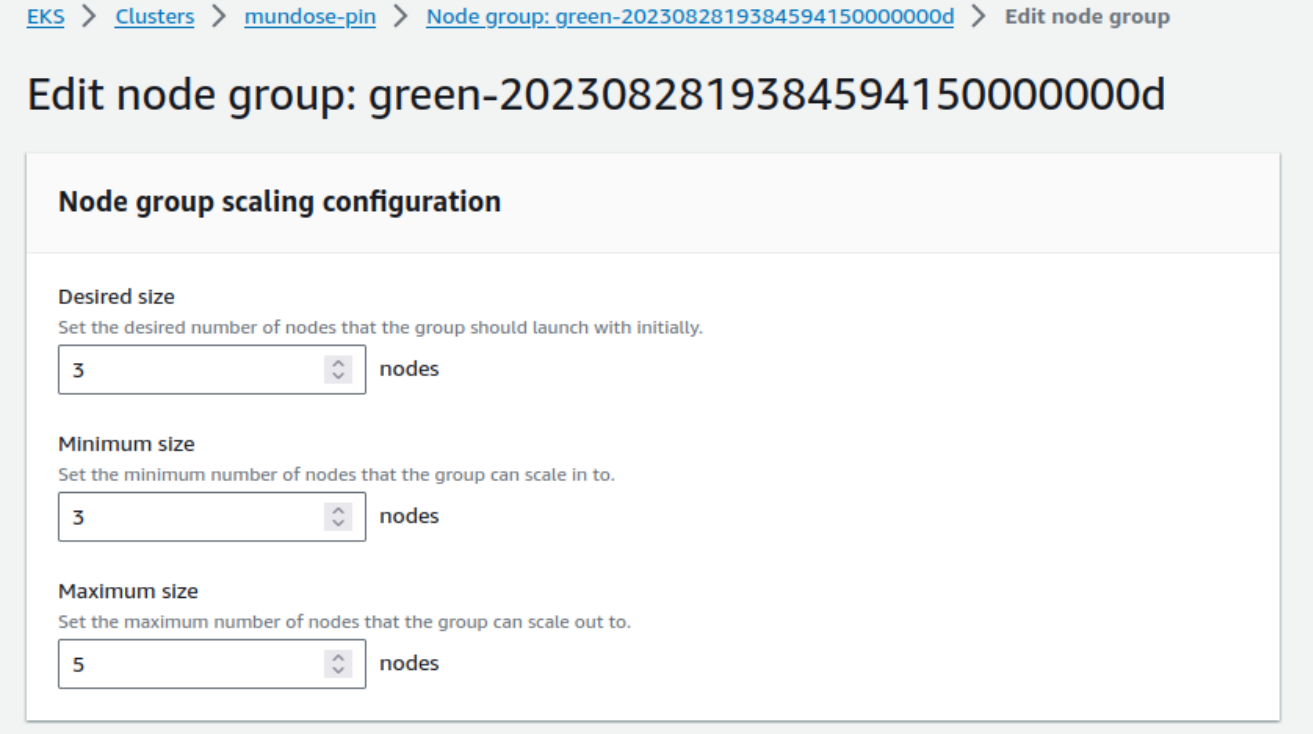


I can add more nodes to the cluster changing on this script

```
eks_managed_node_groups = {

  green = {
    min_size     = 1
    max_size     = 2
    desired_size = 1

    instance_types = ["t2.medium"]
    capacity_type  = "ON_DEMAND" # SPOT
  }
}
```

```
eks_managed_node_groups = {

        green = {
                min_size = 3
                max_size = 5
                desired_size = 3

                instance_types = ["t2.medium"]
                capacity_type = "ON_DEMAND" # SPOT
        }
}
```

