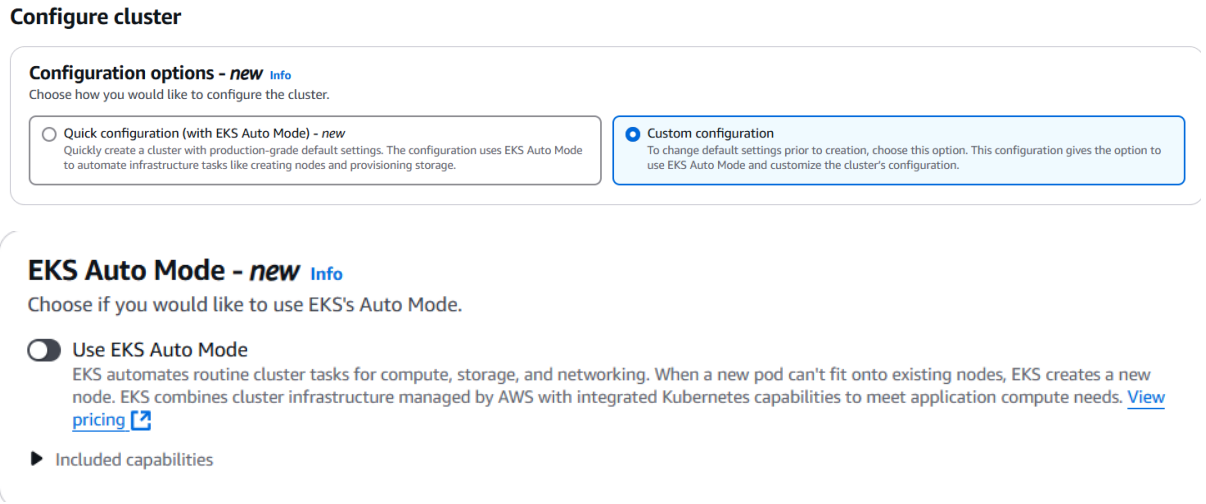
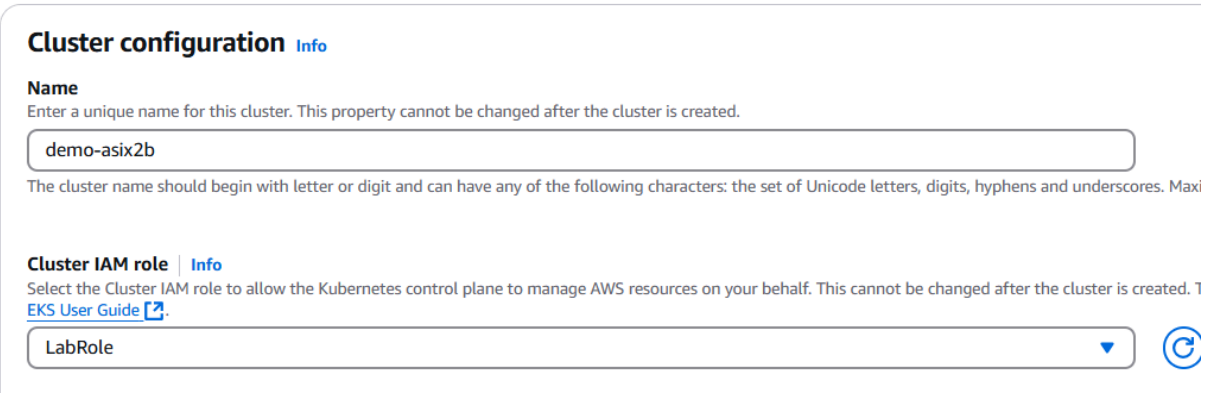
Kubernes inicio

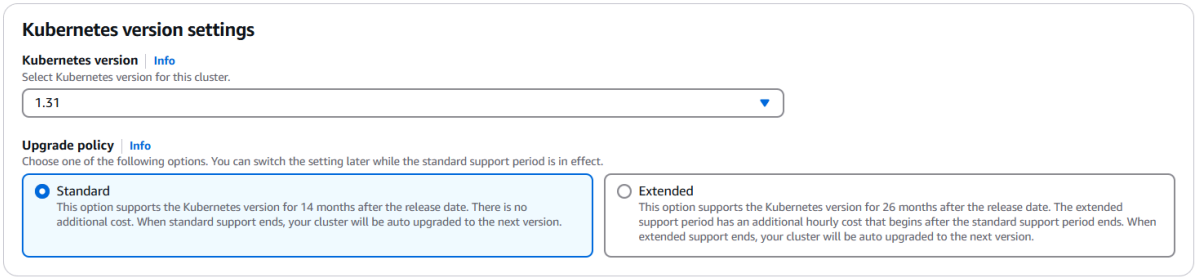
Kubernetes es un lenguaje declarativo (como Terraform), por tanto espera que le digamos cómo queremos que sea la infraestructura que queremos crear. Por tanto, el Control Planer es el elemento que se encarga de comprobar periódicamente que lo que quieres desplegar funciona bien y si no, pues pone en marcha de nuevo los servicios que no estén funcionando según lo esperado en la declaración

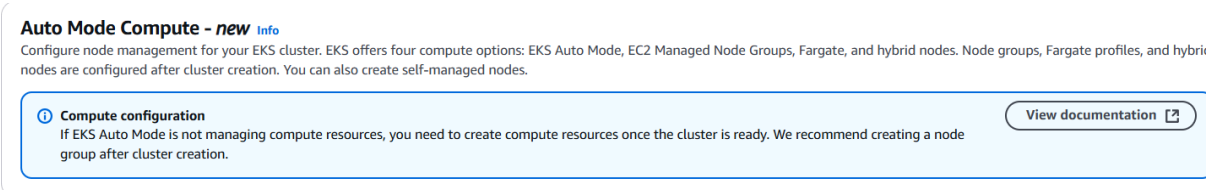
Vamos a crear un Cluster en AWS. Vamos a AWS > EKS ⭐ > Create Cluster.
MUY IMPORTANTE desactivar el EKS Auto Mode, porque si está habilitado nos dará muchos problemas de permisos.



