Proyecto Devops Integrador v2

Intro

Este proyecto tiene como idea principal el aprendizaje y poner en práctica lo aprendido a través de un laboratorio que permitirá integrar diferentes herramientas y tecnologías.

Nos centraremos en la primera parte en crear una instancia de **EC2** en AWS para poder desde allí realizar todas las tareas necesarias. Luego comenzaremos con el despliegue del Cluster de Kubernetes que tiene dos opciones **terraform** o **ekscli.** Una vez configurado el cluster integraremos el mismo con **Azure Devops** y desplegaremos un contenedor de **nginx**. Existe un capítulo opcional para configurar **Route 53**, **dns**, **certificados** y darle una url amigable a nuestro sitio, si deciden no realizarlo les quedará como referencia para proyectos futuros y personales.

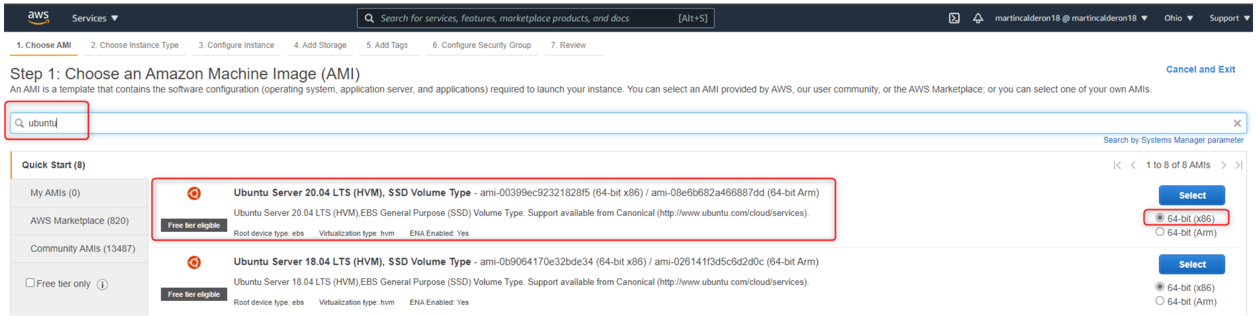
En la segunda parte, configuraremos monitoreo con el stack de **Elastic**, **FluentBit** y **Kibana** y luego desplegaremos **Grafana** y **Prometheus**.

Este será el repositorio de github de referencia para todo el proyecto: Proyecto Integrador Repo

Crear y configurar Máquina EC2

Caracteristicas

Region: **us-east-2**
Sistema Operativo **: Ubuntu Server 20.04**
Family (Tipo): **t2.small**



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

ubuntu

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-00399ec92321828f5 (64-bit x86) / ami-08e6b682a466887dd (64-bit Arm)
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

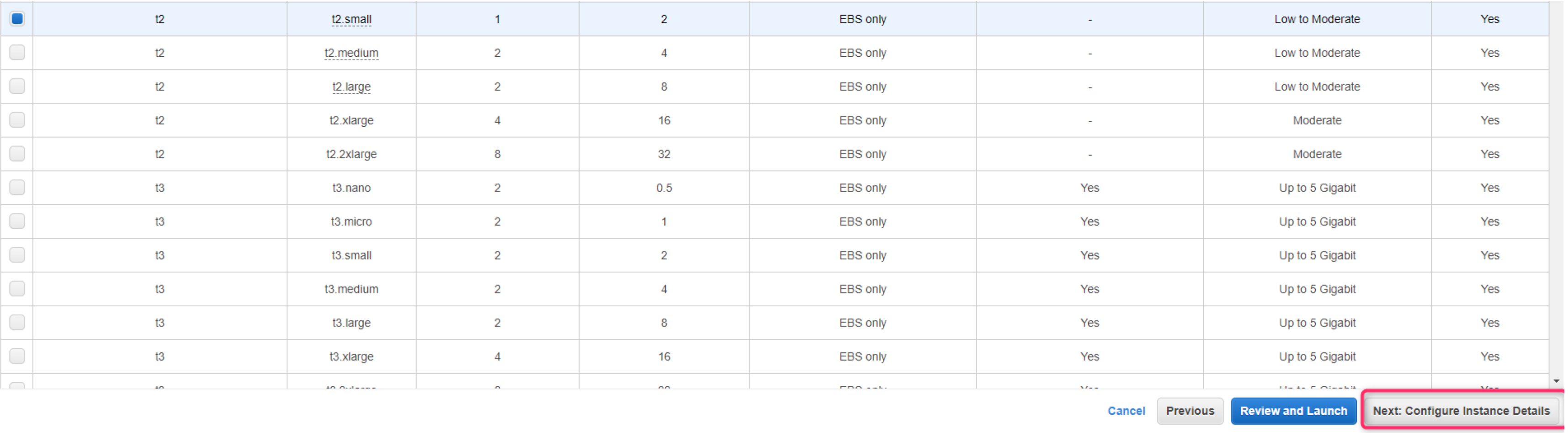
● 64-bit (x86)
○ 64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b9064170e32bde34 (64-bit x86) / ami-026141f3d5c6d2d0c (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Nota: Las instancias de tipo t2.small tiene un cargo. Es necesario eliminarlas inmediatamente luego de terminar el proyecto o si ha decidido no continuar con el por un periodo de tiempo

Configure Instance Details



☑	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes

User Data:
Aquí vamos a pasar un script the Bash que realizará las siguientes actividades en la instancia

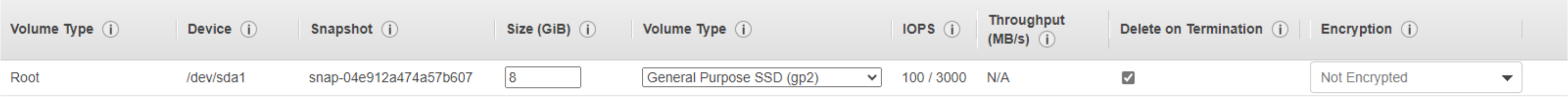
- Instalar unzip

- Descargar AWS CLI
- Instalar AWS CLI
- Instalamos eksctl - CLI for Amazon EKS
- Instalar Docker
- Instalar las herramientas para Kubernetes
    - **kubeadm**: needed for low level node administration (Referencia pero no la instalamos)
    - **kubelet**: low level kubernetes bootstrapper (Referencia pero no la instalamos)
    - **kubectl**: user interface for Kubernetes (Si la instalamos)
- Instalar HELM
- Configurar grupos y permisos
- Instalar Terraform

Pueden obtener el script en el siguiente repositorio ec2_user_data repo

Storage (Almacenamiento):

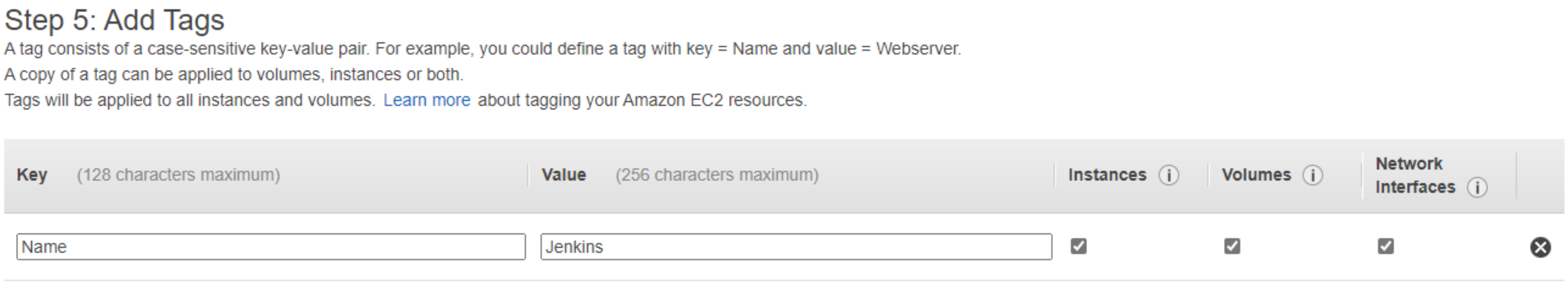
8GB General Purpose SSD (gp2)



Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-04e912a474a57b607	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Tag

Name : Jenkins



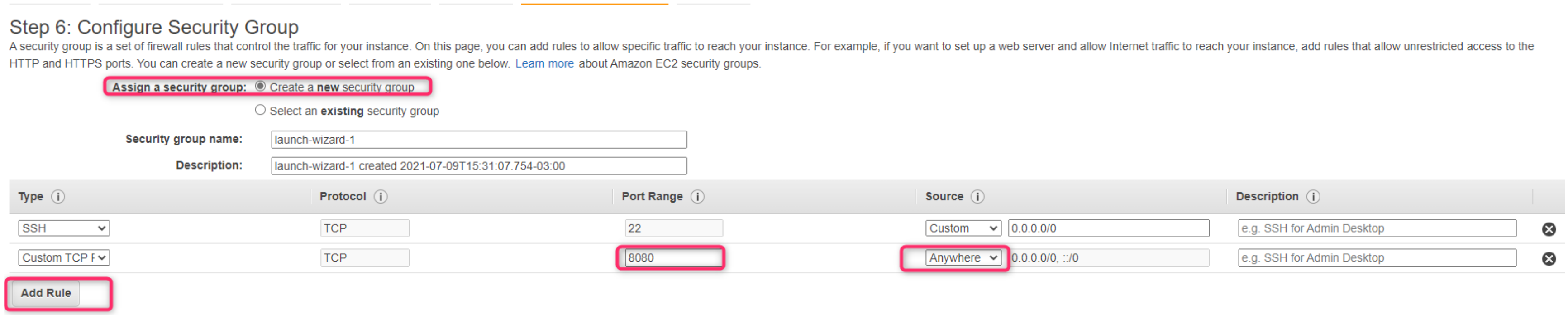
Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	Network Interfaces
Name	Jenkins	☑	☑	☑

Configure Security Group

Con un nuevo grupo de seguridad que habilite el acceso por el puerto 22 desde cualquier red y otra regla que habilite el acceso al puerto 8080 desde cualquier lugar.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

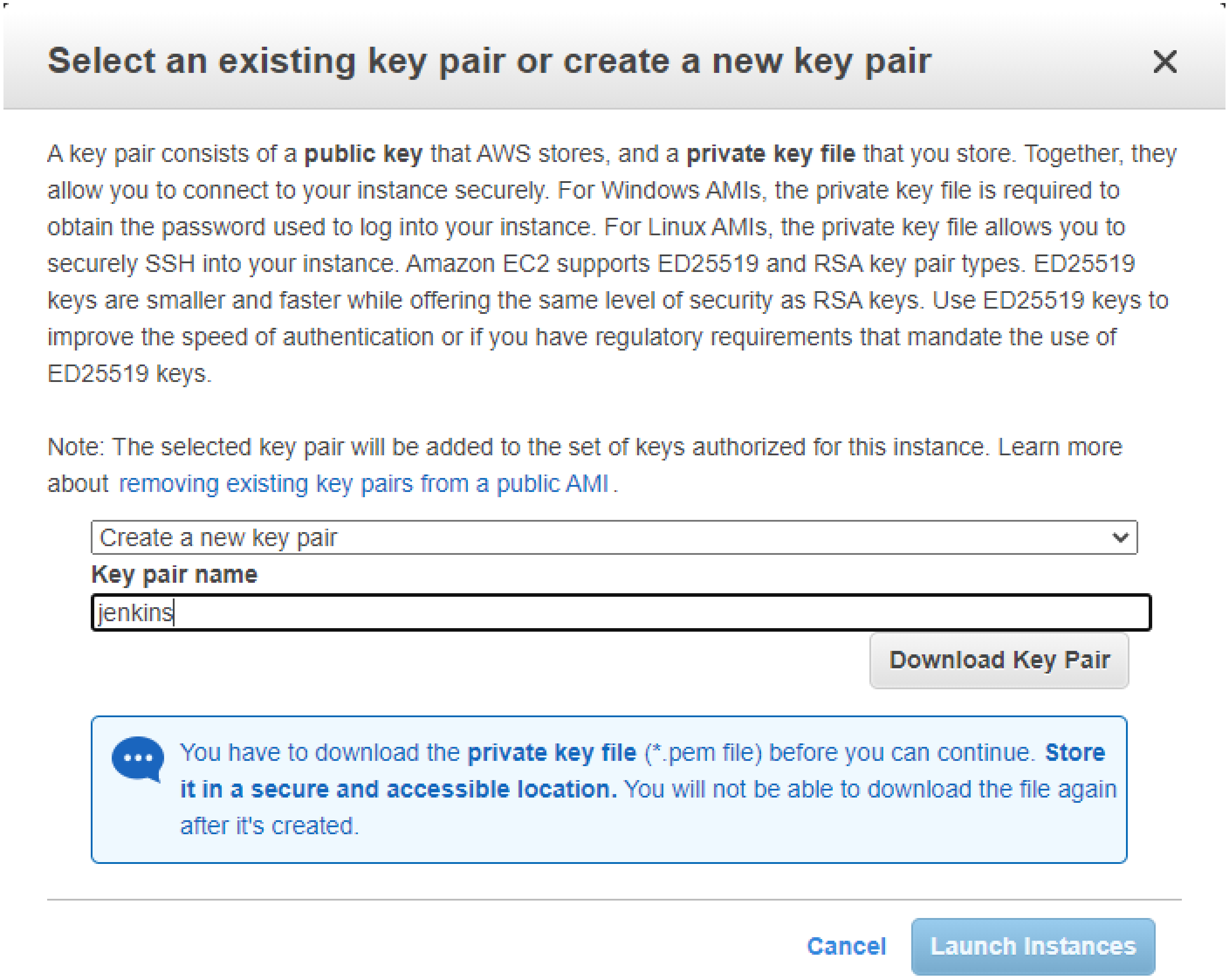
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-1
Description: launch-wizard-1 created 2021-07-09T15:31:07.754-03:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Crear Key Pair

Al Lanzar la creación de la instancia nos abrira el menu para crear una llave de ssh , ingresemos el nombre de **jenkins**



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types. ED25519 keys are smaller and faster while offering the same level of security as RSA keys. Use ED25519 keys to improve the speed of authentication or if you have regulatory requirements that mandate the use of ED25519 keys.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

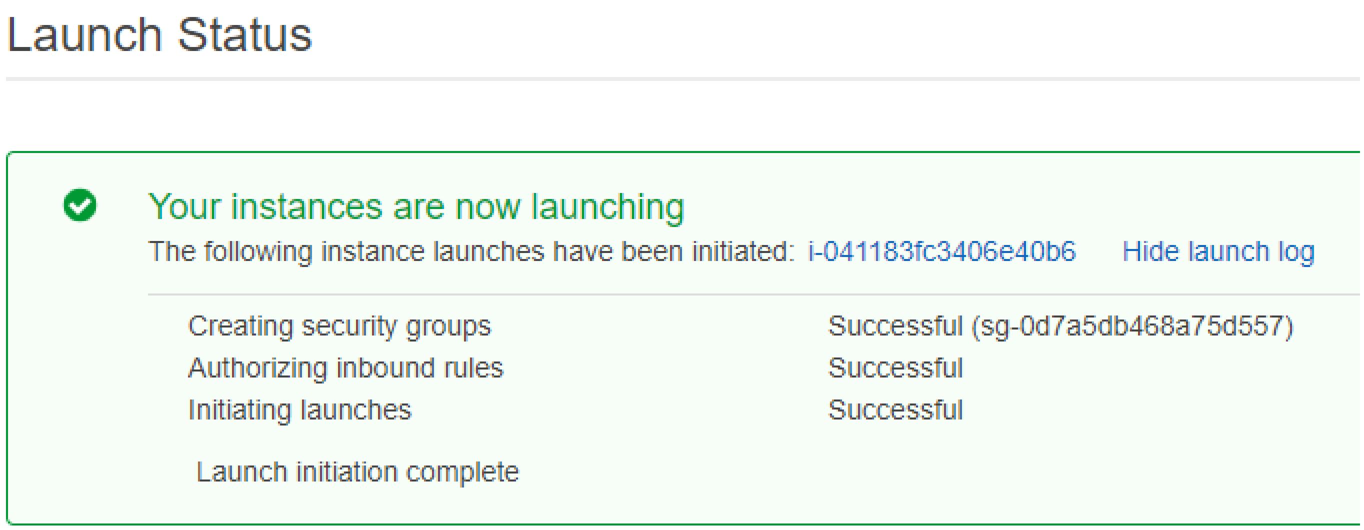
Create a new key pair

Key pair name

jenkins

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Seguir el progreso de la creación



Launch Status

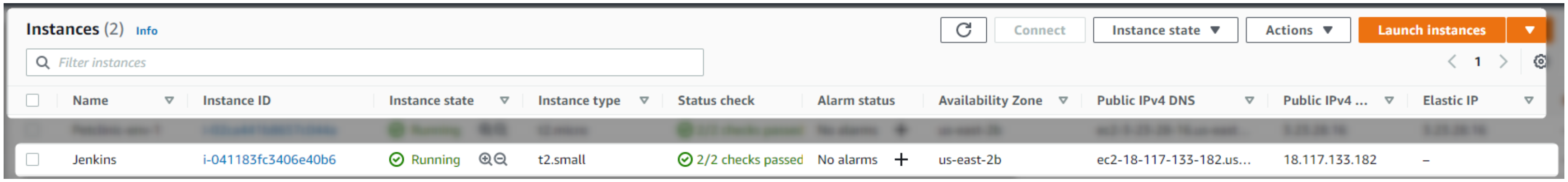
✓ Your instances are now launching

The following instance launches have been initiated: i-041183fc3406e40b6   Hide launch log

Creating security groups	Successful (sg-0d7a5db468a75d557)
Authorizing inbound rules	Successful
Initiating launches	Successful

Launch initiation complete

Una vez finalizado deberían ver la instancia corriendo en la consola de EC2

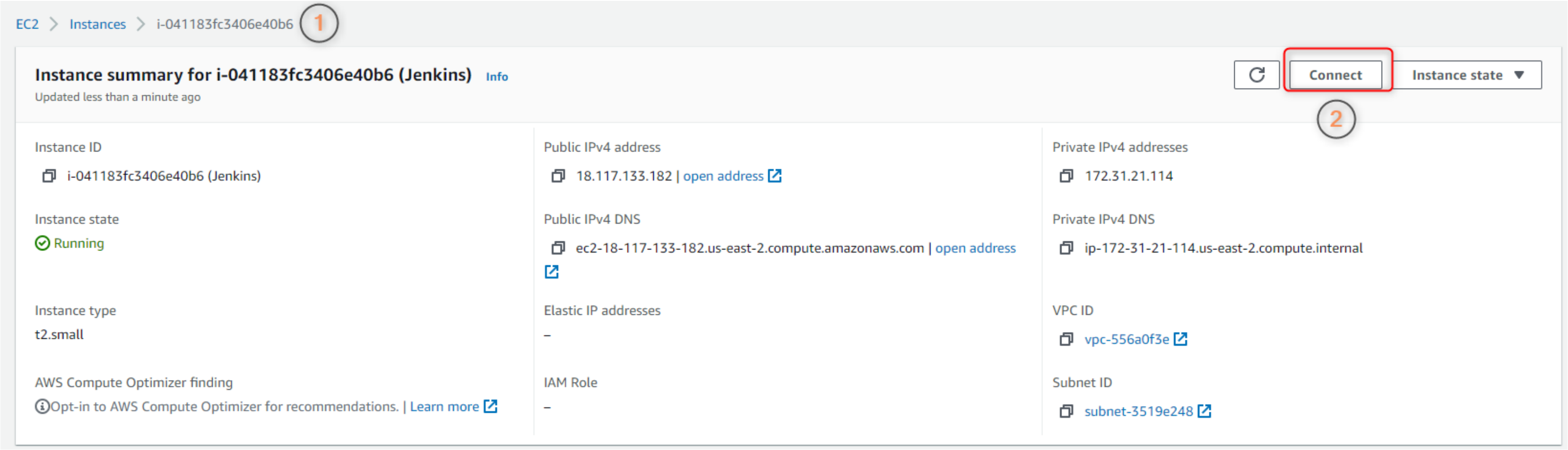


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
□	Jenkins	i-041183fc3406e40b6	✓ Running	t2.small	✓ 2/2 checks passed	No alarms +	us-east-2b	ec2-18-117-133-182.us...	18.117.133.182	–

Conectarse a una máquina EC2

Conectarse a la instancia:

Clic en el Link del instance ID , esto abre las configuraciones.



Instance summary for i-041183fc3406e40b6 (Jenkins) Info

Updated less than a minute ago

Instance ID
i-041183fc3406e40b6 (Jenkins)

Instance state
Running

Instance type
t2.small

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Public IPv4 address
18.117.133.182 | open address

Public IPv4 DNS
ec2-18-117-133-182.us-east-2.compute.amazonaws.com | open address

Elastic IP addresses
–

IAM Role
–

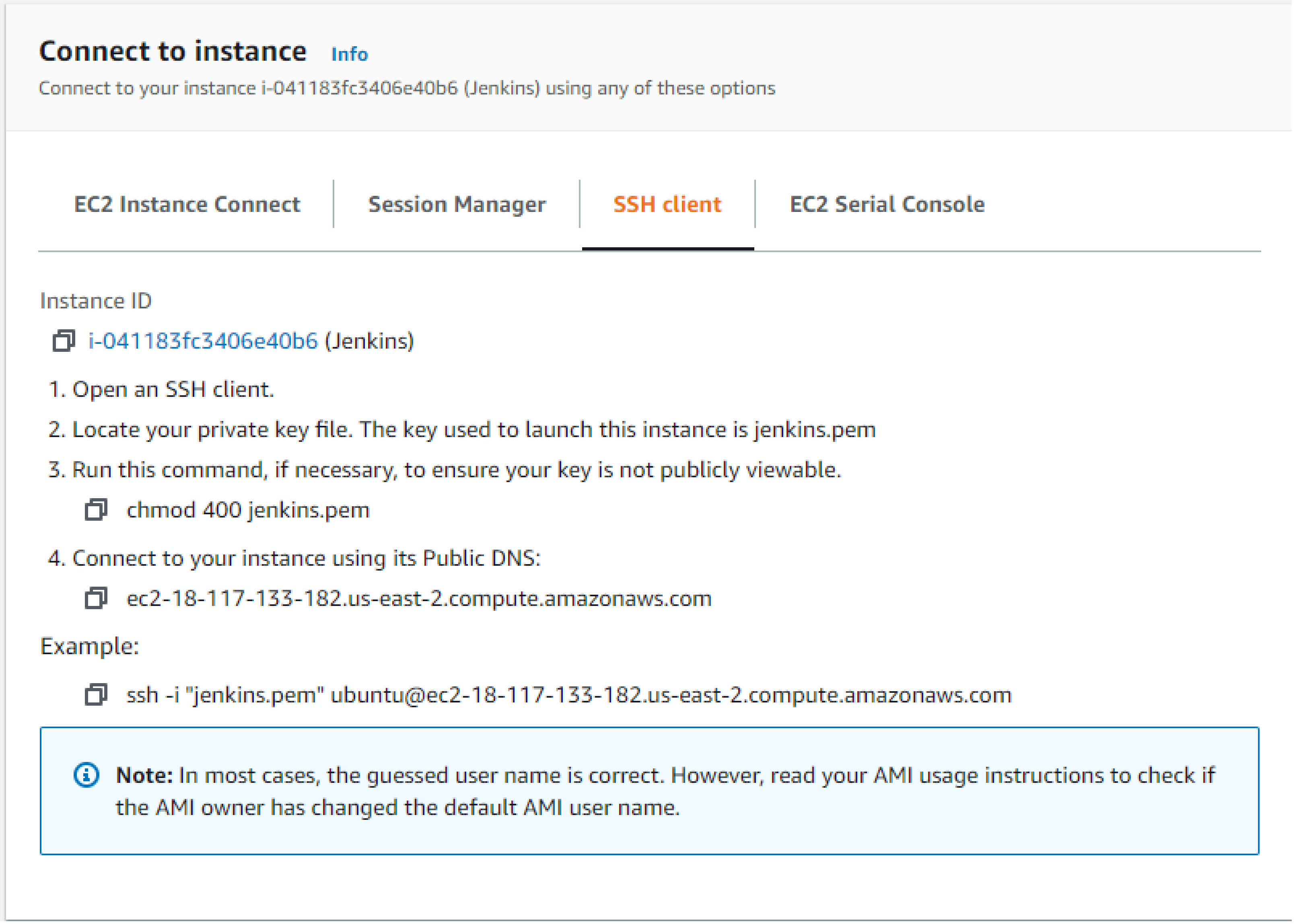
Private IPv4 addresses
172.31.21.114

Private IPv4 DNS
ip-172-31-21-114.us-east-2.compute.internal

VPC ID
vpc-556a0f3e

Subnet ID
subnet-3519e248

Seguir las instrucciones para asignar los permisos correspondientes al archivo .PEM y conectarse a la instancia EC2



Instance ID

i-041183fc3406e40b6 (Jenkins)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is jenkins.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 jenkins.pem
```

4. Connect to your instance using its Public DNS:

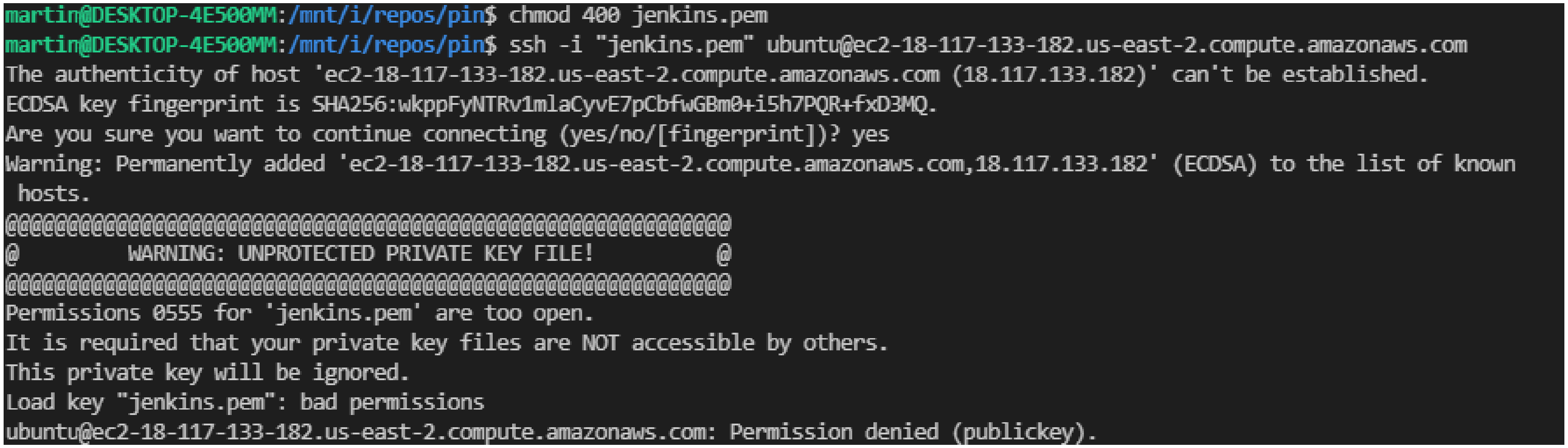
```
ec2-18-117-133-182.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "jenkins.pem" ubuntu@ec2-18-117-133-182.us-east-2.compute.amazonaws.com
```

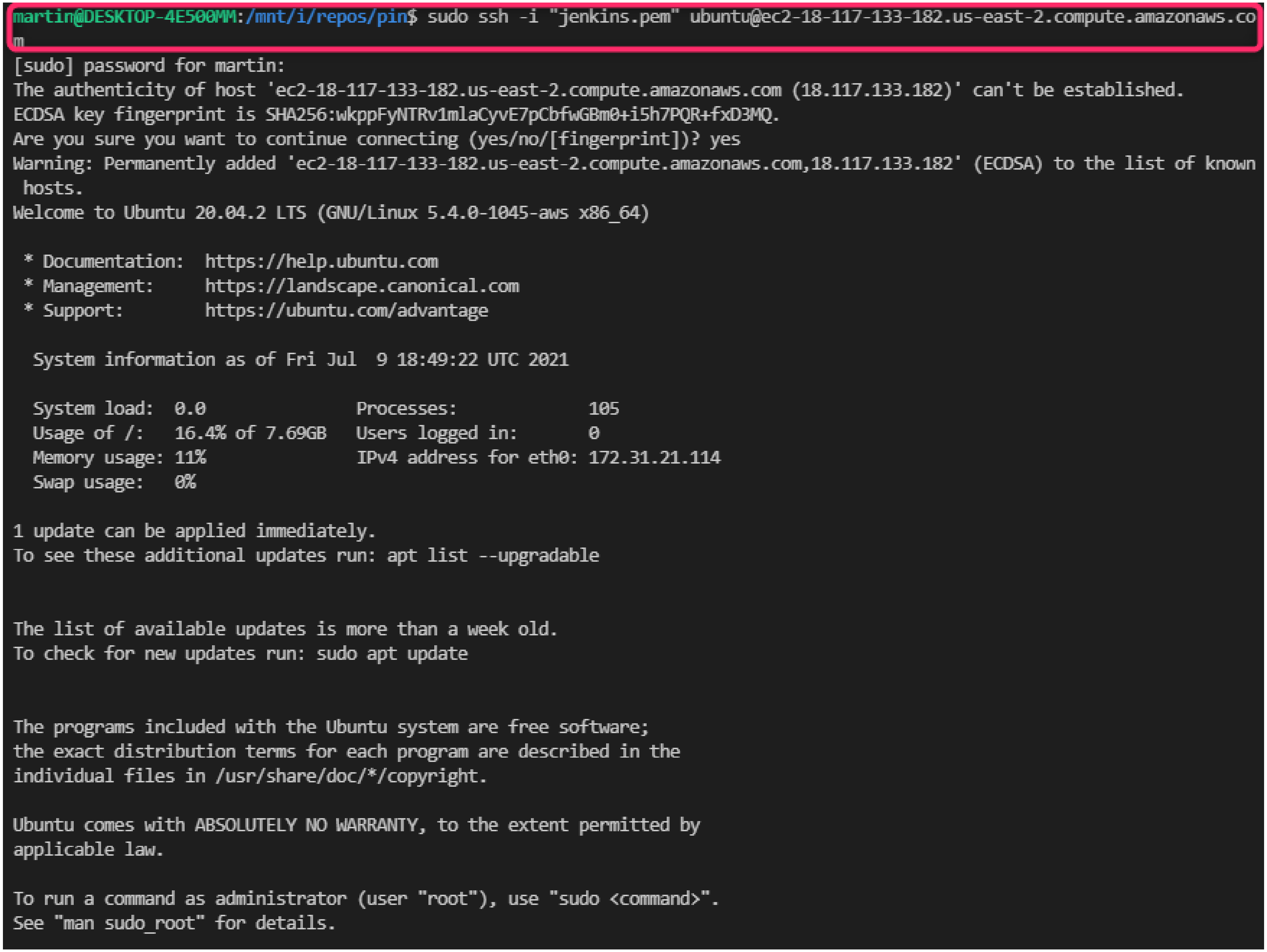
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Nota: Es posible que si están utilizando WSL en Windows (Ubuntu desde Windows), incluso luego de cambiar los permisos arroje un error



```
martin@DESKTOP-4E500MM:/mnt/i/repos/pin$ chmod 400 jenkins.pem
martin@DESKTOP-4E500MM:/mnt/i/repos/pin$ ssh -i "jenkins.pem" ubuntu@ec2-18-117-133-182.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-117-133-182.us-east-2.compute.amazonaws.com (18.117.133.182)' can't be established.
ECDSA key fingerprint is SHA256:wkppFyNTRv1mlaCyvE7pCbfwGBm0+i5h7PQR+fxD3MQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-117-133-182.us-east-2.compute.amazonaws.com,18.117.133.182' (ECDSA) to the list of known
 hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0555 for 'jenkins.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "jenkins.pem": bad permissions
ubuntu@ec2-18-117-133-182.us-east-2.compute.amazonaws.com: Permission denied (publickey).
```

Se soluciona corriendo la conexión SSH con Sudo



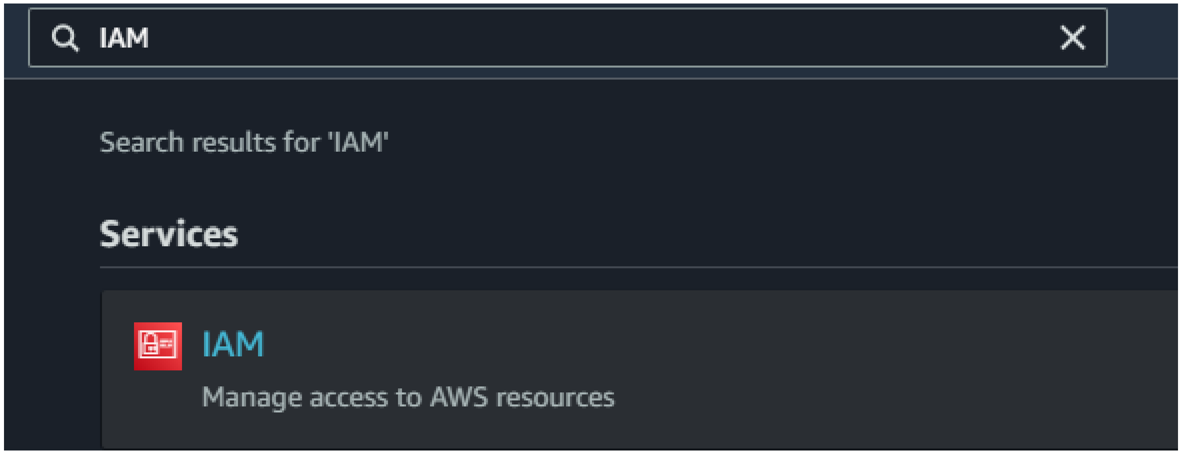
```
martin@DESKTOP-4E500MM:/mnt/i/repos/pin$ sudo ssh -i "jenkins.pem" ubuntu@ec2-18-117-133-182.us-east-2.compute.amazonaws.co
m
[sudo] password for martin:
The authenticity of host 'ec2-18-117-133-182.us-east-2.compute.amazonaws.com (18.117.133.182)' can't be established.
ECDSA key fingerprint is SHA256:wkppFyNTRv1mlaCyvE7pCbfwGBm0+i5h7PQR+fxD3MQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-117-133-182.us-east-2.compute.amazonaws.com,18.117.133.182' (ECDSA) to the list of known
 hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Jul  9 18:49:22 UTC 2021

  System load:  0.0               Processes:             105
  Usage of /:   16.4% of 7.69GB   Users logged in:       0
  Memory usage: 11%               IPv4 address for eth0: 172.31.21.114
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Configurar Instancia y cliente aws

Vamos a realizar una serie de configuraciones para permitir a la instancia de EC2 realizar las diferentes tareas que necesitaremos.

