

PROYECTO FINAL PIN3

MUNDOSE DEVOPS 2303

GONZALO MAHSERDJIAN

DATOS TÉCNICOS

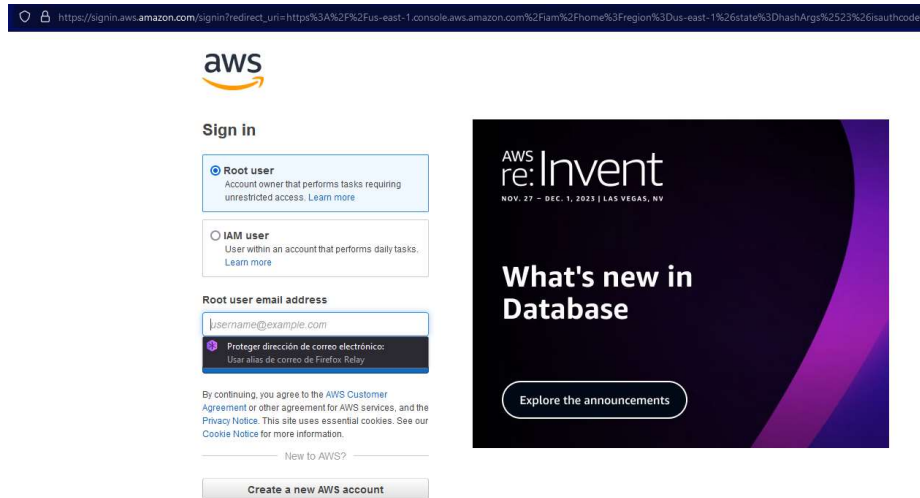
- NOMBRE: Proyecto Final (PIN3) de Diplomatura DevOps de MundosE
- DESCRIPCIÓN: Este es el paso a paso del proceso manual del PIN3.
- REPOSITORIO: <https://github.com/gsmx64/MundosE-DevOps2303-PIN3/tree/main>
- LICENCIA: GNU General Public License v3 (GPLv3)

ÍNDICE

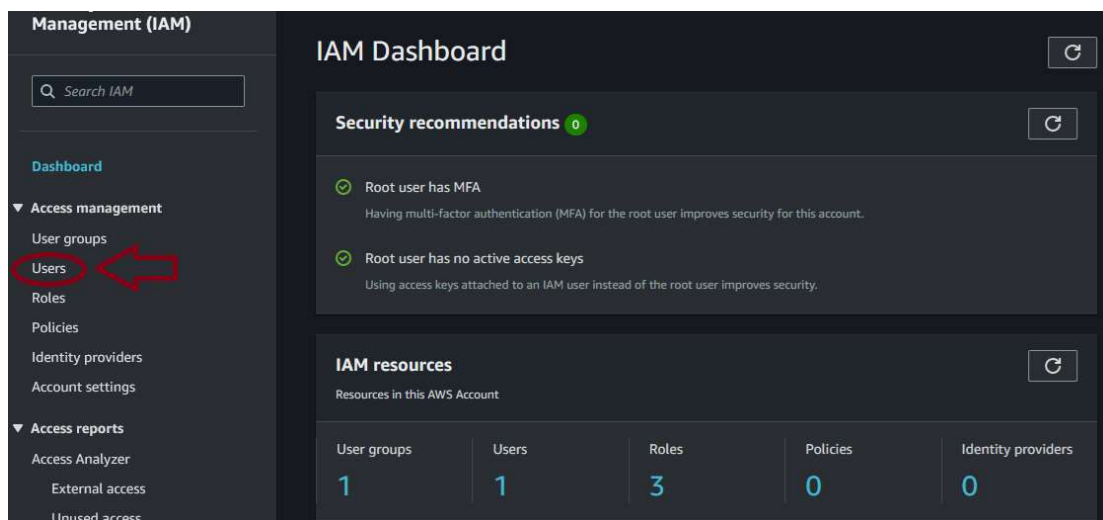
Creación de keys.....	1
Creación de instancia de EC2.....	3
Conectarse a instancia de EC2 por medio de AWS CLI.....	14
Creación del cluster.....	17
Herramientas de monitoreo – EBS Driver.....	19
Herramientas de monitoreo - Prometheus.....	21
Herramientas de monitoreo - Grafana.....	26

CREACIÓN DE KEYS DE AWS

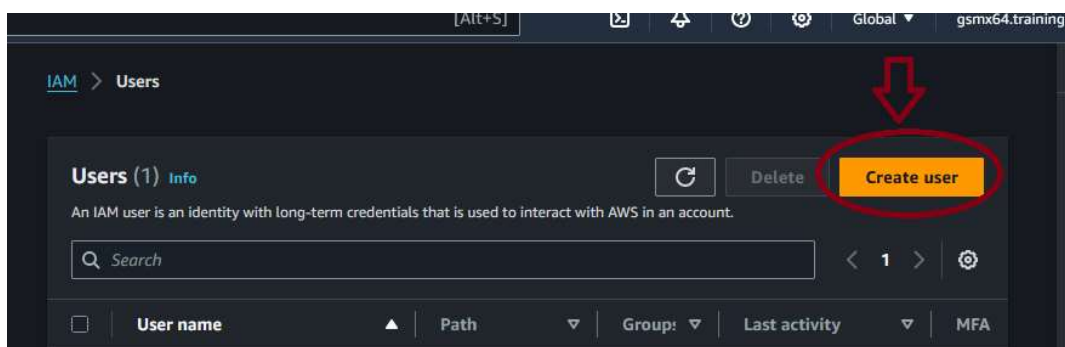
El primer paso es iniciar sesión en AWS con "Root user" o "IAM User":



Luego entramos en IAM y del lado izquierdo en el dashboard vamos a "Users":



Y del lado derecho en el botón naranja vamos a "Create user":



Le ingresamos un nombre de usuario, tildamos la opción para que tenga acceso a consola de AWS "Provide user access to the AWS Management Console – optional", seleccionamos "I want to create an IAM user" y debajo de todo destildamos en mi caso la opción "Users must create a new password at next sign-in" así me genera una contraseña random el mismo sistema y le damos a "Next":

User name
gsm pin3

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - optional
If you're providing [console access](#) to a person, it's a best practice [to](#) manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☒ Autogenerated password
You can view the password after you create the user.

☐ Custom password
Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | ' "

☐ Show password

☐ Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

En el siguiente paso nos consulta sobre los permisos para este usuario, para este escenario lo voy a crear dentro de un grupo, para ello le damos click en “Add user to group” y luego nos aparece un cuadro debajo donde hay que hacerle click en el botón izquierdo llamado “Create group”, y click en “Next”:

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1)

Search

Create group

<input type="checkbox"/>	Group name	Users	Attached policies	Created
<input type="checkbox"/>	pin2-deploy	1	AmazonEC2FullAccess, AmazonS3...	2023-12-11 ...

Ahora consulta sobre qué permisos se asignarán a este nuevo grupo, de momento le estaré dando permisos totales EC2 (para luego cuando funcione todo ir bajando esos permisos al mínimo), así que le asigno las políticas “AmazonEC2FullAccess” y “AWSCloudFormationFullAccess”; finalmente abajo click en el botón amarillo “Create user group”:

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name
Enter a meaningful name to identify this group.
pin3-deploy
Maximum 128 characters. Use alphanumeric and '+,=, @, -, _' characters.

Permissions policies (1/912)

Filter by Type: All types | 1 match

<input checked="" type="checkbox"/>	Policy name	Type	Use...	Descri
<input checked="" type="checkbox"/>	AmazonEC2FullAccess	AWS managed	Permis...	Provid

Cancel Create user group

Esto nos lleva al siguiente apartado donde ya está creado el grupo, y debemos tildarlo y darle click en el botón de abajo derecha y amarillo que dice “Next”, así agrega este grupo al usuario creado:

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- ☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/2)

<input type="checkbox"/>	Group name	Users	Attached policies	Crea...
<input type="checkbox"/>	pin2-deploy	1	AmazonEC2FullAccess, AmazonS3...	2023-1...
<input checked="" type="checkbox"/>	pin3-deploy	0	AmazonEC2FullAccess	2024-0...

► Set permissions boundary - optional

Cancel Previous Next

Entonces llegamos a este apartado donde solo resta darle click en el botón amarillo de abajo derecha que dice “Create user” (y si quisiera también aquí podría relacionarlo con un tag para asociar los recursos):

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
gsm pin3	Autogenerated	No

Permissions summary

Name	Type	Used as
pin3-deploy	Group	Permissions group

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#)
[Previous](#)
[Create user](#)

Una vez que ya tengo el usuario creado, debo hacerle click en “show” para ver la contraseña random que me generó IAM y guardarla en un lugar seguro, porque no la utilizaré por ahora, ya que generaré un certificado pem para utilizarlo en AWS CLI; y hago click en “Return to users list”:

🟢
User created successfully
[View user](#)

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[IAM](#) > [Users](#) > [Create user](#)

Step 1
[Specify user details](#)

Step 2
[Set permissions](#)

Step 3
[Review and create](#)

Step 4
Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL

[https://385694789955.signin.aws.amazon.com/console](#)

User name

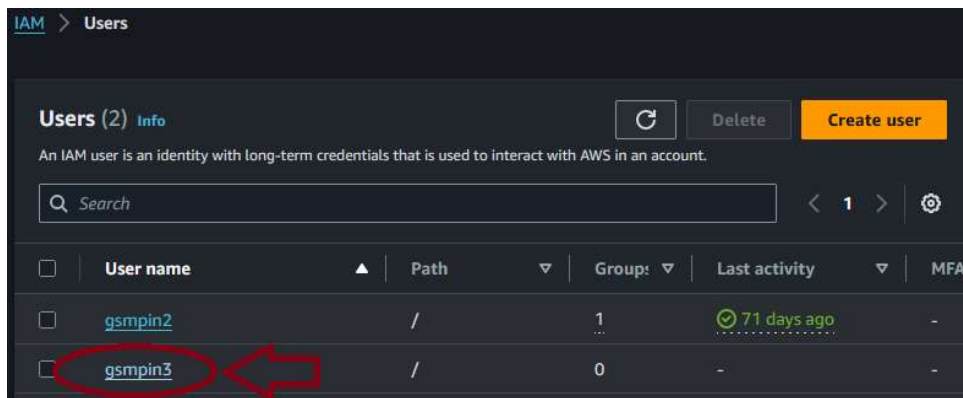
[gsm pin3](#)

