

Kubernetes inicio

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

Vamos a crear un Cluster en AWS. Vamos a AWS > EKS ★ > Create Cluster.

MUY IMPORTANTE desactivar el EKS Auto Mode, porque si está habilitado nos dará muchos problemas de permisos.

Configure cluster

Configuration options - *new* [Info](#)

Choose how you would like to configure the cluster.

☐ Quick configuration (with EKS Auto Mode) - *new*

Quickly create a cluster with production-grade default settings. The configuration uses EKS Auto Mode to automate infrastructure tasks like creating nodes and provisioning storage.

☒ Custom configuration

To change default settings prior to creation, choose this option. This configuration gives the option to use EKS Auto Mode and customize the cluster's configuration.

EKS Auto Mode - *new* [Info](#)

Choose if you would like to use EKS's Auto Mode.

☐ Use EKS Auto Mode

EKS automates routine cluster tasks for compute, storage, and networking. When a new pod can't fit onto existing nodes, EKS creates a new node. EKS combines cluster infrastructure managed by AWS with integrated Kubernetes capabilities to meet application compute needs. [View pricing](#)

► Included capabilities

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

Cluster configuration [Info](#)

Name

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

demo-asix2b

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

Cluster IAM role [Info](#)

Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. [EKS User Guide](#)

LabRole

Kubernetes version settings

Kubernetes version [Info](#)

Select Kubernetes version for this cluster.

1.31

Upgrade policy [Info](#)

Choose one of the following options. You can switch the setting later while the standard support period is in effect.

☒ Standard

This option supports the Kubernetes version for 14 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto upgraded to the next version.

☐ Extended


This option supports the Kubernetes version for 26 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto upgraded to the next version.

Auto Mode Compute - new [Info](#)

Configure node management for your EKS cluster. EKS offers four compute options: EKS Auto Mode, EC2 Managed Node Groups, Fargate, and hybrid nodes. Node groups, Fargate profiles, and hybrid nodes are configured after cluster creation. You can also create self-managed nodes.

Compute configuration

If EKS Auto Mode is not managing compute resources, you need to create compute resources once the cluster is ready. We recommend creating a node group after cluster creation.

[View documentation](#) 

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

Specify networking

Networking [Info](#)

IP address family and service IP address range cannot be changed after cluster creation.

VPC [Info](#)

Select a VPC to use for your EKS cluster resources.

vpc-037977ddab44e0c40 | Default




Subnets [Info](#)

Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet

Select subnets



subnet-00d8add2742e1650e 
us-east-1b 172.31.80.0/20

subnet-0550e4976da5198b4 
us-east-1a 172.31.0.0/20

subnet-02c35ff1924530459 
us-east-1c 172.31.16.0/20

Additional security groups - optional [Info](#)

EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane. Optionally, choose additional interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups




Additional security groups - optional [Info](#)

EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane. Optionally, choose additional interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups



sg-0807496b89cb06344 | default 
default VPC security group

Choose cluster IP address family [Info](#)

Specify the IP address type for pods and services in your cluster.

Queremos añadir 5 pods, que son:

kube-proxy


Amazon VPC CNI

CoreDNS


Amazon EKS Pod Identity Agent

Amazon EFS CSI Driver


El resto aparte de éstos, desmárcalo.

☐


Node monitoring agent [Info](#)
Enable automatic detection of node health issues.
Category
observability
Compatible compute
EC2

☒


kube-proxy [Info](#)
Enable service networking within your cluster.
Category
networking
Compatible compute
EC2, Hybrid Nodes

☒


Amazon VPC CNI [Info](#)
Enable pod networking within your cluster.
Category
networking
Compatible compute
EC2

☒


CoreDNS [Info](#)
Enable service discovery within your cluster.
Category
networking
Compatible compute
EC2, Hybrid Nodes, Fargate, EKS Auto Mode

☒

Amazon EKS Pod Identity Agent [Info](#)
Install EKS Pod Identity Agent to use EKS Pod Identity to grant AWS IAM permissions to pods through Kubernetes service accounts.
Category
security
Compatible compute
EC2, Hybrid Nodes

