

Centro de Procesamiento de Datos



Práctica 5. Creación de máquinas virtuales en AWS. (Opcional)

Objetivo:

Crear un entorno basado en tres máquinas virtuales en AWS y tratar de reproducir la práctica 4.

Presentar un documento pdf en SWAD → Actividades → Práctica 5 con la siguiente información:

- Capturas de la creación de las máquinas virtuales de las distintas formas presentadas en la práctica.
- Para el apartado de GlusterFS, incluir los ficheros utilizados para la creación del entorno virtualizado y capturas de las distintas etapas de instalación y ejecución de GlusterFS.

Desarrollo:

En esta práctica estudiamos cómo automatizar la creación de máquinas virtuales en AWS.

I) Creación de máquinas virtuales desde el entorno WEB de AWS

Primero creamos los elementos necesarios para facilitar cualquier acceso:

- Seleccionamos la región eu-west-1 (Irlanda)
- Creamos un par de claves para el acceso remoto SSH. Almacenamos en nuestro ordenador la clave privada. Por ejemplo, creamos la clave *m1* que utilizaremos posteriormente.
- Editamos en los grupos de seguridad, el grupo por defecto, y añadimos una regla de entrada SSH (puerto 22) para cualquier dirección.

A partir de aquí podemos crear todas las instancias que necesitemos.

Por ejemplo creamos una instancia:

- Basada en la imagen Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-06fd8a495a537da8b (64-bit x86) / ami-03ec287fa560a6ccc (64-bit Arm)
- Tipo: t2.micro
- Grupo de seguridad: (por defecto, con el acceso SSH activo)
- Selección de la clave de acceso:

Una vez creada, podemos probar la conexión SSH:

Conectarse a instancia, cliente SSH, copiamos la orden y probamos el acceso por SSH.

La instancia puede estar :

- En ejecución:
- Detenida: La máquina virtual está parada, pero tiene el coste del almacenamiento .
- Terminada: Borra tanto la máquina como el almacenamiento.

Para continuar con los siguientes apartados creamos unas claves de acceso. Para ello, en el panel IAM (Identity and Access Management), en usuario → credenciales de seguridad → creamos una clave de acceso

En el fichero .aws/config

```
[default]
region = eu-west-1
```

en el fichero .aws/credentials

```
[default]
aws_access_key_id = MI_CLAVE_DE_ACCESO
aws_secret_access_key = MI_CLAVE_SECRETA_DE_ACCESO
```

II) Utilizando AWS CLI (Command Line Interface)

Instalamos el CLI (versión 2) según:

https://docs.aws.amazon.com/es_es/cli/latest/userguide/cli-chap-install.html

Creamos una máquina mediante la siguiente orden:

```
aws ec2 run-instances --image-id ami-06fd8a495a537da8b --key-name m1 --count 1 --
instance-type t2.micro
```

III) Creación mediante Vagrant

Instalamos el plugin de vagrant-aws

```
vagrant plugin install vagrant-aws
```

Para conocer todas las opciones disponibles consultar:

<https://github.com/mitchellh/vagrant-aws>

Instalamos una box "dummy"

```
vagrant box add dummy https://github.com/mitchellh/vagrant-aws/raw/master/dummy.box
```

Creamos un directorio el siguiente fichero Vagrantfile:

```
class Hash
  def slice(*keep_keys)
    h = {}
    keep_keys.each { |key| h[key] = fetch(key) if has_key?(key) }
    h
  end
end
```

```

end unless Hash.method_defined?(:slice)
def except(*less_keys)
  slice(*keys - less_keys)
end unless Hash.method_defined?(:except)
end

Vagrant.configure("2") do |config|
  config.vm.box = "dummy"
  config.ssh.keys_only = false
  config.vm.synced_folder ".", "/vagrant", disabled: true

  config.vm.provider :aws do |aws, override|
    aws.access_key_id = "MI_CLAVE_DE_ACCESO"
    aws.secret_access_key = "MI_CLAVE_SECRETETA_DE_ACCESO"
    aws.keypair_name = "m1"
    aws.instance_type = "t2.micro"
    aws.ami="ami-06fd8a495a537da8b" # ubuntu 20.04 (64-bit x86)
    aws.block_device_mapping = [{ 'DeviceName' => '/dev/sda1', 'Ebs.VolumeSize' => 10 }]
    aws.region = "eu-west-1"
    override.ssh.username = "ubuntu"
    override.ssh.private_key_path = "/Users/antonioidiaz/.ssh/m1.pem"
  end
end

```

Levantamos la máquina:

```
vagrant up --provider=aws
```

Podemos acceder mediante ssh directamente

```
vagrant ssh
```

Parar la máquina

```
vagrant halt
```

Destruir la máquina:

```
vagrant destroy
```

IV) Creación de máquinas con Terraform

Instalamos Terraform: <https://learn.hashicorp.com/tutorials/terraform/install-cli>

Creamos un directorio con el fichero main.tf

```
provider "aws" {  
    region = "eu-west-1"  
}
```

Ejecutamos

```
terraform init
```

Esto instala los controladores para AWS de forma automática.

Podemos incluir nuestras claves de acceso con:

```
AWS_ACCESS_KEY_ID=MI_CLV_DE_A AWS_SECRET_ACCESS_KEY=MI_CLAVE_SEC  
terraform plan
```

Añadimos en el fichero main.tf la información de la instancia que queremos crear

```
resource "aws_instance" "ejemploCPD" {  
    ami          = "ami-06fd8a495a537da8b" # ubuntu AMI (free tier)  
    instance_type = "t2.micro"  
    associate_public_ip_address = true  
    key_name      = "m1"  
}
```

Aplicamos:

```
terraform apply
```

Para conocer los detalles de la máquina:

```
terraform show
```

Conociendo la ip pública podemos acceder como usuario ubuntu por SSH.

Para destruir los recursos:

```
terraform destroy
```