

Proyecto Integrador DevOps 2203 - Grupo 7

En la implementación del presente proyecto se maximizó el uso de infraestructura como código (IaC), el repositorio del proyecto se encuentra aquí. La infraestructura como código forma parte del pilar de **Excelencia Operacional** del AWS Well-Architected Framework. Como herramienta para la gestión de la IaC se decidió usar el AWS Cloud Development Kit (CDK), principalmente por los beneficios de:

- Usar abstracciones de alto nivel para definir infraestructura (Constructos de nivel 1, 2 y 3).
- Diseñar y desarrollar componentes de infraestructura reusables.
- Soporte para lenguajes: Typescript/Javascript, Python, Java, GO, .NET.
- Infraestructura y código de aplicación pueden convivir en el mismo repositorio.
- El background de los integrantes del equipo es orientado al área de desarrollo.

Integrantes

- Luis Miguel Mamani Humpiri
- Carlos Ruiz de la Vega
- Reynaldo Capia Capia

Crear y configurar instancia bastión

El aprovisionamiento y configuración del nodo bastión se encuentra en cdk-bastion, este proyecto define la IaC requeridas para la creación y configuración de la instancia bastión así como del rol que se usará. La definición y configuración del bastión se encuentra en bastion.py e incluye:

1. Tipo y user-data de instancia
2. Storage de instancia
3. Tags de instancia
4. Security group de instancia
5. KeyPair de instancia
6. Rol de instancia
7. Permisos de rol de instancia
8. Tags de rol de instancia
9. Asignación de rol de instancia

```
# 4. Security group de instancia
self._instance_security_group = ec2.SecurityGroup(
    self,
    "InstanceSecurityGroup",
    security_group_name=f"{construct_id}-security-group",
    vpc=vpc,
    allow_all_outbound=True,
```

```

        description="Bastion instance security group",
    )
self._instance_security_group.add_ingress_rule(
    ec2.Peer.any_ipv4(),
    ec2.Port.tcp(22),
    "Allows SSH access from any IP"
)

# 5. KeyPair de instancia
self._instance_key_pair = KeyPair(
    self,
    "InstanceKeyPair",
    name=f"{construct_id}-key-pair",
    resource_prefix=f"{construct_id}",
    store_public_key=True,
)

# 6. Rol de instancia
self._instance_role = iam.Role(
    self,
    "InstanceRole",
    role_name=f"{construct_id}-role",
    assumed_by=iam.ServicePrincipal("ec2.amazonaws.com"),
)

# 7. Permisos de rol de instancia
self._instance_role.add_managed_policy(
    iam.ManagedPolicy.from_aws_managed_policy_name("AdministratorAccess")
)
# SSM Agent
self._instance_role.add_managed_policy(
    iam.ManagedPolicy.from_aws_managed_policy_name("AmazonSSMManagedInstanceCore")
)

# 8. Tags de rol de instancia
Tags.of(self._instance_role).add(
    key='RoleName',
    value='ec2-admin-role',
)

# 2. Storage de instancia
self._root_volume: ec2.BlockDevice = ec2.BlockDevice(
    device_name='/dev/xvda',
    volume=ec2.BlockDeviceVolume.ebs(
        volume_size=8,
        volume_type=ec2.EbsDeviceVolumeType.GP2,

```

```

),
)

# Instancia EC2
self._instance = ec2.Instance(
    self,
    "Instance",
    instance_name=instance_name,
    vpc=vpc,
    vpc_subnets=ec2.SubnetSelection(subnet_type=ec2.SubnetType.PUBLIC),
    # 9. Asignación de rol de instancia
    role=self._instance_role,
    # 4. Security group de instancia
    security_group=self._instance_security_group,
    key_name=self._instance_key_pair.key_pair_name,
    # 1. tipo de instancia
    instance_type=ec2.InstanceType(instance_type),
    # 2. storage de instancia
    block_devices=[self._root_volume],
    machine_image=ec2.MachineImage.from_ssm_parameter(
        # Ubuntu LTS
        parameter_name="/aws/service/canonical/ubuntu/server/focal/stable/current/amd64/hvm",
        os=ec2.OperatingSystemType.LINUX,
    ),
    # 1. user-data de instancia
    user_data=user_data,
)

# 3. Tags de instancia
Tags.of(self._instance).add(
    key='Name',
    value='Jenkins',
)

```

El stack de la instancia bastión se despliega con el comando `cdk deploy`:

Conectándose a la instancia bastión a través de **Session Manager**:

Conectándose a la instancia bastión a través de SSH. Las llaves se encuentran en Secrets Manager:

```

cdk-bastion-bastion: deploying... [1/1]
[0%] start: Publishing 4b53adc667dcab419bec9e03553dce125d10f1d9c6ac1d2c082001b9332f150f:136737878111-us-east-1
[0%] start: Publishing 1df2530a6e480baf0f76ccbb1a01b54e3a318d0caf684ccfb66831cdb8b427:136737878111-us-east-1
[50%] success: Published 4b53adc667dcab419bec9e03553dce125d10f1d9c6ac1d2c082001b9332f150f:136737878111-us-east-1
[100%] success: Published 1df2530a6e480baf0f76ccbb1a01b54e3a318d0caf684ccfb66831cdb8b427:136737878111-us-east-1
cdk-bastion-bastion: creating CloudFormation changeset...

✓ cdk-bastion-bastion

✖ Deployment time: 216.13s

Stack ARN:
arn:aws:cloudformation:us-east-1:136737878111:stack/cdk-bastion-bastion/76268680-e7d5-11ed-a6d2-124d7f36d237

```

Figure 1: img_62.png

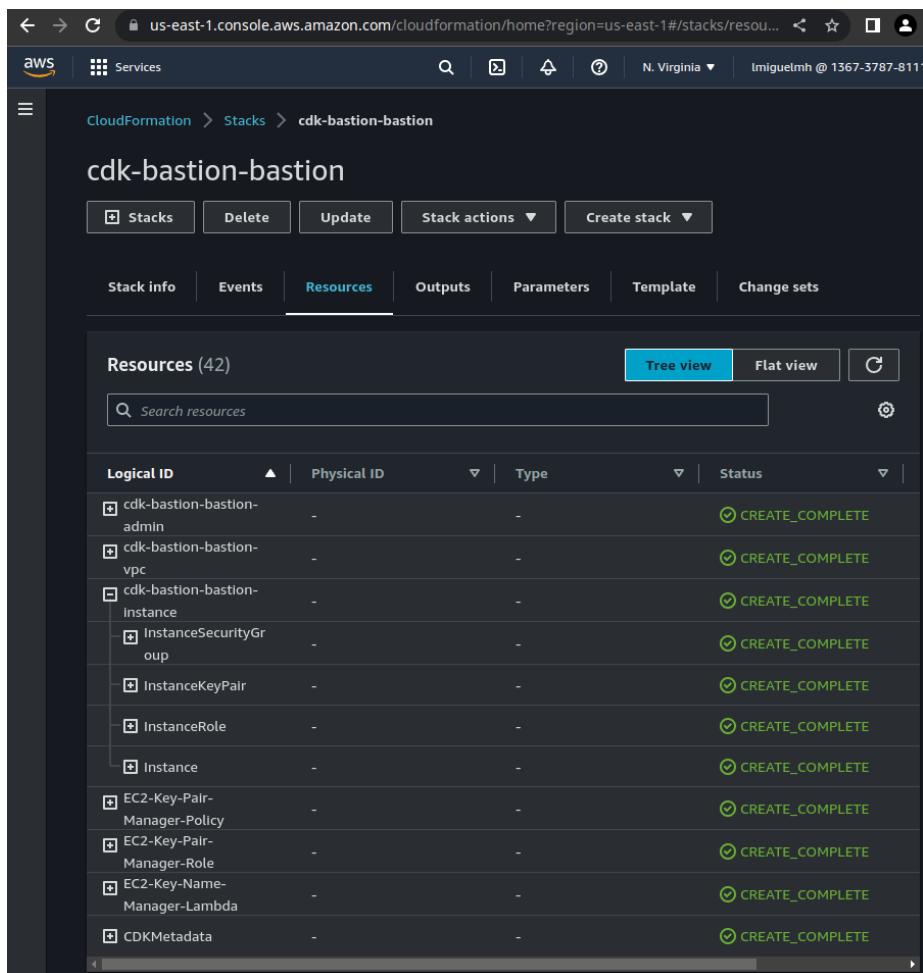


Figure 2: img_63.png

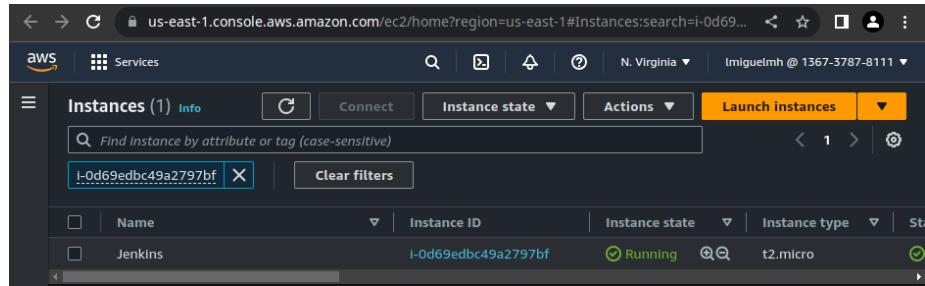


Figure 3: img_64.png

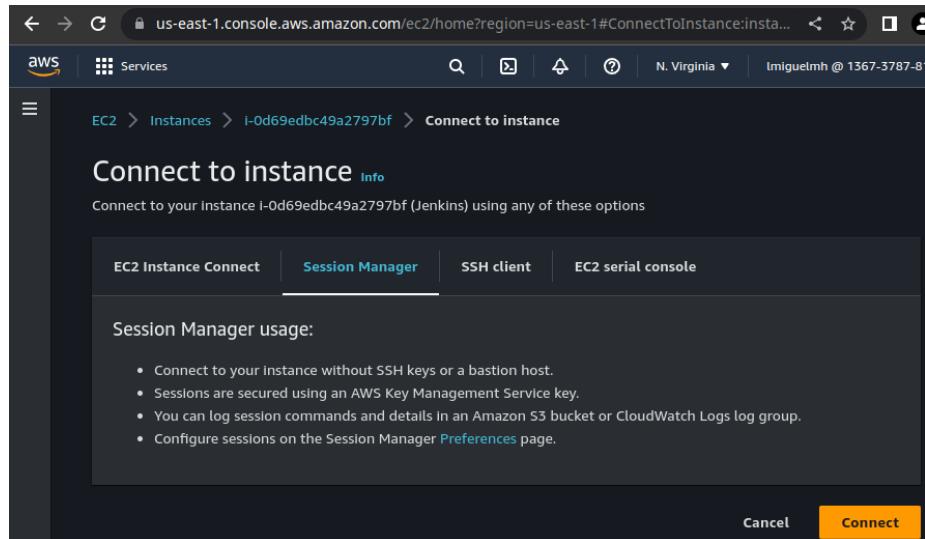
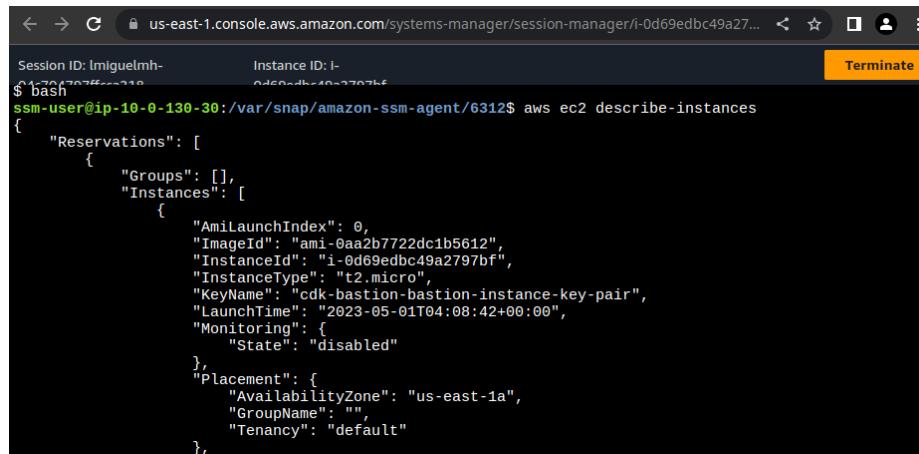
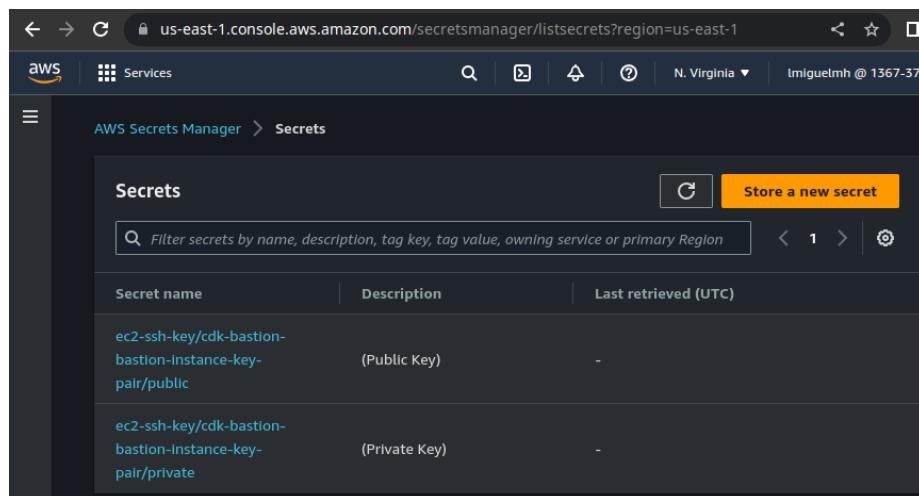


