

PROYECTO: APACHE GUACAMOLE

GUILLERMO CAÑERO PÉREZ





1. ANTECEDENTES	3
2. DEFINICIÓN DE PROBLEMA	4
3. JUSTIFICACIÓN	5
4. OBJETIVOS	6
5. PRESUPUESTO	8
6. LIMITACIONES	9
7. ALCANCE	10
8. MARCO DE REFERENCIA	11
9. PROCEDIMIENTOS	12
10. DESCRIPCIÓN DE LAS ACTIVIDADES	13
11. PROCEDIMIENTO ACTIVIDADES	15
11.2 ESCENARIO PRÁCTICA	15
11.3 INSTALAR DOCKER Y DOCKER-COMPOSE	16
11.4 DESCARGAR FICHERO DOCKER-COMPOSE	17
11.4 REVISAR EL FICHERO DOCKER-COMPOSE	19
11.5 ARRANCAR DOCKER-COMPOSE	20
11.6 CONFIGURAR GUACAMOLE	21
11.7 AÑADIR VPN A DOCKER-COMPOSE	24
11.8 CREAR CONEXIÓN SSH UBUNTU-SERVER	29
11.9 CREAR CONEXIÓN RDP WINDOWS 10	31
11.10 CREAR CONEXIÓN VNC UBUNTU DESKTOP	33
12. ESQUEMA	40
13. CRONOGRAMA	42
14. RIESGOS LABORALES	43
15. CONCLUSIONES	44
a. DEBILIDADES	44
b. AMENAZAS	46
c. FORTALEZAS	48
d. OPORTUNIDADES	50
e. AMPLIACIONES FUTURAS	52
f. ESCABILIDAD	54
16. BIBLIOGRAFÍA	56



1. ANTECEDENTES

Antes de configurar Apache Guacamole, es importante tener conocimientos básicos de redes, servidores y sistemas operativos. Además, es necesario tener conocimientos sobre los siguientes temas:

- Protocolos de acceso remoto:** Apache Guacamole es una herramienta que permite acceder a otros sistemas a través de diferentes protocolos de acceso remoto, como SSH, RDP y VNC. Por lo tanto, es importante tener conocimientos sobre estos protocolos y cómo funcionan.
- Servidores web:** Apache Guacamole se ejecuta en un servidor web, por lo que es importante tener conocimientos sobre cómo configurar y administrar servidores web, como Apache o Nginx.
- Sistemas operativos:** Apache Guacamole se puede ejecutar en diferentes sistemas operativos, como Linux, Windows o macOS. Es importante tener conocimientos sobre el sistema operativo en el que se va a instalar y configurar Apache Guacamole.
- Bases de datos:** Apache Guacamole utiliza una base de datos para almacenar información de configuración y registros de acceso. Es importante tener conocimientos sobre bases de datos y cómo configurarlas.
- Seguridad:** Es importante tener conocimientos sobre seguridad informática y cómo configurar Apache Guacamole de forma segura, como la autenticación de usuarios, el cifrado de datos y la configuración de firewalls.



2. DEFINICIÓN DE PROBLEMA

·El problema que estamos experimentando es la falta de espacio de almacenamiento en nuestros equipos de clase debido al uso de VirtualBox. VirtualBox es un software que permite ejecutar sistemas operativos y aplicaciones en una máquina virtual, lo que puede requerir un gran espacio de almacenamiento en nuestro disco duro.

·Para solucionar este problema, he recurrido a Apache Guacamole, que es una herramienta que permite acceder a otras máquinas remotas a través de diferentes protocolos de acceso remoto como RDP, VNC y SSH, sin la necesidad de instalar ninguna aplicación adicional en tu equipo local. De esta manera, podemos acceder a máquinas remotas con gran capacidad de almacenamiento y procesamiento, sin tener que preocuparnos por el espacio de almacenamiento en nuestro equipo local de clase.

·En resumen, el problema que estamos experimentando es la falta de espacio de almacenamiento en nuestro equipo debido al uso de VirtualBox, y la solución que he encontrado es utilizar Apache Guacamole para acceder a máquinas remotas con mayor capacidad de almacenamiento y procesamiento.



Apache Guacamole™



3. JUSTIFICACIÓN

Instalar y configurar Apache Guacamole puede ayudar a resolver problemas de espacio en varias formas:

- Centralización:** Al utilizar Apache Guacamole para acceder a diferentes máquinas remotas, se evita la necesidad de tener múltiples herramientas de acceso remoto instaladas en la computadora local. Esto reduce la cantidad de espacio que se utiliza en el disco duro y se evita la necesidad de tener que descargar y actualizar múltiples aplicaciones de acceso remoto.

- Virtualización:** Apache Guacamole permite a los usuarios acceder a máquinas virtuales alojadas en servidores remotos. Al hacerlo, se elimina la necesidad de tener múltiples sistemas operativos instalados en el equipo local, lo que puede ocupar mucho espacio en el disco duro.

- Optimización de recursos:** Al utilizar Apache Guacamole, se puede acceder a recursos remotos sin la necesidad de tenerlos instalados localmente. Esto puede incluir software especializado o archivos grandes. Al no tener que mantener estos recursos localmente, se reduce la cantidad de espacio que se necesita en el disco duro.

- En resumen, al instalar y configurar Apache Guacamole, se puede reducir la cantidad de espacio que se necesita en el disco duro, al centralizar el acceso remoto y virtualizar las máquinas y recursos.



4. **OBJETIVOS**

·**OBJETIVOS ESPECÍFICOS:** El objetivo específico de Apache Guacamole es proporcionar un cliente de escritorio remoto de código abierto y sin cliente que permita a los usuarios acceder a sus escritorios y aplicaciones desde cualquier lugar utilizando un navegador web estándar. Su objetivo es simplificar el acceso remoto a diferentes sistemas y aplicaciones al proporcionar una solución unificada y accesible desde cualquier dispositivo, sin necesidad de instalar software adicional o plugins en el dispositivo del usuario final. Además, Apache Guacamole tiene como objetivo mejorar la seguridad al permitir que el acceso remoto se realice a través de conexiones cifradas y autenticadas, y al proporcionar opciones de autenticación multifactor para una mayor protección.

·**OBJETIVOS GENERALES:** El objetivo principal es simplificar el acceso remoto y mejorar la seguridad mediante la eliminación de la necesidad de instalar software adicional o plugins en el dispositivo del usuario final, y al proporcionar opciones de autenticación seguras y cifradas. Apache Guacamole también tiene como objetivo reducir los costos y complejidad de la administración de acceso remoto, al proporcionar una solución centralizada y unificada que pueda ser fácilmente escalada y personalizada según las necesidades del usuario. En general, el objetivo de Apache Guacamole es mejorar la eficiencia y productividad de los usuarios, permitiéndoles acceder a sus sistemas y aplicaciones desde cualquier lugar y en cualquier momento.



·**OBJETIVOS GENÉRICOS:** ·Los objetivos genéricos de Apache Guacamole son los siguientes:

- Proporcionar una solución de acceso remoto de código abierto y sin cliente que permita a los usuarios acceder a sus sistemas y aplicaciones desde cualquier lugar utilizando un navegador web estándar.

- Simplificar el acceso remoto al eliminar la necesidad de instalar software adicional o plugins en el dispositivo del usuario final.

- Mejorar la seguridad del acceso remoto mediante la utilización de conexiones cifradas y autenticadas, así como opciones de autenticación multifactor para una mayor protección.

- Reducir los costos y complejidad de la administración de acceso remoto, al proporcionar una solución centralizada y unificada que pueda ser fácilmente escalada y personalizada según las necesidades del usuario.

- Mejorar la eficiencia y productividad de los usuarios al permitirles acceder a sus sistemas y aplicaciones desde cualquier lugar y en cualquier momento, lo que les permite trabajar de manera remota de manera más efectiva.

·Los objetivos genéricos de Apache Guacamole son mejorar la accesibilidad, seguridad, simplicidad y eficiencia del acceso remoto, proporcionando una solución de escritorio remoto unificada y escalable que puede ser personalizada según las necesidades del usuario.



5. PRESUPUESTO

·El costo de configurar Apache Guacamole depende de varios factores, como el tamaño de la organización, el número de usuarios y dispositivos, y los recursos de hardware y software disponibles.

·En primer lugar, es importante tener en cuenta que Apache Guacamole es un software de código abierto, lo que significa que es gratuito para descargar y usar. Sin embargo, aún existen costos asociados con su implementación y configuración, que pueden incluir:

·**Hardware:** Es posible que se necesite hardware adicional, como servidores o dispositivos de red, para alojar y ejecutar Apache Guacamole, según las necesidades de la organización y la cantidad de usuarios y dispositivos que se conectarán.

·**Personal:** Si no se cuenta con personal interno capacitado para la implementación y configuración de Apache Guacamole, es posible que se necesite contratar a un consultor o proveedor de servicios para ayudar en el proceso.

·**Mantenimiento y soporte:** Se puede requerir la contratación de personal para el mantenimiento y soporte continuo de la solución, incluyendo actualizaciones de seguridad, resolución de problemas y soporte técnico para los usuarios.