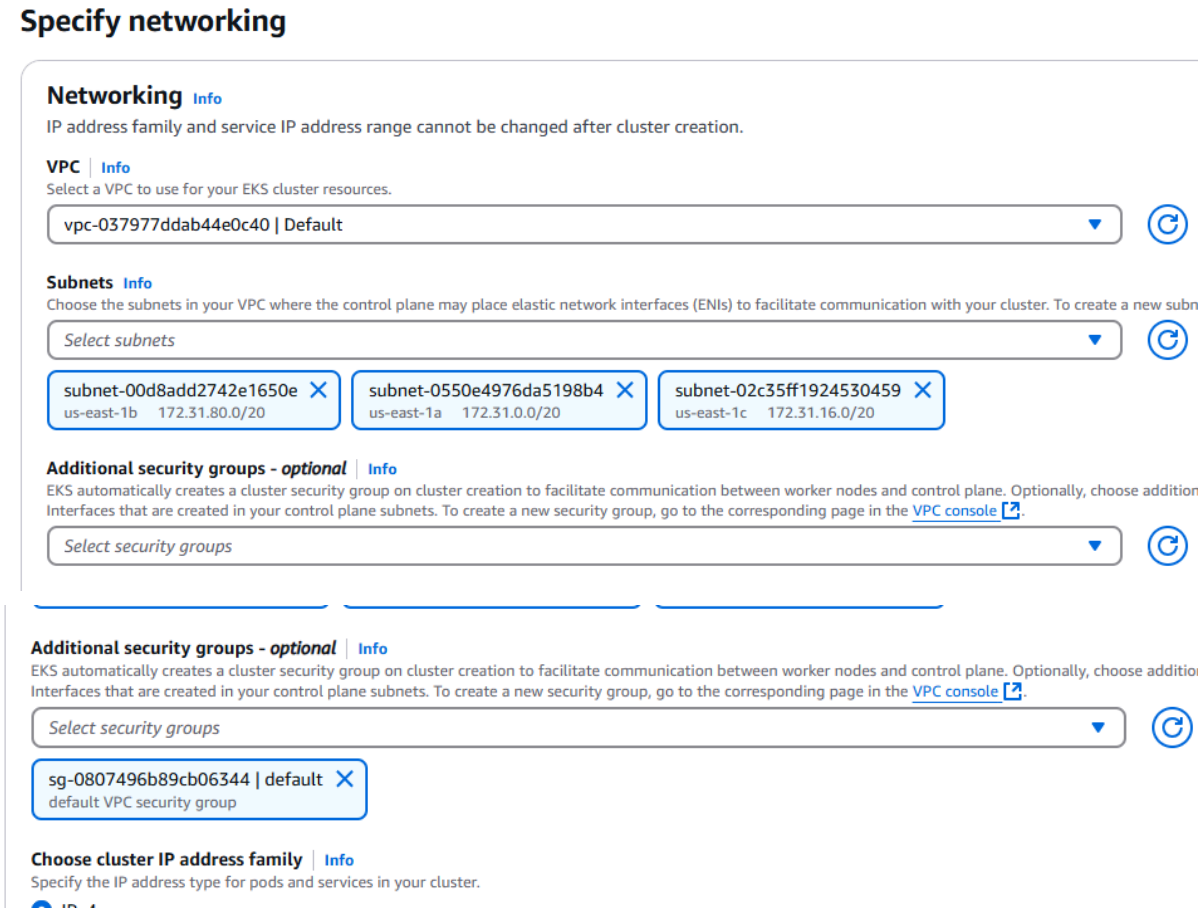
Como hicimos con Form, asignamos Role a LabRole que tiene los permisos más extensos para que no tengamos problemas después.





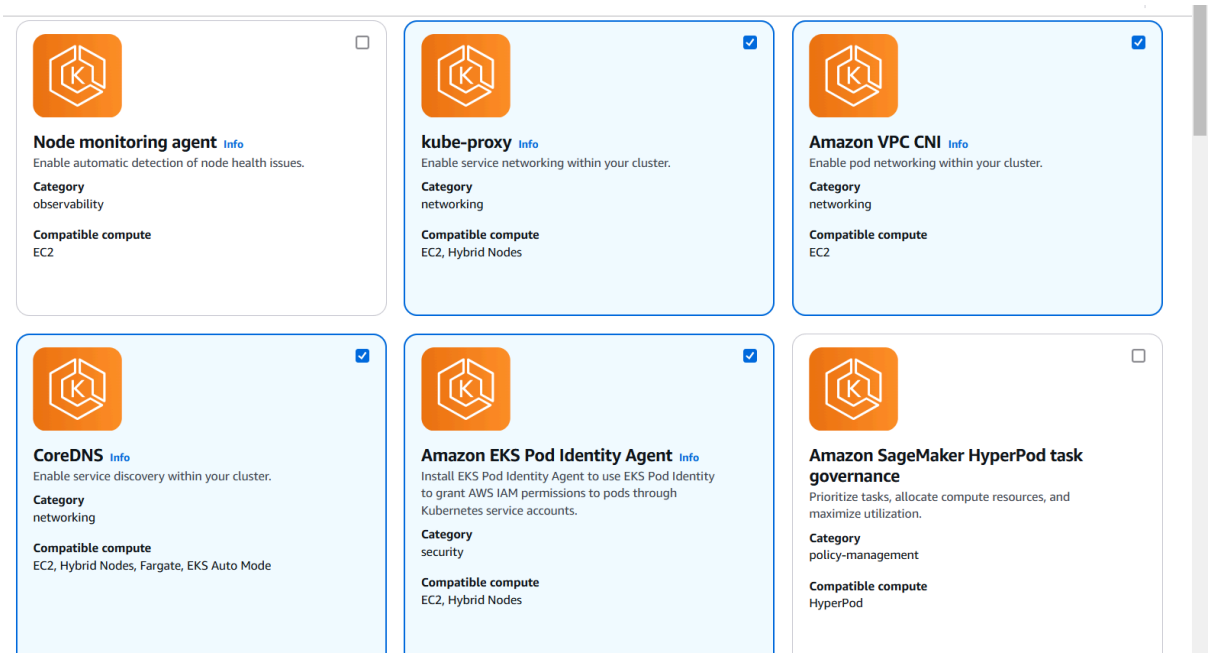


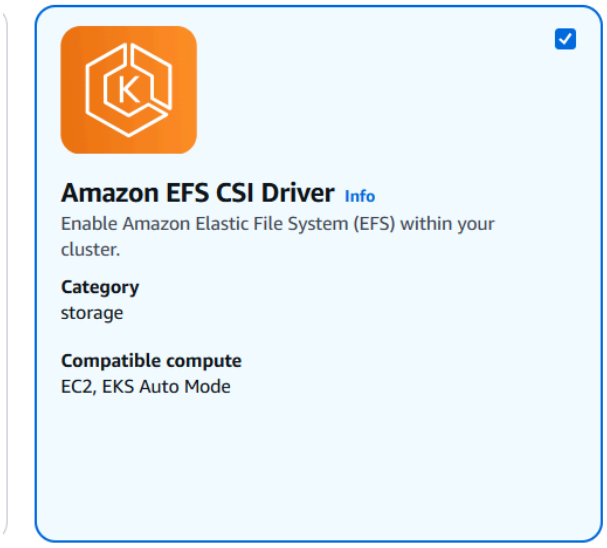
Aquí por defecto tendremos 6 subnets y nosotros no querremos tener las dos últimas (d, e, f), por tanto las sacamos y nos quedamos con us-east1a, b y c.



Queremos añadir 5 pods, que son:
kube-proxy
Amazon VPC CNI
CoreDNS
Amazon EKS Pod Identity Agent
Amazon EFS CSI Driver

El resto aparte de éstos, desmárcalo.