or editing the node group



Test connection between the cluster and the RDS database

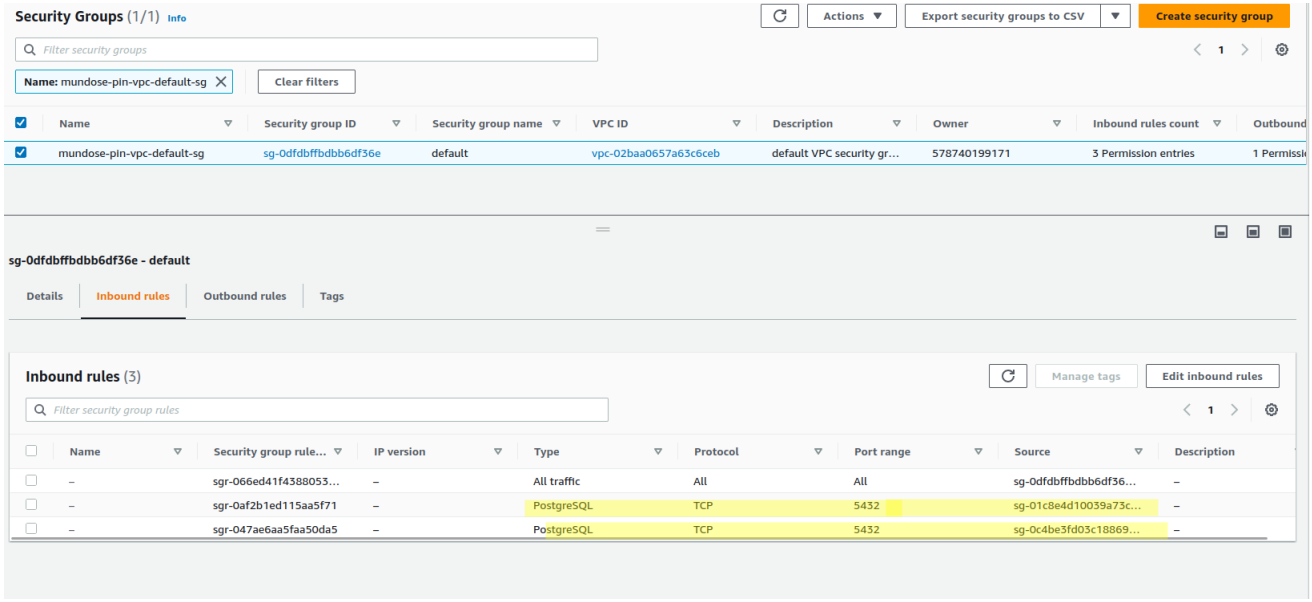
```
kubectl create ns vote
kubectl run alpine --image=alpine -i --tty -n vote
```

the connection fails

```
❯ kubectl create ns vote
namespace/vote created
❯ kubectl run alpine --image=alpine -i --tty -n vote
If you don't see a command prompt, try pressing enter.
/ # apk --update add postgresql-client
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_64/APKINDEX.tar.gz
(1/8) Installing postgresql-common (1.2-r0)
Executing postgresql-common-1.2-r0.pre-install
(2/8) Installing lz4-libs (1.9.4-r4)
(3/8) Installing libpq (15.4-r0)
(4/8) Installing ncurses-terminfo-base (6.4_p20230506-r0)
(5/8) Installing libncursesw (6.4_p20230506-r0)
(6/8) Installing readline (8.2.1-r1)
(7/8) Installing zstd-libs (1.5.5-r4)
(8/8) Installing postgresql15-client (15.4-r0)
Executing busybox-1.36.1-r2.trigger
Executing postgresql-common-1.2-r0.trigger
* Setting postgresql15 as the default version
OK: 12 MiB in 23 packages
/ # psql "postgresql://master:masterpass@mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com:5432/mundose?sslmode=require"

^C
/ # psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose
^C
/ # ping mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com
PING mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com (10.0.3.220): 56 data bytes
^C
--- mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
/ # psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose
^C
/ #
```

Add inbound rule

```
resource "aws_security_group_rule" "postgresql_ec2_instances_sg" {
  # this rule is added to the security group defined by `security_group_id`
  # and this id target the `default` security group associated with the created VPC
  security_group_id = data.aws_security_group.default_security_group.id

  type      = "ingress"
  protocol  = "tcp"
  from_port = 5432
  to_port   = 5432

  # One of ['cidr_blocks', 'ipv6_cidr_blocks', 'self', 'source_security_group_id', 'prefix_list_ids']
  # must be set to create an AWS Security Group Rule
  source_security_group_id = module.eks.eks_managed_node_groups.green.security_group_id

  lifecycle {
    create_before_destroy = true
  }
}
```




```
/ # psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose
^C
/ # psql -h mundose.ckxnxtpwrrto.us-east-2.rds.amazonaws.com -U master -d mundose
Password for user master:
psql (15.4, server 15.3)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, compression: off)
Type "help" for help.

mundose=>
```

