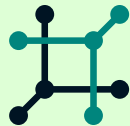


Despliegue de la aplicación Python de gestión de la infraestructura de red NetBox mediante la especificación ASGI y usando Caddy como proxy inverso

Proyecto Integrado de **Francisco Javier Huete**

Alumno del **I.E.S. Gonzalo Nazareno**
Tutorizado por **José Domingo Muñoz**



netbox



caddy[®]

Objetivos del proyecto

Objetivos del proyecto

Objetivo principal

- **Implantar la aplicación NetBox**
 - Entorno para implantar una app Python escrita con Django
 - Postgres y Redis
 - Servidor web

Objetivos del proyecto

Objetivos secundarios

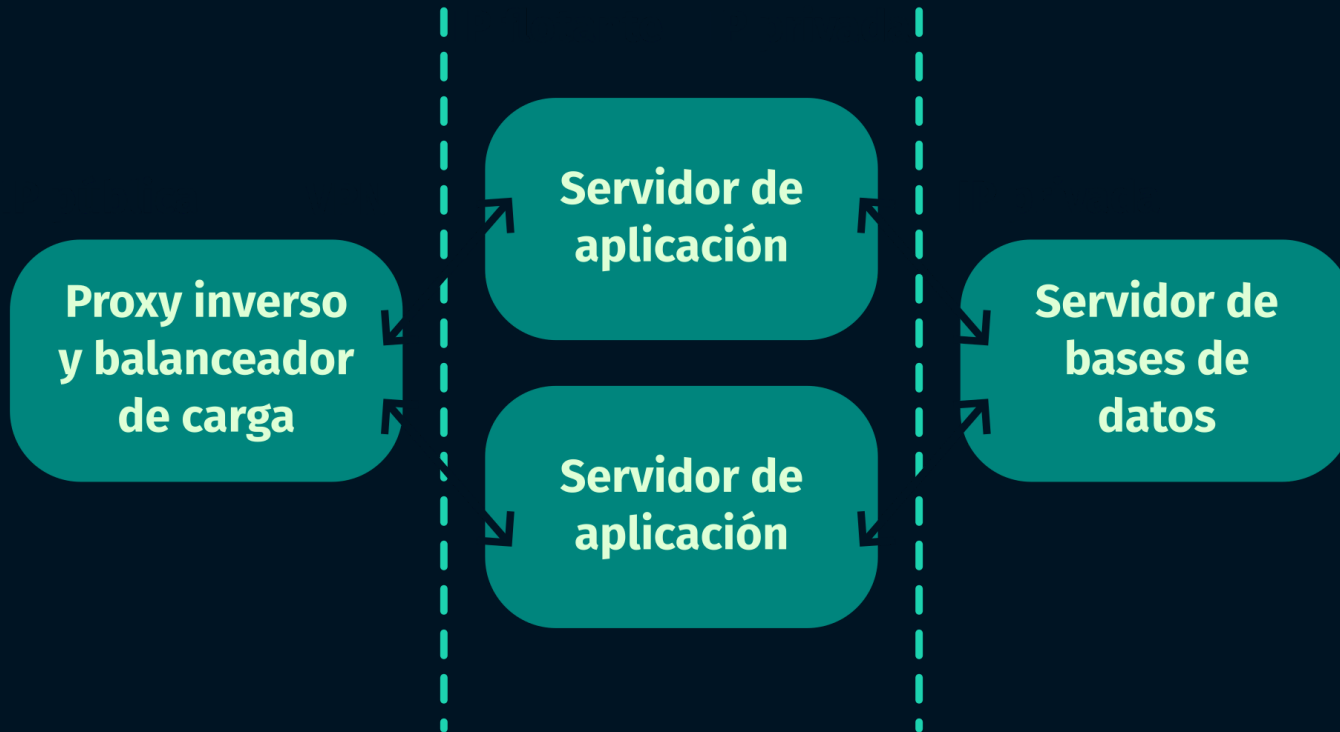
- Usar el protocolo ASGI
- Entender y demostrar el funcionamiento del servidor web Caddy

