Despliegue de aplicaciones web 2.0

▼ SSH

1. Introducción

- SSH es un **protocolo seguro** para acceder remotamente a servidores.
- Sustituye a **Telnet**, que transmite datos en texto plano.
- Permite:
  - Acceso remoto seguro.
  - Transferencia segura de archivos (SFTP).
  - Gestión de claves de cifrado.
  - Creación de túneles SSH.
  - Reenvío de sesiones X-Windows.

Implementaciones

- OpenSSH es la más utilizada.
- Disponible para múltiples sistemas operativos.

---

2. Instalación de OpenSSH

En el servidor

```
# apt install openssh-server
```

- Instala:
  - `openssh-server` (acceso remoto).

- `openssh-sftp-server` (transferencia segura de archivos).
- `ssh-import-id` (gestión de claves).

Gestión del servicio SSH

```
# systemctl start|stop|restart|status ssh
# service ssh start|stop|restart|status
```

- El proceso asociado es `sshd`.

---

3. Conexión remota

- SSH usa el puerto **22** (Telnet usaba el **21**).
- Para conectarse desde un cliente:

```
$ ssh usuario@IP
$ ssh -l usuario IP
```

- La **primera vez** pedirá confirmar la conexión y guardará la clave en `~/.ssh/known_hosts`.
- Para salir, escribe:

```
exit
```

---

4. Configuración del servidor SSH

El archivo de configuración está en: `/etc/ssh/sshd_config`

Parámetros clave

Parámetro	Descripción
`Port`	Cambia el puerto de escucha (por defecto 22).
`ListenAddress`	Especifica en qué IP escucha el servicio.
`PermitRootLogin no`	Prohíbe el acceso remoto como root.
`MaxSessions`	Límite de sesiones simultáneas.

`AllowUsers` / `DenyUsers`	Permite o bloquea usuarios.
`AllowGroups` / `DenyGroups`	Permite o bloquea grupos.
`X11Forwarding yes`	Activa el reenvío de aplicaciones gráficas.
`Subsystem sftp`	Activa SFTP para transferencia segura de archivos.

Para aplicar cambios:

```
# systemctl restart ssh
```

Ejemplo de archivo de configuración:

Aquí tienes un **ejemplo básico del archivo de configuración** de un servidor OpenSSH (`/etc/ssh/sshd_config`):

```
# Puerto en el que escucha SSH (por defecto 22)
Port 2222  # Cambia el puerto por seguridad

# Dirección IP en la que escucha (0.0.0.0 = todas)
ListenAddress 192.168.1.1

# Registro de eventos
SyslogFacility AUTH
LogLevel INFO

# Permisos de acceso
PermitRootLogin no  # No permitir acceso directo como root
MaxSessions 5  # Máximo 5 sesiones simultáneas

# Autenticación
PasswordAuthentication yes  # Permitir contraseñas
PubkeyAuthentication yes  # Permitir claves SSH

# Control de usuarios
AllowUsers usuario1 usuario2  # Solo estos usuarios pueden acceder
DenyUsers hacker1 hacker2  # Bloquear acceso a estos usuarios

# Reenvío de puertos
```

```
AllowTcpForwarding yes  # Permitir túneles SSH
GatewayPorts no  # Solo accesible desde el servidor

# Reenvío de X11 (aplicaciones gráficas)
X11Forwarding yes

# Habilitar SFTP (transferencia de archivos segura)
Subsystem sftp /usr/lib/openssh/sftp-server
```

Para aplicar cambios, reinicia el servicio:

```
# systemctl restart ssh
```

---

5. Métodos de autenticación

Autenticación con usuario y contraseña

- Se requiere conocer la contraseña del usuario remoto.

Autenticación con clave de cifrado

- Más segura que la contraseña.
- Usa un **par de claves**:
  - **Clave pública**: se comparte con el servidor.
  - **Clave privada**: se guarda en el cliente y **no se comparte**.

Generar clave RSA en el cliente:

```
$ ssh-keygen -t rsa
```

- Guarda las claves en `~/.ssh/id_rsa` (privada) y `~/.ssh/id_rsa.pub` (pública).

Copiar clave pública al servidor:

```
$ ssh-copy-id usuario@IP
```

- Luego, el acceso será sin contraseña.