Figure 4: img_65.png



```
Session ID: lmiguelmh-  
Instance ID: i-0d69edbc49a27...  
Terminate  
$ bash  
ssm-user@ip-10-0-130-30:/var/snap/amazon-ssm-agent/6312$ aws ec2 describe-instances  
{  
    "Reservations": [  
        {  
            "Groups": [],  
            "Instances": [  
                {  
                    "AmiLaunchIndex": 0,  
                    "ImageId": "ami-0aa2b7722dc1b5612",  
                    "InstanceId": "i-0d69edbc49a2797bf",  
                    "InstanceType": "t2.micro",  
                    "KeyName": "cdk-bastion-bastion-instance-key-pair",  
                    "LaunchTime": "2023-05-01T04:08:42+00:00",  
                    "Monitoring": {  
                        "State": "disabled"  
                    },  
                    "Placement": {  
                        "AvailabilityZone": "us-east-1a",  
                        "GroupName": "",  
                        "Tenancy": "default"  
                    },  
                    "RootDeviceType": "Amazon EBS",  
                    "State": "running",  
                    "StateReason": {  
                        "Code": "Success",  
                        "Message": "The instance has been successfully started."  
                    },  
                    "SubnetId": "subnet-000000000000000000",  
                    "VpcId": "vpc-000000000000000000",  
                    "NetworkInterfaces": [  
                        {  
                            "Association": {  
                                "AllocationId": "eni-000000000000000000",  
                                "InterfacePrivateDns": "ip-10-0-130-30.eu-central-1.compute.internal",  
                                "InterfacePrivateIp": "10.0.130.30",  
                                "PublicDns": "ip-10-0-130-30.eu-central-1.compute.amazonaws.com",  
                                "PublicIp": "10.0.130.30",  
                                "Region": "eu-central-1",  
                                "Status": "attached",  
                                "VpcId": "vpc-000000000000000000"  
                            },  
                            "Description": "Primary network interface",  
                            "MacAddress": "e8:0c:2d:00:00:00",  
                            "NetworkInterfaceId": "eni-000000000000000000",  
                            "PrivateDns": "ip-10-0-130-30.eu-central-1.compute.internal",  
                            "PrivateIp": "10.0.130.30",  
                            "Status": "in-use",  
                            "Type": "Primary"  
                        }  
                    ]  
                }  
            ]  
        }  
    ]  
}
```

Figure 5: img_66.png



Secret name	Description	Last retrieved (UTC)
ec2-ssh-key/cdk-bastion-bastion-instance-key-pair/public	(Public Key)	-
ec2-ssh-key/cdk-bastion-bastion-instance-key-pair/private	(Private Key)	-

Figure 6: img_67.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ chmod 400 bastion.pem
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ ssh -i bastion.pem ubuntu@ec2-3-231-58-232.compute-1.amazonaws.com
The authenticity of host 'ec2-3-231-58-232.compute-1.amazonaws.com (3.231.58.232)' can't be established.
ED25519 key fingerprint is SHA256:ASeMFNDxpyePtEsH84s2BCXgjBW4dFTVGfdwH9Hpyj8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-231-58-232.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1033-aws x86_64)

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

 System information as of Mon May  1 04:19:21 UTC 2023

 System load:  0.24           Processes:          101
 Usage of /:   32.5% of 7.57GB  Users logged in:     0
 Memory usage: 26%            IPv4 address for eth0: 10.0.130.30
 Swap usage:   0%
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.

ubuntu@ip-10-0-130-30:~$ aws s3 ls
2022-05-15 23:20:43 cdk-hnb659fds-assets-136737878111-us-east-1
| 2023-04-30 21:21:01 cdk-hnb659fds-assets-136737878111-us-west-2
```

Crear cluster EKS

La definición del cluster EKS se encuentra en cluster/component.py e incluye:

1. Cluster EKS.
2. Grupo de autoescalamiento para los worker nodes.
3. Llave SSH para acceder a los workers.

```
# 1. Cluster EKS.
cluster = eks.Cluster(
    self,
    conf.CLUSTER_EKS_NAME,
    cluster_name=conf.CLUSTER_EKS_NAME,
    version=eks.KubernetesVersion.V1_25,
    kubectl_layer=lambda_layer_kubectl_v25.KubectlV25Layer(self, "kubectl"),
    endpoint_access=eks.EndpointAccess.PUBLIC_AND_PRIVATE,
    default_capacity=0, # will customize it using an ASG
    alb_controller=eks.AlbControllerOptions(
        version=eks.AlbControllerVersion.V2_4_1,
    ),
    cluster_logging=[
        eks.ClusterLoggingTypes.API,
        eks.ClusterLoggingTypes.AUDIT,
```

```

        eks.ClusterLoggingTypes.AUTHENTICATOR,
        eks.ClusterLoggingTypes.CONTROLLER_MANAGER,
        eks.ClusterLoggingTypes.SCHEDULER,
    ],
)

# 3. Llave SSH para acceder a los workers.
key_pair = KeyPair(
    self,
    conf.CLUSTER_SSH_KEY_NAME,
    name=conf.CLUSTER_SSH_KEY_NAME,
    resource_prefix=conf.CLUSTER_SSH_KEY_NAME,
)

# 2. Grupo de autoescalamiento para los worker nodes.
cluster.add_auto_scaling_group_capacity(
    conf.CLUSTER_ASG_NAME,
    auto_scaling_group_name=conf.CLUSTER_ASG_NAME,
    instance_type=ec2.InstanceType("t3.large"),
    machine_image_type=eks.MachineImageType.AMAZON_LINUX_2, # or BOTTLEROCKET
    min_capacity=1,
    desired_capacity=1,
    max_capacity=3,
    vpc_subnets=ec2.SubnetSelection(subnet_type=ec2.SubnetType.PUBLIC),
    key_name=key_pair.key_pair_name,
)

```

El stack del cluster EKS se despliega con el comando `cdk deploy`:

```

eks-cluster: deploying... [1/1]
[0%] start: Publishing ebc48a9a74e277f463f78c5a0b3bcf489a891efe25ece75b5b439b9e89bee4ac:136737878111-us-east-1
[0%] start: Publishing 81679ff44041b616804767bcfdcc5d509ff630848eb51df91b0a44351ec90b5f:136737878111-us-east-1
[0%] start: Publishing 4f38ec7164c1b9c7bae70f79b9af9cbcd59f45e4f927cacc58e997783bebd7a:136737878111-us-east-1
[0%] start: Publishing 1eabd374284d340b7419e342908132f5bb0b7b28d472d852807d7f5f9746:136737878111-us-east-1
[0%] start: Publishing 92ea03f882e779503519f7781d06c03f95b46863db85f5c50a4e7debf0d4be02:136737878111-us-east-1
[0%] start: Publishing 159ab89ebd881b09102e294af7d28d2d93b12ed951fab089ece691faef1ffd44:136737878111-us-east-1
[0%] start: Publishing ffa3d5f3876afed62c60735ced8dafb4db2ccbd70a8b2f6bb0d9e6aaa5823d631:136737878111-us-east-1
[0%] start: Publishing a913a2614f4e118cb83fa3c0dd17575c9adbbdbad4da17aeecd505282a25513f8:136737878111-us-east-1
[0%] start: Publishing 4b53adc667dcab419bec9e03553dce125d10f1d9c6ac1d2c082001b9332f150f:136737878111-us-east-1
[0%] start: Publishing 8abb24e0a6f69d09d5c7f743d2c10d7123f28b2d5736792166500ecfb13be:136737878111-us-east-1
[0%] start: Publishing 1a8433f1c759432998a5aa800e84caae3ad9404047e3b9953e1439a2441cd3:136737878111-us-east-1
[0%] start: Publishing f013c2d14f16f8fc674d69770fa52d53c1b9123304d9a3a806bc88be6e0ddedb:136737878111-us-east-1
[8%] success: Published ebc48a9a74e277f463f78c5a0b3bcf489a891efe25ece75b5b439b9e89bee4ac:136737878111-us-east-1
[16%] success: Published 4f38ec7164c1b9c7bae70f79b9af9cbcd59f45e4f927cacc58e997783bebd7a:136737878111-us-east-1
[25%] success: Published 1eabd374284d340b74179e342908132f5bb0b7b28d472d852807d7f5f9746:136737878111-us-east-1
[33%] success: Published 81679ff44041b616804767bcfdcc5d509ff630848eb51df91b0a44351ec90b5f:136737878111-us-east-1
[41%] success: Published 92ea03f8b2e779503519f7781d06c03f95b46863db85f5c50a4e7debf0d4be02:136737878111-us-east-1
[50%] success: Published a913a2614f4e118cb83fa3c0dd17575c9adbbdbad4da17aeecd505282a25513f8:136737878111-us-east-1
[58%] success: Published ffa3d5f3876afed62c60735ced8dafb4db2ccbd70a8b2f6bb0d9e6aaa5823d631:136737878111-us-east-1
[66%] success: Published 4b53adc667dcab419bec9e03553dce125d10f1d9c6ac1d2c082001b9332f150f:136737878111-us-east-1
[75%] success: Published 159ab89ebd881b09102e294af7d28d2d93b12ed951fab089ece691faef1ffd44:136737878111-us-east-1
[83%] success: Published 8abb24e0a6f69d09d5c7f743d2c1dd7123f28b2d5736792166500ecfb13be:136737878111-us-east-1
[91%] success: Published 1a8433f1c759432998a5aa800e84caae3ad9404047e3b9953e1439a2441cd3:136737878111-us-east-1
[100%] success: Published f013c2d14f16f8fc674d69770fa52d53c1b9123304d9a3a806bc88be6e0ddedb:136737878111-us-east-1
eks-cluster: creating CloudFormation changeset...

✓ eks-cluster
✓ eks-cluster

⌚ Deployment time: 1432.35s

Outputs:
eks-cluster.EksUpdateKubeconfig = aws eks update-kubeconfig --region us-east-1 --name eks-cluster-eks --role-arn eks-cluster.ServiceAccountIamRole = arn:aws:iam::136737878111:role/eks-cluster-eksclustereksfluentbitRole5A927374
eks-cluster.eksclustereksConfigCommand02010694 = aws eks update-kubeconfig --name eks-cluster-eks --region us-east-1
eks-cluster.eksclustereksGetTokenCommand086540443 = aws eks get-token --cluster-name eks-cluster-eks --region us-east-1
Stack ARN: arn:aws:cloudformation:us-east-1:136737878111:stack/eks-cluster/70756260-e7a3-11ed-a869-0a5685d227fd
⌚ Total time: 1442.4s

```

Notar que en la consola EKS la pantalla informativa nos indica que nuestro usuario IAM no tiene acceso a los objetos Kubernetes.

Mapear usuarios IAM - cluster EKS

Antes de realizar el mapping se debe configurar kubectl, para lo cual el output del stack tiene el comando.

Podemos continuar con el mapping. Considerar que anteriormente la consola nos advertía que nuestro usuario IAM no tenía acceso a los objetos de Kubernetes.

Ahora, en la consola EKS, la advertencia desaparece y podemos ver los recursos del cluster.

Crear AWS Code Pipeline

Desde bastión sólo es necesario desplegar el pipeline. Por defecto, el pipeline será disparado por cambios en la rama definida en `core.common.PIPELINE_GITHUB_BRANCH`.

Screenshot of the AWS CloudFormation console showing the resources for the 'eks-cluster' stack.

The page title is 'eks-cluster' under 'CloudFormation > Stacks > eks-cluster'. The navigation bar includes 'Stacks', 'Delete', 'Update', 'Stack actions', and 'Create stack'.

The 'Resources' tab is selected, showing 78 resources. The table has columns: Logical ID, Physical ID, Type, and Status.

Logical ID	Physical ID	Type	Status
kubectl	-	-	CREATE_COMPLETE
eks-cluster-eks	-	-	CREATE_COMPLETE
@aws-cdk--aws-eks.ClusterResource...	-	-	CREATE_COMPLETE
@aws-cdk--aws-eks.KubectlProvider...	-	-	CREATE_COMPLETE
eksclustereksclusterek	s790FC687...	-	CREATE_COMPLETE
Custom::AWSCKOOpenIDConnectProvider...	-	-	CREATE_COMPLETE
AWSCKCfnUtilsProviderCustomResource...	-	-	CREATE_COMPLETE
eks-cluster-ssh-key-dev	-	-	CREATE_COMPLETE
EC2-Key-Pair-Manager-Policy	-	-	CREATE_COMPLETE
EC2-Key-Pair-Manager-Role	-	-	CREATE_COMPLETE
EC2-Key-Name-Manager-Lambda	-	-	CREATE_COMPLETE
eks-cluster-fluent-bit-service-account-role...	-	-	CREATE_COMPLETE
CDKMetadata	-	-	CREATE_COMPLETE

Figure 7: img_72.png

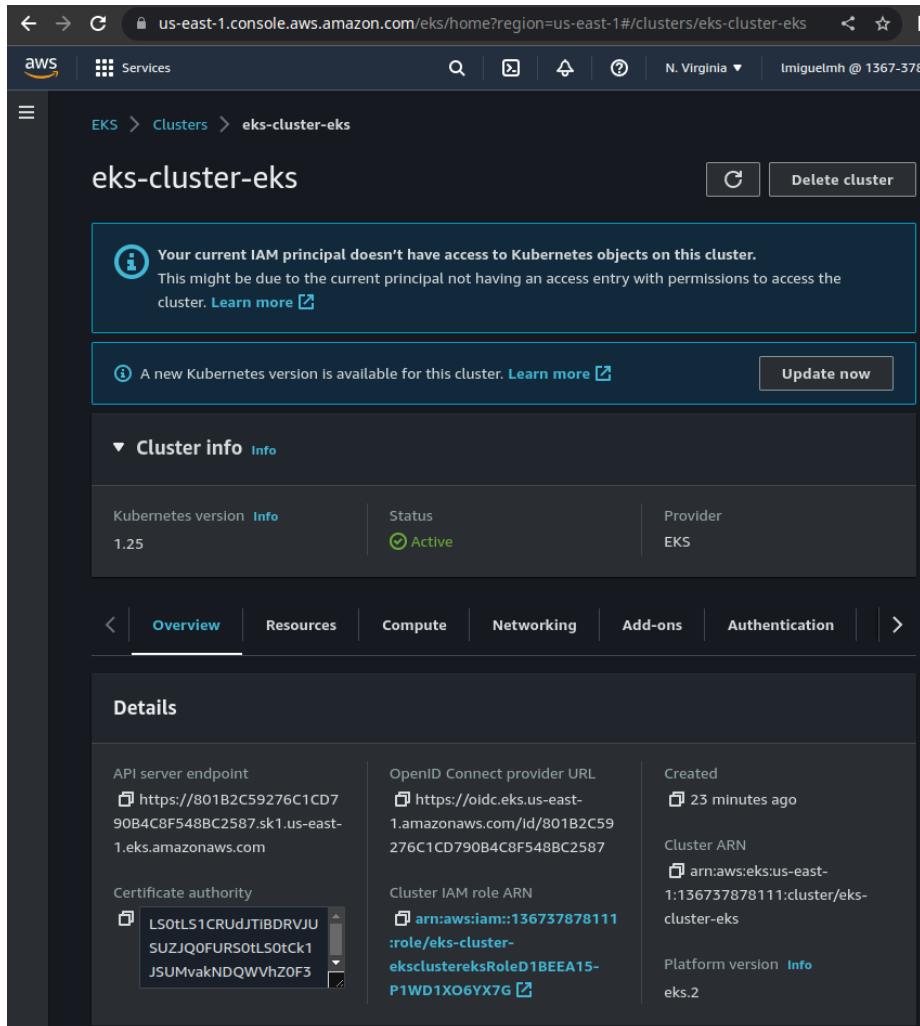


Figure 8: img_73.png

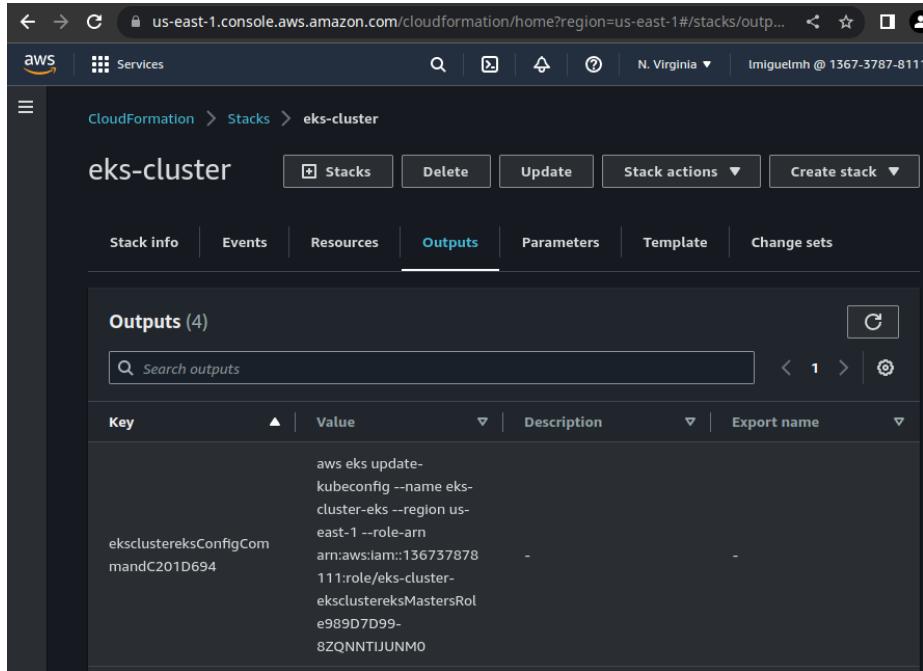


Figure 9: img_74.png