Escenario

Escenario

- OpenStack
 - Servidor de bases de datos (Postgres y Redis)
 - Dos servidores de aplicación
- VPS
 - Expuesto a Internet con Caddy
 - Conectado a los servidores de aplicación a través de una VPN

Escenario



Fundamentos teóricos

Fundamentos teóricos

NetBox

ASGI

Caddy

NetBox

- Aplicación escrita en Python usando el framework Django
- Solución para modelar y documentar redes

NetBox

IPAM

Gestión de direcciones IP

IPAM es una disciplina que se encarga de planificar, organizar, hacer seguimiento, auditar, monitorizar, ajustar la información y administrar las direcciones IP y las subredes usadas en una red de manera centralizada.

NetBox

DCIM

Gestión de la infraestructura en el centro de datos

Se trata de centralizar la monitorización, gestión y planificación de los sistemas críticos que constituyen los centros de datos.

Fundamentos teóricos

NetBox

ASGI

Caddy

ASGI

- Protocolo de Interfaz de Servidor Asíncrono
- Estándar de comunicación entre servidores web y aplicaciones Python
- Evolución del estándar WSGI
 - Tradicional en la comunicación entre servidores web y aplicaciones Python

ASGI

- Maneja operaciones asíncronas
- Permite la transición desde aplicaciones síncronas a nuevas versiones que incorporen elementos asíncronos
- Gestiona peticiones HTTP pero también HTTP/2 y WebSocket

ASGI

- Python 3.5
 - Incluye las palabras reservadas `async` y `await`
 - Se orienta al paradigma de programación asíncrona
- Se ponen de relieve las limitaciones de WSGI para aprovechar estas nuevas capacidades de Python

ASGI

WSGI vs ASGI

WSGI

Modelo de programación síncrono

Recibe las solicitudes de forma secuencial: bloquea su ejecución cada vez que recibe una petición

Limitación cuando las tareas requieren un tiempo de procesamiento largo

Falta de compatibilidad con algunos protocolos actuales

ASGI

Modelo de programación asíncrono

Procesa diferentes peticiones de manera paralela

Mejor capacidad de respuesta y concurrencia

Maneja mejor conexiones en tiempo real con protocolos

ASGI

WSGI vs ASGI

WSGI

Soporte para HTTP/1.1

Consumo más elevado de los recursos del sistema

Se puede usar un middleware síncrono

ASGI

Soporte para HTTP/2 y WebSockets

Menor consumo de recursos

El middleware debe soportar operaciones asíncronas

ASGI

Servidores ASGI

- **Uvicorn**
 - Rapidez, alto rendimiento y bajo consumo de recursos
- **Daphne**
 - Desarrollado por el proyecto Django Channels
 - Integración nativa con Django
- **Hypercorn**
 - Permite alcanzar la compatibilidad con HTTP/3
 - Flexibilidad: Compatible con ASGI y WSGI
- **Granian**
 - Servidor HTTP con soporte para ASGI y WSGI
- **Nginx Unit**
 - Solución integral para el despliegue a aplicaciones web

ASGI

Ventajas y desventajas

Ventajas

Mejor rendimiento

Operaciones simultáneas sin bloquear el proceso principal

Soporte nativo para protocolos más modernos

Mayor concurrencia de procesos

Ejecución del código tras haber enviado la respuesta

Desventajas

Requiere modificar el código de la aplicación

Mayor complejidad en la programación

La depuración del código puede ser más difícil y costosa

Incompatibilidades con WSGI

Puede producirse un cuello de botella en el mapeo relacional de

Fundamentos teóricos

NetBox

ASGI

Caddy

Caddy

- Servidor web de código abierto
- De reciente creación: lanzado en 2015
- Escrito en Go
 - Eficiencia en el uso de recursos del sistema
 - Mayor rendimiento
 - Alto nivel de portabilidad entre plataformas
 - Lenguaje "memory safe": implicaciones en cuanto a seguridad
 - Modelo de concurrencia: Caddy puede manejar múltiples peticiones de forma simultánea