6. LIMITACIONES

· Aunque Apache Guacamole es una solución de escritorio remoto de código abierto y sin cliente altamente efectiva, tiene algunas limitaciones que se deben considerar:

· **Requisitos de hardware:** Para una implementación adecuada, se requiere hardware específico, como servidores o dispositivos de red, que pueden aumentar los costos y la complejidad de la implementación.

· **Compatibilidad de los protocolos de acceso remoto:** Aunque Guacamole es compatible con varios protocolos de acceso remoto, algunos sistemas y aplicaciones no son compatibles con estos protocolos y, por lo tanto, no se pueden acceder mediante esta solución.

· **Limitaciones de rendimiento:** El rendimiento de Guacamole puede verse afectado por factores como el ancho de banda de red, la calidad de la conexión a Internet y la carga del servidor, lo que puede causar retrasos y disminuir la calidad de la experiencia del usuario.

· **Requisitos de seguridad:** La seguridad es un factor crítico en la implementación de Guacamole, y puede requerir un conocimiento especializado para garantizar una configuración segura.



7. ALCANCE

·Apache Guacamole está dirigido a cualquier organización o empresa que necesite una solución de acceso remoto para sus empleados o usuarios, especialmente aquellas que necesitan acceder a diferentes sistemas y aplicaciones desde ubicaciones remotas. Esto incluye a empresas de todos los tamaños y sectores, como agencias gubernamentales, empresas de salud, educación, finanzas, TI y más.

·En particular, Guacamole es útil para empresas con empleados remotos, usuarios móviles, contratistas externos y socios comerciales que necesitan acceso a los recursos de la organización, independientemente de su ubicación física. También puede ser una solución efectiva para organizaciones que buscan reducir los costos y la complejidad asociados con la gestión de múltiples soluciones de acceso remoto.

·En resumen, Apache Guacamole está dirigido a cualquier organización que necesite una solución segura y fácil de usar para el acceso remoto a sus sistemas y aplicaciones.





8. MARCO DE REFERENCIA

- El marco de referencia de Apache Guacamole es el de una solución de escritorio remoto de código abierto y sin cliente, diseñada para permitir el acceso seguro y fácil a sistemas y aplicaciones desde cualquier ubicación, utilizando un navegador web.
- Guacamole se basa en un servidor central que proporciona conexiones a sistemas y aplicaciones a través de una variedad de protocolos de acceso remoto, incluyendo RDP, VNC, SSH y Telnet. Los usuarios pueden acceder a estos recursos mediante un cliente HTML5 en su navegador web, lo que elimina la necesidad de instalar software cliente adicional en sus dispositivos.
- Además, Guacamole proporciona funciones avanzadas de seguridad y autenticación, incluyendo soporte para autenticación multifactor, cifrado de extremo a extremo y la capacidad de integrarse con sistemas de autenticación existentes.
- En definitiva, el marco de referencia de Apache Guacamole es el de una solución de acceso remoto de código abierto y sin cliente que proporciona una experiencia de usuario fácil y segura para el acceso a sistemas y aplicaciones desde cualquier ubicación





9. PROCEDIMIENTOS

·En este proyecto vamos a ver como controlar todos nuestros equipos desde una única interfaz web, tanto si tienen IP pública como privada a través de una VPN.

·Al final veremos controlar remotamente equipos con Windows, Ubuntu Server y Ubuntu Desktop.

·El orden sería el siguiente:

1. ESCENARIO DE LA PRÁCTICA
2. INSTALAR DOCKER Y DOCKER-COMPOSE
3. DESCARGAR FICHERO DOCKER-COMPOSE
4. REVISAR EL FICHERO DOCKER-COMPOSE
5. ARRANCAR DOCKER-COMPOSE
6. CONFIGURAR GUACAMOLE
7. AÑADIR VPN A DOCKER-COMPOSE
8. CREAR CONEXIÓN SSH UBUNTU-SERVER
9. CREAR CONEXIÓN RDP WINDOWS 10
10. CREAR CONEXIÓN VNC UBUNTU DESKTOP



10. DESCRIPCIÓN DE LAS ACTIVIDADES

·**Escenario de la práctica:** Nos referimos al contexto o entorno en el que se llevarán a cabo las actividades para instalar y configurar Apache Guacamole, que podría ser un sistema operativo específico o una infraestructura de red determinada.

·**Instalar Docker y Docker-Compose:** Es la actividad de instalar las herramientas de Docker y Docker-Compose en el sistema o servidor donde se ejecutará Guacamole. Docker es un motor de contenedores que permite crear y ejecutar aplicaciones en un entorno aislado, mientras que Docker-Compose es una herramienta para definir y ejecutar múltiples contenedores Docker como una aplicación.

·**Descargar fichero Docker-Compose:** Consiste en descargar el archivo de configuración de Docker-Compose que define la estructura de contenedores que se ejecutarán para la solución Guacamole.

·**Revisar el fichero Docker-Compose:** Es la actividad de revisar y modificar, si es necesario, el archivo de configuración de Docker-Compose para asegurarse de que se estén utilizando las imágenes correctas de Guacamole y las conexiones a los sistemas y aplicaciones que se necesitan.

·**Arrancar Docker-Compose:** Implica ejecutar el archivo de configuración de Docker-Compose para arrancar los contenedores de Guacamole y las conexiones correspondientes.



- Configurar Guacamole:** Consiste en realizar la configuración inicial de Guacamole, como la definición de los usuarios, las conexiones a los sistemas y aplicaciones, y la configuración de seguridad.
- Añadir VPN a Docker-Compose:** Se refiere a la actividad de agregar un contenedor adicional que proporcione una conexión VPN segura para el acceso remoto a los sistemas y aplicaciones que se encuentran detrás del firewall de la red.
- Crear conexión SSH Ubuntu-Server:** Es la actividad de crear una conexión segura SSH al servidor Ubuntu desde Guacamole, lo que permitirá el acceso remoto a la línea de comandos del servidor.
- Crear conexión RDP Windows 10:** Consiste en crear una conexión de Escritorio Remoto (RDP) al sistema operativo Windows 10 desde Guacamole, lo que permitirá el acceso remoto al escritorio del sistema.
- Crear conexión VNC Ubuntu Desktop:** Se refiere a la actividad de crear una conexión de VNC (Virtual Network Computing) al escritorio de Ubuntu desde Guacamole, lo que permitirá el acceso remoto al entorno gráfico del sistema.



11. PROCEDIMIENTO ACTIVIDADES

11.2 ESCENARIO PRÁCTICA

·En primer lugar, voy a instalar Guacamole en una máquina virtual de Ubuntu Server que me conectaré para via ssh. En lugar de instalar Guacamole directamente en mi servidor principal, estoy optando por utilizar una máquina virtual (VM) de Ubuntu Server. La ventaja de esto es que puedes experimentar con diferentes configuraciones sin afectar a mi equipo principal. La conexión a la VM se realizará a través de SSH, que es un protocolo seguro para conectarse a un servidor remoto.

·También voy a utilizar un Ubuntu Server con IP pública para que los usuarios remotos puedan acceder a Guacamole. Para hacer esto, debemos tener una dirección IP pública asignada a mi Ubuntu. Esto significa que tu servidor será visible desde Internet y podrás acceder a él desde cualquier lugar.

·Guacamole puede conectarse a sistemas remotos que ejecutan diferentes sistemas operativos, como Windows. De esta manera, podrás conectarte a tus servidores de Windows utilizando Guacamole desde cualquier lugar. En este caso tendremos una máquina virtual de Windows.

·Y por último optaré por utilizar Ubuntu Desktop como un cliente para conectarse a Guacamole. Para conectarte a Guacamole de forma segura desde cualquier lugar, debemos utilizar una conexión VPN.