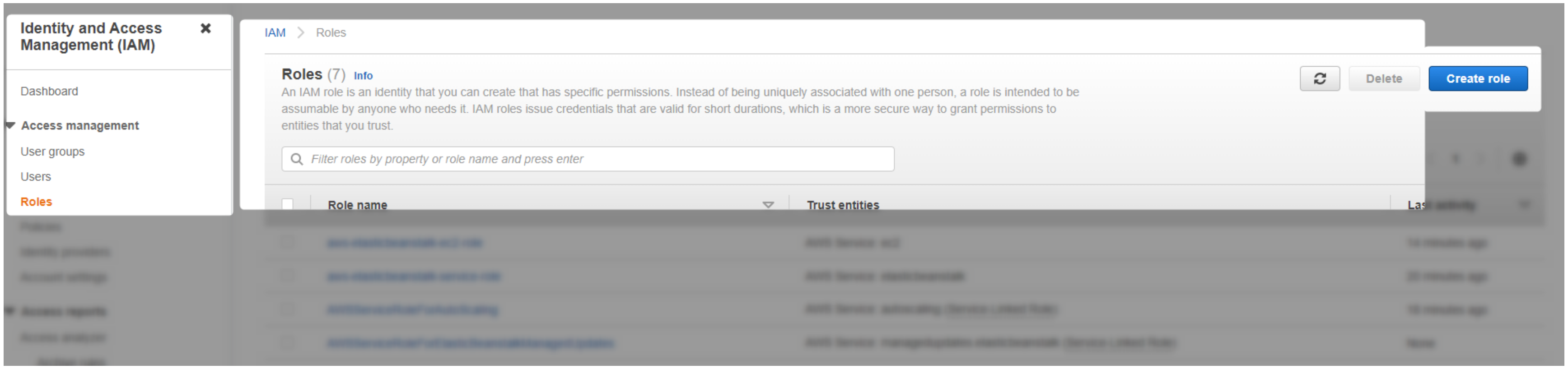
Crear Role

En la barra de busqueda escribir IAM y Abrir la consola



Luego creamos un role siguiente el siguiente flujo

IAM > Roles > Create Role



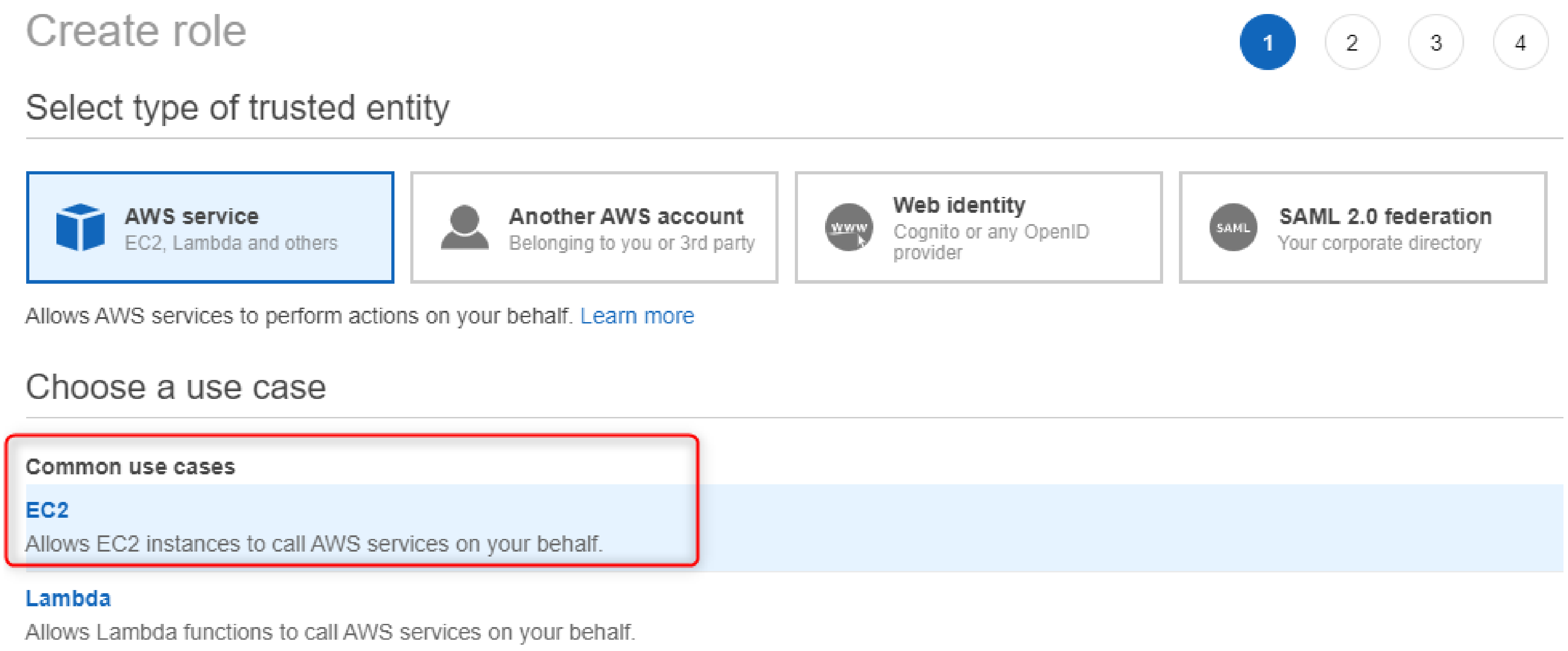
Roles (7) Info

An IAM role is an identity that you can create that has specific permissions. Instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. IAM roles issue credentials that are valid for short durations, which is a more secure way to grant permissions to entities that you trust.

Filter roles by property or role name and press enter

□	Role name	Trust entities	Last activity

Tipo de Role



Create role

Select type of trusted entity

Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

Common use cases

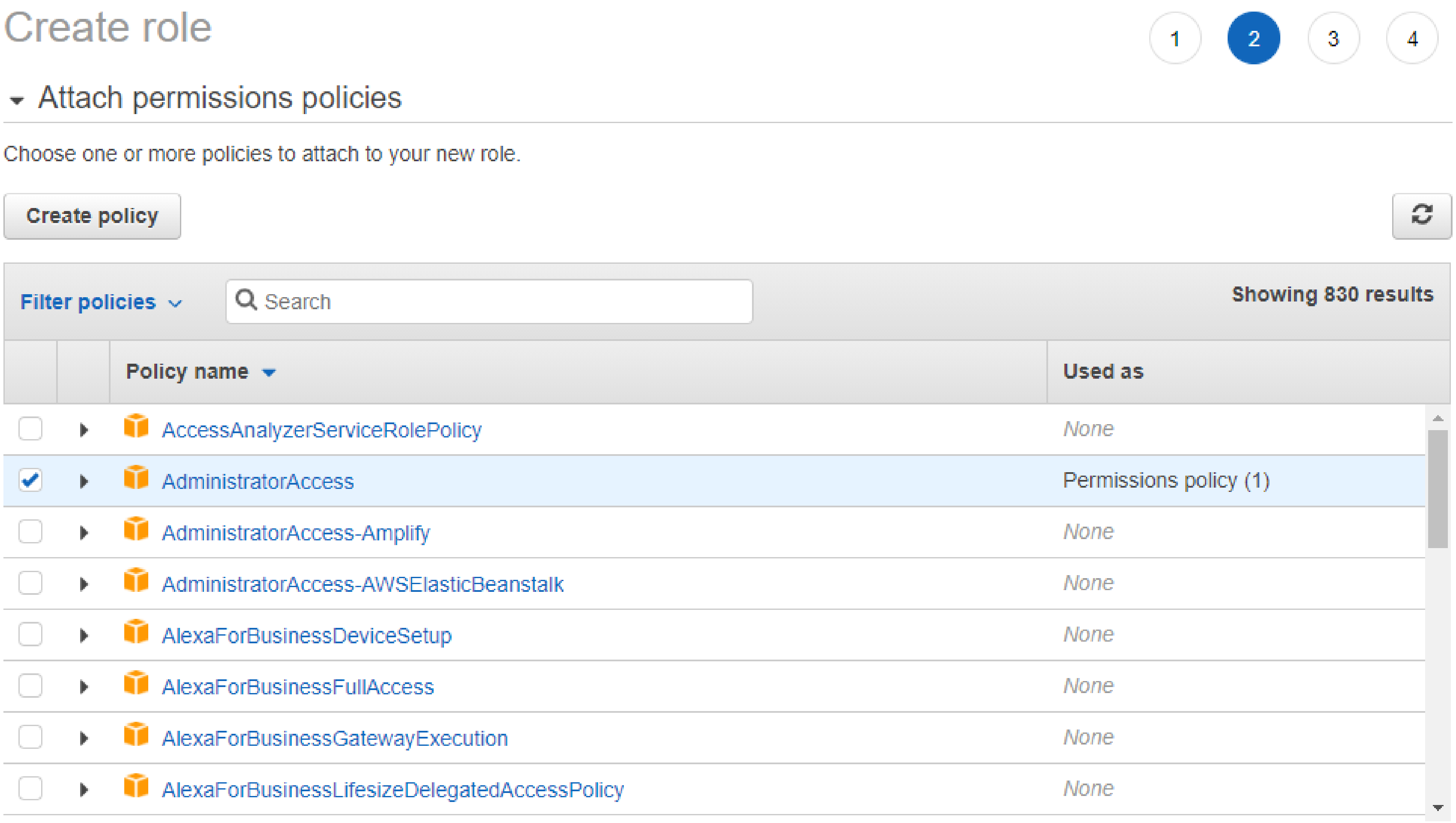
EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Asignar Permisos



Create role

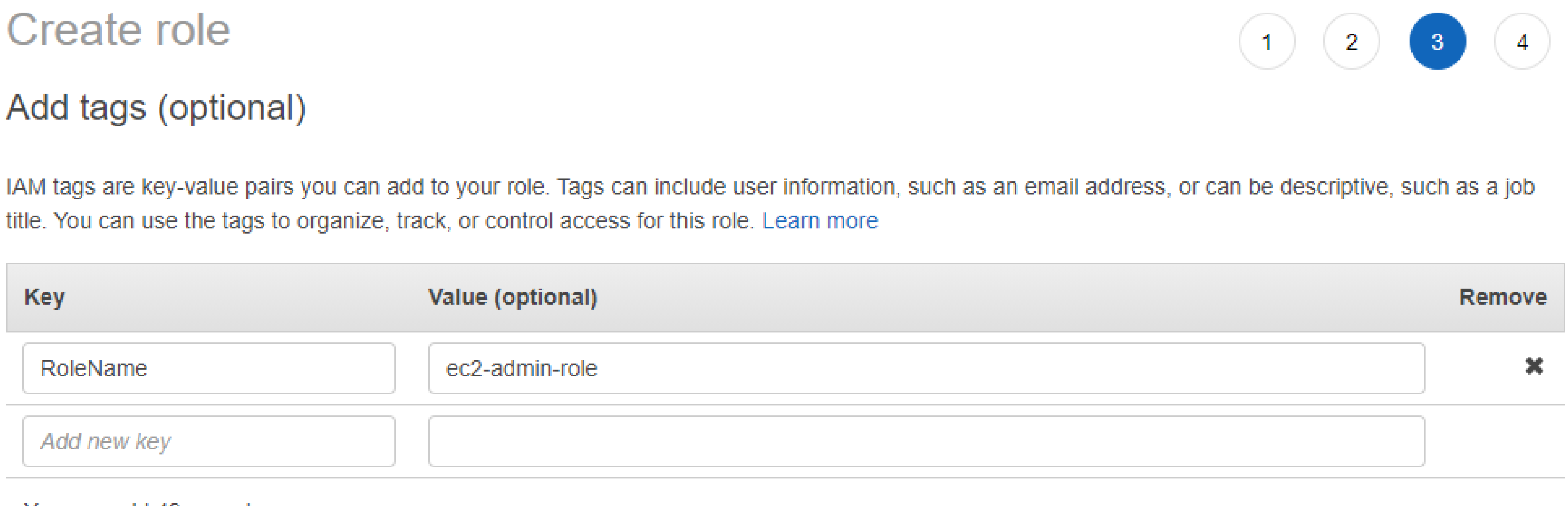
Attach permissions policies

Choose one or more policies to attach to your new role.

Showing 830 results

		Policy name	Used as
□	▸	AccessAnalyzerServiceRolePolicy	*None*
☑	▸	AdministratorAccess	Permissions policy (1)
□	▸	AdministratorAccess-Amplify	*None*
□	▸	AdministratorAccess-AWSElasticBeanstalk	*None*
□	▸	AlexaForBusinessDeviceSetup	*None*
□	▸	AlexaForBusinessFullAccess	*None*
□	▸	AlexaForBusinessGatewayExecution	*None*
□	▸	AlexaForBusinessLifesizeDelegatedAccessPolicy	*None*

Tags



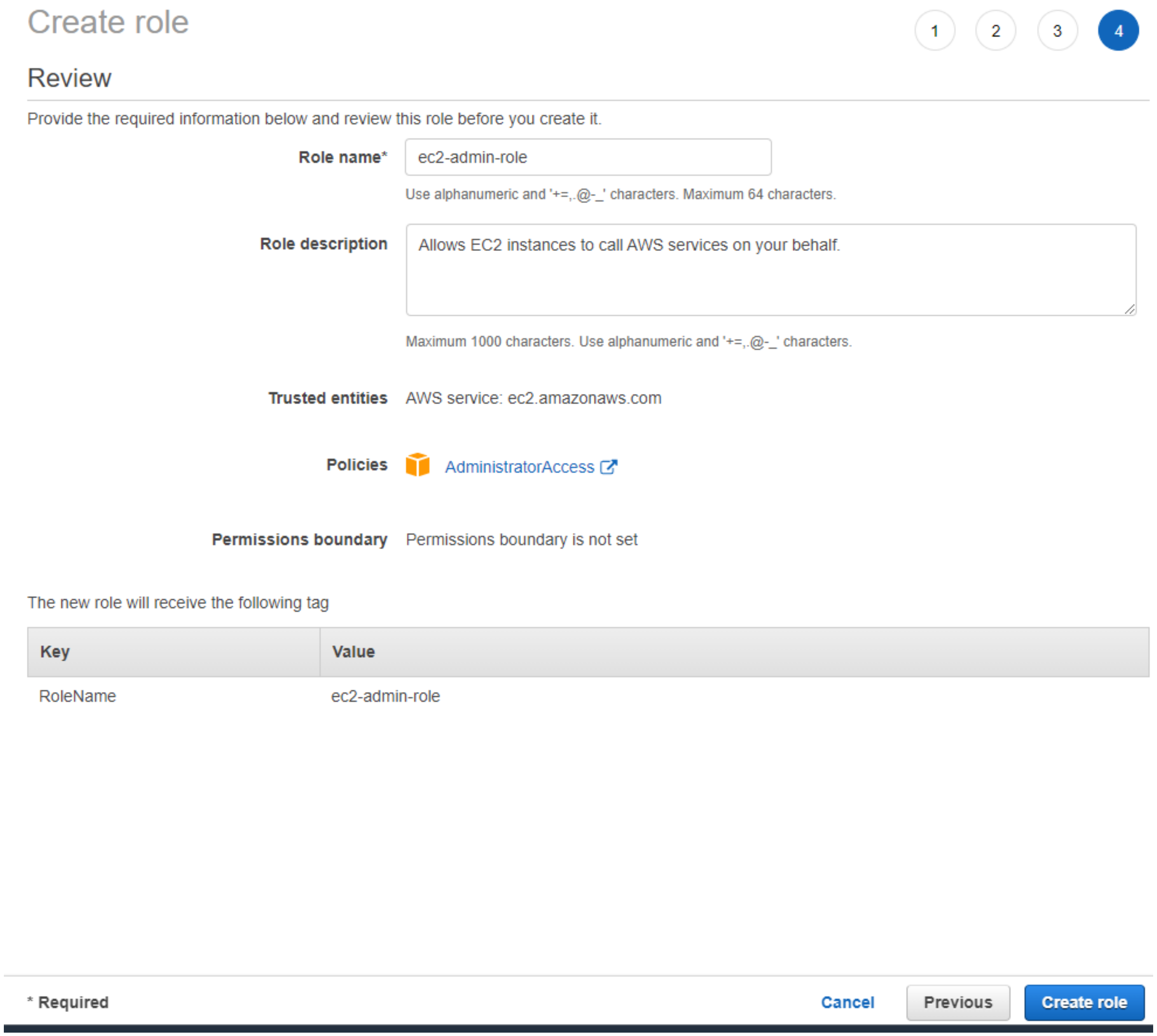
Create role

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. Learn more

Key	Value (optional)	Remove
RoleName	ec2-admin-role	✖
Add new key		

Crear



Review

Provide the required information below and review this role before you create it.

Role name* ec2-admin-role

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

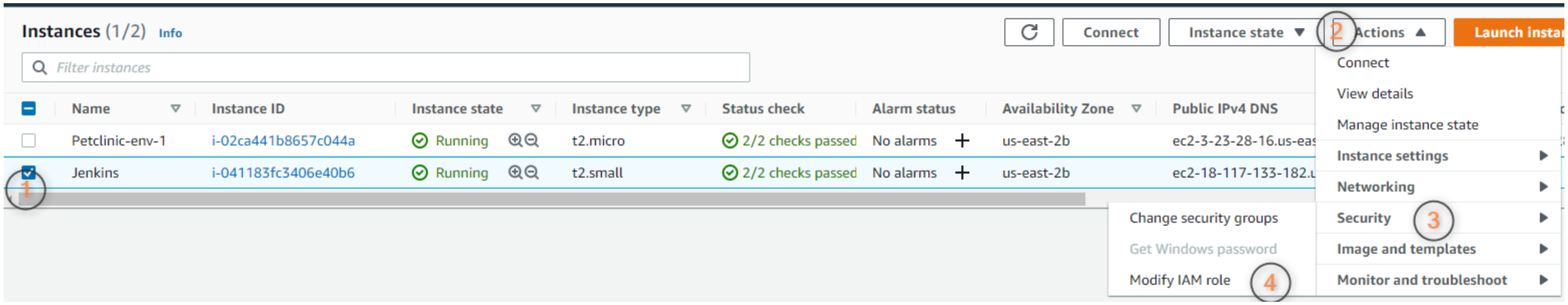
Policies AdministratorAccess

Permissions boundary Permissions boundary is not set

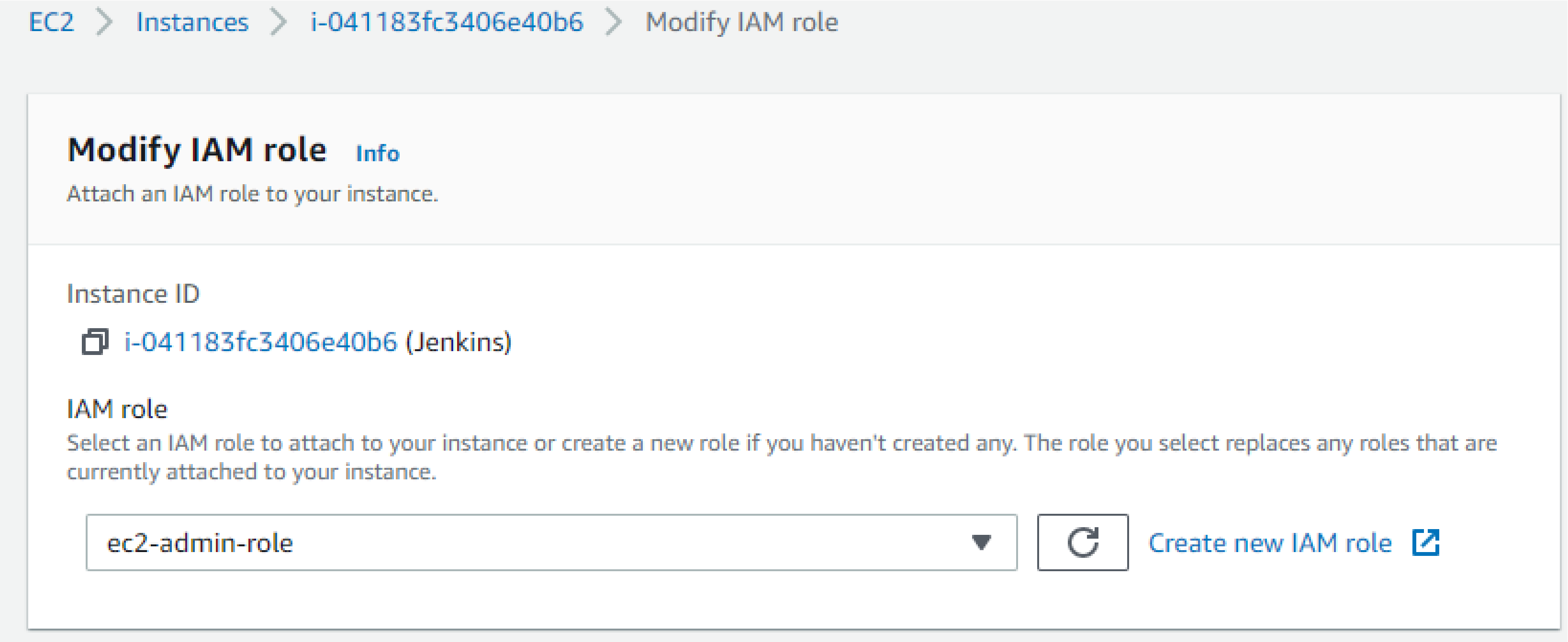
The new role will receive the following tag

Key	Value
RoleName	ec2-admin-role

Asignar Role a la instancia EC2



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	Petclinic-env-1	i-02ca441b8657c044a	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-3-23-28-16.us-eas
☑	Jenkins	i-041183fc3406e40b6	Running	t2.small	2/2 checks passed	No alarms	us-east-2b	ec2-18-117-133-182.u



EC2 > Instances > i-041183fc3406e40b6 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-041183fc3406e40b6 (Jenkins)

IAM role

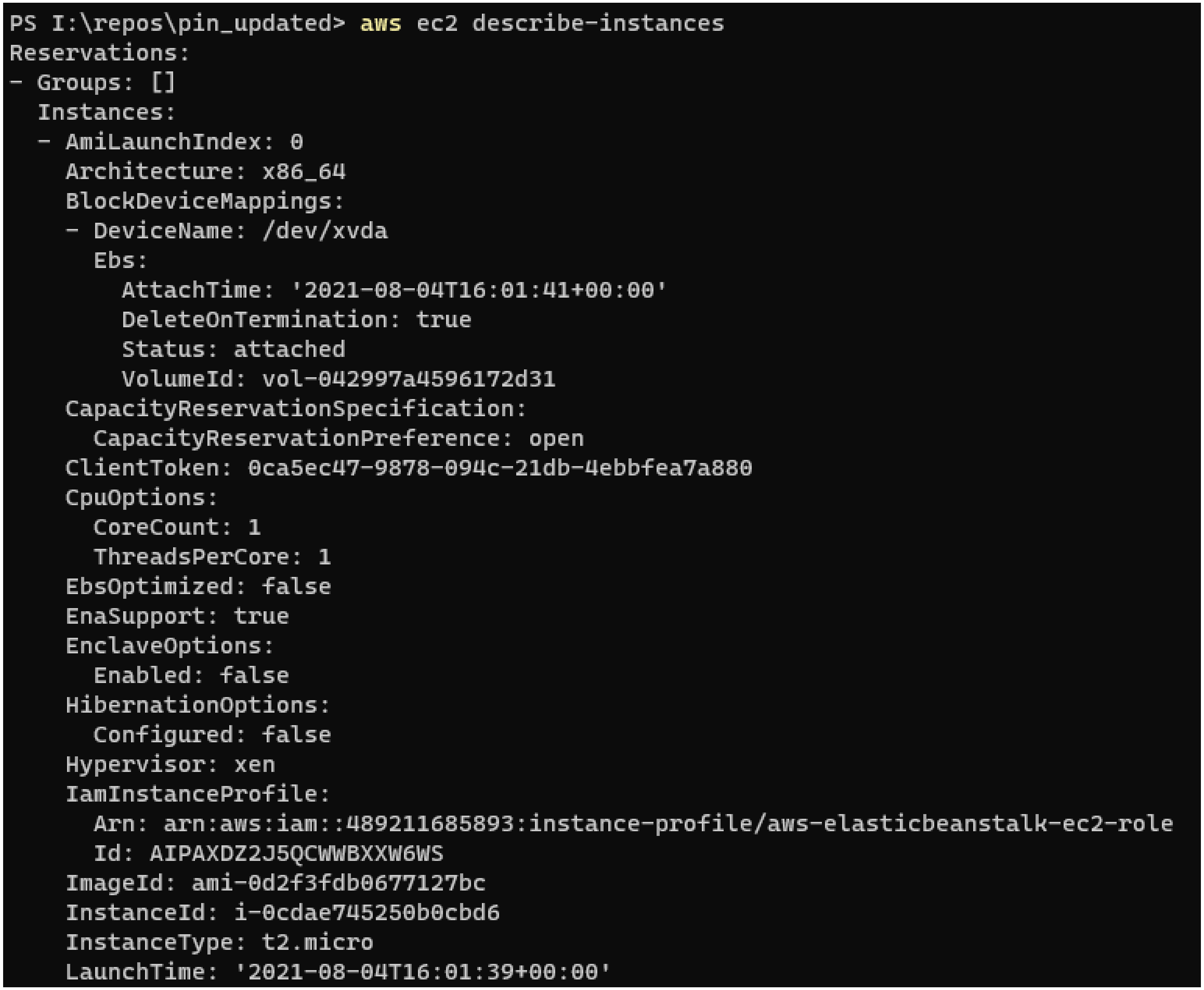
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

ec2-admin-role ▼

Luego en la consola de la instancia de EC2 ejecutar el comando <**aws configure**> para configurar las credenciales. No hace falta pasar Key ID o Secret ya que la instancia utilizara el role para autenticar.