Caddy

- **Objetivo: alojamiento web sencillo, fácil, automatizado y seguro**
- **Gestiona de forma automática los certificados SSL/TLS del dominio**
- **Enfoque minimalista en la configuración**
 - **Sintaxis simple y fácilmente legible**
- **Arquitectura modular**
 - **Comando**
 - **Librería centralizada**
 - **Módulos de configuración (plugins)**
- **Soporte nativo para las principales funciones de un servidor web**
 - **Proxy inverso**
 - **Balanceo de carga**
 - **Gestión de reintentos**
 - **Configuración de timeouts**
 - **Gestión de backends dinámicos**

Caddy

- Otras funciones
 - Permite verificar la salud de los backends a los que dirige el tráfico
 - Soporte para las versiones 2 y 3 de HTTP
 - Servidor de archivos integrado
 - Soporte multiplataforma
 - API de configuración

Métodos de configuración

Caddyfile

Fichero de configuración que utiliza directivas y bloques de sitio.

Directiva	Función
root	Indica la raíz del sitio
file_server	Sirve archivos estáticos
reverse_proxy	Configura un proxy inverso
log	Indica la ruta al fichero de log
rewrite	Reescribe las URL
php_fastcgi	Reenvía la petición a FastCGI para servir aplicaciones PHP
load_balance	Balancea la carga entre varios destinos

Métodos de configuración

JSON

- Automatizaciones o integraciones con herramientas de gestión de la configuración
- Adaptadores de configuración: convierten a JSON ficheros
 - Caddyfile
 - YAML
 - TOML
 - Configuración de Nginx
- Mayor flexibilidad y control
- Más adecuado para configuraciones complejas y automatizadas

Métodos de configuración

API RESTful

- Cargar, descargar y gestionar la configuración
- Sin reiniciar
- Facilita la integración con otras herramientas de orquestación
- Permite flujos de trabajo automatizados
- Gestión dinámica de la configuración
- Fundamental para flujos de integración continua

Caddy

Gestión de HTTPS y certificados

- Gestión automática
 - Protocolo ACME y Let's Encrypt
- Nombre de dominio
 - Caddy intenta resolver el desafío ACME automáticamente para demostrar el control sobre el dominio a Let's Encrypt
- Reproduce los pasos que se siguen manualmente con herramientas como `certbot`
- Redirige el tráfico del puerto 80 al 443 por defecto
- Ofrece el contenido por HTTPS con un certificado autofirmado si el dominio es `localhost`

Caddy

Ventajas y desventajas

Ventajas

Configuración simple e intuitiva

Gestión automática de HTTPS

Configuración dinámica a través de su API RESTful

Recarga de su configuración sin interrupciones en el servicio

Consumo mínimo de recursos

Desventajas

Comunidad más pequeña

Menos recursos para educación y resolución de problemas

Menor control de opciones de configuración

Características no soportadas

Menor rendimiento en solicitudes por segundo bajo carga extrema

Pasos para la realización del proyecto

Instalación de NetBox

Instalación del servidor ASGI

Configuración de Caddy

Instalación de NetBox

- Bases de datos
 - Postgres
 - Instalación
 - Configuración del acceso remoto
 - Creación de la base de datos
 - Redis
 - Instalación
 - Configuración del acceso remoto

```
#En /etc/redis/redis.conf  
bind 0.0.0.0  
...  
protected-mode no
```

Instalación de NetBox

- Instalación de dependencias
 - Python
 - Librerías
 - Git
- Clonación del repositorio de github

```
git clone https://github.com/netbox-community/netbox.git
```

- Creación del usuario de sistema

```
sudo adduser --system --group netbox
```

Configuración de NetBox

```
1 ALLOWED_HOSTS = ['*']
2
3 DATABASES = {
4     'ENGINE': 'django.db.backends.postgresql',
5     'NAME': 'netbox',
6     'USER': 'netbox',
7     'PASSWORD': '/zIwTLL+8TYmmgQwVJeU',
8     'HOST': '10.0.0.160',
9     'PORT': '5432',
10    'CONN_MAX_AGE': 300,
11 }
12
13 REDIS = {
14     'tasks': {
15         'HOST': '10.0.0.160',
16         'PORT': 6379,
17         'PASSWORD': '',
18         'DATABASE': 0,
19         'SSL': False,
20     },
21     'caching': {
```