[santos-pardos (profesantos)]
GITHUB
[Install and Set Up kubectl on Linux | Kubernetes]

Instalar cliente
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/arm64/kubectl"

Instalar la última versión con este comando.

```
ubuntu@ip-172-31-90-175:~$ ls -l
total 54536
-rw-rw-r-- 1 ubuntu ubuntu 55836824 Feb 28 18:52 kubectl
drwx------ 4 ubuntu ubuntu     4096 Feb 28 18:27 snap
ubuntu@ip-172-31-90-175:~$ chmod a+x kubectl
ubuntu@ip-172-31-90-175:~$ ls -l
total 54536
-rwxrwxr-x 1 ubuntu ubuntu 55836824 Feb 28 18:52 kubectl
drwx------ 4 ubuntu ubuntu     4096 Feb 28 18:27 snap
```

aws eks update-kubeconfig --name demo-cluster --region us-east-1
kubectl config get-clusters

Tutorials

```
https://www.devopsworld.co.in/p/kubernetes-tutorials.html
https://github.com/antonputra/tutorials/tree/main
```

Install AWS Tools

```
AWS Tools (in Cloud9 you dont need to install AWS Tools)
aws --version
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
which aws
./aws/install --bin-dir /usr/bin --install-dir /usr/bin/aws-cli --update
aws --version
```

Install KUBECTL

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.3/2023-11-14/bin/linux/amd64/kubectl
chmod +x ./kubectl
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
kubectl version --client

aws eks update-kubeconfig --name demo-cluster --region us-east-1
kubectl config get-clusters
```

Install EKSCTL

aws eks update-kubeconfig --name demo-asix2b --region us-east-1
kubectl config get-clusters

```
ubuntu@ip-172-31-90-175:~$ aws eks update-kubeconfig --name demo-asix2b --region us-east-1
kubectl config get-clusters
Added new context arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b to /home/ubuntu/.kube/config
NAME
arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
ubuntu@ip-172-31-90-175:~$
```

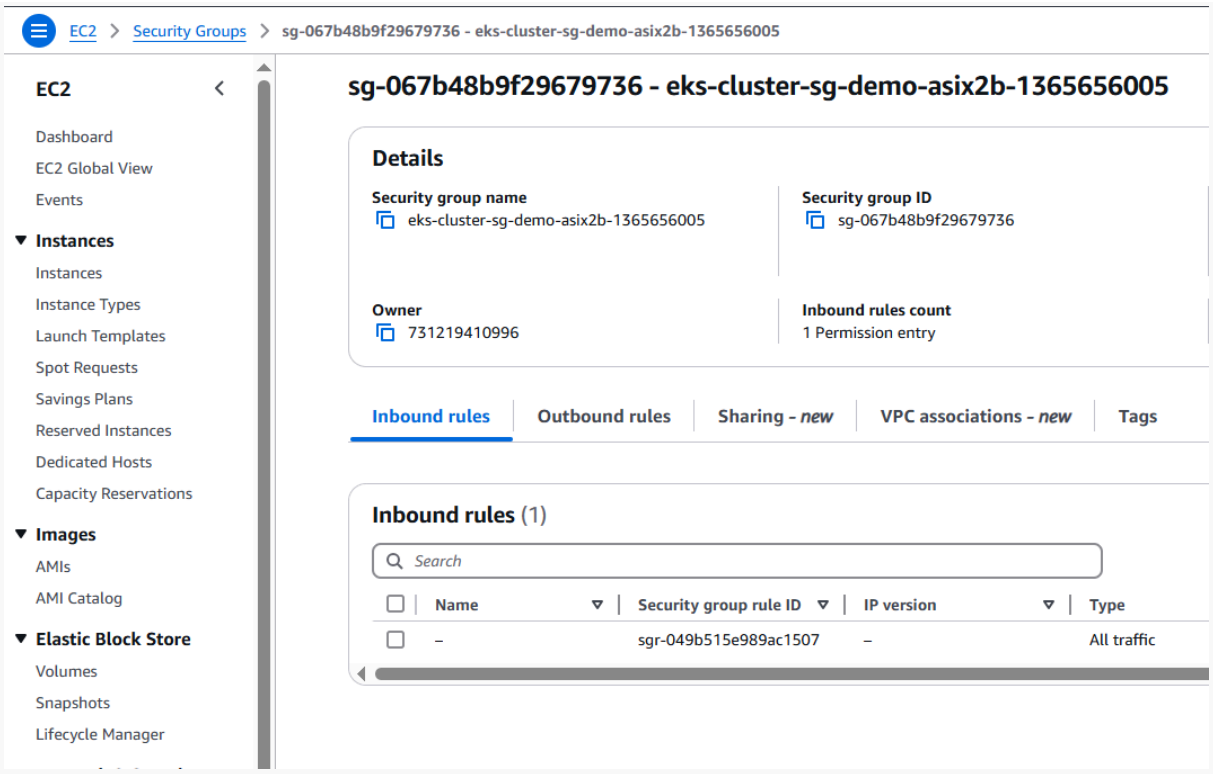
cat -n /home/ubuntu/.kube/config

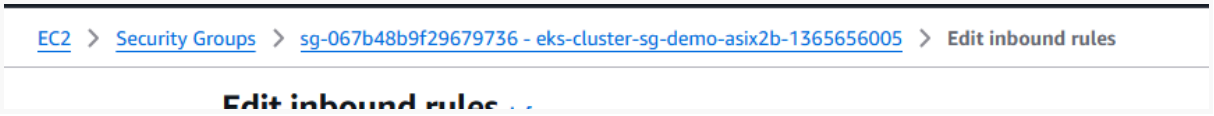
```
ubuntu@ip-172-31-90-175:~$ cat -n /home/ubuntu/.kube/config
     1  apiVersion: v1
     2  clusters:
     3  - cluster:
     4      certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJVU5USTVYczFpU
l3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TlRBeU1qZ3hPRFF4TURkYUZ3MHpOVEF5TWpZeE9
UTJNRGRhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUNkei8vRz
g5OHJyZ3lzajRjSmNES3pTVlVHTmxGSmRHK00rK2NmNVJDU2hUQ3E0YmVrNnRtOWduMVYKNzAyTXJnTFFNWDdqbEV3clNqQjQ0NXE2aENvcXF5WEdubmMw
NDNDd0gwN1lBdkNKV1FYY1J3YVYwbjNKTlVCdQpSRnZadWJqZjUrdnUrNTFXRytxTVA1Tmk5Q1pneFFkdWcxMkJWOFBNL2c1eDJjSzl4Y2FJWklBMnRWLz
NrR1NnCklyNEN0Y2lVaEczTHZQK1FHY1M0eVBOZmVuNHQ1Rm1jSzdkZkl0NUFKMERIRVpPdERzS0RWNGdvQmptMVpqSlYKSW5vOVJnQlJEaHF5L3pjc2Uv
eXlzdjJXeW1XcUdtQ2FORWFJbjRvY1BNbm5hYVZZa3h0a1NlN3NVT081REhoOQo5Nml3Vm9CYjNRY3N1TGoybTVRUlo3cElVYkdqQWdNQkFBR2pXVEJYTU
E0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJTZXlxbmVHM1ZabXJBNG4zQVB2WS9POU5DdWpEQVYK
QmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQXM1OFFkVFdIUgpSRExzR25vWStRMDNqeUIvK29EZVRqT0
9aUXVtZjlGQmFOc2NselVlaXUxZTdzb1E5RW1vQTJ3VENwYXcyUWlCCkVHU1hxN3NiN2JFb25sT3hra0pQTW9LdnZXL0cvQUsyN1dTV2FxdnJhUk96VEFy
VjUrVjlNVk9RbmIyUGEzNEEKNi9naUx5STdDQnBhK1dlckw3aHFNMVlSYzA4VFEzc2VRUkIwVWt3T1lXblRzYXNNVWxXajNicGV6ZmkwbDR2Ugo0MWdiSl
p0VS9iNnR4MDV2czAweERoVVRmU0xIRWIwREFOTndkZFJ0NFU4Z1M1cXdRRXRZUTFCSDdHekxLNkdsCkFyVGRaN3ZwYzNSc2ZsR3JaSU54aG5Xd0lDRmZt
SXF3RjIyT1BBUGJXdUJVUXR1ckwyMXF1UVB1WTAvdUtDZE8KaUJBczI1MENJMmRhCi0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
      5      server: https://2816889824995FF3F866728E9973236F.gr7.us-east-1.eks.amazonaws.com
      6    name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
      7  contexts:
      8  - context:
      9      cluster: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
     10      user: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
     11    name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
     12  current-context: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
     13  kind: Config
     14  preferences: {}
     15  users:
     16  - name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
     17    user:
     18      exec:
     19        apiVersion: client.authentication.k8s.io/v1beta1
     20        args:
     21        - --region
     22        - us-east-1
     23        - eks
     24        - get-token
```

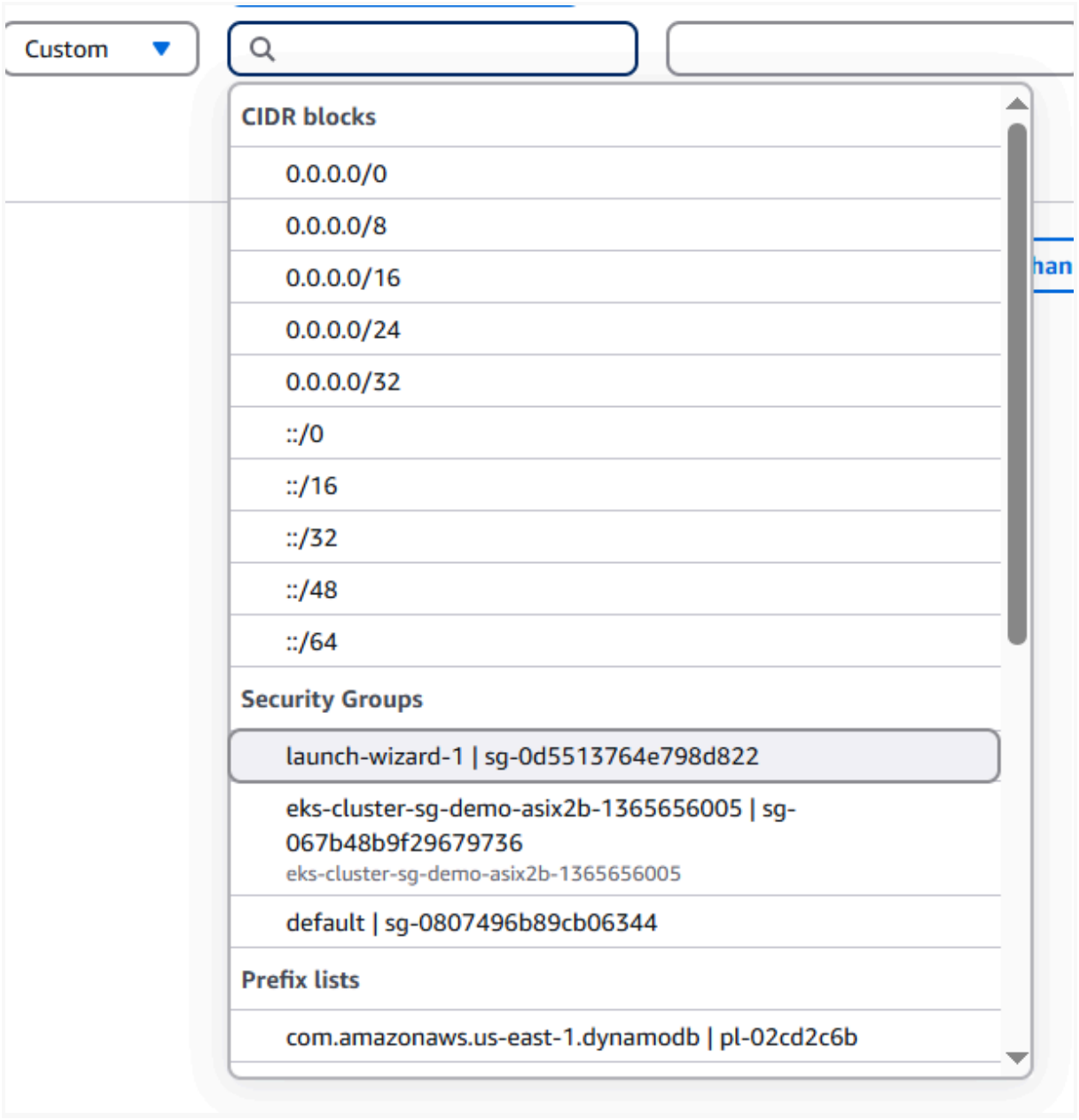
Via externa para conectarme al cluster:

