



Certified Tech Developer

The Ultimate Degree

Infraestructura II

Actividad obligatoria e individual.

Dificultad: Alta

Ejercitación Ansible

El objetivo es crear un par de llaves asociadas para conectarse a la instancia; y, luego, una instancia EC2 con IP pública asociada a las llaves creadas anteriormente.

Servicios

A través de la lista de servicios provista por la documentación de Ansible, encontrá los módulos para AWS que necesitás para completar el objetivo de tu ejercicio. [Link](#)

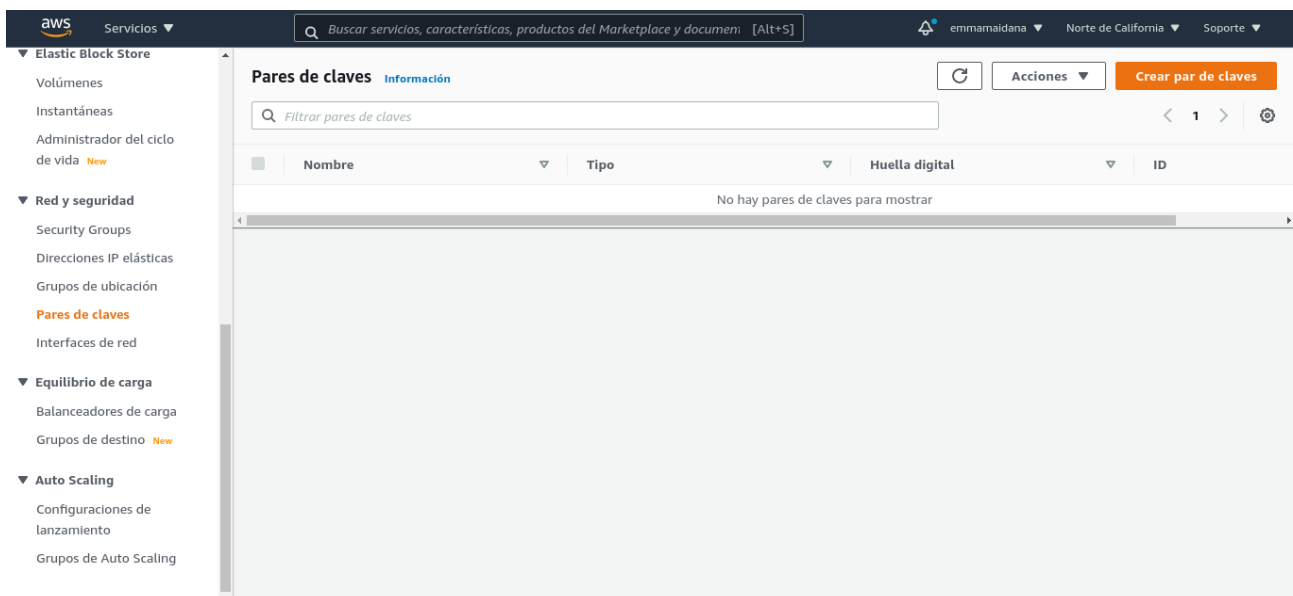
En la siguiente página se encuentra la resolución. Continúa únicamente para realizar una autoevaluación.

Resolución

Antes de comenzar, creá un par de llaves en AWS para asociar a tus instancias, así luego podrás conectarte una vez finalizada la ejecución de tu infraestructura por código.

Vas a poder crear el recurso necesario ingresando de la siguiente manera:

- Ingresá al servicio EC2.
- Luego, al apartado “Red y seguridad” en el menú de la izquierda.
- Hacé clic en “Pares de claves”.



A continuación, completá los siguientes datos como aparecen en la siguiente imagen.

EC2 > Pares de claves > Crear par de claves

Crear par de claves [Información](#)

Par de claves

Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad para demostrar su identidad cuando se conecta a una instancia.