Install vote app

```
make kubectl-vote
```

Get LoadBalancer

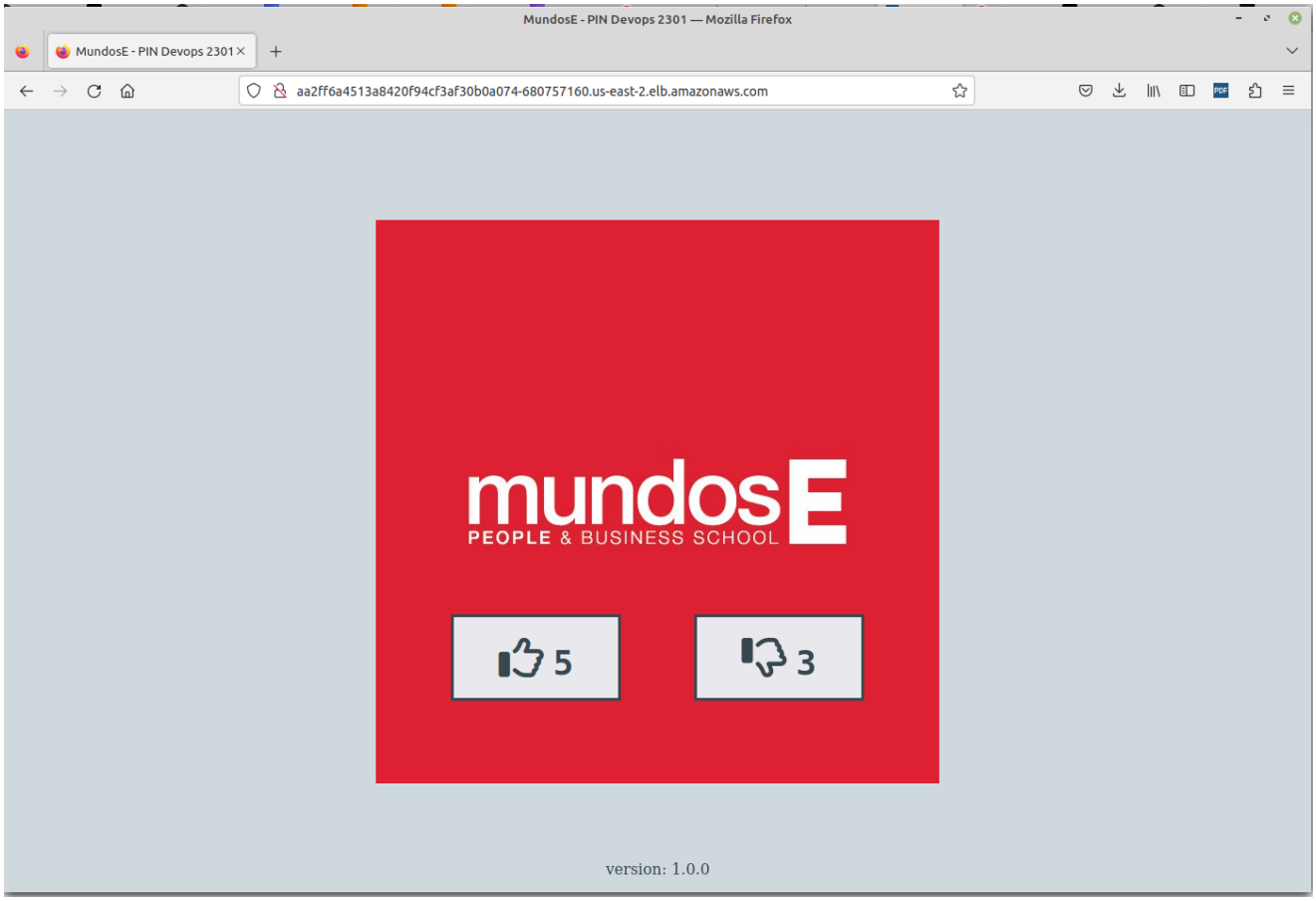
Option1

```
❯ k get all -n vote
NAME                        READY   STATUS    RESTARTS        AGE
pod/alpine                  1/1     Running   1 (3h21m ago)   3h49m
pod/vote-67c9847fcf-ldxxb   1/1     Running   0               3h15m

NAME           TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)        AGE
service/vote   LoadBalancer   172.20.11.120   aa2ff6a4513a8420f94cf3af30b0a074-680757160.us-east-2.elb.amazonaws.com   80:32303/TCP   3h16m

NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/vote   1/1     1            1           3h16m

NAME                              DESIRED   CURRENT   READY   AGE
replicaset.apps/vote-67c9847fcf   1         1         1       3h16m
```