Configuración de NetBox

```
1 ALLOWED_HOSTS = ['*']
2
3 DATABASES = {
4     'ENGINE': 'django.db.backends.postgresql',
5     'NAME': 'netbox',
6     'USER': 'netbox',
7     'PASSWORD': '/zIwTLL+8TYmmgQwVJeU',
8     'HOST': '10.0.0.160',
9     'PORT': '5432',
10    'CONN_MAX_AGE': 300,
11 }
12
13 REDIS = {
14     'tasks': {
15         'HOST': '10.0.0.160',
16         'PORT': 6379,
17         'PASSWORD': '',
18         'DATABASE': 0,
19         'SSL': False,
20     },
21     'caching': {
```

Configuración de NetBox

```
11     }
12
13 REDIS = {
14     'tasks': {
15         'HOST': '10.0.0.160',
16         'PORT': 6379,
17         'PASSWORD': '',
18         'DATABASE': 0,
19         'SSL': False,
20     },
21     'caching': {
22         'HOST': '10.0.0.160',
23         'PORT': 6379,
24         'PASSWORD': '',
25         'DATABASE': 1,
26         'SSL': False,
27     }
28 }
29
30 SECRET_KEY = "h=&#0bILs^0HArFbsRwr@Ws1Z+Ve10mC4ZxPP4nK8IE&WcCd5a"
```

Configuración de NetBox

```
12
13 REDIS = {
14     'tasks': {
15         'HOST': '10.0.0.160',
16         'PORT': 6379,
17         'PASSWORD': '',
18         'DATABASE': 0,
19         'SSL': False,
20     },
21     'caching': {
22         'HOST': '10.0.0.160',
23         'PORT': 6379,
24         'PASSWORD': '',
25         'DATABASE': 1,
26         'SSL': False,
27     }
28 }
29
30 SECRET_KEY = "h=&#0bILs^0HArFbsRwr@Ws1Z+Ve10mC4ZxPP4nK8IE&WcCd5a"
31
32 CSRF_TRUSTED_ORIGINS = ["https://proyecto.javihuete.site"]
```

Configuración de NetBox

```
12
13 REDIS = {
14     'tasks': {
15         'HOST': '10.0.0.160',
16         'PORT': 6379,
17         'PASSWORD': '',
18         'DATABASE': 0,
19         'SSL': False,
20     },
21     'caching': {
22         'HOST': '10.0.0.160',
23         'PORT': 6379,
24         'PASSWORD': '',
25         'DATABASE': 1,
26         'SSL': False,
27     }
28 }
29
30 SECRET_KEY = "h=&#0bILs^0HArFbsRwr@Ws1Z+Ve10mC4ZxPP4nK8IE&WcCd5a"
31
32 CSRF_TRUSTED_ORIGINS = ["https://proyecto.javihuete.site"]
```

Instalación de NetBox

- Script de Instalación
 - Crea entorno virtual
 - Instala requerimientos

```
pip install -r requirements.txt
```

- Migra el esquema de las bases de datos

```
python3 manage.py migrate
```

- Genera la documentación local
 - Agrega los ficheros estáticos al almacenamiento local

```
python3 manage.py collectstatic
```

Instalación de NetBox

- Creación del súper usuario

```
python3 manage.py createsuperuser
```

- Programación de tareas de limpieza con el script proporcionado por NetBox

```
sudo ln -s /opt/netbox/contrib/netbox-housekeeping.sh \  
/etc/cron.daily/netbox-housekeeping
```

- Prueba de funcionamiento de la aplicación

```
python3 manage.py runserver 0.0.0.0:8000 --insecure
```

Instalación del servidor ASGI

```
1 #Instalación de Daphne
2 source /opt/netbox/venv/bin/activate
3 pip install daphne
4
5 #Creación del fichero asgi.py
6 import os
7
8 from django.core.asgi import get_asgi_application
9
10 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'netbox.settings')
11
12 application = get_asgi_application()
13
14 #Configuración del demonio en /etc/systemd/system/netbox.service
15 ExecStart=/opt/netbox/venv/bin/daphne netbox.asgi:application --bind 127.0.0.1 --port 8080
16
17 #Recargar systemd
18 sudo systemctl daemon-reload
19
20 #Arrancar y habilitar los servicios de NetBox
21 sudo systemctl start netbox netbox.service
```