☐

Amazon SageMaker HyperPod task governance
Prioritize tasks, allocate compute resources, and maximize utilization.
Category
policy-management
Compatible compute
HyperPod

☒

Amazon EFS CSI Driver [Info](#)
Enable Amazon Elastic File System (EFS) within your cluster.
Category
storage
Compatible compute
EC2, EKS Auto Mode

[santos-pardos \(profesantos\)](#)

GITHUB

[Install and Set Up kubectl on Linux | Kubernetes](#)

Instalar cliente

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

Instalar la última versión con este comando.

```

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

```

```

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

```

Tutorials

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

Install AWS Tools

```

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

```

🔗 Install KUBECTL

```

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

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

```

Install EKSCTL

```

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

```

```

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

```

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

```
ubuntu@ip-172-31-90-175:~$ cat -n /home/ubuntu/.kube/config
1  apiVersion: v1
2  clusters:
3  - cluster:
4    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJVU5USTVYczFpU
l3RFFZSktvWkloZWVFFFEJQXQdGVEVU0tJFR0ExVUUKQXhNS2EzVmlaWep1WlhbG6N0QWVGdzB5TlRBeU1qZ3hPRFF4TURkYUJ3MHpOVEFSTWpZeE9
UTJNRGRhTUJVeApFekFSQmdOVkxJBTVRDbXQxwW1wewJTVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCKFvSUJBUNkei8vRz
g5OHJyZ3lzaWRjSmNES3pTVlVHTmxGSmRHK00rK2NmNVJDU2hUQ3E0YmVrNnRtOWduMVYKNzAyTXJnTFFNWDdqE3c1NqQjQ0NXE2aENvcXF5WEdubmMw
NDNDd0gwN1lBdkNKV1FY1J3YVYwbjNKTlVCDQpSRnZadWJqZjUrdnUrNTFXRytTVA1Tmk5Q1pneFFkdWcxMkJOwFBNL2c1eDJsZl4Y2FJWk1BMnRWLz
NrR1NncklyNEN0Y2lVaEcZTHZQK1FHY1M0eVBOZmVUNHQ1Rm1jSzdKZk10NUFKMERIRVpPDERzS0RWNGdVQmptMVpqS1YKSW5vOVJnQlJEaHF5L3pjc2Uv
eXlzdjJXew1xUdQtQ2FORWFJbjRvY1BNbm5hYVZza3h0a1NlN3NVt081REhoOQ05Nm13Vm9CYjNRY3N1TGoybTVRUlo3cElVYkdqQWdNqkFBR2pxVEJYU
E0R0ExVWREd0VCL3dRRUF3SUNwREFQckJnTlZlUk1CQWY4RUJUQUJBUUgVUgVUIwR0ExVWREZ1FXQk1JZlR3bWVHM1ZabXJBNG4zQVB2WS9POU5DdWpEQVYK
QmdOVkhSRUVEakFNZ2dwcMRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDU3FHU0lRQkFRQXM1OFFkVfDIUgPSRExR25vWStRMDNqeU1vK29EZVRqT0
9aUUVtZjJlQmF0c2NselVlaXUxZTdzblE5RW1vQTJ3VENwYXcyUWlCCKVHU1hXN3N1N2JFb25sT3hra0pQW9LdnZl0cVQusyN1dTV2FxdnJhUk96VEF
VjUvVjlnVnk9RbmIyUGZzNEEKNi9naUx5STdDQnBhK1d1ckw3aHFMVlSYzA4VFEzC2VRUKIwVWt3T1lXblRzYXNNVWwXajN1cGV6ZmkwbDR2Ugo0MmMdiS
p0VS9iNnR4MDV2czAweERoVVRmU0xIRWIwREFOTndkZFJ0NFU4Z1M1cXdRRXRZUTFCSDdHekxLNkdsCKFyVGRAN3ZwYzNSc2ZSR3JaSU54aG5kd0LDRmZ
SXF3RjIyT1BBUGJXDUJUVXR1ckwyMXF1UVB1WTAvdUtdZE8KaUJBCzI1MENJMMRhCi0tLS0tRU5EIEENFURJRKldQVRFLS0tLS0K
5    server: https://2816889824995FF3F866728E9973236F.gr7.us-east-1.eks.amazonaws.com
6    name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
7  contexts:
8  - context:
9    cluster: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
10   user: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
11   name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
12   current-context: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
13   kind: Config
14   preferences: {}
15   users:
16   - name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
17     user:
18       exec:
19         apiVersion: client.authentication.k8s.io/v1beta1
20         args:
21         - --region
22         - us-east-1
23         - eks
24         - get-token
25         - cluster-name
```