```
ubuntu@ip-172-31-90-175:~$ cat -n /home/ubuntu/.kube/config
     1  apiVersion: v1
     2  clusters:
     3  - cluster:
     4      certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJVU5USTVYczFpUF
l3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TlRBeU1qZ3hPRFF4TURkYUZ3MHpOVEF5TWpZeE9E
UTJNRGRhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUNkei8vRz
g5OHJyZ3lzajRjSmNES3pTVlVHTmxGSmRHK00rK2NmNVJDU2hUQ3E0YmVrNnRtOWduMVYKNzAyTXJnTFFNWDdqbEV3clNqQjQ0NXE2aENvcXF5WEdubmMw
NDNDd0gwN1lBdkNKV1FYY1J3YVYwbjNKTlVCdQpSRnZadWJqZjUrdnUrNTFXRytxTVA1Tmk5Q1pneFFkdWcxMkJWOFBNL2c1eDJjSzl4Y2FJWklBMnRWLz
NrR1NnCklyNEN0Y2lVaEczTHZQK1FHY1M0eVBOZmVuNHQ1Rm1jSzdkZkl0NUFKMERIRVpPdERzS0RWNGdvQmptMVpqSlYKSW5vOVJnQlJEaHF5L3pjc2Uv
eXlzdjJXeW1XcUdtQ2FORWFJbjRvY1BNbm5hYVZZa3h0a1NlN3NVT081REhoOQo5Nml3Vm9CYjNRY3N1TGoybTVRUlo3cElVYkdqQWdNQkFBR2pXVEJYTU
E0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJTZXlxbmVHM1ZabXJBNG4zQVB2WS9POU5DdWpEQVYK
QmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQXM1OFFkVFdIUgpSRExzR25vWStRMDNqeUIvK29EZVRqT0
9aUXVtZjlGQmFOc2NselVlaXUxZTdzb1E5RW1vQTJ3VENwYXcyUWlCCkVHU1hxN3NiN2JFb25sT3hra0pQTW9LdnZXL0cvQUsyN1dTV2FxdnJhUk96VEFy
VjUrVjlNVk9RbmIyUGEzNEEKNi9naUx5STdDQnBhK1dlckw3aHFNMVlSYzA4VFEzc2VRUkIwVWt3T1lXblRzYXNNVWxXajNicGV6ZmkwbDR2Ugo0MWdiSl
p0VS9iNnR4MDV2czAweERoVVRmU0xIRWIwREFOTndkZFJ0NFU4Z1M1cXdRRXRZUTFCSDdHekxLNkdsCkFyVGRaN3ZwYzNSc2ZsR3JaSU54aG5Xd0lDRmZt
SXF3RjIyT1BBUGJXdUJVUXR1ckwyMXF1UVB1WTAvdUtDZE8KaUJBczI1MENJMmRhCi0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
      5      server: https://2816889824995FF3F866728E9973236F.gr7.us-east-1.eks.amazonaws.com
      6    name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
```

https://2816889824995FF3F866728E9973236F.gr7.us-east-1.eks.amazonaws.com


EC2 > Security Groups > sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005
Details
Security group name
eks-cluster-sg-demo-asix2b-1365656005
Security group ID
sg-067b48b9f29679736
Owner
731219410996
Inbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
Inbound rules (1)
Search
Name
Security group rule ID
IP version
Type
–
sgr-049b515e989ac1507
–
All traffic


EC2 > Security Groups > sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005 > Edit inbound rules
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
sgr-049b515e989ac1507	All traffic	All	All	Custom
–	All traffic	All	All	Custom

Add rule

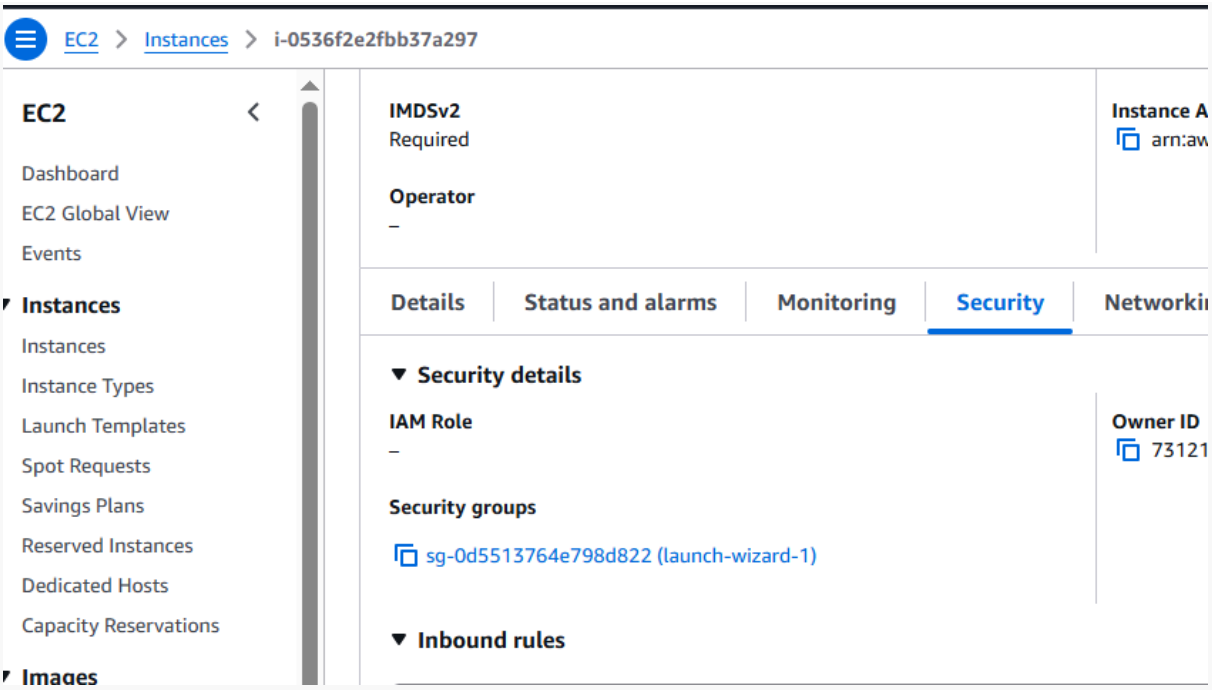
Edit inbound rules Info

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

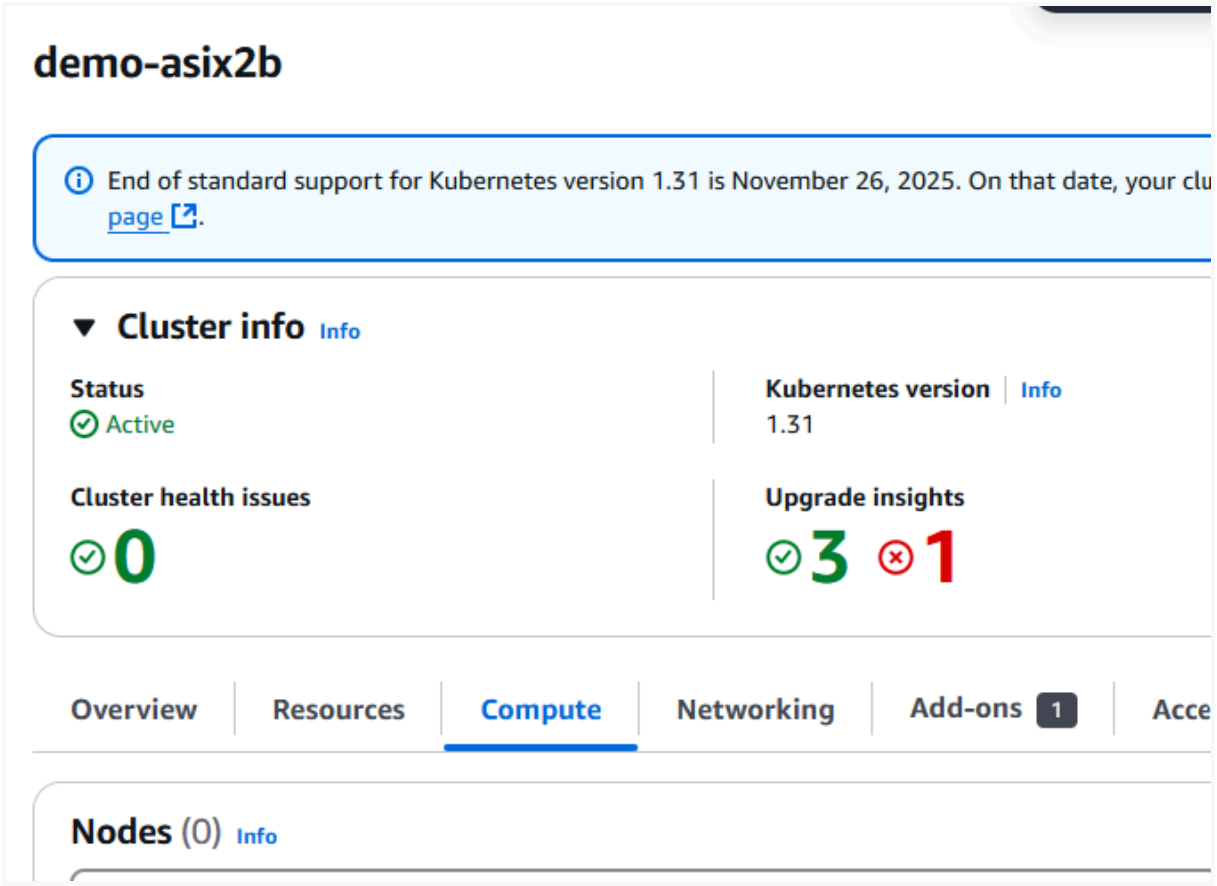
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Descri
sgr-049b515e989ac1507	All traffic	All	All	Custom	sg-067b48b9f29679736
–	All traffic	All	All	Custom	sg-0d5513764e798d822 / sg-0d5513764e798d822

Add rule




