DOCUMENTACION FINAL TFG

PROYECTO DE ADMINISTRACION DE SISTEMAS INFORMATICOS EN RED

6 DE JUNIO DE 2025

IVAN HUMARA MIRANDA – 62204160F

Tabla de contenido

Tabla de contenido tablas

Tabla de contenido imágenes

Resumen

Mi proyecto presenta un sistema de escritorio remoto en entornos de Linux llamado ThinLinc, que permite a los usuarios acceder a entornos de trabajo de manera segura y eficiente. La solución se implementa con la gestión centralizada de usuarios a través de LDAP, facilitando el control y la administración de permisos, y garantizando que solo los usuarios autorizados puedan utilizar el servicio.

Además, el acceso al servicio se restringe usando una VPN, lo que agrega una capa adicional de seguridad para proteger la transmisión de datos en entornos de red no controlados. Este proyecto resuelve el problema de la seguridad en el acceso remoto, al ofrecer una solución unificada y simplificada.

La implementación de ThinLinc permite la conexión remota con un rendimiento óptimo, mientras que LDAP centraliza la administración de usuarios y políticas de acceso. La restricción a través de VPN asegura que la comunicación se realice en un entorno seguro, reduciendo significativamente el riesgo de intrusiones y de ataques informáticos hacia el servicio. En conjunto, esta arquitectura favorece la continuidad operativa, optimiza la gestión de recursos y se adapta a las exigencias de las organizaciones en términos de seguridad, eficiencia y flexibilidad.

Todo esto montado en los servidores de AWS

Palabras clave

Estos son las palabras clave de mi proyecto

- Ubuntu
- Cendio
- ThinLinc
- ThinLinc Maestro
- ThinLinc Agente
- LDAP
- VPN
- WireGuard
- Alta Disponibilidad
- Balanceo de Carga (NBL)
- AWS
- Máquina Virtual

Introducción

Durante mi experiencia diaria en entornos académicos y laborales donde se utilizan sistemas Linux, observé una necesidad, la posibilidad de acceder de forma remota y segura a entornos de trabajo, sin comprometer la integridad de los datos ni la disponibilidad de los servicios. En muchos casos, los usuarios necesitaban acceder a sus escritorios desde distintos sitios, ya fuera desde casa, otras oficinas o incluso durante viajes, y las soluciones disponibles eran poco seguras, difíciles de gestionar o simplemente ineficientes en términos de rendimiento.

A raíz de este problema, me surgió la idea de utilizar un sistema de escritorio remoto basado en entornos de Linux llamado ThinLinc, un servicio desarrollado por Cendio, que permite conexiones rápidas, seguras y estables. La elección de ThinLinc se justifica por su enfoque en el rendimiento, la seguridad y su facilidad de integración con herramientas como LDAP y VPN, elementos clave para la centralización de usuarios y la protección de la red.

El proyecto no está planteado como una empresa comercial, sino como una solución técnica que puede ser implementada en organizaciones que requieran acceso remoto seguro, como instituciones educativas, empresas tecnológicas o departamentos de TI. La inclusión de herramientas como WireGuard para la VPN, scripts automatizados para facilitar la gestión, y la posibilidad de desplegar el sistema en la nube con AWS utilizando máquinas virtuales, responde a la necesidad de una solución escalable, segura y fácil de administrar.

En resumen, este proyecto nace como respuesta a un problema real de acceso remoto poco seguros en entornos Linux, y propone una arquitectura unificada que garantiza alta disponibilidad, balanceo de carga y una administración eficiente, todo ello orientado a mejorar la continuidad operativa de cualquier organización.

Objetivos

Objetivo General:

Desarrollar e implementar un sistema de acceso remoto seguro y eficiente para entornos de trabajo Linux, utilizando ThinLinc para la conexión remota, LDAP para la gestión centralizada de usuarios y VPN para asegurar la transmisión de datos, garantizando la integridad, disponibilidad y confidencialidad de la información.

Objetivos Específicos:

1. Implementar ThinLinc para ofrecer una solución de escritorio remoto que permita el acceso a entornos de Linux, con alta disponibilidad y balanceo de carga.
2. Integrar LDAP para gestionar de forma centralizada los usuarios y las políticas de acceso del servidor ThinLinc, asegurando un control adecuado sobre los permisos y accesos a los recursos de la red.
3. Configurar una VPN para restringir el acceso al servicio de escritorio remoto, agregando una capa adicional de seguridad y protegiendo la transmisión de datos en redes no controladas.
4. Optimizar la seguridad del sistema mediante la implementación de prácticas recomendadas para la protección contra intrusiones y ataques informáticos, garantizando la confidencialidad e integridad de los datos transmitidos.
5. Facilitar la administración de usuarios e instalación de los servicios, permitiendo a los administradores gestionar la creación de usuarios e instalación de los servicios de manera automatizada con la utilización de scripts, minimizando el riesgo de errores humanos.

Análisis del contexto

Análisis del contexto

Después de hacer una larga búsqueda sobre servicios similares he llegado a la conclusión que la principal competencia es:

- **NoMachine:** Ofrece acceso remoto para diversos contenidos, incluyendo audio y video. La empresa que provee el servicio es NoMachine S. El coste es de 40€ por dispositivo.
- **VNC Connect:** Ofrece acceso remoto desde computadoras de escritorio o dispositivos móviles. La empresa que provee el servicio es RealVNC.
- **X2Go:** Proporciona un servicio de acceso remoto de código abierto para Linux que utiliza el protocolo NX. Este servicio no es provisto por ninguna empresa en particular. Es gratuito.
- **mRemoteNG:** Ofrece acceso remoto multiprotocolo con pestañas. Este servicio no es provisto por ninguna empresa en particular. Es Gratuito.
- **Chrome Remote Desktop:** Permite a los usuarios acceder de formar remota a través del navegador Chrome. Es gratuito.

Análisis DAFO

Fortalezas

1. Especialización en Linux: Mejor experiencia de usuario para escritorios Linux que muchas soluciones genéricas (Citrix, VMware, etc).
2. Compatibilidad multiplataforma: Funciona en Windows, macOS, Linux y navegadores web.
3. Seguridad sólida: Basado en SSH, cifrado de extremo a extremo, autenticación fuerte.
4. Bajo consumo de recursos: Funciona bien en conexiones lentas y hardware modesto.
5. Empresa europea: Cumple con estándares de privacidad como GDPR.

Debilidades

1. Menor reconocimiento de marca frente a gigantes como Citrix, Microsoft o VMware.
2. Interfaz técnica: No es la más amigable para usuarios no técnicos o sin experiencia en Linux.
3. Falta de soporte directo a Windows como host: Solo clientes Windows, no servidores.
4. Menos funcionalidades empresariales integradas (auditoría avanzada, balanceo de carga automático, etc.).
5. Dependencia de entornos Linux: Lo que puede limitar su adopción en entornos mixtos.
6. Versión gratuita hasta 10 usuarios: Ideal para pequeñas organizaciones, pruebas y entornos educativos.

Oportunidades

Creciente adopción de Linux en entornos de desarrollo y educación.

1. Demanda por soluciones seguras de trabajo remoto sigue en aumento.
2. Auge del software open source y ético: muchas empresas buscan alternativas a grandes corporaciones.
3. Espacio para integrarse con entornos cloud (AWS, Azure, etc.) para mayor escalabilidad.

4. Mercado educativo y de investigación poco atendido por soluciones comerciales grandes.

Amenazas

1. Competencia de soluciones gratuitas o más conocidas como X2Go, Guacamole o VNC.
2. Empresas muy reconocidas como Microsoft, Citrix y VMware ofrecen soluciones con muchos recursos e integraciones.
3. Cambio de tendencias tecnológicas hacia escritorios totalmente web o aplicaciones SaaS.
4. Riesgo de estancamiento si no se expande a más plataformas o añade funcionalidades colaborativas.
5. Proyectos open source similares y gratuitos que pueden cubrir necesidades básicas.

Estado del arte

Estudio de dominio de aplicación del proyecto

En los últimos años, los servicios de acceso remoto han adquirido una gran relevancia debido al crecimiento del teletrabajo, la virtualización de escritorios y la necesidad de acceder a sistemas desde múltiples ubicaciones y dispositivos. Esta forma de trabajar se ha visto reforzada tras la pandemia, impulsando a muchas instituciones y empresas a implementar soluciones seguras y eficientes para trabajar de forma remota.

Para lograr esta forma de trabajar se utilizan las siguientes tecnologías y dispositivos:

- Protocolos de acceso como RDP, VNC y SSH
- Seguridad utilizando cifrado de extremo a extremo, autentificación por claves, integración de servidores LDAP y Kerberos
- Sesiones persistentes
- Clientes multiplataforma ya puede ser desde Windows, Linux a Android, iOS hasta incluso vía navegadores web

Problemas identificados

Desde que se utiliza esta tecnología del acceso remoto ha habido una serie de problemas recurrentes que afectan a los usuarios como a los propios administradores de sistemas.

- **Latencia y rendimiento**, a veces en las sesiones de acceso remoto se vuelven lentas pudiendo dar tirones o respuestas lentas del teclado y ratón.
- **Problemas de compatibilidad** con ciertos dispositivos como los USB o los escáneres e impresoras y también problemas con los sistemas ya que no todos soportan el escritorio remoto.
- **Dificultad en las configuraciones y el mantenimiento**, esto pasa cuando el usuario no tiene experiencia con estos servicios.
- **Experiencias de usuarios deficientes** ya que no todos los servicios proporcionan una buena calidad grafica.
- **Ausencia de sesiones persistentes** no todos los servicios ofrecen las sesiones persistentes es decir si te desconectas pierdes la sesión.
- **Problemas de seguridad** hay algunas tecnologías que no cifran correctamente la conexión.

Todos estos problemas en mayor o menor medida me afectaran en la realización del proyecto, aunque hay algunos que los puedo mitigar como los problemas de seguridad, ya que al usar una VPN es una barrera de seguridad adicional para poder utilizar el servicio, o la ausencia de sesiones la puedo eliminar ya que uso un servicio que si dispone de ello (ThinLinc), por el otro lado hay algunos problemas que no podre evitar, como la latencia y el rendimiento ya que al no estar directamente en la maquina real del usuario siempre tendrá esa ralentización por culpa de la conexión .

Innovación

El uso de ThinLinc como solución principal en mi proyecto es para mejorar y optimizar lo que ya existe, debido a que ThinLinc se centra en un mayor nivel de seguridad con el uso de SSH, y no crear túneles manuales con VNC, también permitir sesiones persistentes, esta optimizado específicamente para entornos Linux pero a la vez es multiplataforma ya que se puede utilizar en una variedad de SO, incluso en navegadores web, y por ultimo porque es un modelo accesible ya que hasta no llegar a más de 10 usuarios es gratuito.

Todo esto lo hace ideal para ser usados en entornos laborales o educativos que requieran seguridad y fiabilidad sin gastar una gran cantidad de dinero.

Diseño

Diagrama de arquitectura en AWS

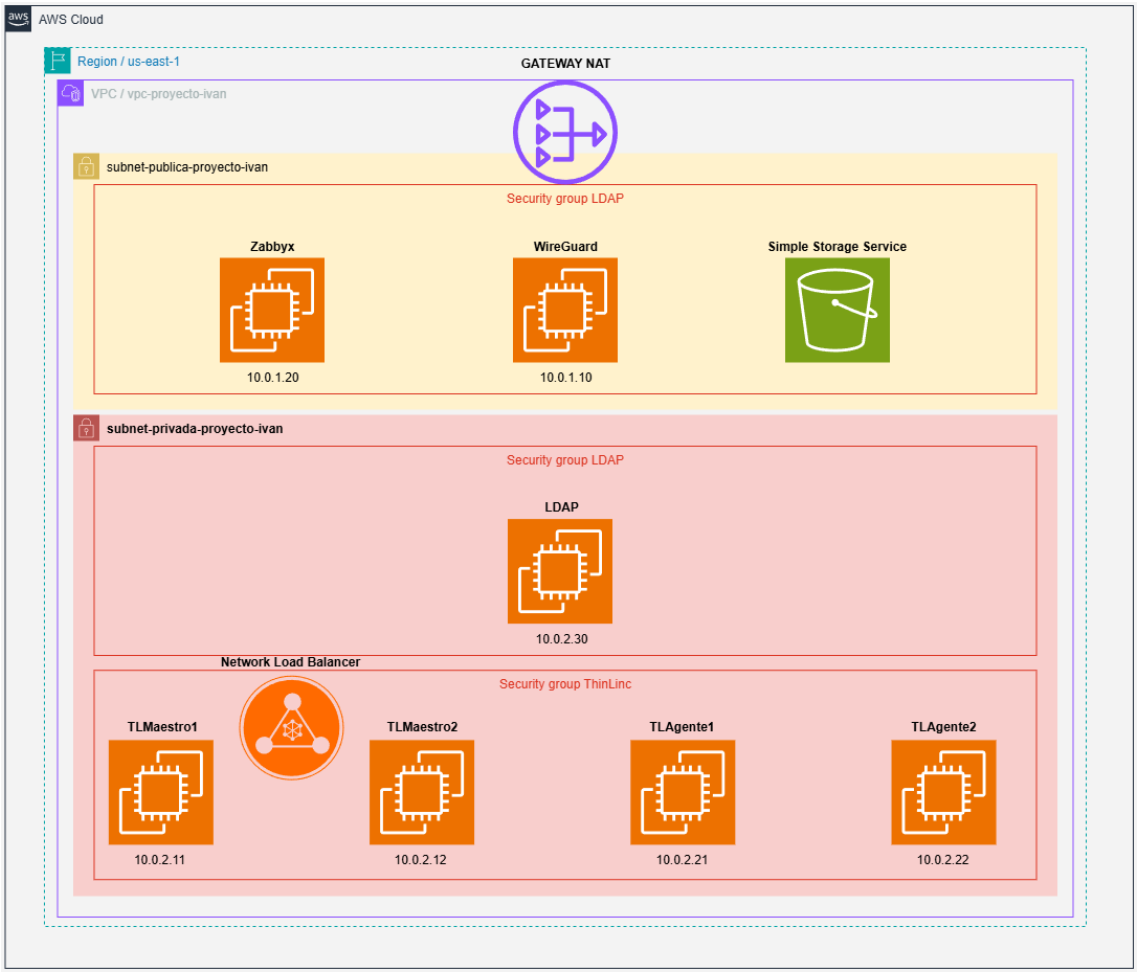


Ilustración 1: Diagrama de red

Listado de tecnologías

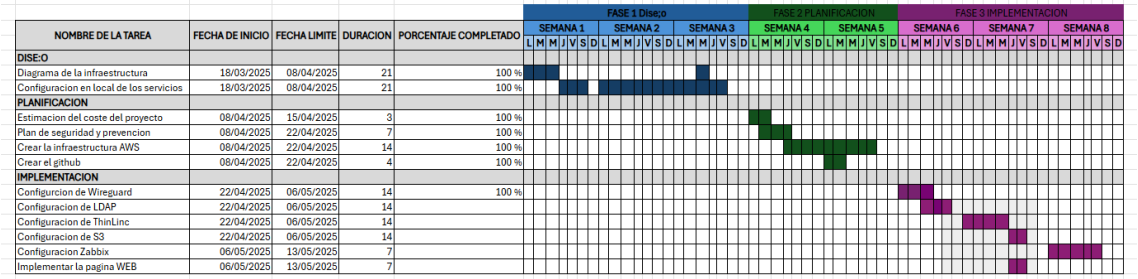
ThinLinc, LDAP, Wireguard, Zabbix

Infraestructura

AWS EC2, S3, Network Load Balancer, Gateway NAT

Planificación

Diagrama de GANT hasta día 1/06/2025



NOMBRE DE LA TAREA	FECHA DE INICIO	FECHA LIMITE	DURACION	PORCENTAJE COMPLETADO
DISE:O				
Diagrama de la infraestructura	18/03/2025	08/04/2025	21	100 %
Configuracion en local de los servicios	18/03/2025	08/04/2025	21	100 %
PLANIFICACION				
Estimacion del coste del proyecto	08/04/2025	15/04/2025	3	100 %
Plan de seguridad y prevencion	08/04/2025	22/04/2025	7	100 %
Crear la infraestructura AWS	08/04/2025	22/04/2025	14	100 %
Crear el github	08/04/2025	22/04/2025	4	100 %
IMPLEMENTACION				
Configurcion de Wireguard	22/04/2025	06/05/2025	14	100 %
Configuracion de LDAP	22/04/2025	06/05/2025	14	
Configuracion de ThinLinc	22/04/2025	06/05/2025	14	
Configuracion de S3	22/04/2025	06/05/2025	14	
Configuracion Zabbix	06/05/2025	13/05/2025	7	
Implementar la pagina WEB	06/05/2025	13/05/2025	7	