Console password

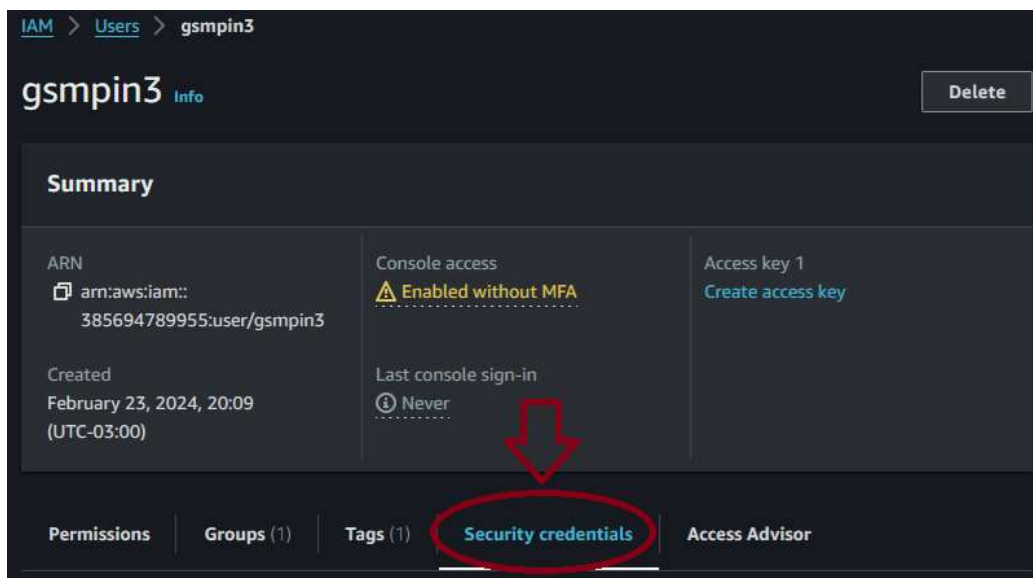
[*****](#) [Show](#)

[Cancel](#)
[Download .csv file](#)
[Return to users list](#)

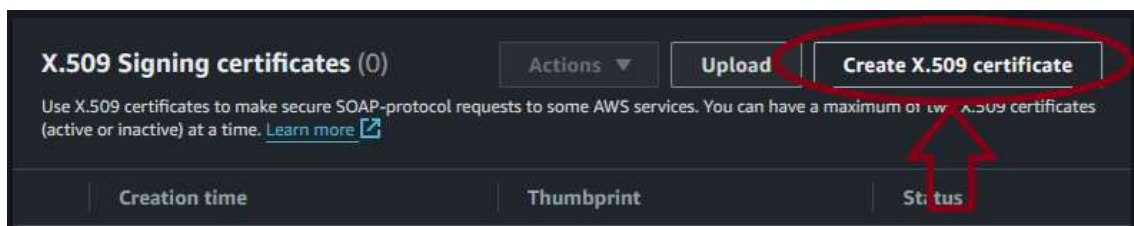
Ahora ingreso al usuario creado haciéndole click en su nombre:



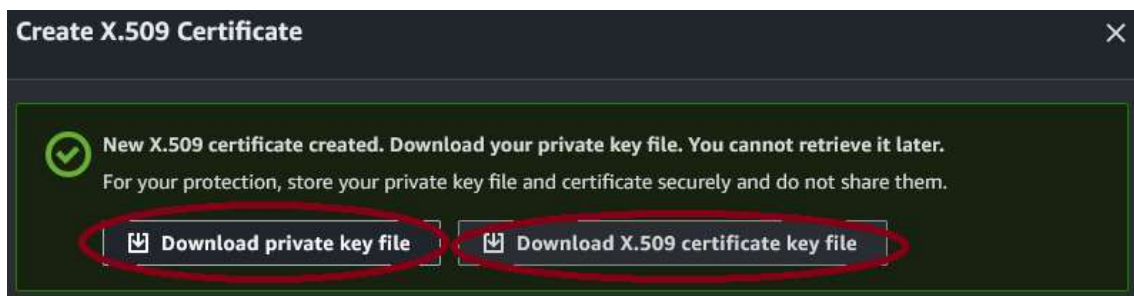
Y voy al tab "Security credentials" y voy debajo de todo a "X.509 Signing certificates":



Le hago click en "Create X.509 certificate":

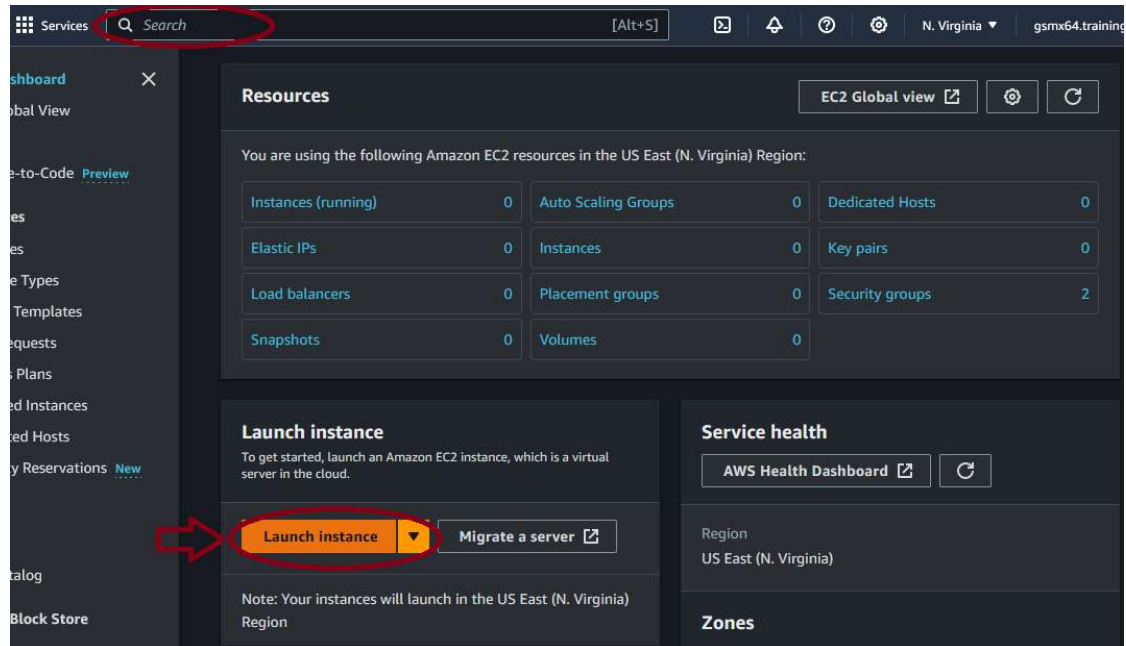


Y descargo la clave privada y el certificado a un lugar seguro, luego click en "Close":

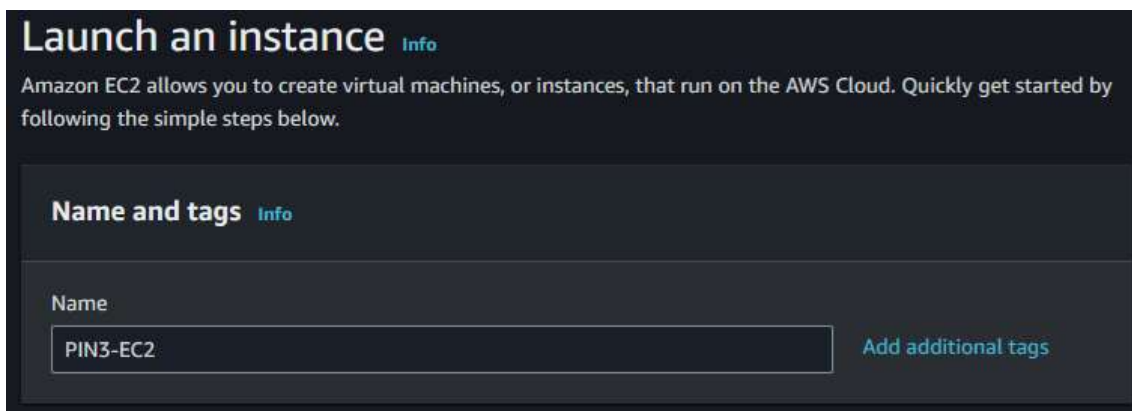


CREACIÓN DE INSTANCIA EC2

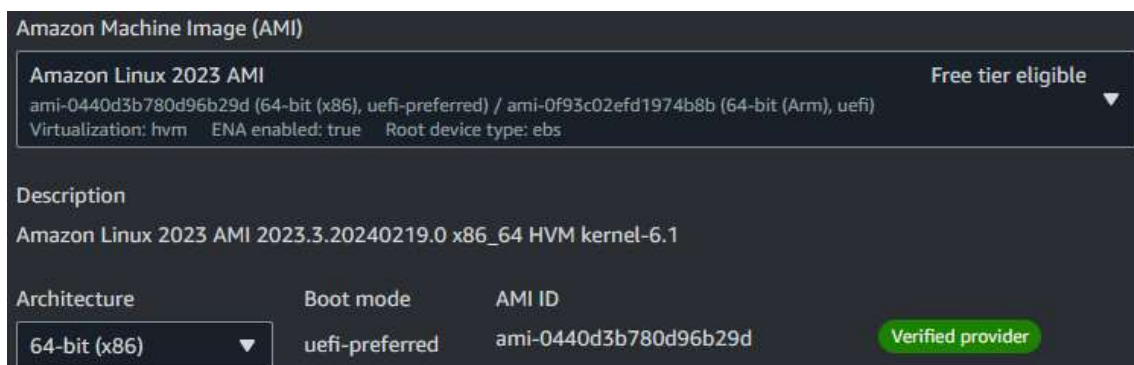
Ahora en el apartado de búsqueda arriba de todo, escribo EC” y entro al dashboard del mismo, para luego hacer click en “Launch Instance”:



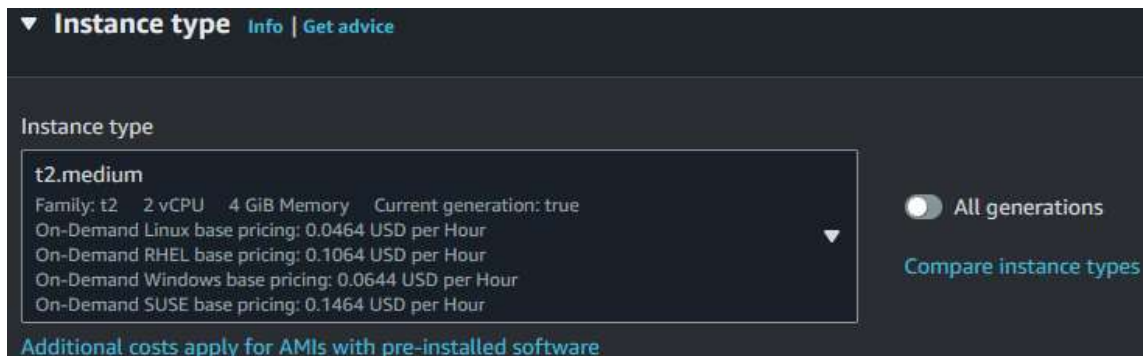
Le ingreso de nombre “PIN3-EC2” ...