11.3 INSTALAR DOCKER Y DOCKER-COMPOSE

Primero nos conectamos vía ssh a la máquina virtual de Guacamole.

```
usuario@us: ~  
usuario@us:~$
```

Ahora ejecutamos el siguiente comando para actualizar el repositorio, para instalar docker.io y docker-compose.

```
usuario@us:~$ sudo apt-get update && sudo apt-get install docker.io docker-compose -y  
[sudo] password for usuario:  
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease  
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Des:5 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [978 kB]  
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [209 kB]  
Des:7 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13,8 kB]  
Des:8 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [731 kB]  
Des:9 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [114 kB]  
Des:10 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [588 B]  
Des:11 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [897 kB]  
Des:12 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [179 kB]  
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18,4 kB]  
Des:14 http://es.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24,1 kB]
```

Comprobamos la versión que hemos instalado.

```
usuario@us:~$ docker -v  
Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.2  
usuario@us:~$ docker-compose -v  
docker-compose version 1.29.2, build unknown  
usuario@us:~$
```




11.4 DESCARGAR FICHERO DOCKER-COMPOSE

- El fichero docker-compose es un archivo de configuración utilizado por la herramienta Docker Compose para definir y ejecutar aplicaciones Docker con múltiples contenedores.
- Este archivo es utilizado para describir la configuración de una aplicación Docker compuesta por varios contenedores, lo que permite definir varios servicios que trabajan en conjunto para proporcionar una aplicación completa.
- En el archivo docker-compose, se pueden especificar los contenedores que compondrán la aplicación, las imágenes de Docker que se utilizarán, las redes y volúmenes que se utilizarán para conectar los contenedores, así como cualquier otra configuración necesaria para la aplicación.
- Primero creamos el directorio de guacamole.

```
usuario@us: ~$ sudo mkdir /home/guacamole
usuario@us: ~$
```



- Descargamos unos ficheros desde github.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole$ sudo git clone "https://github.com/boschkundendienst/guacamole-docker-compose.git"
Cloning into 'guacamole-docker-compose'...
remote: Enumerating objects: 125, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 125 (delta 5), reused 9 (delta 1), pack-reused 108
Receiving objects: 100% (125/125), 44.21 KiB | 310.00 KiB/s, done.
Resolving deltas: 100% (58/58), done.
usuario@us:/home/guacamole$ ls
guacamole-docker-compose
usuario@us:/home/guacamole$
```

- Comprobamos que se han instalado todos los paquetes.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ ls -l
total 68
-rw-r--r-- 1 root root 3215 mar 28 09:12 CODE_OF_CONDUCT.md
-rw-r--r-- 1 root root 6157 mar 28 09:12 docker-compose.yml
-rw-r--r-- 1 root root 35149 mar 28 09:12 LICENSE
drwxr-xr-x 3 root root 4096 mar 28 09:12 nginx
-rwxr-xr-x 1 root root 795 mar 28 09:12 prepare.sh
-rw-r--r-- 1 root root 6406 mar 28 09:12 README.md
-rwxr-xr-x 1 root root 466 mar 28 09:12 reset.sh
usuario@us:/home/guacamole/guacamole-docker-compose$
```

- Ejecutamos el script de prepare.sh.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ sudo ./prepare.sh
Preparing folder init and creating ./init/initdb.sql
Unable to find image 'guacamole/guacamole:latest' locally
latest: Pulling from guacamole/guacamole
74ac377868f8: Pulling fs layer
b9cabe75b440: Pulling fs layer
221c4c5a69b6: Pulling fs layer
b62e7dc9dbe9: Pulling fs layer
16e4b40972e7: Pulling fs layer
2d45560e51e1: Pulling fs layer
2ffcafb29adf: Pulling fs layer
733f8914e1e0: Pulling fs layer
a7fa14ba0a01: Pulling fs layer
6a7b06fb5887: Pulling fs layer
ae30e2e0f5d1: Pulling fs layer
4da50330f55a: Pulling fs layer
2ffcafb29adf: Waiting
```



- Este script es utilizado para preparar un entorno para ejecutar la aplicación Guacamole en un contenedor Docker.
- El script primero verifica si el demonio de Docker está en ejecución y sale si no lo está. Luego, crea una carpeta llamada "init" y otra carpeta llamada "nginx/ssl". A continuación, se ejecuta un contenedor de Guacamole en modo "docker run --rm" con la opción "--postgres" para generar un archivo SQL de inicialización y lo guarda en el directorio "init" creado previamente.
- Por último, se generan certificados SSL auto-firmados utilizando el comando "openssl" y se guardan en la carpeta "nginx/ssl". Estos certificados SSL son necesarios para cifrar la comunicación entre los clientes y el servidor Guacamole.

11.4 REVISAR EL FICHERO DOCKER-COMPOSE

- El fichero docker-compose en Apache Guacamole se utiliza para definir y ejecutar una pila completa de contenedores Docker que se necesitan para ejecutar la aplicación Guacamole.
- El fichero docker-compose de Apache Guacamole incluye la configuración necesaria para lanzar varios contenedores que componen la aplicación, como un servidor de base de datos PostgreSQL, un servidor Guacamole, un servidor web Nginx y un contenedor de autenticación LDAP opcional.
- También se utiliza para definir la forma en que los contenedores se conectan entre sí y a la red host, así como para configurar la persistencia de los datos.



·El uso del fichero docker-compose de Apache Guacamole permite a los usuarios crear rápidamente una pila de contenedores Docker completa para ejecutar la aplicación Guacamole, sin tener que preocuparse por la configuración de cada contenedor individual.

```
usuario@us: /home/guacamol x + v
GNU nano 6.2 docker-compose.yml
# create a network 'guacnetwork_compose' in mode 'bridged'
networks:
  guacnetwork_compose:
    driver: bridge

# services
services:
  # guacd
  guacd:
    container_name: guacd_compose
    image: guacamole/guacd
    networks:
      guacnetwork_compose:
    restart: always
    volumes:
      - ./drive:/drive:rw
      - ./record:/record:rw
  # postgres
  postgres:
    container_name: postgres_guacamole_compose
    environment:
      PGDATA: /var/lib/postgresql/data/guacamole
      POSTGRES_DB: guacamole_db
      POSTGRES_PASSWORD: 'ChooseYourOwnPasswordHere1234'
      POSTGRES_USER: guacamole_user
```

11.5 ARRANCAR DOCKER-COMPOSE

·Arrancamos docker-compose con el siguiente comando.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ sudo docker-compose up -d
[sudo] password for usuario:
Creating network "guacamole-docker-compose_guacnetwork_compose" with driver "bridge"
Pulling guacd (guacamole/guacd:)...
latest: Pulling from guacamole/guacd
63b65145d645: Pull complete
3fb88a9d160c: Pull complete
e1f9afc18875: Downloading [=====] 25.05MB/48.59MB
bd4dfde57ab3: Download complete
b6b349982f16: Download complete
```



- Comprobamos el estado de docker-compose.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ sudo docker-compose ps
-----
Name                                Command                                State                                Ports
-----
guacamole_compose                   /opt/guacamole/bin/start.sh           Up                                0.0.0.0:49153->8080/tcp, :::49153->8080/tcp
guacd_compose                       /bin/sh -c /opt/guacamole/ ...       Up (health: starting)           4822/tcp
nginx_guacamole_compose             /docker-entrypoint.sh nginx ...       Up                                0.0.0.0:8443->443/tcp, :::8443->443/tcp, 80/tcp
postgres_guacamole_compose          docker-entrypoint.sh postgres         Up                                5432/tcp
usuario@us:/home/guacamole/guacamole-docker-compose$
```

11.6 CONFIGURAR GUACAMOLE

- A través de nuestro navegador, comprobamos el acceso.

<https://192.168.137.29:8443/#/>

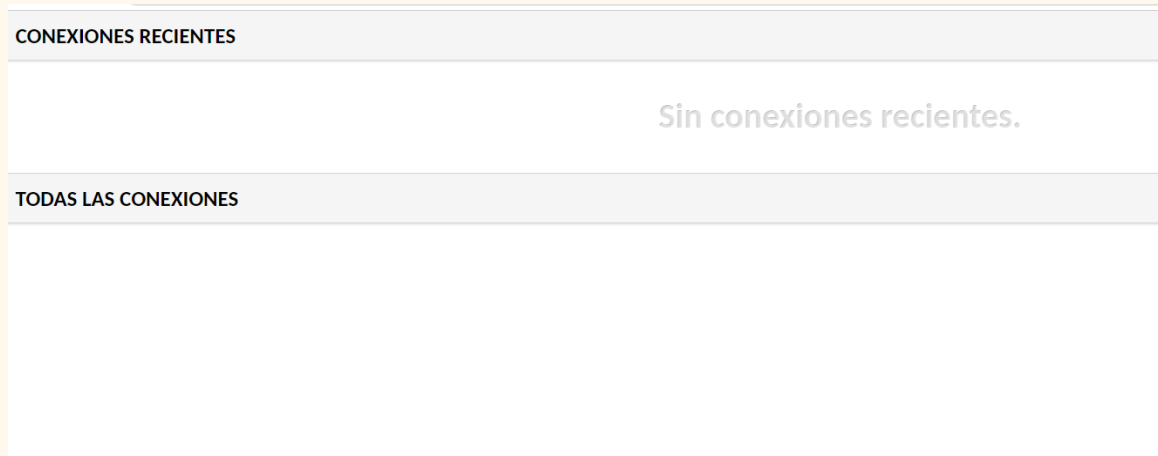


APACHE GUACAMOLE

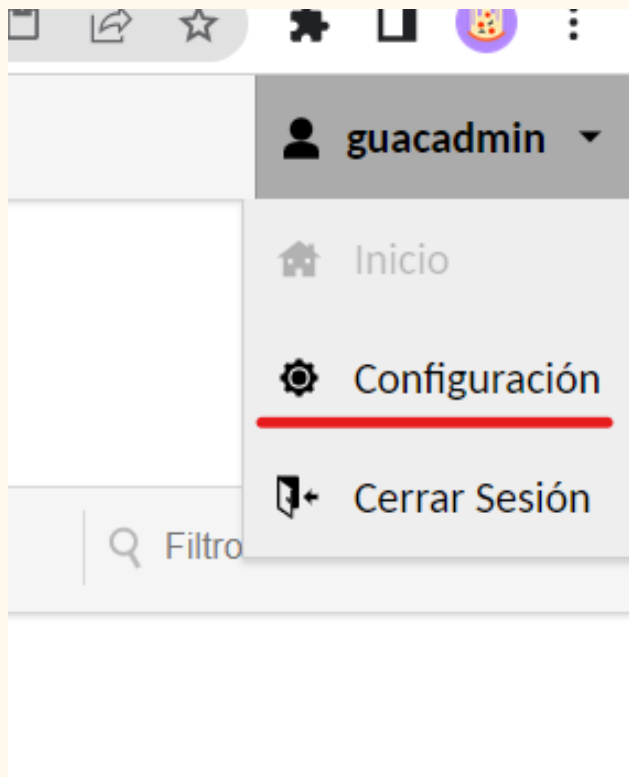
Iniciar Sesión



- Ponemos las credenciales y accedemos a la interfaz de guacamole.



- Ahora tenemos que clonar el usuario guacadmin.





·Clonamos el usuario.

PERMISOS

Administrar sistema:

✓

Crear nuevos usuarios:

✓

Crear nuevos grupo de usuarios:

✓

Crear nuevas conexiones:

✓

Crear nuevos grupos de conexión:

✓

Crear nuevos perfiles de compartir:

✓

Cambiar contraseña:

✓

CONEXIONES

Conexiones actuales

Todas las conexiones



Guardar

Clonar

Cancelar

Borrar

·Comprobamos que se ha creado el nuevo usuario.

Usuario ▾	
 guacadmin	
 usuario	



- Por motivos de seguridad, borramos el usuario guacadmin ya que venía por defecto al instalar guacamole.

11.7 AÑADIR VPN A DOCKER-COMPOSE

- Agregar una VPN a Docker Compose puede ser útil en varias situaciones, principalmente cuando se trabaja con múltiples contenedores que necesitan comunicarse entre sí a través de una red segura.
- Una VPN (Red Privada Virtual) permite crear una conexión segura y encriptada entre los contenedores de Docker Compose, evitando así que los datos sean expuestos en la red pública. Al agregar una VPN a Docker Compose, se puede mejorar la seguridad y privacidad de los datos y comunicaciones entre los diferentes contenedores.
- Además, si se trabaja con servicios que están alojados en diferentes regiones geográficas, una VPN puede ayudar a evitar las restricciones geográficas y a acceder a servicios que están bloqueados en ciertas regiones.
- Primero modificamos el fichero docker-compose.yml.
- Cambiamos la versión al principio del fichero.