```
Added new context arn:aws:eks:us-east-1:136737878111:cluster/eks-cluster-eks to /home/lmiguel/.kube/config
```

Figure 10: img_75.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl describe configmap -n kube-system aws-auth
Name:      aws-auth
Namespace: kube-system
Labels:    aws.cdk.eks/prune-c80fbe9378b5adbd62d19640ec0427f05937061ea7=
Annotations: <none>

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

mapRoles:
-----
[{"rolearn":"arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0","username":"arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0","groups":["system:masters"]},{ "rolearn": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksclusterasInstanceRole-XOF7XUUJDVHL", "username": "system:node:{EC2PrivateDNSName}"}, {"groups": ["system:bootstrappers", "system:nodes"]}], "mapUsers: "
-----
[]

BinaryData
=====
```

Figure 11: img_76.png

```
apiVersion: v1
data:
mapAccounts: []
mapRoles: [{"rolearn": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0", "username": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0", "groups": ["system:masters"]}], "mapUsers: "
-----
[]

BinaryData
=====
```

Figure 12: img_77.png

```
mapAccounts:guel@lmiguel-pc cdk-eks-cluster]$ kubectl edit -n kube-system configmap/aws-auth
-----
[]
mapRoles:
-----
[{"rolearn": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0", "username": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksMastersRole989D7D99-8ZQNNTIJUNM0", "groups": ["system:masters"]},{ "rolearn": "arn:aws:iam::136737878111:role/eks-cluster-eksclustereksclusterasInstanceRole-XOF7XUUJDVHL", "username": "system:node:{EC2PrivateDNSName}"}, {"groups": ["system:bootstrappers", "system:nodes"]}], "mapUsers: "
-----
- username: arn:aws:iam::136737878111:user/lmiguelmh
  username: lmiguelmh
  groups:
    - system:masters

BinaryData
=====

Events: <none>
```

Figure 13: img_78.png

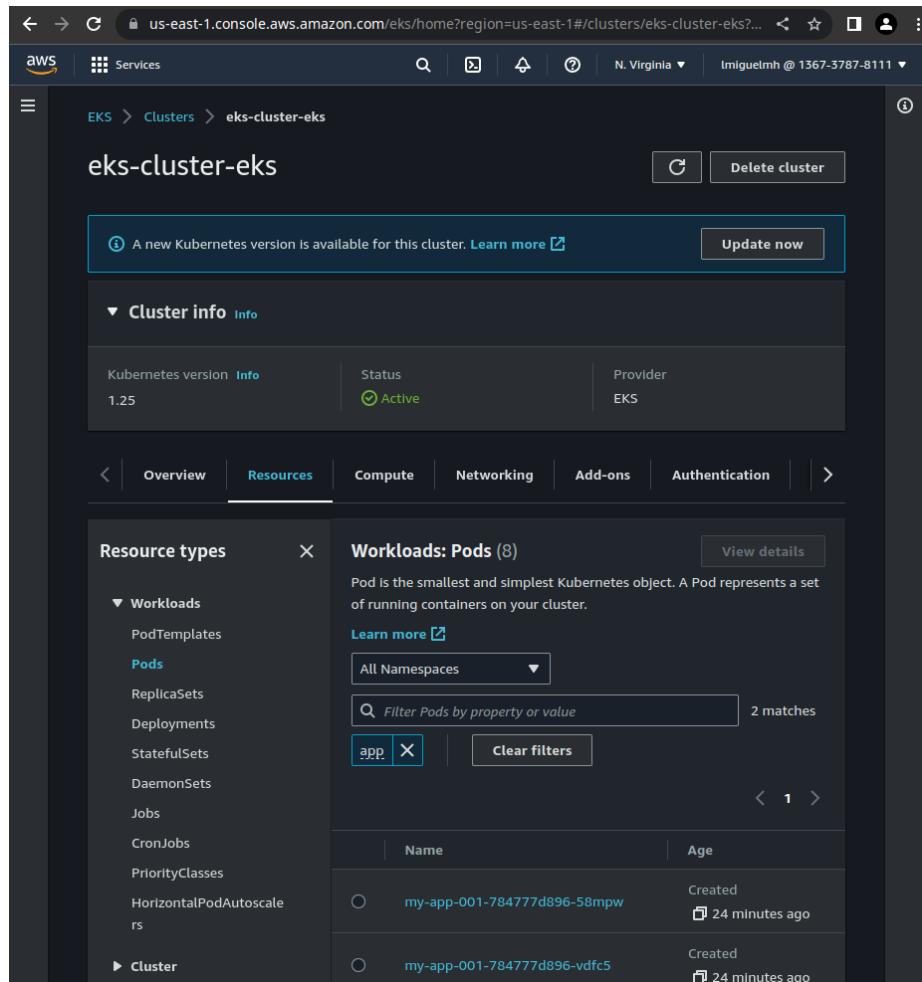


Figure 14: img_79.png

Antes de realizar el despliegue se requiere acceder a la consola y configurar la conexión a Github aquí.

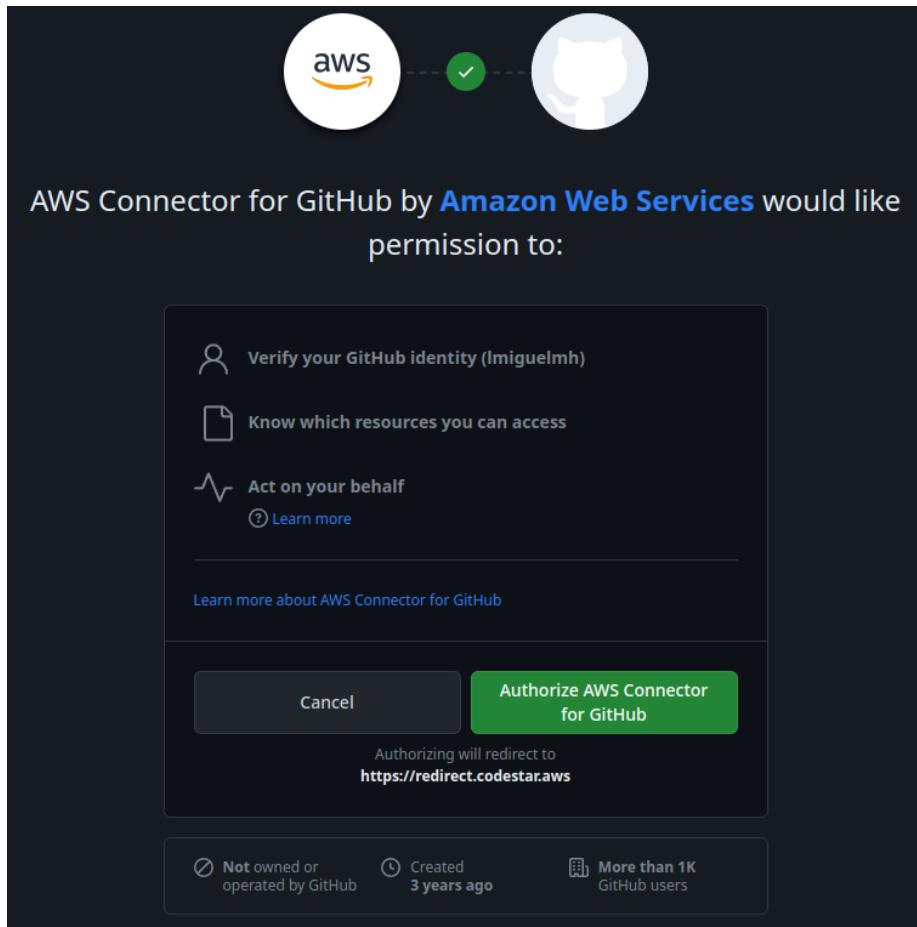


Figure 15: img_28.png

Al finalizar, completar la información en el archivo de configuración `core.conf.{ENV}`.

```
# set the environment/configuration
# on this case we will use the configuration defined on `core.conf.common` and `core.conf.dev`
export ENV=dev

# deploy the pipeline
cdk deploy eks-toolchain
```



Figure 16: img_29.png

Instalar herramientas de monitoreo

Despliegue desde local

El despliegue desde local permite el desarrollo ágil y el despliegue de uno o varios stacks sin necesidad de desplegar toda la aplicación.

```
# set the environment/configuration
export ENV=sandbox

# here we deploy the EKS cluster
cdk deploy eks-cluster

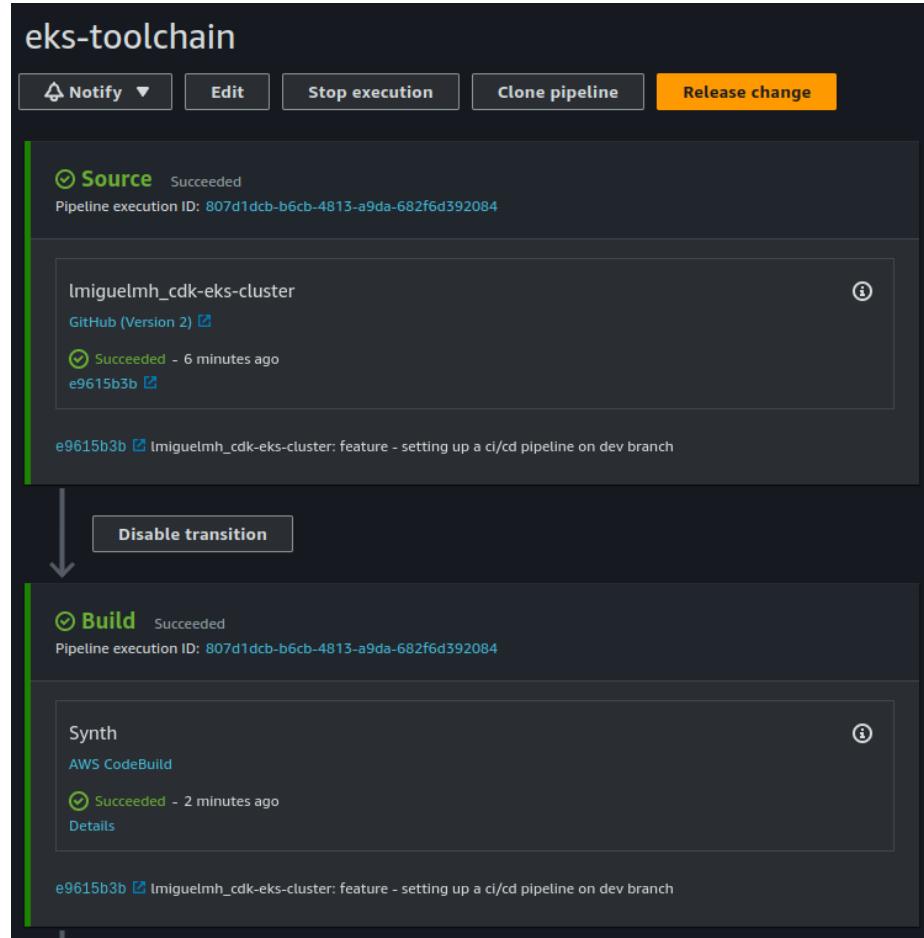
# update kube configuration to access the EKS cluster
# run the command located on the output of ClusterStack
aws eks update-kubeconfig ...

# test kubectl
kubectl get all

# beware that resources created by kubectl need to be deleted manually (ie. load balancers)
kubectl apply -f pod.yml
kubectl get pods
kubectl delete -f pod.yml
```

Despliegue desde pipeline

El despliegue desde el pipeline se dispara automáticamente cuando se realizan cambios en la rama configurada. En entorno `dev` la rama configurada es `dev`.