AWS Access Key ID [None]: **(enter)**
AWS Secret Access Key [None]: **(enter)**
Default region name [None]: **us-east-2**
Default output format [None]:**yaml**

Para comprobar que nuestro role funciona podemos escribir el siguiente comando <**aws ec2 describe-instance**> este comando se conectara a nuestra cuenta y listara las instancias de EC2 que tengamos.



```
PS I:\repos\pin_updated> aws ec2 describe-instances
Reservations:
- Groups: []
  Instances:
  - AmiLaunchIndex: 0
    Architecture: x86_64
    BlockDeviceMappings:
    - DeviceName: /dev/xvda
      Ebs:
        AttachTime: '2021-08-04T16:01:41+00:00'
        DeleteOnTermination: true
        Status: attached
        VolumeId: vol-042997a4596172d31
    CapacityReservationSpecification:
      CapacityReservationPreference: open
    ClientToken: 0ca5ec47-9878-094c-21db-4ebbfea7a880
    CpuOptions:
      CoreCount: 1
      ThreadsPerCore: 1
    EbsOptimized: false
    EnaSupport: true
    EnclaveOptions:
      Enabled: false
    HibernationOptions:
      Configured: false
    Hypervisor: xen
    IamInstanceProfile:
      Arn: arn:aws:iam::489211685893:instance-profile/aws-elasticbeanstalk-ec2-role
      Id: AIPAXDZ2J5QCWWBXXW6WS
    ImageId: ami-0d2f3fdb0677127bc
    InstanceId: i-0cdae745250b0cbd6
    InstanceType: t2.micro
    LaunchTime: '2021-08-04T16:01:39+00:00'
```

Crear cluster con eksctl

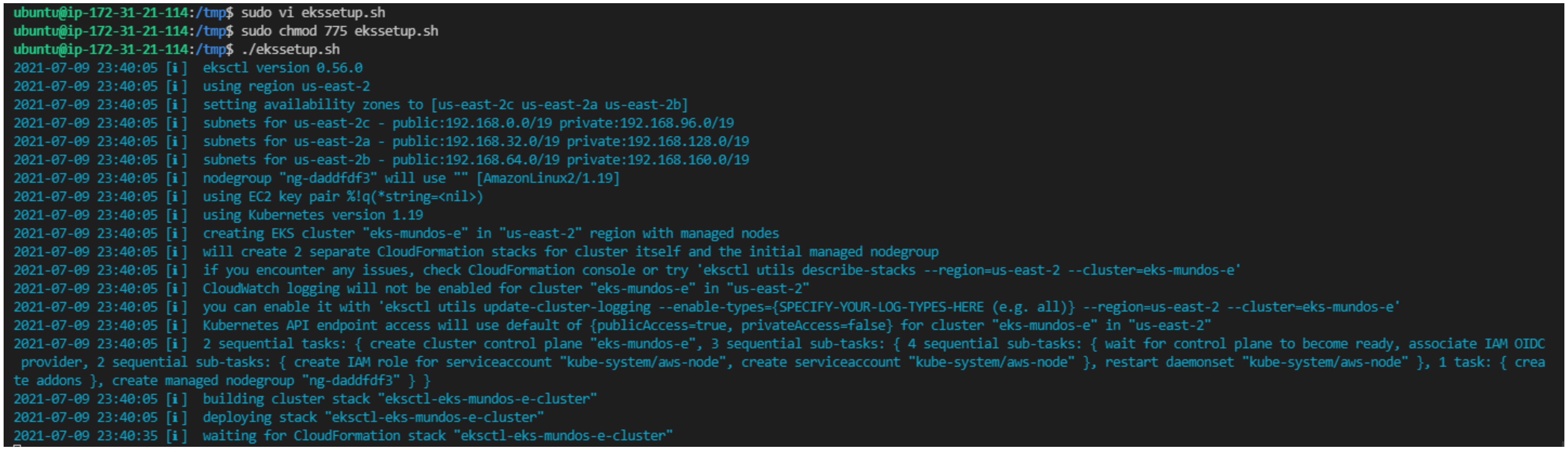
Crear Cluster de EKS

Dentro de la instancia de EC2 corremos el siguiente comando

```
eksctl create cluster \
--name eks-mundos-e \
--region us-east-2 \
--node-type t2.small \
--with-oidc \
--ssh-access \
--ssh-public-key jenkins \
--managed \
--full-ecr-access
--zones us-east-2a,us-east-2b,us-east-2c
```

Nota: El nombre de la ssh-public-key debe ser el mismo que la key generada en el paso Create Key Pair

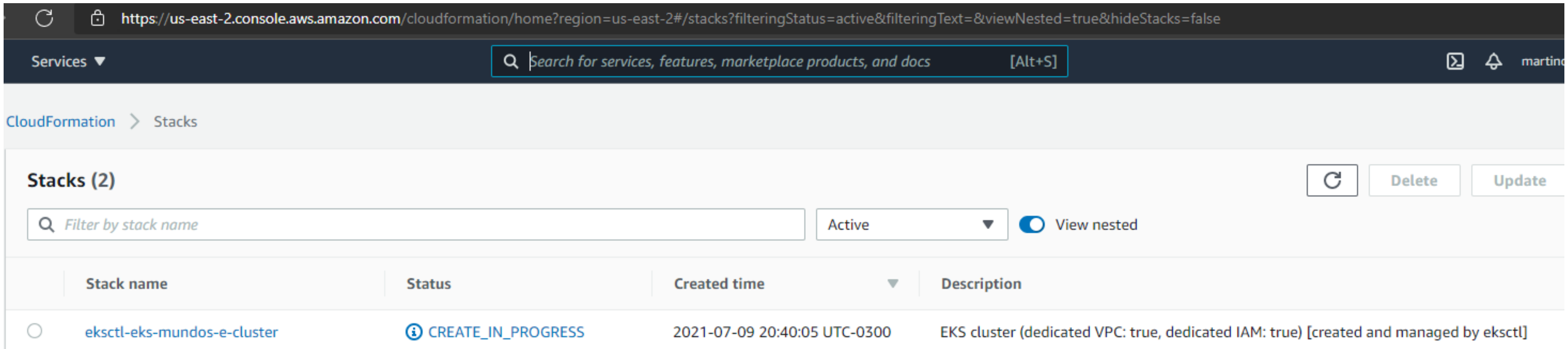
Verificar Progreso shell



```
ubuntu@ip-172-31-21-114:/tmp$ sudo vi ekssetup.sh
ubuntu@ip-172-31-21-114:/tmp$ sudo chmod 775 ekssetup.sh
ubuntu@ip-172-31-21-114:/tmp$ ./ekssetup.sh
2021-07-09 23:40:05 [i]  eksctl version 0.56.0
2021-07-09 23:40:05 [i]  using region us-east-2
2021-07-09 23:40:05 [i]  setting availability zones to [us-east-2c us-east-2a us-east-2b]
2021-07-09 23:40:05 [i]  subnets for us-east-2c - public:192.168.0.0/19 private:192.168.96.0/19
2021-07-09 23:40:05 [i]  subnets for us-east-2a - public:192.168.32.0/19 private:192.168.128.0/19
2021-07-09 23:40:05 [i]  subnets for us-east-2b - public:192.168.64.0/19 private:192.168.160.0/19
2021-07-09 23:40:05 [i]  nodegroup "ng-daddfdf3" will use "" [AmazonLinux2/1.19]
2021-07-09 23:40:05 [i]  using EC2 key pair %!q(*string=<nil>)
2021-07-09 23:40:05 [i]  using Kubernetes version 1.19
2021-07-09 23:40:05 [i]  creating EKS cluster "eks-mundos-e" in "us-east-2" region with managed nodes
2021-07-09 23:40:05 [i]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2021-07-09 23:40:05 [i]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=eks-mundos-e'
2021-07-09 23:40:05 [i]  CloudWatch logging will not be enabled for cluster "eks-mundos-e" in "us-east-2"
2021-07-09 23:40:05 [i]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-2 --cluster=eks-mundos-e'
2021-07-09 23:40:05 [i]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "eks-mundos-e" in "us-east-2"
2021-07-09 23:40:05 [i]  2 sequential tasks: { create cluster control plane "eks-mundos-e", 3 sequential sub-tasks: { 4 sequential sub-tasks: { wait for control plane to become ready, associate IAM OIDC
 provider, 2 sequential sub-tasks: { create IAM role for serviceaccount "kube-system/aws-node", create serviceaccount "kube-system/aws-node" }, restart daemonset "kube-system/aws-node" }, 1 task: { crea
te addons }, create managed nodegroup "ng-daddfdf3" } }
2021-07-09 23:40:05 [i]  building cluster stack "eksctl-eks-mundos-e-cluster"
2021-07-09 23:40:05 [i]  deploying stack "eksctl-eks-mundos-e-cluster"
2021-07-09 23:40:35 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-cluster"
```

Verificar progreso CloudFormation

Se puede verificar desde la consola de CloudFormation su progreso

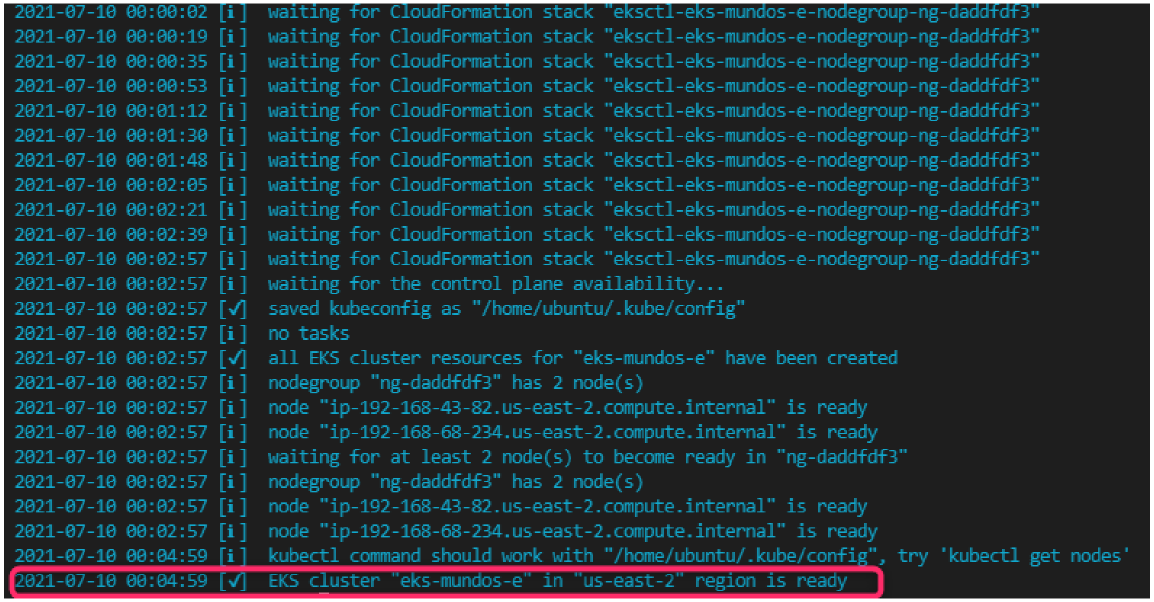


https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks?filteringStatus=active&filteringText=&viewNested=true&hideStacks=false

Stacks (2)

	Stack name	Status	Created time ▼	Description
○	eksctl-eks-mundos-e-cluster	CREATE_IN_PROGRESS	2021-07-09 20:40:05 UTC-0300	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

Resultado Exitoso



```
2021-07-10 00:00:02 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:00:19 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:00:35 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:00:53 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:01:12 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:01:30 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:01:48 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:02:05 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:02:21 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:02:39 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:02:57 [i]  waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-ng-daddfdf3"
2021-07-10 00:02:57 [i]  waiting for the control plane availability...
2021-07-10 00:02:57 [✓]  saved kubeconfig as "/home/ubuntu/.kube/config"
2021-07-10 00:02:57 [i]  no tasks
2021-07-10 00:02:57 [✓]  all EKS cluster resources for "eks-mundos-e" have been created
2021-07-10 00:02:57 [i]  nodegroup "ng-daddfdf3" has 2 node(s)
2021-07-10 00:02:57 [i]  node "ip-192-168-43-82.us-east-2.compute.internal" is ready
2021-07-10 00:02:57 [i]  node "ip-192-168-68-234.us-east-2.compute.internal" is ready
2021-07-10 00:02:57 [i]  waiting for at least 2 node(s) to become ready in "ng-daddfdf3"
2021-07-10 00:02:57 [i]  nodegroup "ng-daddfdf3" has 2 node(s)
2021-07-10 00:02:57 [i]  node "ip-192-168-43-82.us-east-2.compute.internal" is ready
2021-07-10 00:02:57 [i]  node "ip-192-168-68-234.us-east-2.compute.internal" is ready
2021-07-10 00:04:59 [i]  kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2021-07-10 00:04:59 [✓]  EKS cluster "eks-mundos-e" in "us-east-2" region is ready
```

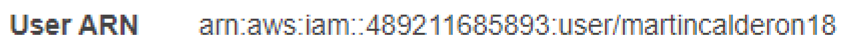
MapUsers

Para poder acceder al cluster en la consola de AWS EKS tenemos que autorizar a nuestro usuario IAM, esto lo vamos a hacer agregando el mismo a las configuraciones. Ref: [Managing users or IAM roles for your cluster - Amazon EKS]()

Corremos los siguientes comandos en la consola de la instancia de EC2

kubectl describe configmap -n kube-system aws-auth Para ver la configuración actual

En la consola de IAM buscamos nuestro usuario, necesitamos capturar el ARN y el usuario (martincalderon18) en este caso



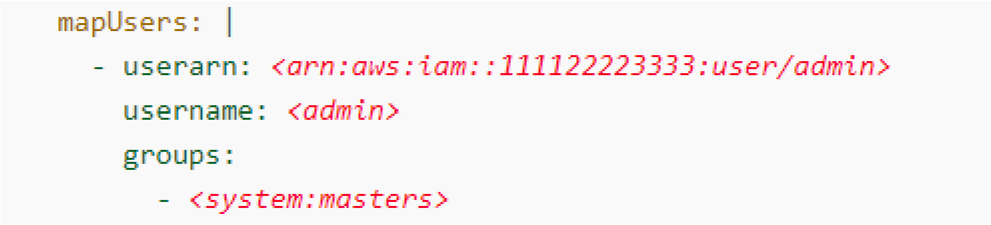
User ARN   arn:aws:iam::489211685893:user/martincalderon18

Agregar usuario IAM de AWS

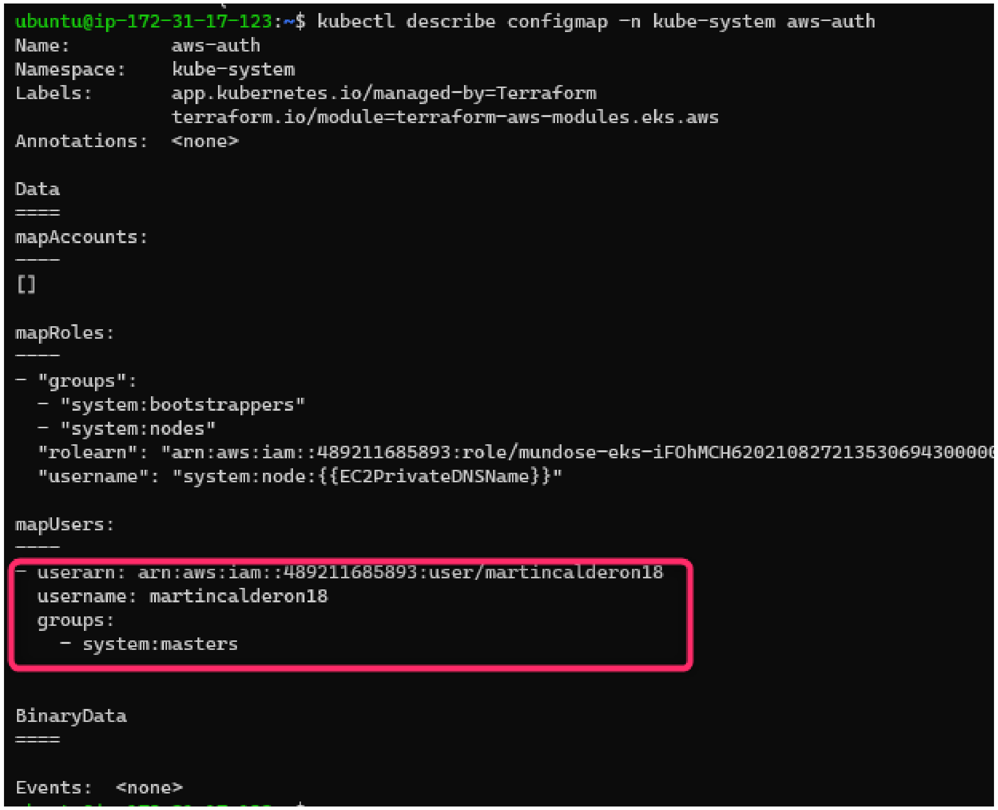
```
kubectl edit -n kube-system configmap/aws-auth
```

El formato pertenece a una lista de objetos en formato yaml

Cuando terminamos de agregar las líneas, salvamos los cambios con los mismos pasos que en VI (:qw! ---Enter)



```
  mapUsers: |
    - userarn: <arn:aws:iam::111122223333:user/admin>
      username: <admin>
      groups:
        - <system:masters>
```



```
ubuntu@ip-172-31-17-123:~$ kubectl describe configmap -n kube-system aws-auth
Name:         aws-auth
Namespace:    kube-system
Labels:       app.kubernetes.io/managed-by=Terraform
              terraform.io/module=terraform-aws-modules.eks.aws
Annotations:  <none>

Data
====
mapAccounts:
----
[]

mapRoles:
----
- "groups":
  - "system:bootstrappers"
  - "system:nodes"
  "rolearn": "arn:aws:iam::489211685893:role/mundose-eks-iFOhMCH620210827213530694300000
  "username": "system:node:{{EC2PrivateDNSName}}"

mapUsers:
----
- userarn: arn:aws:iam::489211685893:user/martincalderon18
  username: martincalderon18
  groups:
    - system:masters


BinaryData
====

Events:  <none>
```

Crea cluster con Terraform

Crear Cluster de EKS

Dentro de la instancia de EC2 clonamos el siguiente repositorio Terraform Module EKS Repo

Luego navegamos a la carpeta **eks_setup_terraform** y ahi primero instalamos el proveedor de AWS EKS, vpc, security groups entre otros y luego desplegamos el cluster. Esto puede tomar 15-20 Minutos

Terraform init (Descarga los proveedores)

Terraform apply (Despliega el cluster)

Configurar kubectl

Para poder conectarnos al cluster tenemos que configurar

```
aws eks --region $(terraform output -raw region) update-kubeconfig --name $(terraform output -raw cluster_name)
```

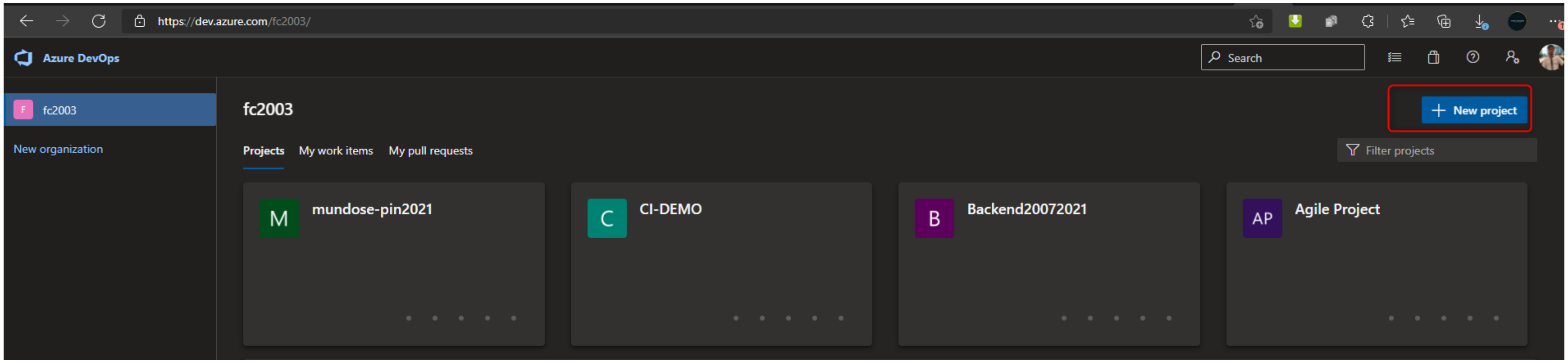
Azure Devops Setup

La configuración de Azure Devops consiste en crear una **cuenta**, crear un **proyecto**, crear un **repositorio** de azure repos y agregar el cluster de kubernetes

Es necesario crear una cuenta gratuita en https://dev.azure.com/

Crear un proyecto azure Devops

Desde la organización creada por defecto al crear la cuenta seleccionar **+ New Project**



Definir un nombre, tipo de proyecto publico y proceso agile.

Una vez creado instanciar un repositorio con un readme por defecto

Agregar el cluster de Kubernetes a Azure Devops

Para poder configurar el cluster en AzDO vamos a crear un ServiceAccount en kubernetes (Provee una identidad para procesos que corren en un pod), luego agregarlo a AzDo como Service Connection para luego poder utilizarla en los pipelines.

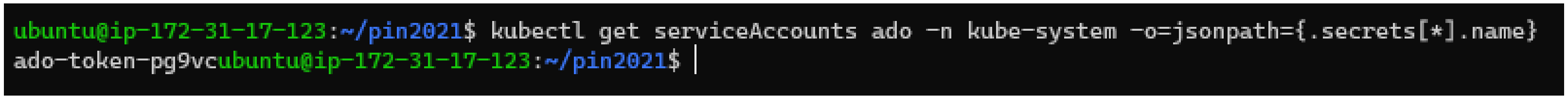
Crear ServiceAccount para Azure Devops

[azdo service account]()

Kubectl apply -f ado-admin-service-account.yaml

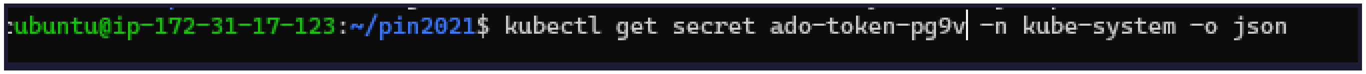
Obtener secret asociado

kubectl get serviceAccounts ado -n kube-system -o=jsonpath={.secrets[*].name}



```
ubuntu@ip-172-31-17-123:~/pin2021$ kubectl get serviceAccounts ado -n kube-system -o=jsonpath={.secrets[*].name}
ado-token-pg9vcubuntu@ip-172-31-17-123:~/pin2021$ |
```

kubectl get secret ado-token-pg9vc -n kube-system -o json

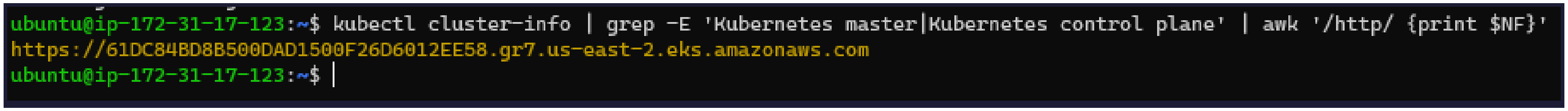


```
:ubuntu@ip-172-31-17-123:~/pin2021$ kubectl get secret ado-token-pg9v| -n kube-system -o json
```

Obtener API URL

Desde la consola de la instancia de EC2

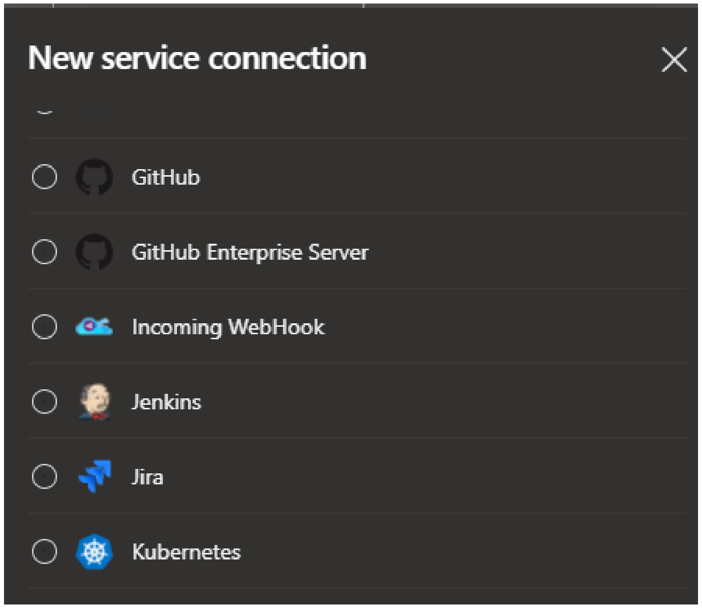
```
kubectl cluster-info | grep -E 'Kubernetes master|Kubernetes control plane' | awk '/http/ {print $NF}'
```



```
ubuntu@ip-172-31-17-123:~$ kubectl cluster-info | grep -E 'Kubernetes master|Kubernetes control plane' | awk '/http/ {print $NF}'
https://61DC84BD8B500DAD1500F26D6012EE58.gr7.us-east-2.eks.amazonaws.com
ubuntu@ip-172-31-17-123:~$ |
```

Agregar la cuenta de Servicio

Project Settings > Service Connection > new service connection > Elegimos Kubernetes > utilizamos los datos del paso anterior



New service connection

○ GitHub

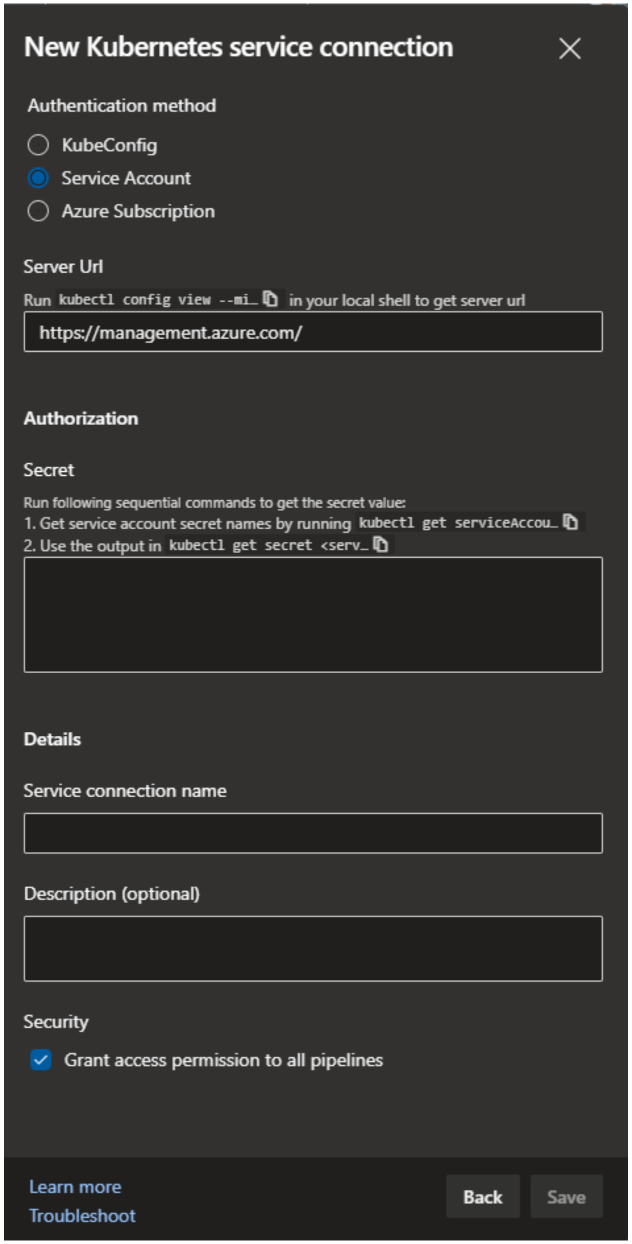
○ GitHub Enterprise Server

○ Incoming WebHook

○ Jenkins

○ Jira

○ Kubernetes



New Kubernetes service connection

Authentication method

○ KubeConfig

● Service Account

○ Azure Subscription

Server Url

Run `kubectl config view --mi…` in your local shell to get server url

https://management.azure.com/

Authorization

Secret

Run following sequential commands to get the secret value:

1. Get service account secret names by running `kubectl get serviceAccou…`
2. Use the output in `kubectl get secret <serv…`

Details

Service connection name

Description (optional)

Security

☑ Grant access permission to all pipelines

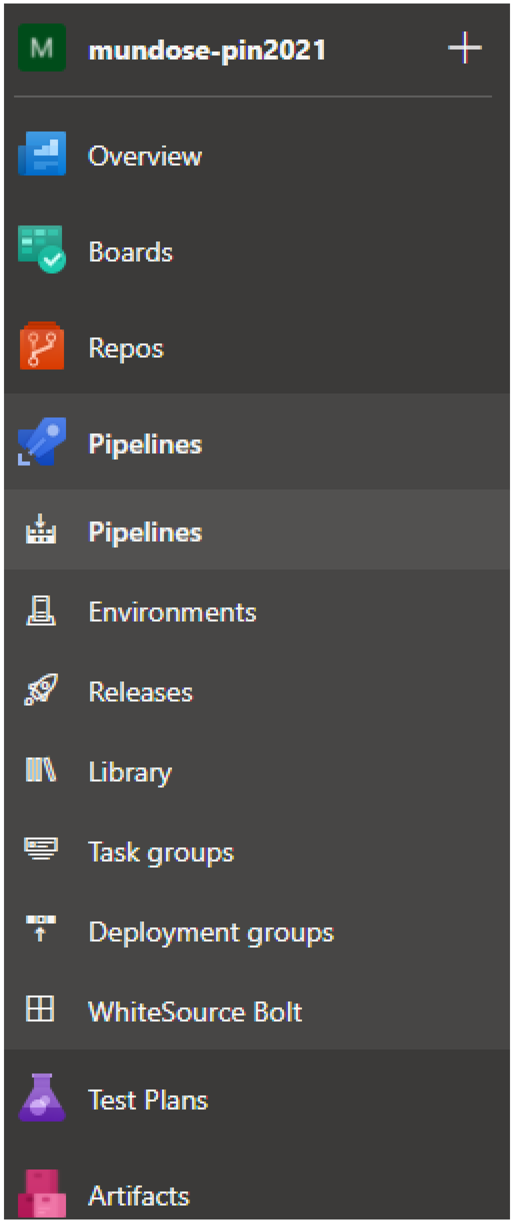
Azure Devops Pipeline Setup

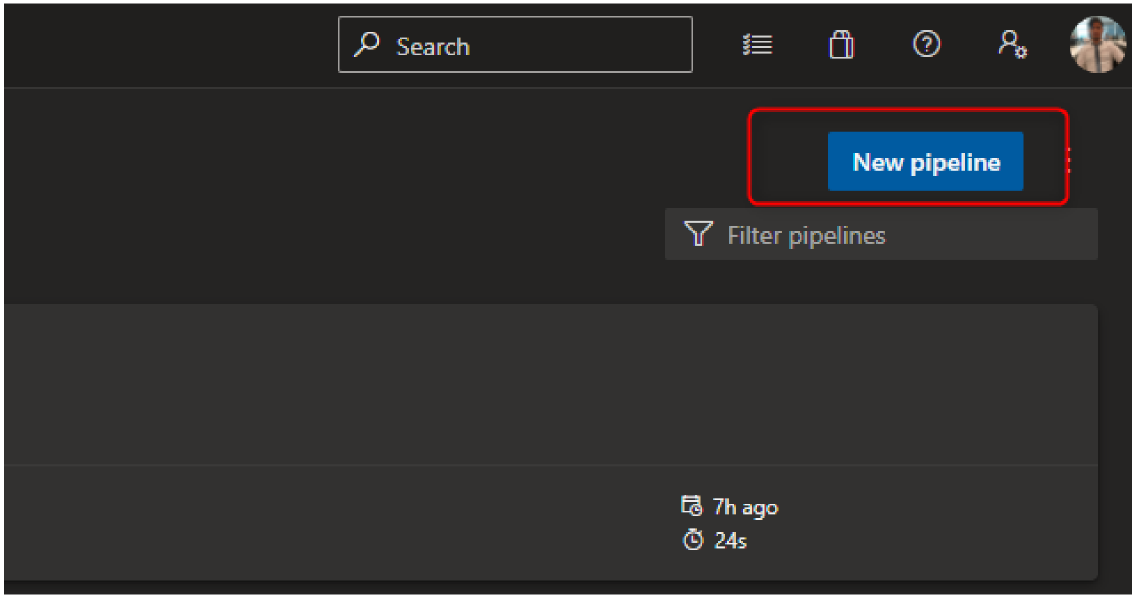
Para realizar el despliegue de nginx (Web) al cluster vamos a utilizar un archivo de deployment que también contiene un servicio del tipo Load Balancer (Externo)

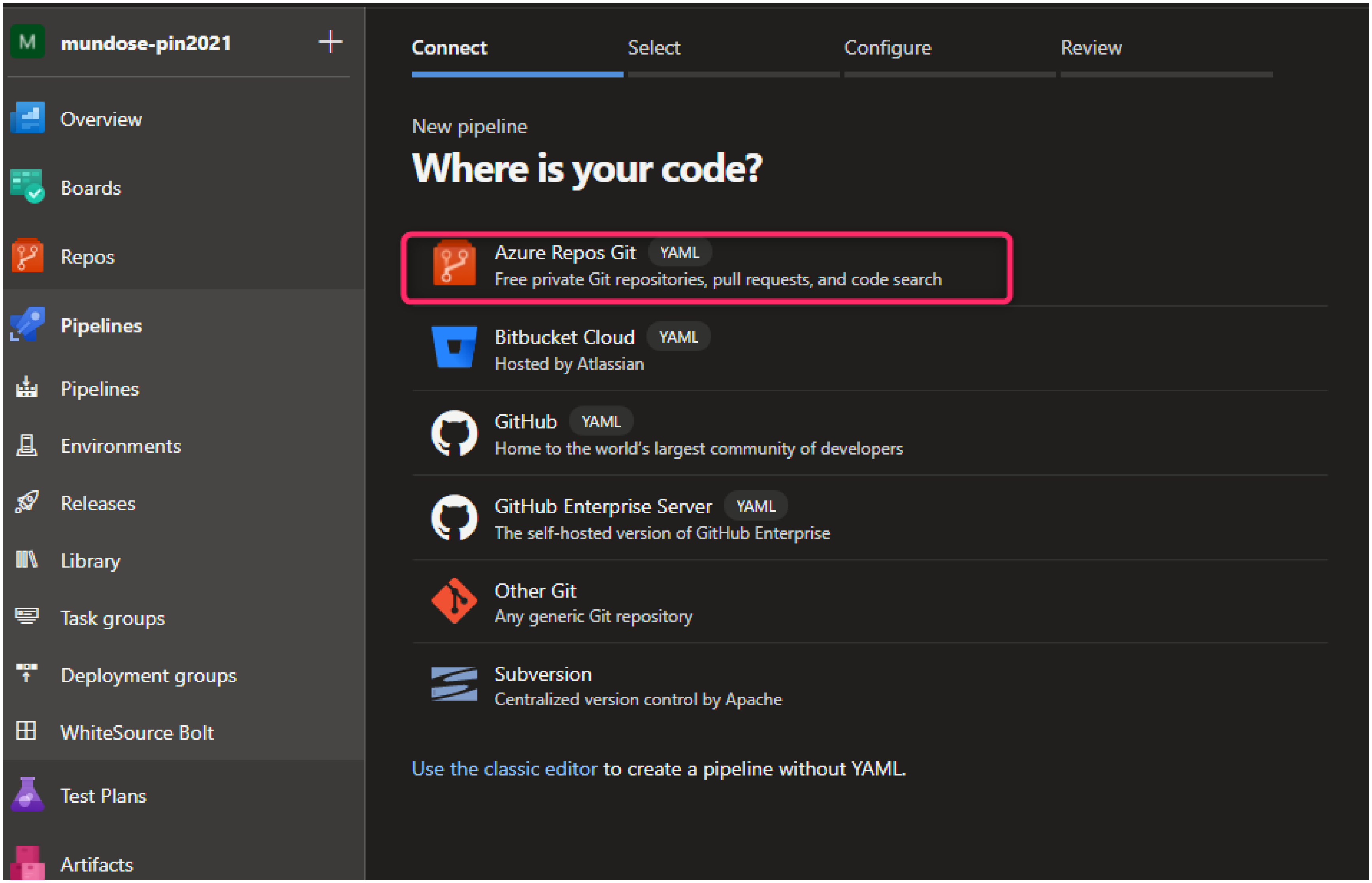
nginx-deployment.yaml repo

Dentro del proyecto de Azure Devops > Pipelines > New Pipeline

Seguimos el navegador con las selecciones que se detallan abajo


M
mundose-pin2021
+
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
WhiteSource Bolt
Test Plans
Artifacts


Search
New pipeline
Filter pipelines
7h ago
24s



New pipeline

Where is your code?

Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

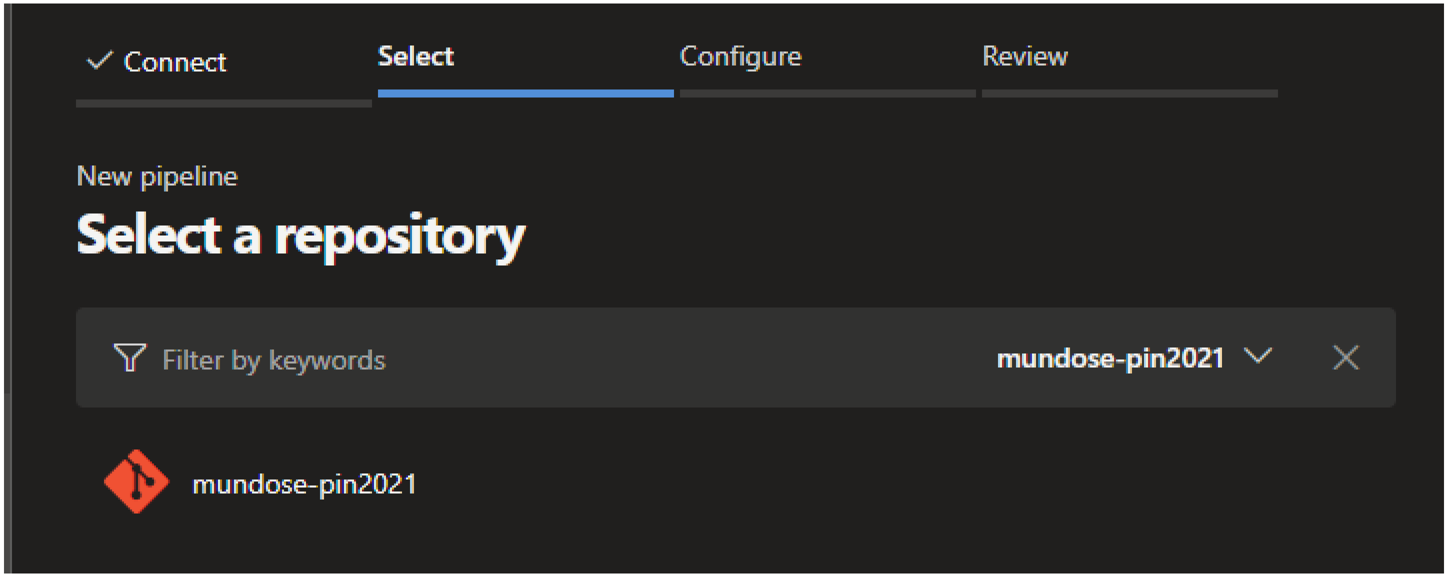
GitHub YAML
Home to the world's largest community of developers

GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

Other Git
Any generic Git repository

Subversion
Centralized version control by Apache

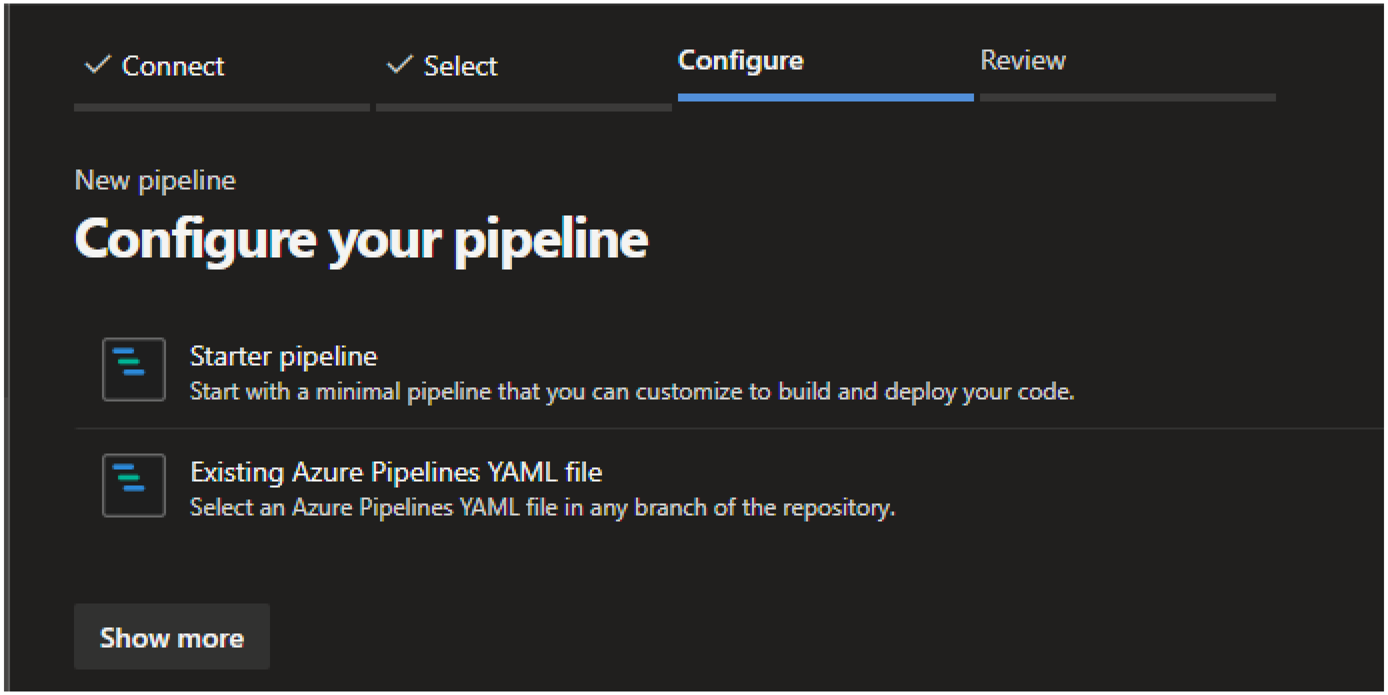
Use the classic editor to create a pipeline without YAML.



New pipeline

Select a repository

mundose-pin2021

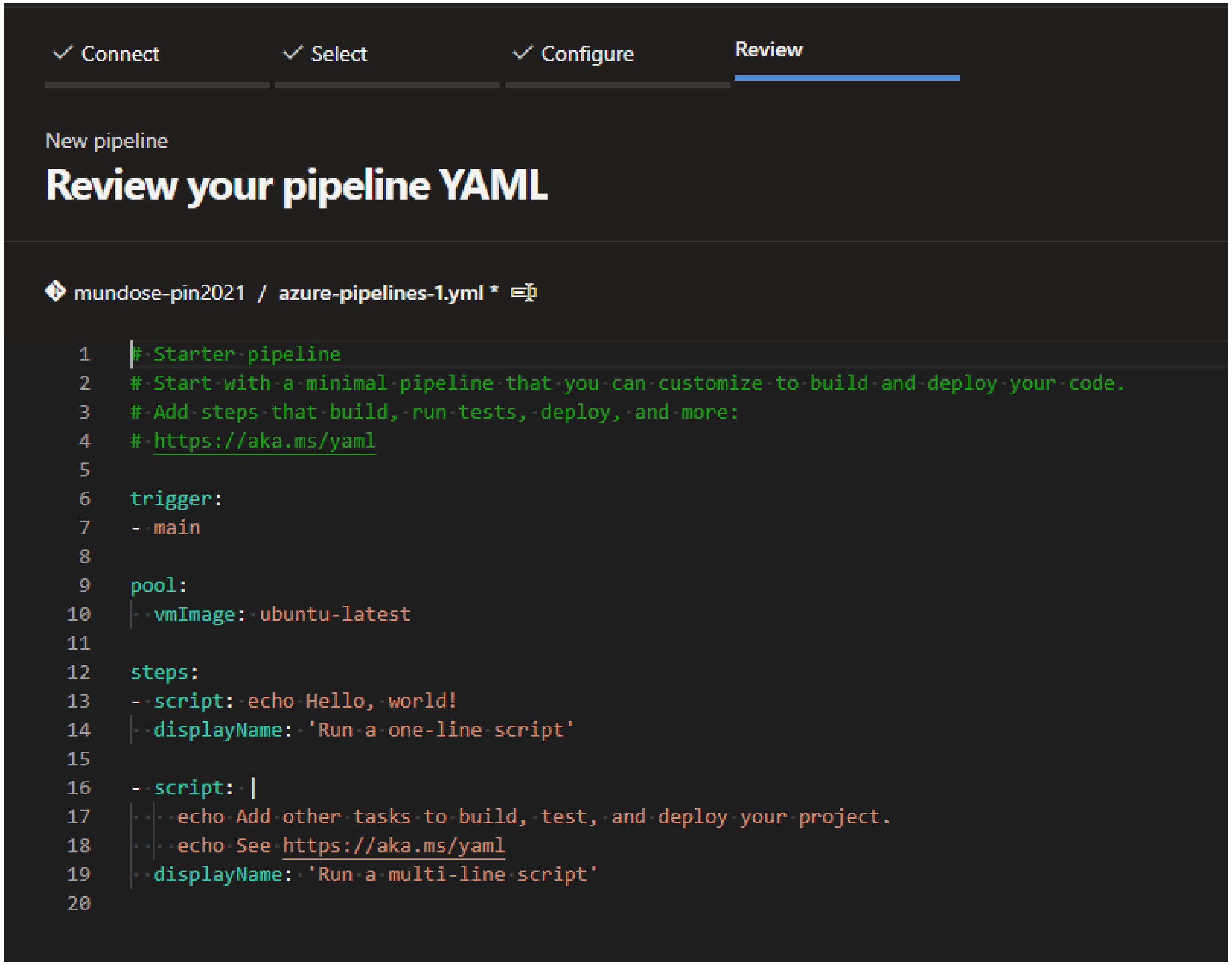


New pipeline

Configure your pipeline

Starter pipeline
Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file
Select an Azure Pipelines YAML file in any branch of the repository.



New pipeline

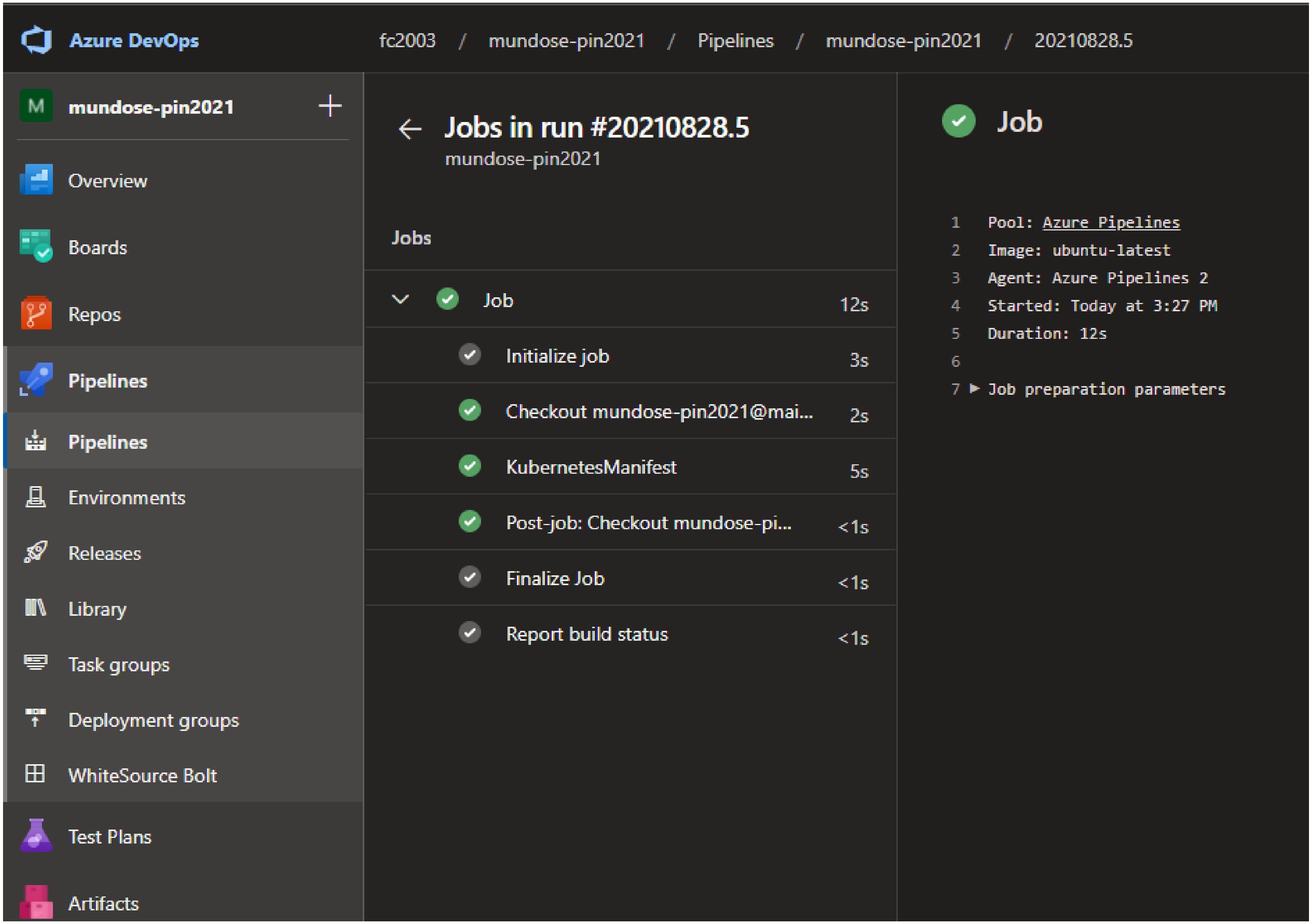
Review your pipeline YAML

mundose-pin2021 / azure-pipelines-1.yml *

```yaml
# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
- main

pool:
  vmImage: ubuntu-latest

steps:
- script: echo Hello, world!
  displayName: 'Run a one-line script'

- script: |
    echo Add other tasks to build, test, and deploy your project.
    echo See https://aka.ms/yaml
  displayName: 'Run a multi-line script'

```

Reemplazamos el código con este aquí azure-pipeline.yaml repo (cambiar el nombre de la conexión por el que hayan seleccionado)

Al salvar el pipeline se va a iniciar el mismo de manera automática y finalmente podemos inspeccionar el job para ver en el paso del manifiesto la url creada para la nginx



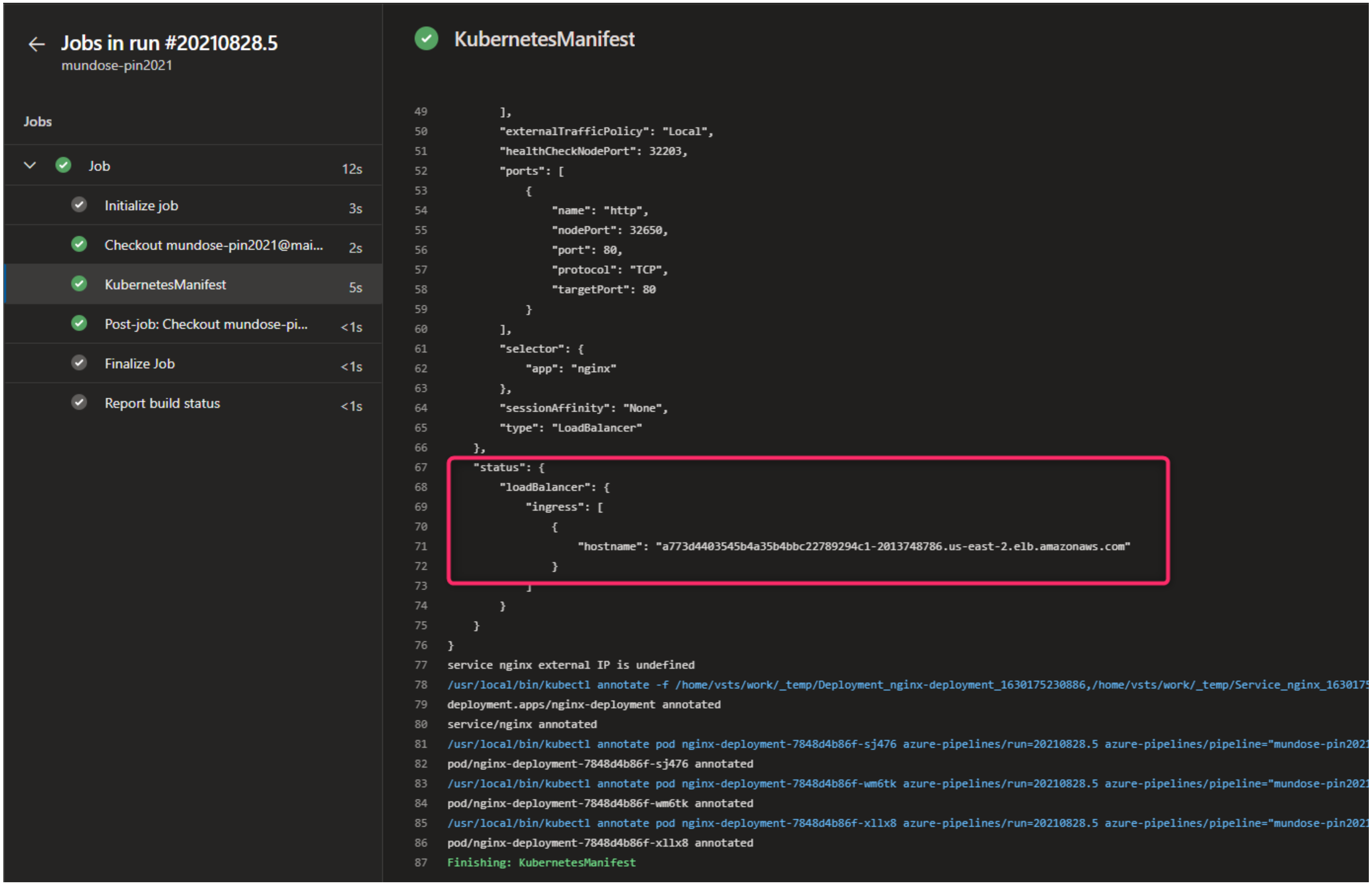
Jobs in run #20210828.5
mundose-pin2021

Jobs

```
Job                                  12s
  Initialize job                      3s
  Checkout mundose-pin2021@mai...     2s
  KubernetesManifest                  5s
  Post-job: Checkout mundose-pi...   <1s
  Finalize Job                       <1s
  Report build status                <1s
```

Job

```
Pool: Azure Pipelines
Image: ubuntu-latest
Agent: Azure Pipelines 2
Started: Today at 3:27 PM
Duration: 12s

▶ Job preparation parameters
```



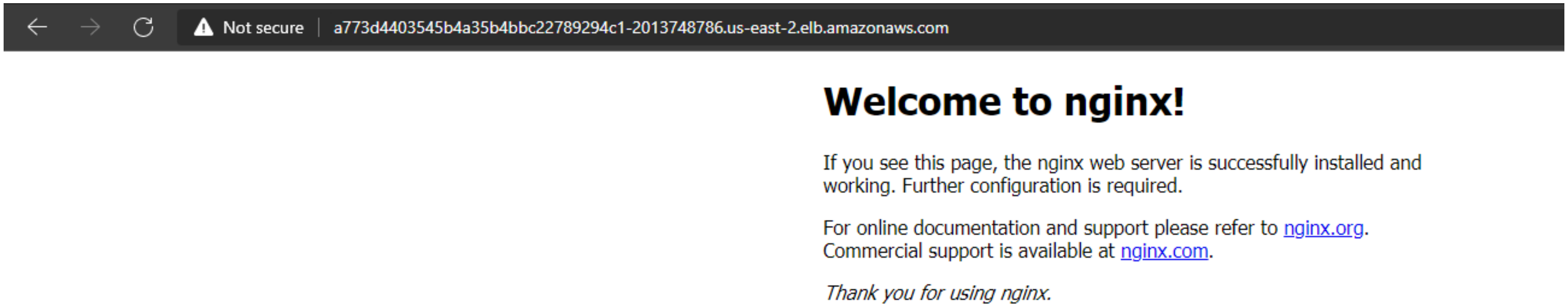
Jobs in run #20210828.5
mundose-pin2021

Jobs

Job	12s
Initialize job	3s
Checkout mundose-pin2021@mai...	2s
KubernetesManifest	5s
Post-job: Checkout mundose-pi...	<1s
Finalize Job	<1s
Report build status	<1s

KubernetesManifest

```
        ],
        "externalTrafficPolicy": "Local",
        "healthCheckNodePort": 32203,
        "ports": [
            {
                "name": "http",
                "nodePort": 32650,
                "port": 80,
                "protocol": "TCP",
                "targetPort": 80
            }
        ],
        "selector": {
            "app": "nginx"
        },
        "sessionAffinity": "None",
        "type": "LoadBalancer"
    },
    "status": {
        "loadBalancer": {
            "ingress": [
                {
                    "hostname": "a773d4403545b4a35b4bbc22789294c1-2013748786.us-east-2.elb.amazonaws.com"
                }
            ]
        }
    }
}
service nginx external IP is undefined
/usr/local/bin/kubectl annotate -f /home/vsts/work/_temp/Deployment_nginx-deployment_1630175230886,/home/vsts/work/_temp/Service_nginx_1630175
deployment.apps/nginx-deployment annotated
service/nginx annotated
/usr/local/bin/kubectl annotate pod nginx-deployment-7848d4b86f-sj476 azure-pipelines/run=20210828.5 azure-pipelines/pipeline="mundose-pin2021
pod/nginx-deployment-7848d4b86f-sj476 annotated
/usr/local/bin/kubectl annotate pod nginx-deployment-7848d4b86f-wm6tk azure-pipelines/run=20210828.5 azure-pipelines/pipeline="mundose-pin2021
pod/nginx-deployment-7848d4b86f-wm6tk annotated
/usr/local/bin/kubectl annotate pod nginx-deployment-7848d4b86f-xllx8 azure-pipelines/run=20210828.5 azure-pipelines/pipeline="mundose-pin2021
pod/nginx-deployment-7848d4b86f-xllx8 annotated
Finishing: KubernetesManifest
```



a773d4403545b4a35b4bbc22789294c1-2013748786.us-east-2.elb.amazonaws.com

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.

LO LOGRAMOS!!! Nuestro nginx se encuentra desplegado y accesible desde internet

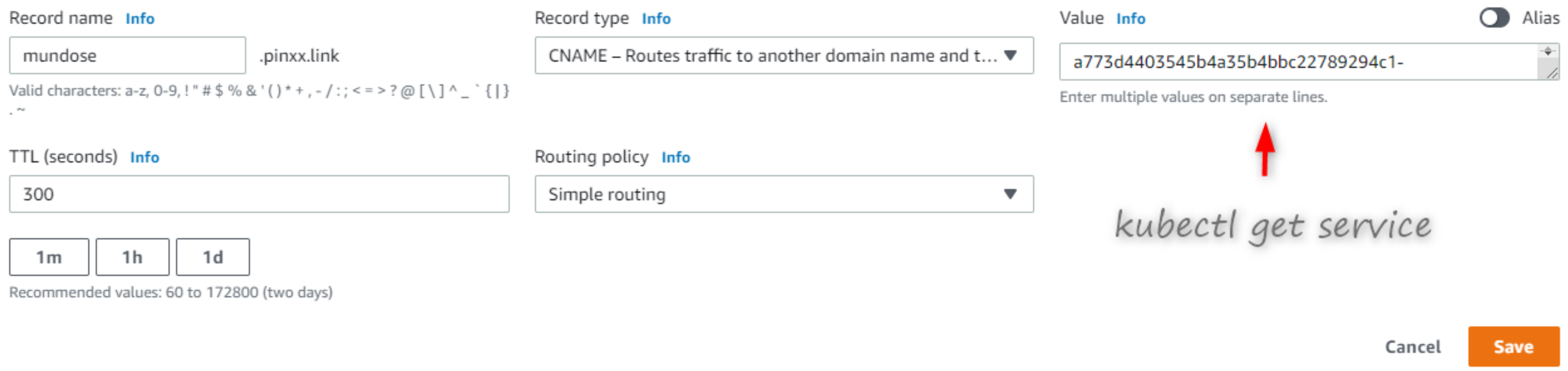
Configurar Route 53

Este punto de la guía es opcional ya que es necesario comprar un dominio (5 USD el más barato y un Certificado)

Crear Registro

Ir a Route 53 en la consola de aws

Hosted Zone > Tu dominio > Create Record



Record name Info

mundose .pinxx.link

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Record type Info

CNAME – Routes traffic to another domain name and t... ▼

Value Info

a773d4403545b4a35b4bbc22789294c1-

Enter multiple values on separate lines.

Alias

TTL (seconds) Info

300

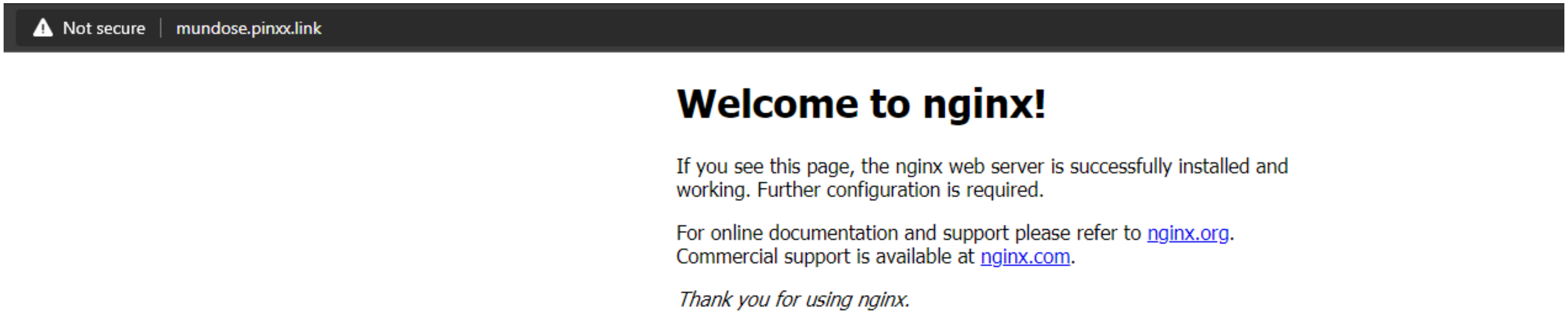
Recommended values: 60 to 172800 (two days)

Routing policy Info

Simple routing ▼

kubectl get service

Podemos ahora acceder a nuestra aplicación desplegada en kubernetes con una url amigable



Not secure | mundose.pinxx.link

Welcome to nginx!

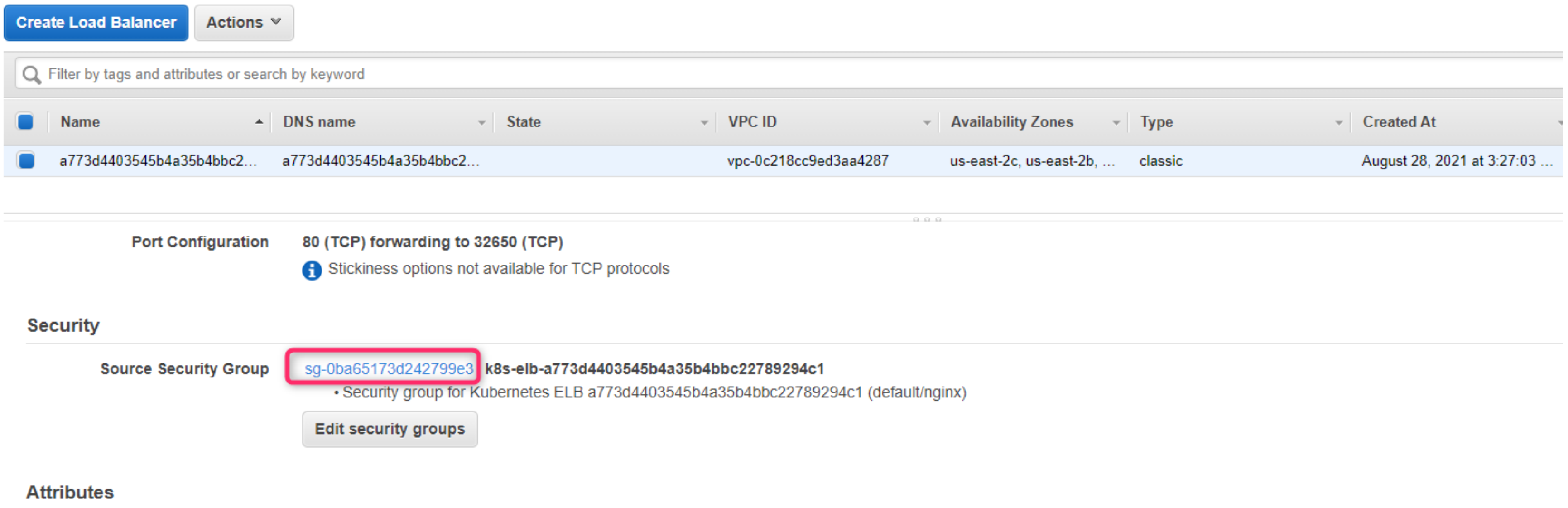
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

Configurar HTTPS

Ir a load balancer en la consola de EC2, luego a security y hacer click en el security group.



	Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
■	a773d4403545b4a35b4bbc2...	a773d4403545b4a35b4bbc2...		vpc-0c218cc9ed3aa4287	us-east-2c, us-east-2b, ...	classic	August 28, 2021 at 3:27:03 ...

Port Configuration 80 (TCP) forwarding to 32650 (TCP)

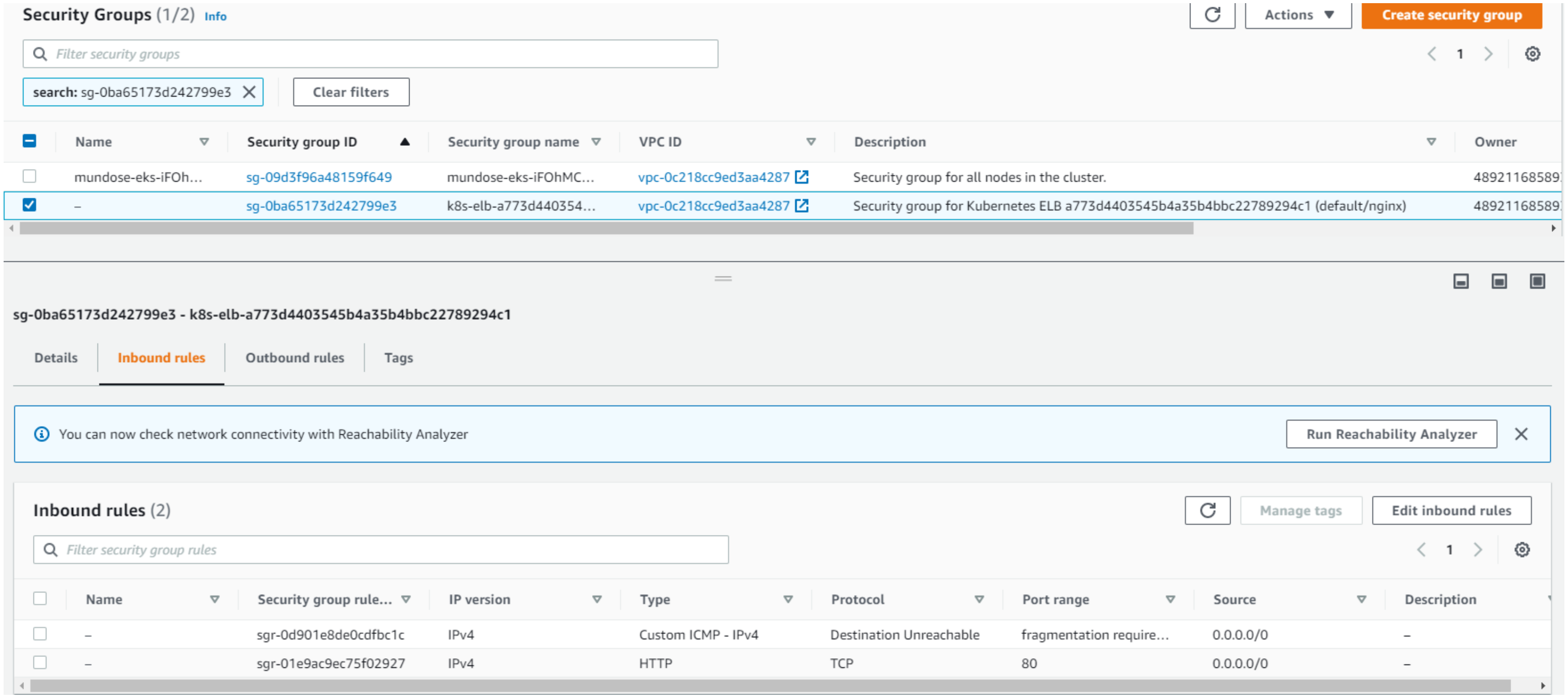
Stickiness options not available for TCP protocols

Security

Source Security Group sg-0ba65173d242799e3 **k8s-elb-a773d4403545b4a35b4bbc22789294c1**

- Security group for Kubernetes ELB a773d4403545b4a35b4bbc22789294c1 (default/nginx)

Attributes



Security Groups (1/2) Info

search: sg-0ba65173d242799e3

	Name	Security group ID	Security group name	VPC ID	Description	Owner
□	mundose-eks-iFOh…	sg-09d3f96a48159f649	mundose-eks-iFOhMC…	vpc-0c218cc9ed3aa4287	Security group for all nodes in the cluster.	48921168589
☑	–	sg-0ba65173d242799e3	k8s-elb-a773d440354…	vpc-0c218cc9ed3aa4287	Security group for Kubernetes ELB a773d4403545b4a35b4bbc22789294c1 (default/nginx)	48921168589

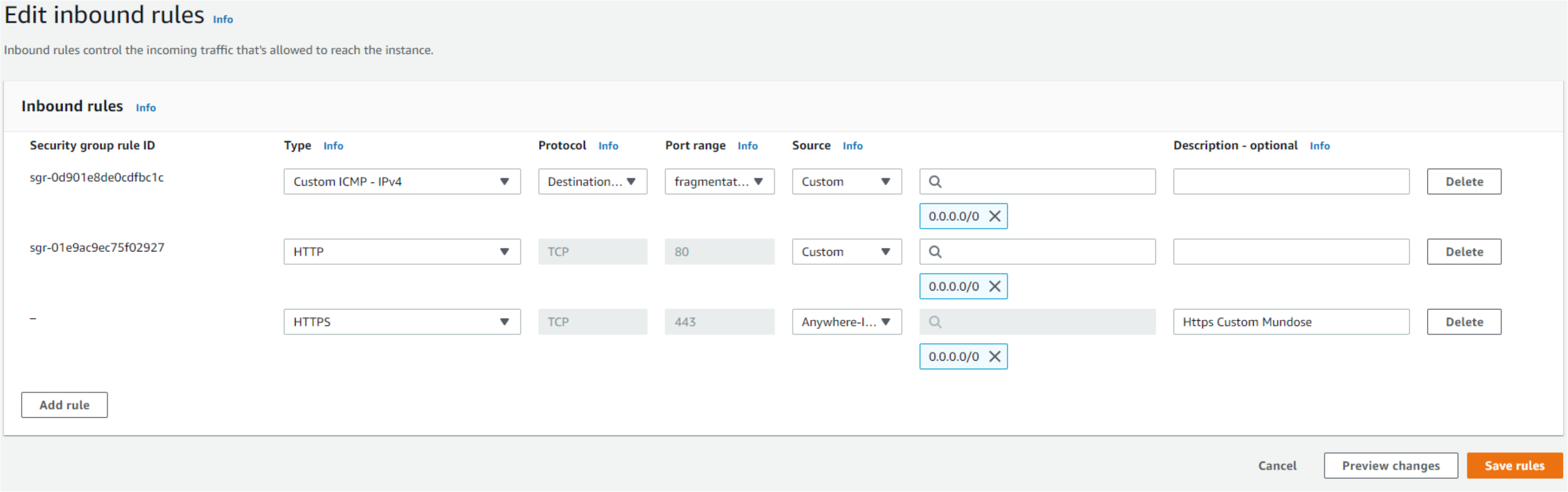
sg-0ba65173d242799e3 - k8s-elb-a773d4403545b4a35b4bbc22789294c1

You can now check network connectivity with Reachability Analyzer

Inbound rules (2)

	Name	Security group rule…	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-0d901e8de0cdfbc1c	IPv4	Custom ICMP - IPv4	Destination Unreachable	fragmentation require…	0.0.0.0/0	–
□	–	sgr-01e9ac9ec75f02927	IPv4	HTTP	TCP	80	0.0.0.0/0	–

Editar inbound rules



Edit inbound rules Info

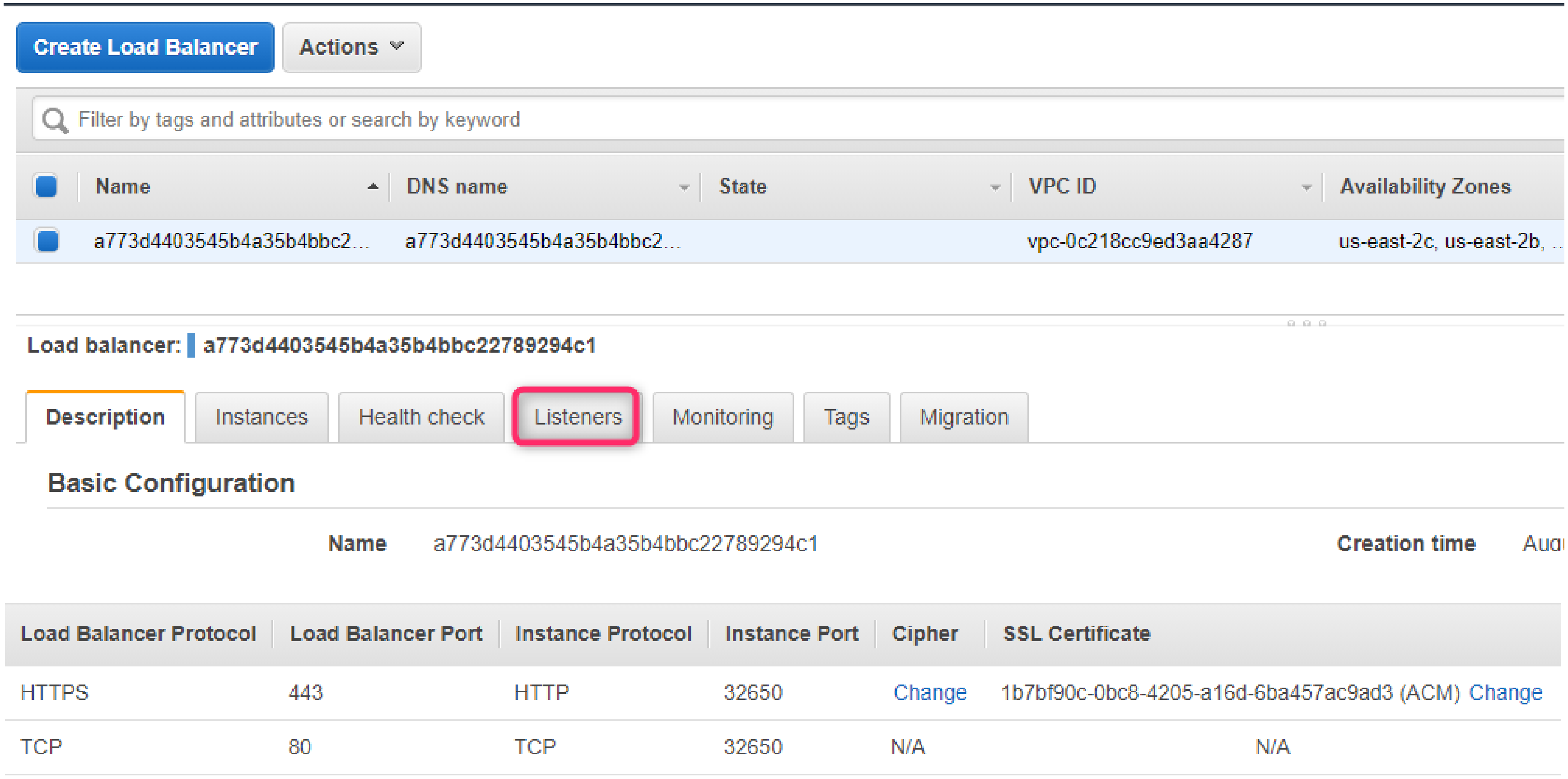
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0d901e8de0cdfbc1c	Custom ICMP - IPv4	Destination…	fragmentat…	Custom	0.0.0.0/0		Delete
sgr-01e9ac9ec75f02927	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
–	HTTPS	TCP	443	Anywhere-I…	0.0.0.0/0	Https Custom Mundose	Delete

Configurar listener y solicitar certificado

Nota: El pedido del certificado puede tardar 30 minutos en procesarse



Name	DNS name	State	VPC ID	Availability Zones
a773d4403545b4a35b4bbc2...	a773d4403545b4a35b4bbc2...		vpc-0c218cc9ed3aa4287	us-east-2c, us-east-2b, ..

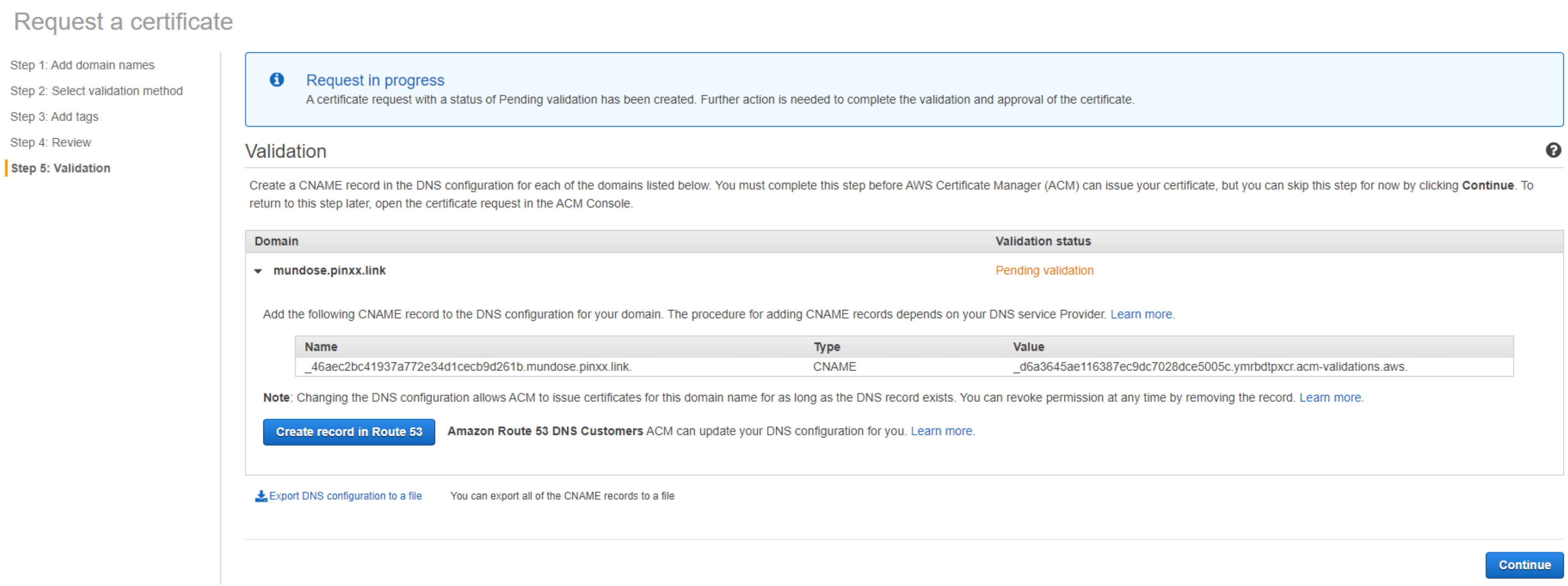
Load balancer: a773d4403545b4a35b4bbc22789294c1

Basic Configuration

Name a773d4403545b4a35b4bbc22789294c1

Creation time Aug

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Cipher	SSL Certificate
HTTPS	443	HTTP	32650	Change	1b7bf90c-0bc8-4205-a16d-6ba457ac9ad3 (ACM) Change
TCP	80	TCP	32650	N/A	N/A



Request a certificate

Request in progress

A certificate request with a status of Pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking **Continue**. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
▾ **mundose.pinxx.link**	Pending validation

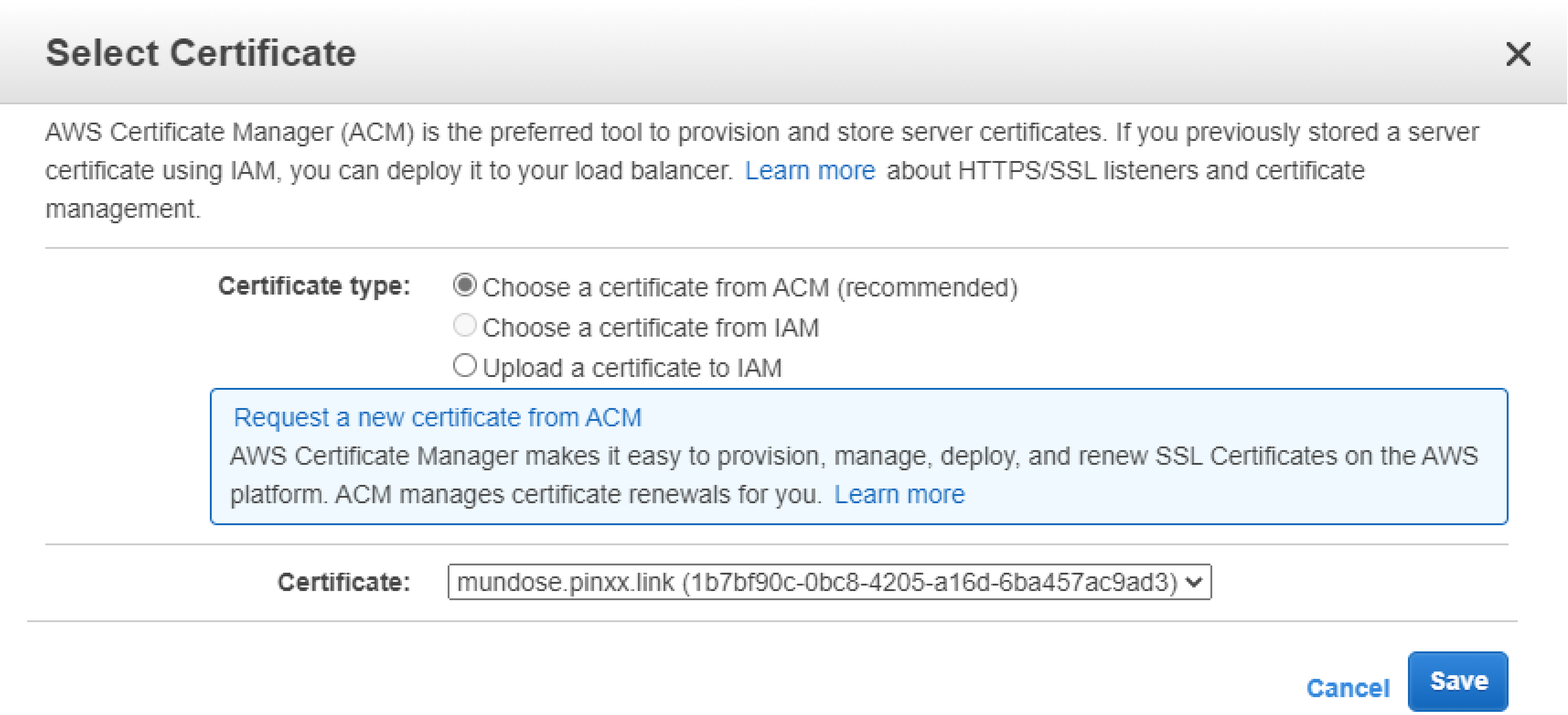
Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. Learn more.

Name	Type	Value
_46aec2bc41937a772e34d1cecb9d261b.mundose.pinxx.link.	CNAME	_d6a3645ae116387ec9dc7028dce5005c.ymrbdtpxcr.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. Learn more.

Amazon Route 53 DNS Customers ACM can update your DNS configuration for you. Learn more.

You can export all of the CNAME records to a file



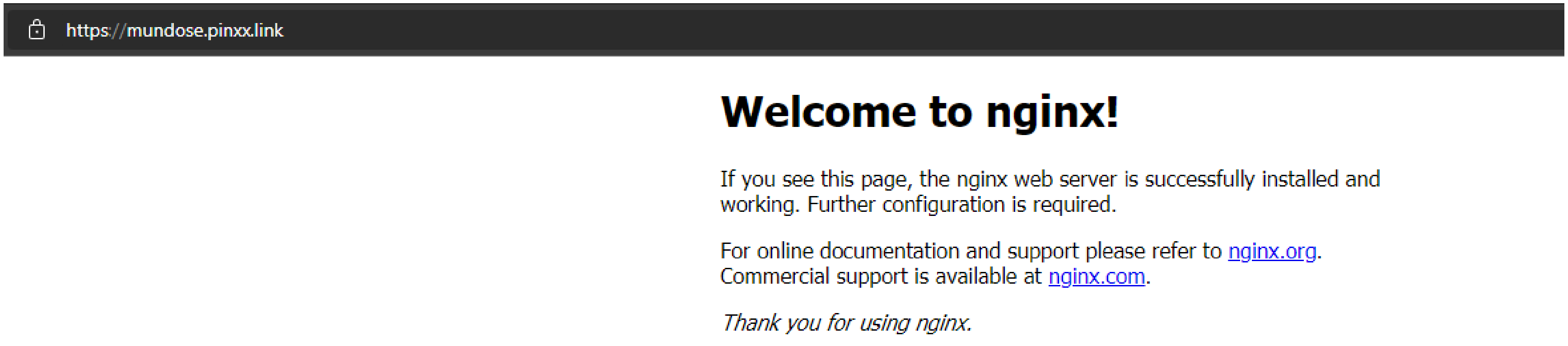
Select Certificate

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more about HTTPS/SSL listeners and certificate management.

Certificate type:
● Choose a certificate from ACM (recommended)
○ Choose a certificate from IAM
○ Upload a certificate to IAM

Request a new certificate from ACM
AWS Certificate Manager makes it easy to provision, manage, deploy, and renew SSL Certificates on the AWS platform. ACM manages certificate renewals for you. Learn more

Certificate: mundose.pinxx.link (1b7bf90c-0bc8-4205-a16d-6ba457ac9ad3)



https://mundose.pinxx.link

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.

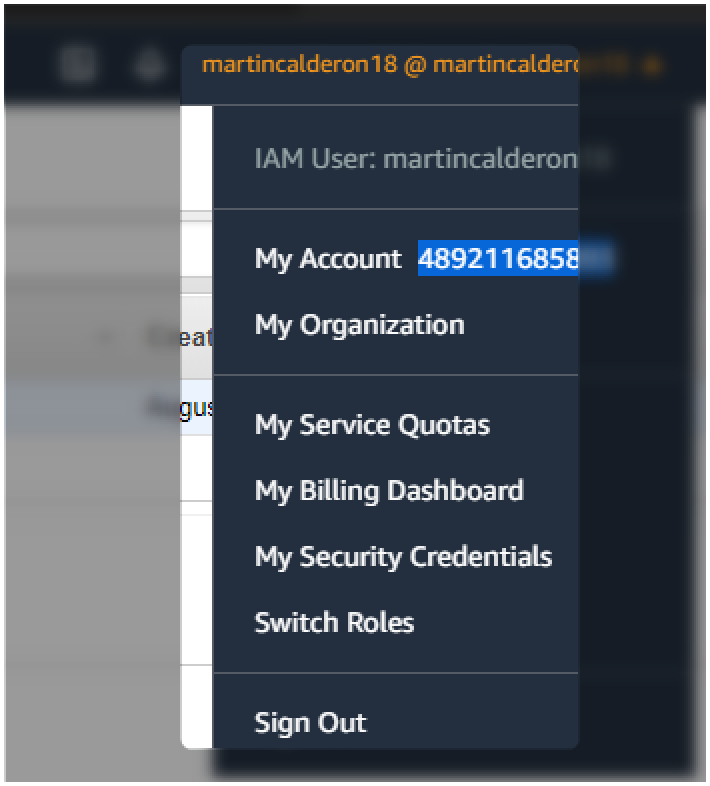
Instalar herramientas de Monitoreo

Stack de EFK

- ElasticSearch
- Fluentbit
- Kibana

Configurar variables de entorno

Capturamos el account id de nuestra cuenta de aws



Definir la región por defecto en una variable de ambiente

```
export AWS_REGION='us-east-2'
```

Definir el id de la cuenta en una variable de ambiente

```
export ACCOUNT_ID=489211685
```

Definir el nombre de dominio para el cluster de Elasticsearch

```
export ES_DOMAIN_NAME="eksworkshop-logging"
```

Elasticsearch version

```
export ES_VERSION="7.4"
```

kibana admin user

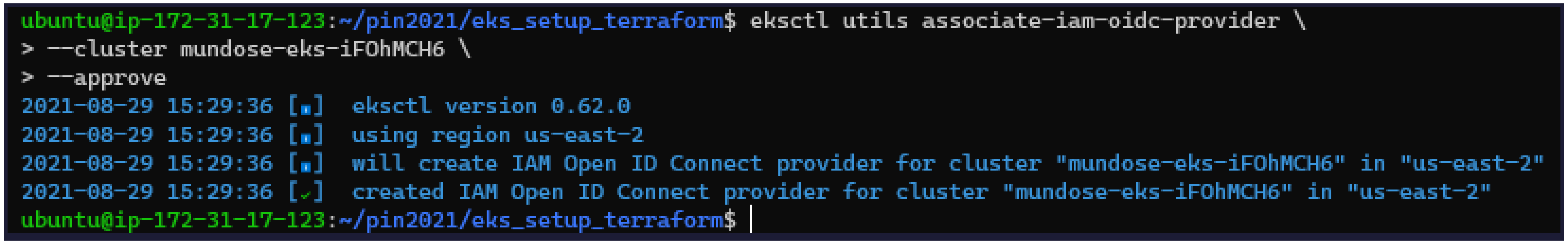
```
export ES_DOMAIN_USER="eksworkshop"
```

kibana admin password

```
export ES_DOMAIN_PASSWORD="$(openssl rand -base64 12)_Ek1$"
```

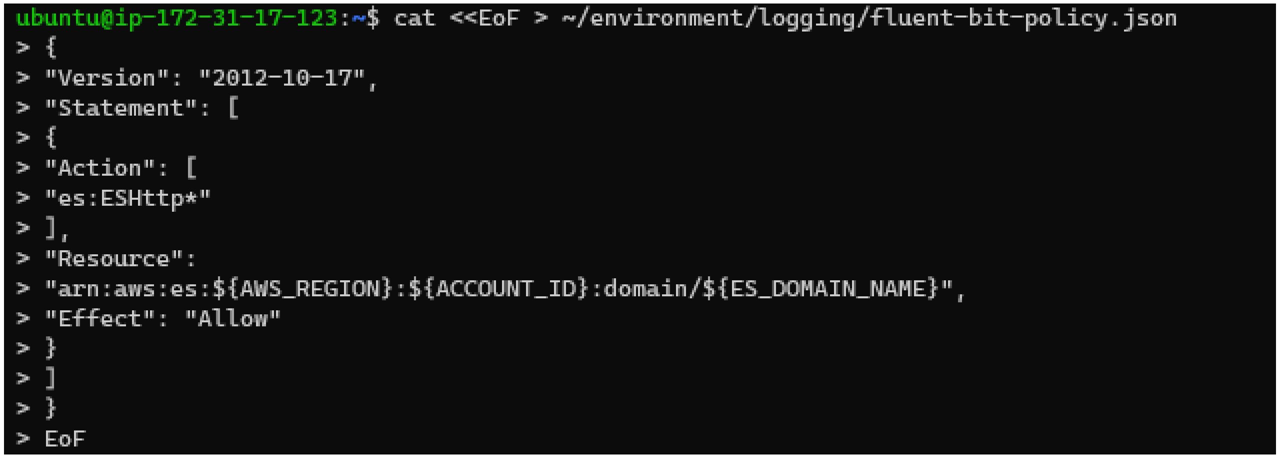
Configurar OpenID Connect

```
eksctl utils associate-iam-oidc-provider \
--cluster mundose-eks-iFOhMCH6 \
--approve
```

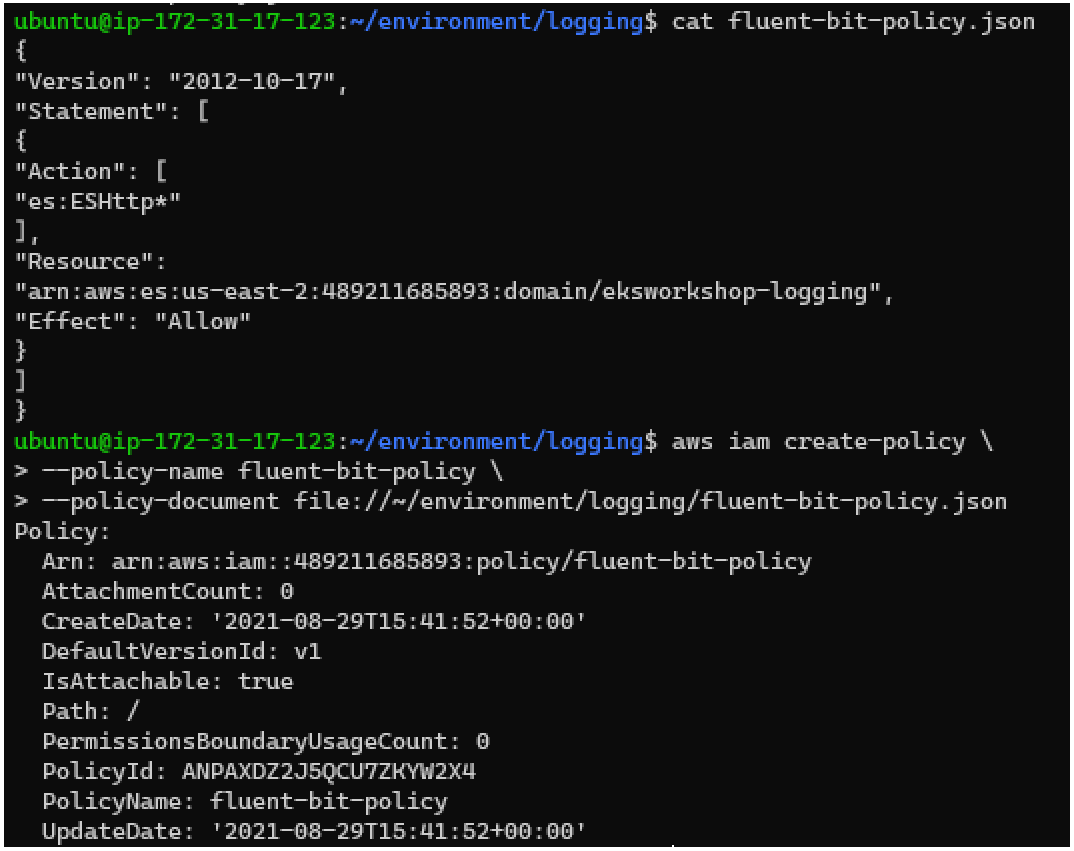


```
ubuntu@ip-172-31-17-123:~/pin2021/eks_setup_terraform$ eksctl utils associate-iam-oidc-provider \
> --cluster mundose-eks-iFOhMCH6 \
> --approve
2021-08-29 15:29:36 [ℹ]  eksctl version 0.62.0
2021-08-29 15:29:36 [ℹ]  using region us-east-2
2021-08-29 15:29:36 [ℹ]  will create IAM Open ID Connect provider for cluster "mundose-eks-iFOhMCH6" in "us-east-2"
2021-08-29 15:29:36 [✔]  created IAM Open ID Connect provider for cluster "mundose-eks-iFOhMCH6" in "us-east-2"
ubuntu@ip-172-31-17-123:~/pin2021/eks_setup_terraform$
```

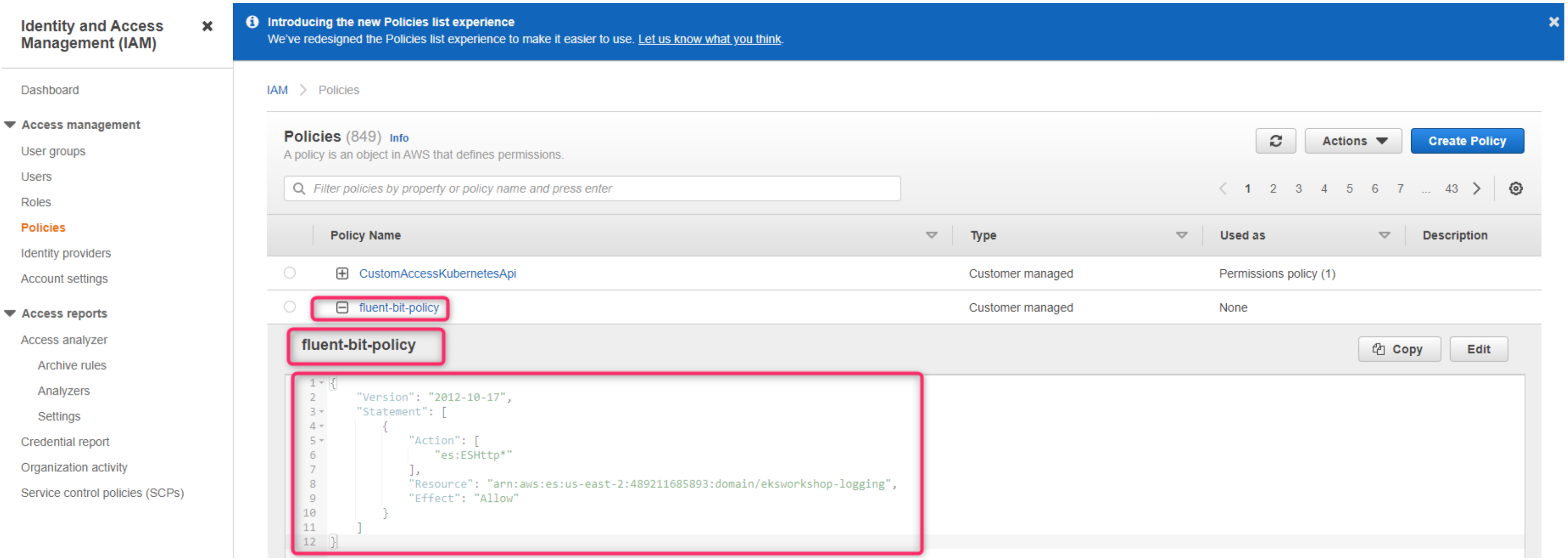
Crear IAM policy con AWS CLI



```
ubuntu@ip-172-31-17-123:~$ cat <<EoF > ~/environment/logging/fluent-bit-policy.json
> {
> "Version": "2012-10-17",
> "Statement": [
> {
> "Action": [
> "es:ESHttp*"
> ],
> "Resource":
> "arn:aws:es:${AWS_REGION}:${ACCOUNT_ID}:domain/${ES_DOMAIN_NAME}",
> "Effect": "Allow"
> }
> ]
> }
> EoF
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ cat fluent-bit-policy.json
{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
"es:ESHttp*"
],
"Resource":
"arn:aws:es:us-east-2:489211685893:domain/eksworkshop-logging",
"Effect": "Allow"
}
]
}
ubuntu@ip-172-31-17-123:~/environment/logging$ aws iam create-policy \
> --policy-name fluent-bit-policy \
> --policy-document file://~/environment/logging/fluent-bit-policy.json
Policy:
  Arn: arn:aws:iam::489211685893:policy/fluent-bit-policy
  AttachmentCount: 0
  CreateDate: '2021-08-29T15:41:52+00:00'
  DefaultVersionId: v1
  IsAttachable: true
  Path: /
  PermissionsBoundaryUsageCount: 0
  PolicyId: ANPAXDZ2J5QCU7ZKYW2X4
  PolicyName: fluent-bit-policy
  UpdateDate: '2021-08-29T15:41:52+00:00'
```



Introducing the new Policies list experience
We've redesigned the Policies list experience to make it easier to use. Let us know what you think.