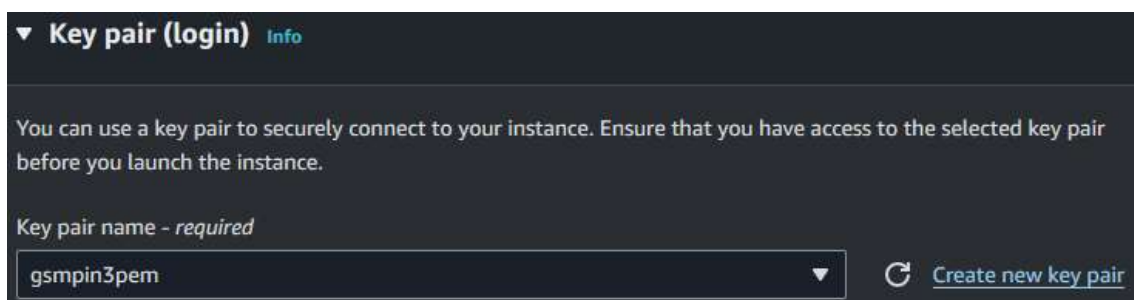
Selecciono como sistema operativo Amazon Linux 2023 AMI, arquitectura 64 bits, tomo nota del AMI ID para guardarlo en el archivo .env ...



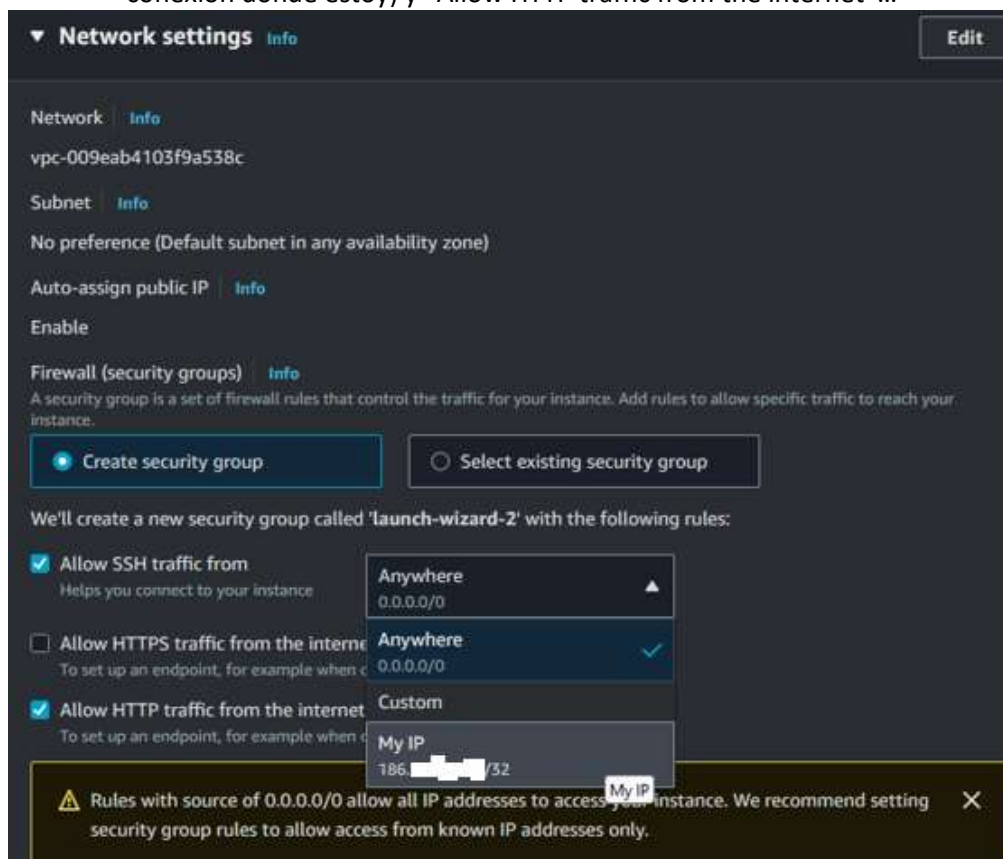
En instancia selecciono “t2.medium” (es la que le sigue al de free tier, y tengo créditos de AWS disponibles con lo cual no sería un costo para este proyecto final):



Selecciono un certificado o genero uno nuevo en “Key pair”...



Ingresa la configuración de la VPC, click e “Create security group” si no tengo alguna definida, selecciono “Allow SSH traffic from” (por seguridad solo desde “My IP” para acceder desde la conexión donde estoy) y “Allow HTTP traffic from the internet”...



Configuro el storage, en 29GiB ó 30GiB...

Configure storage [Info](#) Advanced

1x 29 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

Y todo listo por ahora, le doy click del lado derecho a “Launch instance...”

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...[read more](#)
ami-0440d3b780d96b29d

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 29 GiB

[Cancel](#) [Launch instance](#) [Review commands](#)

Cuando se complete, le hago click al enlace de la instancia creada...

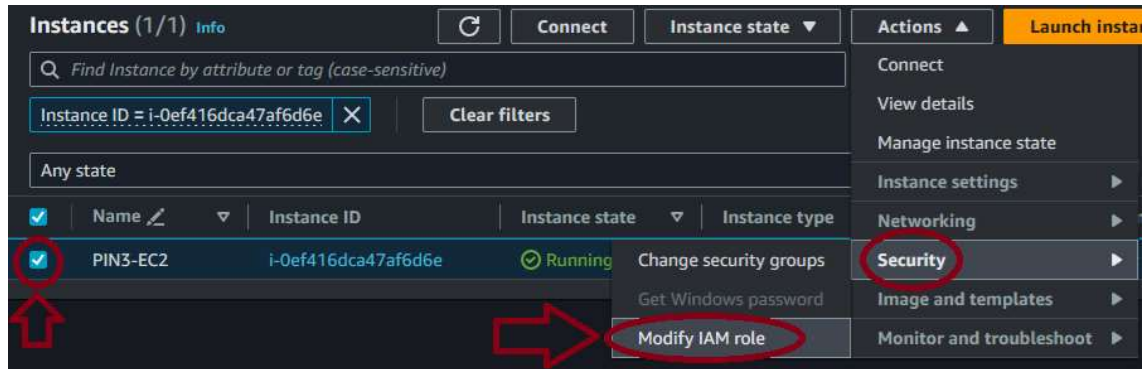
[EC2](#) > [Instances](#) > Launch an instance

Success
Successfully initiated launch of instance **i-0ef416dca47af6d6e**

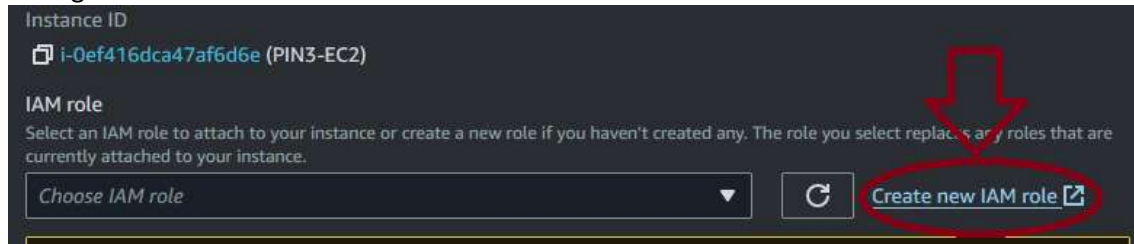
Launch log

Initializing requests	✓ Succeeded
Creating security groups	✓ Succeeded
Creating security group rules	✓ Succeeded
Launch initiation	✓ Succeeded

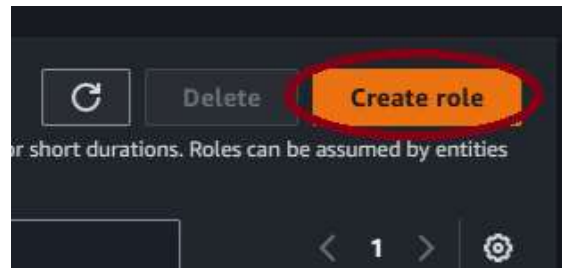
Selecciono la instancia, y arriba a la derecha hago click en “Actions”, “Security”, “Modify IAM role”...



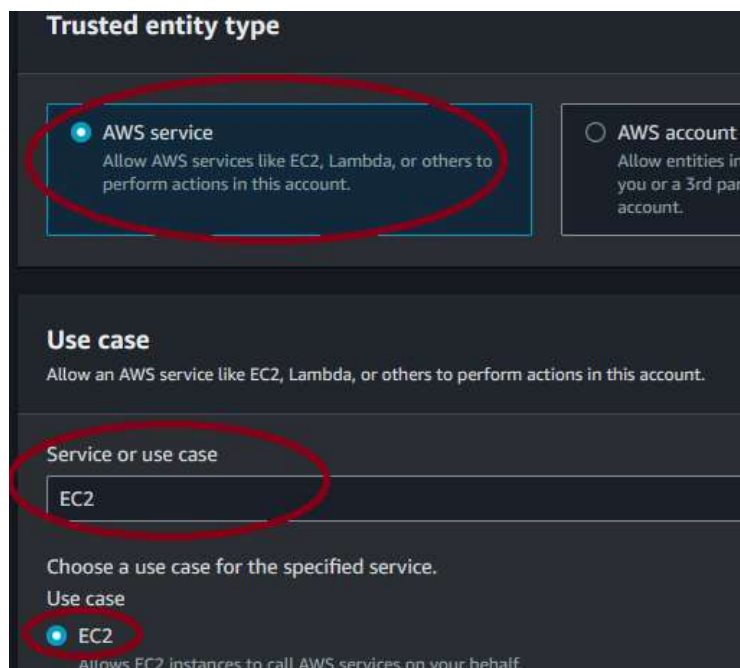
Le hago click en “Create new IAM role”...



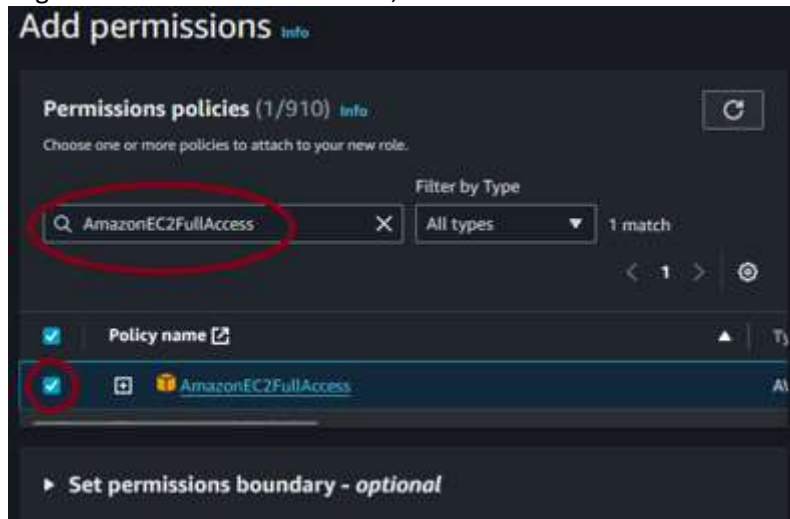
y en la página siguiente nuevamente en “Create role”...