Option2

```
❯ make load-balancer
/home/carlos/Trabajo/Cursos/DevOps2301/mundise-pin
EXECUTE load-balancer
LOAD_BALANCER aa2ff6a4513a8420f94cf3af30b0a074-680757160.us-east-2.elb.amazonaws.com
```

Browse the app



Install Prometheus

```
make kubectl-prometheus
```

```
❯ make kubectl-prometheus
/home/carlos/Trabajo/Cursos/DevOps2301/mundise-pin
 EXECUTE  kubectl-prometheus
anamespace/prometheus created
"prometheus-community" has been added to your repositories
Release "prometheus" does not exist. Installing it now.
NAME: prometheus
LAST DEPLOYED: Mon Aug 28 23:23:18 2023
NAMESPACE: prometheus
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.prometheus.svc.cluster.local


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


The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.prometheus.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been disabled by default since    #####
######            it deprecated after k8s 1.25+. use                        #####
######            (index .Values "prometheus-node-exporter" "rbac"          #####
###### .          "pspEnabled") with (index .Values                         #####
######            "prometheus-node-exporter" "rbac" "pspAnnotations")       #####
######            in case you still need it.                                #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.prometheus.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
```

```
❯ k get all -n prometheus
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0                        0/1     Pending   0          5m58s
pod/prometheus-kube-state-metrics-d96b9df7-c9k4c     1/1     Running   0          5m58s
pod/prometheus-prometheus-node-exporter-b7lwx        1/1     Running   0          5m58s
pod/prometheus-prometheus-pushgateway-799f46dfd8-t5lfq   1/1     Running   0          5m58s
pod/prometheus-server-6c9c9b7b58-hwbxr               0/2     Pending   0          5m58s

NAME                                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
service/prometheus-alertmanager               ClusterIP   172.20.141.180   <none>        9093/TCP   6m
service/prometheus-alertmanager-headless      ClusterIP   None             <none>        9093/TCP   6m
service/prometheus-kube-state-metrics         ClusterIP   172.20.132.55    <none>        8080/TCP   6m
service/prometheus-prometheus-node-exporter   ClusterIP   172.20.205.99    <none>        9100/TCP   6m
service/prometheus-prometheus-pushgateway     ClusterIP   172.20.8.157     <none>        9091/TCP   6m
service/prometheus-server                     ClusterIP   172.20.101.88    <none>        80/TCP     6m

NAME                                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/prometheus-prometheus-node-exporter   1         1         1       1            1           kubernetes.io/os=linux   6m

NAME                                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics       1/1     1            1           6m1s
deployment.apps/prometheus-prometheus-pushgateway   1/1     1            1           6m1s
deployment.apps/prometheus-server                   0/1     1            0           6m1s

NAME                                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-d96b9df7         1         1         1       6m1s
replicaset.apps/prometheus-prometheus-pushgateway-799f46dfd8   1         1         1       6m1s
replicaset.apps/prometheus-server-6c9c9b7b58                   1         1         0       6m1s

NAME                                          READY   AGE
statefulset.apps/prometheus-alertmanager      0/1     6m2s
```