Via externa para conectarme al cluster:

```
ubuntu@ip-172-31-90-175:~$ cat -n /home/ubuntu/.kube/config
1  apiVersion: v1
2  clusters:
3  - cluster:
4    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJVU5USTVYczFpU
l3RFFZSktvWkloZWVFFFEJQXQdGVEVU0tJFR0ExVUUKQXhNS2EzVmlaWep1WlhbG6N0QWVGdzB5TlRBeU1qZ3hPRFF4TURkYUJ3MHpOVEFSTWpZeE9
UTJNRGRhTUJVeApFekFSQmdOVkxJBTVRDbXQxwW1wewJTVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCKFvSUJBUNkei8vRz
g5OHJyZ3lzaWRjSmNES3pTVlVHTmxGSmRHK00rK2NmNVJDU2hUQ3E0YmVrNnRtOWduMVYKNzAyTXJnTFFNWDdqE3c1NqQjQ0NXE2aENvcXF5WEdubmMw
NDNDd0gwN1lBdkNKV1FY1J3YVYwbjNKTlVCDQpSRnZadWJqZjUrdnUrNTFXRytTVA1Tmk5Q1pneFFkdWcxMkJOwFBNL2c1eDJsZl4Y2FJWk1BMnRWLz
NrR1NncklyNEN0Y2lVaEcZTHZQK1FHY1M0eVBOZmVUNHQ1Rm1jSzdKZk10NUFKMERIRVpPDERzS0RWNGdVQmptMVpqS1YKSW5vOVJnQlJEaHF5L3pjc2Uv
eXlzdjJXew1xUdQtQ2FORWFJbjRvY1BNbm5hYVZza3h0a1NlN3NVt081REhoOQ05Nm13Vm9CYjNRY3N1TGoybTVRUlo3cElVYkdqQWdNqkFBR2pxVEJYU
E0R0ExVWREd0VCL3dRRUF3SUNwREFQckJnTlZlUk1CQWY4RUJUQUJBUUgVUgVUIwR0ExVWREZ1FXQk1JZlR3bWVHM1ZabXJBNG4zQVB2WS9POU5DdWpEQVYK
QmdOVkhSRUVEakFNZ2dwcMRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDU3FHU0lRQkFRQXM1OFFkVfDIUgPSRExR25vWStRMDNqeU1vK29EZVRqT0
9aUUVtZjJlQmF0c2NselVlaXUxZTdzblE5RW1vQTJ3VENwYXcyUWlCCKVHU1hXN3N1N2JFb25sT3hra0pQW9LdnZl0cVQusyN1dTV2FxdnJhUk96VEF
VjUvVjlnVnk9RbmIyUGZzNEEKNi9naUx5STdDQnBhK1d1ckw3aHFMVlSYzA4VFEzC2VRUKIwVWt3T1lXblRzYXNNVWwXajN1cGV6ZmkwbDR2Ugo0MmMdiS
p0VS9iNnR4MDV2czAweERoVVRmU0xIRWIwREFOTndkZFJ0NFU4Z1M1cXdRRXRZUTFCSDdHekxLNkdsCKFyVGRAN3ZwYzNSc2ZSR3JaSU54aG5kd0LDRmZ
SXF3RjIyT1BBUGJXDUJUVXR1ckwyMXF1UVB1WTAvdUtdZE8KaUJBCzI1MENJMMRhCi0tLS0tRU5EIEENFURJRKldQVRFLS0tLS0K
5    server: https://2816889824995FF3F866728E9973236F.gr7.us-east-1.eks.amazonaws.com
6    name: arn:aws:eks:us-east-1:731219410996:cluster/demo-asix2b
```