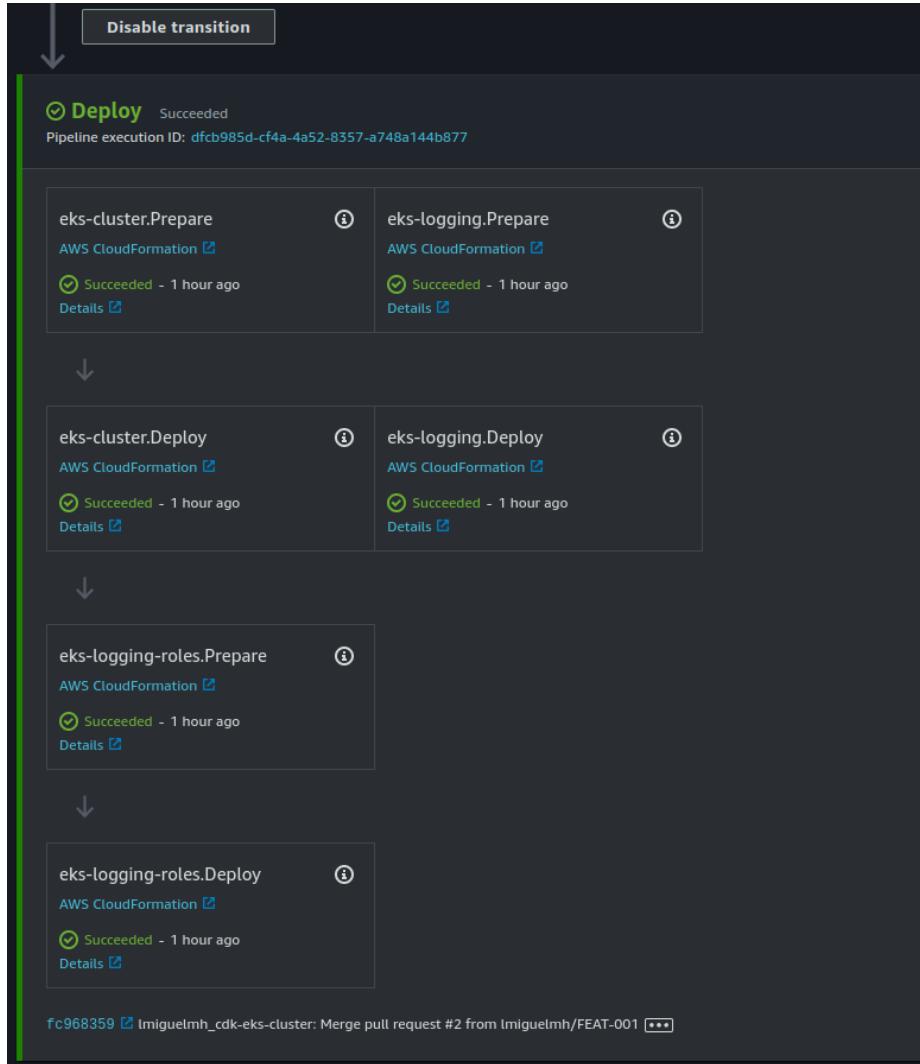
↓

Disable transition

Assets Succeeded
Pipeline execution ID: dfcb985d-cf4a-4a52-8357-a748a144b877

FileAsset1 AWS CodeBuild Succeeded - 24 minutes ago Details	FileAsset10 AWS CodeBuild Succeeded - 23 minutes ago Details	FileAsset11 AWS CodeBuild Succeeded - 22 minutes ago Details
FileAsset2 AWS CodeBuild Succeeded - 17 minutes ago Details	FileAsset3 AWS CodeBuild Succeeded - 16 minutes ago Details	FileAsset4 AWS CodeBuild Succeeded - 14 minutes ago Details
FileAsset7 AWS CodeBuild Succeeded - 18 minutes ago Details	FileAsset8 AWS CodeBuild Succeeded - 11 minutes ago Details	FileAsset9 AWS CodeBuild Succeeded - 10 minutes ago Details

fc968359 [] lmiguelmh_cdk-eks-cluster: Merge pull request #2 from lmiguelmh/FEAT-001 ***



Configuración del Cluster EKS

- La definición del cluster se encuentra en ClusterStack.
- Se usó CDK para la creación de la infraestructura.
 - El aprovisionamiento de los nodos se realiza con un ASG.
 - Otros recursos son aprovisionados
- La definición del servicio y el despliegue de la aplicación de ejemplo también se encuentra en ClusterStack.

The terminal window shows the output of the command `kubectl get all`, listing pods, services, deployments, and replicaset status. The browser window shows the Kubernetes dashboard with a 'Hello world!' message and details about a pod and node.

```

Terminal: Local x + ↵
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all
NAME                               READY   STATUS    RESTARTS   AGE
pod/my-app-001-784777d896-fvq6c   1/1    Running   0          7m11s
pod/my-app-001-784777d896-krz69   1/1    Running   0          7m11s

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/kubernetes   ClusterIP   172.20.8.1    <none>           443/TCP     18m
service/my-app-001   LoadBalancer 172.20.190.68  a7bcd0c4212e8454383ad01a9fba667d-995660036.us-west-2.elb.amazonaws.com  80:30641/TCP  7m11s

NAME             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-app-001  2/2     2           2           7m11s

NAME            DESIRED   CURRENT   READY   AGE
replicaset.apps/my-app-001-784777d896  2         2         2         7m11s

| a7bcd0c4212e8454383ad01a9fba667d-995660036.us-west-2.elb.amazonaws... ☆ □

```

Kubernetes

Hello world!

pod: my-app-001-784777d896-krz69
node: Linux (5.15.102)

Configuración de OpenSearch + Fluent Bit

- La definición del cluster se encuentra en ClusterLoggingStack.
 - Se usó CDK para la creación de la infraestructura.
 - Se usó autenticación por Cognito User Pools en vez de un Master password.
 - Así mismo, se creó un serviceAccount para permitir que los pods puedan acceder al API de ES.
-
- El mapping de los roles de ES/fluent-bit se encuentra en ClusterLoggingRolesStack.
 - Importante. Incluir el rol creado en el paso anterior.

```

# add service account for fluent-bit
service_account: eks.ServiceAccount = cluster.add_service_account(
    id="fluent-bit",
    name="fluent-bit",
)
service_account.add_to_principal_policy(iam.PolicyStatement(
    actions=["es:ESHttp*"],
    resources=["*"], # lax permissions
    effect=iam.Effect.ALLOW
))

```

Figure 17: img_23.png

```

# create fluent-bit
# before, edit the file and change the namespace, cluster endpoint and aws region
kubectl apply -f fluentbit.yaml

# there should be 3 pods for fluent-bit
kubectl get pods

# cleanup
kubectl delete -f fluentbit.yaml

```

Configuración de Prometheus + Grafana

```

# install helm
# helm 3.9+ breaks some packages, awscliv2 should solve this but in my case didn't
# curl -sSL https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
# installing helm 3.8.2
curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2
helm version --short
helm repo add stable https://charts.helm.sh/stable
helm search repo stable

# add prometheus repo
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
# add grafana repo
helm repo add grafana https://grafana.github.io/helm-charts
# add support for volumes on EBS
#helm repo add aws-ebs-csi-driver https://kubernetes-sigs.github.io/aws-ebs-csi-driver
#helm repo update
#helm upgrade --install aws-ebs-csi-driver --namespace kube-system aws-ebs-csi-driver/aws-ebs-csi-driver
# install eksctl - https://github.com/weaveworks/eksctl/releases/

```

Figure 18: img_9.png

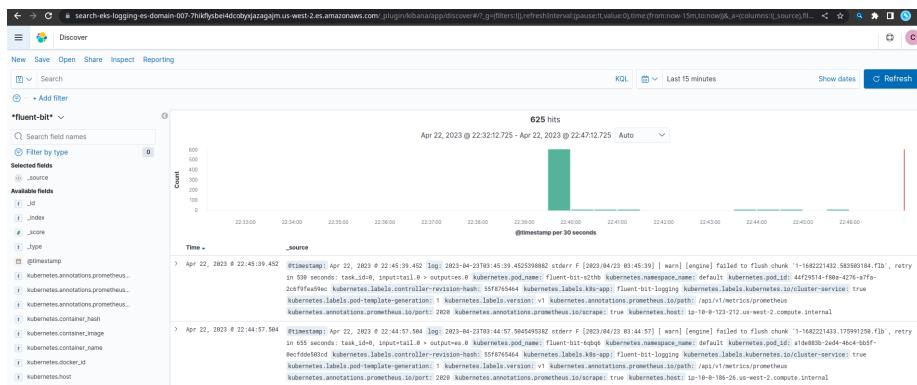


Figure 19: img_10.png

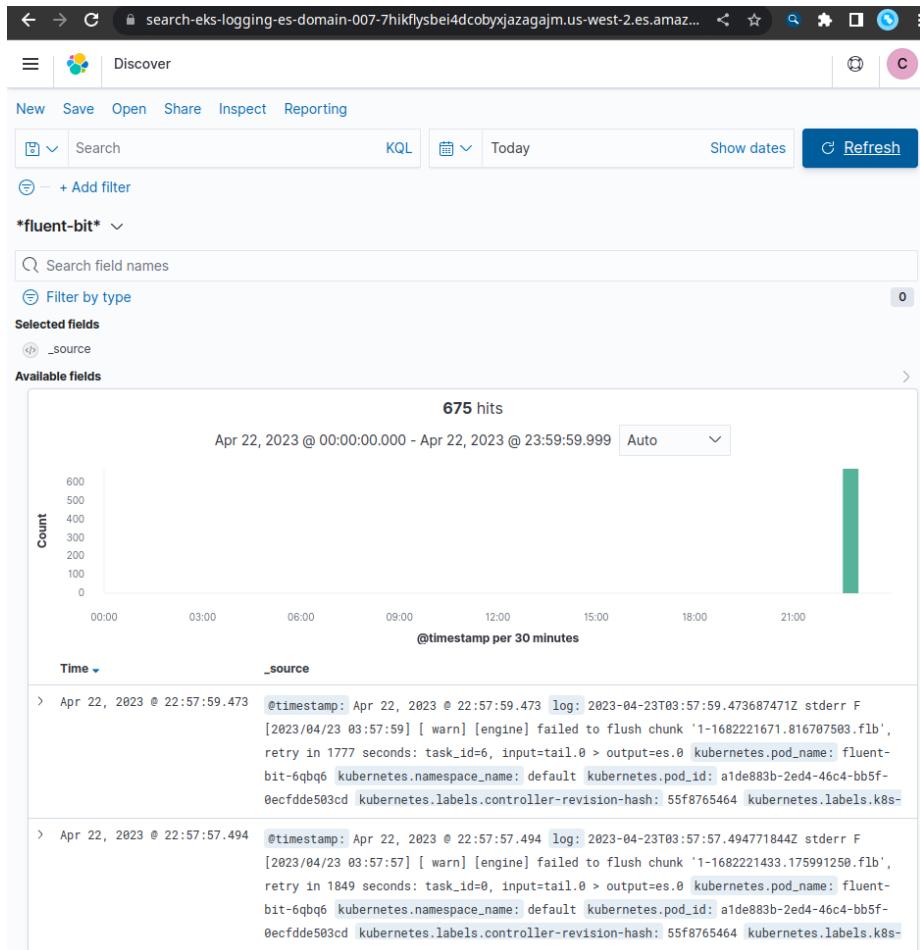


Figure 20: img_11.png

```

# https://docs.aws.amazon.com/eks/latest/userguide/csi-iam-role.html
# it will create a policy/role and *annotate* the **existing** ebs-csi-controller-sa service
# it will NOT update the ebs-csi-controller-sa (it already exists!)
eksctl create iamserviceaccount \
    --name ebs-csi-controller-sa \
    --namespace kube-system \
    --cluster eks-cluster-eks \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
    --approve \
    --role-only \
    --role-name AmazonEKS_EBS_CSI_DriverRole
# https://docs.aws.amazon.com/eks/latest/userguide/managing-ebs-csi.html
# create the addon and attach the role created
eksctl create addon \
    --name aws-ebs-csi-driver \
    --cluster eks-cluster-eks \
    --service-account-role-arn arn:aws:iam::136737878111:role/AmazonEKS_EBS_CSI_DriverRole --i

# test EBS CSI driver
# by creating a StorageClass, a PersistentVolumeClaim (PVC) and a pod
# all at once
#kubectl apply -f dynamic-provisioning/
#kubectl get pods
#kubectl delete -f dynamic-provisioning/
# or step by step
kubectl apply -f gp3-sc.yaml
kubectl apply -f pvc-csi.yaml
kubectl apply -f pod-csi.yaml
# pod status should be RUNNING after ~60s
kubectl get pod --watch
# pvc status should be BOUND
kubectl get pvc
# check more details of PVC
kubectl describe pvc
# cleanup
kubectl delete -f pod-csi.yaml
kubectl delete -f pvc-csi.yaml
kubectl delete -f gp3-sc.yaml

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

```

```

# check pods
kubectl get pods -n prometheus --watch

# does not work
# export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus-pushgateway",
# # kubectl --namespace prometheus port-forward $POD_NAME 9091

# working
kubectl expose deployment prometheus-server --type=LoadBalancer --name prometheus-server-pu

# cleanup
helm uninstall prometheus --namespace prometheus
kubectl delete ns prometheus

```

NAME	READY	STATUS	RESTARTS	AGE
pod/prometheus-alertmanager-0	0/1	Pending	0	46s
pod/prometheus-kube-state-metrics-5fb6fbbf78-5clpv	1/1	Running	0	46s
pod/prometheus-prometheus-node-exporter-7lnbx	1/1	Running	0	46s
pod/prometheus-prometheus-node-exporter-8k2sv	1/1	Running	0	46s
pod/prometheus-prometheus-node-exporter-hm5ph	1/1	Running	0	46s
pod/prometheus-prometheus-pushgateway-7d55869d46-cqwqp	1/1	Running	0	46s
pod/prometheus-server-77df547d88-l8rpn	0/2	Pending	0	46s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/prometheus-alertmanager	ClusterIP	172.20.66.155	<none>	9093/TCP	47s
service/prometheus-alertmanager-headless	ClusterIP	None	<none>	9093/TCP	47s
service/prometheus-kube-state-metrics	ClusterIP	172.20.240.90	<none>	8080/TCP	47s
service/prometheus-prometheus-node-exporter	ClusterIP	172.20.106.67	<none>	9100/TCP	47s
service/prometheus-prometheus-pushgateway	ClusterIP	172.20.29.55	<none>	9091/TCP	47s
service/prometheus-server	ClusterIP	172.20.163.69	<none>	80/TCP	47s