Nombre

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves [Información](#)

☒ RSA

☐ ED25519

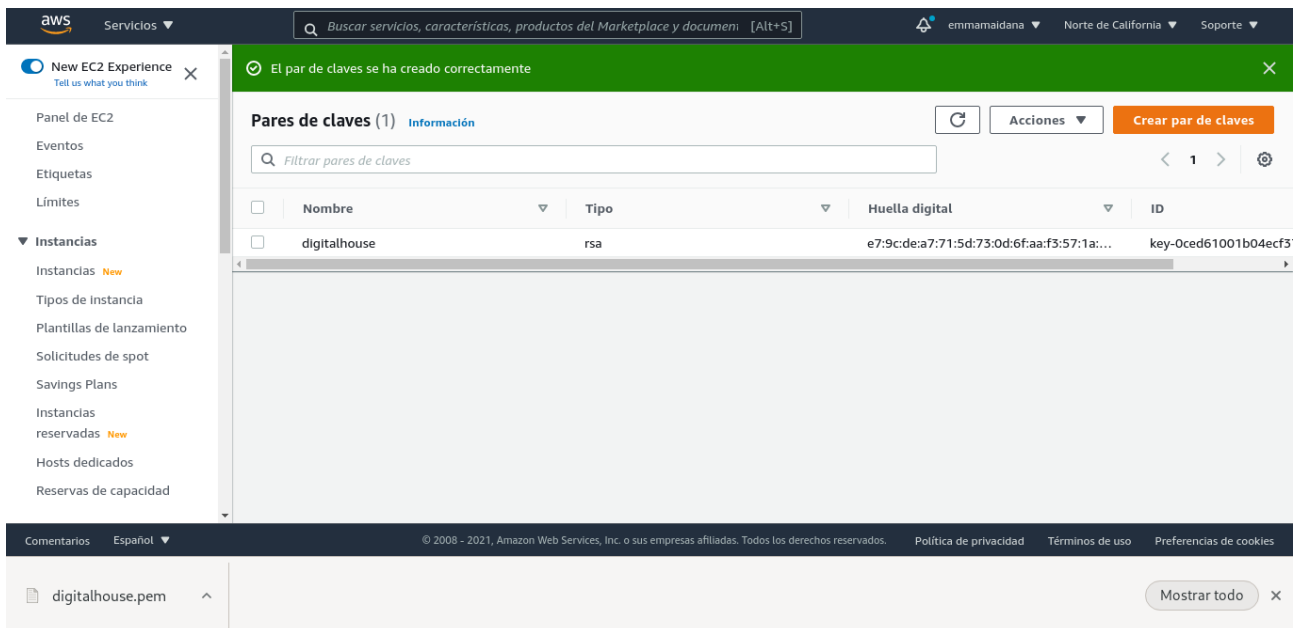
Formato de archivo de clave privada

☒ .pem
Para usar con OpenSSH

☐ .ppk
Para usar con PuTTY

Etiquetas (opcional)

No hay etiquetas asociadas a este recurso.



Una vez seleccionadas estas opciones, se descarga el archivo “digitalhouse.pem” a tu computadora. También vamos a tener que obtener el ID de nuestra VPC por defecto (o crear una nueva) dentro del servicio “VPC” de AWS.

Ahora sí, vamos a aplicar nuestro playbook. El código por parte es el que está a continuación. Primero tenemos que apuntar nuestros hosts, en nuestro caso, vamos a apuntar a “localhost”, ya que no se conecta a otro servidor para crear la instancia.

```
---
- hosts: localhost
```

Definimos las tareas:

```
tasks:
```

Vamos a crear un grupo de seguridad para poder acceder a través de SSH (puerto 22) y conectarnos con nuestro par de llaves. Solo a modo de ejemplo, agregamos también los puertos para HTTP y HTTPS, aunque no son parte de la consigna:

```
- name: Crear grupo de seguridad con HTTPS, HTTP y SSH
```



```
ec2_group:

  name: digitalhouse

  vpc_id: vpc-0a31ba47a968751e4

  description: sg con las reglas

  region: us-west-1

  rules:

    - proto: tcp

      ports:

        - 443

        - 80

        - 22

      cidr_ip: 0.0.0.0/0

      rule_desc: acepto todo el tráfico en puerto 443, 80 y 22
```

