Guía de Configuración de Portainer con Cloudflare Zero Trust

Prerequisitos

- Contenedor de Portainer ya creado y funcionando
- Túnel de Cloudflare configurado con SSH
- Dominio con nameservers apuntando a Cloudflare

1. Configuración del Túnel Cloudflare

Archivo de configuración inicial

Ubicación: `~/.cloudflared/config.yml`

```yaml
tunnel: fb6f53c9-f452-4da0-9001-33d96d2dd220
credentials-file: /home/devcris/.cloudflared/fb6f53c9-f452-4da0-9001-33d96d2dd220.json
ingress:
  - hostname: ssh.cristianalvis.com
    service: ssh://localhost:22
  - service: http_status:404
```

Configuración final con Portainer

```yaml
tunnel: fb6f53c9-f452-4da0-9001-33d96d2dd220
credentials-file: /home/devcris/.cloudflared/fb6f53c9-f452-4da0-9001-33d96d2dd220.json
ingress:
  - hostname: portainer.cristianalvis.com
    service: http://localhost:9000
  - hostname: ssh.cristianalvis.com
    service: ssh://localhost:22
  - service: http_status:404
```

Comandos para reiniciar el túnel

```bash
sudo systemctl restart cloudflared
sudo systemctl status cloudflared
```

2. Configuración DNS en Cloudflare

Agregar registro CNAME

- **Dashboard:** dash.cloudflare.com → DNS → Records
- **Type:** CNAME
- **Name:** portainer
- **Target:** fb6f53c9-f452-4da0-9001-33d96d2dd220.cfargotunnel.com
- **Proxy status:** Proxied (nube naranja activada)
- **TTL:** Auto

3. Configuración de Portainer

Comando para crear contenedor (puerto HTTP)

```bash
docker run -d -p 9000:9000 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v pc
```

Verificación del contenedor

```bash
docker ps | grep portainer
sudo netstat -tlnp | grep 9000
```

4. Configuración de Cloudflare Zero Trust

Crear organización Zero Trust

1. Ir a: one.dash.cloudflare.com
2. Elegir nombre del equipo (ej: cristianalvis)
3. Seleccionar plan gratuito (Zero Trust Free)
4. Completar proceso de pago (gratis)

Crear aplicación en Zero Trust

Navegación: Access → Applications → Add an Application

Basic Information

- **Application name:** portainer
- **Session Duration:** 24 hours

Public hostname

- **Subdomain:** portainer
- **Domain:** cristianalvis.com
- **Path:** (vacío)

Policy Configuration

- **Policy name:** Access Portainer
- **Action:** Allow
- **Rules:** Email (tu email personal)

5. Comandos de Verificación

Verificar servicios

```bash
# Estado del túnel
sudo systemctl status cloudflared

# Logs del túnel
sudo journalctl -u cloudflared -f

# Contenedores corriendo
docker ps

# Puertos en uso
sudo netstat -tlnp | grep 9000
```

Verificar DNS

```bash
# Verificar nameservers
nslookup -type=ns cristianalvis.com

# Verificar resolución del subdominio
nslookup portainer.cristianalvis.com
```

6. Resolución de Problemas Comunes

Error 502 Bad Gateway

Causa: Problemas de certificado HTTPS **Solución:** Usar HTTP interno y puerto 9000

"Client sent an HTTP request to an HTTPS server"

Causa: Mismatch entre HTTP/HTTPS **Solución:** Asegurar que Portainer use puerto 9000 (HTTP) y configuración use `http://localhost:9000`

Sin autenticación Zero Trust

Causa: Aplicación no configurada en Zero Trust **Solución:** Crear aplicación en Access → Applications

7. Acceso Final

URL de acceso

```
https://portainer.cristianalvis.com
```

Flujo de autenticación

1. Cloudflare solicita autenticación
2. Usuario se autentica (email, Google, etc.)
3. Zero Trust verifica política
4. Acceso concedido a Portainer

8. Configuraciones Adicionales

Para agregar más servicios

1. Crear contenedor en puerto específico
2. Agregar entrada al `config.yml`
3. Agregar DNS CNAME
4. Reiniciar túnel
5. (Opcional) Configurar Zero Trust para el nuevo servicio

Ejemplo para nueva aplicación

```yaml
```

```
# En config.yml
- hostname: mi-app.cristianalvis.com
  service: http://localhost:3000

# DNS CNAME
# Name: mi-app
# Target: fb6f53c9-f452-4da0-9001-33d96d2dd220.cfargotunnel.com
```

Notas Importantes

- El plan Zero Trust Free permite hasta 50 usuarios
- Los certificados se manejan automáticamente por Cloudflare
- El túnel se reinicia automáticamente con el sistema
- Los datos de Portainer persisten en el volumen `portainer_data`