

AppDynamics Python Agentのインストール

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[設定](#)

[ステップ1: エージェントのインストールとコードの準備](#)

[ステップ2: Dockerfileの設定](#)

[ステップ3: Docker作成ファイル](#)

[確認](#)

[ステップ1: 構築と実行](#)

[ステップ2: AppDynamicsダッシュボードに移動](#)

[トラブルシューティング](#)

[関連情報](#)

[サポートが必要な場合](#)

はじめに

このドキュメントでは、Dockerコンテナ内のAppDynamics Python Agentを使用して基本的なPython Flaskアプリケーションをインストールメント化する方法について説明します。

前提条件

要件

- DockerおよびDocker Composeがインストールされている
- AppDynamicsコントローラー (SaaSまたはオンプレミス)
- アクセスクレデンシャル (アカウント名、アクセスキー、コントローラホスト)
- Pythonエージェントサポートを含むAppDynamicsライセンス
- GitHubリンクに存在するアプリケーションを使用するか、関連情報セクションから

使用するコンポーネント

- Python: 3.9
- AppDynamics Python Agent: 25.6
- Base Dockerイメージ : python:3.9-slim (Debianベース)

- プラットフォーム：linux/amd64（ARM MACマシンで実行している場合にAppDynamics/バイナリとの互換性を確保するため）

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

AppDynamics Pythonエージェントは、ラッパーのPythonエージェントを介してPythonエントリポイントを開始することで、アプリケーションを計測します。エージェントは、HTTP要求、関数のタイミング、およびエラーに関するテレメトリを収集し、AppDynamicsコントローラーに報告します。

設定

ステップ1：エージェントのインストールとコードの準備

アプリケーション.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
```

ステップ2: Dockerfileの設定

```
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
```

ステップ3: Docker作成ファイル

```
version: "2"

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

確認

ステップ1: 構築と実行

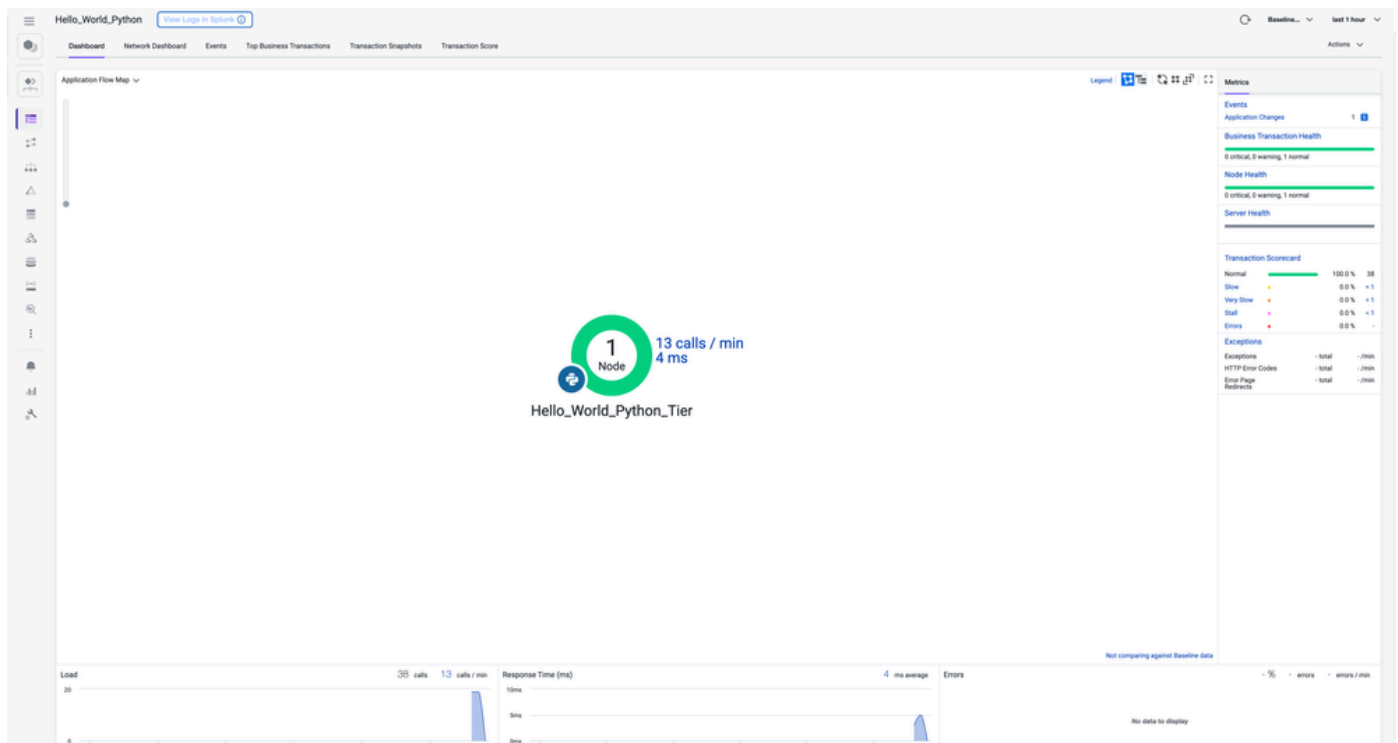
```
docker build --no-cache -t python-flask-appdynamics-demo .
```

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

- アプリケーションに負荷をかける

ステップ2: AppDynamicsダッシュボードに移動

- AppDynamicsコントローラーにログインします。
- Applications > Hello_World_Pythonに移動します。
- 5分以内にメトリックとフロアマップを検索します。



トラブルシューティング

問題	考えられる原因	解決方法
アプリケーションがダッシュボードに表示されない	不正なアクセスキーまたはコントローラ情報	appdynamics.cfgの資格情報を確認してください
AppDynamicsがログに記録されていません	pyagentがトリガーされない	CMDがpyagent runを使用しているかどうかを確認します...

関連情報

[AppDynamicsドキュメント](#)

[Python Agentのインストール](#)

[GitHub Python AppDynamicsデモ](#)

サポートが必要な場合

質問がある場合、または問題が発生した場合は、次の詳細情報を含む[asupport](#) チケットを作成してください。

- エラーの詳細またはスクリーンショット：特定のエラーメッセージまたは問題のスクリーン

ショットを提供します。

- 使用コマンド：問題が発生したときに実行していたコマンドを正確に指定します。
- Python agent logs :/tmp/appd/*/logs/からエージェントログ（生成された場合）を提供します。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。