DuckDNS es un servicio de DNS dinámico que permite asignar un nombre de dominio personalizado a una dirección IP dinámica. Es decir, si tu proveedor de servicios de Internet (ISP) te asigna una dirección IP dinámica que cambia cada vez que te conectas a internet, DuckDNS te permite tener una forma fácil de acceder a tu dispositivo o proyecto en línea sin tener que recordar la dirección IP en constante cambio.

En mi caso, he utilizado DuckDNS para asignar el nombre de dominio fernandoayoso.duckdns.org a la dirección IP de tu Raspberry Pi (192.168.1.94) que aloja diversos proyectos y servicios como Samba, Jellyfin, Transmission, etc. De esta manera, en lugar de tener que recordar la dirección IP dinámica, simplemente puedes acceder a tus servicios a través de este nombre de dominio fácil de recordar.

Entre las principales ventajas de utilizar DuckDNS se encuentran:

* Acceso remoto fácil: Al utilizar un nombre de dominio personalizado, puedes acceder a tus servicios en línea desde cualquier lugar del mundo sin tener que recordar la dirección IP dinámica.
* Seguridad: Utilizando DuckDNS y una conexión segura (HTTPS), puedes cifrar el tráfico de tus servicios y aumentar la seguridad de tu proyecto en línea.
* Gratuito: El servicio es completamente gratuito y no requiere de un registro de tarjeta de crédito u otra información de pago.

Para utilizar DuckDNS, simplemente necesitaremos registrarnos en su sitio web y seguir las instrucciones para crear tu propio nombre de dominio personalizado. Una vez que hayas completado la configuración, podrás acceder a los servicios en línea a través de este nombre de dominio

Estos comandos corresponden a la configuración de DuckDNS para actualizar la dirección IP pública de tu Raspberry Pi. A continuación se detalla el propósito de cada uno de ellos:

1. mkdir duckdns: Este comando crea un directorio llamado "duckdns" en el directorio actual.
2. cd duckdns: Este comando cambia el directorio actual al directorio "duckdns" recién creado.
3. sudo nano duck.sh: Este comando abre el editor de texto "nano" y crea un nuevo archivo llamado "duck.sh" dentro del directorio "duckdns". Este archivo será utilizado para ejecutar la actualización de DuckDNS.
4. Echo url="https://www.duckdns.org/update?domains=fernandoayoso&token=4b70b9bd-cf88-4ef4-aadd-1ef828049ebb&ip=" | curl -k -o /duckdns/duck.log -K -: Este comando escribe una cadena de caracteres en la salida estándar que incluye la URL de DuckDNS con tu token de acceso y el nombre de dominio que deseas actualizar. Luego, esta cadena de caracteres se pasa a curl, que utiliza la URL para enviar una solicitud de actualización a DuckDNS. El resultado de la solicitud se guarda en un archivo llamado "duck.log" en el directorio "duckdns".
5. sudo chmod 700 duck.sh: Este comando cambia los permisos del archivo "duck.sh" para que solo el usuario root pueda leer, escribir y ejecutar el archivo.
6. crontab -e: Este comando abre el archivo de configuración de cron para el usuario actual. Cron es un servicio de programación de tareas en segundo plano que se utiliza comúnmente en sistemas Linux.
7. \*/5 \* \* \* \* ~/duckdns/duck.sh >/dev/null 2>&1: Este comando añade una tarea programada al archivo de configuración de cron que ejecutará el script "duck.sh" cada 5 minutos. La redirección de salida (>/dev/null 2>&1) asegura que no se envíen notificaciones por correo electrónico sobre el resultado de la tarea programada.
8. ./duck.sh: Este comando ejecuta el archivo "duck.sh" manualmente para probar que la actualización de DuckDNS funciona correctamente.
9. cat duck.log: Este comando muestra el contenido del archivo "duck.log" en la salida estándar, que debe contener la respuesta de DuckDNS a la solicitud de actualización.

mkdir duckdns

cd duckdns

sudo nano duck.sh

echo url="https://www.duckdns.org/update?domains=fernandoayoso&token=4b70b9bd-cf88-4ef4-aadd-1ef828049ebb&ip=" | curl -k -o /duckdns/duck.log -K -

sudo chmod 700 duck.sh

crontab -e

\*/5 \* \* \* \* ~/duckdns/duck.sh >/dev/null 2>&1

./duck.sh

cat duck.log