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
daemonset.apps/prometheus-prometheus-node-exporter	3	3	3	3	3	<none>	46s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/prometheus-kube-state-metrics	1/1	1	1	47s
deployment.apps/prometheus-prometheus-pushgateway	1/1	1	1	47s
deployment.apps/prometheus-server	0/1	1	0	47s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/prometheus-kube-state-metrics-5fb6fbbf78	1	1	1	47s
replicaset.apps/prometheus-prometheus-pushgateway-7d55869d46	1	1	1	47s
replicaset.apps/prometheus-server-77df547d88	1	1	0	47s

NAME	READY	AGE
statefulset.apps/prometheus-alertmanager	0/1	47s

Figure 21: img_13.png

```

kubectl create namespace grafana

# doesnt work
helm install grafana grafana/grafana \
--namespace grafana \
--set persistence.storageClassName="gp2" \
--set persistence.enabled=true \

```

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pods -n prometheus --watch
NAME                               READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0          1/1    Running   0          16s
prometheus-kube-state-metrics-5fb6fbbf78-rc5jv  1/1    Running   0          17s
prometheus-prometheus-node-exporter-hhb7w        1/1    Running   0          17s
prometheus-prometheus-pushgateway-7d55869d46-9r55r  0/1    Running   0          17s
prometheus-server-77df547d88-6hb6l      0/2    ContainerCreating   0          17s
prometheus-server-77df547d88-6hb6l      1/2    Running   0          20s
prometheus-prometheus-pushgateway-7d55869d46-9r55r  1/1    Running   0          21s
prometheus-server-77df547d88-6hb6l      2/2    Running   0          49s
```

Figure 22: img_37.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n prometheus
NAME                               READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0          1/1    Running   0          3m16s
pod/prometheus-kube-state-metrics-5fb6fbbf78-rc5jv  1/1    Running   0          3m17s
pod/prometheus-prometheus-node-exporter-hhb7w        1/1    Running   0          3m17s
pod/prometheus-prometheus-pushgateway-7d55869d46-9r55r  1/1    Running   0          3m17s
pod/prometheus-server-77df547d88-6hb6l      2/2    Running   0          3m17s

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/prometheus-alertmanager     ClusterIP   172.20.239.117  <none>       9093/TCP   3m17s
service/prometheus-alertmanager-headless ClusterIP   None           <none>       9093/TCP   3m17s
service/prometheus-kube-state-metrics   ClusterIP   172.20.22.62    <none>       8080/TCP   3m17s
service/prometheus-prometheus-node-exporter   ClusterIP   172.20.255.155  <none>       9100/TCP   3m17s
service/prometheus-prometheus-pushgateway     ClusterIP   172.20.4.161    <none>       9091/TCP   3m17s
service/prometheus-server             ClusterIP   172.20.19.36    <none>       80/TCP     3m17s

NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODES
ELECTOR
daemonset.apps/prometheus-prometheus-node-exporter  1         1         1         1           1           <none>
3m18s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics  1/1     1           1           3m18s
deployment.apps/prometheus-prometheus-pushgateway 1/1     1           1           3m18s
deployment.apps/prometheus-server               1/1     1           1           3m18s

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-5fb6fbbf78  1         1         1         3m18s
replicaset.apps/prometheus-prometheus-pushgateway-7d55869d46  1         1         1         3m18s
replicaset.apps/prometheus-server-77df547d88            1         1         1         3m18s

NAME          READY   AGE
statefulset.apps/prometheus-alertmanager  1/1     3m18s
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$
```

Figure 23: img_38.png

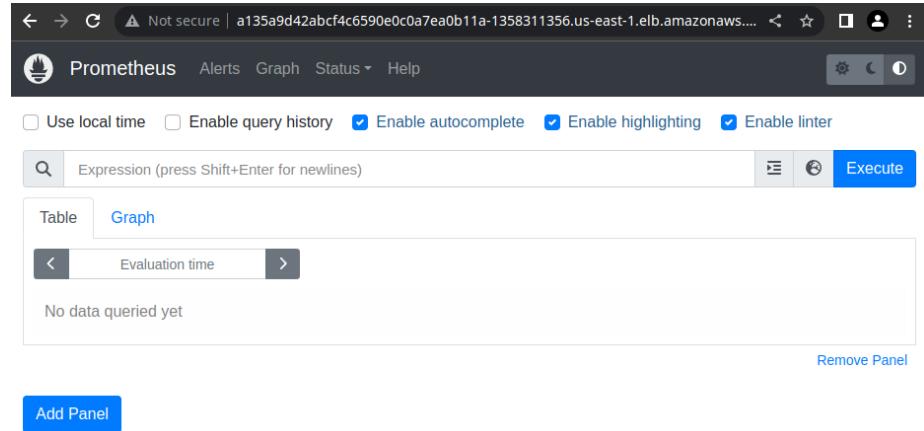


Figure 24: img_39.png

```
--set adminPassword='EKS!sAWSome' \
--values grafana.yml \
--set service.type=LoadBalancer
```

Luego de eliminar, cambiar el tamaño de los worker nodos, el despliegue funcionó correctamente.

← → ⌂ Not secure | a135a9d42abcf4c6590e0c0a7ea0b11a-1358311356.us-east-1.elb.amazonaws.com | 🔍

 Prometheus Alerts Graph Status Help

Targets

All scrape pools ▾ All Unhealthy Collapse All Filter by endpoint: Unknown Unhealthy Healthy

kubernetes-apiservers (2/2 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://10.0.111.211:443/metrics	UP	instance="10.0.111.211:443" job="kubernetes-apiservers"	51.535s ago	124.814ms	
https://10.0.133.206:443/metrics	UP	instance="10.0.133.206:443" job="kubernetes-apiservers"	671.000ms ago	103.736ms	

kubernetes-nodes (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/p-10-0-9-142.ec2.internal/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="t3.medium" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="us-east-1" failure_domain_beta_kubernetes_io_zone="us-east-1a" instance="ip-10-0-9-142.ec2.internal" job="kubernetes-nodes" k8s.io_cloud_provider_aws="7f2d95bed7bb8d2cfbe2ff30992e082c" kubernetes_io_arch="amd64" kubernetes_io_hostname="ip-10-0-9-142.ec2.internal" kubernetes_io_os="linux" lifecycle="OnDemand" node_kubernetes_io_instance_type="t3.medium" topology_ebs_csi_aws_com_zone="us-east-1a" topology_kubernetes_io_region="us-east-1" topology_kubernetes_io_zone="us-east-1a"	10.702s ago	108.549ms	

kubernetes-nodes-cadvisor (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc/api/v1/nodes/p-10-0-9-142.ec2.internal/proxy/metrics	UP	beta_kubernetes_io_arch="amd64"	7.548s ago	254.231ms	

Figure 25: img_40.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl create namespace grafana
namespace/grafana created
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ helm install grafana grafana/grafana \
    --namespace grafana \
    --set persistence.storageClassName="gp2" \
    --set persistence.enabled=true \
    --set adminPassword='EKS!sAWSome' \
    --values grafana.yaml \
    --set service.type=LoadBalancer
NAME: grafana
LAST DEPLOYED: Sun Apr 30 18:47:55 2023
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

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

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

    grafana.grafana.svc.cluster.local

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

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

Figure 26: img_41.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ helm install grafana grafana/grafana \
--namespace grafana \
--set persistence.storageClassName="gp2" \
--set persistence.enabled=true \
--set adminPassword='EKS!sAWSome' \
--values grafana.yaml \
--set service.type=LoadBalancer
NAME: grafana
LAST DEPLOYED: Sun Apr 30 21:07:57 2023
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

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

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

    grafana.grafana.svc.cluster.local

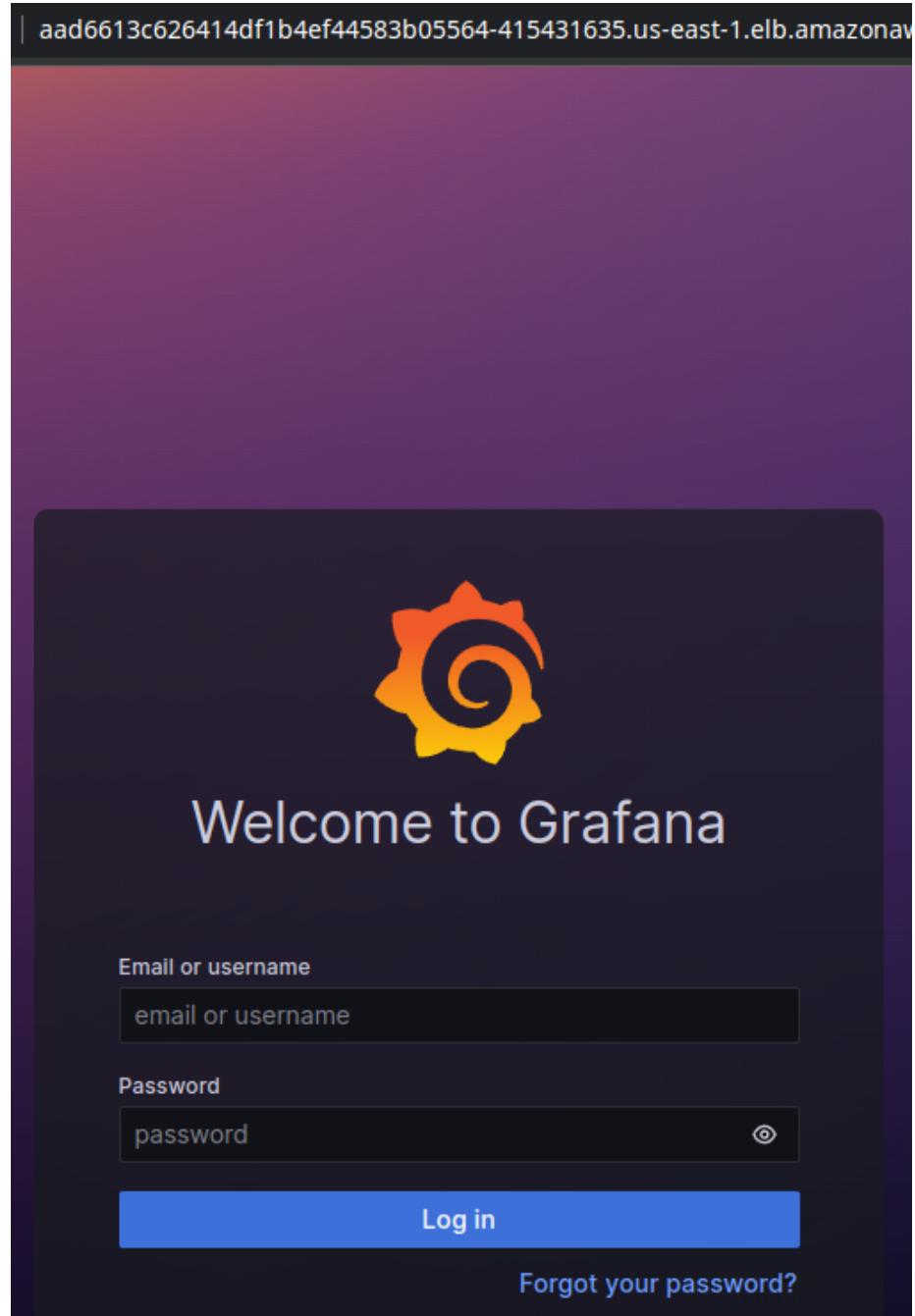
Get the Grafana URL to visit by running these commands in the same shell:
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
      You can watch the status of by running 'kubectl get svc --namespace grafana -w grafana'
      export SERVICE_IP=$(kubectl get svc --namespace grafana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

3. Login with the password from step 1 and the username: admin
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pods -n grafana --watch
NAME           READY   STATUS    RESTARTS   AGE
grafana-5867d65f94-5qrxx  0/1     Init:0/1  0          9s
grafana-5867d65f94-5qrxx  0/1     PodInitializing  0          18s
grafana-5867d65f94-5qrxx  0/1     Running   0          24s
grafana-5867d65f94-5qrxx  1/1     Running   0          36s
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n grafana
NAME           READY   STATUS    RESTARTS   AGE
pod/grafana-5867d65f94-5qrxx  1/1     Running   0          4m43s

NAME            TYPE        CLUSTER-IP      EXTERNAL-IP
              PORT(S)      AGE
service/grafana  LoadBalancer  172.20.38.207  aad6613c626414df1b4ef44583b05564-415431635.us-east-1.elb.amazonaws.com  80:32205/TCP  4m43s

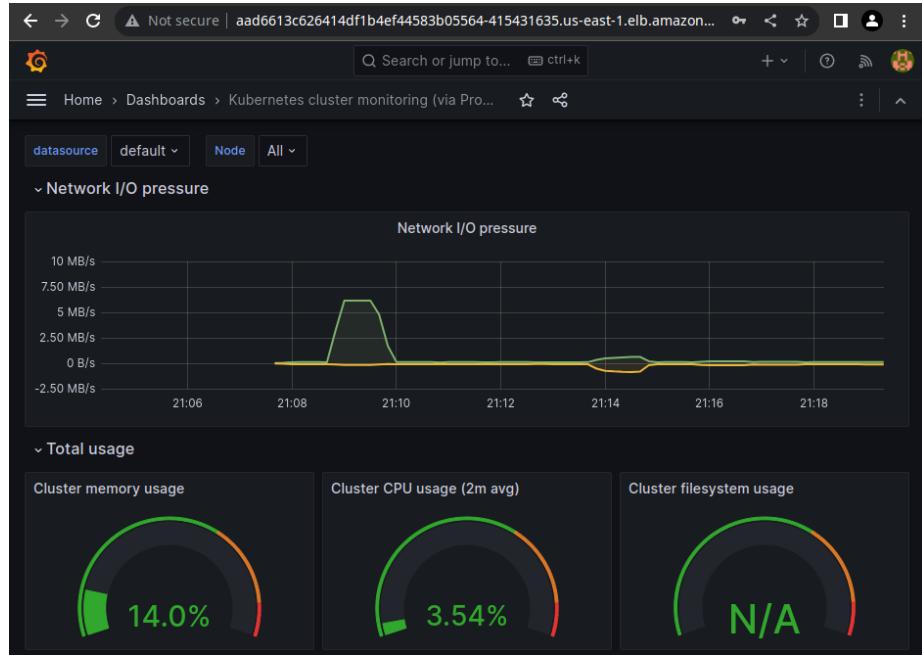
NAME           READY   UP-TO-DATE  AVAILABLE   AGE
deployment.apps/grafana  1/1     1           1           4m44s