Policies (849) Info
A policy is an object in AWS that defines permissions.

Policy Name	Type	Used as	Description
CustomAccessKubernetesApi	Customer managed	Permissions policy (1)	
fluent-bit-policy	Customer managed	None	

fluent-bit-policy

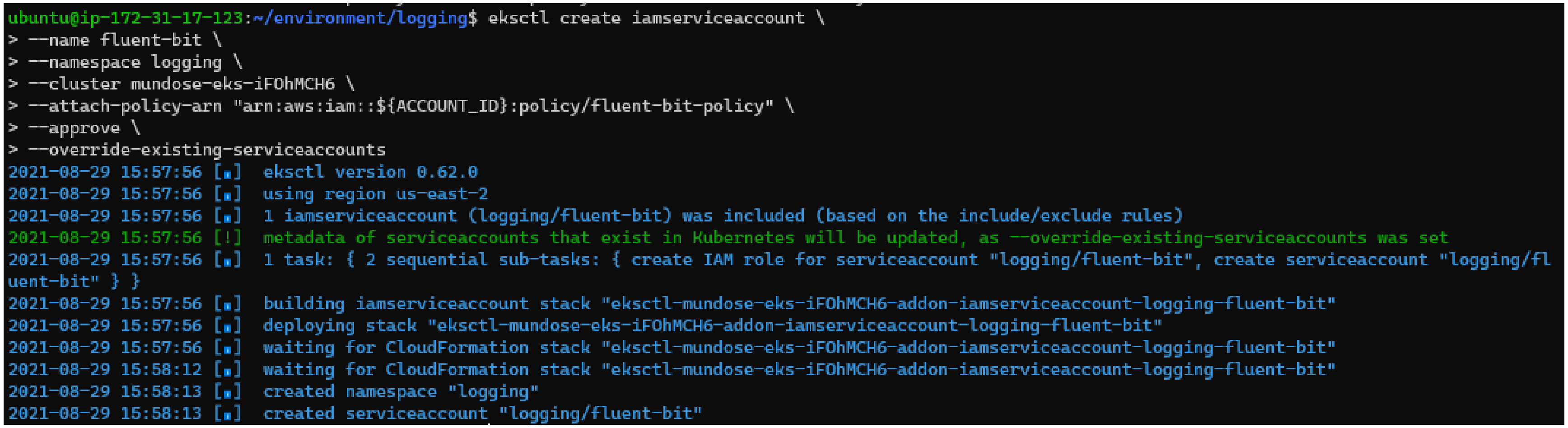
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:ESHttp*"
            ],
            "Resource": "arn:aws:es:us-east-2:489211685893:domain/eksworkshop-logging",
            "Effect": "Allow"
        }
    ]
}
```

Crear el namespace de logging

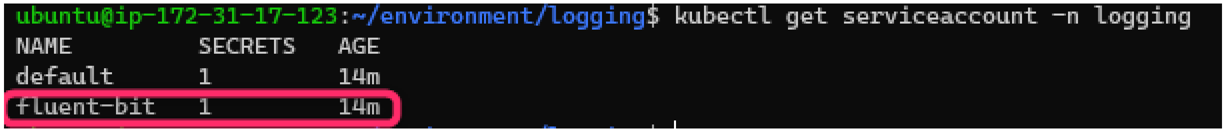
```
kubectl create namespace logging
```

Crear cuenta de servicio

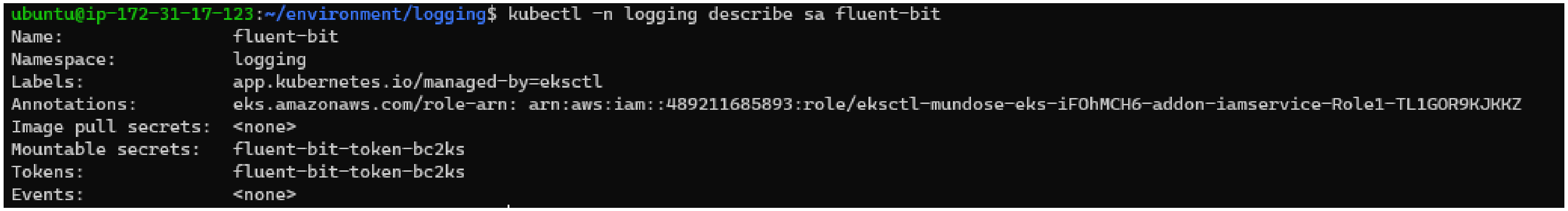
```
eksctl create iamserviceaccount \
--name fluent-bit \
--namespace logging \
--cluster mundose-eks-iFOhMCH6 \
--attach-policy-arn "arn:aws:iam::${ACCOUNT_ID}:policy/fluent-bit-policy" \
--approve \
--override-existing-serviceaccounts
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ eksctl create iamserviceaccount \
> --name fluent-bit \
> --namespace logging \
> --cluster mundose-eks-iFOhMCH6 \
> --attach-policy-arn "arn:aws:iam::${ACCOUNT_ID}:policy/fluent-bit-policy" \
> --approve \
> --override-existing-serviceaccounts
2021-08-29 15:57:56 [ℹ]  eksctl version 0.62.0
2021-08-29 15:57:56 [ℹ]  using region us-east-2
2021-08-29 15:57:56 [ℹ]  1 iamserviceaccount (logging/fluent-bit) was included (based on the include/exclude rules)
2021-08-29 15:57:56 [!]  metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2021-08-29 15:57:56 [ℹ]  1 task: { 2 sequential sub-tasks: { create IAM role for serviceaccount "logging/fluent-bit", create serviceaccount "logging/fl
uent-bit" } }
2021-08-29 15:57:56 [ℹ]  building iamserviceaccount stack "eksctl-mundose-eks-iFOhMCH6-addon-iamserviceaccount-logging-fluent-bit"
2021-08-29 15:57:56 [ℹ]  deploying stack "eksctl-mundose-eks-iFOhMCH6-addon-iamserviceaccount-logging-fluent-bit"
2021-08-29 15:57:56 [ℹ]  waiting for CloudFormation stack "eksctl-mundose-eks-iFOhMCH6-addon-iamserviceaccount-logging-fluent-bit"
2021-08-29 15:58:12 [ℹ]  waiting for CloudFormation stack "eksctl-mundose-eks-iFOhMCH6-addon-iamserviceaccount-logging-fluent-bit"
2021-08-29 15:58:13 [ℹ]  created namespace "logging"
2021-08-29 15:58:13 [ℹ]  created serviceaccount "logging/fluent-bit"
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ kubectl get serviceaccount -n logging
NAME         SECRETS   AGE
default      1         14m
fluent-bit   1         14m
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ kubectl -n logging describe sa fluent-bit
Name:                fluent-bit
Namespace:           logging
Labels:              app.kubernetes.io/managed-by=eksctl
Annotations:         eks.amazonaws.com/role-arn: arn:aws:iam::489211685893:role/eksctl-mundose-eks-iFOhMCH6-addon-iamservice-Role1-TL1GOR9KJKKZ
Image pull secrets:  <none>
Mountable secrets:   fluent-bit-token-bc2ks
Tokens:              fluent-bit-token-bc2ks
Events:              <none>
```

Crear Cluster de Elastic

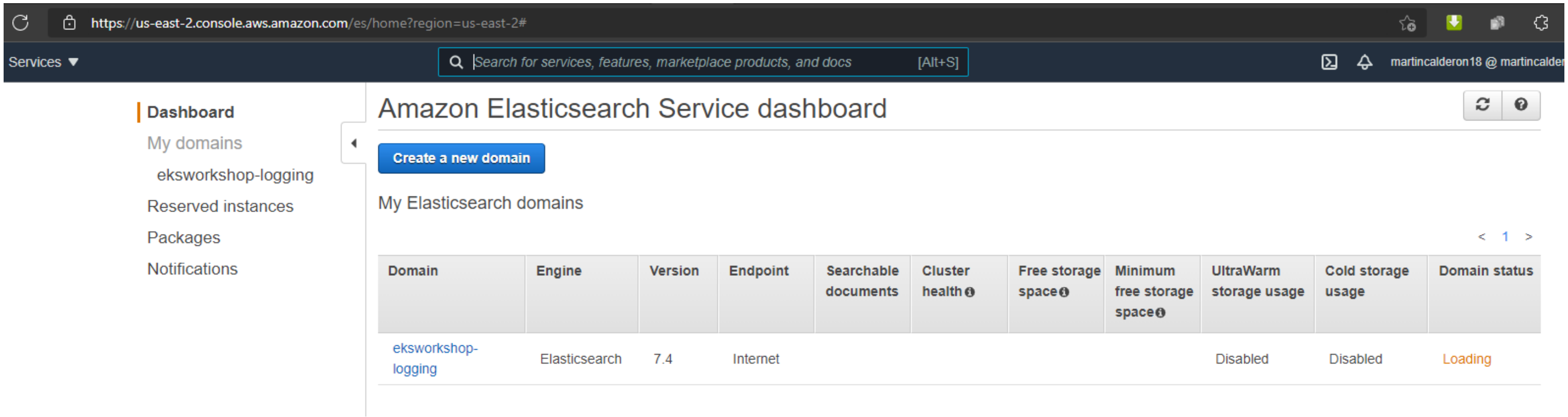
Esto puede tomar hasta 30 minutos

Descargar y actualizar el template usando las variables definidas previamente

```
curl -sS https://www.eksworkshop.com/intermediate/230_logging/deploy.files/es_domain.json \
 | envsubst > ~/environment/logging/es_domain.json
```

Crear el cluster de Elastic

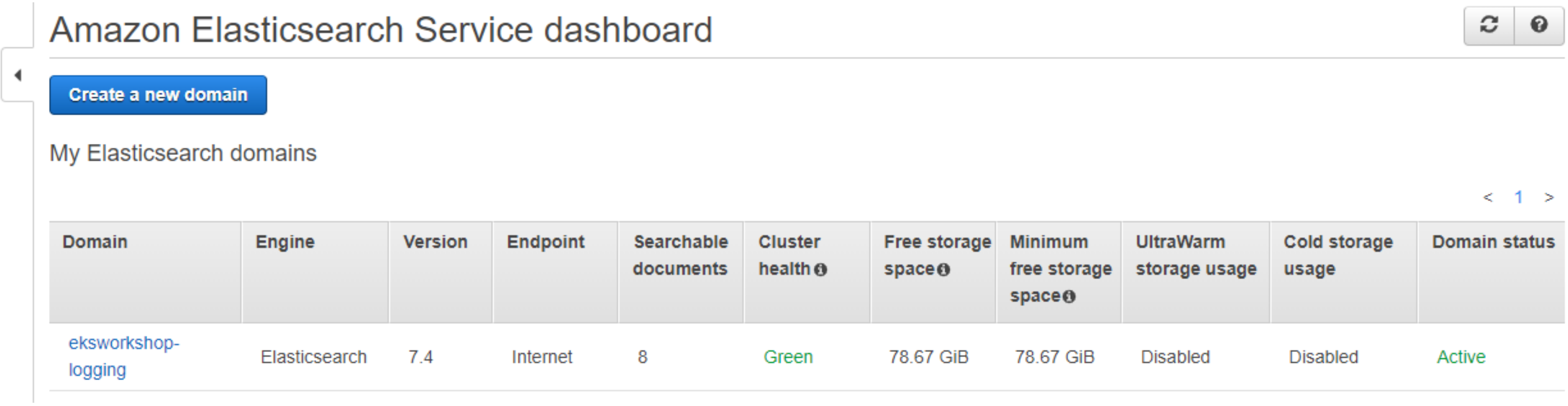
```
aws es create-elasticsearch-domain \
 --cli-input-json  file://~/environment/logging/es_domain.json
```


https://us-east-2.console.aws.amazon.com/es/home?region=us-east-2#
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
martincalderon18 @ martincalder
Dashboard
My domains
eksworkshop-logging
Reserved instances
Packages
Notifications
Create a new domain
&lt; 1 &gt;

Amazon Elasticsearch Service dashboard

My Elasticsearch domains

Domain	Engine	Version	Endpoint	Searchable documents	Cluster health	Free storage space	Minimum free storage space	UltraWarm storage usage	Cold storage usage	Domain status
eksworkshop-logging	Elasticsearch	7.4	Internet					Disabled	Disabled	Loading



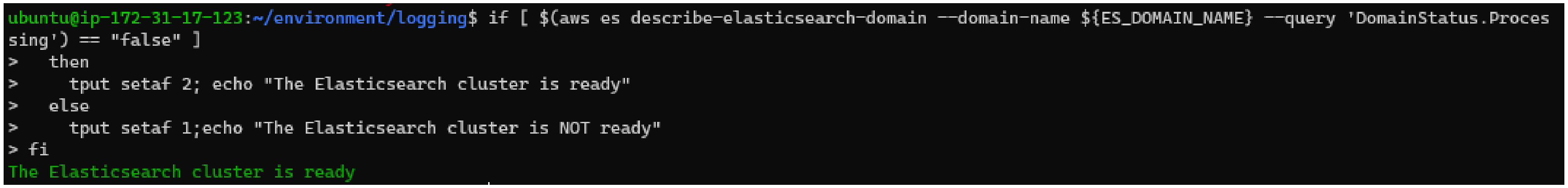
Amazon Elasticsearch Service dashboard

My Elasticsearch domains

Domain	Engine	Version	Endpoint	Searchable documents	Cluster health	Free storage space	Minimum free storage space	UltraWarm storage usage	Cold storage usage	Domain status
eksworkshop-logging	Elasticsearch	7.4	Internet	8	Green	78.67 GiB	78.67 GiB	Disabled	Disabled	Active

También podemos usar el shell para comprobarlo

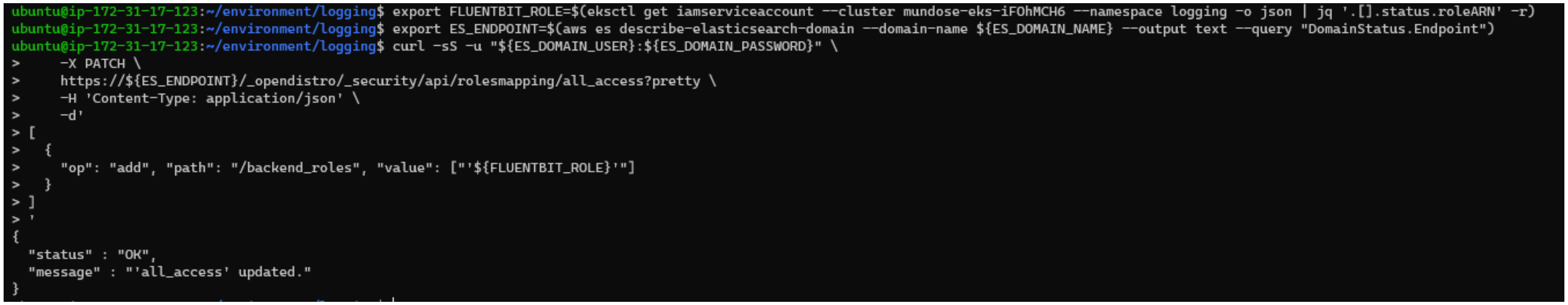
```
if [ $(aws es describe-elasticsearch-domain --domain-name ${ES_DOMAIN_NAME} --query
'DomainStatus.Processing') == "false" ]
 then
  tput setaf 2; echo "The Elasticsearch cluster is ready"
 else
  tput setaf 1;echo "The Elasticsearch cluster is NOT ready"
fi
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ if [ $(aws es describe-elasticsearch-domain --domain-name ${ES_DOMAIN_NAME} --query 'DomainStatus.Proces
sing') == "false" ]
>   then
>     tput setaf 2; echo "The Elasticsearch cluster is ready"
>   else
>     tput setaf 1;echo "The Elasticsearch cluster is NOT ready"
> fi
The Elasticsearch cluster is ready
```

Configurar Acceso ElasticSearch

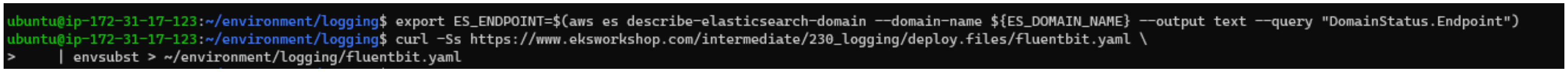
Corremos los siguiente comandos para configurar el acceso a ElasticSearch [Configure Elastic Access Repo]



```
ubuntu@ip-172-31-17-123:~/environment/logging$ export FLUENTBIT_ROLE=$(eksctl get iamserviceaccount --cluster mundose-eks-iFOhMCH6 --namespace logging -o json | jq '.[].status.roleARN' -r)
ubuntu@ip-172-31-17-123:~/environment/logging$ export ES_ENDPOINT=$(aws es describe-elasticsearch-domain --domain-name ${ES_DOMAIN_NAME} --output text --query "DomainStatus.Endpoint")
ubuntu@ip-172-31-17-123:~/environment/logging$ curl -sS -u "${ES_DOMAIN_USER}:${ES_DOMAIN_PASSWORD}" \
>     -X PATCH \
>     https://${ES_ENDPOINT}/_opendistro/_security/api/rolesmapping/all_access?pretty \
>     -H 'Content-Type: application/json' \
>     -d'
> [
>   {
>     "op": "add", "path": "/backend_roles", "value": ["'${FLUENTBIT_ROLE}'"]
>   }
> ]
> '
{
  "status" : "OK",
  "message" : "'all_access' updated."
}
```

Crear Despliegue Fluent Bit

Corremos los siguientes comando para crear el archivo de deployment de fluentbit [Generate Deployment file for Fluent Bit repo]

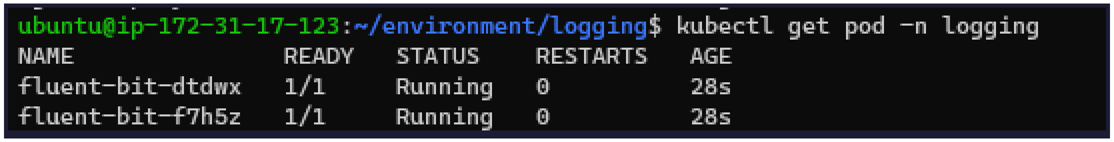


```
ubuntu@ip-172-31-17-123:~/environment/logging$ export ES_ENDPOINT=$(aws es describe-elasticsearch-domain --domain-name ${ES_DOMAIN_NAME} --output text --query "DomainStatus.Endpoint")
ubuntu@ip-172-31-17-123:~/environment/logging$ curl -Ss https://www.eksworkshop.com/intermediate/230_logging/deploy.files/fluentbit.yaml \
>     | envsubst > ~/environment/logging/fluentbit.yaml
```

Desplegar Fluent Bit

kubectl apply -f ~/environment/logging/fluentbit.yaml

kubectl --namespace=logging get pods



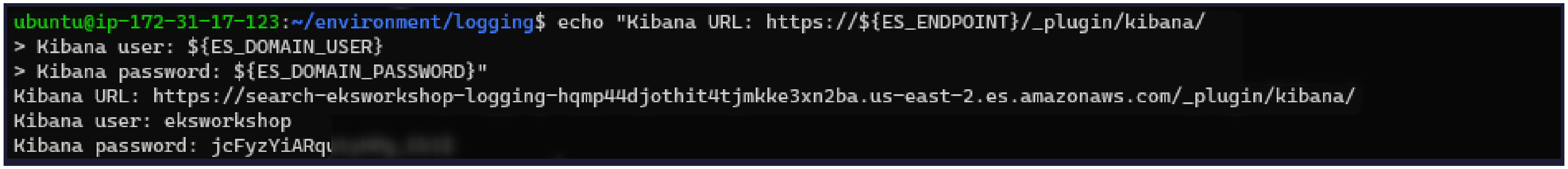
```
ubuntu@ip-172-31-17-123:~/environment/logging$ kubectl get pod -n logging
NAME                READY   STATUS    RESTARTS   AGE
fluent-bit-dtdwx    1/1     Running   0          28s
fluent-bit-f7h5z    1/1     Running   0          28s
```

En este punto FluentBit se desplegó de manera exitosa

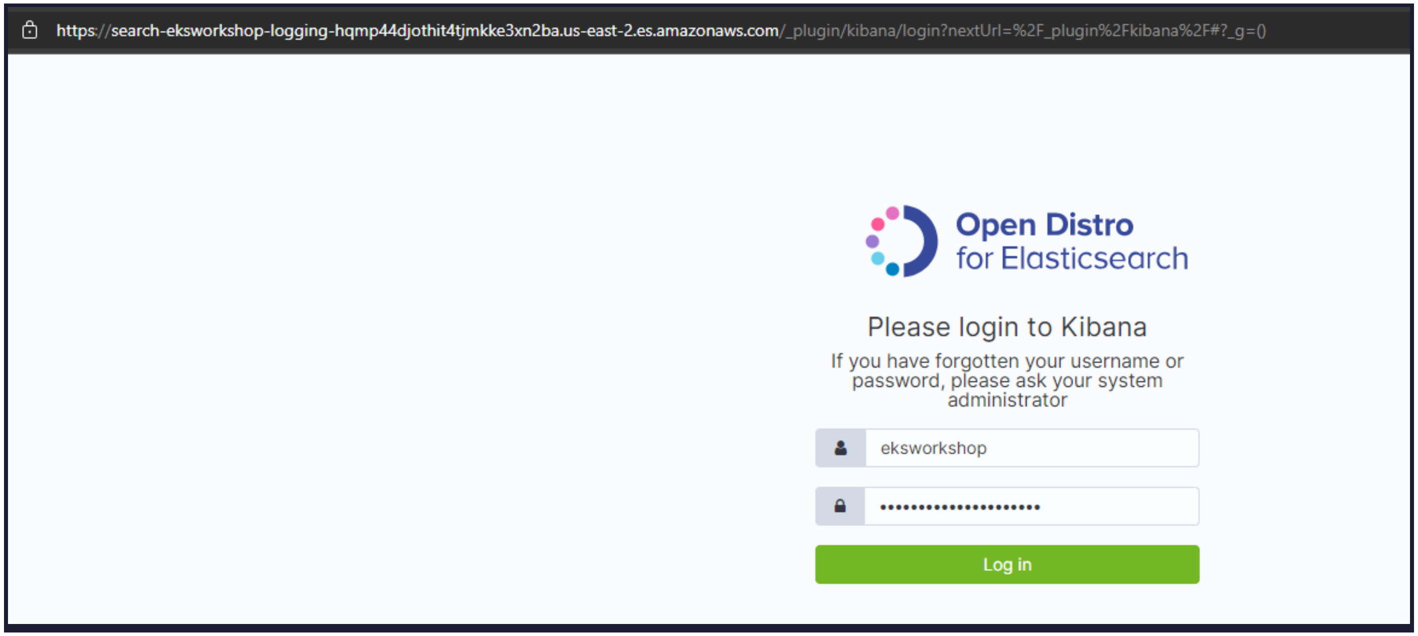
Kibana configuración

En la consola de EC2 corremos los siguientes comandos para obtener la información necesaria

