

Servidor doméstico multipropósito

Este servidor multipropósito tiene las siguientes funciones:

- Alojamiento y compartición de archivos en una red local
- Alojamiento de servidores de Minecraft Java y Bedrock
- Alojamiento de una página web
- Alojamiento de un bot de Discord

Material:

- Gabinete ACTECK Draco GM450 ATX
- ASUS PRIME A320M-K
- 16 Gb RAM
- AMD Athlon 3000G (4) @ 3.500GHz
- ADATA SU630
- Repetidor Steren COM-8300

Procedimiento:

Instalación de Linux Ubuntu

Para este proyecto es necesario instalar Linux Ubuntu, este es un buen sistema operativo para servidores debido a su versatilidad y facilidad de uso en consola.

1. Con una unidad USB flasheada con el sistema operativo, se ejecuta como unidad de memoria a la cual iniciar la máquina.
2. Se siguen los pasos dados por la unidad USB para la instalación.
3. Una vez terminada la instalación se reinicia la máquina y se retira la unidad USB.
4. Encender la máquina con normalidad.

Durante esta instalación se determinó el nombre de usuario como **user-admin**

Instalación de paquetes necesarios

La máquina requiere de paquetes necesarios para el monitoreo y correcto funcionamiento de esta.

1. Ejecutar el comando en consola: **sudo apt update && sudo apt upgrade -y**. Este comando actualiza el sistema operativo a su última versión.
2. **sudo apt install openssh-server**. Este paquete nos permite realizar una conexión ssh desde una computadora al servidor.
3. **sudo apt install btop**. Este paquete nos da el monitoreo a tiempo real de la máquina de forma gráfica.

4. `sudo apt install neofetch`. Este paquete nos permite ver las especificaciones de la máquina.
5. `sudo apt install curl`. Este paquete nos permite descargar archivos de internet sin necesidad de un navegador.

Conexión SSH a través de Remmina

Remmina es un paquete preinstalado en Ubuntu el cual es donde podemos realizar la conexión ssh para poder ejecutar todos los siguientes comandos de consola necesarios para continuar.

1. Ejecutar en consola `sudo service ssh start`
2. Ejecuta `sudo ufw ssh allow` para permitir el tráfico en el puerto.
3. Abrir Remmina en una máquina aparte del servidor para mejor accesibilidad.
4. Click en "Add a new connection profile", el botón de la esquina superior izquierda.
5. Ingresar la dirección IP del servidor (aparece en una sección del comando `btcp`)
6. Ingresar el nombre de usuario y contraseña del servidor

Ahora no es necesario utilizar siempre una pantalla conectada al servidor.

Instalacion de Samba para la compartición de archivos

Samba es un paquete para la compartición de archivos dentro de la misma red local.

1. Ejecutar `sudo apt install samba` en la consola
2. Comprobar la correcta instalación ejecutando el comando `samba --version`
3. Crear una nueva carpeta usando `mkdir sambashare`. Esta va a ser la carpeta que será compartida en la red
4. Ejecutar el comando `sudo nano /etc/samba/smb.conf`. Esto abrirá el archivo de configuración
5. Añadir el siguiente texto al final del archivo:

```
[sambashare]
comment = Samba on Ubuntu
path = /home/user-admin/sambashare
read only = no
browsable = yes
```

6. Presione Ctrl + X, y, Enter
7. Ejecutar el comando `sudo service smbd restart` para reiniciar el servicio y guardar los cambios
8. Ejecute `sudo ufw 20 allow` y `sudo ufw 21 allow` para permitir el tráfico en estos puertos
9. Añada los archivos que quiera compartir a la carpeta.

Ahora la carpeta puede ser accedida en cualquier computadora ya sea de Windows o Linux usando el servidor `smb://192.168.1.199/sambashare` en Linux y usando `\\192.168.1.199\sambashare` en Windows.

