Práctica: Configuración de Vagrant con Autenticación SSH Personalizada

Introducción

En esta práctica, aprenderás a configurar una máquina virtual utilizando Vagrant y a establecer un sistema de autenticación SSH mediante claves personalizadas. Esta configuración se realizará tanto en Windows como en Linux.

El objetivo es automatizar el proceso de configuración para que la máquina virtual acepte una clave pública generada en el host y permita la conexión SSH sin contraseña.

Requisitos previos

1. **Vagrant** instalado en el host.
2. **VirtualBox** como proveedor de máquinas virtuales.
3. Conocimientos básicos de comandos de terminal.

Generación de claves SSH en el host

En Windows:

1. Abre **PowerShell** y ejecuta:

```
ssh-keygen -t rsa -b 2048 -C "vagrant_vm_key" -f
$HOME\.ssh\vagrant_vm_key
```

   - `-t rsa`: Especifica el tipo de clave RSA.
   - `-b 2048`: Define el tamaño de la clave.
   - `-C`: Agrega un comentario para identificar la clave.
   - `-f`: Especifica el archivo donde se guardará la clave.

2. Esto generará dos archivos:

   - `vagrant_vm_key`: Clave privada.
   - `vagrant_vm_key.pub`: Clave pública.

En Linux:

1. Abre una terminal y ejecuta:

```
ssh-keygen -t rsa -b 2048 -C "vagrant_vm_key" -f
~/.ssh/vagrant_vm_key
```

PROF

2. Confirma la generación de los archivos:

```
ls ~/.ssh/vagrant_vm_key*
```

Configuración del `Vagrantfile`

El siguiente script configura una máquina virtual con Vagrant y asegura que la clave SSH generada en el host se copie correctamente y se utilice para la autenticación.

`Vagrantfile` Explicado Paso a Paso

```
Vagrant.configure("2") do |config|
  # Especifica la caja base de Ubuntu
  config.vm.box = "ubuntu/xenial64"

  # Configura una red privada con una IP fija
  config.vm.network "private_network", ip: "192.168.33.10"

  # Configura los recursos de hardware
  config.vm.provider "virtualbox" do |vb|
    vb.memory = "4096"  # 4 GB de memoria
    vb.cpus = 2          # 2 CPUs
  end

  # Copia la clave pública generada en el host a la VM
  config.vm.provision "file", run: "once", source: "#
{Dir.home}/.ssh/vagrant_vm_key.pub", destination:
"/home/vagrant/.ssh/vagrant_vm_key.pub"

  # Configuración inicial del servidor SSH
  config.vm.provision "shell", run: "once" do |s|
    s.inline = <<-SHELL
      # Actualizar e instalar OpenSSH Server
      apt-get update
      apt-get install -y openssh-server

      # Configurar el servidor SSH para aceptar claves públicas
      sed -i 's/#PubkeyAuthentication yes/PubkeyAuthentication yes/'
/etc/ssh/sshd_config
      sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/'
/etc/ssh/sshd_config

      # Crear el directorio .ssh si no existe
      mkdir -p /home/vagrant/.ssh
      chmod 700 /home/vagrant/.ssh
      chown vagrant:vagrant /home/vagrant/.ssh

      # Sobrescribir el contenido de authorized_keys con la clave
pública
```

PROF

```
      cat /home/vagrant/.ssh/vagrant_vm_key.pub >
/home/vagrant/.ssh/authorized_keys

      # Ajustar permisos
      chmod 600 /home/vagrant/.ssh/authorized_keys
      chown vagrant:vagrant /home/vagrant/.ssh/authorized_keys

      # Reiniciar el servidor SSH
      systemctl restart sshd

      loadkeys es
    SHELL
  end

  # Configuración para evitar prompts de SSH
  config.ssh.insert_key = false
  config.ssh.extra_args = ["-o StrictHostKeyChecking=no", "-o
UserKnownHostsFile=/dev/null"]
end
```

Explicación del Script

1. **Copia de la clave pública**:
    - Utiliza `config.vm.provision "file"` para copiar la clave pública desde el host al archivo `/home/vagrant/.ssh/vagrant_vm_key.pub` dentro de la VM.
2. **Configuración del servidor SSH**:
    - Se asegura de que el servidor SSH esté instalado y configurado para aceptar claves públicas (habilitando `PubkeyAuthentication`).
    - La clave pública se sobrescribe en el archivo `authorized_keys` para permitir la autenticación.
    - Ajusta los permisos del directorio `.ssh` y el archivo `authorized_keys`.

PROF

3. **Evitar prompts de SSH**:
    - La opción `config.ssh.insert_key = false` desactiva la generación automática de claves por parte de Vagrant.
    - `config.ssh.extra_args` desactiva verificaciones estrictas del archivo `known_hosts`.