Install ebs csi driver

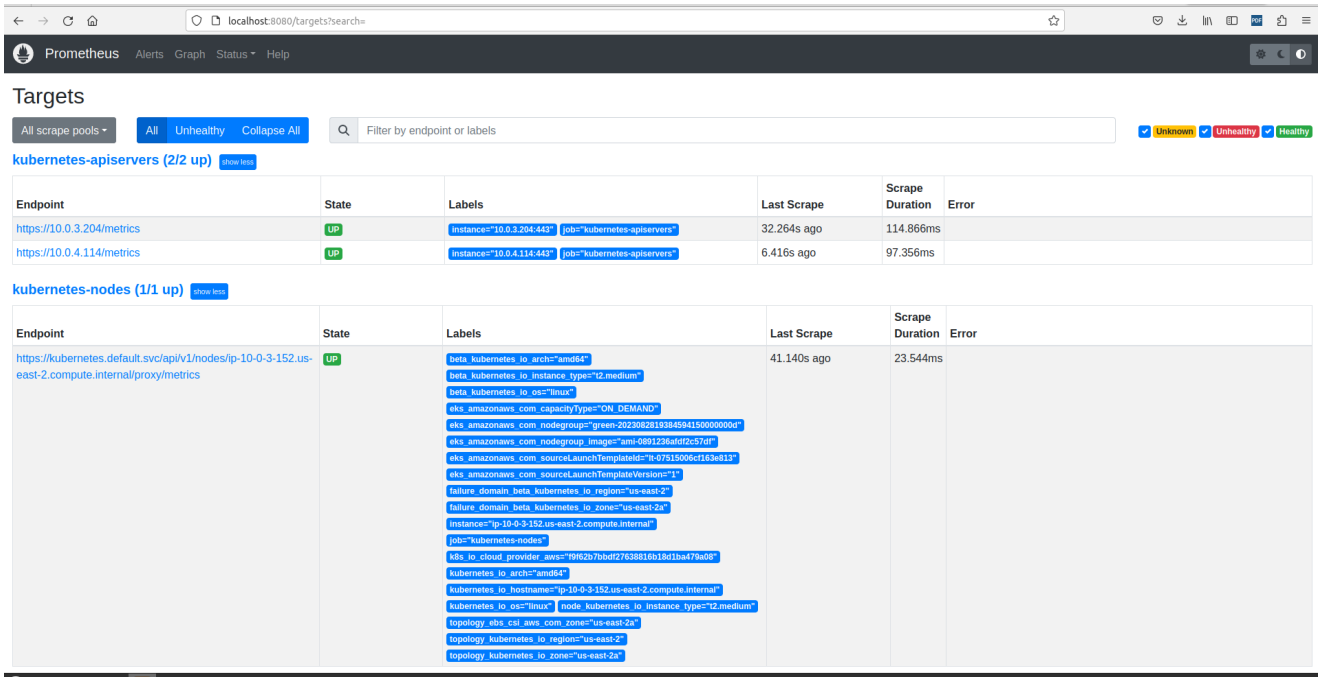
```
make kubectl-ebs-csi-driver
```

```
❯ make kubectl-ebs-csi-driver
/home/carlos/Trabajo/Cursos/DevOps2301/mundise-pin
 EXECUTE  kubectl-ebs-csi-driver
 AWS_ACCOUNT_ID  578740199171
2023-08-28 23:39:33 [ℹ]  1 iamserviceaccount (kube-system/ebs-csi-controller-sa) was included (based on the include/exclude rules)
2023-08-28 23:39:33 [!]  serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2023-08-28 23:39:33 [ℹ]  1 task: { create IAM role for serviceaccount "kube-system/ebs-csi-controller-sa" }
2023-08-28 23:39:33 [ℹ]  building iamserviceaccount stack "eksctl-mundose-pin-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-08-28 23:39:33 [ℹ]  deploying stack "eksctl-mundose-pin-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-08-28 23:39:34 [ℹ]  waiting for CloudFormation stack "eksctl-mundose-pin-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-08-28 23:40:04 [ℹ]  waiting for CloudFormation stack "eksctl-mundose-pin-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-08-28 23:40:07 [ℹ]  Kubernetes version "1.23" in use by cluster "mundose-pin"
2023-08-28 23:40:07 [ℹ]  using provided ServiceAccountRoleARN "arn:aws:iam::578740199171:role/AmazonEKS_EBS_CSI_DriverRole"
2023-08-28 23:40:07 [ℹ]  creating addon
2023-08-28 23:41:03 [ℹ]  addon "aws-ebs-csi-driver" active
```

```
❯ k get all -n prometheus
NAME                                                READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0                       1/1     Running   0          36m
pod/prometheus-kube-state-metrics-d96b9df7-c9k4c    1/1     Running   0          36m
pod/prometheus-prometheus-node-exporter-b7lwx       1/1     Running   0          36m
pod/prometheus-prometheus-pushgateway-799f46dfd8-t5lfq   1/1     Running   0          36m
pod/prometheus-server-6c9c9b7b58-hwbxr              2/2     Running   0          36m

NAME                                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
service/prometheus-alertmanager               ClusterIP   172.20.141.180   <none>        9093/TCP   36m
service/prometheus-alertmanager-headless      ClusterIP   None             <none>        9093/TCP   36m
service/prometheus-kube-state-metrics         ClusterIP   172.20.132.55    <none>        8080/TCP   36m
service/prometheus-prometheus-node-exporter   ClusterIP   172.20.205.99    <none>        9100/TCP   36m
service/prometheus-prometheus-pushgateway     ClusterIP   172.20.8.157     <none>        9091/TCP   36m
service/prometheus-server                     ClusterIP   172.20.101.88    <none>        80/TCP     36m

NAME                                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/prometheus-prometheus-node-exporter   1         1         1       1            1           kubernetes.io/os=linux   36m

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics       1/1     1            1           36m
deployment.apps/prometheus-prometheus-pushgateway   1/1     1            1           36m
deployment.apps/prometheus-server                   1/1     1            1           36m

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-d96b9df7         1         1         1       36m
replicaset.apps/prometheus-prometheus-pushgateway-799f46dfd8   1         1         1       36m
replicaset.apps/prometheus-server-6c9c9b7b58                   1         1         1       36m

NAME                                       READY   AGE
statefulset.apps/prometheus-alertmanager   1/1     36m
```