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

EC2 > Security Groups > sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005

EC2

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005

Details

Security group name

eks-cluster-sg-demo-asix2b-1365656005

Security group ID

sg-067b48b9f29679736

Owner

731219410996

Inbound rules count

1 Permission entry

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (1)

Search

	Name	Security group rule ID	IP version	Type
	-	sgr-049b515e989ac1507	-	All traffic

EC2 > Security Groups > sg-067b48b9f29679736 - eks-cluster-sg-demo-asix2b-1365656005 > Edit inbound rules

Edit inbound rules

Custom ▼

Q

CIDR blocks

0.0.0.0/0

0.0.0.0/8

0.0.0.0/16

0.0.0.0/24

0.0.0.0/32

::/0

::/16

::/32

::/48

::/64

Security Groups

launch-wizard-1 | sg-0d5513764e798d822

eks-cluster-sg-demo-asix2b-1365656005 | sg-067b48b9f29679736

eks-cluster-sg-demo-asix2b-1365656005

default | sg-0807496b89cb06344

Prefix lists

com.amazonaws.us-east-1.dynamodb | pl-02cd2c6b

Edit inbound rules [Info](#)

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

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info
sgr-049b515e989ac1507	All traffic ▼	All	All	Custom ▼
-	All traffic ▼	All	All	Custom ▼

Add rule

Edit inbound rules [Info](#)

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

Inbound rules [Info](#)

Security group rule ID

sg-r-049b515e989ac1507

Type [Info](#)

All traffic

Protocol [Info](#)

All

Port range [Info](#)

All

Source [Info](#)

Custom

Q

sg-067b48b9f29679736

Description

-

All traffic

All

All

Custom

Q sg-0d5513764e798d822

sg-0d5513764e798d822

Add rule



[EC2](#) > [Instances](#) > i-0536f2e2fbb37a297

EC2

Dashboard
EC2 Global View
Events

Instances

Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Images

IMDSv2
Required

Operator

-

Instance A
 arn:aws

Details

Status and alarms

Monitoring

Security

Networking

▼ Security details

IAM Role

-

Owner ID
 73121


Security groups

sg-0d5513764e798d822 (launch-wizard-1)

▼ Inbound rules

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


demo-asix2b

 End of standard support for Kubernetes version 1.31 is November 26, 2025. On that date, your cluster will be deprecated. For more information, see the [Kubernetes 1.31 deprecation page](#).

▼ Cluster info [Info](#)

Status

 Active

Kubernetes version [Info](#)

1.31

Cluster health issues

 0

Upgrade insights

 3  1

[Overview](#)

[Resources](#)

[Compute](#)

[Networking](#)

[Add-ons](#)

1

[Access](#)

Nodes (0) [Info](#)

Compute no tengo nada

Queremos máquina virtuales donde están

Configure node group [Info](#)

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

Node group configuration

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

Name

Assign a unique name for this node group.

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

Node IAM role [Info](#)

Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).

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

[Learn more](#)

Launch template [Info](#)

- Step 1
- Configure node group**
- Step 2
- Set compute and scaling configuration
- Step 3
- Specify networking
- Step 4
- Review and create

Configure node group [Info](#)

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

Node group configuration

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

Name

Assign a unique name for this node group.

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

Node IAM role [Info](#)

Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).

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

[Learn more](#)

Launch template [Info](#)

- Step 1
- Configure node group
- Set compute and scaling configuration**
- Step 3
- Specify networking
- Step 4
- Review and create

Set compute and scaling configuration

Node group compute configuration

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

AMI type [Info](#)

Select the EKS-optimized Amazon Machine Image for nodes.

Capacity type

Select the capacity purchase option for this node group.

Instance types [Info](#)

Select instance types you prefer for this node group.

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

Disk size

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

Node group scaling configuration

Desired size

Set the desired number of nodes that the group should launch with initially.

1 nodes

Desired node size must be greater than or equal to 0

Minimum size

Set the minimum number of nodes that the group can scale in to.