```
ubuntu@ip-172-31-90-175:~$ kubectl get nodes
No resources found
ubuntu@ip-172-31-90-175:~$
```



Compute no tengo nada
Queremos máquina virtuales donde están

Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

mogollon-ec2-asxi2b

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

LabRole

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

Launch template Info

Step 1
Configure node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple no

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

mogollon-ec2-asxi2b

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphe

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

Select role

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

Launch template Info

Step 1
Configure node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Set compute and scaling configuration

Node group compute configuration

These properties cannot be changed after the node group is created.

AMI type Info

Select the EKS-optimized Amazon Machine Image for nodes.

Amazon Linux 2023 (x86_64) Standard (AL2023_x86_64_STANDARD)

Capacity type

Select the capacity purchase option for this node group.

On-Demand

Instance types Info

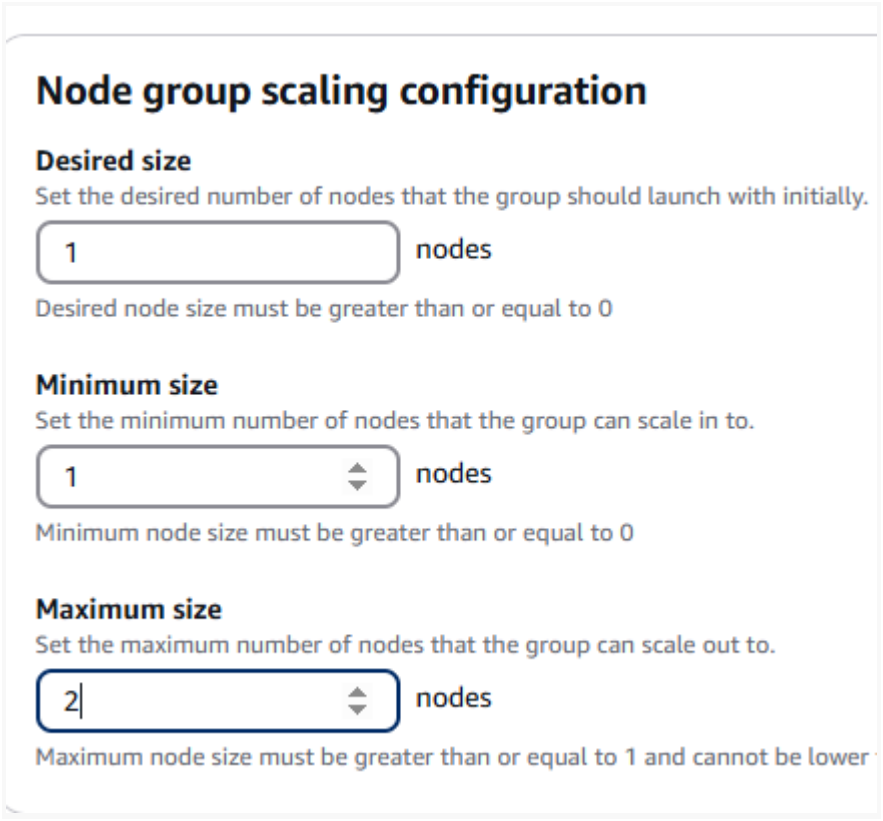
Select instance types you prefer for this node group.

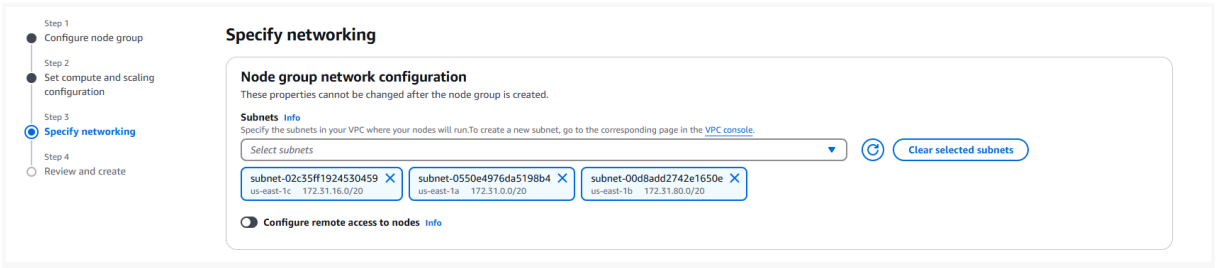
Enter an instance type

t3.large
vCPU: 2 vCPUs Memory: 8 GiB Network: Up to 5 Gigabit Max ENI: 3 Max IPs: 36

Disk size

Select the size of the attached EBS volume for each node.





next
CREATEEE