Ilustración 2: Diagrama de GANT

Definición de actividades y tareas

1. Diseño
    a. Diagrama de la infraestructura
        i. Definir la VPC, las zonas de disponibilidad, las subredes y los grupos de seguridad
    b. Configuración en local de los servicios
        i. Realizar las instalaciones y configuraciones de los servicios en proxmox
2. Planificación
    a. Estimación del coste del proyecto
        i. Hacer una estimación del coste del laboratorio de AWS
    b. Plan de seguridad y prevención
        i. Configurar los grupos de seguridad
        ii. Visualiza los riesgos y prevenirlos
3. Implementación
    a. Creación de la Infraestructura
    b. Configuración de VPN
    c. Configuración de LDAP
    d. Configuración de ThinLinc
        i. Creación de usuarios
    e. Configuración de S3
        i. Implementar la página WEB
    f. Configuración de Zabbix
4. Comprobación
    a. Pruebas de funcionamiento
5. Presentación
    a. Realizar la presentación del proyecto

Identificación de riesgos y prevención

RIESGO	Prevención
Fallos en los scripts	Probar los scripts de forma local antes de meterlos al laboratorio
Gasto total de saldo en AWS	No apurar el saldo de los laboratorios de AWS y cambiar con tiempo
Falta de tiempo	Priorizar las tareas importantes como la VPN y la infraestructura ya que sin ellas no funcionaría nada

Tabla 1: Identificación de riesgos y prevención

Cálculo de coste del proyecto

Este es el coste promedio que cuesta mantener el servicio en AWS

Servicio	Coste mensual estimado cada mes
EC2 – 7 instancias	46.52€
S3	1.05€
IP elásticas	3.15€
Gateway NAT	28,36€
TOTAL	**79,08€**

Tabla 2: Calculo de coste del proyecto

Organigrama jerárquico

Fase 1: Diseño	Diagrama de la infraestructura
	Configuración en local de los servicios
Fase 2: Planificación	Estimación de coste del proyecto
	Plan de seguridad y prevención
Fase 3: Implementación	Creación de la infraestructura
	Configuración de Wireguard
	Configuración de LDAP
	Configuración de ThinLinc
	Creación de S3
Fase 4: Comprobación	Pruebas de funcionamiento
Fase 5: Presentación	Presentar el proyecto

Tabla 3: Organigrama jerárquico

Definición de recursos y logística necesaria para el proyecto

Orden lógico

Es muy importante seguir el orden ya que si se hace desordenado el servicio no funcionaría hasta tenerlo todo montado

1- Diseño: Diagrama de la infraestructura > Configuración en local de los servicios

2- Planificación: Estimación del coste del proyecto > Plan de seguridad y prevención

3- Implementación: Creación de la Infraestructura > Configuración de VPN > Configuración de LDAP > Configuración de ThinLinc > Creación de S3

4- Comprobación: Pruebas de funcionamiento

5- Presentación: Presentación

Asignación de tiempos y recursos

SEMANA	ACTIVIDADES	RECURSOS UTILIZADOS
18/03/2025 8/04/2025	Fase 1 diseño	Proxmox, ThinLinc, LDAP
08/04/2025 22/04/2025	Fase 2 planificación	AWS CLI, GitHub, ThinLinc y LDAP en bash
22/04/2025 13/05/2025	Fase 3 implementación	ThinLinc, LDAP, Wireguard, Zabbix, S3
13/05/2025 1/06/2025	Fase 4 – 5 Comprobación y presentación	Wireguard cliente, ThinLinc cliente, Word

Tabla 4: Asignación de tiempos y recursos

Implementación

Una de las partes clave de los scripts de instalación del servicio es el script de instalación de ThinLinc. Automatizar este proceso representó un desafío importante, ya que durante la instalación el sistema requiere múltiples interacciones manuales a través del teclado.

Para resolver este problema, investigué alternativas que permitieran automatizar dichas interacciones. Encontré que la herramienta expect es ideal para este tipo de situaciones, ya que permite simular entradas de teclado en función de palabras clave o eventos específicos durante la ejecución de un script.

A continuación, se presenta un fragmento del script desarrollado utilizando expect, el cual permitió automatizar completamente la instalación de ThinLinc.

```
# Script Expect
expect <<EOF
set timeout -1
spawn $INSTALLER --no-gui

expect {
    -re {.*\[(yes|Yes)/[Nn]o\]\?.*} {
        send "yes\r"
        exp_continue
    }
    -re "Run ThinLinc setup now.*\[Yes/no\]\?" {
        send "yes\r"
        exp_continue
    }
    -re "Enter.*continue.*" {
        send "\r"
        exp_continue
    }
    -re "Server type.*\[Master/agent\]" {
        send "Master\r"
        exp_continue
    }
    -re "Externally reachable address.*\[ip/hostname/manual\]" {
        send "ip\r"
        exp_continue
    }
    -re "Administrator email.*" {
        send "ihumaram01@educantabria.es\r"
        exp_continue
    }
    -re "Web Administration password.*" {
        send -- "Admin1\r"
        exp_continue
    }
    eof
}
EOF
```

Ilustración 3: Script automatización de ThinLinc expect

Otra sección destacada del proyecto es la automatización de la configuración de un bucket en Amazon S3 para su uso como alojamiento web estático.

Esta parte del script se encarga de crear dinámicamente el bucket, configurarlo para permitir el acceso público de lectura, definirlo como sitio web estático y subir los archivos web necesarios.

Un aspecto importante de la automatización fue garantizar el correcto funcionamiento del sitio web, se desbloquearon las restricciones de acceso público, se añadió una política de bucket que permite la lectura de objetos a cualquier usuario y se configuró el comportamiento de la página de inicio (index.html).

Finalmente, el script automatiza también la subida de los archivos web (index.html, linux.html, windows.html, movil.html) y valida si existen antes de intentar cargarlos.

```
# Crear el bucket S3
if [ "$REGION" == "us-east-1" ]; then
    aws s3api create-bucket \
        --bucket "$BUCKET_NAME" \
        --region "$REGION" > /dev/null
else
    aws s3api create-bucket \
        --bucket "$BUCKET_NAME" \
        --region "$REGION" \
        --create-bucket-configuration LocationConstraint="$REGION" > /dev/null
fi

# Desbloquear acceso público
aws s3api put-public-access-block \
    --bucket "$BUCKET_NAME" \
    --public-access-block-configuration \
        BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false

# Agregar política pública de lectura
aws s3api put-bucket-policy --bucket "$BUCKET_NAME" --policy "{
  \"Version\": \"2012-10-17\",
  \"Statement\": [
    {
      \"Sid\": \"PublicReadGetObject\",
      \"Effect\": \"Allow\",
      \"Principal\": \"*\",
      \"Action\": \"s3:GetObject\",
      \"Resource\": \"arn:aws:s3:::$BUCKET_NAME/*\"
    }
  ]
}"

echo "Habilitando el sitio web estático..."

# Configurar como sitio web estático
aws s3api put-bucket-website --bucket "$BUCKET_NAME" --website-configuration '{
  "IndexDocument": { "Suffix": "index.html" },
  "ErrorDocument": { "Key": "index.html" }
}'

# Subir archivo index.html
if [ -f "proyecto/www/index.html" ]; then
    aws s3 cp proyecto/www/index.html s3://$BUCKET_NAME/index.html > /dev/null
else
    echo "⚠️  El archivo proyecto/www/index.html no existe. No se subió nada."
fi
```

Ilustración 4: Script configuración de lS3

También es importante el script automatizado de la configuración de wireguard, el script automatiza la configuración de clientes para una VPN con WireGuard, asignándoles IPs, claves públicas y privadas y reglas de red. También habilita el reenvío de tráfico y permite la navegación de los clientes a través del servidor VPN

A continuación, se muestra un fragmento del script:

```
# Crear configuración de clientes y añadirlos al servidor como peers
for CLIENT in "${CLIENTS[@]}"; do
    CLIENT_PRIV=$(cat "$WG_DIR/$CLIENT/client_private.key")
    CLIENT_PUB=$(cat "$WG_DIR/$CLIENT/client_public.key")
    CLIENT_IP="10.0.212.$IP_BASE"

    # Configuración del cliente
    cat > "$WG_DIR/$CLIENT/$CLIENT.conf" <<EOF
[Interface]
PrivateKey = $CLIENT_PRIV
Address = $CLIENT_IP/32
DNS = $DNS_SERVER

[Peer]
PublicKey = $SERVER_PUB
Endpoint = vpn-ivanhumara.duckdns.org:$PORT
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25
EOF

    # Agregar peer al servidor
    cat >> "$WG_DIR/wg0.conf" <<EOF
[Peer]
PublicKey = $CLIENT_PUB
AllowedIPs = $CLIENT_IP/32
EOF

    IP_BASE=$((IP_BASE + 1))
done

# Activar el reenvío de IPs en el sistema
sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' /etc/sysctl.conf
sysctl -p

# + Punto 4: Aceptar tráfico hacia y desde wg0 a nivel global
iptables -A FORWARD -i wg0 -j ACCEPT
iptables -A FORWARD -o wg0 -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.0.212.0/24 -o $NET_IFACE -j MASQUERADE

# Establecer permisos correctos
chmod 600 "$WG_DIR"/*.key "$WG_DIR"/*/*.key "$WG_DIR"/*.conf "$WG_DIR"/*/*.conf
```

Ilustración 5: Script configuración WireGuard

Este script automatiza la configuración inicial de Zabbix con base de datos PostgreSQL. Crea el usuario y la base de datos necesarios, importa el esquema predeterminado, ajusta el archivo de configuración del servidor y define los parámetros de conexión para la interfaz web. Además, reinicia y habilita los servicios y aplica los permisos necesarios para que el sistema funcione correctamente desde un navegador web

A continuación, se muestra un fragmento del script:

```
# Crear el script expect en el mismo archivo
expect << EOF
  set timeout -1
  spawn sudo -u postgres createuser --pwprompt zabbix
  expect "Enter password for new role:"
  send "Admin1\r"
  expect "Enter it again:"
  send "Admin1\r"
  expect eof
EOF

# Crear la base de datos inicial
sudo -u postgres createdb -O zabbix zabbix
zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u zabbix psql zabbix

# Configuración de Zabbix Server
echo "Configurando Zabbix Server..."
sudo sed -i 's/# DBPassword=/DBPassword=Admin1/' /etc/zabbix/zabbix_server.conf

# Reiniciar los servicios
sudo systemctl restart zabbix-server zabbix-agent apache2
sudo systemctl enable zabbix-server zabbix-agent apache2

# Configuracion inicial
sudo tee /etc/zabbix/web/zabbix.conf.php > /dev/null <<EOF
<?php
\$DB['TYPE']     = 'POSTGRESQL';
\$DB['SERVER']   = 'localhost';
\$DB['PORT']     = '5432';
\$DB['DATABASE'] = 'zabbix';
\$DB['USER']     = 'zabbix';
\$DB['PASSWORD'] = 'Admin1';
\$DB['SCHEMA']   = 'public';

\$ZBX_SERVER      = 'localhost';
\$ZBX_SERVER_PORT = '10051';
\$ZBX_SERVER_NAME = 'Zabbix Server';

\$IMAGE_FORMAT_DEFAULT = IMAGE_FORMAT_PNG;
EOF

sudo chown www-data:www-data /etc/zabbix/web/zabbix.conf.php
```