NAME           DESIRED  CURRENT  READY   AGE
replicaset.apps/grafana-5867d65f94  1       1       1       4m44s
```



The top screenshot shows the Grafana 'Welcome' page. It features a 'Basic' section with instructions for finishing setup, a 'Tutorial' section for 'DATA SOURCE AND DASHBOARDS' and 'Grafana fundamentals', a 'COMPLETE' section for adding a first data source, and a 'DASHBOARD' section for creating your own dashboard. The bottom screenshot shows the 'Import dashboard' dialog box, which is importing a dashboard from Grafana.com. The dashboard is titled 'Kubernetes cluster monitoring (via Prometheus)' and was published by 'Jjo Org' on '2017-09-08 10:22:08'. The 'Options' section includes fields for 'Name' (set to 'Kubernetes cluster monitoring (via Prometheus)'), 'Folder' (set to 'General'), and 'Unique Identifier (UID)' (set to 'prometheus'). The 'Prometheus' icon is selected. At the bottom are 'Import' and 'Cancel' buttons.

Luego de seleccionar el ID **3119** y el datasource **Prometheus**:



The screenshot shows the 'Import dashboard' dialog in Grafana. At the top, it says 'Importing dashboard from Grafana.com'. Below that, it shows the dashboard's details: 'Published by' sekka1 and 'Updated on' 2018-06. Under 'Options', there are fields for 'Name' (set to 'Kubernetes Cluster (Prometheus)'), 'Folder' (set to 'General'), and 'Unique identifier (UID)' (set to '4XuMd2iz'). There is also a 'prometheus' entry under 'Datasources'. At the bottom are 'Import' and 'Cancel' buttons.

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by sekka1

Updated on 2018-06

Name

Kubernetes Cluster (Prometheus)

Folder

General

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

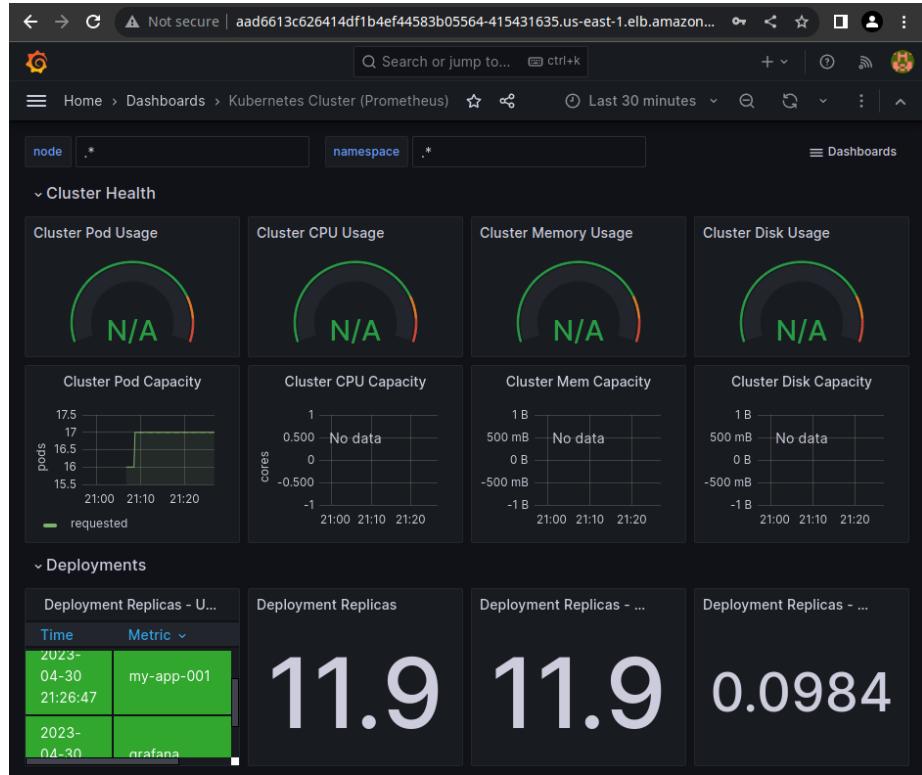
4XuMd2iz

prometheus

Prometheus

Import Cancel

Repetimos para el ID **6417** y el datasource **Prometheus**:



Configuración manual

```

eksctl create cluster -f config.yaml
kubectl config current-context
kubectl apply -f gp3-sc.yaml
kubectl apply -f pvc-csi.yaml
# check resources
eksctl get cluster
eksctl get addon --name aws-ebs-csi-driver --cluster ebs-demo-cluster
kubectl get sc

# install
helm install prometheus prometheus-community/prometheus \
--namespace prometheus \
--set alertmanager.persistentVolume.storageClass="gp3" \
--set server.persistentVolume.storageClass="gp3"
# check prometheus
kubectl get all -n prometheus

# cleanup
helm uninstall prometheus --namespace prometheus

```

```

kubectl delete -f pvc-csi.yaml
kubectl delete -f gp3-sc.yaml
eksctl delete addon --name aws-ebs-csi-driver --cluster ebs-demo-cluster

2023-04-25 21:18:06 [i] waiting for CloudFormation stack "eksctl-ebs-demo-cluster-nodegroup-managed-ng-1"
2023-04-25 21:18:06 [i] waiting for the control plane to become ready
2023-04-25 21:18:06 [✓] saved kubeconfig as "/home/lmiguel/.kube/config"
2023-04-25 21:18:06 [i] no tasks
2023-04-25 21:18:06 [✓] all EKS cluster resources for "ebs-demo-cluster" have been created
2023-04-25 21:18:07 [i] nodegroup "managed-ng-1" has 1 node(s)
2023-04-25 21:18:07 [i] node "ip-192-168-173-173.us-west-2.compute.internal" is ready
2023-04-25 21:18:07 [i] waiting for at least 1 node(s) to become ready in "managed-ng-1"
2023-04-25 21:18:07 [i] nodegroup "managed-ng-1" has 1 node(s)
2023-04-25 21:18:07 [i] node "ip-192-168-173-173.us-west-2.compute.internal" is ready
2023-04-25 21:18:09 [i] creating role using provided policies ARNs
2023-04-25 21:18:10 [i] deploying stack "eksctl-ebs-demo-cluster-addon-aws-ebs-csi-driver"
2023-04-25 21:18:10 [i] waiting for CloudFormation stack "eksctl-ebs-demo-cluster-addon-aws-ebs-csi-driver"
2023-04-25 21:18:41 [i] waiting for CloudFormation stack "eksctl-ebs-demo-cluster-addon-aws-ebs-csi-driver"
2023-04-25 21:18:41 [i] creating addon
2023-04-25 21:19:28 [i] addon "aws-ebs-csi-driver" active
2023-04-25 21:19:30 [i] kubectl command should work with "/home/lmiguel/.kube/config", try 'kubectl get nodes'
2023-04-25 21:19:30 [✓] EKS cluster "ebs-demo-cluster" in "us-west-2" region is ready
NAME          REGION      EKSCONTROLLED
ebs-demo-cluster    us-west-2    True
2023-04-25 21:19:35 [i] Kubernetes version "1.22" in use by cluster "ebs-demo-cluster"
2023-04-25 21:19:36 [i] to see issues for an addon run 'eksctl get addon --name <addon-name> --cluster <cluster-name>'

NAME          VERSION      STATUS   ISSUES   IAMROLE
ONFIGURATION VALUES
aws-ebs-csi-driver    v1.5.2-eksbuild.1    ACTIVE   0        arn:aws:iam::719602558560:role/eksctl-ebs-csi-driver
  kubectl --namespace prometheus port-forward $POD_NAME 9091
#####
##### WARNING: Pod Security Policy has been disabled by default since #####
##### it deprecated after k8s 1.25+. use #####
##### (index .Values "prometheus-node-exporter" "rbac" #####
##### .      "pspEnabled") with (index .Values #####
##### "prometheus-node-exporter" "rbac" "pspAnnotations") #####
##### in case you still need it. #####
#####

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

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

For more information on running Prometheus, visit:
https://prometheus.io/
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$
```

```
(.venv) [lmiiguel@lmiiguel-pc cdk-eks-cluster]$ kubectl get all -n prometheus
NAME                                         READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0                0/1     Pending   0          4m24s
pod/prometheus-kube-state-metrics-6dc44cc4d9-kv7m8 1/1     Running   0          4m24s
pod/prometheus-prometheus-node-exporter-b4kl4    1/1     Running   0          4m24s
pod/prometheus-prometheus-pushgateway-5fdcccb6f7-8xj5b 0/1     Pending   0          4m24s
pod/prometheus-server-76c879bccf-v44sm         2/2     Running   0          4m24s

NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/prometheus-alertmanager   ClusterIP   10.100.63.205 <none>        9093/TCP   4m24s
service/prometheus-alertmanager-headless ClusterIP   None            <none>        9093/TCP   4m24s
service/prometheus-kube-state-metrics   ClusterIP   10.100.132.135 <none>        8080/TCP   4m24s
service/prometheus-prometheus-node-exporter ClusterIP   10.100.159.97  <none>        9100/TCP   4m24s
service/prometheus-prometheus-pushgateway   ClusterIP   10.100.169.133 <none>        9091/TCP   4m24s
service/prometheus-server             ClusterIP   10.100.227.81  <none>        80/TCP     4m24s

NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODES
ELECTOR     AGE
daemonset.apps/prometheus-prometheus-node-exporter 1         1         1         1           1           <none>
4m24s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics 1/1     1           1           4m25s
deployment.apps/prometheus-prometheus-pushgateway 0/1     1           0           4m25s
deployment.apps/prometheus-server               1/1     1           1           4m25s

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-6dc44cc4d9 1         1         1         4m25s
replicaset.apps/prometheus-prometheus-pushgateway-5fdcccb6f7 1         1         0         4m25s
replicaset.apps/prometheus-server-76c879bccf           1         1         1         4m25s

NAME          READY   AGE
statefulset.apps/prometheus-alertmanager 0/1     4m25s
```

Cleanup

Eliminando pasos manuales

```
kubectl delete -f fluentbit.yaml
helm uninstall prometheus -n prometheus
kubectl delete ns prometheus
helm uninstall grafana -n grafana
kubectl delete ns grafana
```

```
eksctl delete addon \
--name aws-ebs-csi-driver \
--cluster eks-cluster-eks
eksctl delete iamserviceaccount \
--name ebs-csi-controller-sa \
--namespace kube-system \
--cluster eks-cluster-eks \
--wait
```

```

(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ helm uninstall prometheus -n prometheus
release "prometheus" uninstalled
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl delete ns prometheus
namespace "prometheus" deleted
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ helm uninstall grafana -n grafana
release "grafana" uninstalled
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl delete ns grafana
namespace "grafana" deleted
2023-04-30 21:36:32 [i] Kubernetes version "1.25" in use by cluster "eks-cluster-eks"
2023-04-30 21:36:32 [i] deleting addon: aws-ebs-csi-driver
2023-04-30 21:36:33 [i] deleted addon: aws-ebs-csi-driver
2023-04-30 21:36:33 [i] no associated IAM stacks found
2023-04-30 21:36:35 [i] 1 iamserviceaccount (kube-system/ebs-csi-controller-sa) was included (based on the
include/exclude rules)
2023-04-30 21:36:36 [i] 1 task: {
    2 sequential sub-tasks: {
        delete IAM role for serviceaccount "kube-system/ebs-csi-controller-sa",
        delete serviceaccount "kube-system/ebs-csi-controller-sa",
    } }2023-04-30 21:36:36 [i] will delete stack "eksctl-eks-cluster-eks-addon-iamserviceaccount-kube-syste
m-ebs-csi-controller-sa"
2023-04-30 21:36:36 [i] waiting for stack "eksctl-eks-cluster-eks-addon-iamserviceaccount-kube-system-ebs-c
si-controller-sa" to get deleted
2023-04-30 21:36:36 [i] waiting for CloudFormation stack "eksctl-eks-cluster-eks-addon-iamserviceaccount-ku
be-system-ebs-csi-controller-sa"
2023-04-30 21:37:07 [i] waiting for CloudFormation stack "eksctl-eks-cluster-eks-addon-iamserviceaccount-ku
be-system-ebs-csi-controller-sa"
2023-04-30 21:37:07 [i] 1 error(s) occurred and IAM Role stacks haven't been deleted properly, you may wish
to check CloudFormation console
2023-04-30 21:37:07 [*] checking whether serviceaccount "kube-system/ebs-csi-controller-sa" exists: Unautho
rized
Error: failed to delete iamserviceaccount(s)

```

Eliminando stacks de ES/OS y EKS.

Eliminando pipeline.

Reporte de costos generados por el proyecto

Algunos de los costos se reducieron debido a la Free Tier de AWS: 100 minutos gratuitos en CodeBuild, 720 horas de instancia EC2 t2.min, etc.

Historial de cambios a repositorios

Problemas

- El despliegue falló porque se llegó al límite de 5 IPs por región.
 - Se solicitó el incremento de número de IPs.
 -
- El despliegue falló por un error en el manifest.
 - Se eliminó el manifest para culminar el despliegue.
 -
- No se pudo usar kubectl desde local.
 - Se eliminó la versión de kubectl 1.26.3-1.
 - Se recreó la carpeta `~/.kube/`
 - Se probó con la versión 1.27, 1.26, 1.25, 1.24, finalmente la versión 1.23.17

Stacks (38)		
	Stack name	Status
○	eks-cluster	DELETE_COMPLETE
○	eks-logging-roles	DELETE_COMPLETE
○	eks-cluster-awscdkawseksKubectlProviderNestedStackawscdkawseksKubectlProviderNested5ta-41RCFWZ15FNP	DELETE_COMPLETE
○	eks-cluster-awscdkawseksClusterResourceProviderNestedStackawscdkawseksClusterResourceProviderNest	DELETE_COMPLETE
○	eks-cluster	DELETE_COMPLETE
○	eks-logging	DELETE_COMPLETE

Figure 27: img_59.png

CloudFormation > Stacks		
Stacks (1)		
	Stack name	Status
○	eks-toolchain	DELETE_COMPLETE

Figure 28: img_61.png

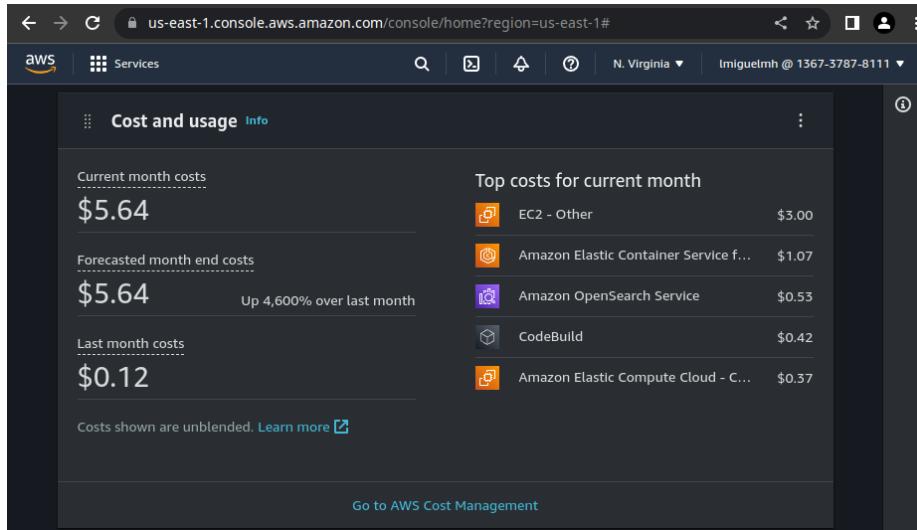


Figure 29: img_60.png

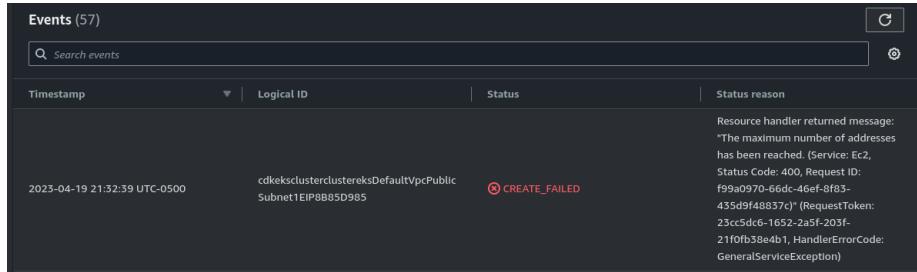


Figure 30: img.png

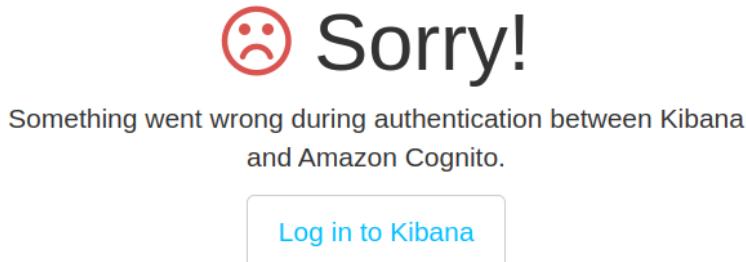
Events (200+)			
Timestamp	Logical ID	Status	Status reason
2023-04-19 22:04:57 UTC-0500	cdkeksclusterclustereksmanifestSamplePod05868EE3	CREATE_FAILED	<p>Received response status [FAILED] from custom resource. Message returned: Error: b'error: error validating "/tmp/manifest.yaml": error validating data: apiVersion not set; if you choose to ignore these errors, turn validation off with --validate=false'\n Logs: /aws/lambda/cdk-eks-cluster-cluster-awsckavse-Handler896CB40B-GmWVQAsudsV at invokeUserFunction (/var/task/framework.js:102:19) at processTicksAndRejections (Internal/process/task_queues.js:95:5) at async onEvent (/var/task/framework.js:22:27) at async Runtime.handler (/var/task/cfn-response.js:52:15) (RequestId: 0047d963-3280-48eb-bad7-178cbc708c0f)</p>

Figure 31: img_1.png