```
version: '2.1'
# networks
# create a network 'guacnetwork_compose' in mode 'bridged'
networks:
  guacnetwork_compose:
    driver: bridge
```




·Al final del documento, añadimos las siguientes líneas para configurar nuestra VPN.

```
# WIREGUARD VPN

wireguard:
  image: lscr.io/linuxserver/wireguard
  container_name: wireguard
  cap_add:
    - NET_ADMIN
    - SYS_MODULE
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=Europe/London
    - SERVERURL=192.168.137.29
    - SERVERPORT=51820 #opcional
    - PEERDNS=auto #opcional
    - INTERNAL_SUBNET=10.13.13.0 #opcional
    - ALLOWEDIPS=0.0.0.0/0 #opcional
  volumes:
    - /opt/wireguard-server/config:/config
    - /lib/modules:/lib/modules
  networks:
    guacnetwork_compose:
  ports:
    - 51820:51820/udp
  sysctls:
    - net.ipv4.conf.all.src_valid_mark=1
  restart: unless-stopped
```

·**image:** lscr.io/linuxserver/wireguard: Esta línea indica la imagen de Docker que se utilizará para el contenedor del servicio WireGuard. En este caso, se utiliza la imagen de LinuxServer.io para WireGuard.

·**container_name:** Esta línea especifica el nombre del contenedor para este servicio.



·**cap_add**: Estas líneas añaden permisos adicionales al contenedor, en este caso, permitiendo que el contenedor pueda configurar la red (NET_ADMIN) y cargar módulos del kernel (SYS_MODULE).

·**Environment**: Estas líneas establecen variables de entorno que se utilizarán en el contenedor de WireGuard. Por ejemplo, se especifica la dirección IP del servidor (SERVERURL), el puerto (SERVERPORT) y otras configuraciones opcionales, como la subred interna (INTERNAL_SUBNET) y la configuración de DNS para los clientes (PEERDNS).

·**volumes**: Estas líneas especifican los directorios que se montarán en el contenedor de WireGuard como volúmenes. En este caso, se monta el directorio /opt/wireguard-server/config como /config en el contenedor, y el directorio de los módulos del kernel (/lib/modules) se monta como /lib/modules en el contenedor.

·**networks**: Esta línea especifica la red en la que se ejecutará el contenedor de WireGuard. En este caso, se utiliza una red llamada guacnetwork_compose.

·**ports**: Esta línea especifica el mapeo de puertos entre el host y el contenedor de WireGuard. En este caso, el puerto UDP 51820 del host se mapea al puerto UDP 51820 del contenedor.

·**sysctls**: Esta línea establece un valor del kernel de Linux para permitir el marcado válido de paquetes de origen (src_valid_mark) en la red.



·**restart:** Esta línea especifica la política de reinicio del contenedor en caso de fallo o detención. En este caso, el contenedor se reiniciará automáticamente a menos que se detenga de manera explícita mediante el comando docker stop.

·Ahora creamos el directorio de trabajo.

```
usuario@us: /home/guacamol x + v
usuario@us: /home/guacamole/guacamole-docker-compose$ sudo mkdir /opt/wireguard-server
usuario@us: /home/guacamole/guacamole-docker-compose$
```

·Le damos los permisos necesarios para la configuración.

```
usuario@us: /home/guacamol x + v
usuario@us: /home/guacamole/guacamole-docker-compose$ sudo chown usuario:usuario /opt/wireguard-server/
usuario@us: /home/guacamole/guacamole-docker-compose$
```

·Lanzamos de nuevo docker-compose.

```
usuario@us: /home/guacamole/guacamole-docker-compose$ sudo docker-compose up -d
Pulling wireguard (lscr.io/linuxserver/wireguard:...)
latest: Pulling from linuxserver/wireguard
38a8b9979b0c: Downloading [=====] 27.53MB/32.82MB
48f754c837b5: Download complete
e62f0384a36b: Download complete
776e56018534: Download complete
22883c3dda69: Download complete
996a56d8d5ee: Downloading [==>] 8.632MB/322.9MB
f10c0ae476f4: Download complete
```



- Comprobamos el estado de docker-compose.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ sudo docker-compose ps
-----
Name                                Command                                State                                Ports
-----
guacamole_compose                   /opt/guacamole/bin/start.sh           Up                                0.0.0.0:49153->8080/tcp, :::49153->8080/tcp
guacd_compose                       /bin/sh -c /opt/guacamole/ ...        Up (healthy)                    4822/tcp
nginx_guacamole_compose             /docker-entrypoint.sh nginx ...       Up                                0.0.0.0:8443->443/tcp, :::8443->443/tcp,
80/tcp
postgres_guacamole_compose          docker-entrypoint.sh postgres         Up                                5432/tcp
wireguard                           /init                                  Up                                0.0.0.0:51820->51820/udp, :::51820-
>51820/udp
usuario@us:/home/guacamole/guacamole-docker-compose$
```

- Ejecutamos el siguiente comando.

```
usuario@us: /home/guacamol x + v
usuario@us:/home/guacamole/guacamole-docker-compose$ sudo ip route add 10.13.13.0/24 via 172.18.0.6
usuario@us:/home/guacamole/guacamole-docker-compose$
```

· El comando "sudo ip route add 10.13.13.0/24 via 172.18.0.6" se utiliza para agregar una ruta a una red específica en un sistema Linux.

"**sudo**" se utiliza para ejecutar el comando como superusuario o como un usuario con privilegios elevados.

"**ip route**" se utiliza para administrar las tablas de enrutamiento del sistema.

"**add**" se utiliza para agregar una nueva ruta.

"**10.13.13.0/24**" es la red de destino que se agregará a la tabla de enrutamiento. "/24" especifica la máscara de subred y significa que



los primeros 24 bits de la dirección IP son la parte de la red y los últimos 8 bits son la parte del host.

"via 172.18.0.6" especifica la dirección IP del siguiente salto o el siguiente nodo que se utilizará para alcanzar la red de destino.

11.8 CREAR CONEXIÓN SSH UBUNTU-SERVER

·Vamos a crear una conexión vía ssh a nuestro Ubuntu Server desde Guacamole.

·Indicamos de nuestra conexión y el método que vamos a utilizar.

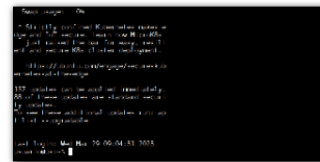
EDITAR CONEXIÓN	
Nombre:	<input type="text" value="UBUNTU SERVER"/>
Ubicación:	<input type="text" value="ROOT"/>
Protocolo:	<input type="text" value="SSH"/>

·También hay que poner la IP del servidor , el puerto que en este caso será el 22 (ssh) y la autenticación.