Instalación del servidor ASGI

```
1 #Instalación de Daphne
2 source /opt/netbox/venv/bin/activate
3 pip install daphne
4
5 #Creación del fichero asgi.py
6 import os
7
8 from django.core.asgi import get_asgi_application
9
10 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'netbox.settings')
11
12 application = get_asgi_application()
13
14 #Configuración del demonio en /etc/systemd/system/netbox.service
15 ExecStart=/opt/netbox/venv/bin/daphne netbox.asgi:application --bind 127.0.0.1 --port 8080
16
17 #Recargar systemd
18 sudo systemctl daemon-reload
19
20 #Arrancar y habilitar los servicios de NetBox
21 sudo systemctl start netbox netbox.service
```

Instalación del servidor ASGI

```
2 source /opt/netbox/venv/bin/activate
3 pip install daphne
4
5 #Creación del fichero asgi.py
6 import os
7
8 from django.core.asgi import get_asgi_application
9
10 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'netbox.settings')
11
12 application = get_asgi_application()
13
14 #Configuración del demonio en /etc/systemd/system/netbox.service
15 ExecStart=/opt/netbox/venv/bin/daphne netbox.asgi:application --bind 127.0.0.1 --port 8080
16
17 #Recargar systemd
18 sudo systemctl daemon-reload
19
20 #Arrancar y habilitar los servicios de NetBox
21 sudo systemctl start netbox netbox-rq
22 sudo systemctl enable netbox netbox-rq
```

Instalación del servidor ASGI

```
2 source /opt/netbox/venv/bin/activate
3 pip install daphne
4
5 #Creación del fichero asgi.py
6 import os
7
8 from django.core.asgi import get_asgi_application
9
10 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'netbox.settings')
11
12 application = get_asgi_application()
13
14 #Configuración del demonio en /etc/systemd/system/netbox.service
15 ExecStart=/opt/netbox/venv/bin/daphne netbox.asgi:application --bind 127.0.0.1 --port 8080
16
17 #Recargar systemd
18 sudo systemctl daemon-reload
19
20 #Arrancar y habilitar los servicios de NetBox
21 sudo systemctl start netbox netbox-rq
22 sudo systemctl enable netbox netbox-rq
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en los servidores de aplicación

```
1 :80 {
2   log {
3     output file /var/log/caddy/access.log
4     format filter {
5       wrap console
6       fields {
7         request>headers>Authorization delete
8       }
9     }
10  }
11  encode gzip
12  @excludeDirs {
13    not path /static/* /media/*
14  }
15  reverse_proxy @excludeDirs localhost:8001
16  file_server {
17    root /opt/netbox/netbox/
18  }
19 }
```

Configuración de Caddy en el servidor expuesto a Internet

```
1 proyecto.javihuete.site {
2     reverse_proxy 172.22.201.4:80 172.22.201.219:80 {
3         lb_policy ip_hash
4
5         health_uri /login/?next=/
6         health_interval 60s
7         health_timeout 3s
8         health_status 2xx
9     }
10 }
```

Configuración de Caddy en el servidor expuesto a Internet

```
1 proyecto.javihuete.site {
2     reverse_proxy 172.22.201.4:80 172.22.201.219:80 {
3         lb_policy ip_hash
4
5         health_uri /login/?next=/
6         health_interval 60s
7         health_timeout 3s
8         health_status 2xx
9     }
10 }
```

Configuración de Caddy en el servidor expuesto a Internet

```
1 proyecto.javihuete.site {
2     reverse_proxy 172.22.201.4:80 172.22.201.219:80 {
3         lb_policy ip_hash
4
5         health_uri /login/?next=/
6         health_interval 60s
7         health_timeout 3s
8         health_status 2xx
9     }
10 }
```

Demostraciones de funcionamiento y configuración