```
Terminal: Local x + ↗
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get nodes
error: exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl version
error: exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ █
```

Figure 32: img_2.png

- No se puede acceder a Kibana con el usuario de Cognito.
 - Posiblemente un error de integración entre UserPool y el IdentityPool. Se añadieron roles y redesplegó.



What happened?

com.amazonaws.services.cognitoidentity.model.InvalidIdentityPoolConfigurationException: Invalid identity pool configuration. Check assigned IAM roles for this pool. (Service: AmazonCognitoIdentity; Status Code: 400; Error Code: InvalidIdentityPoolConfigurationException; Request ID: d0fa1b39-090c-4082-a1de-3450a9119699; Proxy: null)

What should I do?

Try logging in again. If the problem persists, please review the [troubleshooting guide](#) for information on resolving common issues.

Figure 33: img_3.png

- El despliegue de un manifiesto (service + deployment) falla.
 - Errores de versión de manifiesto. Se corrigió y cambiaron algunos nombres.

```
Stack Deployments Failed: Error: The stack named eks-cluster failed to deploy: UPDATE_ROLLBACK_COMPLETE: Received response status [FAILED] from custom resource. Message returned: Error: b'error: error validating "/tmp/manifest.yaml": error validating data: ValidationError(Service.spec.ports[0]): unknown field "target_port" in io.k8s.api.core.v1.ServicePort; if you choose to ignore these errors, turn validation off with --validate=false\n'
```

Figure 34: img_4.png

- Problemas al eliminar el Cluster EKS, al parecer algunas VPCs, IGs y subnets no pueden eliminarse.

- Un balanceador de carga creado con kubectl (manualmente) no podía ser eliminado. Se identificó el balanceador y tuvo que ser eliminado manualmente, luego el stack pudo ser eliminado.

Events (100+)			
Timestamp	Logical ID	Status	Status reason
2023-04-22 16:56:15 UTC-0500	eksclustereksDefaultVpc76B6D86C	✖ DELETE_FAILED	Resource handler returned message: "The vpc 'vpc-0f5425119ad6e9a46' has dependencies and cannot be deleted. (Service: Ec2, Status Code: 400, Request ID: 50b9b2d3-3fef-469f-a80b-b04a46072ef4" (RequestToken: acdfbf75c-f3e7-c605-e680-43143abdf4ab, HandlerErrorCode: InvalidRequest)
2023-04-22 16:42:45 UTC-0500	eksclustereksDefaultVpcIGW06F48215	✖ DELETE_FAILED	The InternetGateway 'igw-0f99db17ebc41c72' has dependencies and cannot be deleted. (Service: Ec2, Status Code: 400, Request ID: e6745108-06c6-4600-beb6-afcb5796e144)

Figure 35: img_8.png

Resources (73)					
Logical ID	Physical ID	Type	Status	Module	
eksclustereksDefaultVpc76B6D86C	vpc-0f5425119ad6e9a46	AWS::EC2::VPC	✖ DELETE_FAILED	-	
eksclustereksDefaultVpcIGW06F48215	igw-0f99db17ebc41c72	AWS::EC2::InternetGateway	✖ DELETE_FAILED	-	
eksclustereksDefaultVpcPublicSubnet1Subnet6780E4D2	subnet-083c4f141ad0475ab	AWS::EC2::Subnet	✖ DELETE_FAILED	-	
eksclustereksDefaultVpcVPCGW915C0673	eks-c-eksc1-1W6J0VYEBXTP6	AWS::EC2::VPCHostAttachment	✖ DELETE_FAILED	-	

Figure 36: img_7.png

- Error al desplegar Prometheus: INSTALLATION FAILED: Kubernetes cluster unreachable: exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"
 - Al parecer es un problema de Helm 3.9 + AWS cli v1.
 - Instalando AWS cli v2 no funcionó (<https://github.com/helm/helm/issues/10975#issuecomment-1132139799>)
 - Tuve que revertir y usar la v3.8.2 de Helm.
 -
- El pod de helm se queda en *Pending*.

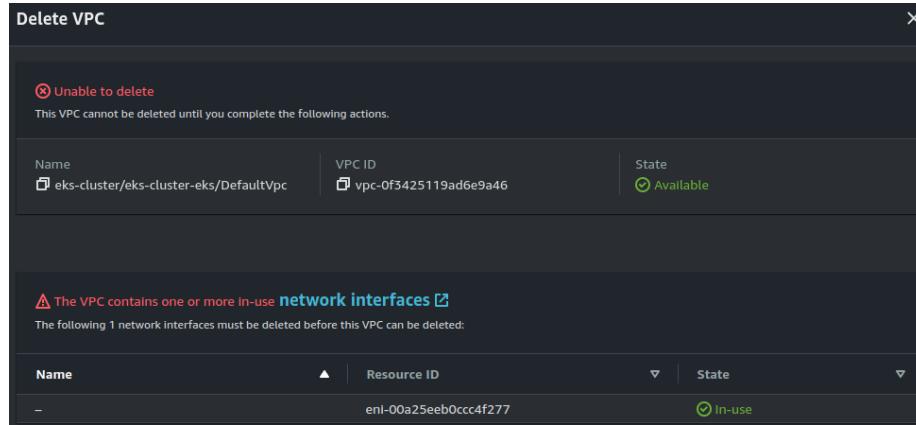


Figure 37: img_6.png

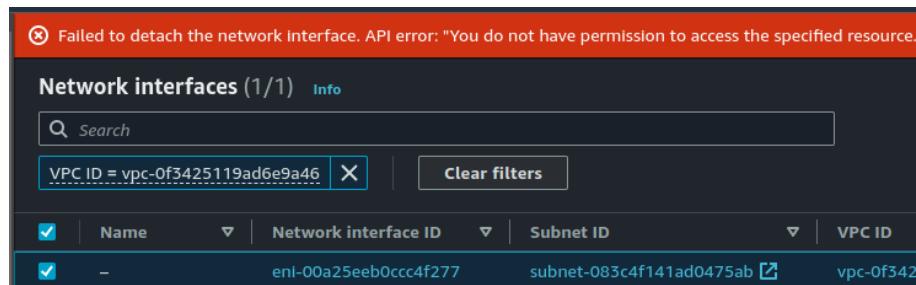


Figure 38: img_5.png

```
(.venv) [tmiiguez@lmiguel-pc cdk-eks-cluster]$ helm install prometheus prometheus-community/prometheus \
--namespace prometheus \
--set alertmanager.persistentVolume.storageClass="gp2" \
--set server.persistentVolume.storageClass="gp2"
Error: INSTALLATION FAILED: Kubernetes cluster unreachable: exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"
```

Figure 39: img_12.png

```
(.venv) [tmiiguez@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n prometheus -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
pod/prometheus-alertmanager-0   0/1     Pending   0          4m7s   <none>        <none>    <none>        <none>
pod/prometheus-kube-state-metrics-5fb6bbff78-5clpv   1/1     Running   0          4m7s   10.0.4.196   ip-10-0-38-75.us-west-2.compute.internal   <none>    <none>
pod/prometheus-prometheus-node-exporter-71lnx   1/1     Running   0          4m7s   10.0.186.26   ip-10-0-186-26.us-west-2.compute.internal   <none>    <none>
pod/prometheus-prometheus-node-exporter-8k2sv   1/1     Running   0          4m7s   10.0.123.212  ip-10-0-123-212.us-west-2.compute.internal   <none>    <none>
pod/prometheus-prometheus-node-exporter-hesph   1/1     Running   0          4m7s   10.0.30.75   ip-10-0-30-75.us-west-2.compute.internal   <none>    <none>
pod/prometheus-prometheus-pushgateway-7d5580d46-cqwp   1/1     Running   0          4m7s   10.0.105.219  ip-10-0-123-212.us-west-2.compute.internal   <none>    <none>
pod/prometheus-server-77df547d88-18rpn   0/2     Pending   0          4m7s   <none>        <none>    <none>        <none>
```

Figure 40: img_14.png

-
- No hay logs en `kubectl logs -n prometheus pod/prometheus-server-77df547d88-18rpn -c prometheus-server`.
- `kubectl describe -n prometheus pods/prometheus-server-77df547d88-bxtdc` no ayuda:

```
Volumes:
  config-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      prometheus-server
    Optional:  false
  storage-volume:
    Type:      PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName: prometheus-server
    ReadOnly:   false
  kube-api-access-l2pcb:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
  QoS Class:      BestEffort
  Node-Selectors: <none>
  Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type     Reason          Age     From           Message
  ----     -----          ---     ----          -----
  Warning  FailedScheduling 4m13s  default-scheduler  0/1 nodes are available: 1 Too many pods. preemption: 0/1
          nodes are available: 1 No preemption victims found for incoming pod.
```

Figure 41: img_17.png

- *
- `kubectl describe pvc -n prometheus` parece un problema de volúmenes. Al parecer no puede crear algun volumen.
- *
- Se instaló `aws-ebs-csi-driver`, ahora todos los pods en *Pending*.
- *
- Se siguió el siguiente post para habilitar el almacenamiento persistente en EKS
- Se encontró un problema al crear el ServiceAccount y realizar el despliegue. Solucionado al desinstalar `aws-ebs-csi-driver`, instalado previamente.
- *
- Se intentó la configuración del despliegue usando el add-on de EKS para el driver EBS CSI. Pero el pod de prueba de AWS se queda en *Pending*.
- *
- Se intentó la instalación del driver EBS CSI usando helm
- *
- Se volvió a reintentar, esta vez siquiendo este blog de AWS para usar el EBS CSI driver como un add-on de EKS

```

Name:      storage-prometheus-alertmanager-0
Namespace:  prometheus
StorageClass: gp2
Status:     Pending
Volume:
Labels:    app.kubernetes.io/instance=prometheus
           app.kubernetes.io/name=alertmanager
Annotations: <none>
Finalizers: [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode: Filesystem
Used By:    prometheus-alertmanager-0
Events:
  Type   Reason          Age            From               Message
  ----  ----             ---           ----              -----
  Normal  WaitForFirstConsumer  11m           persistentvolume-controller  waiting for first consumer to b
e created before binding
  Normal  WaitForPodScheduled  81s (x41 over 11m)  persistentvolume-controller  waiting for pod prometheus-aler
tmanager-0 to be scheduled
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$
```

Figure 42: img_18.png

```

(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pods -n prometheus
NAME                               READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0           0/1     Pending   0          4m39s
prometheus-kube-state-metrics-5fb6fbbf78-ss9nt   0/1     Pending   0          4m39s
prometheus-prometheus-node-exporter-jr8x9       0/1     Pending   0          4m39s
prometheus-prometheus-pushgateway-7d55869d46-vklx6  0/1     Pending   0          4m39s
prometheus-server-77df547d88-tbvdb        0/2     Pending   0          4m39s
```

Figure 43: img_19.png

```

eks-cluster: creating CloudFormation changeset...
1:02:08 PM | CREATE_FAILED      | AWS::EKS::Addon           | awsebscsidriver
1 validation error detected: Value 'Addon moved to failed status during Create operation.
Code: ConfigurationConflict, Message: Conflicts found when trying to apply. Will not continue due to resolve
conflicts mode. Conflicts:
PodDisruptionBudget.policy ebs-csi-controller - .metadata.labels.app.kubernetes.io/managed-by
```

Figure 44: img_20.png

