### ***Documentación profesional***

**Fase 1: Configuración Inicial del Sistema**

1. **Actualizar el sistema:** Lo primero que hice fue actualizar todos los programas del servidor. Esto es como cuando actualizas las aplicaciones en tu móvil para que todo funcione bien y seguro. Usé unos comandos para que el servidor buscara las últimas versiones de todo y las instalara.
2. **Configurar la zona horaria:** Es importante que el servidor tenga la hora correcta para que los registros y las tareas programadas funcionen bien. Configuramos la hora para que fuera la de Madrid.
3. **Cambiar el nombre del equipo:** Para identificar mi servidor fácilmente, le puse un nombre más específico: "srv-base-daviniaseSierra". Es como ponerle una etiqueta a una caja para saber qué hay dentro.
4. **Crear un usuario normal:** Después de la instalación, usaba el usuario "root", que tiene muchísimos permisos (demasiados para el día a día). Así que creé un usuario nuevo para mí, llamado "daviniase". Este es mi usuario principal para trabajar, y es más seguro que "root". Luego le di permisos especiales (de "sudo") para poder hacer cosas importantes cuando las necesitara, pero sin usar "root" todo el tiempo.

**Fase 2: Conectividad y Acceso Remoto**

1. **Instalar OpenSSH Server:** Para poder conectarme a mi servidor desde lejos, instalé un programa llamado OpenSSH Server. Este programa me permite acceder al servidor de forma segura a través de la terminal, usando algo llamado "SSH". Es como tener una puerta segura a mi servidor.
2. **Generar claves SSH:** Para que la conexión sea aún más segura y no tener que usar contraseñas cada vez, generé unas "claves SSH". Son como una llave especial que solo yo tengo. Generé una "llave pública" (que dejé en el servidor) y una "llave privada" (que guardé en mi ordenador). Así, solo con mi llave puedo entrar, y es mucho más seguro que una contraseña.
3. **Configurar la conexión SSH con claves:** Copié mi llave pública al servidor para que me reconociera. Luego, configuré mi programa Putty (en mi ordenador Windows) para que usara mi llave privada al intentar conectarse al servidor. De esta forma, pude acceder a mi servidor de forma segura y sin teclear contraseña.

**Fase 3: Seguridad Mínima Obligatoria**

1. **Activar el firewall (UFW):** Instalé y activé un "firewall" llamado UFW. Un firewall es como un guarda de seguridad que decide qué tráfico puede entrar y salir de mi servidor. Configuramos reglas para permitir solo lo necesario (como el acceso SSH y el acceso web) y bloqueamos todo lo demás. Esto ayuda a prevenir ataques.
2. **Cambiar el puerto SSH:** Por seguridad, cambié el puerto por defecto que usa SSH (que es el 22) a otro número. Es como cambiar la cerradura de mi puerta a un número secreto que solo yo sé. Esto hace más difícil que intenten entrar atacantes automáticos. Abrí el nuevo puerto en el firewall y cerré el antiguo.
3. **Desactivar el acceso SSH de root:** Impotantísimo: quité la posibilidad de que el usuario "root" (el que tiene todos los permisos) pueda conectarse directamente por SSH. Esto es una medida de seguridad muy fuerte. Así, si alguien intenta adivinar contraseñas, no podrá hacerlo con la cuenta más peligrosa. Para hacer tareas de "root", ahora tengo que conectarme con mi usuario normal ("daviniase") y luego usar sudo para hacer cosas de administrador.

**Fase 4: Estructura de Carpetas y Servicios Iniciales**

1. **Crear estructura de carpetas:** Organicé el espacio en mi servidor creando varias carpetas especiales dentro de /srv/:  
   * /srv/www: Para guardar los proyectos de páginas web.
   * /srv/repositorios: Para guardar el código de mis programas.
   * /srv/docs: Para guardar la documentación técnica.
2. **Establecer permisos de carpetas:** Di permisos específicos a estas carpetas:  
   * Solo el usuario "desarrollador" puede escribir en /srv/www.
   * Solo mi usuario principal ("daviniase") puede entrar en /srv/repositorios.
   * Las otras carpetas tienen permisos más generales.
3. **Instalar servidor web Apache2:** Elegí e instalé Apache2, que es un programa muy popular que convierte mi servidor en un "servidor web". Esto significa que puede mostrar páginas web a cualquiera que las pida.
4. **Configurar Apache2 y página de prueba:** Configuramos Apache2 para que mostrara las páginas desde mi carpeta /srv/www (en lugar de la carpeta que trae por defecto). También creé una página web muy simple (un index.html) dentro de /srv/www con un mensaje. Después de configurar todo correctamente y reiniciar Apache2, ¡pude ver mi página de prueba en el navegador! Esto significa que el servidor web funciona perfectamente.

**CAPTURAS DE PANTALLA**

ESTÁN EN CADA UNA DE LAS FASES DEL BLOQUE 2, DÍA 7

**ERRORES:**

En la fase 2 tenía problemas a la hora de habilitar el servicio ssh.

Compruebo que el Firewall no esté habilitado, cambio la “Red” por “Adaptador puente”, luego por “NAT”, después por “NatNetwork”, después inicio con CDHP, cambio por nuevo dirección Ping 10.0.2.1 y vuelvo a cambiar a “Red” “ NAT”, y por fin funciona…