Selecciono “AWS Service”, busco y selecciono “EC2”, y nuevamente click en “EC2”, click en “Next”...

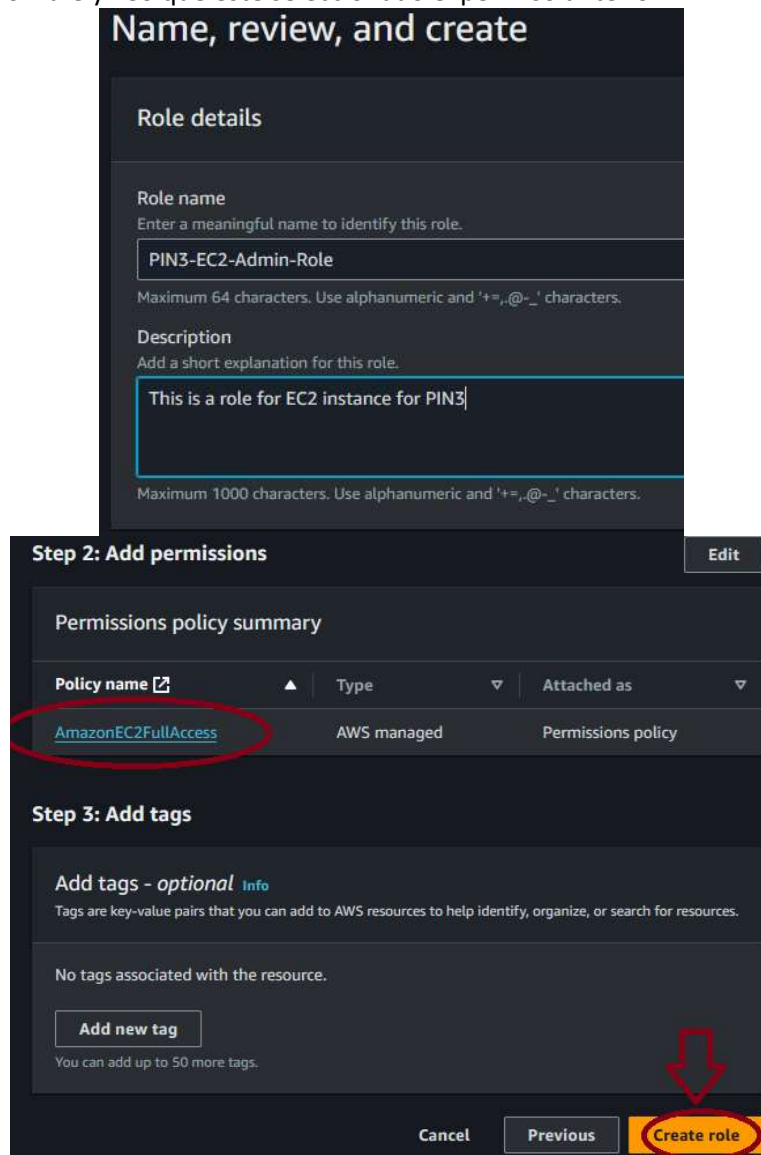


Busco el rol a asignar: “AmazonEC2FullAccess”, lo selecciono...

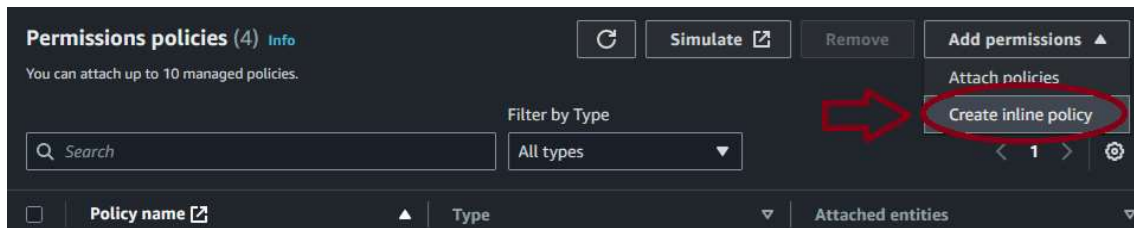


Tambien hay que agregar “AWSCloudFormationFullAccess”, le damos en siguiente...

Le ingreso un nombre y veo que esté seleccionado el permiso anterior...



Despues hay que editar el rol y agregarle 2 policias más, llendo a “Create inline policy”:



Agregar:

Con nombre “PIN3_EKS_Policy”:

PIN3_EKS_Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "eks:*",
      "Resource": "*"
    },
    {
      "Action": [
        "ssm:GetParameter",
        "ssm:GetParameters"
      ],
      "Resource": [
        "arn:aws:ssm:*:<account_id>:parameter/aws/*",
        "arn:aws:ssm:*:parameter/aws/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "logs:PutRetentionPolicy"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Con nombre “PIN3_IamLimitedAccess”:

PIN3_IamLimitedAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateInstanceProfile",
        "iam:DeleteInstanceProfile",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:GetRole",
        "iam:CreateRole",

```

Permissions policies (4) Info

↺

Simulate ↗

Remove

Add permissions ▾





You can attach up to 10 managed policies.

🔍 Search

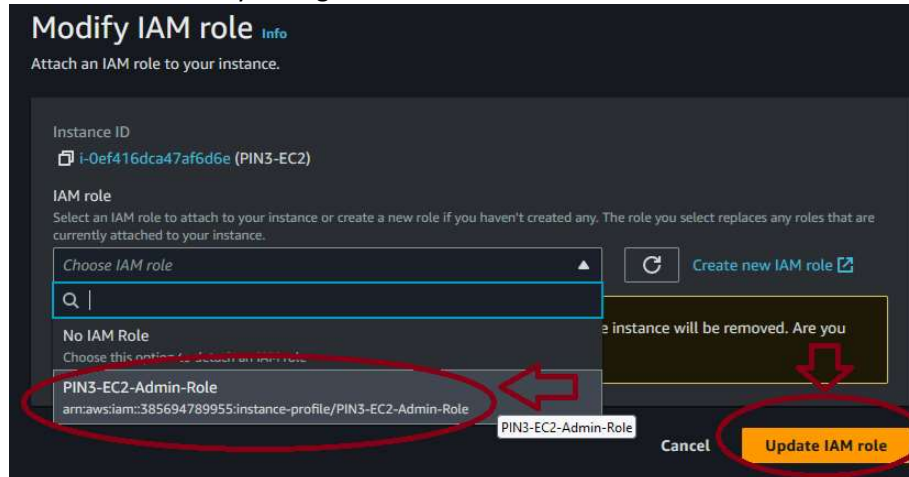
Filter by Type

All types ▾

< 1 > ⚙️

<input type="checkbox"/>	Policy name ↗	Type ▾	Attached entities ▾
<input type="checkbox"/>	 PIN3_CloudFormation_Policy	Customer inline	0
<input type="checkbox"/>	 PIN3_EC2_Policy	Customer inline	0
<input type="checkbox"/>	 PIN3_EKS_Policy	Customer inline	0
<input type="checkbox"/>	 PIN3_IamLimitedAccess	Customer inline	0

Vuelvo a la instancia de EC2 y le asigno el nuevo rol creado...



CONECTARSE A INSTANCIA EC2 POR MEDIO DE AWS CLI

Paso siguiente, se debe conectar a la instancia de EC2, para eso transfiero el certificado privado (como nano pin3.pem y le pego el texto del mismo, y debe tener permisos chmod 400 pin3.pem) sino dará error.

```
ec2-user@ip-172-31-94-34:~
root@pin3-aws:/home/gsmppin3/pin3# ssh -i pin.pem ec2-user@ec2-34-207-157-43.compute-1.amazonaws.com
#_
~\#### Amazon Linux 2023
~~\#####
~~\###|
~~\#| https://aws.amazon.com/linux/amazon-linux-2023
~~V~' '->
~~~
~~~.~.~
~~~/_m/' '->
[ec2-user@ip-172-31-94-34 ~]$
```

La instancia de EC2 no tiene git por defecto, lo instalo mediante:
yum install git

Ahora descargo el repositorio en la EC2:

git clone <https://github.com/gsmx64/MundosE-DevOps2303-PIN3.git>

```
[ec2-user@ip-172-31-94-34 ~]$ git clone https://github.com/gsmx64/MundosE-DevOps2303-PIN3.git
Cloning into 'MundosE-DevOps2303-PIN3'...
remote: Enumerating objects: 214, done.
remote: Counting objects: 100% (214/214), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 214 (delta 112), reused 161 (delta 64), pack-reused 0
Receiving objects: 100% (214/214), 3.66 MiB | 35.70 MiB/s, done.
Resolving deltas: 100% (112/112), done.
[ec2-user@ip-172-31-94-34 ~]$ ls
MundosE-DevOps2303-PIN3
[ec2-user@ip-172-31-94-34 ~]$
```


Ahora le asigno permisos de ejecución al run.sh que previamente armé y actualicé en el repositorio, esto me instalará todas las dependencias, configuraciones e iniciará el cluster...

```
01_ec2 02_eksctl 03_monitoring Readme.md img run.sh
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ chmod +x run.sh
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ ls
01_ec2 02_eksctl 03_monitoring Readme.md img run.sh
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ ./run.sh
```

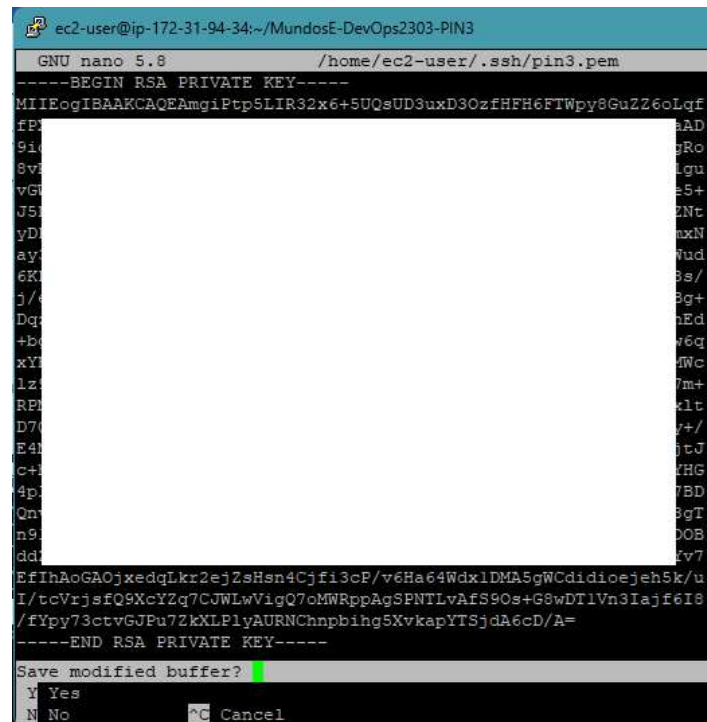
```
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ ./run.sh
Copying the env file.sample to .env
-----
> Please fill the .env file with your AWS credentials and settings:
-----
```