```

(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
app                                0/1     Pending   0          6m2s
my-app-001-784777d896-4m8xh     1/1     Running   0          102m
my-app-001-784777d896-9nthn     1/1     Running   0          102m
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl delete -f dynamic-provisioning/
persistentvolumeclaim "ebs-claim" deleted
pod "app" deleted
storageclass.storage.k8s.io "ebs-sc" deleted
```

Figure 45: img_21.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-ebs-csi-driver,app.kubernetes.io/instance=aws-ebs-csi-driver"
NAME                               READY   STATUS    RESTARTS   AGE
ebs-csi-controller-587f5768b4-hfbzl  5/5     Running   0          4m32s
ebs-csi-controller-587f5768b4-rxxfr  0/5     Pending   0          4m32s
ebs-csi-node-wbwjl                 3/3     Running   0          4m33s
```

Figure 46: img_22.png

- * Repitiendo los pasos se encontró que el Service Account fue creado en el namespace `default` cuando debió ser creado en el namespace `kube-system`.
- * Así mismo se encontró algunas otras herramientas para diagnosticar los componentes del add-on:
 - `kubectl get deploy,ds -l=app.kubernetes.io/name=aws-ebs-csi-driver -n kube-system`
 - `kubectl get po -n kube-system -l 'app in (ebs-csi-controller,ebs-csi-node)'`
 - `kubectl get -n kube-system pod/ebs-csi-controller-CHANGE_ME -o jsonpath='{.spec.containers[*].name}'`
- * El problema persiste, pero ahora al hacer describe del PersistentVolumeClaim (PVC) obtenemos varios errores.

```
Warning ProvisioningFailed 10m ebs.csi.aws.com_ebs-csi-controller-6cbc456c69-hk5g4_931cdd83-7f6f-4045-b922-16c455234ed4 failed to provision volume with StorageClass "gp3": rpc error: code = Internal desc = Could not create volume "pvc-52f74507-8607-49a1-ac97-4ccfdb23eee3": could not create volume in EC2: WebIdentityErr: failed to retrieve credentials
caused by: AccessDenied: Not authorized to perform sts:AssumeRoleWithWebIdentity
    status code: 403, request id: 1d1ffaa6-981d-41da-bcd0-9866975ac09d
Normal Provisioning 3m58s (x11 over 13m) ebs.csi.aws.com_ebs-csi-controller-6cbc456c69-hk5g4_931cdd83-7f6f-4045-b922-16c455234ed4 External provisioner is provisioning volume for claim "default/pvc-csi"
Warning ProvisioningFailed 3m58s (x2 over 8m58s) ebs.csi.aws.com_ebs-csi-controller-6cbc456c69-hk5g4_931cdd83-7f6f-4045-b922-16c455234ed4 (combined from similar events): failed to provision volume with StorageClass "gp3": rpc error: code = Internal desc = Could not create volume "pvc-52f74507-8607-49a1-ac97-4ccfdb23eee3": could not create volume in EC2: WebIdentityErr: failed to retrieve credentials
caused by: AccessDenied: Not authorized to perform sts:AssumeRoleWithWebIdentity
    status code: 403, request id: e5c64678-904a-4de9-9359-536c9a927dc1
Normal ExternalProvisioning 3m56s (x40 over 13m) persistentvolume-controller waiting for a volume to be created, either by external provisioner "ebs.csi.aws.com" or manually created by system administrator
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ 
```

Figure 47: img_24.png

- * Redesplegando el stack, para acelerar las cosas se puede usar otra región para el despliegue, y no esperar a que el cluster se elimine por completo.
 - * Problema persiste.
- Creando el cluster con kubectl según el blog, el PVC llega a estado BOUND, y el pod a RUNNING! El problema debe estar en la forma en cómo CDK crea el cluster EKS o algún policy o recurso fallido.
 - * Se crea el storageClass “gp3”.
 - * Se reintenta el comando usando “gp3” como storageClass. Al-

gunos recursos funcionan y otros ya no.

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n prometheus
NAME                                         READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0                0/1     Pending   0          4m24s
pod/prometheus-kube-state-metrics-6dc44cc4d9-kv7m8 1/1     Running   0          4m24s
pod/prometheus-prometheus-node-exporter-b4kl4   1/1     Running   0          4m24s
pod/prometheus-prometheus-pushgateway-5fdcccb6f7-8xj5b 0/1     Pending   0          4m24s
pod/prometheus-server-76c879bccf-v44sm        2/2     Running   0          4m24s

NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/prometheus-alertmanager      ClusterIP   10.100.63.205 <none>        9093/TCP    4m24s
service/prometheus-alertmanager-headless ClusterIP   None           <none>        9093/TCP    4m24s
service/prometheus-kube-state-metrics   ClusterIP   10.100.132.135 <none>        8080/TCP    4m24s
service/prometheus-prometheus-node-exporter ClusterIP   10.100.159.97  <none>        9100/TCP    4m24s
service/prometheus-prometheus-pushgateway ClusterIP   10.100.169.133 <none>        9091/TCP    4m24s
service/prometheus-server             ClusterIP   10.100.227.81  <none>        80/TCP      4m24s

NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE(S)
ELECTOR   AGE
daemonset.apps/prometheus-prometheus-node-exporter 1         1         1         1           1           <none>
                                                    4m24s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics 1/1     1           1           4m25s
deployment.apps/prometheus-prometheus-pushgateway 0/1     1           0           4m25s
deployment.apps/prometheus-server               1/1     1           1           4m25s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-6dc44cc4d9 1         1         1         4m25s
replicaset.apps/prometheus-prometheus-pushgateway-5fdcccb6f7 1         1         0         4m25s
replicaset.apps/prometheus-server-76c879bccf            1         1         1         4m25s

NAME           READY   AGE
statefulset.apps/prometheus-alertmanager 0/1     4m25s
```

Figure 48: img_27.png

- * No se puede obtener mayor detalle de porqué los pods fallaron.
- Revisitando el problema y leyendo detenidamente estas instrucciones:

When the plugin is deployed, it creates and is configured to use a service account that's named `ebs-csi-controller-sa`. The service account is bound to a Kubernetes clusterrole that's assigned the required Kubernetes permissions.

Figure 49: img_33.png

- *
- *
- * Nuestro error fue ignorar que el service account `ebs-csi-controller-sa` ya existía y por lo tanto **NO** debía ser creado por nosotros. Sólo necesitábamos *anotar* el `ebs-csi-controller-sa` con el rol creado.
- * Ejecutando las instrucciones y desplegando la aplicación de prueba, el PVC llega a estado BOUND y el pod a RUNNING.

Create an IAM role and attach the required AWS managed policy with the following command. Replace `my-cluster` with the name of your cluster. The command deploys an AWS CloudFormation stack that creates an IAM role, attaches the IAM policy to it, and annotates the `existing` `ebs-csi-controller-sa` service account with the Amazon Resource Name (ARN) of the IAM role. If your cluster is in the AWS GovCloud (US-East) or

Figure 50: img_34.png

```
# https://docs.aws.amazon.com/eks/latest/userguide/csi-iam-role.html
# it will create a policy/role and *annotate* the **existing** ebs-csi-controller-sa service account
# it will NOT create NOR update the ebs-csi-controller-sa (it already exists!)
eksctl create iamserviceaccount \
--name ebs-csi-controller-sa \
--namespace kube-system \
--cluster eks-cluster-eks \
--attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
--approve \
--role-only \
--role-name AmazonEKS_EBS_CSI_DriverRole
# https://docs.aws.amazon.com/eks/latest/userguide/managing-ebs-csi.html
# create the addon and attach the role created
eksctl create addon \
--name aws-ebs-csi-driver \
--cluster eks-cluster-eks \
--service-account-role-arn arn:aws:iam::136737878111:role/AmazonEKS_EBS_CSI_DriverRole --force
```

Figure 51: img_35.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl apply -f gp3-sc.yaml
storageclass.storage.k8s.io/gp3 created
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl apply -f pvc-csi.yaml
persistentvolumeclaim/pvc-csi created
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl apply -f pod-csi.yaml
pod/app-gp3 created
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pod --watch
NAME           READY   STATUS        RESTARTS   AGE
app-gp3        0/1     ContainerCreating   0          11s
my-app-001-784777d896-mxzh6   1/1     Running      0          19m
my-app-001-784777d896-zdn9n   1/1     Running      0          19m
app-gp3        1/1     Running      0          15s
^C(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pvc
NAME      STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pvc-csi   Bound     pvc-a5e02a5b-64d8-4225-a3ca-dfac0cf63a35   1Gi        RWO          gp3          39s
```

Figure 52: img_36.png

- Problemas al instalar Grafana.
 - Pod se queda en estado Pending. EBS CSI driver está instalado.

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n grafana
NAME           READY   STATUS    RESTARTS   AGE
pod/grafana-6959759d4-rtkv2   0/1     Pending   0          9m6s

NAME            TYPE        CLUSTER-IP      EXTERNAL-IP
              PORT(S)     AGE
service/grafana LoadBalancer   172.20.35.43   ad976b8f002dd43749003be614d1e3b1-250774179.us-east-1.elb.amazonaws.com   80:31293/TCP   9m7s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana   0/1     1           0          9m7s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-6959759d4   1         1         0         9m7s
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$
```

Figure 53: img_42.png

-
- kubectl get ev -n grafana
-
- kubectl get pvc -n grafana
-
- Al ver los eventos creí que era el LB que reiniciaba el pod, al no estar healthy
- Seguí estas instrucciones para desplegarlo “por partes” (sin LB)

<https://grafana.com/docs/agent/latest/operator/helm-getting-started/>
- El problema persiste:

*

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get ev -n grafana
LAST SEEN   TYPE      REASON          OBJECT                               MESSAGE
14m         Warning   FailedScheduling   pod/grafana-5fdbdb5dc5-m7j8l   0/1 nodes are available: 1 Too many
           pods. preemption: 0/1 nodes are available: 1 No preemption victims found for incoming pod.
13m         Warning   FailedScheduling   pod/grafana-5fdbdb5dc5-m7j8l   skip schedule deleting pod: grafana
/grafana-5fdbdb5dc5-m7j8l
25m         Normal    SuccessfulCreate  replicaset/grafana-5fdbdb5dc5   Created pod: grafana-5fdbdb5dc5-m7j
8l
27m         Warning   FailedScheduling   pod/grafana-6959759d4-4h2tw   0/1 nodes are available: 1 Too many
           pods. preemption: 0/1 nodes are available: 1 No preemption victims found for incoming pod.
4m58s       Warning   FailedScheduling   pod/grafana-6959759d4-rtkv2   0/1 nodes are available: 1 Too many
           pods. preemption: 0/1 nodes are available: 1 No preemption victims found for incoming pod.
12m         Warning   FailedScheduling   pod/grafana-6959759d4-x2cx4   0/1 nodes are available: 1 Too many
           pods. preemption: 0/1 nodes are available: 1 No preemption victims found for incoming pod.
10m         Warning   FailedScheduling   pod/grafana-6959759d4-x2cx4   skip schedule deleting pod: grafana
/grafana-6959759d4-x2cx4
32m         Normal    SuccessfulCreate  replicaset/grafana-6959759d4   Created pod: grafana-6959759d4-4h2t
w
12m         Normal    SuccessfulCreate  replicaset/grafana-6959759d4   Created pod: grafana-6959759d4-x2cx
4
4m58s       Normal    SuccessfulCreate  replicaset/grafana-6959759d4   Created pod: grafana-6959759d4-rtkv
2
9m41s       Warning   FailedScheduling   pod/grafana-7859566f74-r54pl   0/1 nodes are available: 1 Too many
           pods. preemption: 0/1 nodes are available: 1 No preemption victims found for incoming pod.
5m51s       Warning   FailedScheduling   pod/grafana-7859566f74-r54pl   skip schedule deleting pod: grafana
/grafana-7859566f74-r54pl
9m41s       Normal    SuccessfulCreate  replicaset/grafana-7859566f74   Created pod: grafana-7859566f74-r54
pl
32m         Normal    WaitForFirstConsumer persistentvolumeclaim/grafana  waiting for first consumer to be cr
eated before binding
32m         Normal    EnsuringLoadBalancer service/grafana                Ensuring load balancer
32m         Normal    ScalingReplicaSet   deployment/grafana              Scaled up replica set grafana-69597
59d4 to 1
```

Figure 54: img_43.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get pvc -n grafana
NAME      STATUS  VOLUME   CAPACITY  ACCESS MODES  STORAGECLASS  AGE
grafana  Pending           gp2        8m10s
```

Figure 55: img_44.png

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n grafana
NAME                  READY   STATUS    RESTARTS   AGE
pod/grafana-5f7f6d4d8c-mrvd6   1/1     Running   0          16m

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana  1/1     1           1          16m

NAME                  DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-5f7f6d4d8c  1        1        1       16m
```

Figure 56: img_46.png

- Al revisar el manifiesto yml vi que requiere 700MiB de memoria. Eliminé los despliegues de prometheus y volví a ejecutar el despliegue:

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n grafana
NAME                         READY   STATUS    RESTARTS   AGE
pod/grafana-5f7f6d4d8c-mrvd6  1/1     Running   0          16m

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana 1/1     1           1           16m

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana 1         1         1       16m
```

Figure 57: img_45.png

- *
 - Eliminé el despliegue por partes y ejecuté el despliegue usando helm y el pod está en Running!

```
(.venv) [lmiguel@lmiguel-pc cdk-eks-cluster]$ kubectl get all -n grafana
NAME                         READY   STATUS    RESTARTS   AGE
pod/grafana-b45b4c6c7-f25bn  1/1     Running   0          17s

NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/grafana  ClusterIP  172.20.42.163  <none>        80/TCP      18s

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana 1/1     1           1           18s

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-b45b4c6c7  1         1         1       18s
```

Figure 58: img_47.png

- *
 - Redesplegué con nodos de mayor tamaño (8GB de memoria).

Instances (2) Info					
<input type="text"/> Find Instance by attribute or tag (case-sensitive)					
	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	eks-cluster/eks-cluster-eks-cluster-...	i-08c8e00fec5611b52	Running Q Q	t3.large	Initializing
<input type="checkbox"/>	eks-cluster/eks-cluster-eks-cluster-...	i-0eed2a6cdeb960e37	Shutting-down Q Q	t3.medium	-

Figure 59: img_48.png

*