PARÁMETROS	
Red	
Nombre de Host:	<input type="text" value="192.168.137.151"/>
Puerto:	<input type="text" value="22"/>
Clave pública host (Base64):	<input type="text"/>
Autenticación	
Usuario:	<input type="text" value="usuario"/>
Contraseña:	<input type="password" value="....."/>
Llave Privada:	<input type="text"/>



- Guardamos la configuración.
- Comprobamos que podemos conectarnos remotamente.

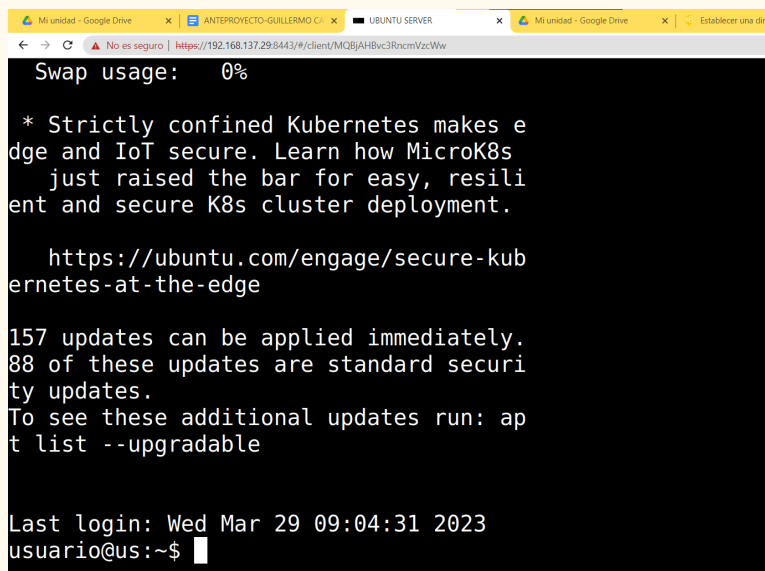


UBUNTU SERVER

TODAS LAS CONEXIONES

➤ UBUNTU SERVER

- Como podemos ver, se ha creado una conexión vía ssh a la máquina de Ubuntu Server.



- Vemos que nos conectamos a través de Internet.



11.9 CREAR CONEXIÓN RDP WINDOWS 10

· Ahora vamos a crear una conexión vía RDP con una máquina virtual de Windows 10.

EDITAR CONEXIÓN

Nombre:

WINDOWS 10

Ubicación:

ROOT

Protocolo:

RDP

· También hay que poner la IP de la máquina , el puerto que en este caso será el 33 (rdp) , la autenticación y algunos otros parámetros..

PARÁMETROS

Red

Nombre de Host:

192.168.137.106

Puerto:

3389

Autenticación

Usuario:

usuario

Contraseña:

Dominio:

Modo seguridad:

NLA (Autenticación de nivel de red)

Desactivar autenticación:

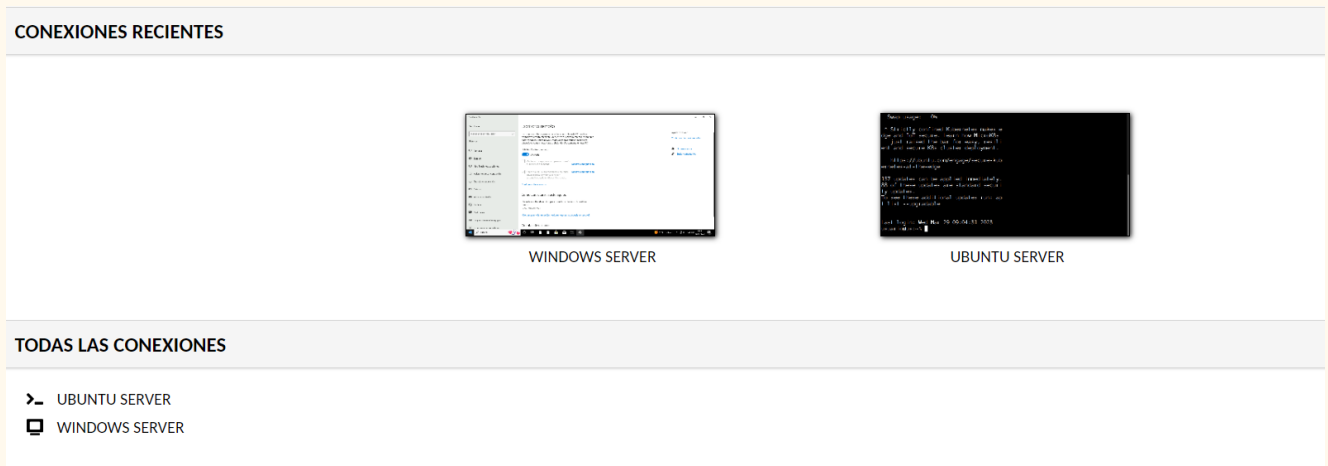
☐

Ignorar certificado del servidor:

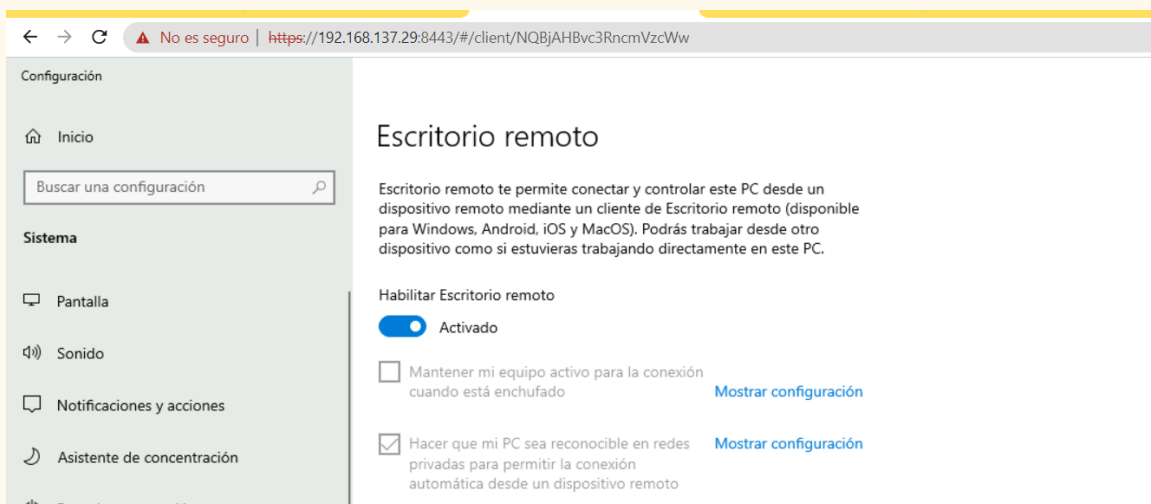
☒



- Guardamos cambios.
- Comprobamos que nos podemos conectar.



- Para que todo esto funcione, tenemos que tener el Escritorio Remoto de Windows activado.





11.10 CREAR CONEXIÓN VNC UBUNTU DESKTOP

·En primer lugar, actualizaremos los repositorios e instalamos wireguard con el siguiente comando.

```
Actividades Terminal 29 de mar 12:38 guille@guille-VirtualBox: ~
guille@guille-VirtualBox:~$ sudo apt-get update
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2.465 kB]
Des:6 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [574 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [803 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [420 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [275 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata
```

```
guille@guille-VirtualBox:~$ sudo apt-get install wireguard resolvconf -y
[sudo] contraseña para guille:
Lo sentimos, vuelva a intentarlo.
[sudo] contraseña para guille:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  wireguard-tools
Se instalarán los siguientes paquetes NUEVOS:
  resolvconf wireguard wireguard-tools
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 401 no actualizados
Se necesita descargar 141 kB de archivos.
Se utilizarán 544 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 wireguard-tools amd64 1.0.20200513-1~20.04.2 [83,3 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 wireguard
```



- Ahora nos tenemos que descargar un fichero que configuración en el Ubuntu Desktop.
- Para ello usamos el siguiente comando vía scp.

```
Actividades Terminal 30 de mar 09:27
guille@guille-VirtualBox: ~
guille@guille-VirtualBox:~$ scp usuario@192.168.137.29:/opt/wireguard-server/co
nfig/peer1/peer1.conf ./
usuario@192.168.137.29's password:
peer1.conf 100% 307 66.2KB/s 00:00
guille@guille-VirtualBox:~$
```

```
guille@guille-VirtualBox:~$ ls
Descargas Escritorio Música Plantillas Vídeos
Documentos Imágenes peer1.conf Público
guille@guille-VirtualBox:~$
```

- Le cambiamos el nombre y ubicación al archivo que nos hemos descargado.

```
Actividades Terminal 30 de mar 09:30
guille@guille-VirtualBox: ~
guille@guille-VirtualBox:~$ sudo mv peer1.conf /etc/wireguard/w0.conf
guille@guille-VirtualBox:~$
```



- Comprobamos el contenido del fichero.

```
GNU nano 4.8          w0.conf
[Interface]
Address = 10.13.13.2
PrivateKey = IAYEW72V1Wc8d+ckbj8h55YleQZXiXd0S1zjKDGbAkY=
ListenPort = 51820
DNS = 10.13.13.1

[Peer]
PublicKey = ke083z58Pecf6sdxow1by9piZ9Ek1o1YJSTBACQYAAA=
PresharedKey = UV22EQJyk82Q1ViLfGbUBn8Rd/ILtaCFZxrUvpeiEoQ=
Endpoint = 192.168.137.29:51820
AllowedIPs = 0.0.0.0/0
```

- El fichero es una configuración para una conexión VPN utilizando el protocolo WireGuard. La configuración se divide en dos secciones: Interface y Peer.
- La sección Interface incluye la dirección IP del servidor VPN, la clave privada del servidor, el puerto de escucha, y la dirección IP del servidor DNS.
- La sección Peer describe un par de claves, una pública y otra compartida. Además, incluye la dirección del servidor remoto con el que se establecerá la conexión VPN, y la red permitida a través de la conexión VPN. En este caso, AllowedIPs se establece en 0.0.0.0/0, lo que significa que todo el tráfico se enruta a través de la conexión VPN.