```
ubuntu@ip-172-31-90-175:~$ kubectl get nodes
NAME                            STATUS   ROLES    AGE     VERSION
ip-172-31-27-119.ec2.internal   Ready    <none>   3m36s   v1.31.5-eks-5d632ec
ubuntu@ip-172-31-90-175:~$
```

Ya tenemos una ec2 donde podemos desplegar los contenedores.
Comprobaciones
https://github.com/santos-pardos/K8s-Eks.git
git clone

```
ubuntu@ip-172-31-90-175:~$ kubectl get nodes
NAME                           STATUS   ROLES    AGE     VERSION
ip-172-31-27-119.ec2.internal   Ready    <none>   3m36s   v1.31.5-eks-5d632ec
ubuntu@ip-172-31-90-175:~$ git clone https://github.com/santos-pardos/K8s-Eks.git
Cloning into 'K8s-Eks'...
remote: Enumerating objects: 1288, done.
remote: Counting objects: 100% (930/930), done.
remote: Compressing objects: 100% (548/548), done.
remote: Total 1288 (delta 370), reused 872 (delta 346), pack-reused 358 (from 1)
Receiving objects: 100% (1288/1288), 12.16 MiB | 30.46 MiB/s, done.
Resolving deltas: 100% (471/471), done.
ubuntu@ip-172-31-90-175:~$
```