Le ingreso el AWS Account ID y salvo el archive .env :

```
ec2-user@ip-172-31-94-34:~/MundosE-DevOps2303-PIN3
GNU nano 5.8 .env
# AWS VARIABLES
AWS_REGION=us-east-1
AWS_ZONES=us-east-1a,us-east-1b,us-east-1c
AWS_ACCOUNT_ID=
# EC2 VARIABLES
AWS_EC2_AMI=ami-0440d3b780d96b29d
```

```
-----
> Please fill the .env file with your AWS credentials and settings:
-----
> Making the scripts executable.
> Runing EC2 user data scripts.
-----
> Initial setup
-----
> Updating the system.
Last metadata expiration check: 0:47:34 ago on Sat Feb 24 15:44:00 2024.
Dependencies resolved.
Nothing to do.
Complete!
> Installing the required openssl package.
Last metadata expiration check: 0:47:38 ago on Sat Feb 24 15:44:00 2024.
Error: No matching Packages to list
Last metadata expiration check: 0:47:39 ago on Sat Feb 24 15:44:00 2024.
Dependencies resolved.
Nothing to do.
Complete!
OpenSSL 3.0.8 7 Feb 2023 (Library: OpenSSL 3.0.8 7 Feb 2023)
```

En el siguiente paso ingreso la clave privada (borre parte en la captura por un tema de seguridad):



```

ec2-user@ip-172-31-94-34:~/MundosE-DevOps2303-PIN3
GNU nano 5.8 /home/ec2-user/.ssh/pin3.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEEgIBAAKCAQEAmgiPtp5LIR32x6+5UQsUD3uxD3OzfHfH6FTWpy8GuZZ6oLqf
fP...
9i...
8v...
vGT...
J5...
yD...
ay...
6K...
j/...
Dq...
+bo...
xY...
lz...
RPM...
D7...
E4...
c+...
4p...
Qm...
n9...
dd...
EfIhAoGAOjxedqLkr2ejZsHsn4Cjfi3cP/v6Ha64WdxlDMA5gWCdidioejeh5k/u
I/tcVrjsfQ9XcY2q7CJWLwVigQ7oMWRppAgSPNTLvAfs90s+G8wDT1Vn3Iajf6I8
/fYpy73ctvGJPu7ZkXLPlyAURNChnpbing5XvkapYTSjdA6cD/A=
-----END RSA PRIVATE KEY-----
Save modified buffer?
Y Yes
N No
^C Cancel

```

Y corre el script que actualiza el sistema, le da permisos correctos al certificado, actualiza AWS CLI, e instala: Kubectl, EKSctl, Docker y Docker Compose, Helm y Terraform.

```

> Script completed!
> AWS Version:
aws-cli/2.15.23 Python/3.11.6 Linux/6.1.77-99.164.amzn2023.x86_64 exe/x86_64.amz
n.2023 prompt/off
> Kubectl Version:
Client Version: v1.28.5-eks-5e0fdde
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
> EKSctl Version:
0.172.0
> Docker Version:
Docker version 24.0.5, build ced0996
> Docker Compose Version:
Docker Compose version v2.24.6
> Helm Version:
version.BuildInfo{Version:"v3.14.2", GitCommit:"c309b6f0ff63856811846cel8f3bdc93
d2b4d54b", GitTreeState:"clean", GoVersion:"go1.21.7"}
> Terraform Version:
Terraform v1.7.4
on linux_amd64

```

Y a continuación hace la creación del cluster.

CREACION DEL CLUSTER

Se ejecuta el script situado en “02_eksctl/create-cluster.sh” en el repositorio...

```
ec2-user@ip-172-31-94-34:~/MundosE-DevOps2303-PIN3
2024-02-24 18:35:42 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-addon-iamserviceaccount-kube-system-aws-node"
2024-02-24 18:36:12 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-addon-iamserviceaccount-kube-system-aws-node"
2024-02-24 18:36:12 [ ] serviceaccount "kube-system/aws-node" already exists
2024-02-24 18:36:12 [ ] updated serviceaccount "kube-system/aws-node"
2024-02-24 18:36:12 [ ] daemonset "kube-system/aws-node" restarted
2024-02-24 18:36:12 [ ] building managed nodegroup stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:36:13 [ ] deploying stack "eksctl-mundose-cluster-pin3-g19-nodegr
oup-standard-workers"
2024-02-24 18:36:13 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:36:43 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:37:38 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:38:32 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:39:57 [ ] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-nodegroup-standard-workers"
2024-02-24 18:39:57 [ ] waiting for the control plane to become ready
2024-02-24 18:39:58 [✓] saved kubeconfig as "/home/ec2-user/.kube/config"
2024-02-24 18:39:58 [ ] no tasks
2024-02-24 18:39:58 [✓] all EKS cluster resources for "mundose-cluster-pin3-g19
" have been created
2024-02-24 18:39:58 [ ] nodegroup "standard-workers" has 3 node(s)
2024-02-24 18:39:58 [ ] node "ip-192-168-58-111.ec2.internal" is ready
2024-02-24 18:39:58 [ ] node "ip-192-168-75-158.ec2.internal" is ready
2024-02-24 18:39:58 [ ] node "ip-192-168-9-183.ec2.internal" is ready
2024-02-24 18:39:58 [ ] waiting for at least 3 node(s) to become ready in "stan
dard-workers"
2024-02-24 18:39:58 [ ] nodegroup "standard-workers" has 3 node(s)
2024-02-24 18:39:58 [ ] node "ip-192-168-58-111.ec2.internal" is ready
2024-02-24 18:39:58 [ ] node "ip-192-168-75-158.ec2.internal" is ready
2024-02-24 18:39:58 [ ] node "ip-192-168-9-183.ec2.internal" is ready
2024-02-24 18:39:59 [ ] kubectl command should work with "/home/ec2-user/.kube/
config", try 'kubectl get nodes'
2024-02-24 18:39:59 [✓] EKS cluster "mundose-cluster-pin3-g19" in "us-east-1" r
egion is ready
```

Aquí se pueden ver los 3 workers listos:

Instances (4) Info

🔄

Connect

Instance state ▾

Actions ▾

Launch instance

🔍 Find Instance by attribute or tag (case-sensitive)

Instance state = running ✕

Clear filters

Any state ▾

< 1

<input type="checkbox"/>	Name 🔗 ▾	Instance ID	Instance state ▾	Instance type ▾	Status 📊
<input type="checkbox"/>	mundose-cluster-pin3-g19-standard-workers-Node	i-0543d26c9977e8ad0	🟢 Running 🔍 🔍	t2.medium	🟢 2/2 ch
<input type="checkbox"/>	mundose-cluster-pin3-g19-standard-workers-Node	i-0f219fd131a50428e	🟢 Running 🔍 🔍	t2.medium	🟢 2/2 ch
<input type="checkbox"/>	PIN3-EC2	i-0ef416dca47af6d6e	🟢 Running 🔍 🔍	t2.medium	🟢 2/2 ch
<input type="checkbox"/>	mundose-cluster-pin3-g19-standard-workers-Node	i-08436b2c7dea47381	🟢 Running 🔍 🔍	t2.medium	🟢 2/2 ch

Y dentro una vez creado el cluster se ejecuta el comando “kubectl apply -f \$PWD/02_eksctl/nginx-deployment.yaml” que levanta el nginx:


```

ec2-user@ip-172-31-94-34:~/MundosE-DevOps2303-PIN3
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-58-111.ec2.internal      Ready    <none>   46m   v1.28.5-eks-5e0fdde
ip-192-168-75-158.ec2.internal      Ready    <none>   46m   v1.28.5-eks-5e0fdde
ip-192-168-9-183.ec2.internal       Ready    <none>   46m   v1.28.5-eks-5e0fdde
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl cluster-info
Kubernetes control plane is running at https://2534F2ADDE11D95DA09B76693BD69CD7.gr7.us-east-1.eks.
CoreDNS is running at https://2534F2ADDE11D95DA09B76693BD69CD7.gr7.us-east-1.eks.amazonaws.com/api/
xy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl get service
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes ClusterIP   10.100.0.1     <none>         443/TCP     55m
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
nginx-deployment-7ffd9c9dbb-9mszl   1/1      Running   0            27m
nginx-deployment-7ffd9c9dbb-kgrfx   1/1      Running   0            27m
nginx-deployment-7ffd9c9dbb-nq8qn   1/1      Running   0            27m
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl get deployments
NAME      READY    UP-TO-DATE    AVAILABLE    AGE
nginx-deployment  3/3      3             3            28m
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl describe pod ^C
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl describe pod nginx-deployment
Name:      nginx-deployment-7ffd9c9dbb-9mszl
Namespace: default
Priority:   0
Service Account: default
Node:      ip-192-168-75-158.ec2.internal/192.168.75.158
Start Time: Sat, 24 Feb 2024 18:57:07 +0000
Labels:    app=nginx
           pod-template-hash=7ffd9c9dbb
Annotations: <none>
Status:    Running
IP:        192.168.84.186
IPs:
  IP:      192.168.84.186
Controlled By: ReplicaSet/nginx-deployment-7ffd9c9dbb
Containers:
  nginx:
    Container ID: containerd://cd5d85e3bddf539cc724d3789982f4e7a89941a07585476918ab4bf3b81db14c

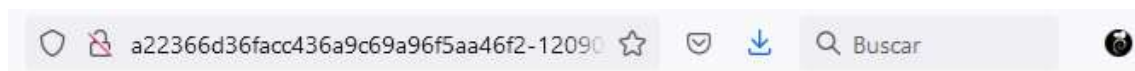
```

Ahora lo expongo con:

```
kubectl -n default patch svc nginx-deployment -p '{"spec": {"type": "LoadBalancer"}}'
```

