

Infraestructura III

Instalación de Kubernetes

Objetivo

En el siguiente ejercicio vamos a instalar Kubernetes sobre una instancia de EC2 en AWS.

Consigna

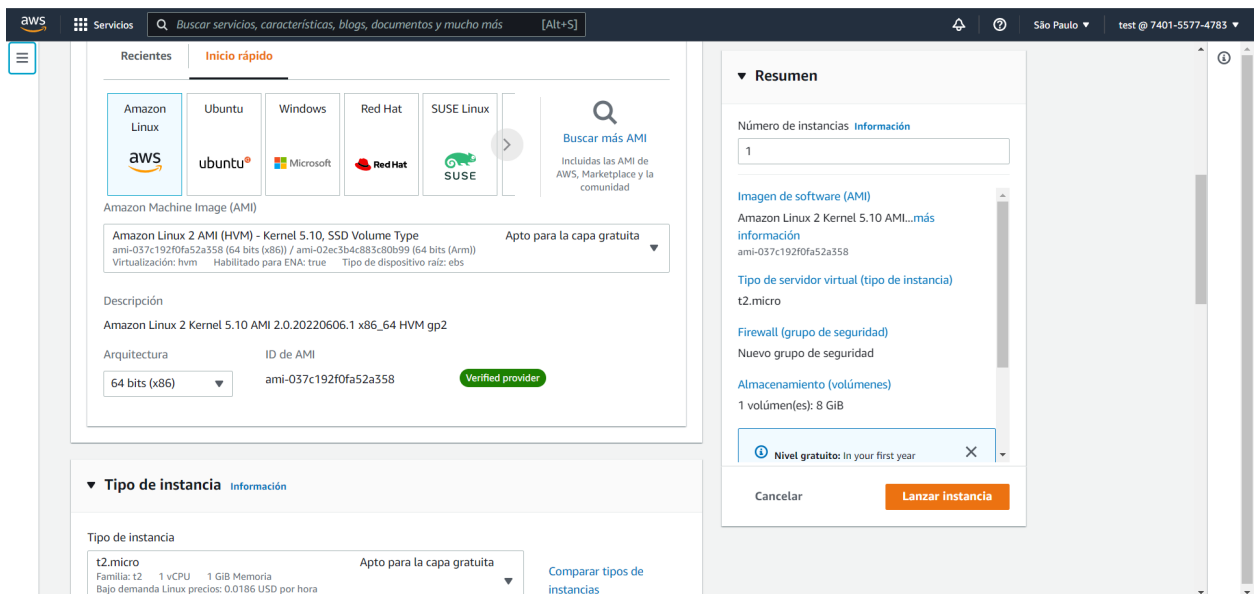
Veamos los pasos a seguir para realizar la instalación.

Instancia

Crear una instancia tipo Amazon Linux y seleccionar **t2.medium**:

D ■

1



The screenshot shows the AWS Management Console interface for creating a new EC2 instance. The 'Amazon Linux 2 AMI' is selected as the image, and 't2.micro' is selected as the instance type. The 'Lanzar instancia' button is visible at the bottom right of the console.



Conectarse

Conectarse a la instancia vía **ssh**:

```
ssh -i tu-key-pair.pem ec2-data@xxx.xxx.xxx.xxx
```

Recordemos abrir la conexión al puerto 22 en nuestro security group.

Instalación de Docker

Actualizar los paquetes:

```
sudo yum update -y
```

Instalar el motor:

```
sudo amazon-linux-extras install docker
```

Instalar Docker:

```
sudo yum install docker
```

Inicia Docker:

```
sudo service docker start
```

Utilizar el comando **docker** sin el sudo previo añadiendo el usuario al grupo:

```
sudo usermod -a -G docker ec2-user
```

Hacer dueño al usuario **ec2-user** del daemon de docker:

```
sudo chown ec2-user:docker /var/run/docker.sock
```

Probarlo:

```
docker version
```



```
ec2-user@ip-172-31-0-4:~$ docker version
Client:
Version:      20.10.13
API version:  1.41
Go version:   go1.16.15
Git commit:   a224086
Built:        Thu Mar 31 19:20:32 2022
OS/Arch:      linux/amd64
Context:      default
Experimental: true

Server:
Engine:
Version:      20.10.13
API version:  1.41 (minimum version 1.12)
Go version:   go1.16.15
Git commit:   906f57f
Built:        Thu Mar 31 19:21:13 2022
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.4.13
GitCommit:    9cc61520f4cd876b86e77edfeb88fbcd536d1f9d
runc:
Version:      1.0.3
GitCommit:    f46b6ba2c9314cfc8caae24a32ec5fe9ef1059fe
docker-init:
Version:      0.19.0
GitCommit:    de40ad0
ec2-user@ip-172-31-0-4 ~]$
```

Instalación del servidor

En este caso para trabajar con entornos locales vamos a utilizar [minikube](https://minikube.sigs.k8s.io/docs/start/).

Instalar contrack:

```
sudo yum install contrack
```

Descarga **minikube**:

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

Instalar minikube:

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

Probarlo:

```
minikube version
```

Iniciar minikube:

```
minikube start
```



```
ec2-user@ip-172-31-0-4:~  
[ec2-user@ip-172-31-0-4 ~]$ minikube start  
* minikube v1.26.0 on Amazon 2 (xen/amd64)  
* Automatically selected the docker driver. Other choices: none, ssh  
  
X Exiting due to RSRC_INSUFFICIENT_CORES: Requested cpu count 2 is greater than the available cpus of 1  
[ec2-user@ip-172-31-0-4 ~]$ |
```

Si nos encontramos con este error, no se preocupen. Es normal, básicamente el servidor de Kubernetes (en este caso, minikube) necesita al menos dos CPUs para correr, y cuando seleccionamos en el primer paso **t2.micro** nos indica que tiene un solo CPU. Si cuando ejecutamos el comando **minikube version**, nos brinda la versión, significa que lo hemos instalado correctamente. Si repetimos el paso a paso en una instancia **t2.medium**, funcionará sin problemas.