Ilustración 6: Script instalación automática Zabbix

Una parte fundamental del despliegue automatizado fue la creación y configuración de los Grupos de Seguridad (Security Groups) asociados a cada uno de los servicios principales de la infraestructura: WireGuard VPN, Zabbix, LDAP y ThinLinc. Esta sección del script utiliza la CLI de AWS para definir reglas precisas de entrada con el fin de garantizar tanto el acceso autorizado como la seguridad de la red.

```
179  echo "Creando Grupos de Seguridad..."
180
181  # Grupo de seguridad para WireGuard VPN
182  SG_WIREGUARD_ID=$(aws ec2 create-security-group --group-name "sg_wireguard" --description "SG para WireGuard VPN" --vpc-id "$VPC_ID" --query
183  aws ec2 authorize-security-group-ingress --group-id "$SG_WIREGUARD_ID" --protocol udp --port 51820 --cidr "0.0.0.0/0" # WireGuard
184  aws ec2 authorize-security-group-ingress --group-id "$SG_WIREGUARD_ID" --protocol tcp --port 10050 --cidr "0.0.0.0/0" # Zabbix agente
185  aws ec2 authorize-security-group-ingress --group-id "$SG_WIREGUARD_ID" --protocol tcp --port 22 --cidr "0.0.0.0/0" # SSH
186  aws ec2 authorize-security-group-ingress --group-id "$SG_WIREGUARD_ID" --protocol icmp --port -1 --cidr "10.0.2.0/24" # PING
187
188  # Grupo de seguridad para Zabbix
189  SG_ZABBIX_ID=$(aws ec2 create-security-group --group-name "sg_zabbix" --description "SG para Zabbix Server" --vpc-id "$VPC_ID" --query 'Grou
190  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol tcp --port 10050 --cidr "0.0.0.0/0" # Zabbix agente
191  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol tcp --port 10051 --cidr "0.0.0.0/0" # Zabbix server
192  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol tcp --port 80 --cidr "0.0.0.0/0" # Zabbix
193  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol tcp --port 443 --cidr "0.0.0.0/0" # Zabbix
194  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol tcp --port 22 --cidr "0.0.0.0/0" # SSH
195  aws ec2 authorize-security-group-ingress --group-id "$SG_ZABBIX_ID" --protocol icmp --port -1 --cidr "0.0.0.0/0" # PING
196
197  # Grupo de seguridad para LDAP
198  SG_LDAP_ID=$(aws ec2 create-security-group --group-name "sg_ldap" --description "SG para LDAP" --vpc-id "$VPC_ID" --query 'GroupId' --output
199  aws ec2 authorize-security-group-ingress --group-id "$SG_LDAP_ID" --protocol tcp --port 389 --cidr "10.0.2.0/24" # LDAP
200  aws ec2 authorize-security-group-ingress --group-id "$SG_LDAP_ID" --protocol tcp --port 10050 --cidr "0.0.0.0/0" # Zabbix agente
201  aws ec2 authorize-security-group-ingress --group-id "$SG_LDAP_ID" --protocol tcp --port 22 --cidr "0.0.0.0/0" # SSH
202  aws ec2 authorize-security-group-ingress --group-id "$SG_LDAP_ID" --protocol icmp --port -1 --cidr "10.0.2.0/24" # PING
203
204  # Grupo de seguridad para ThinLinc
205  SG_THINLINC_ID=$(aws ec2 create-security-group --group-name "sg_thinlinc" --description "SG para ThinLinc" --vpc-id "$VPC_ID" --query 'Group
206  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 300 --cidr "0.0.0.0/0" # ThinLinc
207  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 443 --cidr "0.0.0.0/0" # ThinLinc
208  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 904 --cidr "0.0.0.0/0" # ThinLinc
209  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 5901-5999 --cidr "0.0.0.0/0" # ThinLinc
210  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 389 --cidr "10.0.2.0/24" # LDAP
211  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 10050 --cidr "0.0.0.0/0" # Zabbix agente
212  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol tcp --port 22 --cidr "0.0.0.0/0" # SSH
213  aws ec2 authorize-security-group-ingress --group-id "$SG_THINLINC_ID" --protocol icmp --port -1 --cidr "0.0.0.0/0" # PING
```

Ilustración 7: Script grupos de seguridad

Grupo de Seguridad para WireGuard

Se creó un grupo de seguridad específico para el servidor VPN que permite:

- Tráfico UDP en el puerto 51820, necesario para WireGuard.
- Tráfico TCP en el puerto 10050, utilizado por el agente de Zabbix.
- Acceso SSH (puerto 22) desde cualquier origen.
- Paquetes ICMP desde la red 10.0.2.0/24 para permitir PING.

Grupo de Seguridad para Zabbix

Este grupo habilita todos los accesos necesarios para el monitoreo:

- Puertos 10050 y 10051 para la comunicación entre agentes y servidor Zabbix.
- Acceso HTTP (80) y HTTPS (443) para la interfaz web.
- SSH (22) para administración remota.
- ICMP desde cualquier origen para diagnóstico de red.

Grupo de Seguridad para LDAP

Se configuró con un enfoque en accesos restringidos:

- Puerto 389 (LDAP) limitado a la red 10.0.2.0/24, lo que refuerza la seguridad.
- Puerto 10050 para integración con Zabbix.
- SSH y PING para administración y conectividad.

Grupo de Seguridad para ThinLinc

Este grupo está diseñado para soportar múltiples componentes del entorno de escritorio remoto:

- Puertos 300, 443, 904 y 5901-5999, todos relacionados con el servicio ThinLinc.
- Acceso al puerto 389 para integrarse con el servidor LDAP.
- Puerto **10050** para Zabbix y **22** para administración remota.
- Tráfico ICMP para conectividad y diagnóstico general.

Este script automatiza el registro de instancias EC2 en grupos de destino y la creación de listeners en un balanceador de carga (NLB) en AWS. Primero obtiene los IDs de dos instancias específicas mediante sus direcciones IP. Luego, las registra en los grupos de destino asociados a diferentes puertos (22 para SSH y 300 para el acceso web de thinlinc). Por último, crea listeners en el NLB para redirigir automáticamente el tráfico entrante a las instancias correspondientes según el puerto.

A continuación, se muestra un fragmento del script:

```
# Obtener IDs de las instancias
ID_MAESTRO1=$(aws ec2 describe-instances \
  --filters "Name=private-ip-address,Values=10.0.2.11" \
  --region $REGION \
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)

ID_MAESTRO2=$(aws ec2 describe-instances \
  --filters "Name=private-ip-address,Values=10.0.2.12" \
  --region $REGION \
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)

# Registrar instancias en los Target Groups
aws elbv2 register-targets --target-group-arn $TG_SSH_ARN \
  --targets Id=$ID_MAESTRO1 Id=$ID_MAESTRO2 \
  --region $REGION

aws elbv2 register-targets --target-group-arn $TG_300_ARN \
  --targets Id=$ID_MAESTRO1 Id=$ID_MAESTRO2 \
  --region $REGION

# Crear Listeners en el NLB
aws elbv2 create-listener \
  --load-balancer-arn $NLB_ARN \
  --protocol TCP \
  --port 22 \
  --default-actions Type=forward,TargetGroupArn=$TG_SSH_ARN \
  --region $REGION

aws elbv2 create-listener \
  --load-balancer-arn $NLB_ARN \
  --protocol TCP \
  --port 300 \
  --default-actions Type=forward,TargetGroupArn=$TG_300_ARN \
  --region $REGION
```

Ilustración 8: Script configuración NLB

Puesta en marcha, explotación

Cambios de Configuración, Seguridad y Legalidad Previos a la Puesta en Producción

1. Cambios de Configuración:

- Verificar la correcta configuración de los servicios críticos (VPN, ThinLinc, LDAP y Zabbix) para garantizar la continuidad operativa.
- Ajustar los parámetros de rendimiento de las instancias para que soporten la carga en el entorno de producción.
- Implementar balanceo de carga en los servidores ThinLinc (Maestro1 y Maestro2) para garantizar alta disponibilidad.
- Actualizar configuraciones en el archivo de inventario para incluir las IPs de producción.

2. Seguridad:

- Realizar un análisis de vulnerabilidades en las instancias EC2 mediante herramientas como OpenVAS o Nessus.
- Realizar pruebas de penetración en el servicio VPN para asegurar la protección frente a ataques externos.
- Revisar los permisos en los grupos de seguridad de AWS para garantizar que solo las IPs autorizadas tengan acceso.
- Implementar monitoreo de tráfico en los servidores VPN y ThinLinc para detectar posibles accesos no autorizados.
- Revisar las reglas de iptables en el servidor VPN para asegurar el acceso solo desde la red interna y clientes autorizados.

3. Legalidad:

- Asegurarse de que los datos personales tratados mediante LDAP cumplan con la legislación vigente (por ejemplo, GDPR).
- Actualizar las políticas de privacidad en caso de que los datos gestionados cambien al pasar a producción.
- Garantizar el cifrado de todas las conexiones de usuarios externos mediante el uso de VPN.

Pasos para la Puesta en Producción

1. **Despliegue en el Entorno de Producción:**
    - Ejecutar el script de infraestructura en AWS para desplegar la VPC, subredes y las instancias necesarias.
    - Verificar que el balanceador de carga esté correctamente configurado para redirigir el tráfico a los servidores ThinLinc.
2. **Verificación de Servicios:**
    - Comprobar el correcto funcionamiento de la VPN y acceso a los recursos internos.
    - Verificar que el servicio de escritorio remoto ThinLinc esté operativo y accesible desde los clientes.
    - Realizar pruebas de monitoreo en Zabbix para asegurarse de que los agentes informen correctamente.
3. **Pruebas de Seguridad:**
    - Ejecutar pruebas de penetración nuevamente en el entorno de producción para identificar posibles vulnerabilidades no detectadas previamente.
    - Revisar los registros de acceso y errores para identificar posibles problemas o configuraciones incorrectas.
4. **Optimización y Monitoreo:**
    - Ajustar el monitoreo en Zabbix para incluir las métricas críticas en producción (CPU, RAM, uso de red).
    - Implementar alertas proactivas para identificar caídas del servicio o anomalías en el tráfico.

Pruebas y control de calidad

Pruebas de funcionamiento de una página web alojada en un bucket S3. La primera página que se carga es la página de inicio, la cual presenta una introducción al proyecto y ofrece tres enlaces que dirigen a diferentes páginas con tutoriales.

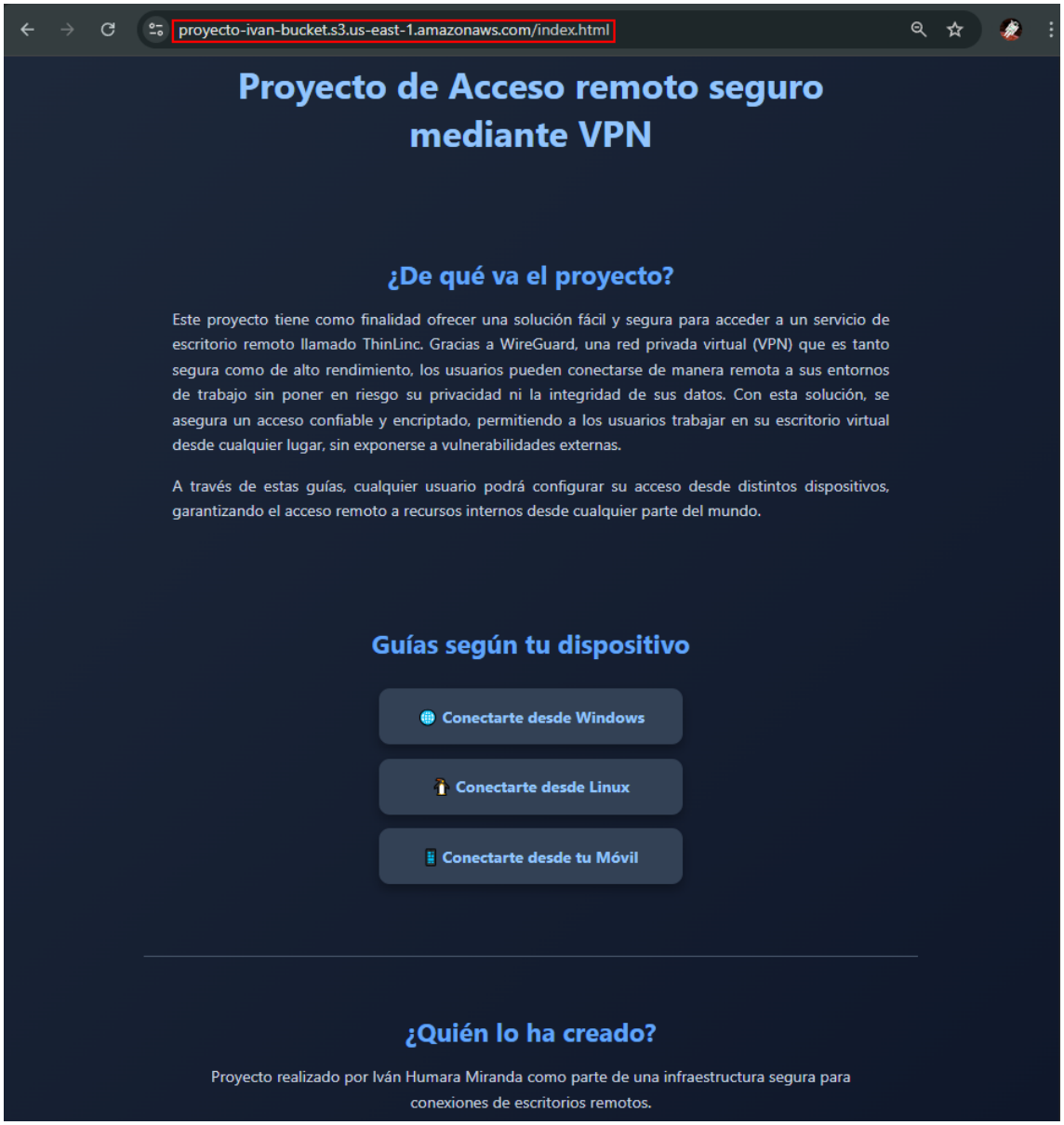


Ilustración 9: Prueba página principal

Esta es la página del tutorial de conexión a Windows, contiene una guía paso a paso sobre cómo descargar y configurar el cliente WireGuard.

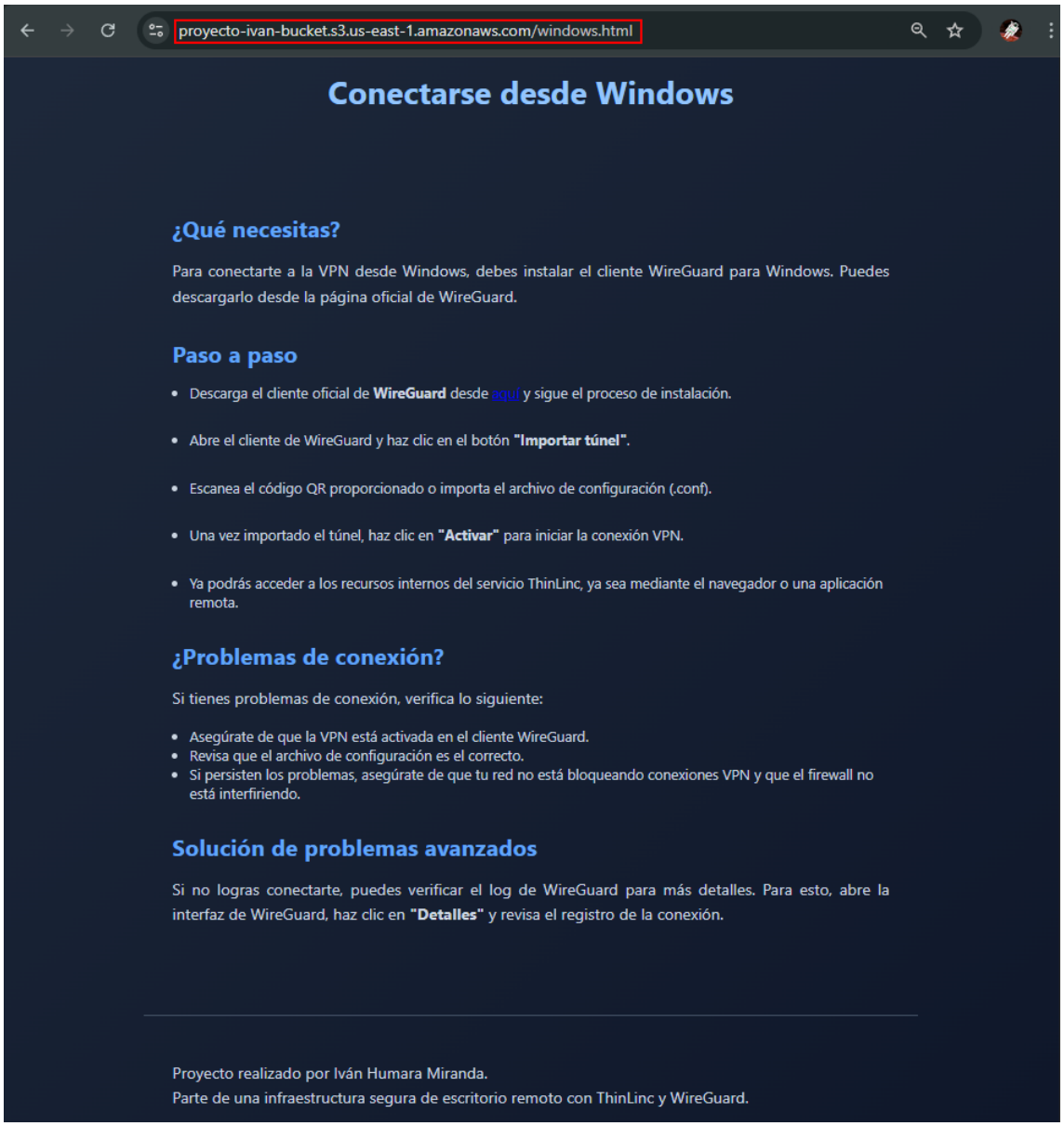


Ilustración 10: Prueba página tutorial Windows

Esta es la página del tutorial de conexión a Ubuntu, proporciona una guía paso a paso para descargar, instalar y configurar el cliente WireGuard

