

Guía de Despliegue Completa del Orquestador en un VPS (Ubuntu 22.04)

Esta guía te llevará paso a paso a través del proceso de instalación y configuración de tu servidor orquestador en un VPS público.

Paso 0: Requisitos Previos

Un Servidor VPS: Un VPS con Ubuntu 22.04. Necesitarás la dirección IP pública y acceso root (o un usuario con privilegios sudo).

Un Nombre de Dominio (Recomendado): Aunque puedes usar la IP, un dominio es más profesional. Apunta tu dominio (ej: tu-dominio.com) o un subdominio (ej: api.tu-dominio.com) a la IP pública de tu VPS a través de los registros DNS de tu proveedor de dominio.

El archivo orchestrator.py: Ten a mano la última versión del código del orquestador en tu ordenador local.

Paso 1: Conexión y Actualización Inicial del Servidor

Conéctate a tu VPS a través de SSH. Reemplaza `tu_usuario` y `tu_ip_publica`.

```
ssh tu_usuario@tu_ip_publica
```

Una vez dentro, actualiza los paquetes del sistema:

```
sudo apt update && sudo apt upgrade -y
```

Paso 2: Instalar Dependencias del Sistema

Instala Python, su gestor de paquetes pip, la herramienta de entorno virtual y el servidor web NGINX.

```
sudo apt install python3 python3-pip python3-venv nginx -y
```

Paso 3: Preparar el Entorno de la Aplicación

Crearemos una carpeta dedicada para el proyecto y un entorno virtual de Python para mantener las dependencias aisladas.

Crea el directorio del proyecto:

```
sudo mkdir -p /var/www/helios_orchestrator
```

Crea un entorno virtual:

```
sudo python3 -m venv /var/www/helios_orchestrator/venv
```

Asigna permisos a tu usuario para gestionar los archivos:

```
sudo chown -R $USER:$USER /var/www/helios_orchestrator
```

Sube tu archivo orchestrator.py:

Desde tu ordenador local, usa el comando scp.

Ejecuta esto en tu máquina local, no en el VPS

```
scp /ruta/a/tu/orchestrator.py tu_usuario@tu_ip_publica:/var/www/helios_orchestrator/
```

Crea la carpeta uploads dentro del directorio del proyecto en el VPS:

Vuelve a la terminal de tu VPS

```
mkdir /var/www/helios_orchestrator/uploads
```

Paso 4: Instalar las Dependencias de Python

Ahora, instalaremos las librerías de Python necesarias dentro del entorno virtual.

Activa el entorno virtual:

```
source /var/www/helios_orchestrator/venv/bin/activate
```

(Verás (venv) al principio de la línea de tu terminal).

Instala las librerías:

```
pip install "fastapi[all]" gunicorn
```

Desactiva el entorno por ahora:

```
deactivate
```

Paso 5: Crear el Servicio systemd

Esto ejecutará nuestro orquestador como un servicio del sistema, asegurando que se inicie al arrancar y se reinicie si falla.

Crea y edita el archivo de servicio:

```
sudo nano /etc/systemd/system/helios-orchestrator.service
```

Copia y pega el contenido completo del archivo helios-orchestrator.service proporcionado en esta guía. Guarda (Ctrl+O, Enter) y sal (Ctrl+X).

Inicia y habilita el servicio:

```
sudo systemctl start helios-orchestrator
```

```
sudo systemctl enable helios-orchestrator
```

Comprueba su estado:

```
sudo systemctl status helios-orchestrator
```

(Deberías ver un mensaje en verde que dice "active (running)").

Paso 6: Configurar NGINX como Reverse Proxy

NGINX gestionará el tráfico de internet y lo reenviará de forma segura a nuestra aplicación.

Crea y edita el archivo de configuración de NGINX:

```
sudo nano /etc/nginx/sites-available/helios-orchestrator
```

Copia y pega el contenido completo del archivo `helios-orchestrator.nginx` proporcionado en esta guía. Importante: Reemplaza `tu_dominio.com` con tu nombre de dominio real. Si no tienes un dominio, usa tu dirección IP pública.

Habilita la configuración creando un enlace simbólico:

```
sudo ln -s /etc/nginx/sites-available/helios-orchestrator /etc/nginx/sites-enabled
```

Prueba la configuración de NGINX para buscar errores:

```
sudo nginx -t
```

(Debería decir "syntax is ok" y "test is successful").

Reinicia NGINX para aplicar los cambios:

```
sudo systemctl restart nginx
```

Paso 7: Configurar el Firewall

Permite que NGINX reciba tráfico.

```
sudo ufw allow 'Nginx Full'
```

```
sudo ufw enable
```

En este punto, ya deberías poder acceder a tu aplicación en `http://tu_dominio.com`.

Paso 8: (Muy Recomendado) Asegurar con HTTPS

Este último paso encripta toda la comunicación con tu servidor.

Instala Certbot, el cliente de Let's Encrypt:

```
sudo apt install certbot python3-certbot-nginx -y
```

Obtén e instala el certificado SSL. Reemplaza `tu_dominio.com` con tu dominio.

```
sudo certbot --nginx -d tu_dominio.com
```

Certbot te hará preguntas (tu email, aceptar los términos). Te preguntará si quieres redirigir el tráfico HTTP a HTTPS. Elige la opción 2 (Redirect).

¡Enhorabuena! Tu orquestador está ahora desplegado y funcionando de forma segura 24/7 en `https://tu_dominio.com`. Recuerda actualizar la variable `ORCHESTRATOR_PUBLIC_URL` en tu archivo `worker.py` antes de distribuirlo.