```
echo "Kibana URL: https://${ES_ENDPOINT}/_plugin/kibana/
Kibana user: ${ES_DOMAIN_USER}
Kibana password: ${ES_DOMAIN_PASSWORD}"
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ echo "Kibana URL: https://${ES_ENDPOINT}/_plugin/kibana/
> Kibana user: ${ES_DOMAIN_USER}
> Kibana password: ${ES_DOMAIN_PASSWORD}"
Kibana URL: https://search-eksworkshop-logging-hqmp44djothit4tjmkke3xn2ba.us-east-2.es.amazonaws.com/_plugin/kibana/
Kibana user: eksworkshop
Kibana password: jcFyzYiARq[redacted]
```



https://search-eksworkshop-logging-hqmp44djothit4tjmkke3xn2ba.us-east-2.es.amazonaws.com/_plugin/kibana/login?nextUrl=%2F_plugin%2Fkibana%2F#?_g=()

Open Distro for Elasticsearch

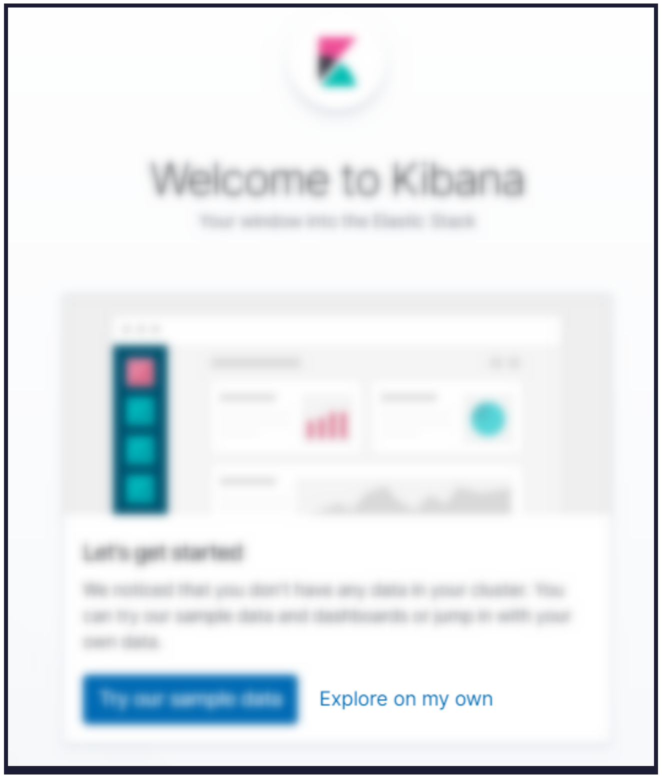
Please login to Kibana

If you have forgotten your username or password, please ask your system administrator

eksworkshop

•••••••••••••••••••••

Seleccionar Explore on my own y luego connect to your Elasticsearch index

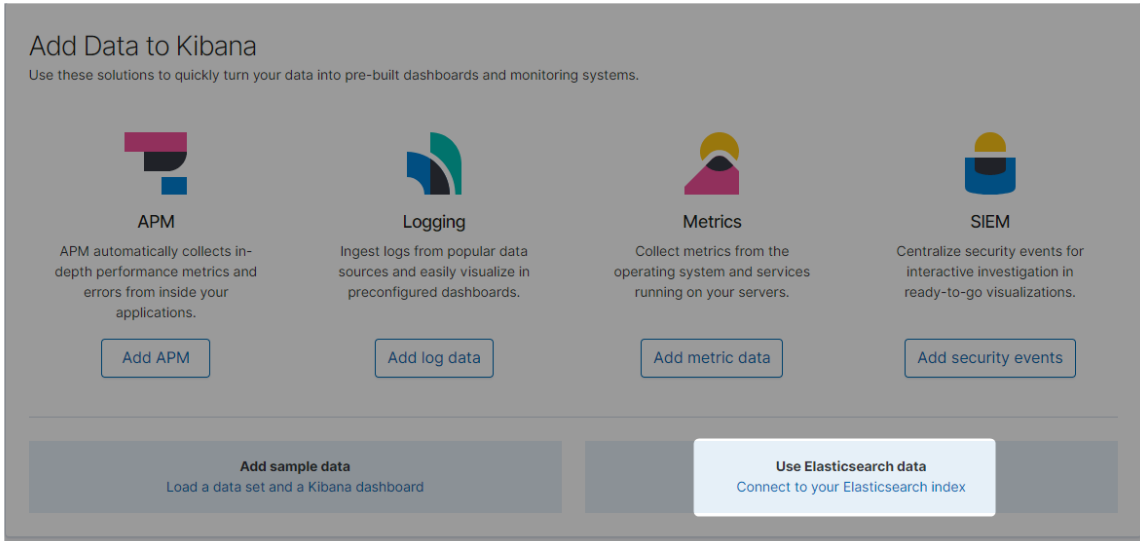


Welcome to Kibana

Your window into the Elastic Stack

Let's get started

We noticed that you don't have any data in your cluster. You can try our sample data and dashboards or jump in with your own data.



Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Metrics

Collect metrics from the operating system and services running on your servers.

SIEM

Centralize security events for interactive investigation in ready-to-go visualizations.

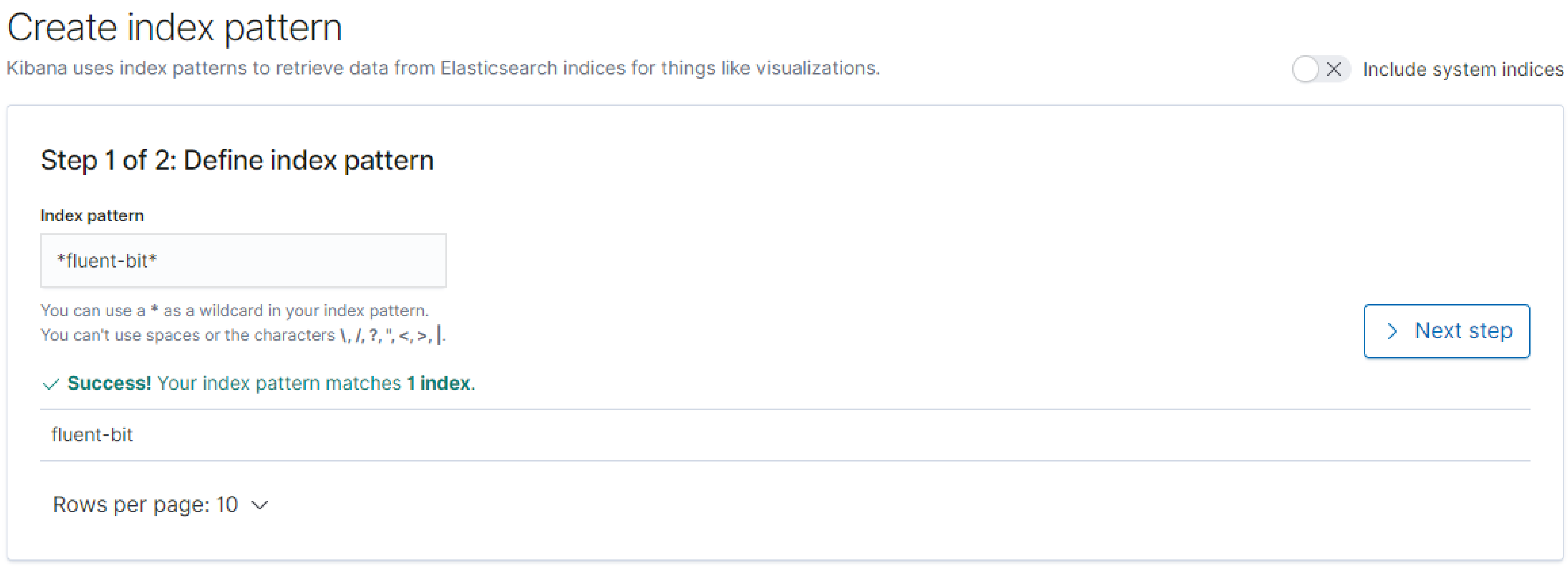
Add sample data

Load a data set and a Kibana dashboard

Use Elasticsearch data

Connect to your Elasticsearch index

Crear index



Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Step 1 of 2: Define index pattern

Index pattern

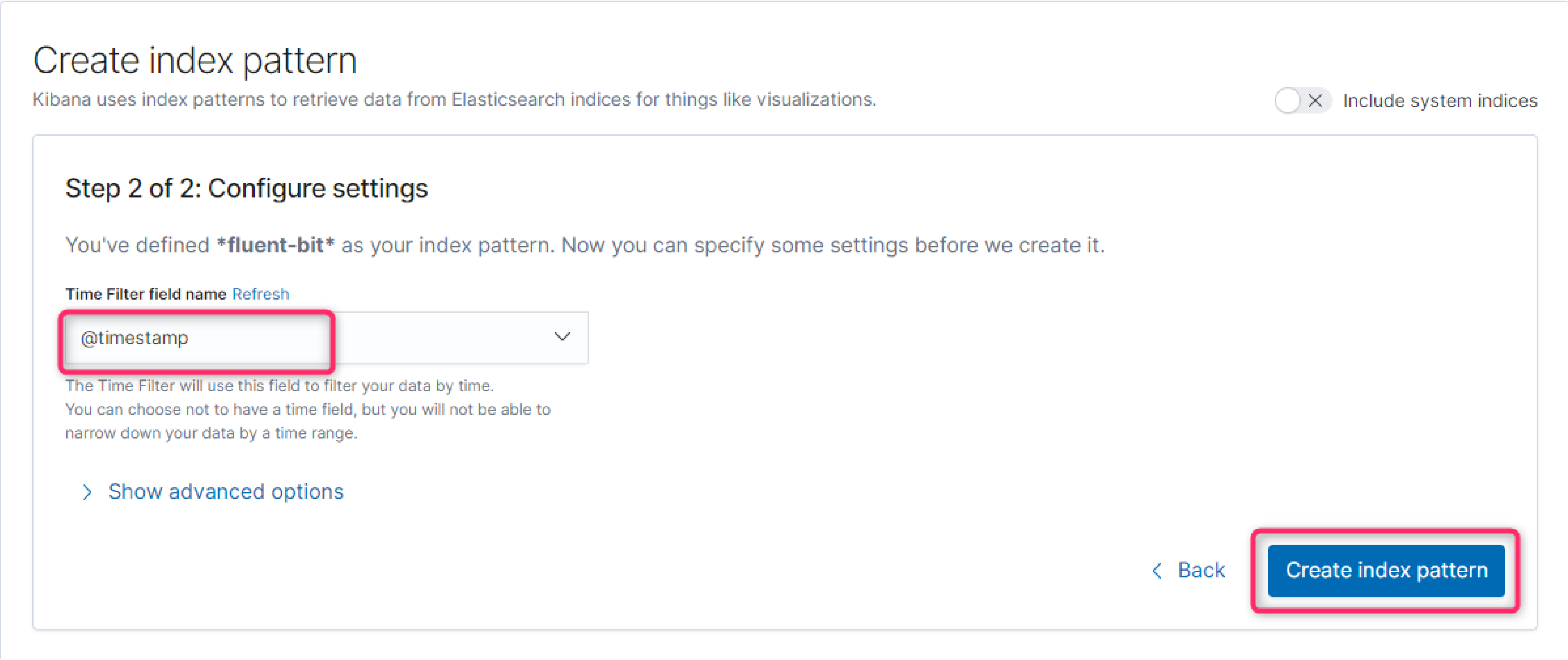
fluent-bit

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ **Success!** Your index pattern matches **1 index**.

fluent-bit

Rows per page: 10



Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Include system indices

Step 2 of 2: Configure settings

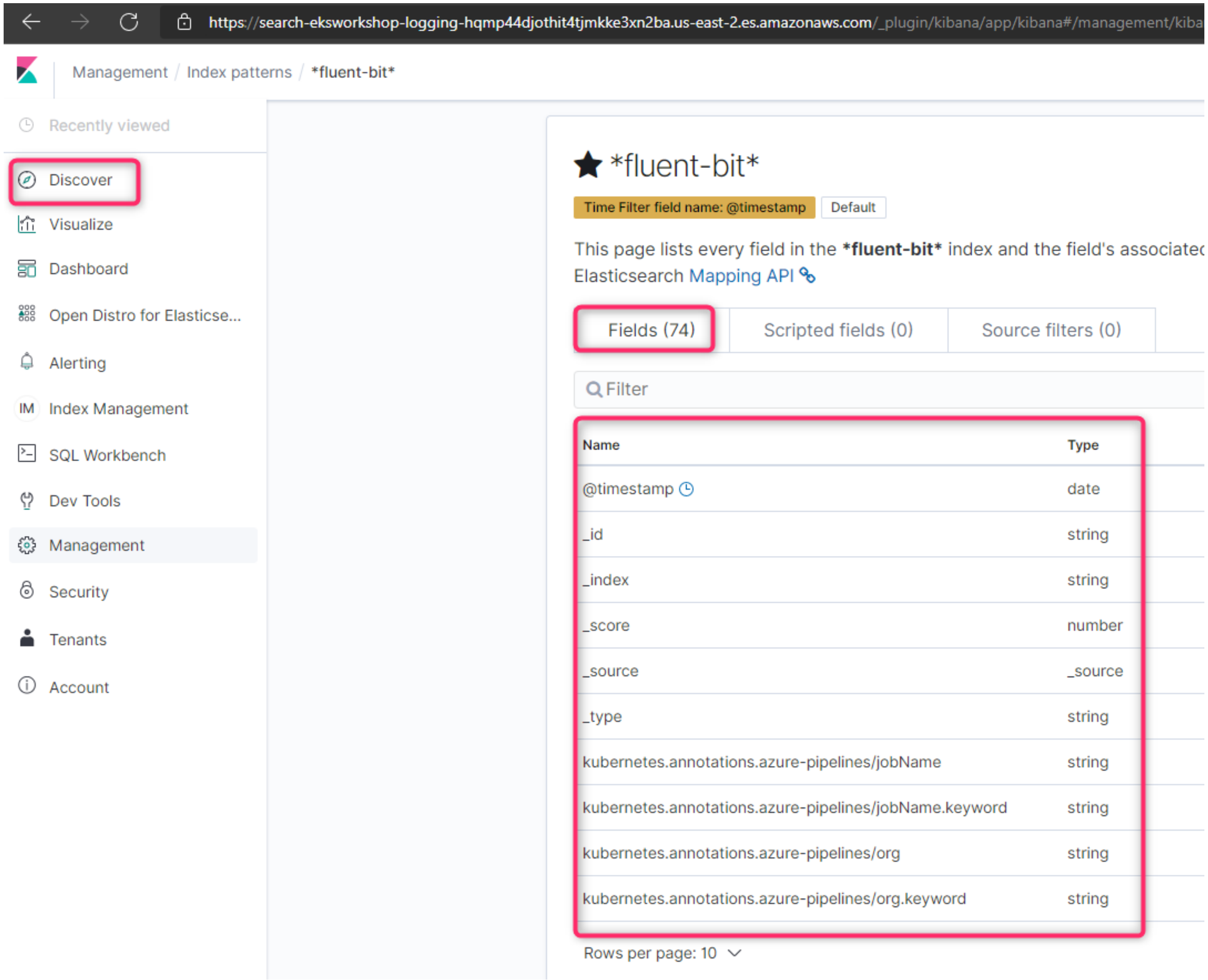
You've defined ***fluent-bit*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

@timestamp

The Time Filter will use this field to filter your data by time. You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

Finalizar & Discover



https://search-eksworkshop-logging-hqmp44djothit4tjmkke3xn2ba.us-east-2.es.amazonaws.com/_plugin/kibana/app/kibana#/management/kiba

Management / Index patterns / *fluent-bit*

★ *fluent-bit*

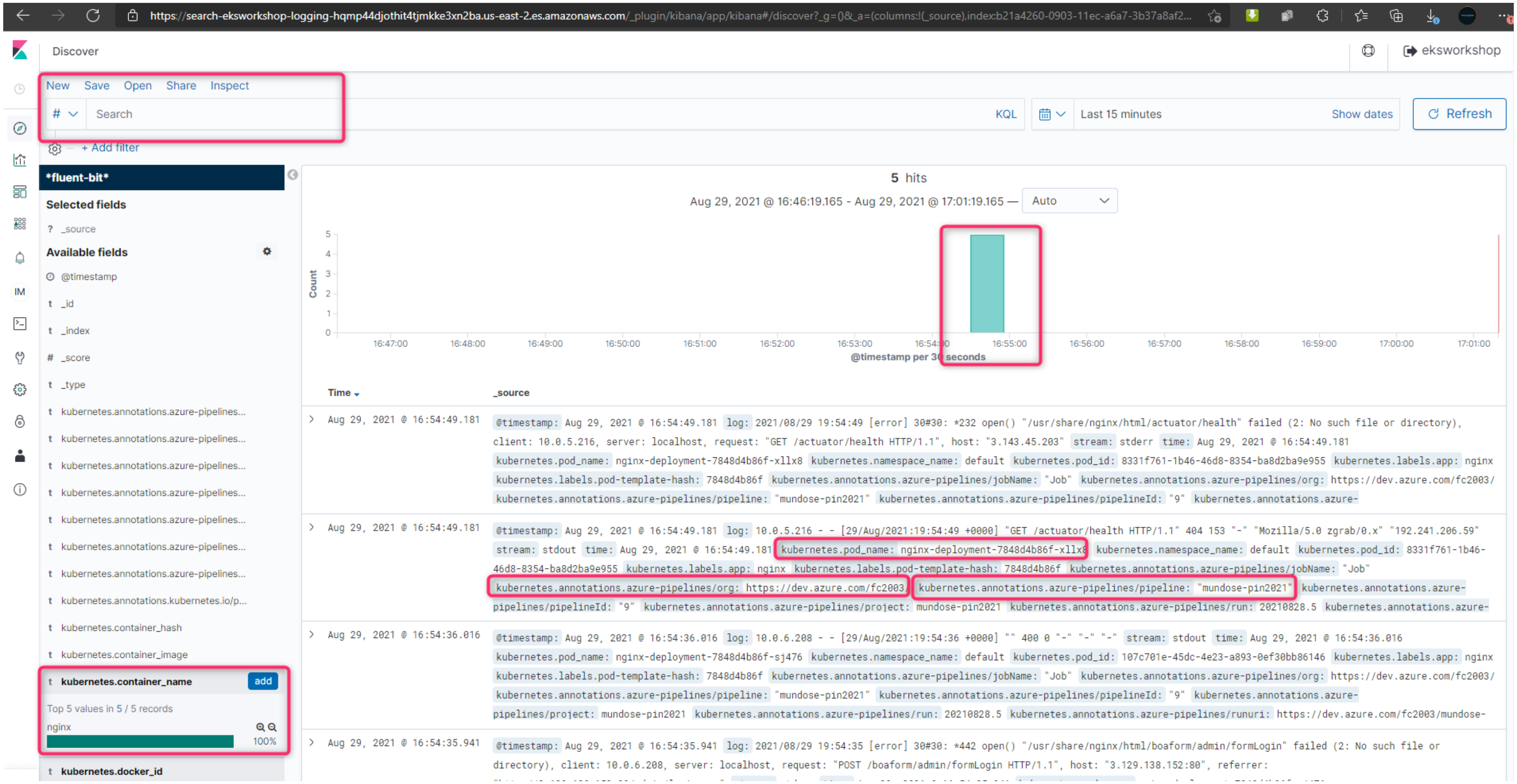
Time Filter field name: @timestamp | Default

This page lists every field in the ***fluent-bit*** index and the field's associatec Elasticsearch Mapping API

Name	Type
@timestamp	date
_id	string
_index	string
_score	number
_source	_source
_type	string
kubernetes.annotations.azure-pipelines/jobName	string
kubernetes.annotations.azure-pipelines/jobName.keyword	string
kubernetes.annotations.azure-pipelines/org	string
kubernetes.annotations.azure-pipelines/org.keyword	string

Rows per page: 10

Navegar datos


https://search-eksworkshop-logging-hqmp44djothit4tjmkke3xn2ba.us-east-2.es.amazonaws.com/_plugin/kibana/app/kibana#/discover?_g=()&_a=(columns:!(_source),index:b21a4260-0903-11ec-a6a7-3b37a8af2...
Discover
eksworkshop
New
Save
Open
Share
Inspect
Search
KQL
Last 15 minutes
Show dates
Refresh
+ Add filter
fluent-bit
Selected fields
_source
Available fields
@timestamp
_id
_index
_score
_type
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.azure-pipelines...
kubernetes.annotations.kubernetes.io/p...
kubernetes.container_hash
kubernetes.container_image
kubernetes.container_name
add
Top 5 values in 5 / 5 records
nginx
100%
kubernetes.docker_id
5 hits
Aug 29, 2021 @ 16:46:19.165 - Aug 29, 2021 @ 17:01:19.165 —
Auto
Count
5
4
3
2
1
0
16:47:00
16:48:00
16:49:00
16:50:00
16:51:00
16:52:00
16:53:00
16:54:00
16:55:00
16:56:00
16:57:00
16:58:00
16:59:00
17:00:00
17:01:00
@timestamp per 30 seconds

```
Time                        _source

> Aug 29, 2021 @ 16:54:49.181   @timestamp: Aug 29, 2021 @ 16:54:49.181 log: 2021/08/29 19:54:49 [error] 30#30: *232 open() "/usr/share/nginx/html/actuator/health" failed (2: No such file or directory),
                                client: 10.0.5.216, server: localhost, request: "GET /actuator/health HTTP/1.1", host: "3.143.45.203" stream: stderr time: Aug 29, 2021 @ 16:54:49.181
                                kubernetes.pod_name: nginx-deployment-7848d4b86f-xllx8 kubernetes.namespace_name: default kubernetes.pod_id: 8331f761-1b46-46d8-8354-ba8d2ba9e955 kubernetes.labels.app: nginx
                                kubernetes.labels.pod-template-hash: 7848d4b86f kubernetes.annotations.azure-pipelines/jobName: "Job" kubernetes.annotations.azure-pipelines/org: https://dev.azure.com/fc2003/
                                kubernetes.annotations.azure-pipelines/pipeline: "mundose-pin2021" kubernetes.annotations.azure-pipelines/pipelineId: "9" kubernetes.annotations.azure-

> Aug 29, 2021 @ 16:54:49.181   @timestamp: Aug 29, 2021 @ 16:54:49.181 log: 10.0.5.216 - - [29/Aug/2021:19:54:49 +0000] "GET /actuator/health HTTP/1.1" 404 153 "-" "Mozilla/5.0 zgrab/0.x" "192.241.206.59"
                                stream: stdout time: Aug 29, 2021 @ 16:54:49.181 kubernetes.pod_name: nginx-deployment-7848d4b86f-xllx8 kubernetes.namespace_name: default kubernetes.pod_id: 8331f761-1b46-
                                46d8-8354-ba8d2ba9e955 kubernetes.labels.app: nginx kubernetes.labels.pod-template-hash: 7848d4b86f kubernetes.annotations.azure-pipelines/jobName: "Job"
                                kubernetes.annotations.azure-pipelines/org: https://dev.azure.com/fc2003/ kubernetes.annotations.azure-pipelines/pipeline: "mundose-pin2021" kubernetes.annotations.azure-
                                pipelines/pipelineId: "9" kubernetes.annotations.azure-pipelines/project: mundose-pin2021 kubernetes.annotations.azure-pipelines/run: 20210828.5 kubernetes.annotations.azure-

> Aug 29, 2021 @ 16:54:36.016   @timestamp: Aug 29, 2021 @ 16:54:36.016 log: 10.0.6.208 - - [29/Aug/2021:19:54:36 +0000] "" 400 0 "-" "-" "-" stream: stdout time: Aug 29, 2021 @ 16:54:36.016
                                kubernetes.pod_name: nginx-deployment-7848d4b86f-sj476 kubernetes.namespace_name: default kubernetes.pod_id: 107c701e-45dc-4e23-a893-0ef30bb86146 kubernetes.labels.app: nginx
                                kubernetes.labels.pod-template-hash: 7848d4b86f kubernetes.annotations.azure-pipelines/jobName: "Job" kubernetes.annotations.azure-pipelines/org: https://dev.azure.com/fc2003/
                                kubernetes.annotations.azure-pipelines/pipeline: "mundose-pin2021" kubernetes.annotations.azure-pipelines/pipelineId: "9" kubernetes.annotations.azure-
                                pipelines/project: mundose-pin2021 kubernetes.annotations.azure-pipelines/run: 20210828.5 kubernetes.annotations.azure-pipelines/runuri: https://dev.azure.com/fc2003/mundose-

> Aug 29, 2021 @ 16:54:35.941   @timestamp: Aug 29, 2021 @ 16:54:35.941 log: 2021/08/29 19:54:35 [error] 30#30: *442 open() "/usr/share/nginx/html/boaform/admin/formLogin" failed (2: No such file or
                                directory), client: 10.0.6.208, server: localhost, request: "POST /boaform/admin/formLogin HTTP/1.1", host: "3.129.138.152:80", referrer:
```

Desplegar Prometheus

Agregar repositorios de HELM

Ejecutar los siguientes comandos Prometheus-Grafana-Deploy Repo

Agregar prometheus Helm repo

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

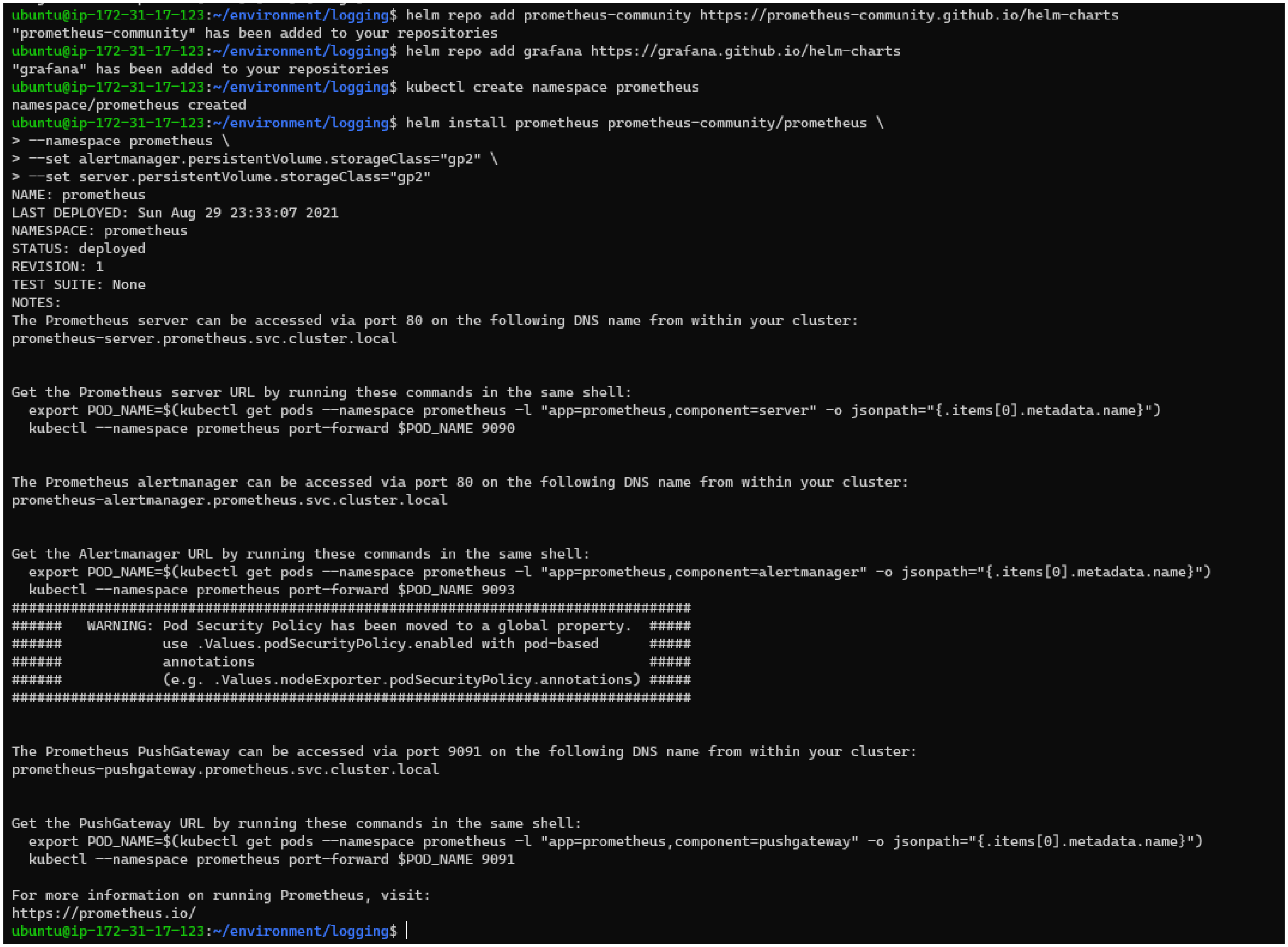
Agregar grafana Helm repo

```
helm repo add grafana https://grafana.github.io/helm-charts
```

Desplegar Prometheus

```
kubectl create namespace prometheus
```

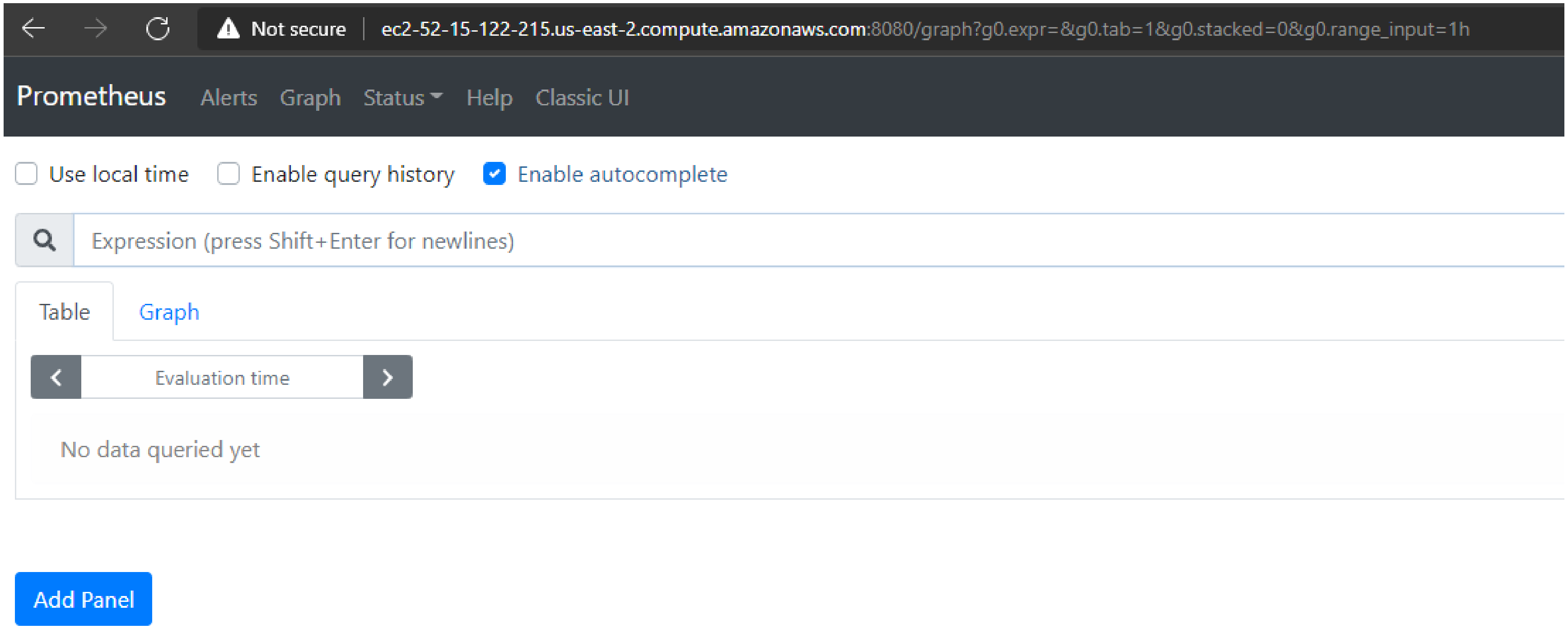
```
helm install prometheus prometheus-community/prometheus \
    --namespace prometheus \
    --set alertmanager.persistentVolume.storageClass="gp2" \
    --set server.persistentVolume.storageClass="gp2"
```



```
ubuntu@ip-172-31-17-123:~/environment/logging$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
ubuntu@ip-172-31-17-123:~/environment/logging$ helm repo add grafana https://grafana.github.io/helm-charts
"grafana" has been added to your repositories
ubuntu@ip-172-31-17-123:~/environment/logging$ kubectl create namespace prometheus
namespace/prometheus created
ubuntu@ip-172-31-17-123:~/environment/logging$ helm install prometheus prometheus-community/prometheus \
> --namespace prometheus \
> --set alertmanager.persistentVolume.storageClass="gp2" \
> --set server.persistentVolume.storageClass="gp2"
NAME: prometheus
LAST DEPLOYED: Sun Aug 29 23:33:07 2021
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=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 80 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=prometheus,component=alertmanager" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been moved to a global property.  #####
######            use .Values.podSecurityPolicy.enabled with pod-based      #####
######            annotations                                               #####
######            (e.g. .Values.nodeExporter.podSecurityPolicy.annotations) #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
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,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/
ubuntu@ip-172-31-17-123:~/environment/logging$ 
```

Exponer prometheus en la instancia de EC2 en el puerto 8080