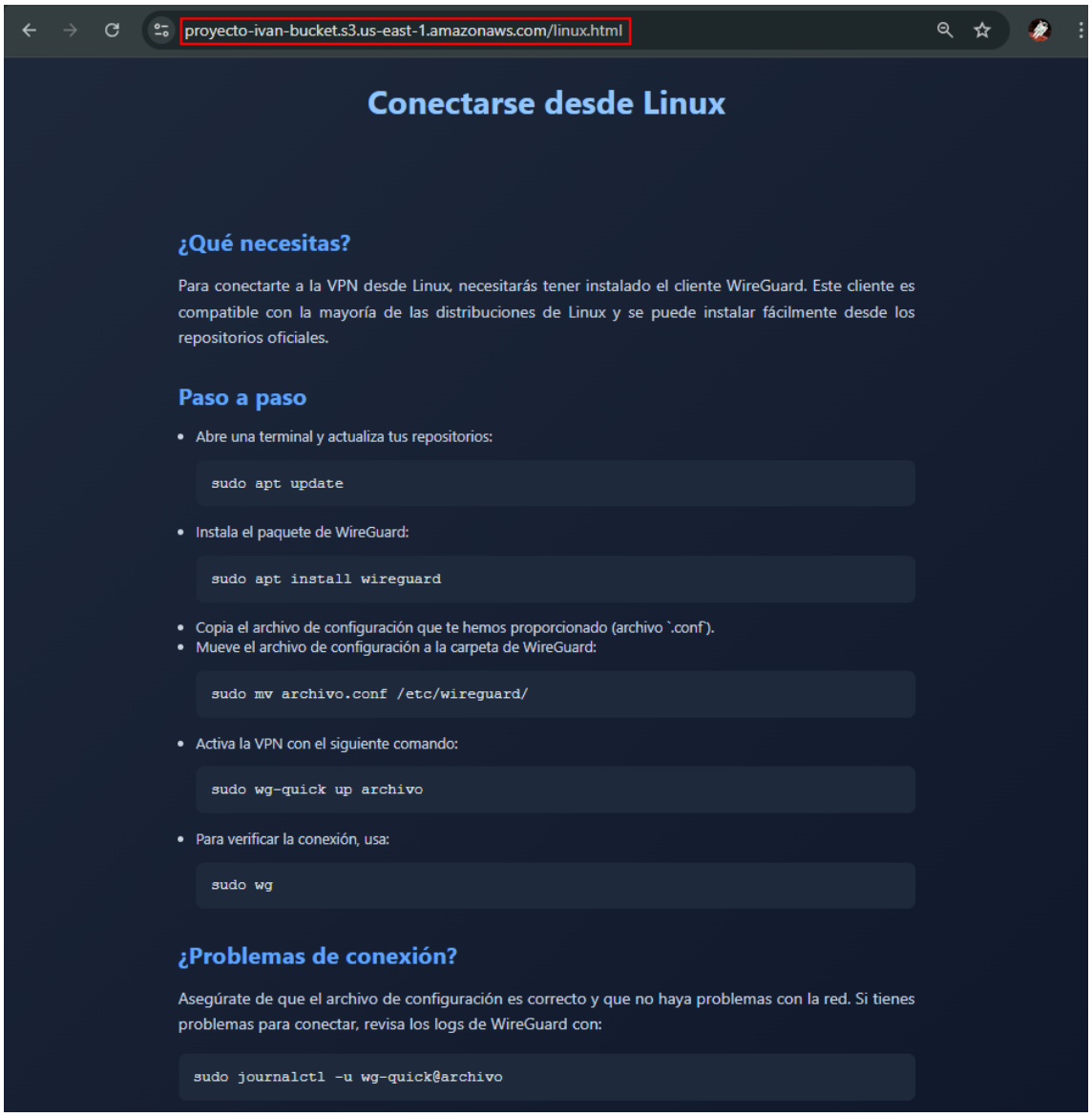


Ilustración 11: Pruebas página tutorial Ubuntu

Esta es la página del tutorial de conexión desde dispositivos móviles, ofrece una guía paso a paso para instalar la aplicación de WireGuard

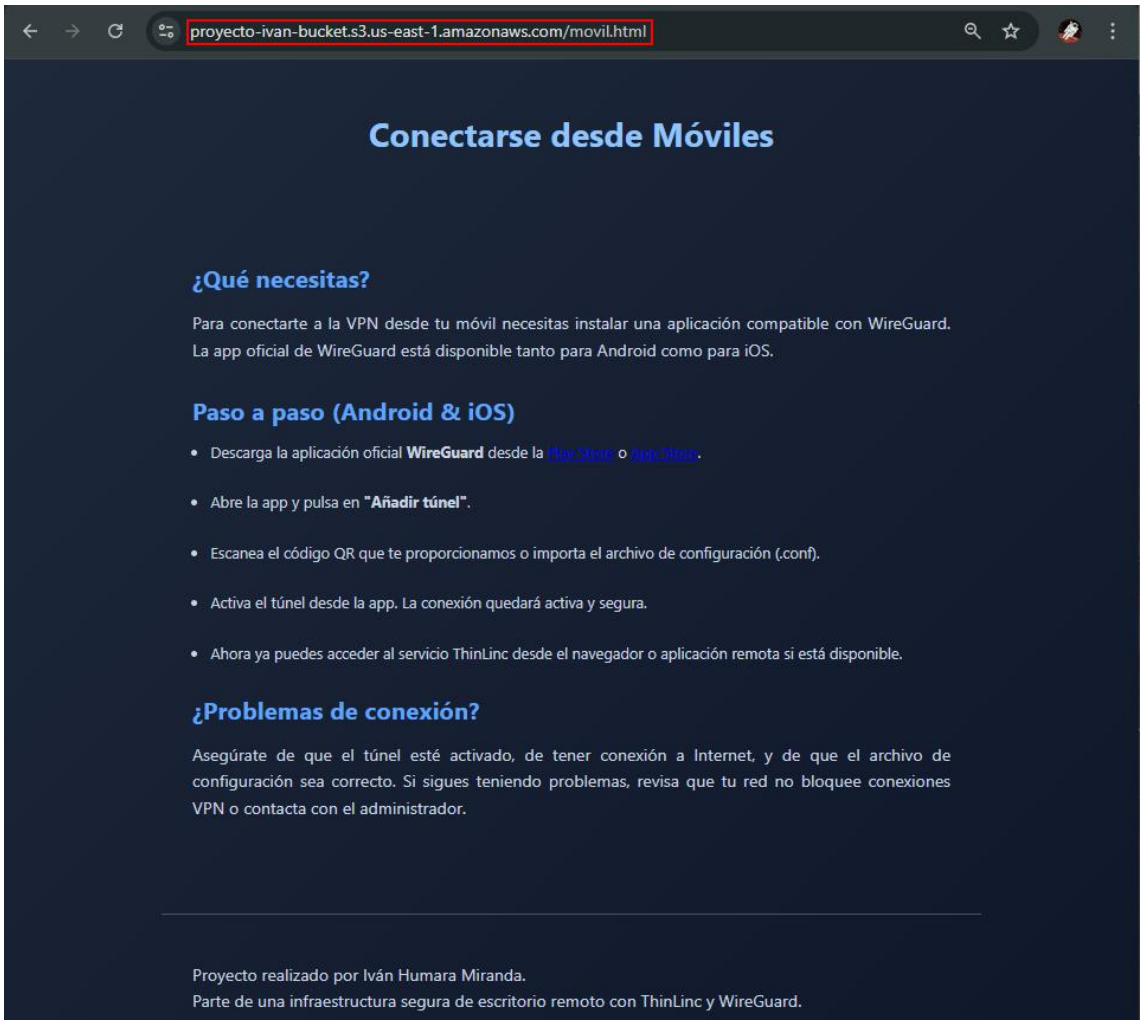


Ilustración 12: Pruebas página tutorial móvil

Una vez completado alguno de los tutoriales, como el de Windows, deberíamos ver que el cliente comienza a enviar y recibir datos con el servidor WireGuard

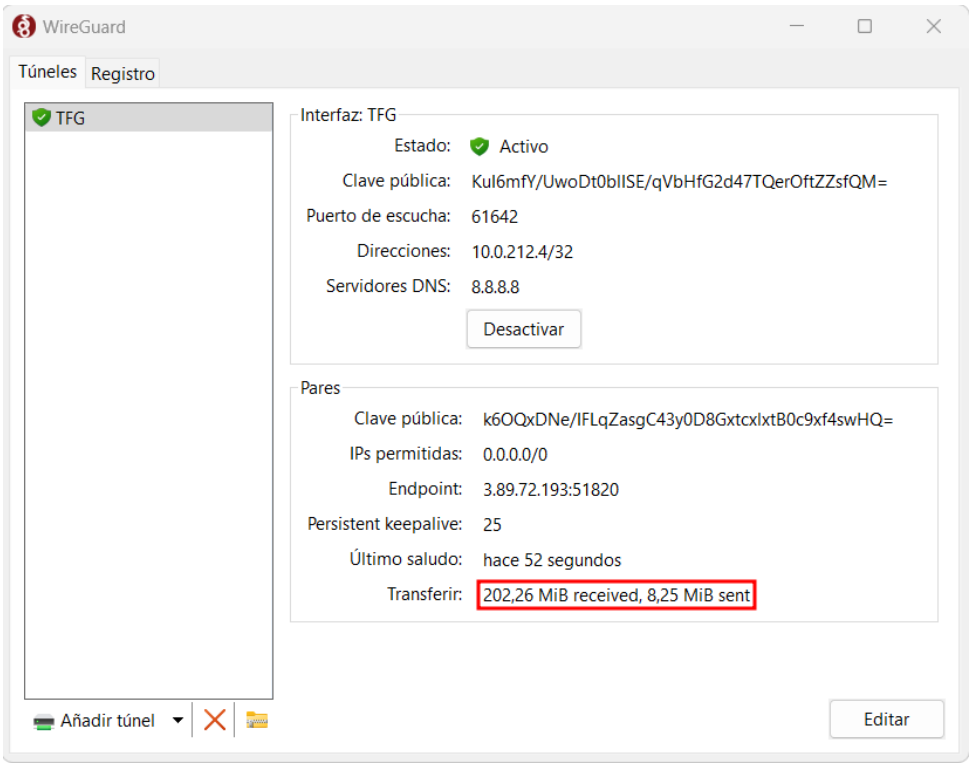


Ilustración 13: Pruebas comunicación Wireguard

Una vez conectados a la VPN, ya podremos acceder al servicio ThinLinc. En la primera prueba, con todos los equipos en ejecución, deberíamos ser redirigidos automáticamente a cualquiera de los servidores disponibles

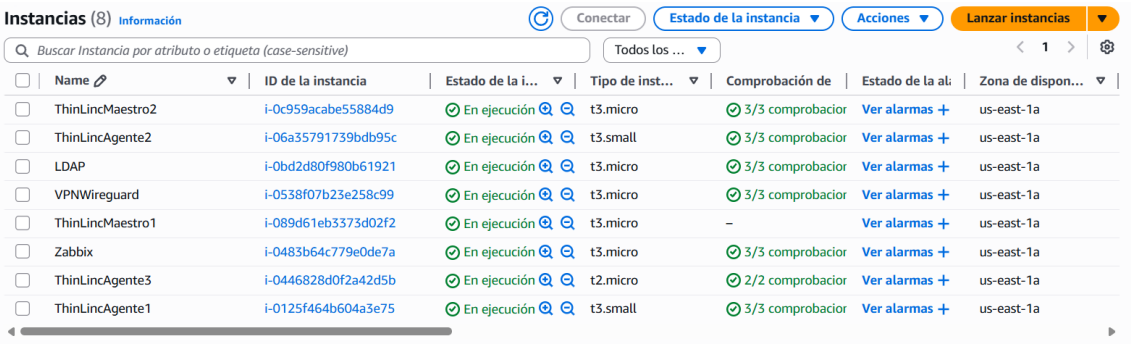


Ilustración 14: Pruebas listado de instancias 1

En el cliente ThinLinc, ya sea en la aplicación o vía web, se debe ingresar el DNS de nuestro balanceador de carga (NLB). Este se encargará de redirigir la conexión a uno de los servidores disponibles. Además, es necesario iniciar sesión con un usuario registrado en el servidor LDAP

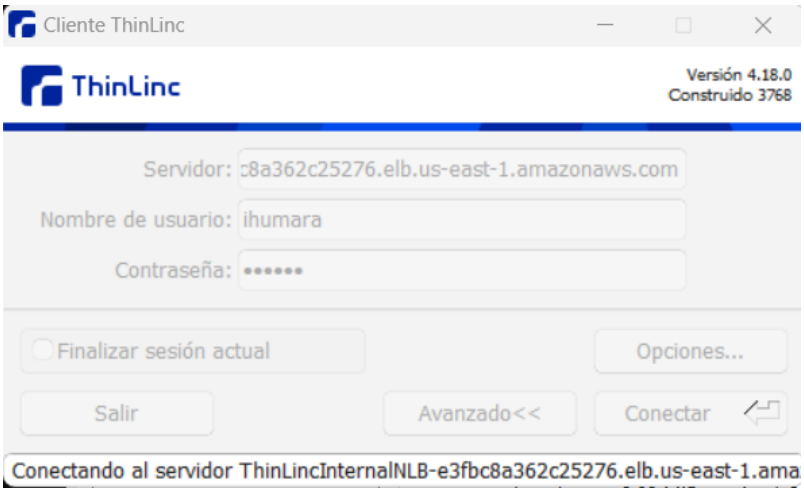


Ilustración 15: Pruebas iniciando sesión en ThinLinc

Una vez establecida la conexión con el servidor, se mostrará el escritorio correspondiente al usuario con el que se ha iniciado sesión