Browse the console

```
❯ kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090 --address 0.0.0.0
Forwarding from 0.0.0.0:8080 -> 9090
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```

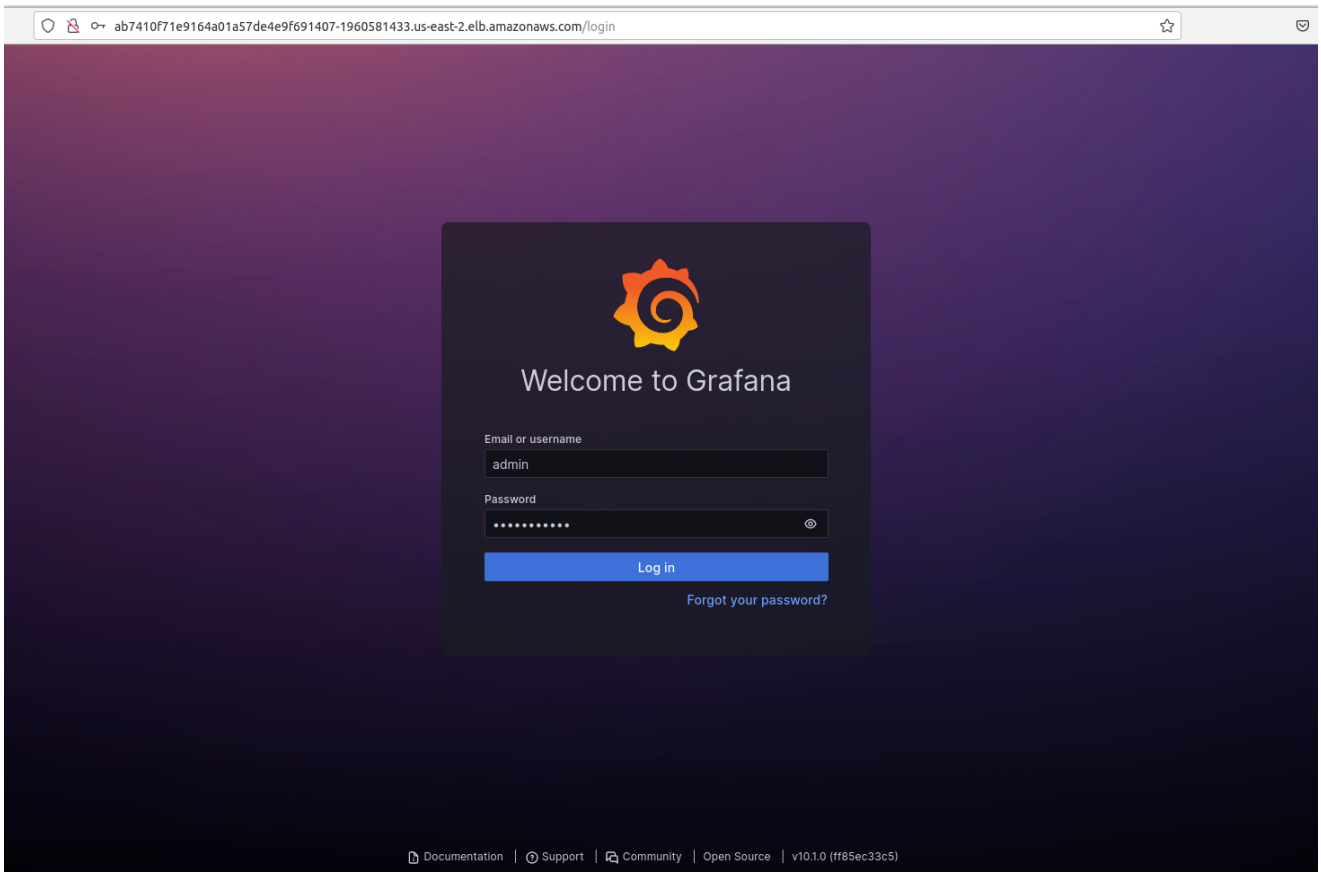


Install grafana

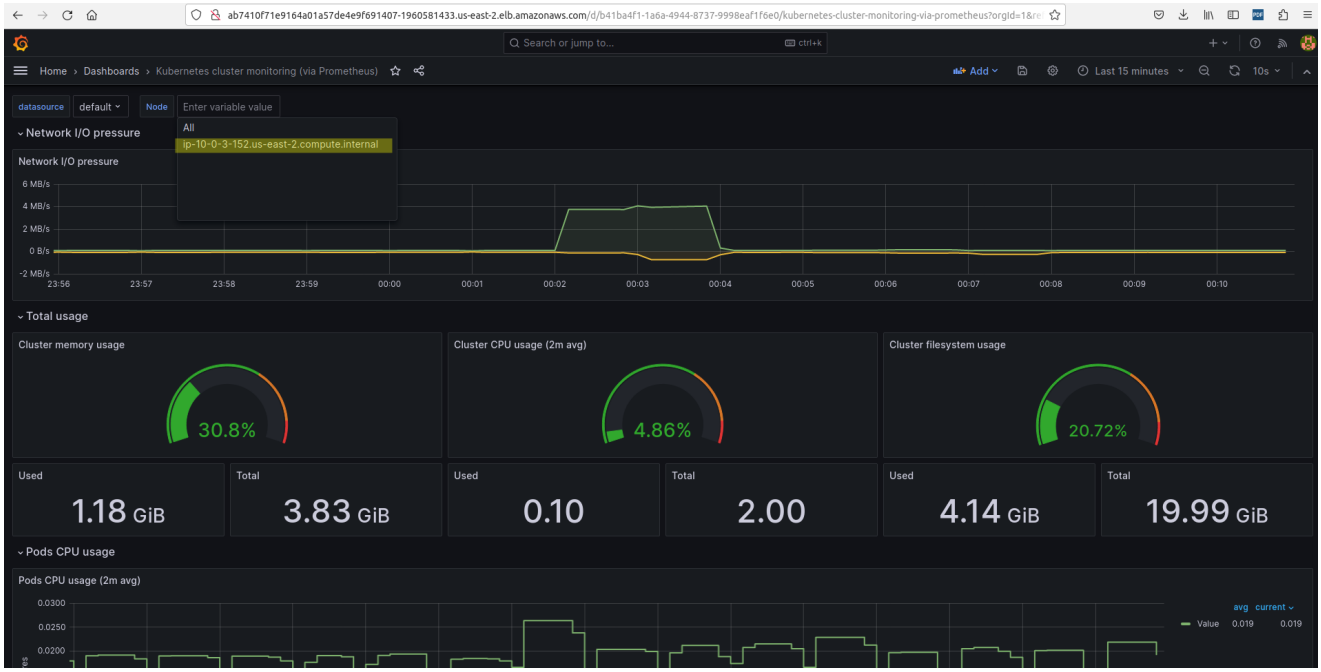
```
make kubectl-grafana
```