Creamos nuestra tarea, es decir, la instancia usando el módulo "ec2".

```
- name: Creamos nuestro servidor

  ec2:
```

Luego, completamos la información necesaria para crear la instancia:

```
ec2:

  region: us-west-1

  instance_type: t2.micro

  image: ami-0ed05376b59b90e46

  wait: yes
```

```
wait_timeout: 500

volumes:

  - device_name: /dev/xvda

    volume_type: gp2

    volume_size: 8

vpc_subnet_id: subnet-070d3b818d23ea3cf

assign_public_ip: yes

key_name: digitalhouse
```

También registramos la salida en una variable que se llama "info". Luego, utilizamos el módulo "debug" para mostrar esa información en el log de salida de Ansible:

```
register: info

- name: IP Pública de nuestro servidor

  debug:

    var: info.instances[0].public_ip

- name: DNS Público de nuestro servidor

  debug:

    var: info.instances[0].public_dns_name
```

La instancia se creó con éxito, se puede evidenciar en la consola de AWS y en los logs de Ansible. Además, comprobamos que nos podemos conectar por ssh correctamente desde nuestras computadoras. Notemos cómo lo que mostramos con "debug" lo utilizamos para conectar a nuestra instancia sin necesidad de entrar a AWS, solo hay que saber el usuario por defecto, que en una "Amazon Linux 2 AMI" siempre es "ec2-user".



```
[enuel@enuel practica_digital_house]$ ansible-playbook ec2.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [localhost] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [Crear grupo de seguridad con HTTPS, HTTP y SSH] *****
changed: [localhost]

TASK [Creamos nuestro servidor] *****
changed: [localhost]

TASK [IP Publica de nuestro servidor] *****
ok: [localhost] => {
  "info.instances[0].public_ip": "52.53.190.226"
}

TASK [DNS Publico de nuestro servidor] *****
ok: [localhost] => {
  "info.instances[0].public_dns_name": "ec2-52-53-190-226.us-west-1.compute.amazonaws.com"
}

PLAY RECAP *****
localhost                : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[enuel@enuel practica_digital_house]$ ssh -i "digitalhouse.pem" ec2-user@ec2-52-53-190-226.us-west-1.compute.amazonaws.com
Last login: Wed Aug 18 19:57:39 2021 from 181.230.142.175

 _ | _ | _ |
 _ | ( _ | /
 _ | \ _ | _ |
  Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
18 package(s) needed for security, out of 23 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-0-29 ~]$ whoami
ec2-user
[ec2-user@ip-10-0-0-29 ~]$
```

ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación ...	Estado de la ...	Zona de dispon...	DNS de IPv4 pública
i-04dd886604a17385f	En ejecución	t2.micro	2/2 comprobador	Sin alarmas	us-west-1c	ec2-52-53-190-226.us

Nuestro código completo es el siguiente:

```
---
- hosts: localhost

  tasks:

  - name: Crear grupo de seguridad con HTTPS, HTTP y SSH

    ec2_group:

      name: digitalhouse

      vpc_id: vpc-0a31ba47a968751e4

      description: sg con las reglas

      region: us-west-1

      rules:

        - proto: tcp
```



```
    ports:

      - 443

      - 80

      - 22

    cidr_ip: 0.0.0.0/0

    rule_desc: acepto todo el trafico en puerto 443, 80 y 22
- name: Creamos nuestro servidor
ec2:

  region: us-west-1

  group: digitalhouse

  instance_type: t2.micro

  image: ami-0ed05376b59b90e46

  wait: yes

  wait_timeout: 500

  volumes:

    - device_name: /dev/xvda

      volume_type: gp2

      volume_size: 8

  vpc_subnet_id: subnet-070d3b818d23ea3cf

  assign_public_ip: yes

  key_name: digitalhouse

  register: info
- name: IP Pública de nuestro servidor
debug:

  var: info.instances[0].public_ip
```




```
- name: DNS Público de nuestro servidor  
  
debug:  
  
  var: info.instances[0].public_dns_name
```