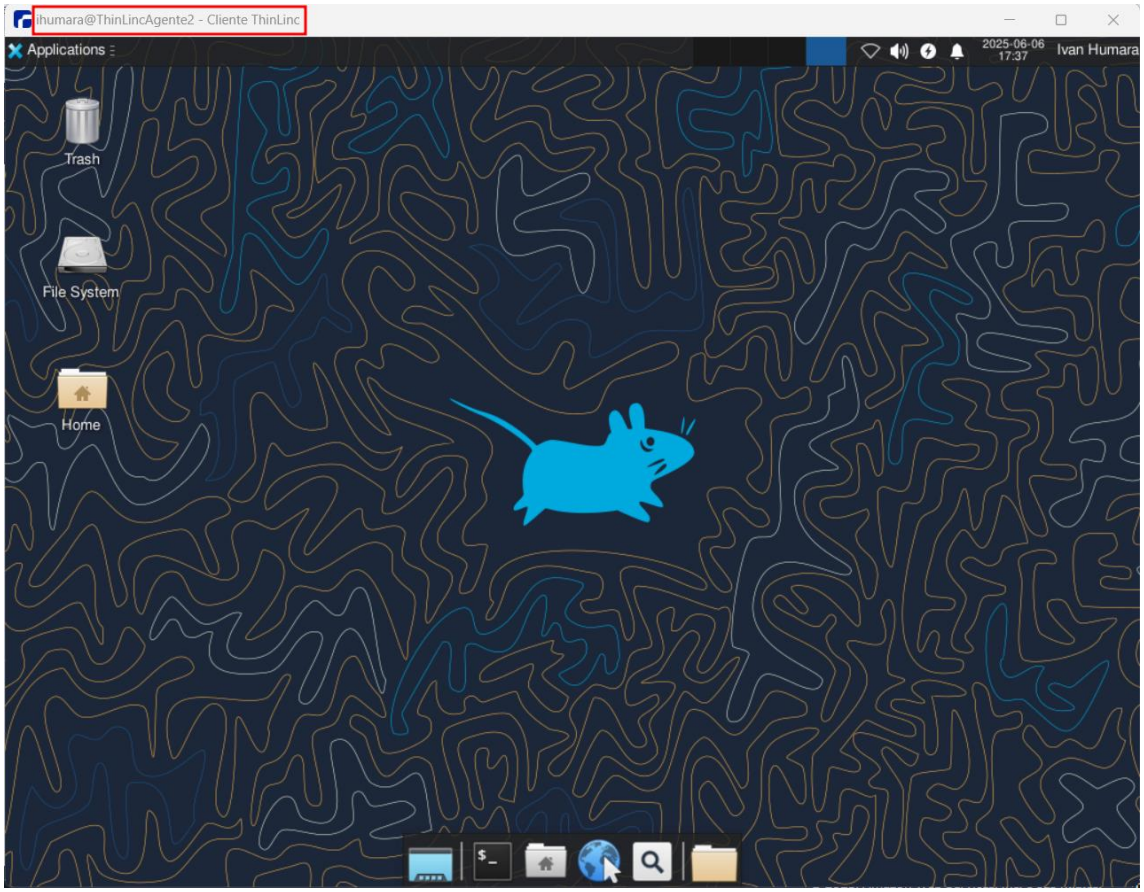


Ilustración 16: Pruebas escritorio servidor maestro 1

A continuación, probaremos el funcionamiento de la alta disponibilidad. Para ello, apagaremos el servidor 1, que fue al que nos conectamos anteriormente

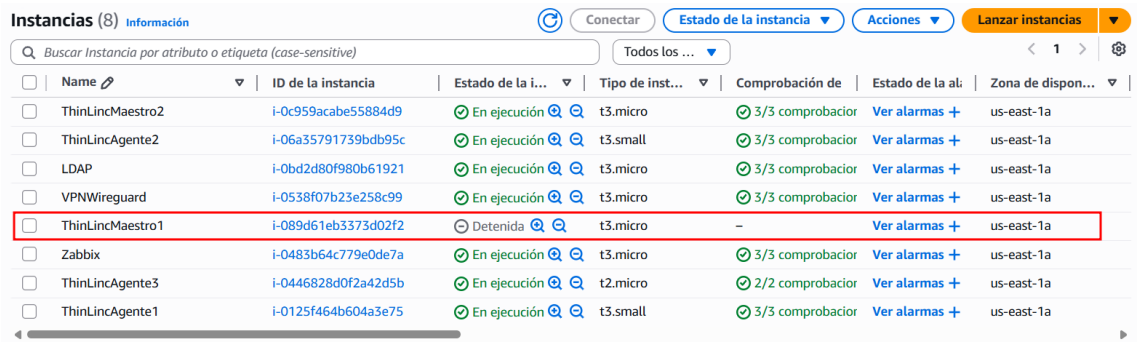


Ilustración 17: Pruebas listado de instancias 2

Volvemos a iniciar sesión utilizando el DNS del balanceador y el mismo usuario. En esta ocasión, la conexión es redirigida al servidor agente 3, el cual está asociado al servidor maestro 2

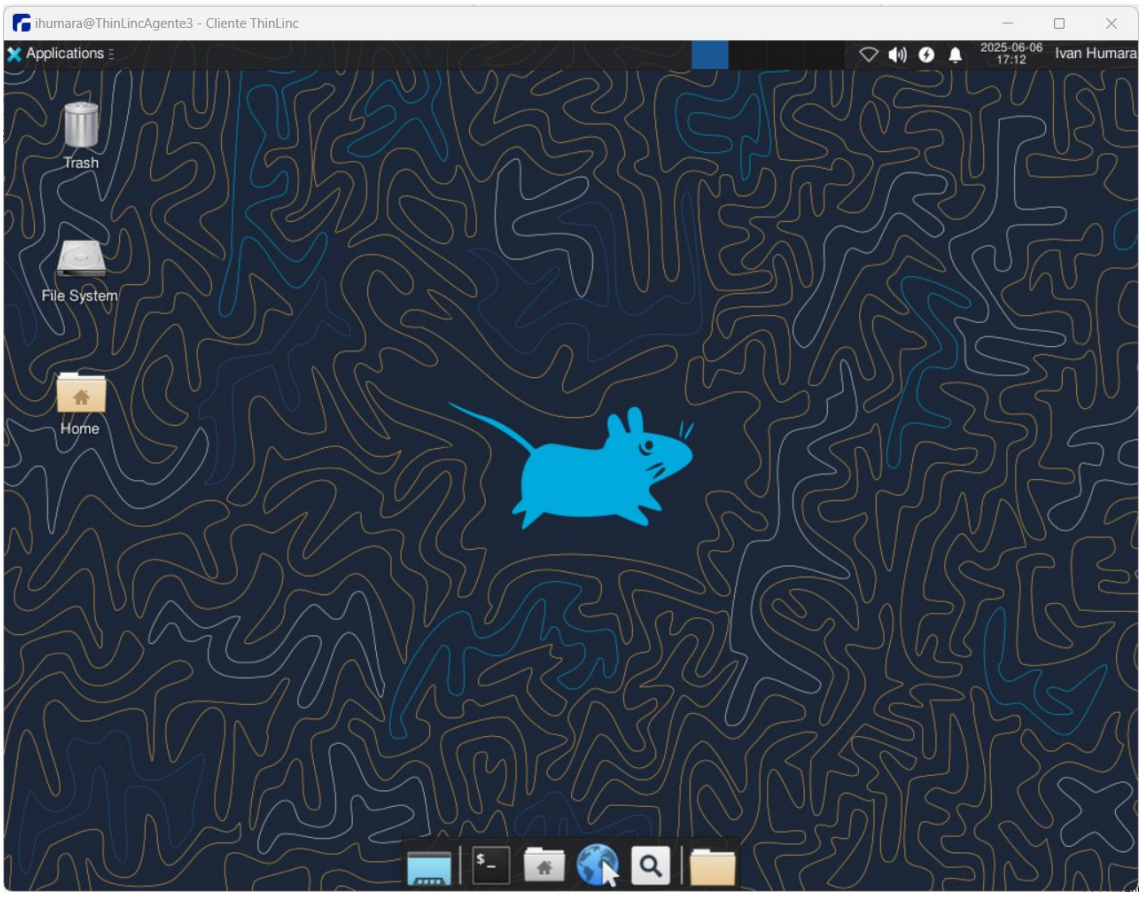


Ilustración 18: Pruebas escritorio utilizando Maestro 2

Para garantizar que el producto final cumple con los requisitos funcionales y de calidad, se ha definido un plan de pruebas, en él se describen las distintas pruebas que se realizarán, detallando las entradas, salidas esperadas, resultados obtenidos y la figura responsable de su ejecución.

El objetivo principal de este plan es asegurar que todos los servicios desplegados (VPN, ThinLinc, LDAP, portal web en S3 y monitorización con Zabbix) funcionen correctamente antes de pasar el sistema a producción.

A continuación, se detalla cada prueba de forma estructurada:

Prueba	Descripción	Entrada	Salida esperada	Resultado obtenido	Responsable
1	Monitorización de servidores en Zabbix	Registrar el servidor ThinLinc en Zabbix y monitorear	El agente Zabbix reporta correctamente CPU, RAM, disco	Servidor reporta correctamente sin errores de conexión	Iván (Administrador)
2	Acceso al portal web en S3	Navegador accede a la URL pública	Carga de index.html correctamente	Página web visible sin errores 404	Iván (Administrador)
3	Conexión a la VPN	Cliente WireGuard configurado	Conexión establecida con IP de la VPN	Conexión exitosa, IP en el rango VPN y acceso a internet	Iván (Administrador)
4	Comprobar HA con NLB	Apagar el servidor Maestro 1	El NLB reenvía el tráfico al servidor Maestro 2	El NLB reenvía correctamente el trafica	Iván (Administrador)
5	Acceso al escritorio ThinLinc	Usuario LDAP válido se conecta	Inicio de sesión exitoso en escritorio remoto	Escritorio cargado sin errores	Iván (Administrador)
6	Prueba de desconexión VPN	Desconexión manual del cliente	Cliente pierde acceso a la red privada	Sin acceso a ThinLinc ni recursos internos	Iván (Administrador)

Tabla 5: Pruebas y control de calidad

Gestión económica o plan de empresa

Gestión Económica del Proyecto

El proyecto tiene como objetivo la creación de una infraestructura en la nube (AWS) que permita el acceso remoto seguro a escritorios ThinLinc a través de una VPN (WireGuard), además de integrar servicios de autenticación LDAP y monitoreo con Zabbix. A continuación, se presentan los costos asociados con el desarrollo, implementación y mantenimiento de esta infraestructura:

1. **Recursos Materiales**

    - **Infraestructura en la Nube**:

        - **Costo de las** instancias EC2 **t3.micro:** 46,52€ al mes.
        - **Costo de** alma**cenamiento (S3/EBS):** 1,05€ mensual.
        - **Costo del** tráf**ico de datos:** 28,36€ mensual.
    - **Software**:
        - **Licencia de ThinLinc**: 0€ por usuario/mes.
        - **Licencia de Zabbix:** 0€ mensual
        - **Licencia de Wireguard:** 0€ mensual

2. **Proveedores**
    - **Servicios de la Nube (AWS)**:
        - **AWS EC2, S3 y ancho de banda**: 80.34€ mensual.
    - **Servicios de Internet**:
        - Proveedor de internet: 0€ mensual.
    - **Energía**: 0€ mensual.
3. **Coste de Desarrollo del Proyecto**
    - **Fases del Proyecto**:
        - **Análisis y diseño**: 8h x 18€/h = 144€.
        - **Implementación de servicios (WireGuard, LDAP, ThinLinc, Zabbix)**: 400€ total.
        - **Pruebas**: 80€ total.
        - **Elaboración de documentación y guías**: 200€ total.
4. **Coste de Perfiles**
    - **Desarrolladores/Administradores de Sistemas**: 18€ por hora, estimado 40 horas.
5. **Coste Total del Proyecto**
    - **Inversiones Iniciales**: 900€ (costo total para comenzar el proyecto incluyendo infraestructura y mano de obra).
    - **Costos Operativos Anuales**: 960€.

Categoría	Detalle	Coste
1. Recursos Materiales		
Infraestructura en la Nube	Costo de las instancias EC2 t3.micro (53.14$/mes)	46,52€ / mes
	Costo de almacenamiento (S3/EBS) (1.20$/mes)	1.05€ / mes
	Costo de tráfico de datos (32.40$/mes)	28,36€ / mes
Software		
Licencia de ThinLinc	Licencia sin costo por usuario/mes	0€ / mes
Licencia de Zabbix	Licencia sin costo mensual	0€ / mes
Licencia de WireGuard	Licencia sin costo mensual	0€ / mes
2. Proveedores		
Servicios de la Nube (AWS)	EC2, S3 y ancho de banda (80.34€/mes)	80.34€ / mes
Servicios de Internet	Proveedor de Internet sin coste mensual	0€ / mes
Energía	Costo energético sin coste mensual	0€ / mes
3. Coste de Desarrollo del Proyecto		
Análisis y Diseño	8 horas a 18€/h (144€)	144€
Implementación de Servicios	WireGuard, LDAP, ThinLinc, Zabbix (costo total)	400€
Pruebas	Coste total de pruebas	80€
Documentación y Guías	Elaboración de documentación y guías	200€
4. Coste de Perfiles		
Desarrolladores/Administradores de Sistemas	40 horas a 18€/h	720€
5. Coste Total del Proyecto		
Inversiones Iniciales	Infraestructura, mano de obra, etc.	900€
Costos Operativos Anuales	Coste de la infraestructura y servicios mensuales	960€ / año

Tabla 6: Gestión económica

Conclusiones y valoración personal

Este proyecto me ha servido mucho para poner en práctica todo lo que he aprendido durante el ciclo. He podido ver cómo se conectan todos los servicios que hemos visto en clase y cómo aplicarlos en una infraestructura real, como por ejemplo montar una VPN, configurar escritorios remotos, integrar un servidor LDAP y usar Zabbix para el monitoreo de los equipos.

Me gustó especialmente la parte de automatizar la instalación y configuración, porque te das cuenta de lo útil que es tenerlo todo bien organizado para ahorrar

tiempo y evitar errores. También me pareció muy interesante poder trabajar con servicios en la nube como AWS, que es algo que usan muchas empresas hoy en día.

Las FCTs me sirvieron bastante porque en la empresa donde estuve configuré algunos de los servicios que usé en este proyecto, así que no partía de cero. Ya tenía una pequeña base, y eso me ayudó a avanzar más seguro y con más confianza.