---

6. Reenvío X11 (Ejecutar apps gráficas remotas)

- Permite abrir aplicaciones gráficas de un servidor remoto.

Configurar servidor (editar `/etc/ssh/sshd_config`):

```
X11Forwarding yes
```

Configurar cliente (editar `/etc/ssh/ssh_config`):

```
ForwardX11 yes
```

Ejecutar conexión con reenvío gráfico:

```
$ ssh -X usuario@servidor
$ firefox &
```

---

7. Transferencia segura de archivos

Con `scp` (copia de archivos)

```
$ scp archivo usuario@servidor:/ruta/destino
$ scp usuario@servidor:/ruta/archivo ./local
```

Con `sftp` (modo interactivo)

```
$ sftp usuario@servidor
sftp> put archivo  # Subir archivo
sftp> get archivo  # Descargar archivo
```

---

8. Reenvío de puertos (SSH Tunneling)

- Permite **cifrar tráfico** de otras aplicaciones o **pasar por cortafuegos**.

Reenvío de puertos locales (`L`)

```
$ ssh -L puerto_local:sistema_destino:puerto_destino usuario@servidor SSH
```

Ejemplo:

```
$ ssh -L 8001:192.168.1.100:8080 usuario@servidor
```

- Accede a `localhost:8001`, pero realmente conectas a `192.168.1.100:8080`.

Reenvío de puertos remotos (`R`)

```
$ ssh -R puerto_remoto:sistema_local:puerto_local usuario@servidorSSH
```

Ejemplo:

```
$ ssh -R 8002:127.0.0.1:80 usuario@servidor
```

- Un usuario en `servidorSSH:8002` accederá al servicio en tu `localhost:80`.

Opciones de configuración en `/etc/ssh/sshd_config`

Parámetro	Descripción
`AllowTcpForwarding yes`	Habilita el reenvío de puertos.
`GatewayPorts yes`	Permite conexiones externas a los puertos reenviados.

9. VPN con SSH

- Crea una **VPN de nivel 3 (Red OSI)** usando SSH.
- Útil para conexiones seguras puntuales.

Configuración en el servidor

```
PermitTunnel yes
PermitRootLogin yes
```

Configuración en el cliente

```
$ ssh -f -w 0:0 usuario@servidorSSH -N
$ ifconfig tun0 10.10.10.2 pointopoint 10.10.10.1 netmask 255.255.255.252
```

Configuración en el servidor

```
$ ifconfig tun0 10.10.10.1 pointopoint 10.10.10.2 netmask 255.255.255.252
```

- Ahora los **clientes y el servidor se comunican cifradamente**.

▼ Tomcat

1. Introducción

- Un servidor de aplicaciones es un software que proporciona servicios a aplicaciones web.
- Tomcat es un servidor de aplicaciones de código abierto que implementa las tecnologías de Jakarta EE.
- Permite desplegar y administrar aplicaciones web en Java.

2. Instalación de Apache Tomcat 10 en Ubuntu 22.04

1. Crear un usuario para Tomcat:

```
sudo useradd -m -d /opt/tomcat -U -s /bin/false tomcat
```

1. Instalar JDK:

```
sudo apt update
sudo apt install default-jdk
```

1. Descargar e instalar Tomcat:

```
cd /tmp
wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.34/bin/apache-tomcat-10.1.34.tar.gz
```

```
sudo tar xzvf apache-tomcat-10*.tar.gz -C /opt/tomcat --strip-compone
nts=1
sudo chown -R tomcat:tomcat /opt/tomcat/
sudo chmod -R u+x /opt/tomcat/bin
```

3. Configuración de usuarios administrativos

1. Editar el archivo de usuarios:

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

1. Agregar los usuarios:

```
<role rolename="manager-gui" />
<user username="manager" password="manager_password" roles="m
anager-gui" />

<role rolename="admin-gui" />
<user username="admin" password="admin_password" roles="manage
r-gui,admin-gui" />
```

1. Eliminar restricciones de acceso en los archivos `context.xml` de `manager` y `host-manager`.

4. Creación de un servicio systemd para Tomcat

1. Crear el archivo de servicio:

```
sudo nano /etc/systemd/system/tomcat.service
```

1. Agregar el siguiente contenido:

```
[Unit]
Description=Tomcat
After=network.target

[Service]
Type=forking
```

```
User=tomcat
Group=tomcat
Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="CATALINA_HOME=/opt/tomcat"
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

1. Cargar y habilitar el servicio:

```
sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo systemctl enable tomcat
```

5. Acceso a la interfaz web

1. Permitir tráfico en el puerto 8080:

```
sudo ufw allow 8080
```

1. Acceder a Tomcat en el navegador:

```
http://<IP_DEL_SERVIDOR>:8080
```

1. Ingresar a "Manager App" con las credenciales configuradas.

6. Despliegue de aplicaciones en Tomcat

1. Copiar el archivo `.war` de la aplicación a `/opt/tomcat/webapps/`:

```
sudo cp aplicacion.war /opt/tomcat/webapps/
```

1. Acceder a la aplicación en:

```
http://<IP_DEL_SERVIDOR>:8080/aplicacion
```

7. Registro de accesos en Tomcat

1. Configurar el archivo `server.xml` para habilitar logs:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
       directory="logs" prefix="access_log" suffix=".txt"
       pattern="common"/>
```

1. Reiniciar Tomcat:

```
sudo systemctl restart tomcat
```

▼ WildFly

1. Introducción

- WildFly (anteriormente JBoss) es un **servidor de aplicaciones de código abierto** orientado a **aplicaciones empresariales (e-business)**.
- Ofrece un entorno ligero y flexible para el desarrollo y despliegue de aplicaciones Jakarta EE.
- Cuenta con herramientas avanzadas de **gestión de memoria, configuración centralizada y administración eficiente**.

2. Instalación y Configuración de WildFly 35 en Ubuntu

2.1. Descargar e instalar WildFly

```
wget https://github.com/wildfly/wildfly/releases/download/35.0.0.Final/
wildfly-35.0.0.Final.tar.gz
tar xvf wildfly-35.0.0.Final.tar.gz
sudo mv wildfly-35.0.0.Final /opt/wildfly
```

2.2. Crear usuario y grupo para ejecutar WildFly

```
sudo groupadd --system wildfly
sudo useradd -s /sbin/nologin --system -d /opt/wildfly -g wildfly wildfly
```

2.3. Configurar servicio systemd

```
sudo mkdir /etc/wildfly
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wild
fly/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/s
ystemd/system/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfl
y/bin/
sudo chmod +x /opt/wildfly/bin/launch.sh
sudo chown -R wildfly:wildfly /opt/wildfly
```

2.4. Iniciar y habilitar WildFly

```
sudo systemctl daemon-reload
sudo systemctl start wildfly
sudo systemctl enable wildfly
```

2.5. Verificar el estado del servicio

```
sudo systemctl status wildfly
```

2.6. Confirmar que WildFly está escuchando en el puerto 8080

```
ss -tunelp | grep 8080
```

3. Creación de usuario administrador

3.1. Ejecutar el script de configuración de usuarios

```
sudo /opt/wildfly/bin/add-user.sh
```

3.2. Configurar usuario administrador

- Seleccionar **opción a** (Management User).
- Introducir un nombre de usuario y contraseña.
- Confirmar la configuración y finalizar.

4. Acceso a la Consola de Administración

4.1. Configurar variables de entorno

```
cat >> ~/.bashrc <<EOF
export WildFly_BIN="/opt/wildfly/bin/"
export PATH=\$PATH:\$WildFly_BIN
EOF
source ~/.bashrc
```

4.2. Conectar con la consola CLI

```
jboss-cli.sh --connect
```

4.3. Verificar la versión de WildFly

```
[standalone@localhost:9990 /] version
```

4.4. Acceso a la consola web

- URL: `http://<IP_DEL_SERVIDOR>:9990`
- Iniciar sesión con las credenciales creadas.

5. Despliegue de una aplicación en WildFly

5.1. Instalación de Java y Maven

```
java -version
mvn -version
```

Si no están instalados:

```
sudo apt install default-jdk maven
```

6.2. Crear aplicación Jakarta EE con Maven

```
mvn archetype:generate -DarchetypeGroupId=org.wildfly.archetype -D
archetypeArtifactId=wildfly-getting-started-archetype
```

6.3. Construcción de la aplicación

```
cd getting-started
mvn package verify
```

6.4. Iniciar la aplicación en WildFly

```
./target/server/bin/standalone.sh
```

- Acceder a la aplicación en `http://localhost:8080/`