- Conectamos con el servidor.

```
guille@guille-VirtualBox:~$ sudo wg-quick up w0
[#] ip link add w0 type wireguard
[#] wg setconf w0 /dev/fd/63
[#] ip -4 address add 10.13.13.2 dev w0
[#] ip link set mtu 1420 up dev w0
[#] resolvconf -a tun.w0 -m 0 -x
[#] wg set w0 fwmark 51820
[#] ip -4 route add 0.0.0.0/0 dev w0 table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] iptables-restore -n
guille@guille-VirtualBox:~$
```

- Comprobamos el estado de la conexión.

```
guille@guille-VirtualBox:~$ sudo wg show
interface: w0
  public key: zcWixlSc3Bku0Y+5UI8faXyU6pXaPohGVV5gsRinJXQ=
  private key: (hidden)
  listening port: 51820
  fwmark: 0xca6c

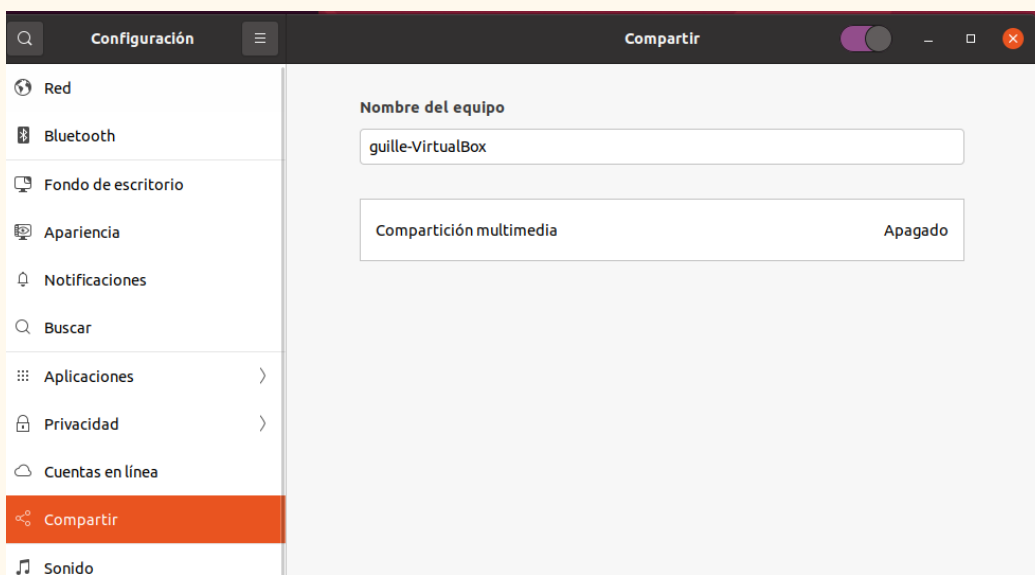
peer: ke083z58Pecf6sdxow1by9piZ9Ek1o1YJSTBACQYAAA=
  preshared key: (hidden)
  endpoint: 192.168.137.29:51820
  allowed ips: 0.0.0.0/0
  transfer: 0 B received, 3.90 KiB sent
guille@guille-VirtualBox:~$
```



- Comprobamos también la conexión desde el servidor de Guacamole.

```
usuario@us: ~  
usuario@us:~$ sudo docker exec -it wireguard wg  
interface: wg0  
  public key: ke083z58Pecf6sdxow1by9piZ9Ek1o1YJSTBACQYAAA=  
  private key: (hidden)  
  listening port: 51820  
  
peer: zcWixlSc3Bku0Y+5UI8faXyU6pXaPohGVV5gsRinJXQ=  
  preshared key: (hidden)  
  endpoint: 192.168.137.230:51820  
  allowed ips: 10.13.13.2/32  
  latest handshake: 16 minutes, 10 seconds ago  
  transfer: 3.08 KiB received, 1.64 KiB sent  
  
peer: S3ET4cp3F+iEuox+5i3iGzolRnDw8HJZ6T0St9HrIgQ=  
  preshared key: (hidden)  
  allowed ips: 10.13.13.3/32  
  
peer: ooHk+p5yXSmrq8dwJvQeyxqgyB9vLcUFiInt9ts0Yyw=  
  preshared key: (hidden)  
  allowed ips: 10.13.13.4/32  
usuario@us:~$
```

- En el Ubuntu Desktop activamos la opción de compartir.





- A continuación, creamos la conexión con las credenciales correctas

PARÁMETROS

Red

Nombre de Host:

Puerto:


Autenticación

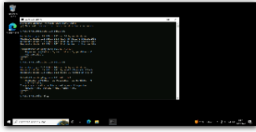
Usuario:


Contraseña: 

- Comprobamos que podemos acceder a la máquina de ubuntu-desktop.


CONEXIONES RECIENTES



UBUNTU-DESKTOP



WINDOWS 10


UBUNTU SERVER

TODAS LAS CONEXIONES

 UBUNTU SERVER

 UBUNTU-DESKTOP

 WINDOWS 10

Actualmente



- Como podemos comprobar, se conecta correctamente a nuestra máquina.

```
← → ↻ ⚠ No es seguro | https://192.168.1.143:8443/#/client/NABjAHBvc3RncmVzcWw
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Se pueden aplicar 316 actualizaciones de forma inmediata.
196 de estas son actualizaciones de seguridad estándares.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

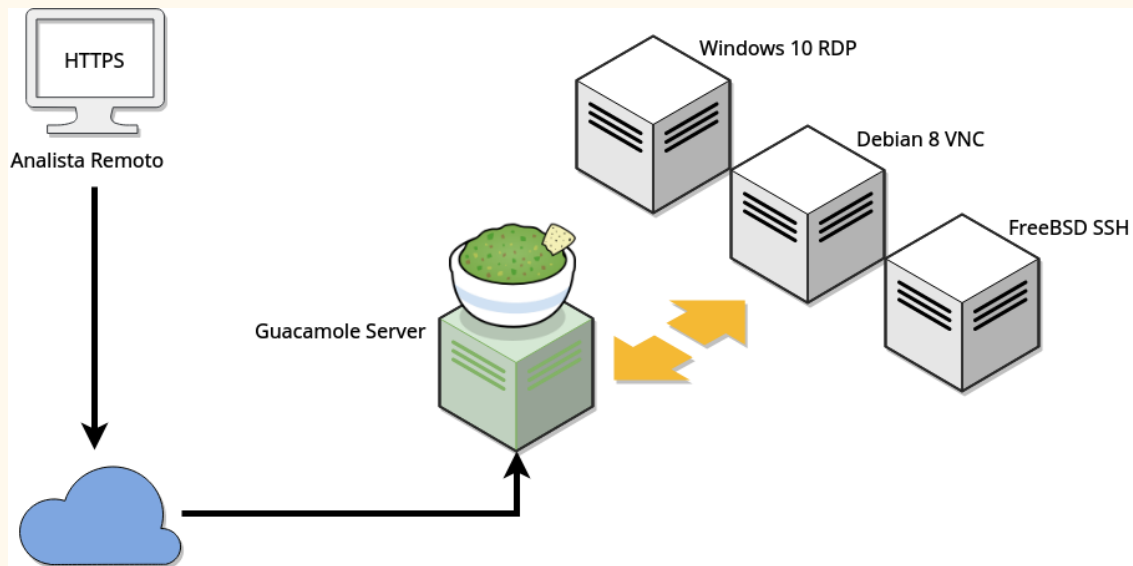
Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Wed Apr 12 12:21:21 2023 from 192.168.1.143
guille@guille-VirtualBox:~$
```

- Ya tendríamos instaladas y configuradas las 3 máquinas en Apache Guacamole.



12. ESQUEMA

·En este esquema podemos observar cómo funciona Apache Guacamole.