En general, ha sido una experiencia muy completa y útil, y me ha hecho ver que todo lo que hemos estudiado tiene una aplicación práctica en el mundo real.

Bibliografía

A continuación, se listan todas las fuentes que consulté para la realización del proyecto, incluyendo páginas oficiales y documentación de configuración:

- **WireGuard (VPN)**
  - Página oficial: https://www.wireguard.com/
  - Instalación y configuración: https://www.wireguard.com/install/
- **ThinLinc (escritorio remoto)**
  - Página oficial: https://www.cendio.com/thinlinc/
  - Manual de administración: https://www.cendio.com/resources/docs/tag/
- **Zabbix (monitorización de los servidores)**
  - Página oficial: https://www.zabbix.com/
  - Documentación oficial: https://www.zabbix.com/documentation/current/manual
- **LDAP (autenticación de usuarios)**
  - Página oficial: https://www.openldap.org/
  - Guía en Ubuntu: https://help.ubuntu.com/community/OpenLDAPServer
- **AWS (Amazon Web Services)**
  - Sitio oficial: https://aws.amazon.com/
  - Documentación general: https://docs.aws.amazon.com/
  - Calculadora de costes: https://calculator.aws.amazon.com/

- **Amazon S3 (Simple Storage Service)**
    - Página oficial: https://aws.amazon.com/s3/
    - Documentación: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html
- **AWS Network Load Balancer**
    - Descripción general: https://aws.amazon.com/elasticloadbalancing/network-load-balancer/
    - Guía de uso: https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html
- **Keepalived (alta disponibilidad)**
    - Proyecto en GitHub: https://github.com/acassen/keepalived
    - Documentación oficial: https://keepalived.readthedocs.io/en/latest/
- **Xfce4 (entorno de escritorio ligero)**
    - Página oficial: https://xfce.org/
    - Documentación y ayuda: https://docs.xfce.org/
- **Otras fuentes de apoyo**
    - Stack Overflow: https://stackoverflow.com/
    - Ubuntu Forums: https://ubuntuforums.org/
    - DigitalOcean Community: https://www.digitalocean.com/community/tutorials

Anexos

Configuración zabbix

A continuación, se muestra la documentación de la configuración web del servidor zabbix

Monitoreo de equipos

Lo primero que hay que hacer es ir al apartado de Monitoreo > Equipos, después pulsamos en crear host y ponemos el nombre, también añadimos al grupo de equipos al que pertenece, que en este caso es Zabbix servers y por último la IP del servidor

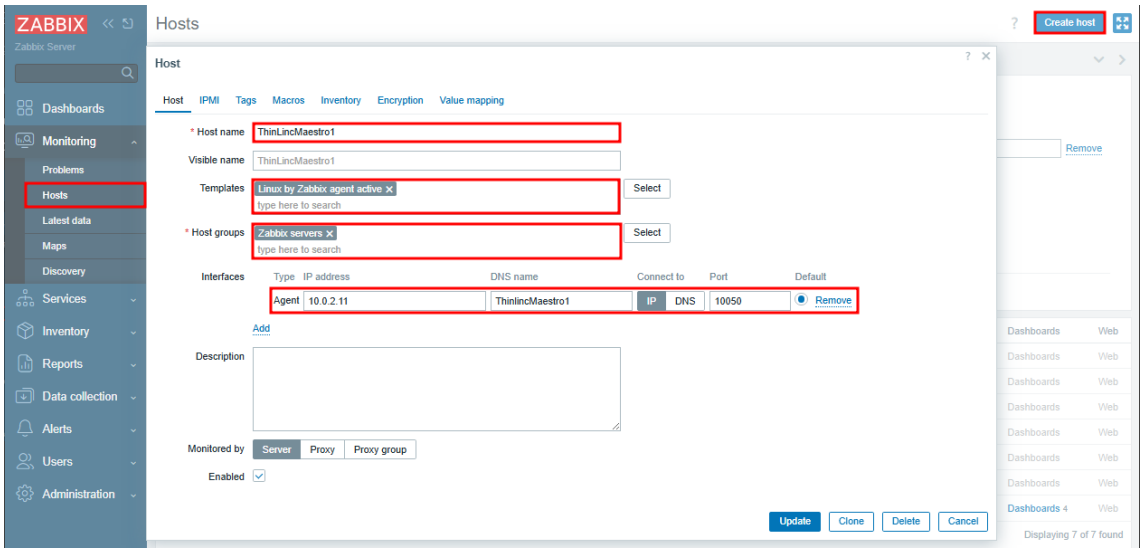


Ilustración 19: Configuración zabbix, monitoreo de equipos

Para comprobar el correcto funcionamiento del monitoreo de los equipos tendremos que volver a Monitoreo > Equipos y saldrá una lista con todos los equipos

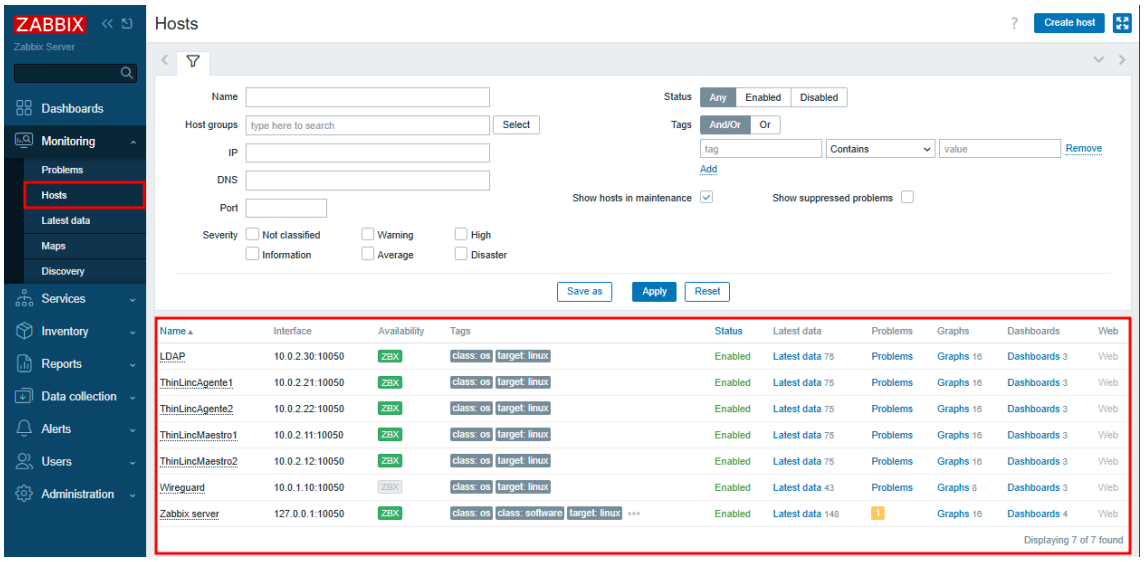


Ilustración 20: Configuración zabbix, Comprobación de monitoreo de equipos

Una vez se tengan los equipos configurados se podrá acceder a los gráficos de rendimiento de cada servidor, eso se hace yendo a un Equipo > Gráficos o Dashboard y hay encontraremos los gráficos de rendimiento de almacenamiento, memoria RAM etc.

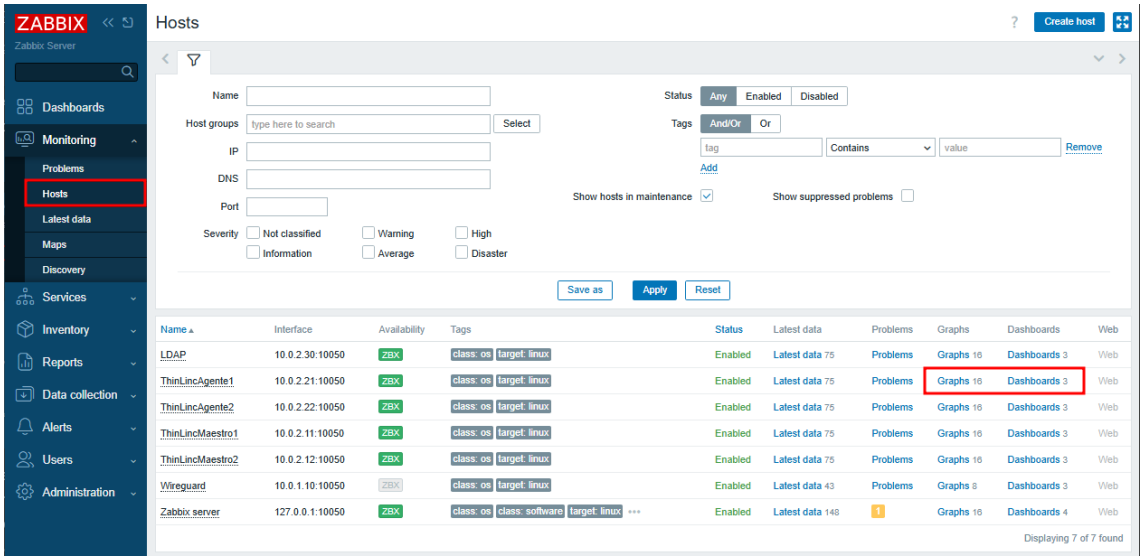


Ilustración 21: Configuración zabbix, monitoreo de equipos gráficos

Y te llevara a los gráficos del servidor seleccionado

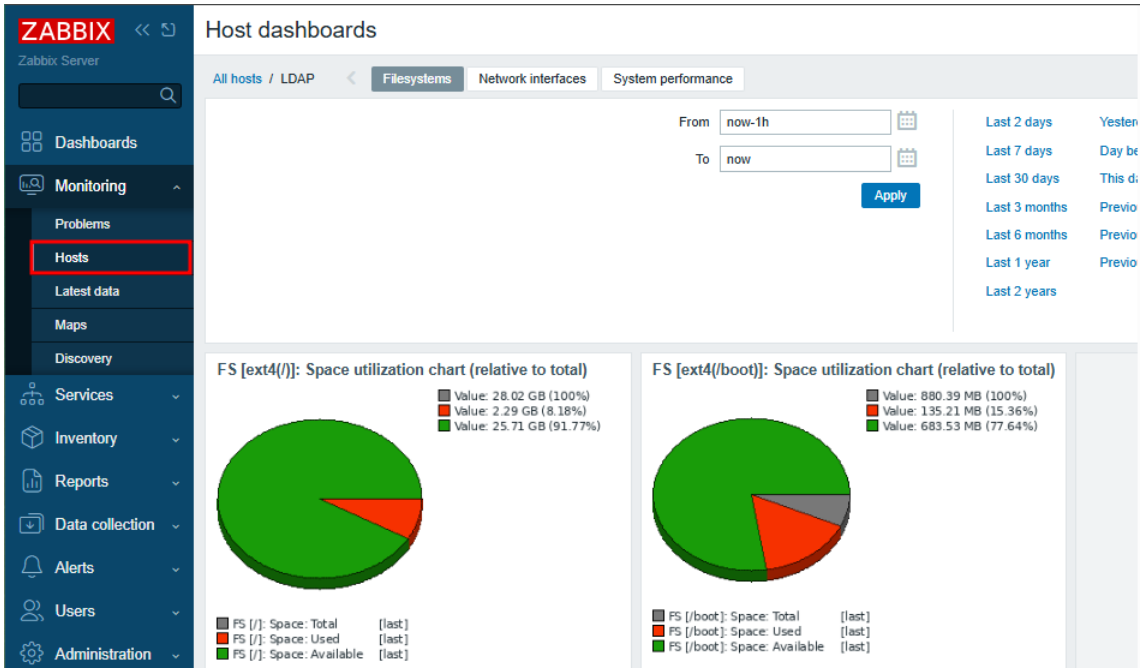


Ilustración 22: Configuración zabbix, monitoreo de equipos gráficos 2

Reglas de descubrimiento

Ahora vamos al apartado de Recopilación > Descubrimientos, ponemos un nombre a la regla de descubrimiento asignamos en un rango en el que quieras descubrir los servidores que haya en mi caso de la IP 10.0.2.10-30 y por último hay que poner porque puertos va a buscar en este caso serán el puerto 389 (LDAP) y el 300 (ThinLinc)

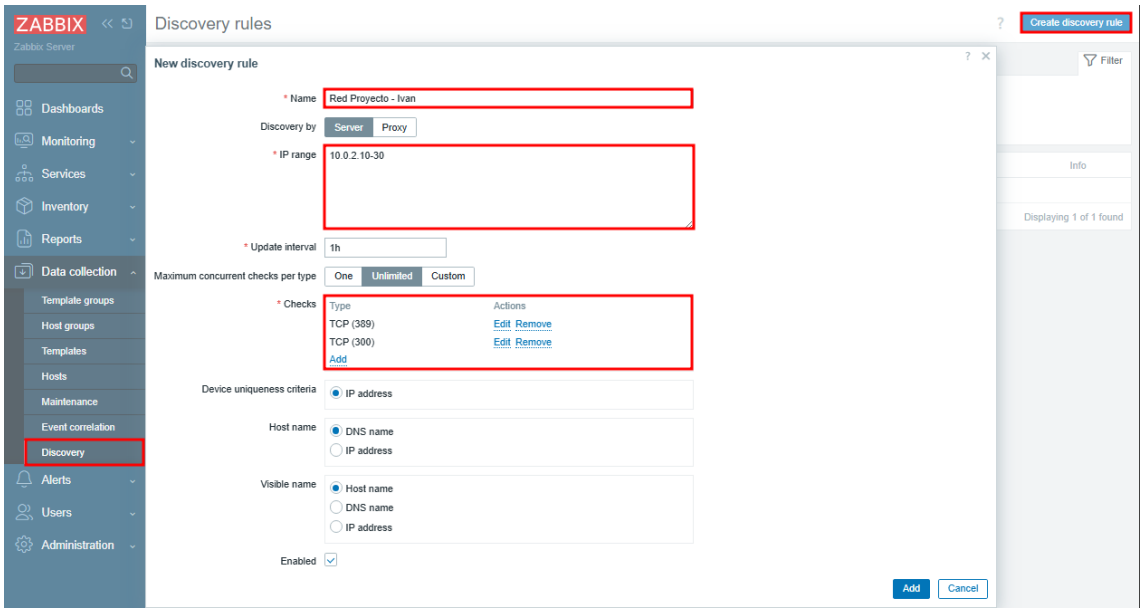


Ilustración 23: Configuración zabbix, reglas de descubrimiento de equipos

Una vez creada la primera regla de descubrimiento añadiremos otra para la red pública 10.0.1.0 y buscaremos el puerto 51820 (Wireguard)

