

AppDynamics Python 에이전트 설치

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[1단계: 에이전트 설치 및 코드 준비](#)

[2단계: Dockerfile 설정](#)

[3단계: Docker 작성 파일](#)

[다음을 확인합니다.](#)

[1단계: 빌드 및 실행](#)

[2단계: AppDynamics 대시보드로 이동](#)

[문제 해결](#)

[관련 정보](#)

[추가 지원이 필요하십니까?](#)

소개

이 문서에서는 Docker 컨테이너 내에서 AppDynamics Python Agent를 사용하여 기본 Python Flask 응용 프로그램을 계측하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

- Docker 및 Docker 작성 설치됨
- AppDynamics 컨트롤러(SaaS 또는 온프레미스)
- 액세스 자격 증명(계정 이름, 액세스 키, 컨트롤러 호스트)
- Python 에이전트 지원을 포함하는 AppDynamics 라이선스
- Related Information 섹션의 GitHub 링크에 있는 응용 프로그램 또는 응용 프로그램 사용

사용되는 구성 요소

- 파이썬: 3.9
- AppDynamics Python 에이전트: 25.6
- 기본 Docker 이미지: python:3.9-슬림(데비안 기반)

- 플랫폼: linux/amd64(ARM MAC 컴퓨터에서 실행 중인 경우 AppDynamics 바이너리와 호환성 보장)

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

AppDynamics Python 에이전트는 래퍼 pyagent를 통해 Python 진입점을 시작하여 앱을 계측합니다. 에이전트는 HTTP 요청, 기능 타이밍 및 오류에 대한 텔레메트리를 수집하고 이를 AppDynamics 컨트롤러에 보고합니다.

구성

1단계: 에이전트 설치 및 코드 준비

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
```

2단계: Dockerfile 설정

```
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
```

3단계: Docker 작성 파일

```
version: "2"

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

다음을 확인합니다.

1단계: 빌드 및 실행

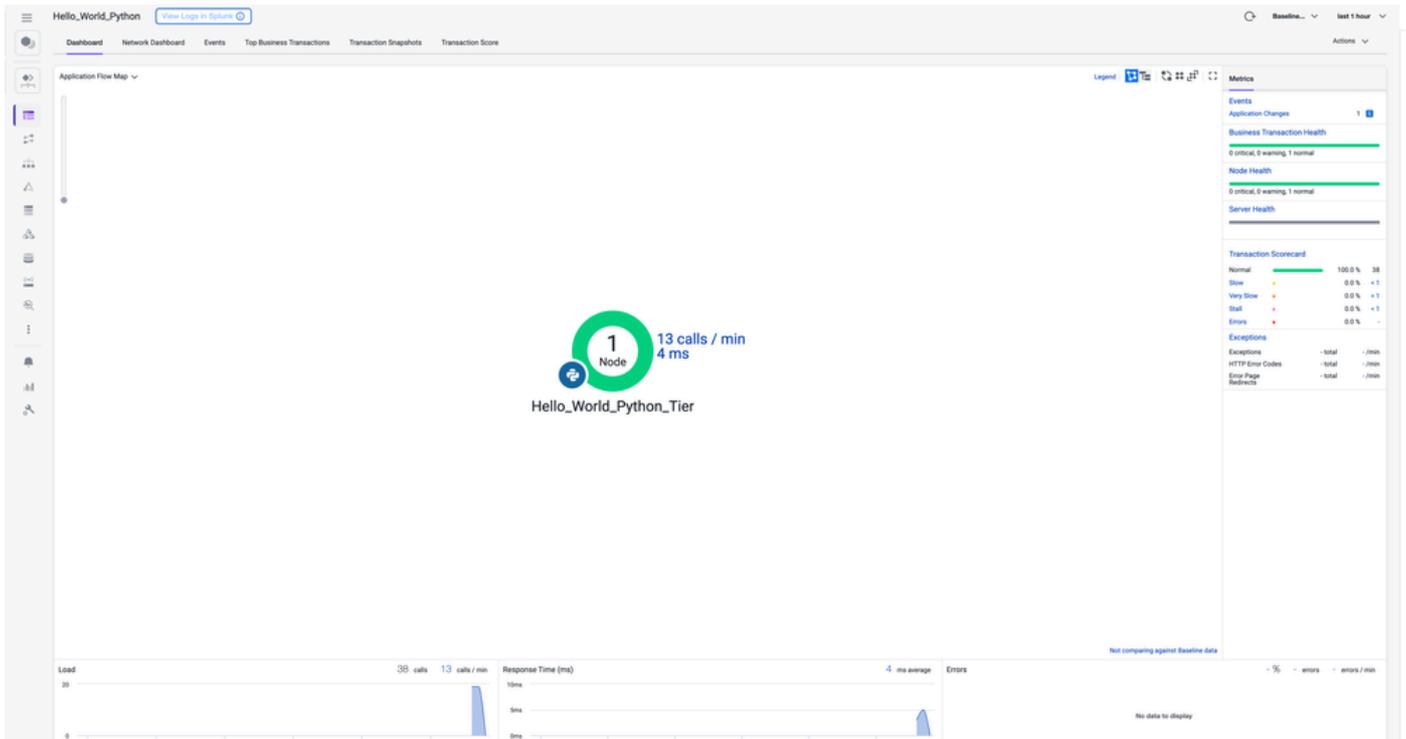
```
docker build --no-cache -t python-flask-appdynamics-demo .
```

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

- 응용 프로그램에 부하 적용

2단계: AppDynamics 대시보드로 이동

- AppDynamics 컨트롤러에 로그인합니다.
- Applications(애플리케이션) > Hello_World_Python으로 이동합니다.
- 5분 내에 메트릭 및 플로우 맵을 찾습니다.



문제 해결

문제	가능한 원인	해결
대시보드에 앱이 표시되지 않음	잘못된 액세스 키 또는 컨트롤러 정보	appdynamics.cfg 자격 증명 확인
AppDynamics가 기록되지 않음	pyagent가 트리거되지 않음	CMD에서 pyagent run을 사용하고 있는지 확인 ...

관련 정보

[AppDynamics 설명서](#)

[Python Agent 설치](#)

[GitHub Python AppDynamics 데모](#)

추가 지원이 필요하십니까?

질문이 있거나 문제가 있는 경우 다음 세부 정보가 포함된 [지원](#) 티켓을 만드십시오.

- 오류 세부사항 또는 스크린샷: 특정 오류 메시지 또는 문제의 스크린샷을 제공하십시오.
- 사용된 명령: 문제가 발생했을 때 실행 중이던 정확한 명령을 지정합니다.

- Python 상담원 로그: /tmp/appd/*/logs/(생성된 경우)의 상담원 로그를 제공합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.