Y obtengo el dominio para accederlo con:

```
kubectl -n default get svc nginx-deployment
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

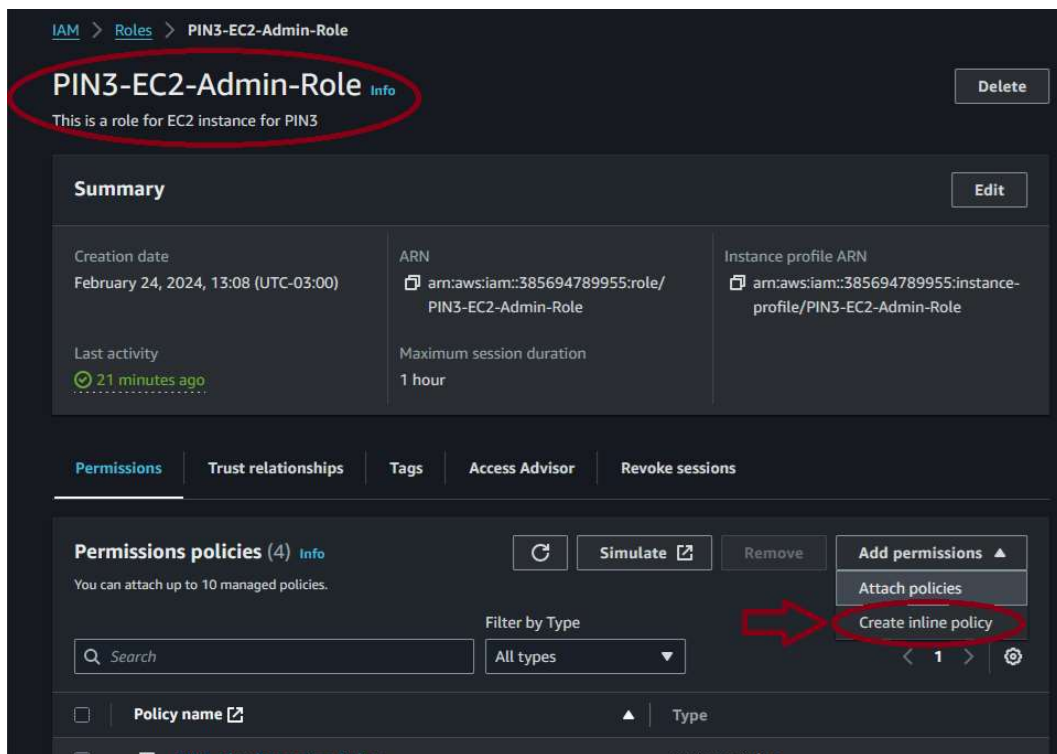
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Dado por completada esta parte, ahora sigue las herramientas de monitoreo.

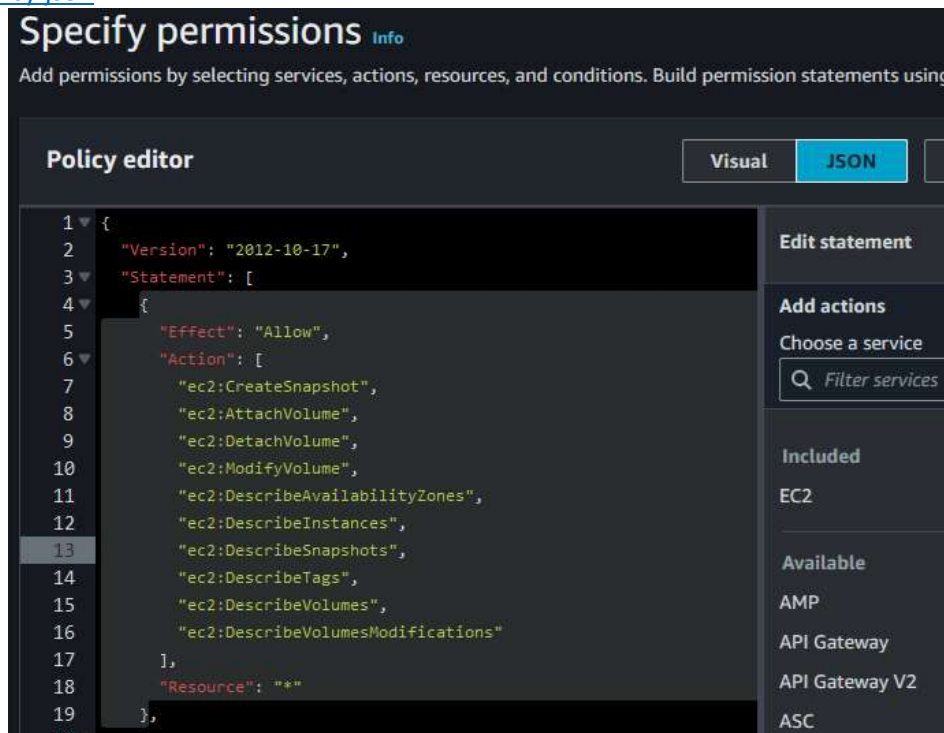
HERRAMIENTAS DE MONITOREO – DRIVER EBS

Para la instalación de este driver, nuevamente hay que ir a donde está la policy “PIN3-EC2-Admin-Role” que creé previamente:



Y dale click en “Create inline policy”. Luego agregar dentro el código de este enlace:

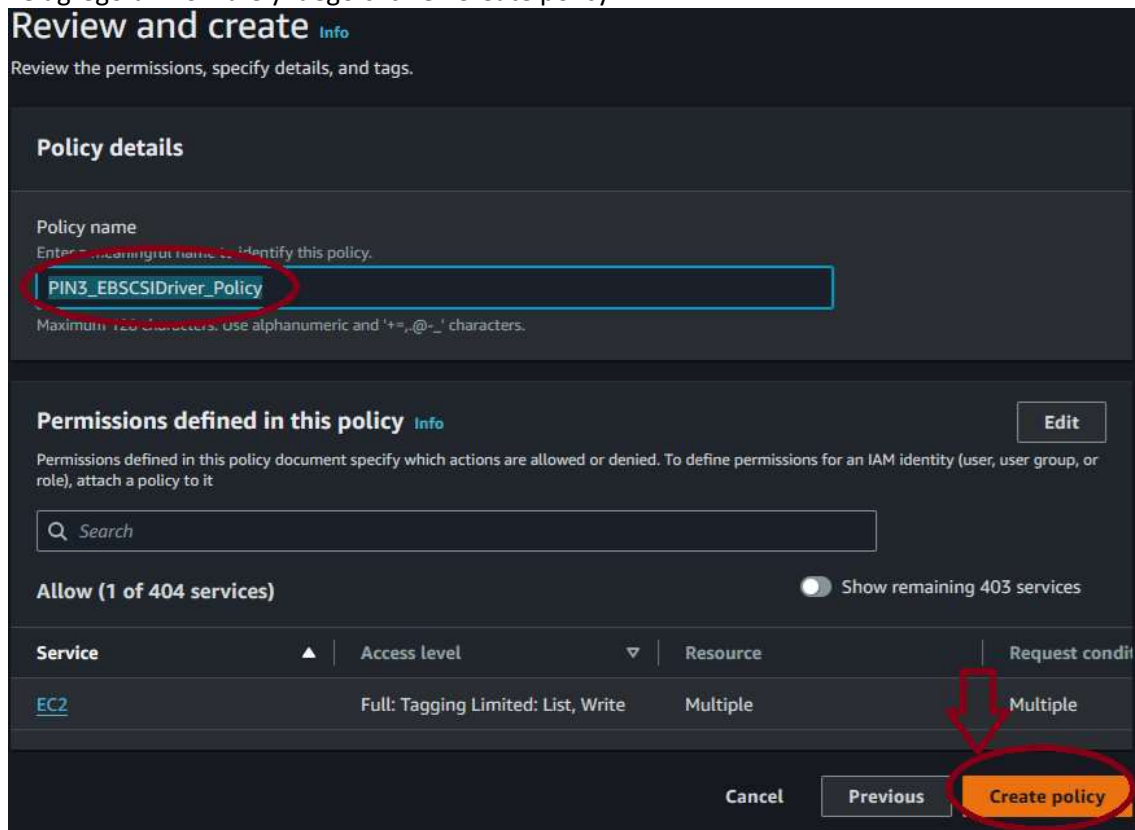
- <https://github.com/kubernetes-sigs/aws-ebs-csi-driver/blob/master/docs/example-iam-policy.json>



Puedo copiar y pegarlo, o bien hacer click en el enlace de github para descargarlo como raw y adjuntarlo como archivo externo:



Le agrego un nombre y luego click e "Create policy":



Ahora agrego el EBS Driver al repositorio de Helm:

```
helm repo add aws-ebs-csi-driver https://kubernetes-sigs.github.io/aws-ebs-csi-driver
```

```
helm repo update
```

A continuación, instalo el driver e su última versión:

```
helm upgrade --install aws-ebs-csi-driver \
  --namespace kube-system \
  aws-ebs-csi-driver/aws-ebs-csi-driver
```

Cuando ya esté realizado el deploy, verifico con:

```
kubectl get pods -n kube-system -l app.kubernetes.io/name=aws-ebs-csi-driver
```



```
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ helm repo add aws-ebs-csi-driver https://kubernetes-sigs.github.io/aws-ebs-csi-driver/
"aws-ebs-csi-driver" has been added to your repositories
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "aws-ebs-csi-driver" chart repository
Update Complete. 🎉Happy Helming!🎉
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ helm upgrade --install aws-ebs-csi-driver \
  --namespace kube-system \
  aws-ebs-csi-driver/aws-ebs-csi-driver
Release "aws-ebs-csi-driver" does not exist. Installing it now.
NAME: aws-ebs-csi-driver
LAST DEPLOYED: Sat Feb 24 22:29:37 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
NOTES:
To verify that aws-ebs-csi-driver has started, run:

    kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-ebs-csi-driver,app.kubernetes.io/component=controller"

NOTE: The [CSI Snapshotter] (https://github.com/kubernetes-csi/external-snapshotter) controller is a prerequisite of this chart and moving forward will be a prerequisite of using the snapshotting functionality.

