

Installa agente AppDynamics Python

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Passaggio 1: Installare l'agente e preparare il codice](#)

[Passaggio 2: Installazione Dockerfile](#)

[Passaggio 3: Componi file Docker](#)

[Verifica](#)

[Passaggio 1: Compilazione ed esecuzione](#)

[Passaggio 2: Vai a AppDynamics Dashboard](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

[Ulteriori informazioni](#)

Introduzione

Questo documento descrive come instrumentare un'applicazione Python Flask di base con l'agente AppDynamics Python all'interno di un contenitore Docker.

Prerequisiti

Requisiti

- Docker e Docker Compose installati
- Un controller AppDynamics (SaaS o locale)
- Credenziali di accesso (Nome account, Chiave di accesso, Host controller)
- Una licenza di AppDynamics che include il supporto per gli agenti Python
- Applicazione funzionante o utilizzo dell'applicazione presente nel collegamento GitHub dalla sezione Informazioni correlate

Componenti usati

- Python 3.9
- Agente AppDynamics Python: 25.6

- Immagine Docker di base: python:3,9-slim (basato su Debian)
- Piattaforma: Linux/amd64 (per garantire la compatibilità con i file binari di AppDynamics se si utilizza un computer MAC ARM)

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

L'agente AppDynamics Python strumentalizza l'app avviando il punto di ingresso Python tramite un agente python wrapper. L'agente raccoglie dati di telemetria su richieste HTTP, intervalli di funzioni ed errori e li segnala al controller AppDynamics.

Configurazione

Passaggio 1: Installare l'agente e preparare il codice

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
```

Passaggio 2: Installazione 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
```

Passaggio 3: Componi file Docker

```
version: "2"

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

Verifica

Passaggio 1: Compilazione ed esecuzione

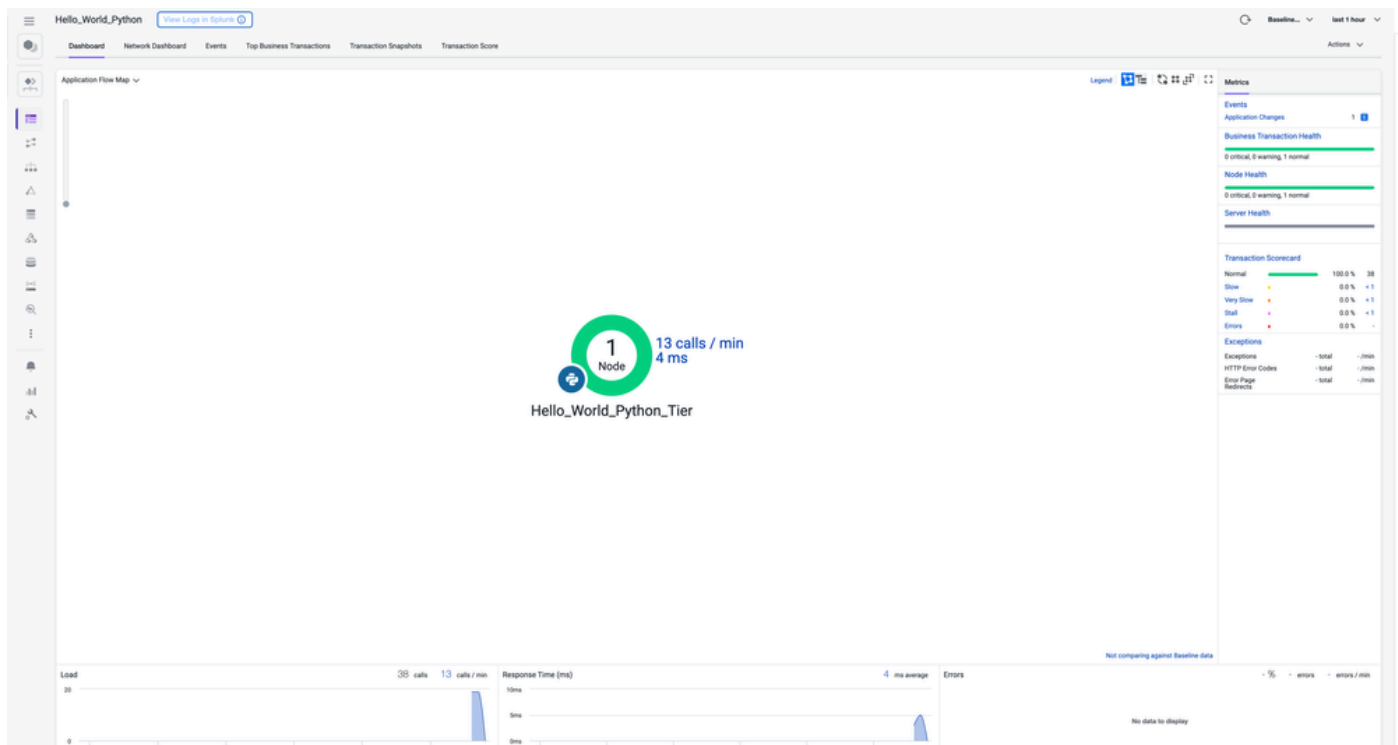
```
docker build --no-cache -t python-flask-appdynamics-demo .
```

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

- Applicazione del carico sull'applicazione

Passaggio 2: Vai a AppDynamics Dashboard

- Accedere a AppDynamics Controller.
- Selezionare Applicazioni > Hello_World_Python.
- Cercare metriche e mappe di flusso entro 5 minuti.



Risoluzione dei problemi

Problema	Possibile causa	Risoluzione
App non visualizzata nel dashboard	Chiave di accesso o informazioni sul controller errate	Controllare le credenziali di appdynamics.cfg
AppDynamics non registra	pyagent non attivato	Verificare se CMD utilizza l'esecuzione pyagent...

Informazioni correlate

[Documentazione di AppDynamics](#)

[Installare l'agente Python](#)

[Demo di GitHub Python AppDynamics](#)

Ulteriori informazioni

In caso di domande o di problemi, creare un biglietto di [assistenza](#) con i seguenti dettagli:

- Dettagli errore o schermata:fornire un messaggio di errore specifico o una schermata del problema.

- Comando utilizzato: specificare il comando in esecuzione quando si è verificato il problema.
- Log agente Python: fornire i log dell'agente da /tmp/appd/*/logs/, se generati.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).