```
ubuntu@ip-172-31-90-175:~$ cd K8s-Eks/
ubuntu@ip-172-31-90-175:~/K8s-Eks$ cd HelloWorld/
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ ls -l
total 16
-rw-rw-r-- 1 ubuntu ubuntu 511 Feb 28 19:22 README.md
-rw-rw-r-- 1 ubuntu ubuntu 561 Feb 28 19:22 helloworld-deployment.yaml
-rw-rw-r-- 1 ubuntu ubuntu 568 Feb 28 19:22 helloworld-svc-https.yaml
-rw-rw-r-- 1 ubuntu ubuntu 186 Feb 28 19:22 helloworld-svc.yaml
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ cat -n helloworld-deployment.yaml
     1  ---
     2  apiVersion: apps/v1
     3  kind: Deployment
     4  metadata:
     5    name: hello-kubernetes
     6  spec:
     7    replicas: 3
     8    selector:
     9      matchLabels:
    10        app: hello-kubernetes
    11    template:
    12      metadata:
    13        labels:
    14          app: hello-kubernetes
    15      spec:
    16        containers:
    17        - name: hello-kubernetes
    18          image: santospardos/sanvalero:hello-k8s
    19          ports:
    20          - containerPort: 8080
    21          resources:
    22            requests:
    23              memory: "64Mi"
    24              cpu: "80m"
    25            limits:
    26              memory: "128Mi"
    27              cpu: "250m"
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

Que hago para despegar o poenr en marcha?
kubectl apply -f helloworld-deployment.yaml

Pero antes:kubectl get pods

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ #kubectl apply -f helloworld-deployment.yam
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get pods
No resources found in default namespace.
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

No hay contenedores.

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                  READY   STATUS    RESTARTS   AGE
kube-system   aws-node-xzlck                        2/2     Running   0          8m54s
kube-system   coredns-789f8477df-8799p              1/1     Running   0          35m
kube-system   coredns-789f8477df-wdqgn              1/1     Running   0          35m
kube-system   ebs-csi-controller-84fd6498d4-bp2qk   6/6     Running   0          35m
kube-system   ebs-csi-controller-84fd6498d4-fqhq9   6/6     Running   0          35m
kube-system   ebs-csi-node-846vt                    3/3     Running   0          8m54s
kube-system   eks-pod-identity-agent-m7tsp          1/1     Running   0          8m54s
kube-system   kube-proxy-mxgxc                      1/1     Running   0          8m54s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

Estos son los plugins que hemos seleccionado antes.

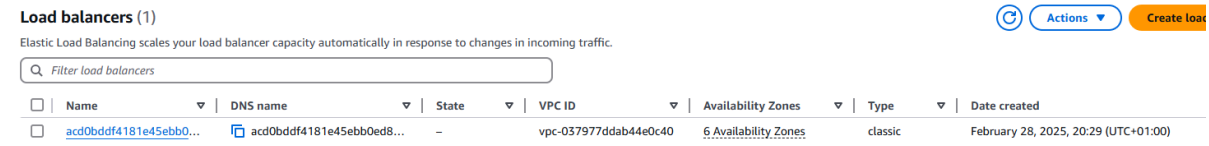
```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-deployment.yaml
deployment.apps/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
hello-kubernetes-654f5d8885-9xwmh   1/1     Running   0          8s
hello-kubernetes-654f5d8885-mmfp8   1/1     Running   0          8s
hello-kubernetes-654f5d8885-vhm4t   1/1     Running   0          8s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

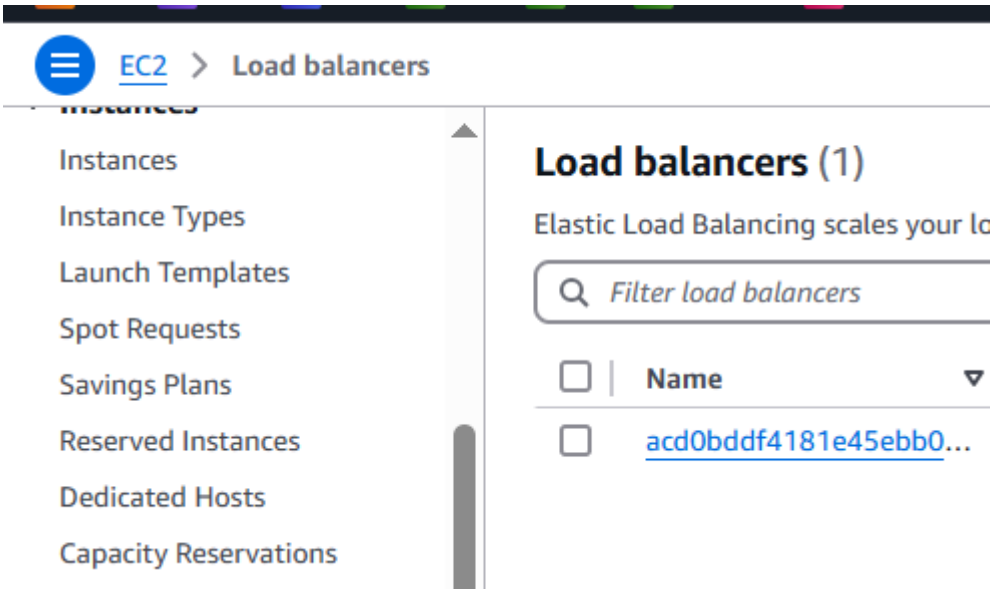
```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/hello-kubernetes-654f5d8885-9xwmh   1/1     Running   0          27s
pod/hello-kubernetes-654f5d8885-mmfp8   1/1     Running   0          27s
pod/hello-kubernetes-654f5d8885-vhm4t   1/1     Running   0          27s

NAME                 TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.100.0.1    <none>        443/TCP   41m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes    3/3     3            3           27s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-654f5d8885   3         3         3       27s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ S
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ ls -l
total 16
-rw-rw-r-- 1 ubuntu ubuntu 511 Feb 28 19:22 README.md
-rw-rw-r-- 1 ubuntu ubuntu 561 Feb 28 19:22 helloworld-deployment.yaml
-rw-rw-r-- 1 ubuntu ubuntu 568 Feb 28 19:22 helloworld-svc-https.yaml
-rw-rw-r-- 1 ubuntu ubuntu 186 Feb 28 19:22 helloworld-svc.yaml
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-svc.yaml
service/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