WARNING: Upgrading the EBS CSI Driver Helm chart with --reuse-values will no longer be supported. See https://github.com/kubernetes-sigs/aws-ebs-csi-driver/issues/1864 for more details.
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$
```

Aquí ya está instalado:

```
Last login: Sat Feb 24 22:28:29 2024 from 186.182.26.45
[ec2-user@ip-172-31-94-34 ~]$ kubectl get pods -n kube-system -l app.kubernetes.io/name=aws-ebs-csi-driver
NAME                                READY   STATUS    RESTARTS   AGE
ebs-csi-controller-8649db944-28gtx  5/5     Running   0           5m22s
ebs-csi-controller-8649db944-knllj  5/5     Running   0           5m22s
ebs-csi-node-8p2df                  3/3     Running   0           5m22s
ebs-csi-node-gcml2                  3/3     Running   0           5m22s
ebs-csi-node-hzsk4                  3/3     Running   0           5m22s
[ec2-user@ip-172-31-94-34 ~]$
```

HERRAMIENTAS DE MONITOREO – PROMETHEUS

Se instala Prometheus con el script del repositorio:

```
BLE  NODE SELECTOR          AGE
daemonset.apps/prometheus-prometheus-node-exporter 3      3      0      3      0
      kubernetes.io/os=linux 38m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics 1/1     1            1           38m
deployment.apps/prometheus-prometheus-pushgateway 1/1     1            1           38m
deployment.apps/prometheus-server 0/1     1            0           38m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-5d5d748fcb 1         1         1       38m
replicaset.apps/prometheus-prometheus-pushgateway-8647d94cf6 1         1         1       38m
replicaset.apps/prometheus-server-66997ccd8b 1         1         0       38m

NAME                                READY   AGE
statefulset.apps/prometheus-alertmanager 0/1     38m
> Exposing Prometheus on the EC2 instance on port 9000 (Default: tcp/8080)
Error from server (NotFound): pods "prometheus-server" not found
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$
```

Y efectivamente, quedan en estado “Pending” el pod de prometheus-server y el de alertmanager:

```
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ kubectl get pods -n prometheus
```

NAME	READY	STATUS	RESTARTS	AGE
prometheus-alertmanager-0	0/1	Pending	0	41m
prometheus-kube-state-metrics-5d5d748fcb-9qj69	1/1	Running	0	41m
prometheus-prometheus-node-exporter-dr4ds	1/1	Running	0	41m
prometheus-prometheus-node-exporter-kwf86	1/1	Running	0	41m
prometheus-prometheus-node-exporter-mwsjc	1/1	Running	0	41m
prometheus-prometheus-pushgateway-8647d94cf6-tlmbt	1/1	Running	0	41m
prometheus-server-66997ccd8b-4w7wk	0/2	Pending	0	41m

```
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$
```

Entonces se ejecuta el comando sugerido, modificándolo con las variables de entorno, y ya se agrega en el archivo de configuración del driver, para que en el próximo despliegue no vuelva a dar el error:

Código:

```
eksctl create iamserviceaccount \
  --region $AWS_REGION \
  --name ebs-csi-controller-sa \
  --namespace kube-system \
  --cluster $EKSCTL_CLUSTER_NAME \
  --attach-policy-arn arn:aws:iam::aws:policy/service-
role/AmazonEBSCSIDriverPolicy \
  --approve \
  --role-only \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

```
[ec2-user@ip-172-31-94-34 MundosE-DevOps2303-PIN3]$ eksctl create iamserviceaccount \
  --region $AWS_REGION \
  --name ebs-csi-controller-sa \
  --namespace kube-system \
  --cluster $EKSCTL_CLUSTER_NAME \
  --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --approve \
  --role-only \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

```
2024-02-25 02:05:05 [!] 1 existing iamserviceaccount(s) (kube-system/aws-node)
will be excluded
2024-02-25 02:05:05 [!] 1 iamserviceaccount (kube-system/ebs-csi-controller-sa)
was included (based on the include/exclude rules)
2024-02-25 02:05:05 [!] serviceaccounts in Kubernetes will not be created or mo
dified, since the option --role-only is used
2024-02-25 02:05:05 [!] 1 task: { create IAM role for serviceaccount "kube-syst
em/ebs-csi-controller-sa" }
2024-02-25 02:05:05 [!] building iamserviceaccount stack "eksctl-mundose-cluste
r-pin3-g19-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-02-25 02:05:06 [!] deploying stack "eksctl-mundose-cluster-pin3-g19-addon-
iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-02-25 02:05:06 [!] waiting for CloudFormation stack "eksctl-mundose-cluste
r-pin3-g19-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
```

Pero siguen sin levantar prometheus-server y alertmanager, estimo que el driver no está operativo, con lo cual con el comando:

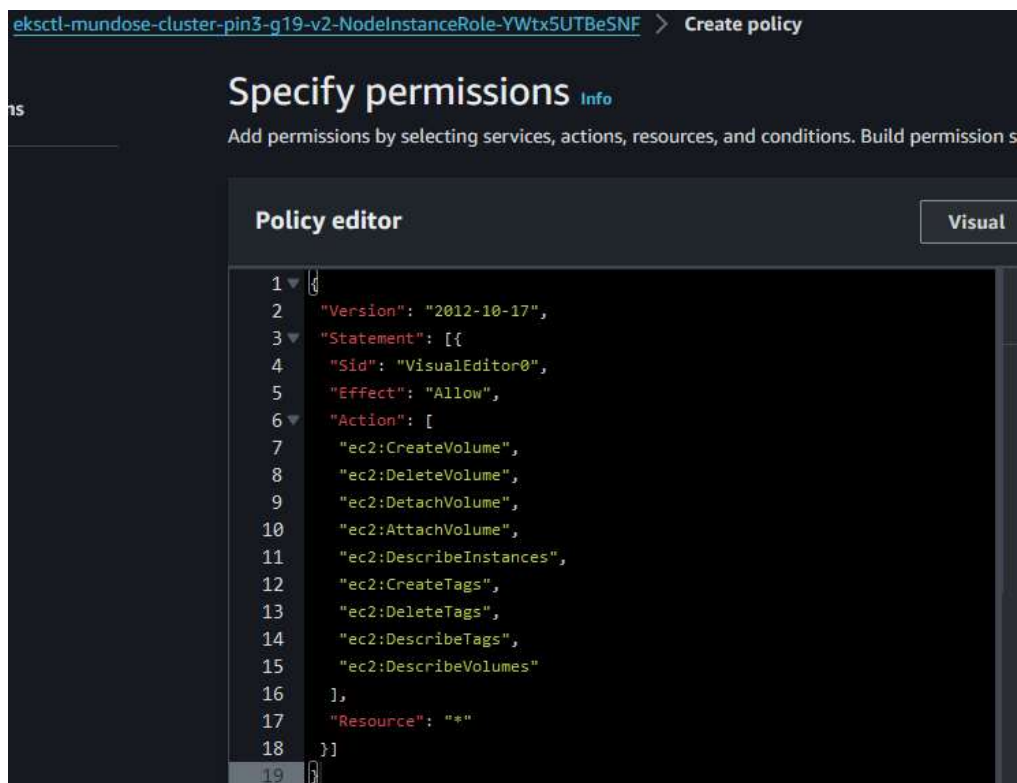
```
>>> kubectl describe pvc
```

Y me tira varios errores similares a este (ver ebs_error_log.log en repositorio, carpeta 03_monitoring):

Warning ProvisioningFailed 7m41s (x6 over 27m) ebs.csi.aws.com_ebs-csi-controller-cc4db67f7-9npxm_866150cc-f575-4300-aeb9-8230a0294172 (combined from similar events): failed to provision volume with StorageClass "aws-ebs": rpc error: code = Internal desc = Could not create volume "pvc-41a89e99-97e5-47b3-801c-bcfddd056de2": could not create volume in EC2: UnauthorizedOperation: You are not authorized to perform this operation. User: arn:aws:sts::385694789955:assumed-role/eksctl-mundose-cluster-pin3-g19-v2-NodeInstanceRole-YWtx5UTBeSNF/i-0a3571a9e38be2f9e is not authorized to perform: ec2:CreateVolume on resource: arn:aws:ec2:us-east-1:385694789955:volume/* because no identity-based policy allows the ec2:CreateVolume action. Encoded authorization failure message:

Ahí veo que falta agregar una política, pero al nodo, esto no estaba en los pasos de instalación del repositorio oficial del AWS EBS CSI Driver, investigo y encuentro las políticas faltantes y creo una nueva política al rol del nodo eksctl-mundose-cluster-pin3-g19-v2-NodeInstanceRole-YWtx5UTBeSNF:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateVolume",
      "ec2>DeleteVolume",
      "ec2:DetachVolume",
      "ec2:AttachVolume",
      "ec2:DescribeInstances",
      "ec2:CreateTags",
      "ec2>DeleteTags",
      "ec2:DescribeTags",
      "ec2:DescribeVolumes"
    ],
    "Resource": "*"
  }]
}
```



Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+,.,@,-,_' characters.

Permissions defined in this policy [Info](#) [Edit](#)

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 404 services) ☐ Show remaining 403 services

Service	Access level	Resource	Request condition
EC2	Full: Tagging Limited: List, Write	All resources	None

[Cancel](#) [Previous](#) [Create policy](#)

Y finalmente todo queda operativo...

```
[ec2-user@ip-172-31-85-117 ~]$ kubectl describe pvc
Name:          pvc
Namespace:     default
StorageClass:  aws-ebs
Status:        Bound
Volume:        pvc-41a89e99-97e5-47b3-801c-bcfddd056de2
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner: ebs.csi.aws.com
               volume.kubernetes.io/storage-provisioner: ebs.csi.aws.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      2Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Used By:       <none>
Events:        <none>
[ec2-user@ip-172-31-85-117 ~]$
```

```
[ec2-user@ip-172-31-85-117 ~]$ kubectl get pods -n prometheus
NAME                                READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0          1/1     Running   0           11h
prometheus-kube-state-metrics-5d5d748fcb-gkvpm  1/1     Running   0           11h
prometheus-prometheus-node-exporter-js8tz      1/1     Running   0           11h
prometheus-prometheus-node-exporter-nbdkz      1/1     Running   0           11h
prometheus-prometheus-node-exporter-z899b      1/1     Running   0           11h
prometheus-prometheus-pushgateway-8647d94cf6-gckg9  1/1     Running   0           11h
prometheus-server-66997ccd8b-gl4lg           2/2     Running   0           11h
```

Patcheo el servicio a LoadBalancer:

```
kubectl -n prometheus patch svc prometheus-server -p '{"spec": {"type": "LoadBalancer"}}'
```

Y con este commando obtengo la url para ingresar:

```
kubectl -n prometheus get svc prometheus-server
```


← → ↻ ae2bb410ec66e4c888ae18c365660eb0-1406583938.us-east-1.elb.amazonaws.com/graph

Prometheus Alerts Graph Status ▾ Help

☐ Use local time ☐ Enable query history ☒ Enable autocomplete ☒ Enable highlighting ☒ Enable linter

🔍 Expression (press Shift+Enter for newlines)

Table **Graph**

< Evaluation time >

No data queried yet

Add Panel

← → ↻ ae2bb410ec66e4c888ae18c365660eb0-1406583938.us-east-1.elb.amazonaws.com/graph

Prometheus Alerts Graph Status ▾ Help

Targets

All scrape pools ▾ All Unhealthy Collapse All 🔍 Filter by endpoint or labels ☒ Unknown ☒ Unhealthy

kubernetes-apiservers (2/2 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.96.181/metrics	UP	instance="192.168.96.181:443" job="kubernetes-apiservers" ▾	8.369s ago	155.205ms	
https://192.168.131.85/metrics	UP	instance="192.168.131.85:443" job="kubernetes-apiservers" ▾	56.231s ago	117.990ms	

kubernetes-nodes (3/3 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/ip-192-168-0-42.ec2.internal/proxy/metrics	UP	alpha_eksctl.io_cluster_name="mundose-cluster-pin3-g19-v2" alpha_eksctl.io_nodegroup_name="standard-workers" beta_kubernetes.io_arch="amd64" beta_kubernetes.io_instance_type="t2.medium" beta_kubernetes.io_os="linux"	28.388s ago	52.712ms	