```
kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090 --address 0.0.0.0
```



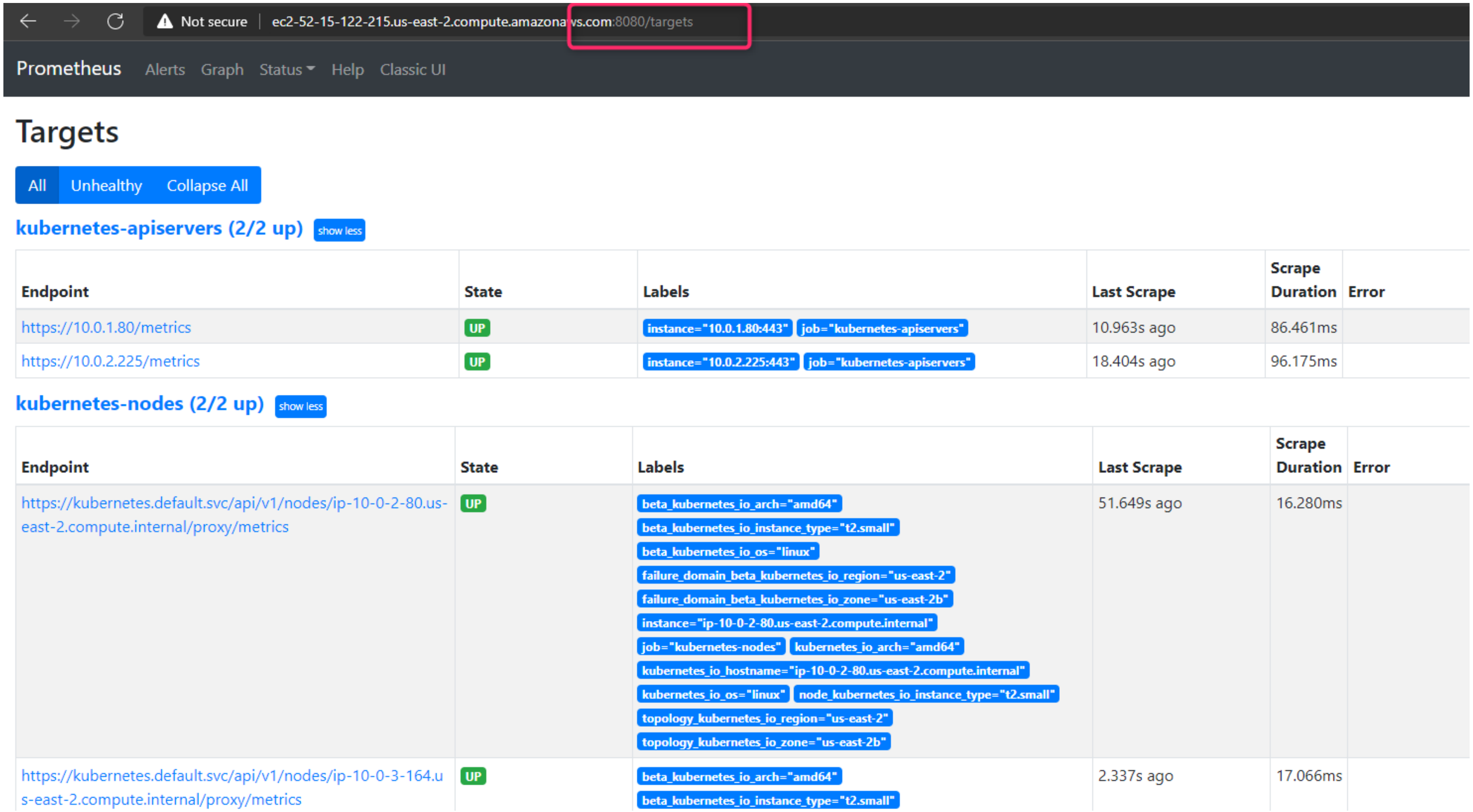
Not secure | ec2-52-15-122-215.us-east-2.compute.amazonaws.com:8080/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.range_input=1h

□ Use local time □ Enable query history ☑ Enable autocomplete

Expression (press Shift+Enter for newlines)

No data queried yet

Navegar a /targets



Targets

kubernetes-apiservers (2/2 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://10.0.1.80/metrics	UP	instance="10.0.1.80:443" job="kubernetes-apiservers"	10.963s ago	86.461ms	
https://10.0.2.225/metrics	UP	instance="10.0.2.225:443" job="kubernetes-apiservers"	18.404s ago	96.175ms	

kubernetes-nodes (2/2 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/ip-10-0-2-80.us-east-2.compute.internal/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="t2.small" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="us-east-2" failure_domain_beta_kubernetes_io_zone="us-east-2b" instance="ip-10-0-2-80.us-east-2.compute.internal" job="kubernetes-nodes" kubernetes_io_arch="amd64" kubernetes_io_hostname="ip-10-0-2-80.us-east-2.compute.internal" kubernetes_io_os="linux" node_kubernetes_io_instance_type="t2.small" topology_kubernetes_io_region="us-east-2" topology_kubernetes_io_zone="us-east-2b"	51.649s ago	16.280ms	
https://kubernetes.default.svc/api/v1/nodes/ip-10-0-3-164.us-east-2.compute.internal/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="t2.small"	2.337s ago	17.066ms	

En este punto Prometheus está funcionando correctamente

Desplegar Grafana

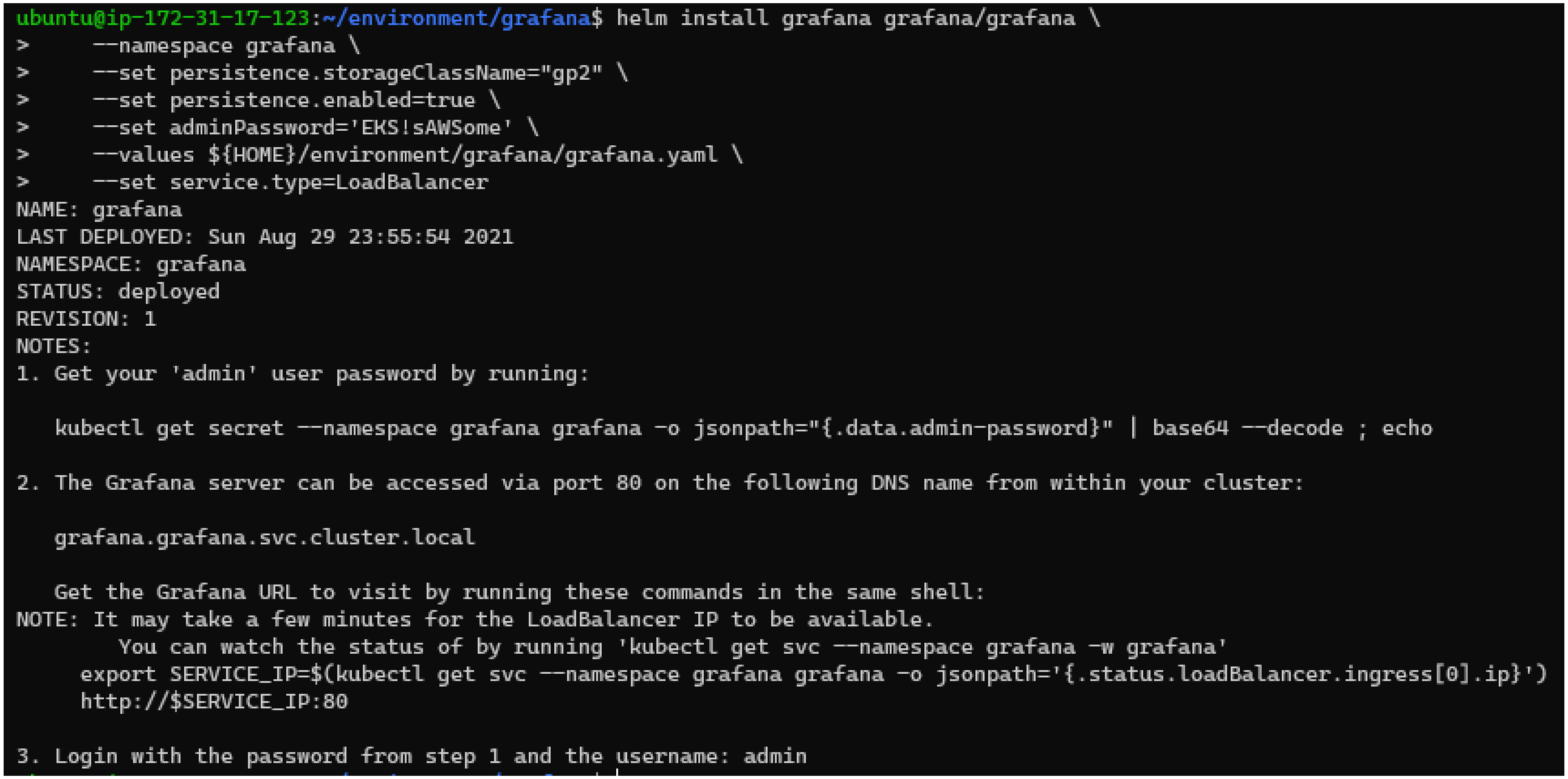
Crear YAML Grafana

Crear directorio grafana dentro de environment y depositar el archivo YAML de Grafana grafana.yaml repo

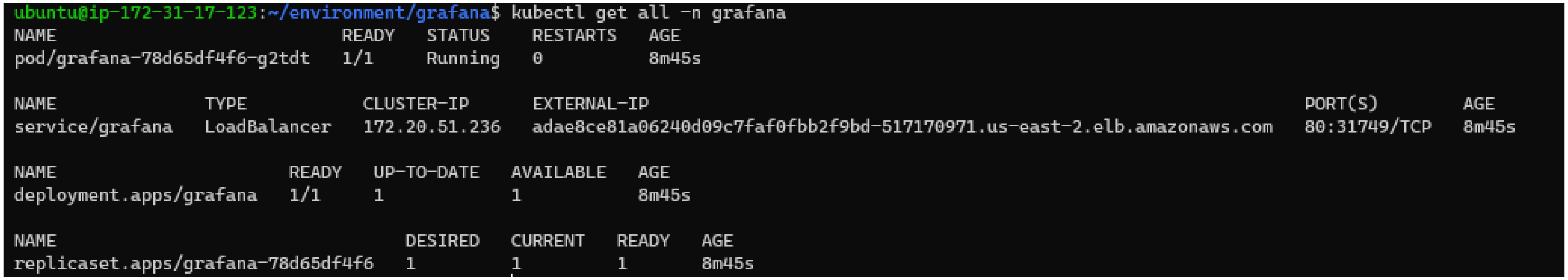
```
mkdir ${HOME}/environment/grafana

cat << EoF > ${HOME}/environment/grafana/grafana.yaml
datasources:
  datasources.yaml:
    apiVersion: 1
    datasources:
    - name: Prometheus
      type: prometheus
      url: http://prometheus-server.prometheus.svc.cluster.local
      access: proxy
      isDefault: true
EoF
```

Desplegar Grafana

Ejecutar los siguientes comandos para crear el namespace y desplegar el chart de Helm grafana deployment repo



```
ubuntu@ip-172-31-17-123:~/environment/grafana$ helm install grafana grafana/grafana \
>     --namespace grafana \
>     --set persistence.storageClassName="gp2" \
>     --set persistence.enabled=true \
>     --set adminPassword='EKS!sAWSome' \
>     --values ${HOME}/environment/grafana/grafana.yaml \
>     --set service.type=LoadBalancer
NAME: grafana
LAST DEPLOYED: Sun Aug 29 23:55:54 2021
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
```



```
ubuntu@ip-172-31-17-123:~/environment/grafana$ kubectl get all -n grafana
NAME                           READY   STATUS    RESTARTS   AGE
pod/grafana-78d65df4f6-g2tdt   1/1     Running   0          8m45s

NAME              TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)        AGE
service/grafana   LoadBalancer   172.20.51.236   adae8ce81a06240d09c7faf0fbb2f9bd-517170971.us-east-2.elb.amazonaws.com   80:31749/TCP   8m45s

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana   1/1     1            1           8m45s

NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-78d65df4f6   1         1         1       8m45s
```

Obtener url de Grafana

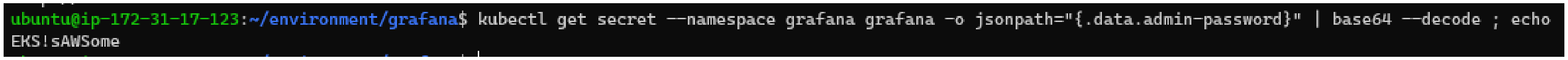
```
export ELB=$(kubectl get svc -n grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].hostname}')
```

```
echo "http://$ELB"
```

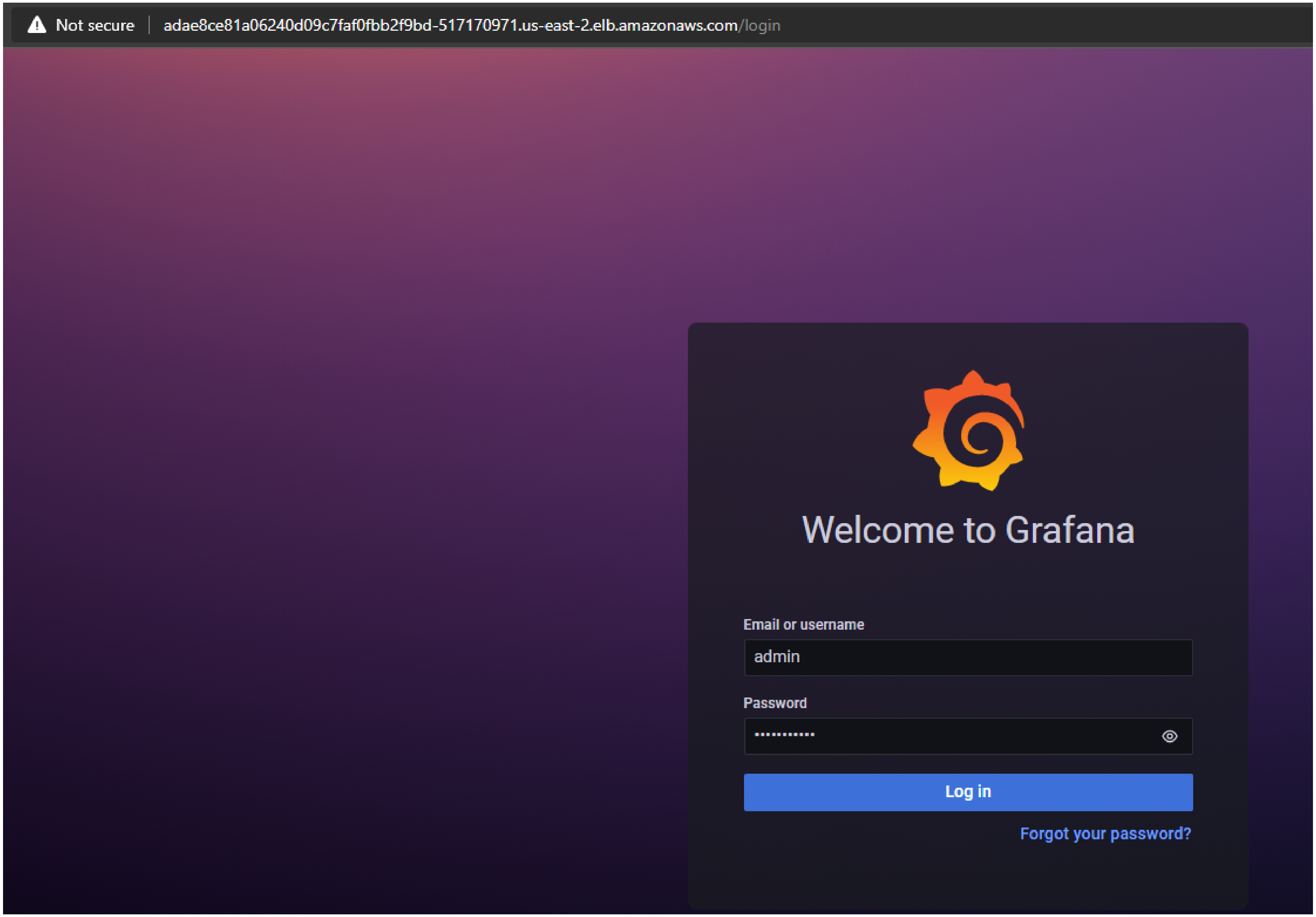
Ingresar a Grafana

Utilizar el usuario **admin** y obtener la contraseña desde el **secret**. **Nota: Definimos esta contraseña en el paso de despliegue**

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



```
ubuntu@ip-172-31-17-123:~/environment/grafana$ kubectl get secret --namespace grafana grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo
EKS!sAWSome
```



Not secure | adae8ce81a06240d09c7faf0fbb2f9bd-517170971.us-east-2.elb.amazonaws.com/login

Welcome to Grafana

Email or username: admin

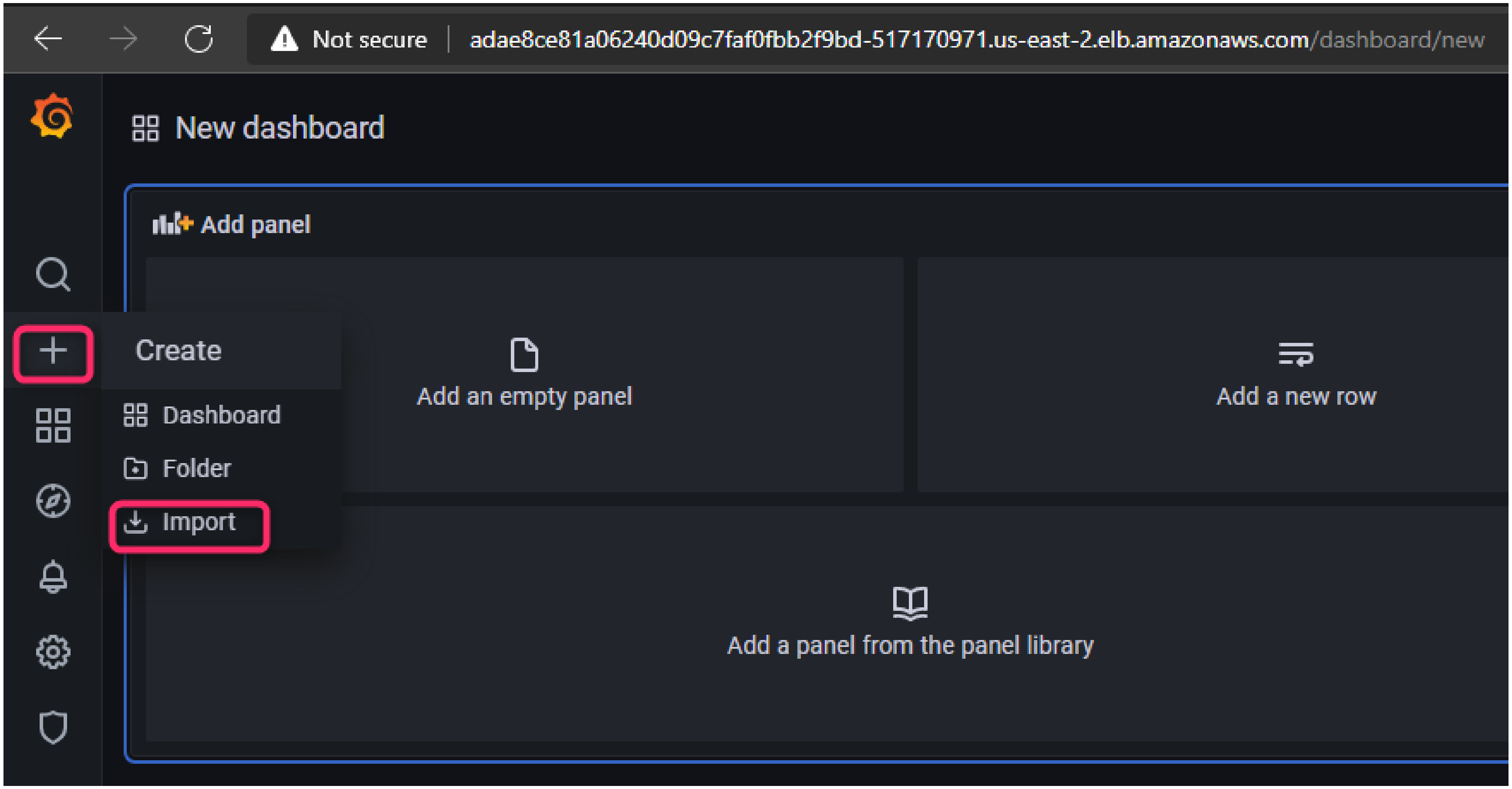
Password: ••••••••••••

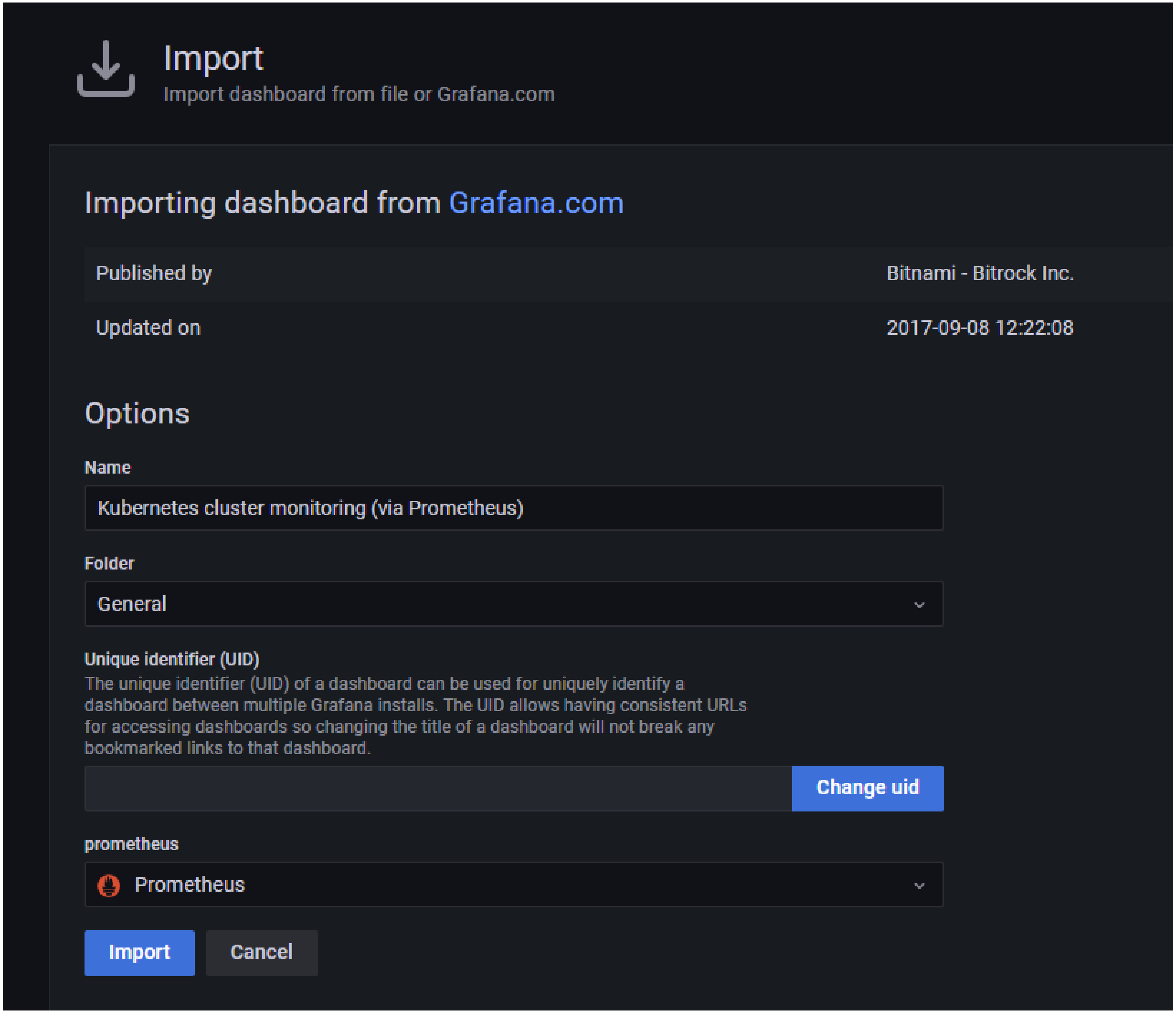
Configurar Grafana

Importar Cluster Monitoring Dashboard

Hacemos clic en + > Import >

Escribimos 3119 > Load > Seleccionamos prometheus como el datasource > Import


Not secure
adae8ce81a06240d09c7faf0fbb2f9bd-517170971.us-east-2.elb.amazonaws.com/dashboard/new
New dashboard
Add panel
Create
Dashboard
Folder
Import
Add an empty panel
Add a new row
Add a panel from the panel library



Import

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by	Bitnami - Bitrock Inc.
Updated on	2017-09-08 12:22:08

Options

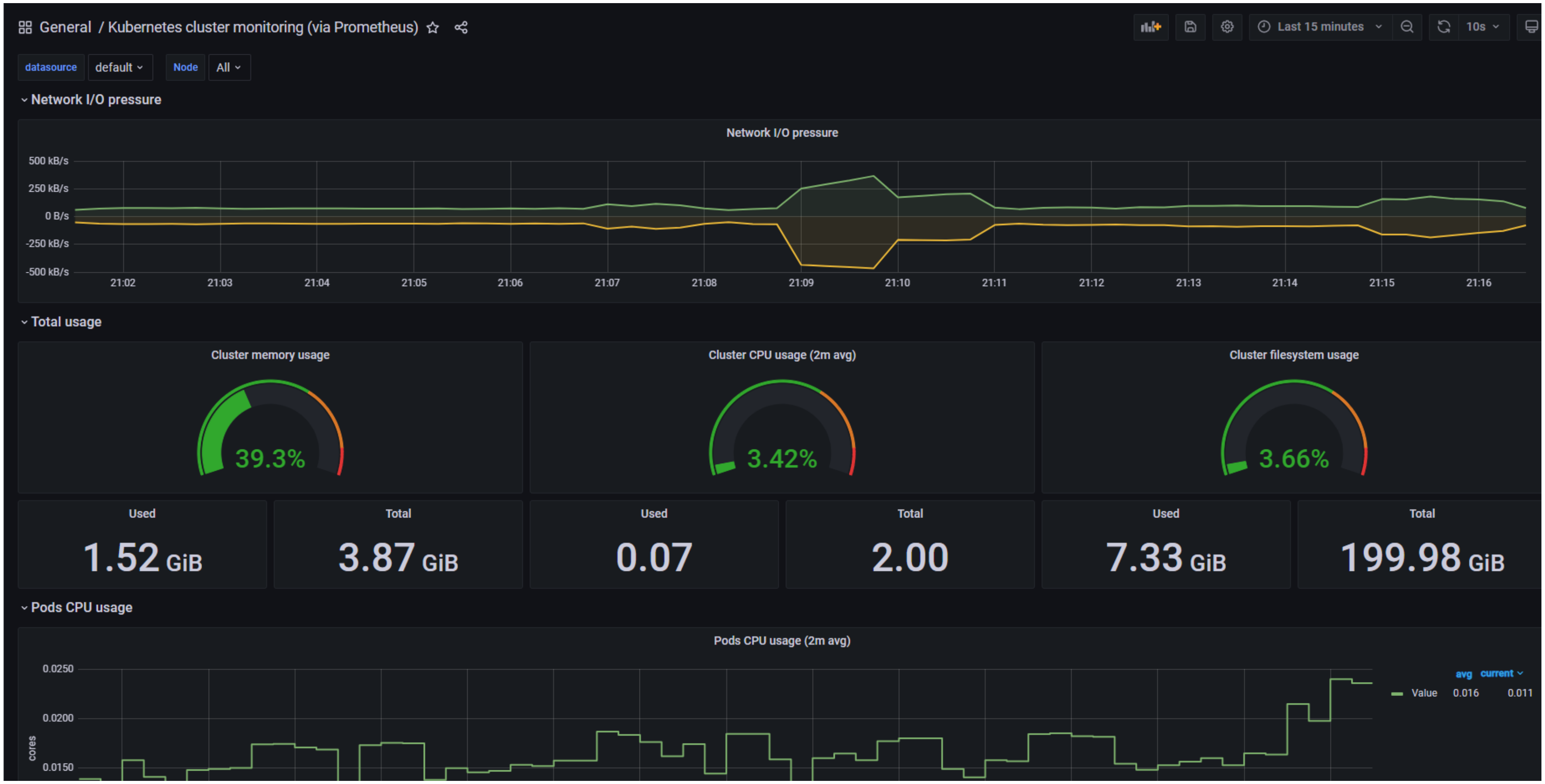
Name: Kubernetes cluster monitoring (via Prometheus)

Folder: General

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.

prometheus: Prometheus

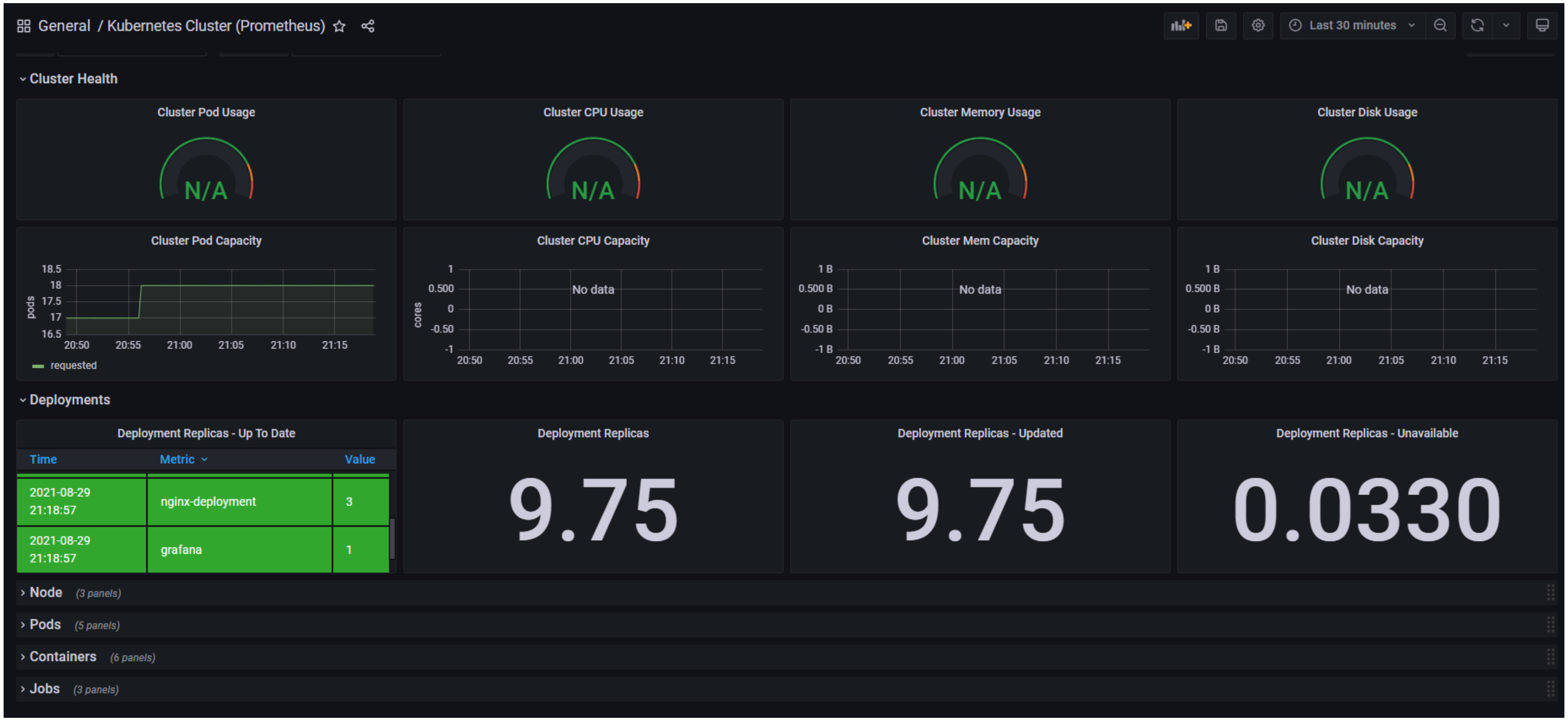


Importar Pods Monitoring Dashboard

Repetimos el procedimiento pero esta vez importando el dashboard 6417

Hacemos clic en + > Import >

Escribimos 6417 > Load > Seleccionamos prometheus como el datasource > Import



Time	Metric	Value
2021-08-29 21:18:57	nginx-deployment	3
2021-08-29 21:18:57	grafana	1

Cleanup de recursos

Borrar FluentBit y Elastic

```
cd  ~/environment/
```

```
kubectl delete -f ~/environment/logging/fluentbit.yaml

aws es delete-elasticsearch-domain \
    --domain-name ${ES_DOMAIN_NAME}

eksctl delete iamserviceaccount \
    --name fluent-bit \
    --namespace logging \
    --cluster eksworkshop-eksctl \
    --wait

aws iam delete-policy   \
  --policy-arn "arn:aws:iam::${ACCOUNT_ID}:policy/fluent-bit-policy"

kubectl delete namespace logging

rm -rf ~/environment/logging

unset ES_DOMAIN_NAME
unset ES_VERSION
unset ES_DOMAIN_USER
unset ES_DOMAIN_PASSWORD
unset FLUENTBIT_ROLE
unset ES_ENDPOINT
```

Borrar Prometheus y Grafana

```
helm uninstall prometheus --namespace prometheus
```

```
kubectl delete ns prometheus
```

```
helm uninstall grafana --namespace grafana
kubectl delete ns grafana
```

```
rm -rf ${HOME}/environment/grafana
```

Borrar Cluster EKS

Si lo crearon con eksctl

```
eksctl delete cluster --name “Nombre del cluster”
```

Si lo crearon con terraform

```
Terraform destroy
```