Cuando un cliente se conecta a tu servidor Apache Guacamole, se siguen los siguientes pasos:

- El cliente envía una solicitud de conexión al servidor Apache Guacamole a través de la red.
- Apache Guacamole recibe la solicitud y la redirige a su motor de protocolo, que es capaz de interpretar varios protocolos de escritorio remoto, como RDP, VNC y SSH.
- El motor de protocolo selecciona el protocolo adecuado para la solicitud y establece una conexión con el servidor remoto que aloja la sesión de escritorio remoto.



- Una vez establecida la conexión, el motor de protocolo recupera los datos de la sesión de escritorio remoto y los transmite al cliente a través de la conexión establecida.
- El cliente recibe los datos y los muestra en su pantalla.
- Cualquier entrada del usuario, como movimientos del mouse o pulsaciones de teclas, se envía de vuelta al motor de protocolo a través de la conexión establecida.
- El motor de protocolo transmite la entrada del usuario al servidor remoto, que la procesa y devuelve los resultados de la acción, como una actualización de la pantalla.
- Los resultados se envían al cliente a través de la conexión establecida y se muestran en la pantalla del cliente.

Estos pasos se repiten continuamente mientras la sesión de escritorio remoto está activa. Cuando se cierra la sesión, el motor de protocolo cierra la conexión con el servidor remoto y la conexión con el cliente se termina.





13. DIAGRAMA DE GANTT

Activity / Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Instalar docker y docker-compose	✓																			
Descargar fichero docker-compose		✓																		
Revisar el fichero docker-compose			✓	✓	✓															
Arrancar docker-compose						✓	✓													
Configurar Guacamole								✓	✓	✓	✓	✓	✓	✓	✓	✓				
Añadir VPN a docker-compose											✓	✓	✓	✓						
Crear conexión ssh a Ubuntu-Server y crear conexión rdp a Windows																	✓	✓	✓	
Crear conexión vnc a Ubuntu																✓	✓	✓	✓	✓

- He realizado el proyecto en unos 20 días.
- La parte más complicada como se puede comprobar fue configurar Apache Guacamole, ya que había bastantes ficheros de configuración.
- A pesar de ello y de algunos otros contratiempos, conseguí realizar mi proyecto en relativamente pocos días y estoy muy satisfecho por ello.



14. RIESGOS LABORALES

Como software, Apache Guacamole no tiene riesgos laborales por sí mismo. Sin embargo, si una persona usa Apache Guacamole de manera habitual en su trabajo, es importante tener en cuenta los posibles riesgos laborales asociados con su entorno de trabajo y la actividad laboral que realiza a través de Guacamole.

Algunos riesgos laborales que podrían estar asociados con el uso de Apache Guacamole incluyen:

- Fatiga visual:** Si la persona trabaja durante largos períodos de tiempo en Guacamole, podría experimentar fatiga visual debido a la pantalla del ordenador y el uso continuado de ratón y teclado.
- Problemas posturales:** Si la persona trabaja en una posición incómoda o durante largos períodos de tiempo sin tomar descansos, podría experimentar problemas posturales, como dolor de cuello, hombros o espalda.
- Estrés mental:** Si la persona utiliza Guacamole para realizar tareas que requieren alta concentración y precisión, podría experimentar estrés mental y emocional, lo que podría afectar su salud mental.
- Riesgos de seguridad de la red:** Si la persona trabaja en una red de ordenadores que no está bien protegida, podría estar expuesta a riesgos de seguridad, como ataques de malware, robo de datos, entre otros.



15. CONCLUSIONES

a. DEBILIDADES

Aunque Apache Guacamole es una solución sólida para el acceso remoto y el control de escritorio, también presenta algunas debilidades que es importante tener en cuenta:

Requisitos de infraestructura: Apache Guacamole requiere una infraestructura adecuada para su implementación, lo que puede incluir servidores y recursos de red adicionales. Esto puede generar costos adicionales y requerir experiencia técnica para su configuración y mantenimiento.

Dependencia de la conexión a Internet: Como Guacamole es una solución basada en la web, su rendimiento y accesibilidad dependen en gran medida de la calidad y la velocidad de la conexión a Internet. Si la conexión es lenta o inestable, podría afectar negativamente la experiencia del usuario y la productividad.

Limitaciones de rendimiento: Aunque Guacamole está diseñado para funcionar de manera eficiente incluso en entornos de red de baja velocidad, es posible que experimente ciertas limitaciones en términos de rendimiento, especialmente cuando se trata de transferencia de archivos grandes o ejecución de aplicaciones de alto rendimiento gráfico.



Complejidad en la configuración inicial: Configurar y poner en marcha Guacamole puede requerir cierto nivel de conocimientos técnicos, especialmente para integrarlo con sistemas de autenticación existentes o para habilitar protocolos específicos. Esto puede representar un desafío para los usuarios menos experimentados.

Soporte y comunidad: Aunque Guacamole cuenta con una comunidad de usuarios y desarrolladores activa, puede ser menos conocido y tener menos recursos disponibles en comparación con soluciones de acceso remoto más establecidas y comerciales. Esto puede afectar la disponibilidad de soporte técnico y documentación en algunos casos.

A pesar de estas debilidades, Apache Guacamole sigue siendo una solución popular y efectiva para el acceso remoto y el control de escritorio, y muchas organizaciones encuentran que sus beneficios superan con creces sus limitaciones.





b. AMENAZAS

Si bien Apache Guacamole es una solución robusta para el acceso remoto y el control de escritorio, también existen algunas amenazas potenciales que se deben tener en cuenta:

Vulnerabilidades de seguridad: Al ser una herramienta de acceso remoto, Apache Guacamole puede presentar vulnerabilidades de seguridad si no se implementa y configura correctamente. Las vulnerabilidades podrían ser explotadas por actores maliciosos para acceder a sistemas o robar información confidencial. Es importante seguir buenas prácticas de seguridad, como aplicar parches y actualizaciones regulares, utilizar autenticación sólida y mantener las configuraciones adecuadas de firewall.

Acceso no autorizado: Si las credenciales de acceso a Apache Guacamole son comprometidas o si se utilizan contraseñas débiles, existe el riesgo de que personas no autorizadas obtengan acceso a los escritorios o aplicaciones remotas. Esto podría conducir a un acceso no autorizado a sistemas críticos o a la divulgación de información sensible.

Ataques de fuerza bruta: Los atacantes pueden intentar descubrir las credenciales de acceso de Apache Guacamole mediante ataques de fuerza bruta, donde prueban una gran cantidad de combinaciones de nombres de usuario y contraseñas. Para mitigar este riesgo, se deben implementar medidas de seguridad adecuadas, como políticas de bloqueo de cuentas después de varios intentos fallidos de inicio de sesión.



Problemas de compatibilidad: Dependiendo de los sistemas operativos y aplicaciones específicas utilizados en la infraestructura, podría haber problemas de compatibilidad al utilizar Apache Guacamole. Algunas funciones o características pueden no ser completamente compatibles o requerir configuraciones adicionales para funcionar correctamente.

Dependencia de terceros: Apache Guacamole utiliza una variedad de componentes y bibliotecas de terceros para su funcionamiento. Si alguno de estos componentes presenta una vulnerabilidad de seguridad o deja de recibir mantenimiento, podría afectar la seguridad y la funcionalidad general de Guacamole.

Para mitigar estas amenazas, es fundamental mantener Apache Guacamole actualizado con las últimas versiones y parches de seguridad, seguir las mejores prácticas de seguridad, implementar autenticación sólida y realizar auditorías regulares de seguridad. Además, es importante contar con personal capacitado y experto en la implementación y administración de Guacamole para garantizar una configuración segura y eficiente.





c. **FORTALEZAS**

Apache Guacamole ofrece varias fortalezas como solución de acceso remoto y control de escritorio:

Acceso desde cualquier lugar: Guacamole permite acceder a escritorios y aplicaciones de forma remota desde cualquier dispositivo con un navegador web, lo que facilita la colaboración y el trabajo remoto. Los usuarios pueden acceder a sus recursos desde cualquier lugar, lo que aumenta la flexibilidad y la productividad.

Arquitectura basada en web: Guacamole utiliza una arquitectura basada en web, lo que significa que no se requiere la instalación de software adicional en los dispositivos de los usuarios. Esto simplifica la implementación y el uso, ya que solo se necesita un navegador web compatible.

Amplia compatibilidad de protocolos: Guacamole es compatible con varios protocolos de escritorio remoto, como RDP, VNC y SSH. Esto permite acceder a una amplia gama de sistemas y plataformas, lo que lo hace adecuado para entornos heterogéneos.

Integración con sistemas existentes: Guacamole se puede integrar con sistemas de autenticación existentes, como LDAP o Active Directory, lo que facilita la administración de usuarios y la implementación de políticas de seguridad consistentes en la organización.



Seguridad mejorada: Guacamole proporciona una capa adicional de seguridad al permitir el acceso remoto a través de conexiones encriptadas. Esto ayuda a proteger los datos y la información confidencial de accesos no autorizados.