Alojamiento de servidores de Minecraft

Java

Un servidor de Minecraft Java es esencialmente un archivo donde se aloja la información del mundo. Pero para ello se necesita de algunas pre-dependencias:

1. `sudo apt install openjdk-19-jre` se ejecuta para la instalación del paquete de ejecución del Java Runtime environment versión 19
2. Se comprueba la correcta instalación usando `java -version` en la consola
3. Se crea una nueva carpeta usando el comando `mkdir minecraft`
4. `cd minecraft` para acceder a la carpeta
5. `curl -O https://api.papermc.io/v2/projects/paper/versions/1.19.4/builds/550/downloads/paper-1.19.4-550.jar` para descargar el archivo de creación del servidor. En este caso se está descargando la versión 1.19.4 para evitar problemas de compatibilidad mientras se ejecuta una versión reciente del juego.
6. `java -jar paper-1.19.4-550.jar` para ejecutar el archivo jar y crear los archivos para el juego
7. Después de ejecutar el comando se detendrá el juego, parándose abruptamente. Esto es normal para la primera ejecución. Para evitar que vuelva a pasar se ejecuta `nano eula.txt`
8. Una vez en el editor de archivo, cambie "false" por "true" en la 3ra línea. Presione Ctrl + X, y, Enter
9. Ejecuta `sudo ufw 25565 allow`. Este comando permite el tráfico en el puerto.
10. Vuelva a ejecutar el comando `java -jar paper-1.19.4-550.jar` para ejecutar el archivo jar y el juego

Pasos opcionales para facilidad

Para facilitar la adición de plugins al juego se comparte temporalmente la carpeta `~/minecraft/plugins` a través de samba.

1. Ejecute el comando `sudo nano /etc/samba/smb.conf` para abrir la configuración.
2. Añada el siguiente texto al final de archivo:

```
[plugins]
    comment = Server's plugins
    path = /home/user-admin/minecraft/plugins
    read only = no
    browsable = yes
```

3. Presione Ctrl + X, y, Enter para guardar los cambios

Ahora para añadir los plugins solo se descargan el a maquina que esta y se arrastran los archivos jar a la carpeta.

Bedrock

Para facilitar la creación del mundo del juego se comparte temporalmente la carpeta donde se alojará a través de samba.

1. Ejecuta `mkdir bedrock` para crear la carpeta.
2. Ejecute el comando `sudo nano /etc/samba/smb.conf` para abrir la configuración.
3. Añada el siguiente texto al final de archivo:

```
[bedrock]
comment = Server folder
path = /home/user-admin/bedrock
read only = no
browsable = yes
```

4. Presione Ctrl + X, y, Enter para guardar los cambios
5. Se descarga el archivo zip de <https://www.minecraft.net/en-us/download/server/bedrock>
6. Una vez descargado, se arrastra el archivo zip a la carpeta compartida
7. Una vez en la carpeta, se extraen los archivos
8. Cuando termine de extraer los archivos, se ejecuta el comando `cd bedrock` para entrar a la carpeta por consola y `LD_LIBRARY_PATH=. ./bedrock_server` para ejecutar el servidor
9. Ejecute el comando `sudo ufw 19132 allow` para permitir el tráfico a este puerto

Alojamiento de pagina web

El propósito original de este servicio es para que sea posible compartir archivos seleccionados con usuarios móviles.

1. Ejecutamos el comando `sudo apt install apache2`
2. Comprobamos la correcta instalación con ingresar la dirección IP en el navegador (en este caso 192.168.1.199)
3. Una vez confirmado, se crea un archivo html aparte
4. Para la facilidad del usuario, se ejecuta el comando `sudo nano /var/www/html/index.html`
5. Una vez abierto el editor de texto, se elimina todo el contenido del archivo y se pega el contenido del html creado previamente
6. Presione Ctrl + X, y, Enter para guardar los cambios
7. Ejecute el comando `sudo ufw http allow` y `sudo ufw https allow` para permitir el tráfico al servicio

Alojamiento de un bot de Discord

El bot usado en este caso es JMusicBot (<https://github.com/jagrosh/MusicBot>), un bot de código abierto con soporte a varias plataformas. Para la ejecución de este bot se necesita de Java JRE el cual ya ha sido instalado previamente. Para la comprensión de esta documentación, se le ha dado al bot el nombre de "Nokia"

1. Se crea una carpeta usando el comando `mkdir nokia`
2. `cd nokia`
3. Ejecute el comando `curl -O`
`https://github.com/jagrosh/MusicBot/releases/download/0.3.9/JMusicBot-0.3.9.jar`
para descargar el archivo en la carpeta
4. Una vez descargado, ejecute el comando `java -jar JMusicBot-0.3.9.jar`
5. Al ejecutar el comando, el archivo pedirá un Bot Token de Discord, ingreselo
6. Después de esto, pedirá el User ID del usuario que creó la aplicación de Discord. Ingreselo
7. Una vez ingresados los datos pedidos el bot se ejecutará y conectará a Discord correctamente.

Con esto concluye la creación de este servidor.