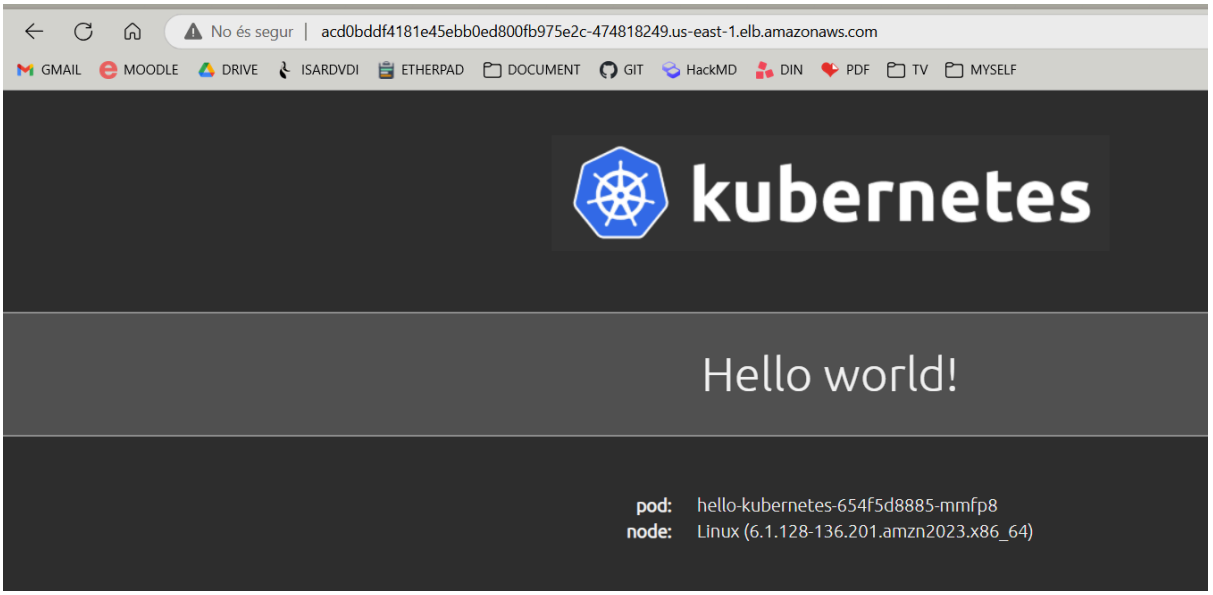
```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ cat -n helloworld-svc.yaml
     1  ---
     2  apiVersion: v1
     3  kind: Service
     4  metadata:
     5    name: hello-kubernetes
     6  spec:
     7    type: LoadBalancer
     8    ports:
     9    - port: 80
    10      targetPort: 8080
    11    selector:
    12      app: hello-kubernetesubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWor
```





O tambien

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/hello-kubernetes-654f5d8885-9xwmh    1/1     Running   0          5m39s
pod/hello-kubernetes-654f5d8885-mmfp8    1/1     Running   0          5m39s
pod/hello-kubernetes-654f5d8885-vhm4t    1/1     Running   0          5m39s

NAME                         TYPE           CLUSTER-IP      EXTERNAL-IP
                                    PORT(S)        AGE
service/hello-kubernetes     LoadBalancer   10.100.231.37   acd0bddf4181e45ebb0ed800fb975e2c-474
818249.us-east-1.elb.amazonaws.com   80:31344/TCP   3m36s
service/kubernetes           ClusterIP      10.100.0.1      <none>
                                    443/TCP        46m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes    3/3     3            3           5m39s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-654f5d8885    3         3         3       5m39s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```



Objetivo cambiar esto y que aparezca el pupito

Debemos eliminar los dos kubernetas que hemos desplegado antes por no tener problemas.

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl delete -f helloworld-svc.yaml
service "hello-kubernetes" deleted
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl delete -f helloworld-deployment.yaml
deployment.apps "hello-kubernetes" deleted
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   54m
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get deployments
No resources found in default namespace.
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl create deployment podinfo --image=stefan
prodan/podinfo
deployment.apps/podinfo created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
podinfo-849bfb5c8d-6grjj   1/1     Running   0          19s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

Editamos el yaml de deployment para cambiar la imagen de los containers por la de stefanprodan o el containerPort abierto a 9898.

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ cat -n helloworld-deployment.yaml
     1  ---
     2  apiVersion: apps/v1
     3  kind: Deployment
     4  metadata:
     5    name: hello-kubernetes
     6  spec:
     7    replicas: 3
     8    selector:
     9      matchLabels:
    10        app: hello-kubernetes
    11    template:
    12      metadata:
    13        labels:
    14          app: hello-kubernetes
    15      spec:
    16        containers:
    17        - name: hello-kubernetes
    18          image: stefanprodan/podinfo
    19          ports:
    20          - containerPort: 9898
    21          resources:
    22            requests:
    23              memory: "64Mi"
    24              cpu: "80m"
    25            limits:
    26              memory: "128Mi"
    27              cpu: "250m"
```

Lo mismo con el yaml del servicio LoadBalancer.

```
  GNU nano 7.2                                  helloworld-svc.yaml
---
apiVersion: v1
kind: Service
metadata:
  name: hello-kubernetes
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 9898
  selector:
    app: hello-kubernetes
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ nano helloworld-deployment.yaml
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-deployment.yaml
deployment.apps/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ nano helloworld-svc.yaml
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-svc.yaml
service/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                                 READY   STATUS             RESTARTS   AGE
pod/hello-kubernetes-d98f94888-d9l6z   0/1     ImagePullBackOff   0          72s
pod/hello-kubernetes-d98f94888-qbfxr   0/1     ImagePullBackOff   0          72s
pod/hello-kubernetes-d98f94888-rjpxf   0/1     ImagePullBackOff   0          72s
pod/podinfo-849bfb5c8d-6grjj           1/1     Running            0          3m53s

NAME                       TYPE           CLUSTER-IP      EXTERNAL-IP
                                  PORT(S)        AGE
service/hello-kubernetes   LoadBalancer   10.100.23.187   ad5508ce8339a4879803479f130f68fd-177
1309034.us-east-1.elb.amazonaws.com   80:32554/TCP   28s
service/kubernetes         ClusterIP      10.100.0.1      <none>
                                  443/TCP        62m

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes   0/3     3            0           72s
deployment.apps/podinfo            1/1     1            1           3m53s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-d98f94888    3         3         0       72s
replicaset.apps/podinfo-849bfb5c8d            1         1         1       3m53s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl delete -f helloworld-deployment.yaml
deployment.apps "hello-kubernetes" deleted
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-svc.yaml
service/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl apply -f helloworld-deployment.yaml
deployment.apps/hello-kubernetes created
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                                   READY   STATUS    RESTARTS   AGE
pod/hello-kubernetes-74594654bb-lsxt4   1/1     Running   0          10s
pod/hello-kubernetes-74594654bb-p7lnt   1/1     Running   0          10s
pod/hello-kubernetes-74594654bb-stpqj   1/1     Running   0          10s

NAME                       TYPE           CLUSTER-IP      EXTERNAL-IP
                                  PORT(S)        AGE
service/hello-kubernetes   LoadBalancer   10.100.236.74   aeb15947d37384636a0af39e589882d3-181
7738793.us-east-1.elb.amazonaws.com   80:31624/TCP   20s
service/kubernetes         ClusterIP      10.100.0.1      <none>
                                  443/TCP        74m

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-kubernetes   3/3     3            3           10s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-kubernetes-74594654bb   3         3         3       10s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
NAME                                   READY   STATUS    RESTARTS   AGE
pod/hello-kubernetes-74594654bb-lsxt4   1/1     Running   0          46s
pod/hello-kubernetes-74594654bb-p7lnt   1/1     Running   0          46s
pod/hello-kubernetes-74594654bb-stpqj   1/1     Running   0          46s

NAME                       TYPE           CLUSTER-IP      EXTERNAL-IP
```