Escalabilidad y rendimiento: Guacamole es escalable y se puede implementar en arquitecturas de nube, lo que permite adaptarse a las necesidades cambiantes de la organización. Además, su diseño ligero y optimizado garantiza un rendimiento eficiente incluso en entornos de red de baja velocidad.

Código abierto: Guacamole es una solución de código abierto, lo que significa que está respaldado por una comunidad activa de desarrolladores y usuarios. Esto proporciona una mayor transparencia, flexibilidad y posibilidad de personalización para adaptarse a las necesidades específicas de cada organización.

En general, Apache Guacamole es una solución confiable y efectiva para el acceso remoto y el control de escritorio. Sus fortalezas en términos de accesibilidad, seguridad, compatibilidad y escalabilidad lo convierten en una opción popular para muchas organizaciones.





d. OPORTUNIDADES

Apache Guacamole ofrece varias oportunidades para mejorar la accesibilidad y la eficiencia en el acceso remoto y el control de escritorio. Algunas de las oportunidades que ofrece Guacamole incluyen:

Trabajo remoto y colaboración: Guacamole permite a las organizaciones adoptar modelos de trabajo remoto y facilitar la colaboración entre equipos distribuidos. Los empleados pueden acceder a sus escritorios y aplicaciones desde cualquier lugar, lo que promueve la flexibilidad y la productividad.

Reducción de costos: Al ser una solución de código abierto, Guacamole ofrece una alternativa rentable a las soluciones propietarias de acceso remoto. No se requiere la compra de licencias y se pueden utilizar recursos de hardware existentes, lo que ayuda a reducir los costos de implementación y mantenimiento.

Acceso universal desde cualquier dispositivo: Guacamole es compatible con cualquier dispositivo que tenga un navegador web, lo que facilita el acceso remoto desde computadoras de escritorio, laptops, tabletas y teléfonos móviles. Esto brinda a los usuarios la libertad de acceder a sus recursos desde el dispositivo que les resulte más conveniente.

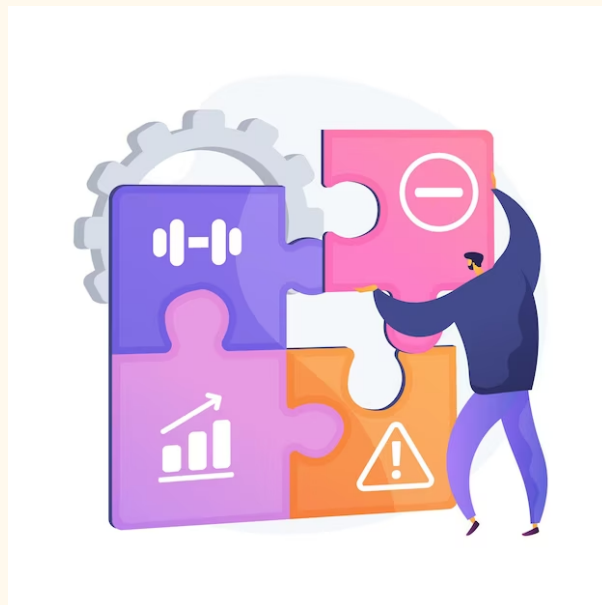
Integración con sistemas existentes: Guacamole se puede integrar con sistemas de autenticación existentes, como LDAP o Active Directory, lo que simplifica la gestión de usuarios y garantiza una autenticación segura y consistente.



Mejora de la seguridad: Guacamole ofrece características de seguridad, como la encriptación de extremo a extremo y el soporte para autenticación de dos factores, lo que mejora la protección de los datos y la información confidencial durante el acceso remoto. Esto ayuda a cumplir con los estándares de seguridad y cumplimiento.

Experiencia de usuario optimizada: Guacamole proporciona una interfaz de usuario intuitiva y fácil de usar, lo que facilita la navegación y el acceso a los escritorios y aplicaciones remotos. Además, su diseño ligero y optimizado garantiza un rendimiento fluido y una experiencia de usuario satisfactoria.

Comunidad de desarrollo activa: Guacamole cuenta con una comunidad de desarrollo activa y creciente, lo que significa que se están realizando constantemente mejoras y nuevas características. Esto brinda oportunidades para personalizar y adaptar la solución a las necesidades específicas de cada organización.





e. AMPLIACIONES FUTURAS

Apache Guacamole tiene un gran potencial para futuras expansiones y mejoras. Algunas áreas en las que podrían realizarse ampliaciones futuras incluyen:

Soporte para más protocolos: Actualmente, Guacamole es compatible con protocolos como RDP, VNC y SSH. Sin embargo, podría haber oportunidades para agregar soporte para otros protocolos de acceso remoto, lo que permitiría a los usuarios acceder a una mayor variedad de sistemas y plataformas.

Mejoras en la experiencia del usuario: A medida que la tecnología y las expectativas de los usuarios evolucionan, Guacamole podría beneficiarse de mejoras en la interfaz de usuario y la experiencia del usuario. Esto podría incluir opciones de personalización más avanzadas, una interfaz más intuitiva y un flujo de trabajo mejorado.

Funcionalidades de colaboración en tiempo real: Actualmente, Guacamole se centra principalmente en el acceso remoto individual a escritorios y aplicaciones. Sin embargo, podría haber oportunidades para agregar funcionalidades de colaboración en tiempo real, como la capacidad de compartir pantallas o trabajar simultáneamente en un escritorio remoto.

Capacidades de grabación y auditoría: Agregar capacidades de grabación y auditoría a Guacamole permitiría a las organizaciones registrar y auditar las actividades de acceso remoto. Esto puede ser valioso para fines de cumplimiento, seguimiento de incidentes y análisis de seguridad.



Integración con herramientas de gestión de TI: Ampliar la integración con otras herramientas de gestión de TI, como sistemas de ticketing, monitoreo o administración de activos, podría mejorar la administración y la eficiencia operativa en entornos empresariales.

Mejoras en la seguridad: A medida que surgen nuevas amenazas y se desarrollan tecnologías de seguridad, Guacamole podría beneficiarse de mejoras en la seguridad. Esto podría incluir el fortalecimiento de la autenticación, la implementación de tecnologías de cifrado más fuertes o la adopción de prácticas de seguridad avanzadas.

Soporte para entornos de contenedores y orquestación: Dado el creciente interés en la contenerización y la orquestación de aplicaciones, Guacamole podría explorar opciones para soportar entornos de contenedores y orquestación, lo que facilitaría su implementación y gestión en infraestructuras modernas.

Estas son solo algunas de las posibles ampliaciones futuras que podrían beneficiar a Apache Guacamole. La dirección y las prioridades específicas dependerán de las necesidades y demandas de la comunidad de usuarios, así como de los avances tecnológicos y las tendencias del mercado.



f. ESCABILIDAD

Apache Guacamole es una solución escalable que se adapta a diferentes necesidades de implementación y demandas de usuarios. Su arquitectura basada en web y su enfoque ligero y eficiente permiten una escalabilidad eficaz. Algunos aspectos relevantes sobre la escalabilidad de Guacamole son los siguientes:

Arquitectura de cliente-servidor: Guacamole se basa en una arquitectura cliente-servidor, lo que permite una distribución eficiente de la carga de trabajo. El servidor de Guacamole puede gestionar múltiples sesiones de usuario simultáneamente, lo que facilita el soporte para un número creciente de usuarios.

Escalabilidad horizontal: Guacamole puede ser escalado horizontalmente al agregar más instancias del servidor Guacamole según sea necesario. Esto permite distribuir la carga de trabajo y manejar un mayor número de conexiones y usuarios. Mediante la configuración de equilibradores de carga y la implementación de clústeres, es posible lograr una escalabilidad horizontal efectiva.

Soporte para infraestructuras en la nube: Guacamole es compatible con infraestructuras en la nube, lo que facilita la escalabilidad y el despliegue en entornos dinámicos y de rápida expansión. Puede ser implementado en plataformas de nube como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform, aprovechando sus características de escalabilidad automática y elástica.



Eficiencia en recursos: Guacamole está diseñado para ser ligero y eficiente, lo que permite un uso eficaz de los recursos del sistema. Esto es especialmente importante al escalar la solución, ya que minimiza la carga en los servidores y reduce los requisitos de hardware.

Gestión centralizada: Guacamole permite la gestión centralizada de usuarios, conexiones y recursos. Esto simplifica la administración y el escalado al proporcionar una única interfaz para controlar y gestionar todas las sesiones y usuarios conectados.

Integración con herramientas de gestión: Guacamole se puede integrar con herramientas de gestión de TI existentes, como sistemas de orquestación y automatización, lo que facilita el escalado y la gestión de forma más eficiente y coherente.





16. BIBLIOGRAFÍA

<https://guacamole.apache.org/>

<https://help.clouding.io/hc/es/articles/360019699179-C%C3%B3mo-instalar-y-configurar-Apache-Guacamole>

<https://www.ochobitshacenunbyte.com/2020/10/21/administracion-remota-con-apache-guacamole/>

<https://firstcommit.dev/2022/04/02/guacamole/>