

# 【Sentry】安裝 Sentry (docker)

Sentry 是一個開源的錯誤追蹤系統，可用於收集應用程式中的例外與錯誤資訊。你可以透過 Docker 或直接在 Ubuntu 上安裝。以下是兩種方式的詳細步驟。

## □ 使用 Docker 安裝 Sentry

Sentry 官方推薦使用 Docker 來部署，方便快捷、維護簡單。

### 1. 安裝 Docker 與 Docker Compose（若尚未安裝）

```
sudo apt update
sudo apt install docker.io docker-compose -y
sudo systemctl enable docker
sudo systemctl start docker
```

### 2. 下載 Sentry 安裝腳本

```
git clone https://github.com/getsentry/self-hosted.git sentry-self-hosted
cd sentry-self-hosted
```

### 3. 設定環境變數

你可以修改 `.env` 或使用預設值即可。

### 4. 啟動安裝腳本

```
./install.sh
```

這個腳本會下載相關 images，初始化資料庫與設定等，需要一點時間（約 10-30 分鐘）。

### 5. 啟動服務

```
docker-compose up -d
```

### 6. 瀏覽器開啟 Sentry UI

打開 `http://<your-server-ip>:9000`（預設埠號是 `9000`），開始進行帳號註冊與專案設定。

## □ 在 Ubuntu 上原生安裝 Sentry（不建議，較複雜）

Sentry 的原生安裝比較繁瑣，不建議除非你有特別需求（如不使用 Docker 的環境）。以下是基本流程：

### 1. 安裝必要套件

```
sudo apt update
sudo apt install python3 python3-pip python3-venv redis postgresql libpq-dev build-essential -y
```

## 2. 建立 PostgreSQL 資料庫

```
sudo -u postgres psql
CREATE DATABASE sentry;
CREATE USER sentry WITH PASSWORD 'sentry';
GRANT ALL PRIVILEGES ON DATABASE sentry TO sentry;
\q
```

## 3. 安裝並設定虛擬環境

```
python3 -m venv sentry-venv
source sentry-venv/bin/activate
pip install --upgrade pip setuptools wheel
```

## 4. 安裝 Sentry

```
pip install sentry
```

## 5. 初始化 Sentry

```
sentry upgrade
```

這一步會要求你建立一個帳號。

## 6. 啟動服務

```
sentry run web
```

你可以另開終端機，啟動 worker 與 cron：

```
sentry run worker
sentry run cron
```

你可能還需要用 `nginx` 或 `gunicorn` 建立完整服務。

## □ 總結比較

方式	優點	缺點
Docker	快速、官方支援、易維護	需安裝 Docker
原生 Ubuntu	更靈活、自行控管細節	安裝繁瑣、難維護、容易踩雷

# Sentry CSRF Verification Failed

<https://github.com/getsentry/self-hosted/issues/2865>

## CSRF Verification Failed

A required security token was not found or was invalid.

If you're continually seeing this issue, try the following:

1. Clear cookies (at least for Sentry's domain).
2. Reload the page you're trying to submit (don't re-submit data).
3. Re-enter the information, and submit the form again.
4. Ensure the URL does not contain an extra "/" anywhere (eg: `https://foo//saml` -> `https://foo/saml`).

Read more about [CSRF on Wikipedia](#).

# 出現錯誤，請在以下檔案

sentry/sentry.conf.py

# 填入你機器真實ip:port

```
CSRF_TRUSTED_ORIGINS = ["sentry.mydomain.com", "127.0.0.1:9000"]
```

ex.

```
CSRF_TRUSTED_ORIGINS = ["10.10.0.1:9000"]
```

🕒 修訂版本 #2

★ 由 treeman 建立於 7 🕒🕒🕒 2025 14:06:02

✍ 由 treeman 更新於 15 🕒🕒🕒 2025 15:19:40