Comandos Importantes

1. Remover entradas conflictivas en `known_hosts`

Si la máquina ya ha sido recreada y el host guarda una clave antigua, podría aparecer un error de autenticación. Usa este comando para eliminar la entrada antigua:

```
ssh-keygen -R [127.0.0.1]:2222
```

- **127.0.0.1:2222**: Indica la IP y el puerto de la máquina virtual en el host.
- Este comando elimina la clave asociada a esa IP/puerto en el archivo `known_hosts`.

2. Probar la conexión SSH manualmente

Desde el host, prueba conectarte a la máquina virtual con:

```
ssh -i ~/.ssh/vagrant_vm_key -p 2222 vagrant@127.0.0.1
```

- `-i ~/.ssh/vagrant_vm_key`: Especifica la clave privada.
- `-p 2222`: Indica el puerto configurado por Vagrant.

Si todo está configurado correctamente, deberías conectarte sin contraseña.

Resolución de Problemas

1. La clave no se copia a la VM

- Verifica que la ruta de la clave en el `Vagrantfile` sea correcta:

  ```
  source: "#{Dir.home}/.ssh/vagrant_vm_key.pub"
  ```

- Usa una ruta absoluta si es necesario:

  ```
  source: "C:/Users/tu_usuario/.ssh/vagrant_vm_key.pub"
  ```

2. Error de permisos en `authorized_keys`

PROF

Si el archivo `authorized_keys` no tiene permisos correctos, configura manualmente desde la VM:

```
chmod 600 /home/vagrant/.ssh/authorized_keys
chmod 700 /home/vagrant/.ssh
chown -R vagrant:vagrant /home/vagrant/.ssh
```

3. SSH no permite la conexión

- Asegúrate de que el servidor SSH esté corriendo:

  ```
  sudo systemctl restart sshd
  ```

- Revisa los logs de SSH para más detalles:

```
sudo journalctl -u ssh
```

**4. Sale este error:

```
192.168.56.10 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh:
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n@
WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
@\r\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\nIT
IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!\r\nSomeone could be
eavesdropping on you right now (man-in-the-middle attack)!\r\nIt is also
possible that a host key has just been changed.\r\nThe fingerprint for
the ED25519 key sent by the remote host
is\nSHA256:MR+lefcqmw4HL4Kz86LBJrS9DsLMWT54qNoLp2VdPMk.\r\nPlease
contact your system administrator.\r\nAdd correct host key in
/home/deivit/.ssh/known_hosts to get rid of this message.\r\nOffending
ECDSA key in /home/deivit/.ssh/known_hosts:3\r\n  remove with:\r\n  ssh-
keygen -f '/home/deivit/.ssh/known_hosts' -R '192.168.56.10'\r\nHost key
for 192.168.56.10 has changed and you have requested strict
checking.\r\nHost key verification failed.",
    "unreachable": true
}
```

El problema que describes está relacionado con el cambio de clave de host SSH. Esto ocurre porque el servidor gestionado ha cambiado su clave de host, pero tu archivo `~/.ssh/known_hosts` aún tiene la clave anterior registrada.

Sigue estos pasos para solucionarlo:

PROF

Solución: Limpieza de la clave de host anterior

1. **Eliminar la clave anterior del archivo `known_hosts`**:

   Ejecuta el siguiente comando para eliminar la entrada del host problemático en el archivo `known_hosts`:

   ```
   ssh-keygen -f "/home/deivit/.ssh/known_hosts" -R "192.168.56.10"
   ```

   Esto elimina la entrada específica para `192.168.56.10`.

---

2. **Verificar la conectividad SSH manualmente**:

Prueba conectarte al servidor gestionado manualmente para registrar la nueva clave de host:

```
ssh -i ~/.ssh/vagrant_vm_key vagrant@192.168.56.10
```

Deberías recibir un mensaje similar a este:

```
The authenticity of host '192.168.56.10' can't be established.
ED25519 key fingerprint is
SHA256:MR+lefcqmw4HL4Kz86LBJrS9DsLMWT54qNoLp2VdPMk.
Are you sure you want to continue connecting (yes/no)?
```

Escribe `yes` para aceptar la nueva clave y registrarla.

---

3. **Configurar Ansible para evitar problemas futuros (opcional)**:

   Si prefieres que Ansible ignore los cambios de claves de host, agrega la configuración `host_key_checking` en tu archivo `ansible.cfg`:

   ```
   [defaults]
   host_key_checking = False
   ```

   Esto deshabilita la verificación de claves de host en el futuro. **Nota:** Este ajuste tiene implicaciones de seguridad, ya que no detectará ataques de tipo "man-in-the-middle".

---

4. **Ejecutar el comando Ansible de nuevo**:

   Ahora intenta ejecutar el comando Ansible nuevamente:

   ```
   ansible -i inventario all -m ping
   ```

   Si todo está configurado correctamente, deberías ver algo como esto:

   ```
   192.168.56.10 | SUCCESS => {
       "changed": false,
       "ping": "pong"
   }
   ```

PROF