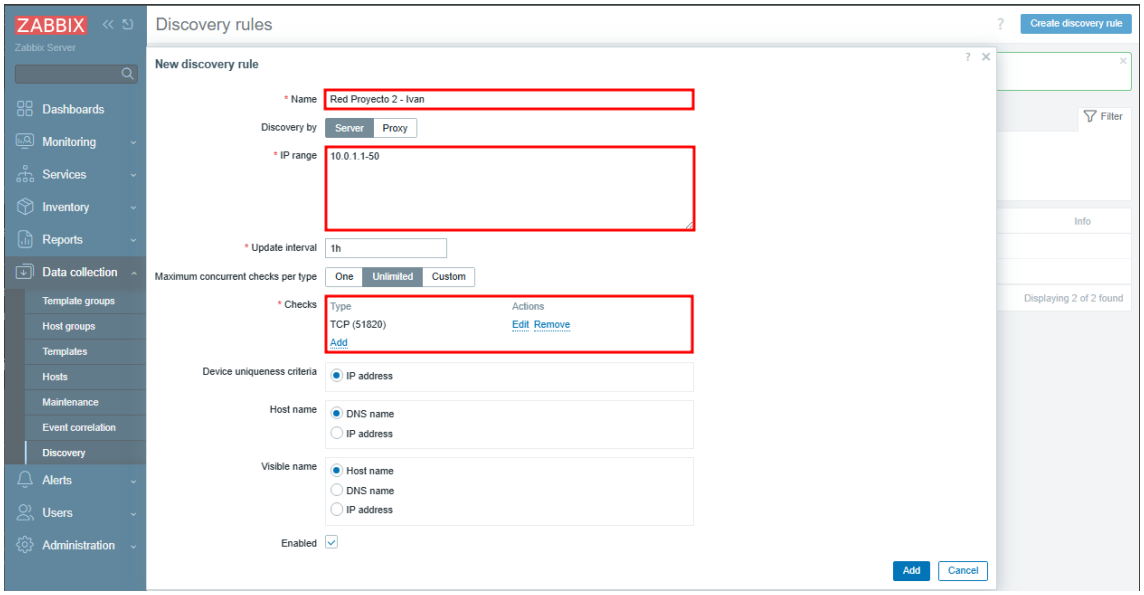


Ilustración 24: Configuración zabbix, reglas de descubrimiento de equipos 2

Para comprobar que las reglas de descubrimiento funcionan hay que ir a Monitoreo > Descubrir

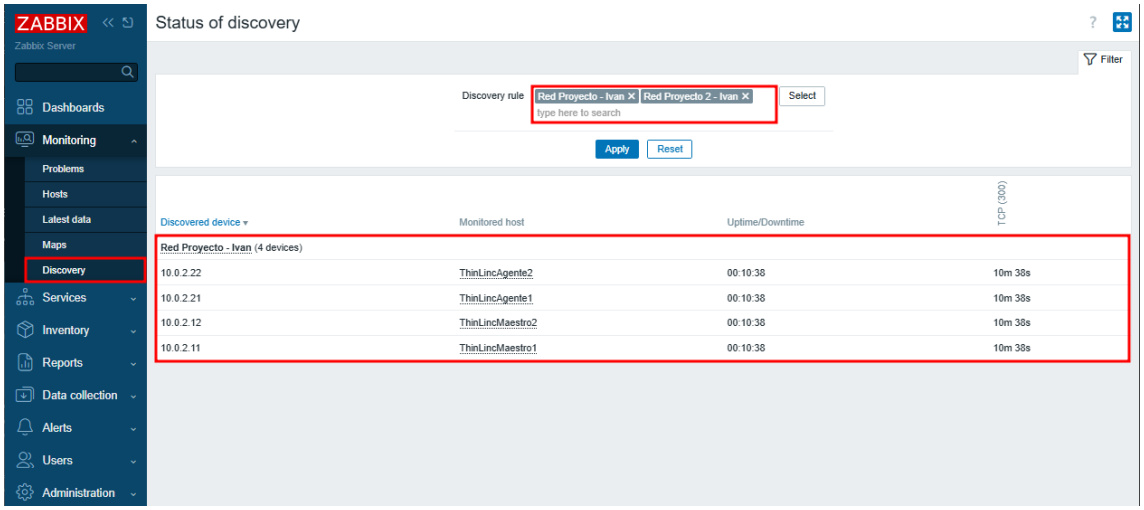


Ilustración 25: Configuración zabbix, comprobación de descubrimiento de equipos

Monitorear accesos por SSH

Crear monitores

Para crear los monitores hay que ir al apartado de Recopilación de datos > Equipos > Item > Añadir item, una vez en la creación del nuevo item le asignamos un nombre, un tipo y la función que va a hacer que en este caso es leer el archivo auth.log y comprobar si se ha completado con éxito el acceso:

log[/var/log/auth.log,Accepted,utf-8,100]

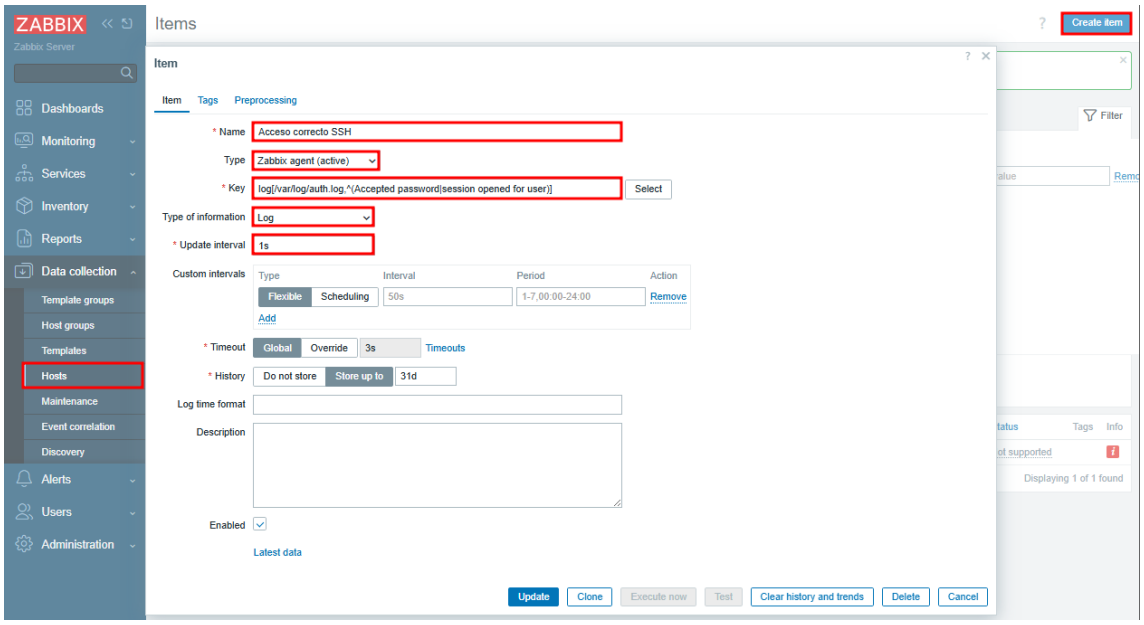


Ilustración 26: Configuración zabbix, creación de monitores

Una vez tenemos el monitor de acceso correcto ahora tenemos que hacer el de acceso incorrecto:

log[/var/log/auth.log,^(Failed password|.*authentication failure)]

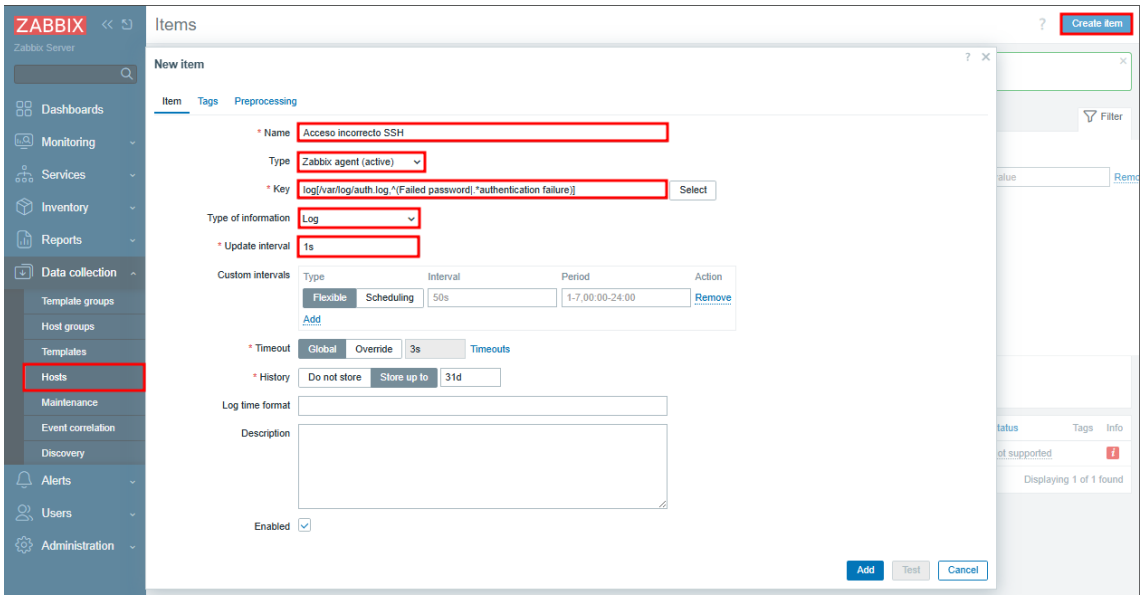


Ilustración 27: Configuración zabbix, creación de monitores 2

Configuración del LDAP

A continuación, se muestra la configuración del servicio LDAP tanto del servidor como del cliente

Configuración del servidor LDAP

Para la configuración del LDAP servidor hay que ejecutar el script subido en mi GitHub

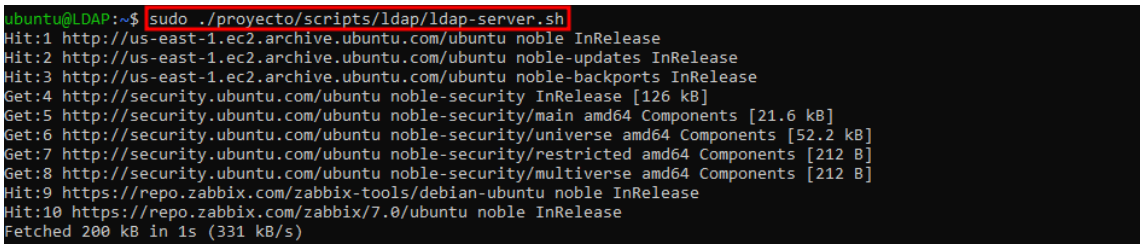


Ilustración 28: Comando ejecutar script de servidor LDAP

Una vez ejecutado el script van a ir saliendo una serie de pantallas moradas que tenemos que rellenar para completar la configuración, en este paso nos preguntan por la contraseña de administrador

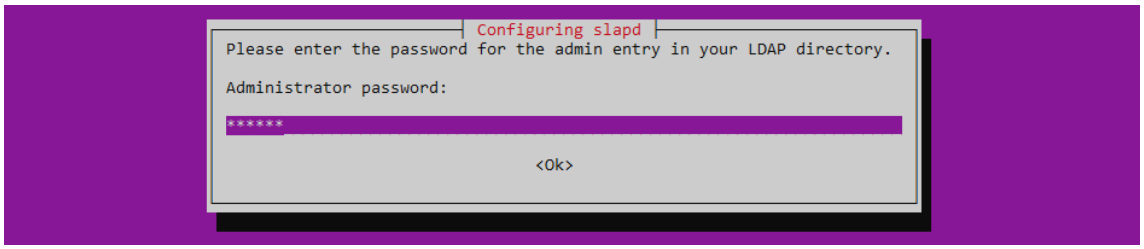


Ilustración 29: Configuración server LDAP, contraseña administradora

En la siguiente pantalla tenemos que confirmar la contraseña anteriormente puesta, nos irán preguntando periódicamente por la contraseña para ir aplicando configuraciones

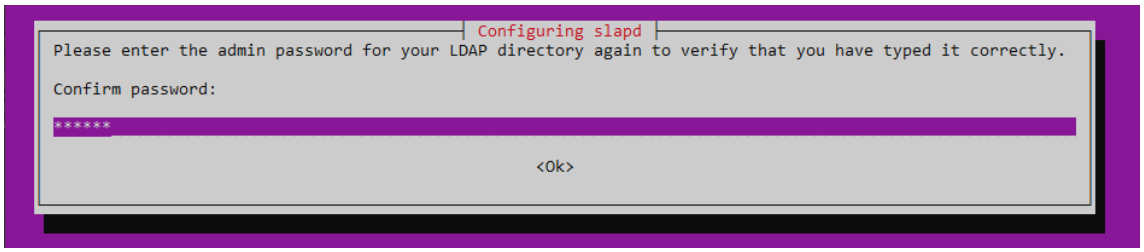


Ilustración 30: Configuración server LDAP, confirmar contraseña

Ahora seleccionamos que no nos haga una configuración inicial de la base de datos, ya que la vamos a hacer manualmente

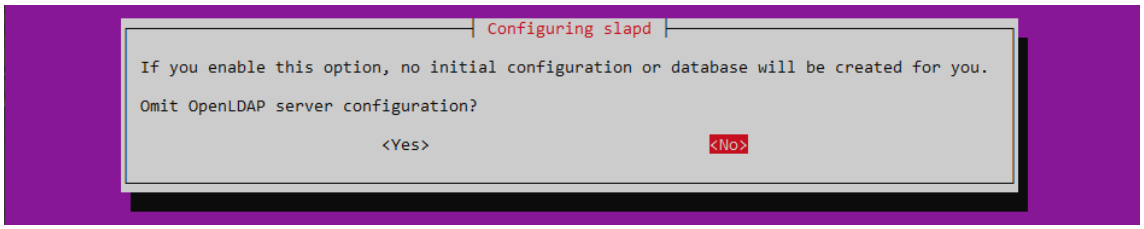


Ilustración 31: Configuración server LDAP, configuración inicial de la BDD

Aquí ponemos el DNS que utilizara nuestro servicio LDAP, que usaran los demás servidores para conectarse, en mi caso pondré LDAP.local