```
❯ make kubectl-grafana
/home/carlos/Trabajo/Cursos/DevOps2301/mundise-pin
 EXECUTE  kubectl-grafana
namespace/grafana created
"grafana" has been added to your repositories
NAME: grafana
LAST DEPLOYED: Tue Aug 29 00:01:29 2023
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

   kubectl get secret --namespace grafana grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo


2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

   grafana.grafana.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
   NOTE: It may take a few minutes for the LoadBalancer IP to be available.
        You can watch the status of by running 'kubectl get svc --namespace grafana -w grafana'
     export SERVICE_IP=$(kubectl get svc --namespace grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
     http://$SERVICE_IP:80

3. Login with the password from step 1 and the username: admin
```

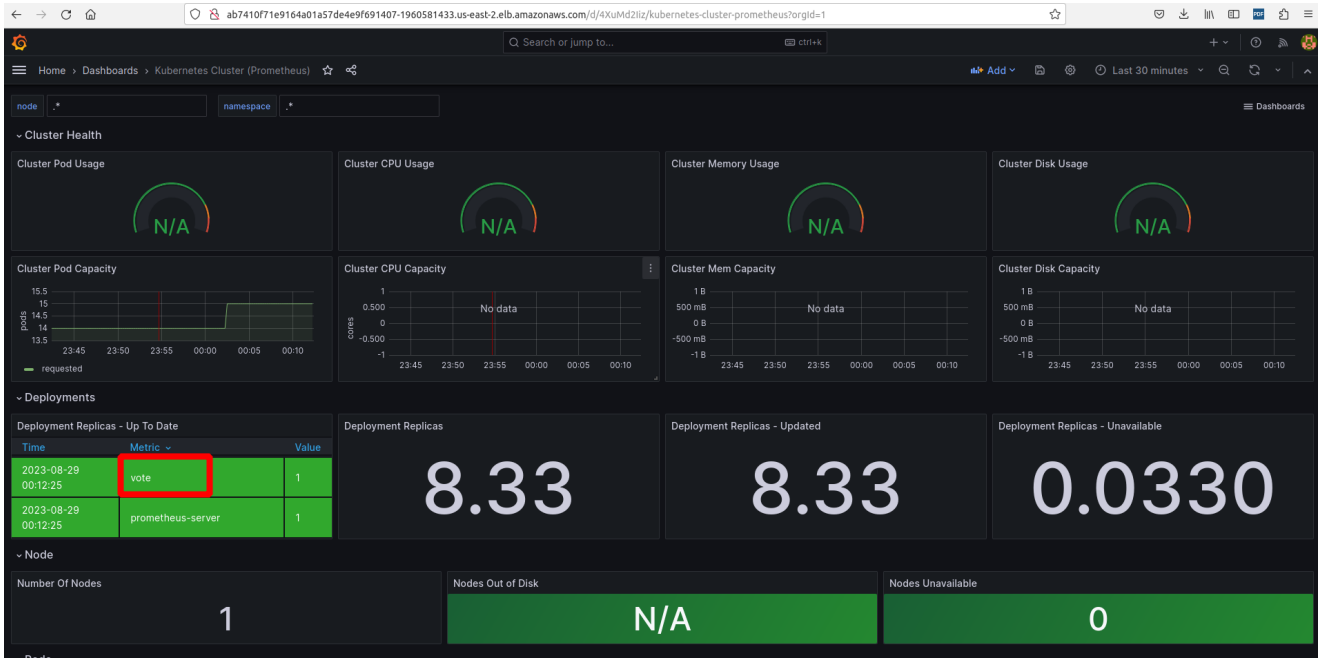
```
❯ kubectl get all -n grafana
NAME                           READY   STATUS    RESTARTS   AGE
pod/grafana-54d8b6bbb-4xvjc    0/1     Running   0          43s

NAME              TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)        AGE
service/grafana   LoadBalancer   172.20.198.113   ab7410f71e9164a01a57de4e9f691407-1960581433.us-east-2.elb.amazonaws.com   80:31741/TCP   45s

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana   0/1     1            0           46s

NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-54d8b6bbb    1         1         0       46s
```



Import 3119



Import 6417



The End