1 nodes

Minimum node size must be greater than or equal to 0

Maximum size

Set the maximum number of nodes that the group can scale out to.

2 nodes

Maximum node size must be greater than or equal to 1 and cannot be lower

- Step 1
Configure node group
- Step 2
Set compute and scaling configuration
- Step 3
Specify networking
- Step 4
Review and create

Specify networking

Node group network configuration

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

Subnets info

Specify the subnets in your VPC where your nodes will run. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets

subnet-02c35f1924530459 ☐ subnet-0550e4976da5198b4 ☐ subnet-00d8add2743e1650e ☐
us-east-1c 172.31.16.0/20 us-east-1a 172.31.0.0/20 us-east-1b 172.31.80.0/20



Clear selected subnets

☒ Configure remote access to nodes [info](#)

next

CREATEEE

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

Ya tenemos una ec2 donde podemos desplegar los contenedores.

Comprobaciones

<https://github.com/santos-pardos/K8s-Eks.git>

git clone

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

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

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

Que hago para desplegar o poenr en marcha?

`kubectl apply -f helloworld-deployment.yaml`

Pero antes:kubectl get pods

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

No hay contenedores.

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
kube-system    aws-node-xzlcx                         2/2     Running   0           8m54s
kube-system    coredns-789f8477df-8799p              1/1     Running   0           35m
kube-system    coredns-789f8477df-wdqgn              1/1     Running   0           35m
kube-system    ebs-csi-controller-84fd6498d4-bp2qk   6/6     Running   0           35m
kube-system    ebs-csi-controller-84fd6498d4-fqh99   6/6     Running   0           35m
kube-system    ebs-csi-node-846vt                    3/3     Running   0           8m54s
kube-system    eks-pod-identity-agent-m7tsp          1/1     Running   0           8m54s
kube-system    kube-proxy-mxgxc                      1/1     Running   0           8m54s
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

Estos son los plugins que hemos seleccionado antes.

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

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

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

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

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

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

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

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

Load balancers (1) Actions Create load balancer

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input type="checkbox"/>	acd0bddf4181e45ebb0...	acd0bddf4181e45ebb0ed8...	-	vpc-037977ddab44e0c40	6 Availability Zones	classic	February 28, 2025, 20:29 (UTC+01:00)

EC2 > Load balancers

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Load balancers (1)

Elastic Load Balancing scales your lo

☐

Name

☐

[acd0bddf4181e45ebb0...](#)