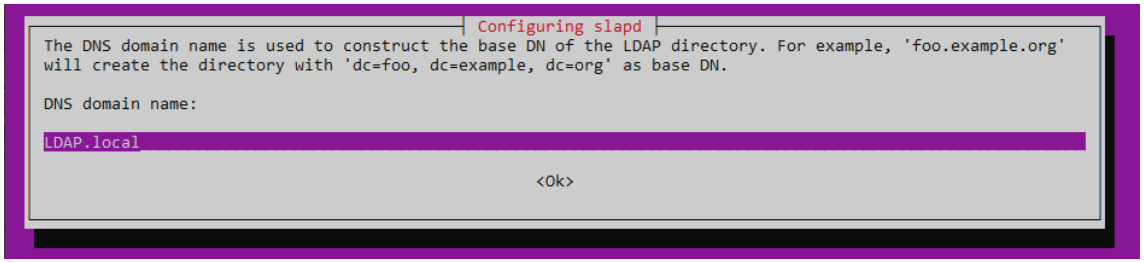


Ilustración 32: Configuración servidor LDAP, asignación de DNS

Después ponemos el nombre de la organización que tendrá el LDAP

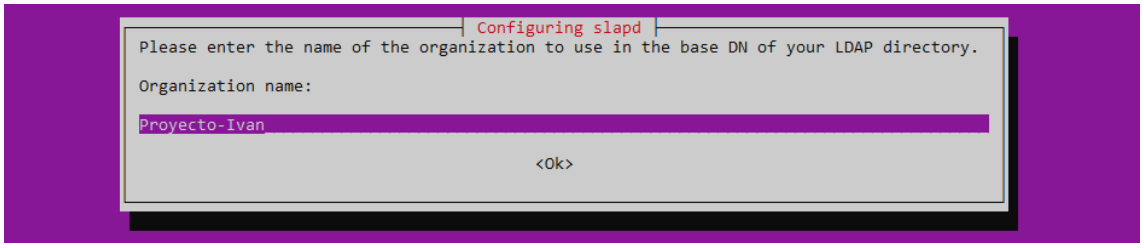


Ilustración 33: Configuración servidor LDAP, nombre de la organización

Ahora se están ejecutando unos scripts de creación de UOs y usuarios, para que se agreguen al LDAP hay que poner la contraseña de administrador que anteriormente configuramos

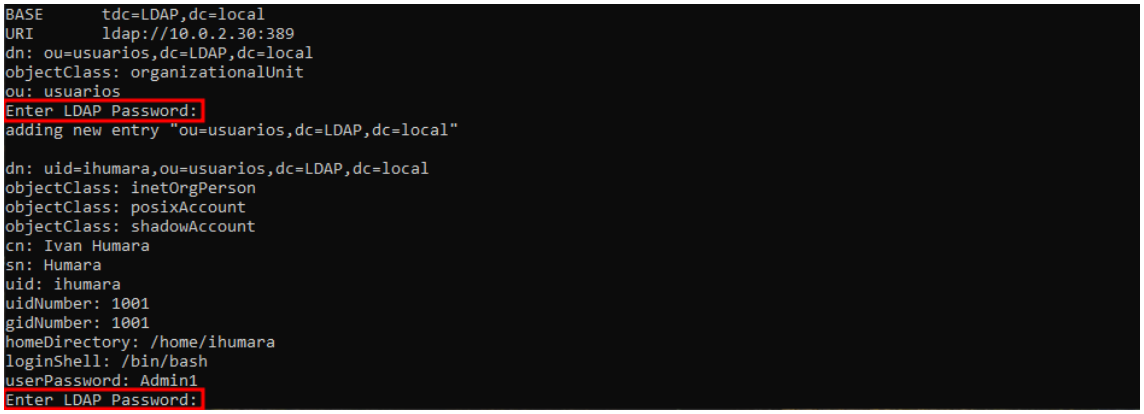


Ilustración 34: Configuración servidor LDAP, scripts de creación de UO y usuarios

Ahora ponemos la IP de nuestro servidor LDAP

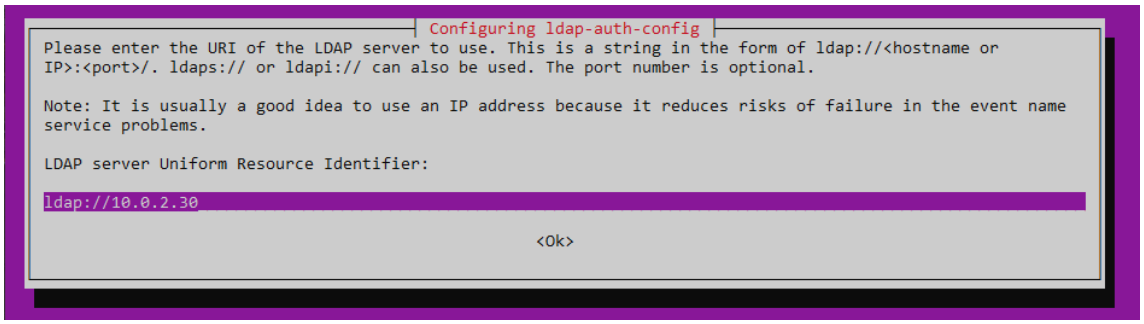


Ilustración 35: Configuración servidor LDAP, asignación de IP

Y por último nos saldrá una lista de opciones que podremos marcar y desmarcar, tendremos que marcar la primera opción, para que cuando un usuario inicie sesión por primera vez se le cree un directorio en el servidor donde se conecte

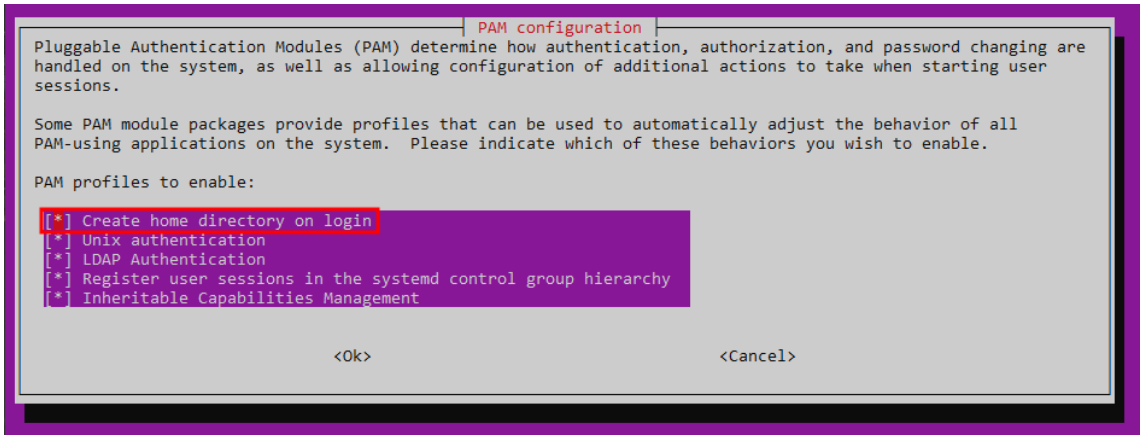


Ilustración 36: Configuración server LDAP, configuración PAM

Configuración de los clientes LDAP

Una vez tengamos configurado el servidor iremos a los servidores que actúan como cliente y ejecutaremos los scripts de configuración de clientes

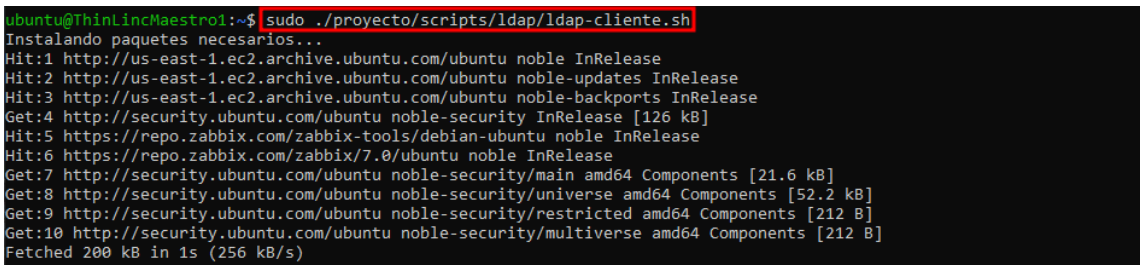


Ilustración 37: Comando ejecutar script de configuración del cliente LDAP

La configuración es muy similar a la del servidor, tendremos que poner y confirmar nuestra contraseña de administrador

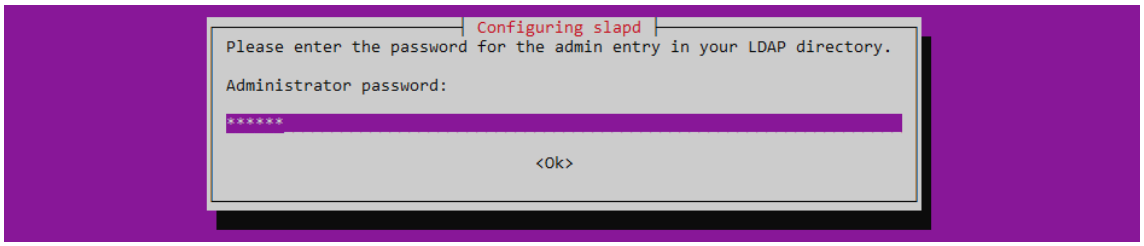


Ilustración 38: Configuración cliente LDAP, contraseña de administrador

Despues tenemos que poner la IP de nuestro servidor LDAP

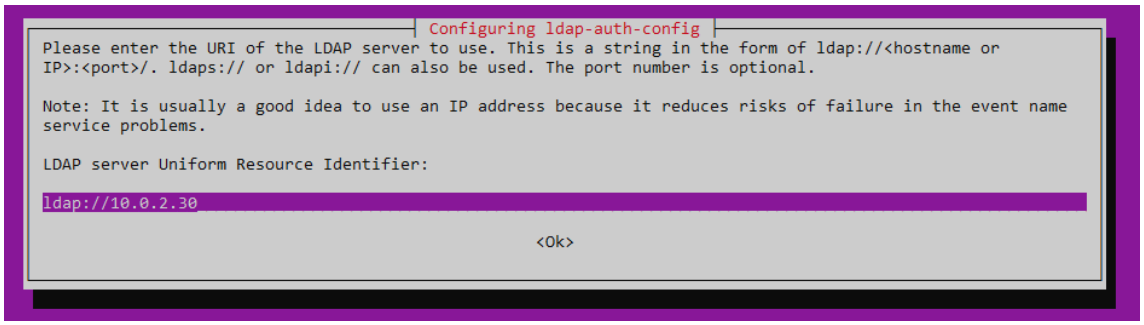


Ilustración 39: Configuración cliente LDAP, asignacion de IP

Y aquí tenemos que poner el DNS anteriormente agregado, es importante separar lo que va antes y después de la coma con “dc=”

Ilustración 40: Configuración cliente LDAP, asignar DNS del servidor

Seleccionamos la versión a usar que en mi caso serio la 3

Ilustración 41: Configuración cliente LDAP, versión del cliente LDAP

En esta pantalla pondremos que si para que solo el usuario root pueda crear la base de datos local

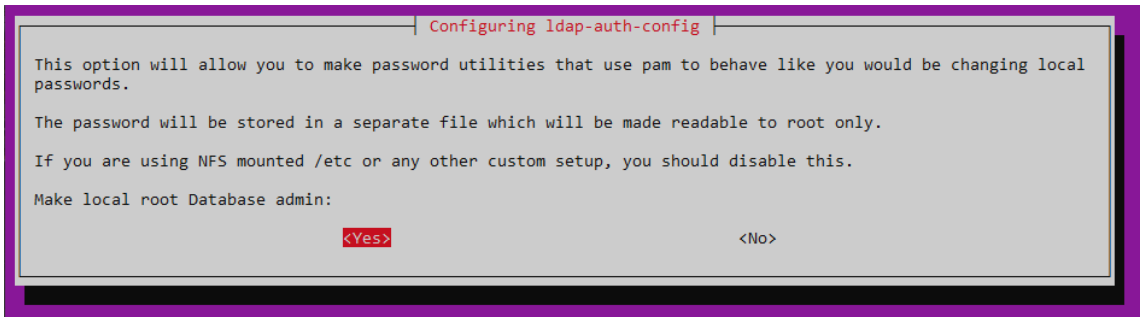


Ilustración 42: Configuración cliente LDAP, cambios en la BDD

Ahora podemos elegir si la base de datos de LDAP tiene login para acceder a los datos en mi caso pondré que no

Ilustración 43: Configuración cliente LDAP, cambios en la BDD 2

Este apartado es muy importante, hay que poner el DNS del servidor como se configuro anteriormente y también el usuario que se creó en los scripts de creación de UOs ..., el usuario es “admin”

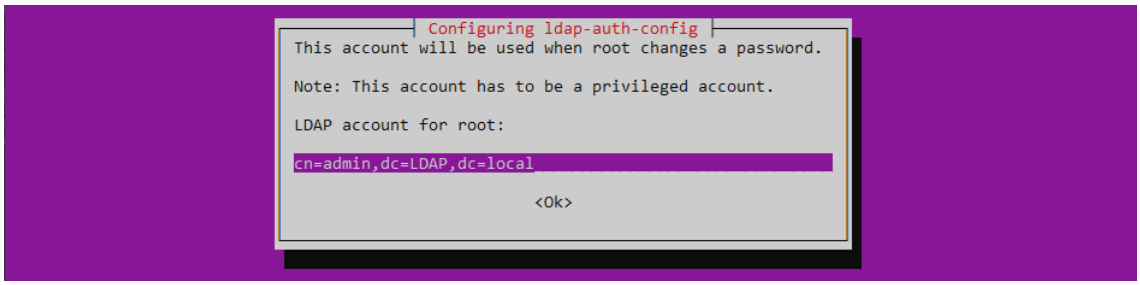


Ilustración 44: Configuración cliente LDAP, conexión con el servidor

Y por último en la lista seleccionamos la primera opción de crear el directorio al logearse en el servidor

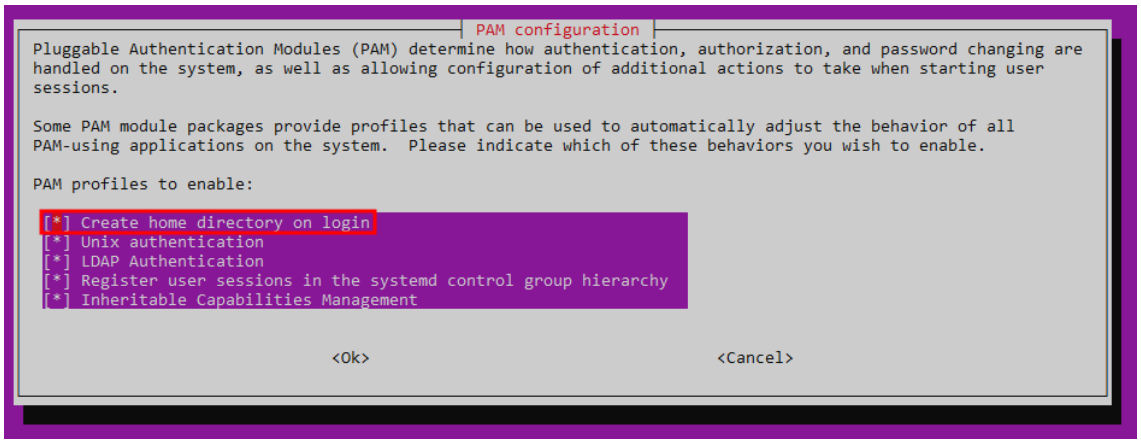


Ilustración 45: Configuración cliente LDAP, configuración de PAM

Esta configuración es replicable en el número de clientes que se necesite, no hay que cambiar ningún parámetro