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows navigation options like 'Panel de EC2', 'Eventos', and 'Instancias'. The main content area is titled 'Resumen de instancia de i-0188fea72b6870018' and provides a detailed overview of the instance's configuration, including its ID, public IP address, state (En ejecución), and instance type (t2.medium). It also lists associated VPC, subnets, and IAM roles.



```
ec2-user@ip-172-31-36-119:~  
[ec2-user@ip-172-31-36-119 ~]$ minikube start  
* minikube v1.26.0 on Amazon 2 (xen/amd64)  
* Automatically selected the docker driver. Other choices: ssh, none  
* Using Docker driver with root privileges  
* Starting control plane node minikube in cluster minikube  
* Pulling base image ...  
* Downloading Kubernetes v1.24.1 preload ...  
  > preloaded-images-k8s-v18-v1...: 405.83 MiB / 405.83 MiB 100.00% 99.55 Mi  
  > gcr.io/k8s-minikube/kicbase: 385.99 MiB / 386.00 MiB 100.00% 18.88 MiB p  
  > gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p/s 13s  
* Creating docker container (CPUs=2, Memory=2200MB) ...  
* Preparing Kubernetes v1.24.1 on Docker 20.10.17 ...  
  - Generating certificates and keys ...  
  - Booting up control plane ...  
  - Configuring RBAC rules ...  
* Verifying Kubernetes components...  
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5  
* Enabled addons: storage-provisioner, default-storageclass  
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'  
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
[ec2-user@ip-172-31-36-119 ~]$
```

Instalación del cliente

kubectl es el cliente de Kubernetes, veamos cómo instalarlo.

Descargar kubectl:

curl -o kubectl

<https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl>

Añadir permisos de ejecución:

chmod +x ./kubectl

Copiar el binario a nuestra carpeta y exportar el path para que el comando **kubectl** quede disponible desde cualquier lugar:

**mkdir -p \$HOME/bin && cp ./kubectl \$HOME/bin/kubectl && export
PATH=\$PATH:\$HOME/bin**

Probar el comando:

kubectl version --short --client



```
ec2-user@ip-172-31-0-4:~  
[ec2-user@ip-172-31-0-4 ~]$ kubectl version --short --client  
Client Version: v1.22.6-eks-7d68063  
[ec2-user@ip-172-31-0-4 ~]$ |
```

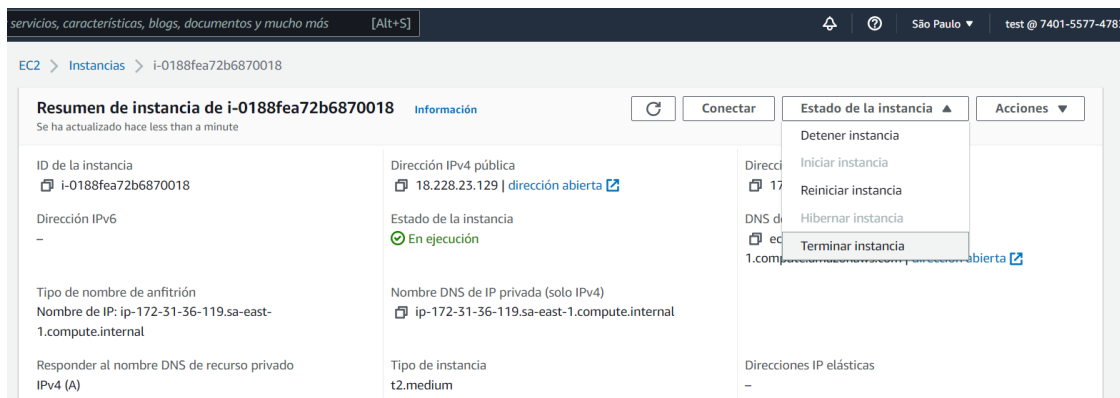
Comprobar la conexión con el servidor:

kubectl cluster-info

```
ec2-user@ip-172-31-36-119:~  
[ec2-user@ip-172-31-36-119 ~]$ kubectl cluster-info  
Kubernetes control plane is running at https://192.168.49.2:8443  
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy  
  
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.  
[ec2-user@ip-172-31-36-119 ~]$
```

Finalizar la instancia

Es importante finalizar la instancia cuando terminamos nuestro trabajo. Mantenerla preñida generará costos innecesarios. Amazon cuenta con varios artículos donde fomenta la reducción de costos. Podemos ver un ejemplo en el siguiente [enlace](#).



Instalación local

Como hemos visto, Kubernetes funciona tanto en servidores Windows, Linux e incluso en computadoras que no son servidores, por ejemplo nuestra computadora personal. Los procedimientos pueden variar de acuerdo al sistema operativo en el que trabajemos. Hay muchos factores que pueden hacer que la instalación no sea sencilla. Por ejemplo, la cantidad de CPUs que tengamos, no tener habilitada la virtualización desde el setup, cantidad de memoria RAM, entre muchos otros (cada computadora es un mundo). Sin embargo, es una buena experiencia intentar instalarlo en nuestra computadora e ir sorteando las dificultades que se vayan presentando.

[Docker Desktop](#) está disponible para Windows, Mac y Linux. Es una muy buena opción para probar, ya que permite habilitar Kubernetes desde su configuración. Si nuestra computadora cumple con todas las condiciones, solamente lo instalamos con un “doble clic” y luego habilitamos Kubernetes.

