

Installer l'agent AppDynamics Python

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurer](#)

[Étape 1 : Installer l'agent et préparer le code](#)

[Étape 2 : Installation de Dockerfile](#)

[Étape 3 : Fichier de composition Docker](#)

[Vérifier](#)

[Étape 1 : Création et exécution](#)

[Étape 2 : Accéder au tableau de bord AppDynamics](#)

[Dépannage](#)

[Informations connexes](#)

[Besoin d'aide supplémentaire ?](#)

Introduction

Ce document décrit comment instrumenter une application Python Flask de base avec l'agent AppDynamics Python dans un conteneur Docker.

Conditions préalables

Exigences

- Docker et Docker Compose installés
- Un contrôleur AppDynamics (SaaS ou On-Prem)
- Identifiants d'accès (nom du compte, clé d'accès, hôte du contrôleur)
- Une licence AppDynamics qui inclut la prise en charge des agents Python
- Une application opérationnelle ou utilisez l'application présente dans le lien GitHub de la section Informations connexes

Composants utilisés

- Python : 3.9
- Agent AppDynamics Python : 25.6

- Image Docker de base : python : 3.9-slim (basé sur Debian)
- Plate-forme : linux/amd64 (pour assurer la compatibilité avec les binaires AppDynamics si vous utilisez une machine MAC ARM)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

L'agent AppDynamics Python instrumente votre application en lançant votre point d'entrée Python via un pyagent wrapper. L'agent collecte la télémétrie sur les demandes HTTP, les minutages de fonction et les erreurs, et les signale au contrôleur AppDynamics.

Configurer

Étape 1 : Installer l'agent et préparer le code

app.py

```
from flask import Flask
import os

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!";

if __name__ == "__main__":
    port = int(os.environ.get("PORT", 3000))
    app.run(debug=True, host='0.0.0.0', port=port)
```

appdynamics.cfg

```
[controller-info]
host=your-controller-url
port=443
ssl=true
account=your-account-name
accesskey=your-access-key
application=Hello_World_Python
tier=Hello_World_Python_Tier
node=Hello_World_Python_Node
```

Étape 2 : Installation de Dockerfile

```
FROM --platform=linux/amd64 python:3.9-slim
ENV APPD_AGENT_VERSION=25.6.0.7974
COPY . /app
WORKDIR /app
RUN chmod +x ./app.py
RUN pip install -U "appdynamics==${APPD_AGENT_VERSION}" -r requirements.txt
CMD pyagent run -c appdynamics.cfg -- python ./app.py
```

Étape 3 : Fichier de composition Docker

```
version: "2"

services:
  apps:
    build:
      context: .
    image: python-flask-appdynamics-demo
    platform: linux/amd64
    ports:
      - "3000:3000"
```

Vérifier

Étape 1 : Création et exécution

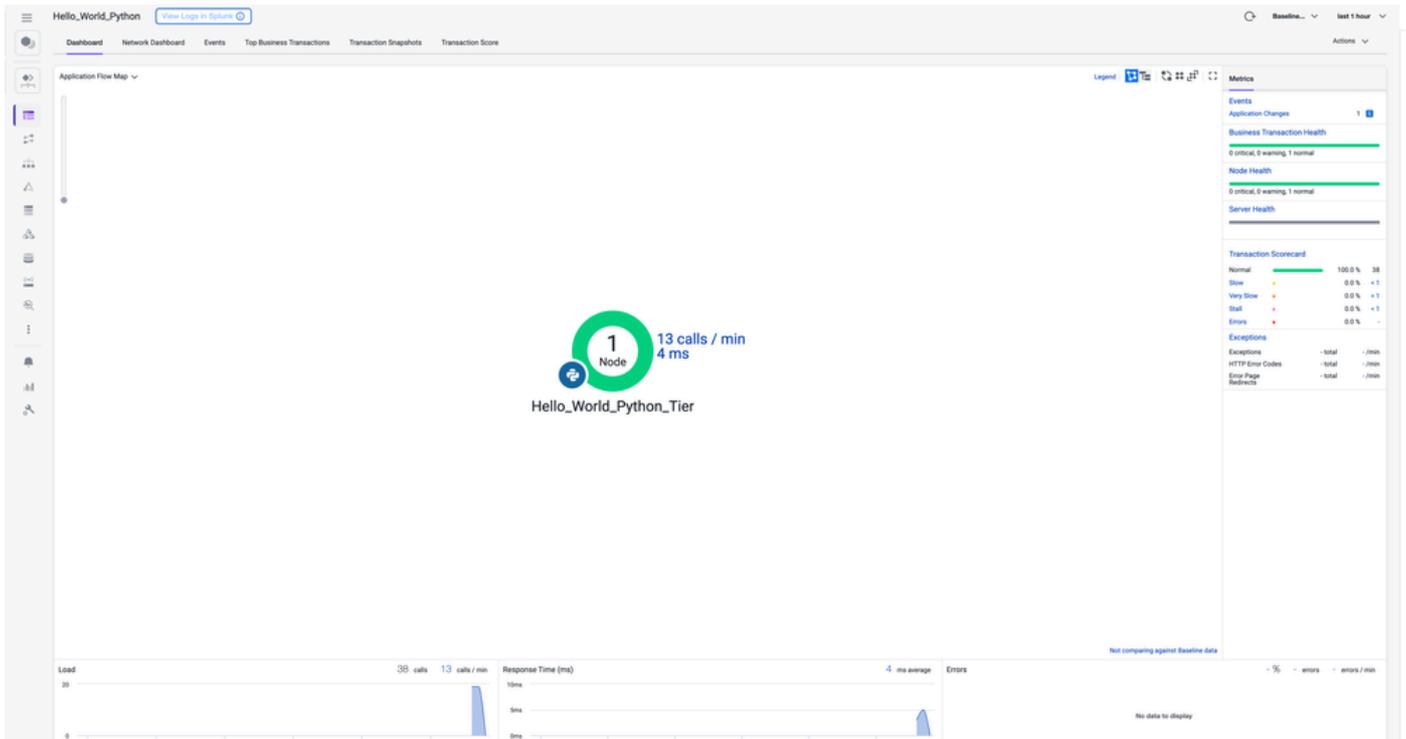
```
docker build --no-cache -t python-flask-appdynamics-demo .
```

```
docker run -p 3000:3000 python-flask-appdynamics-demo
```

- Appliquer la charge à l'application

Étape 2 : Accéder au tableau de bord AppDynamics

- Connectez-vous à votre AppDynamics Controller.
- Accédez à Applications > Hello_World_Python.
- Recherchez des mesures et des cartes de flux dans les 5 minutes.



Dépannage

Problème	Cause possible	Résolution
Application non affichée dans le tableau de bord	Clé d'accès ou informations de contrôleur incorrectes	Vérifiez les informations d'identification appdynamics.cfg
AppDynamics ne se connecte pas	pyagent non déclenché	Vérifiez si CMD utilise pyagent run ...

Informations connexes

[Documentation AppDynamics](#)

[Installation de l'agent Python](#)

[Démonstration AppDynamics de GitHub Python](#)

Besoin d'aide supplémentaire ?

Si vous avez une question ou rencontrez des problèmes, veuillez créer [un](#) ticket d'[assistance](#) avec ces détails :

- Error Details or Screenshot : fournissez un message d'erreur spécifique ou une capture

d'écran du problème.

- Commande utilisée : spécifiez la commande exacte que vous étiez en train d'exécuter lorsque le problème s'est produit.
- Journaux de l'agent Python : Fournissez les journaux de l'agent à partir de `/tmp/appd/*/logs/` s'ils sont générés.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.