O tambien

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
```

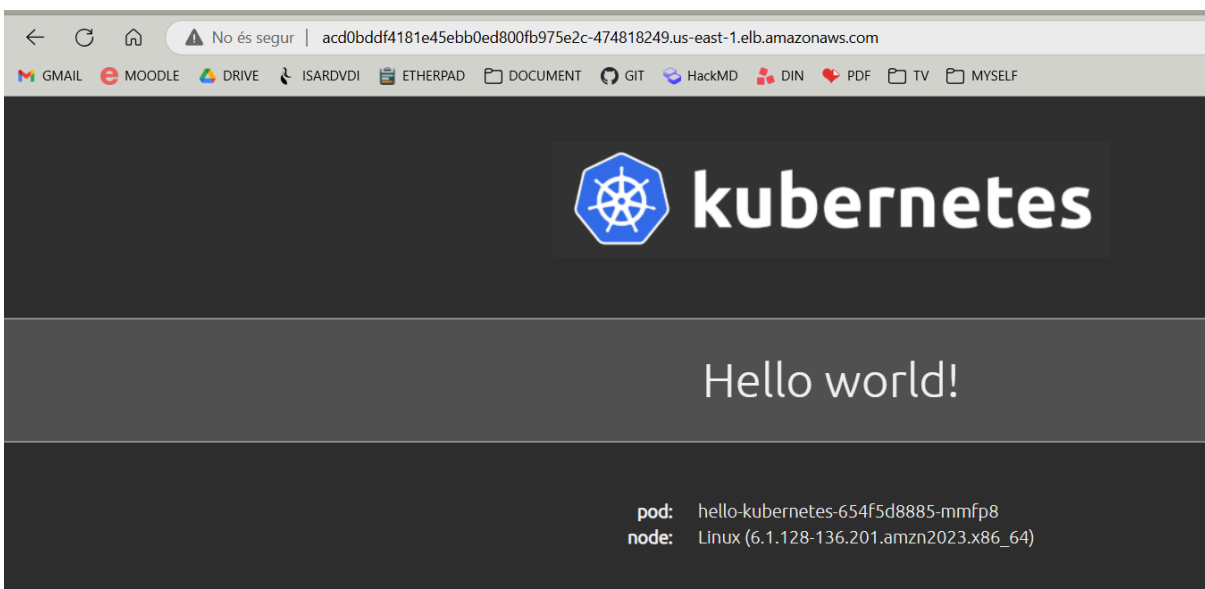
NAME	READY	STATUS	RESTARTS	AGE
pod/hello-kubernetes-654f5d8885-9xwmh	1/1	Running	0	5m39s
pod/hello-kubernetes-654f5d8885-mmfp8	1/1	Running	0	5m39s
pod/hello-kubernetes-654f5d8885-vhm4t	1/1	Running	0	5m39s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/hello-kubernetes	LoadBalancer	10.100.231.37	acd0bddf4181e45ebb0ed800fb975e2c-474818249.us-east-1.elb.amazonaws.com
service/kubernetes	ClusterIP	10.100.0.1	<none>

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

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/hello-kubernetes-654f5d8885	3	3	3	5m39s

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```



Objetivo cambiar esto y que aparezca el pupito

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

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl delete -f helloworld-svc.yaml
service "hello-kubernetes" deleted
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl delete -f helloworld-deployment.yaml
deployment.apps "hello-kubernetes" deleted
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	54m

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$
```

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

```

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

```

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

```

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

```

Lo mismo con el yaml del servicio LoadBalancer.


```

---
apiVersion: v1
kind: Service
metadata:
  name: hello-kubernetes
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 9898
  selector:
    app: hello-kubernetes

```

```

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

```

```

ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all

```

NAME	READY	STATUS	RESTARTS	AGE
pod/hello-kubernetes-d98f94888-d9l6z	0/1	ImagePullBackOff	0	72s
pod/hello-kubernetes-d98f94888-qbfxr	0/1	ImagePullBackOff	0	72s
pod/hello-kubernetes-d98f94888-rjpxf	0/1	ImagePullBackOff	0	72s
pod/podinfo-849bfb5c8d-6grjj	1/1	Running	0	3m53s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/hello-kubernetes	LoadBalancer	10.100.23.187	ad5508ce8339a4879803479f130f68fd-1771309034.us-east-1.elb.amazonaws.com
service/kubernetes	ClusterIP	10.100.0.1	<none>

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

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/hello-kubernetes-d98f94888	3	3	0	72s
replicaset.apps/podinfo-849bfb5c8d	1	1	1	3m53s

```

ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$

```

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

NAME	READY	STATUS	RESTARTS	AGE
pod/hello-kubernetes-74594654bb-lsxt4	1/1	Running	0	10s
pod/hello-kubernetes-74594654bb-p7lnt	1/1	Running	0	10s
pod/hello-kubernetes-74594654bb-stpqj	1/1	Running	0	10s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/hello-kubernetes	LoadBalancer	10.100.236.74	aeb15947d37384636a0af39e589882d3-1817738793.us-east-1.elb.amazonaws.com
service/kubernetes	ClusterIP	10.100.0.1	<none>

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

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/hello-kubernetes-74594654bb	3	3	3	10s

```
ubuntu@ip-172-31-90-175:~/K8s-Eks/HelloWorld$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/hello-kubernetes-74594654bb-lsxt4	1/1	Running	0	46s
pod/hello-kubernetes-74594654bb-p7lnt	1/1	Running	0	46s
pod/hello-kubernetes-74594654bb-stpqj	1/1	Running	0	46s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/hello-kubernetes	LoadBalancer	10.100.236.74	aeb15947d37384636a0af39e589882d3-1817738793.us-east-1.elb.amazonaws.com
service/kubernetes	ClusterIP	10.100.0.1	<none>

