

AppDynamics Python-Agent installieren

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Konfigurieren](#)

[Schritt 1: Installieren des Agenten und Vorbereiten von Code](#)

[Phase 2: Dockerfile einrichten](#)

[Schritt 3: Docker-Kompositionsdatei](#)

[Überprüfung](#)

[Schritt 1: Erstellung und Ausführung](#)

[Phase 2: Zum AppDynamics Dashboard wechseln](#)

[Fehlerbehebung](#)

[Zugehörige Informationen](#)

[Benötigen Sie weitere Unterstützung?](#)

Einleitung

In diesem Dokument wird beschrieben, wie eine einfache Python-Flask-Anwendung mit dem AppDynamics Python-Agent in einem Docker-Container instrumentiert wird.

Voraussetzungen

Anforderungen

- Docker und Docker Composite installiert
- Ein AppDynamics-Controller (SaaS oder vor Ort)
- Zugangsdaten (Kontoname, Zugriffsschlüssel, Controller-Host)
- Eine AppDynamics-Lizenz, die Python-Agent-Unterstützung enthält.
- Eine funktionierende Anwendung oder verwenden Sie die Anwendung in GitHub Link aus Related Information Abschnitt

Verwendete Komponenten

- Python: 3.9
- AppDynamics Python-Agent: 25.6

- Basisdocker-Bild: python:3.9-slim (Debian-basiert)
- Plattform: linux/amd64 (um die Kompatibilität mit AppDynamics-Binärdateien sicherzustellen, wenn Sie auf einem ARM-MAC-System ausgeführt werden)

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Der AppDynamics Python-Agent instrumentiert Ihre App, indem er Ihren Python-Einstiegspunkt über einen Wrapper pyagent startet. Der Agent erfasst Telemetriedaten zu HTTP-Anforderungen, Funktionsabläufen und Fehlern und meldet diese an den AppDynamics-Controller.

Konfigurieren

Schritt 1: Installieren des Agenten und Vorbereiten von 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
```

Phase 2: Dockerfile einrichten

```
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
```

Schritt 3: Docker-Kompositionsdatei

```
version: "2"

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

Überprüfung

Schritt 1: Erstellung und Ausführung

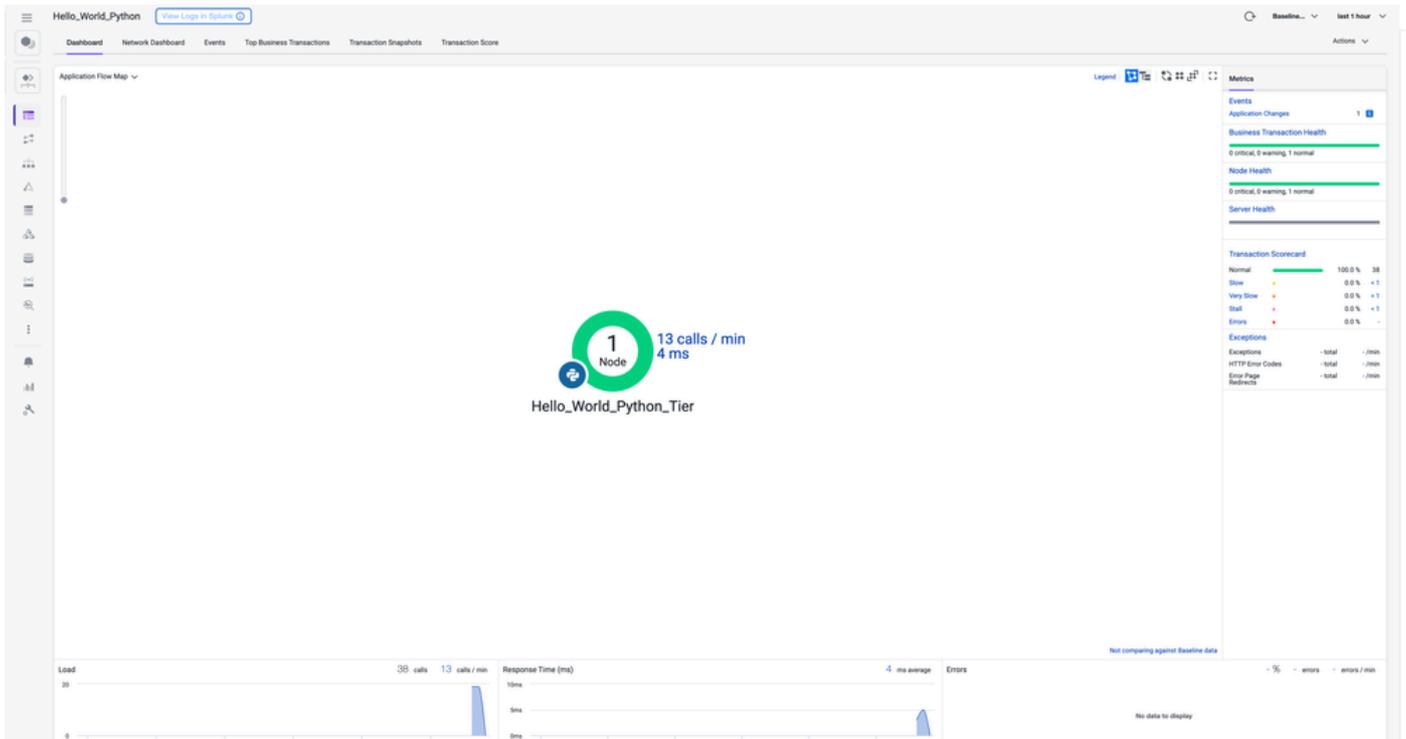
```
docker build --no-cache -t python-flask-appdynamics-demo .
```

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

- Auslastung der Anwendung

Phase 2: Zum AppDynamics Dashboard wechseln

- Melden Sie sich bei Ihrem AppDynamics Controller an.
- Navigieren Sie zu Anwendungen > Hello_World_Python.
- Suchen Sie innerhalb von 5 Minuten nach Metriken und Flow Maps.



Fehlerbehebung

Problem	Mögliche Ursache	Auflösung
App wird im Dashboard nicht angezeigt	Falscher Zugriffsschlüssel oder Controller-Informationen	Überprüfen Sie die Anmeldeinformationen von appdynamics.cfg
AppDynamics wird nicht protokolliert	Pyagent nicht ausgelöst	Überprüfen Sie, ob CMD pyagent run verwendet ...

Zugehörige Informationen

[AppDynamics-Dokumentation](#)

[Installation des Python-Agenten](#)

[GitHub Python AppDynamics - Demo](#)

Benötigen Sie weitere Unterstützung?

Wenn Sie Fragen haben oder Probleme auftreten, erstellen Sie ein [Support-Ticket](#) mit folgenden Informationen:

- Fehlerdetails oder Screenshot: Stellen Sie eine spezifische Fehlermeldung oder einen Screenshot des Problems bereit.
- Verwendeter Befehl: Geben Sie den genauen Befehl an, der ausgeführt wurde, als das Problem auftrat.
- Python-Agentenprotokolle: Geben Sie die Agentenprotokolle aus `/tmp/appd/*/logs/` an, falls generiert.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.