Unidad 1

Introducción. VPS y SSH

SERRANO ALBERT, ALFONSO
DESPLIEGUE DE APLICACIONES WEB

Contenido

1. Introducción

En este módulo vamos a simular escenarios reales donde apenas trabajaremos en local, en nuestro propio ordenador. Simularemos, mediante una máquina virtual que es en la que realmente trabajaremos, que todos nuestros despliegues ocurren en una máquina remota, tal y como ocurre en la realidad.

De hecho, se simulará un escenario donde tengamos contratado un VPS (Virtual Private Server) y debamos conectarnos de forma remota al mismo para poder trabajar. Un escenario muy común en el mundo real.

2. ¿Qué es un VPS?

Un servidor es una computadora en la que tu proveedor de alojamiento web almacena los archivos y las bases de datos necesarios para tu sitio web. Cada vez que un visitante en línea quiere acceder a tu sitio web, su navegador le envía una solicitud a tu servidor y transfiere los archivos necesarios a través de Internet. El alojamiento VPS te proporciona un servidor en la nube que simula un servidor físico; sin embargo, en realidad, la máquina se comparte entre varios usuarios.

Al usar la tecnología de virtualización, tu proveedor de alojamiento web instala una capa virtual sobre el sistema operativo del servidor. Esta capa divide el servidor en particiones y le permite a cada usuario instalar su propio sistema operativo y software.

Por lo tanto, un servidor privado virtual (VPS) es tanto virtual como privado porque tienes control absoluto. Está separado de otros usuarios del servidor a nivel del sistema operativo. De hecho, la tecnología VPS es similar a la creación de particiones en tu computadora cuando quieres ejecutar más de un sistema operativo (por ejemplo, Windows y Linux) sin tener que reiniciar.

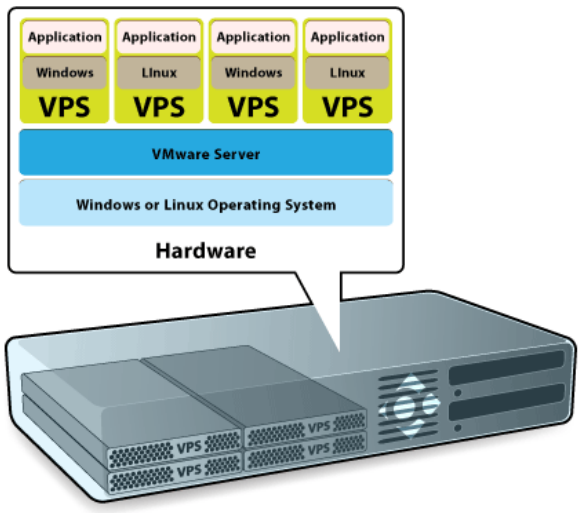

Ilustración 1 Ejemplo de VPS.

Un VPS te permite configurar tu sitio web dentro de un contenedor seguro con recursos garantizados (memoria, espacio en disco, núcleos de CPU, etc.) que no tienes que compartir con otros usuarios. Con el hosting VPS, tienes el mismo acceso de nivel raíz que si alquilaras un servidor dedicado, pero a un costo mucho más bajo.

El VPS es una solución más segura y estable que el hosting compartido, con el que no obtienes espacio de servidor dedicado. Sin embargo, es de menor escala y más barato que alquilar un servidor completo.

El hosting VPS generalmente es elegido por los propietarios de sitios web que tienen un tráfico de nivel medio que excede los límites de los planes de hosting compartido pero que aún no necesitan los recursos de un servidor dedicado.

3. Conexión mediante SSH

Aunque nuestra máquina virtual esté en nuestro ordenador, ya hemos dicho que estamos simulando un VPS remoto. Para conectarnos a una máquina de forma remota y segura, la opción más recomendable es SSH.

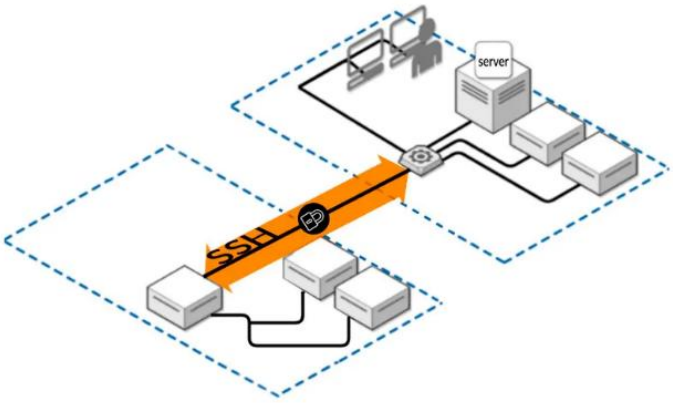

Ilustración 2: Conexión SSH.

SSH o Secure Shell es un protocolo de red criptográfico para operar servicios de red de forma segura a través de una red no protegida. Las aplicaciones típicas incluyen línea de comandos remota, inicio de sesión y ejecución de comandos remota, pero cualquier servicio de red puede protegerse con SSH.

SSH proporciona un canal seguro a través de una red no segura mediante el uso de una arquitectura cliente-servidor, conectando una aplicación cliente SSH con un servidor SSH. El puerto TCP estándar para SSH es 22 y se usa generalmente para acceder a sistemas operativos similares a Unix, pero también se puede usar en Microsoft Windows.

Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente.

SSH tiene muchas aplicaciones diferentes:

- Gestión de servidores a los que no se puede acceder localmente
- Transferencia segura de archivos
- Creación de copias de seguridad
- Conexión entre dos ordenadores con encriptación de extremo a extremo
- Mantenimiento remoto desde otros ordenadores

4. Autenticación

Los dos métodos de autenticación de usuario SSH más comunes que se utilizan son las contraseñas (cifrado simétrico) y las claves SSH (cifrado asimétrico o de clave pública). Los clientes envían contraseñas cifradas al servidor de forma segura. Sin embargo, las contraseñas son un método de autenticación arriesgado porque su solidez depende de que el usuario sepa qué hace que una contraseña sea segura.

Los pares de claves pública-privada SSH encriptados asimétricamente son una mejor opción. Una vez que el cliente descifra el mensaje, el servidor le otorga acceso al sistema.

Es decir, SSH opta por el cifrado híbrido, donde se utiliza el cifrado asimétrico para intercambiar unas claves que serán las que se utilizarán posteriormente en el intercambio de información.

Este tipo de cifrado utiliza la misma clave para cifrar y para descifrar la información. Por este motivo, la clave debe ser secreta y sólo conocida por el emisor y el receptor del mensaje.

5. Cifrados simétricos o de clave privada

Este tipo de cifrado utiliza la misma clave para cifrar y para descifrar la información. Por este motivo, la clave debe ser secreta y sólo conocida por el emisor y el receptor del mensaje.

5.1. Ventajas

- Muy rápidos → cifrar y descifrar un mensaje cada vez requiere un cierto tiempo que, si el algoritmo es complejo, puede ser elevado.

5.2. Inconvenientes

- Si alguien no autorizado consigue la clave, podrá espiar la comunicación sin problemas
- ¿Cómo hacemos para que emisor y receptar conozcan la clave en un primer momento? → no se puede transmitir por el canal inseguro → hay que transmitirla por otro canal seguro Ejemplos: PIN de la tarjeta del banco o archivo comprimido con contraseña

6. Cifrados asimétricos o de clave pública

En este tipo de cifrados cada usuario utiliza un par de claves: una clave pública y una clave privada. Un mensaje cifrado con la clave pública sólo se puede descifrar con su correspondiente clave privada y viceversa.

La *clave pública* es accesible a cualquier persona que quiera consultarla, no hace falta que sea transmitida por un canal seguro como en el caso anterior.

La *clave privada* sólo la debe conocer su dueño.

6.1. Funcionamiento:

1. El emisor cifra un mensaje con la clave pública del receptor
2. El receptor recibe el mensaje y es el único que podrá descifrarlo porque es el único que posee la clave cifrada asociada

6.2. Ventajas

- No se necesita un nuevo canal independiente y seguro para transmitir la clave

6.3. Inconvenientes

- Son más lentos que los cifrados simétricos.
- Hay que proteger muy bien la clave privada y tenerla siempre disponible para poder descifrar los mensajes (no es una contraseña).
- Hay que asegurarse de que la clave pública es de quién dice ser y no de un impostor que se esté haciendo pasar por él.

6.4. Resumen

Característica	Cifrado Simétrico	Cifrado Asimétrico
Nº de claves	1 (compartida)	2 (pública y privada)
Velocidad	Muy rápido	Más lento
Seguridad en distribución	Problema al compartir clave	Más seguro (clave pública es pública)
Uso típico	Cifrado de datos en masa (archivos, discos, VPNs)	Intercambio seguro de claves, autenticación, firmas digitales

7.Ejemplo práctico en Linux

7.1. SSH y los tipos de cifrado

1. **Primera conexión: Cifrado simétrico (contraseña)**
    - Al conectarnos por primera vez a un servidor vía SSH, normalmente introducimos un **usuario y contraseña**.
    - Aquí se está aplicando el **cifrado simétrico**, porque tanto el cliente como el servidor conocen la misma "clave" (la contraseña).
    - Problema: **inseguro y poco práctico**, ya que:
        - La contraseña puede ser interceptada o robada.
        - Hay que introducirla cada vez que se establece la conexión.

2. **Conexiones seguras posteriores: Cifrado asimétrico (par de claves)**
    - Para mejorar la seguridad y la comodidad, SSH permite usar un **par de claves**:
        - **Clave pública**: se guarda en el servidor (~/.ssh/authorized_keys).
        - **Clave privada**: se queda en el ordenador del cliente y **no debe compartirse nunca**.
    - Proceso:
        1. Cuando intentas conectarte, el servidor cifra un desafío con tu **clave pública**.
        2. Solo tu **clave privada** puede descifrarlo.
        3. Si lo resuelves correctamente, la conexión se establece sin necesidad de contraseña.
    - Ventajas:
        - Mucho más seguro (no viaja ninguna contraseña por la red).
        - Cómodo: ya no necesitas escribir tu clave cada vez.
        - Permite automatizar conexiones y despliegues.

7.2. Ejemplo de uso

1. **Generar par de claves en el cliente**:

```
ssh-keygen -t rsa -b 4096
```

2. **Copiar clave pública al servidor**:

```
ssh-copy-id usuario@servidor
```

3. **Conectarse sin contraseña**:

```
ssh usuario@servidor
```