HERRAMIENTAS DE MONITOREO – GRAFANA

Se instala Grafana con el script del repositorio.

```
[ec2-user@ip-172-31-85-117 ~]$ kubectl get pods -n grafana
NAME                                READY   STATUS    RESTARTS   AGE
grafana-848f88c944-sjn9d            1/1     Running   0           26h

[ec2-user@ip-172-31-85-117 ~]$ kubectl get all -n grafana
NAME                                READY   STATUS    RESTARTS   AGE
pod/grafana-848f88c944-sjn9d        1/1     Running   0           26h

NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/grafana                     LoadBalancer       10.100.55.5      af904788c877947fdalc84898253525d-1112927394.us-east-1.elb.amazonaws.com  80:32727/TCP      26h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana              1/1     1             1           26h

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-848f88c944  1         1         1       26h
[ec2-user@ip-172-31-85-117 ~]$
```

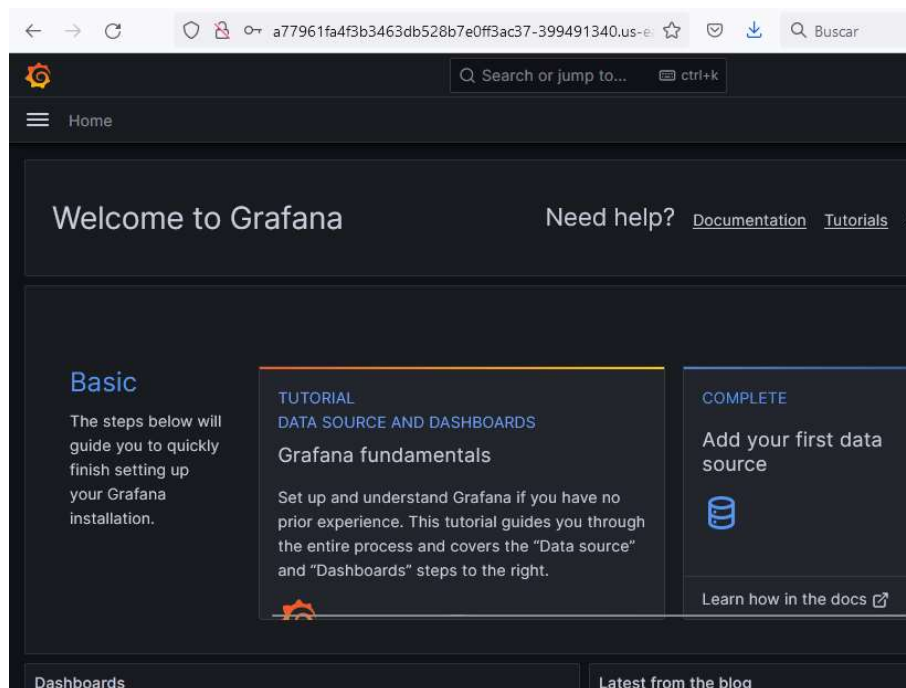
Patcheo el servicio a LoadBalancer:

```
kubectl -n grafana patch svc grafana -p '{"spec": {"type": "LoadBalancer"}}'
```

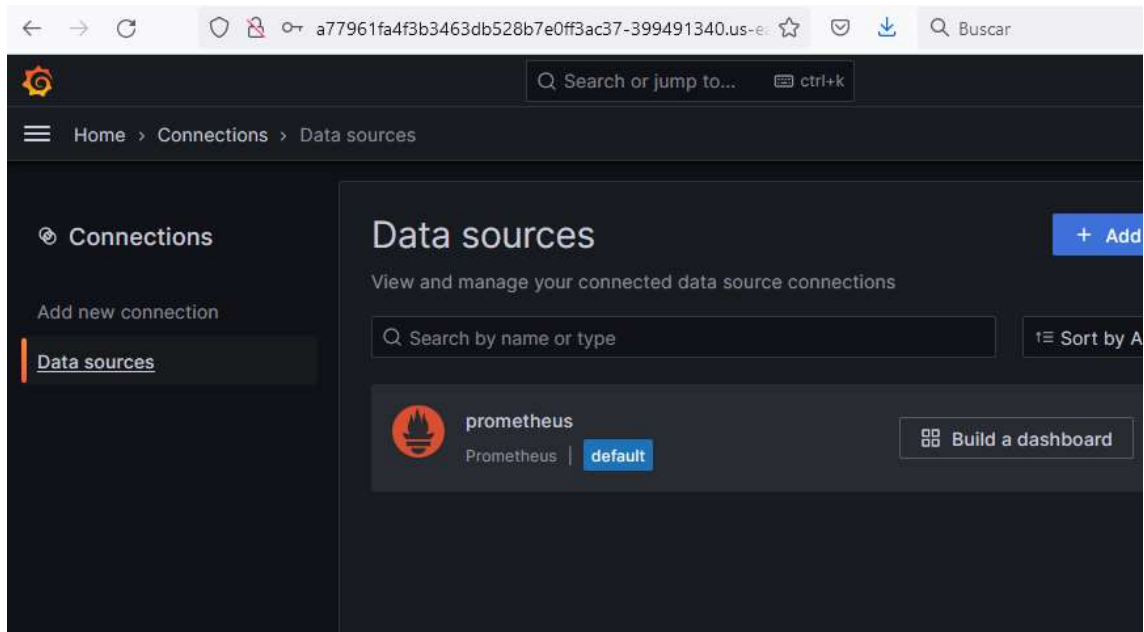
Y con este commando obtengo la url para ingresar:

```
kubectl -n grafana get svc grafana
```

```
[ec2-user@ip-172-31-85-117 ~]$ GRAFANA_PUBLIC_DOMAIN=$(kubectl -n grafana get svc grafana | awk '{print $4}' | grep -v 'EXTERNAL-IP')
[ec2-user@ip-172-31-85-117 ~]$ echo $GRAFANA_PUBLIC_DOMAIN
af904788c877947fdalc84898253525d-1112927394.us-east-1.elb.amazonaws.com
[ec2-user@ip-172-31-85-117 ~]$
```



Ahora procedo a configurar los datasources de Prometheus y de AlertManager:



a77961fa4f3b3463db528b7e0ff3ac37-399491340.us-east-1.elb.amazonaws.com

Search or jump to... **ctrl+k**

data sources > prometheus

Name ⓘ prometheus Default ☒

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with * are required

Connection

Prometheus server URL * ⓘ 5660eb0-1406583938.us-east-1.elb.amazonaws.com/

Authentication

Authentication methods

Choose an authentication method to access the data source

No Authentication ▼

✓ Successfully queried the Prometheus API.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

Siguiendo con el dashboard 3119:

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from [Grafana.com](#)

Published by Jfo Org

Updated on 2017-09-08 12:22:08

Options

Name

Kubernetes cluster monitoring (via Prometheus)

Folder

Dashboards

Unique Identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

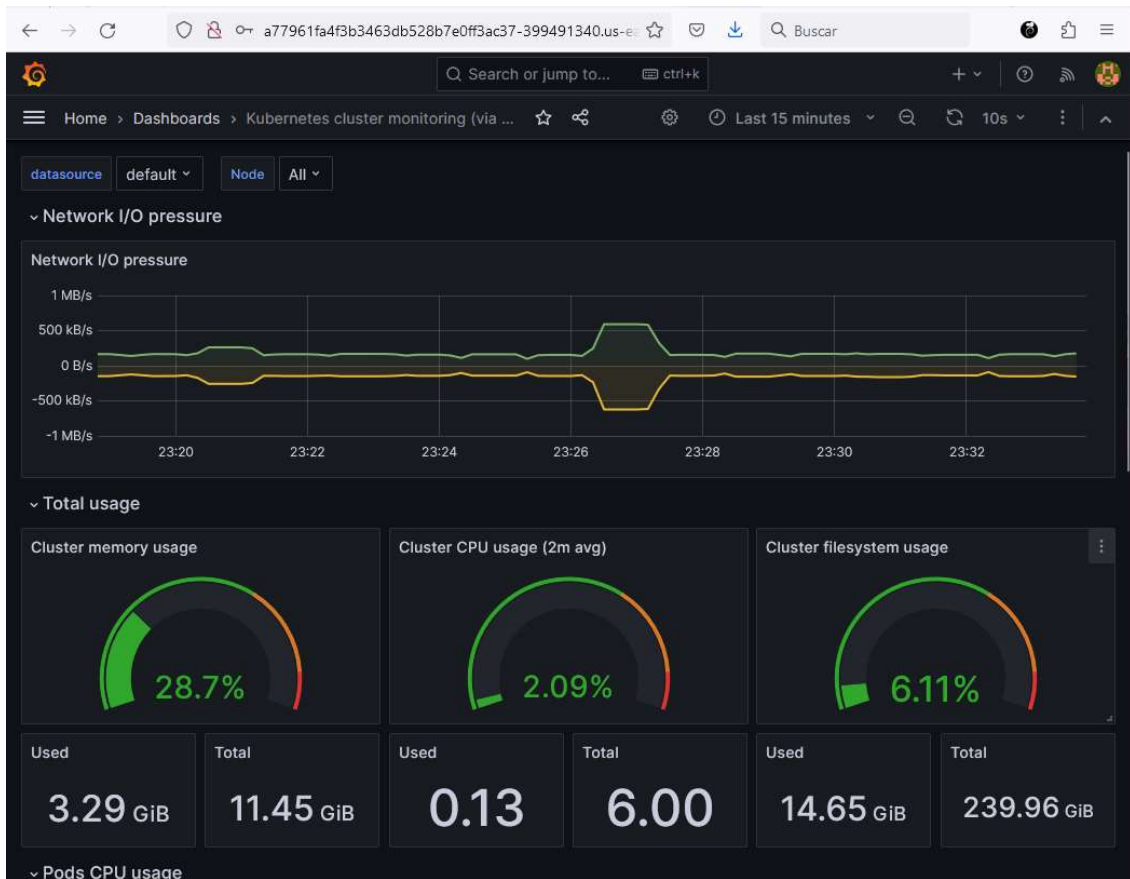
Change uid

prometheus

prometheus

Import Cancel

Quedando operative de forma correcta, levanta sus datos desde Prometheus directamente.



Y finalizando con el dashboard 6417:

The screenshot shows the 'Import dashboard' form in Grafana. The form is titled 'Import dashboard' and includes the following fields and options:

- Import dashboard from file or Grafana.com:** A dropdown menu set to 'Grafana.com'.
- Published by:** sekka1
- Updated on:** 2018-06-06 20:51:56
- Options:**
 - Name:** Kubernetes Cluster (Prometheus)
 - Folder:** Dashboards
 - Unique identifier (UID):** 4XuMd2liz (with a 'Change uid' button)
 - prometheus:** A dropdown menu with 'prometheus' selected.

Quedando igual al PDF de los requerimientos del proyecto final integrador.

