

# AppDynamics Node.js 에이전트 설치

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[1단계: 에이전트 설치](#)

[2단계: 코드에 구성 포함](#)

[3단계: 애플리케이션 로직 추가](#)

[다음을 확인합니다.](#)

[1단계: 앱 시작](#)

[2단계: AppDynamics 대시보드로 이동](#)

[문제 해결](#)

[관련 정보](#)

[추가 지원이 필요하십니까?](#)

---

## 소개

이 문서에서는 AppDynamics Node.js 에이전트를 설치 및 구성하여 Node.js 응용 프로그램에서 성능 메트릭을 모니터링하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

설치를 계속하려면 다음을 충족해야 합니다.

- Node.js 및 Express 프레임워크에 대한 친숙도
- Related Information 섹션의 GitHub 링크에 있는 응용 프로그램 또는 응용 프로그램 사용
- 환경 변수 및 애플리케이션 구조에 대한 기본 이해
- AppDynamics SaaS 또는 온프레미스 컨트롤러에 액세스
- API 액세스 자격 증명(계정 이름, 액세스 키, 컨트롤러 호스트/포트)

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 버전을 기반으로 합니다.

- Node.js v18.x
- AppDynamics Node.js Agent 24.x
- AppDynamics 컨트롤러(온프레미스 또는 SaaS)
- 관련 정보 섹션에 완전한 애플리케이션이 있는 GitHub 링크가 있습니다

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 Node.js 에이전트를 사용하면 Node.js 응용 프로그램을 계측하고 모니터링할 수 있습니다. 비즈니스 트랜잭션 데이터, 백엔드 응답 시간, 메모리 사용량, CPU 메트릭을 캡처하여 AppDynamics Controller로 전송합니다.

메트릭을 올바르게 수집하려면 다른 사용자 애플리케이션 모듈이 로드되기 전에 에이전트를 초기화해야 합니다. 에이전트는 HTTP(S)를 통해 컨트롤러와 통신하며 올바르게 작동하려면 컨트롤러 호스트, 포트 및 액세스 키 세부사항이 필요합니다.

## 구성



팁: 구성 조각에서 모든 변수를 실제 AppDynamics 컨트롤러 세부 정보로 바꾸십시오.

### 1단계: 에이전트 설치

NPM을 사용하여 AppDynamics Node.js 에이전트를 설치합니다.

```
npm install appdynamics --save
```

### 2단계: 코드에 구성 포함

주 파일(app.js 또는 index.js)의 맨 위에 다른 모듈 앞에 코드를 추가합니다.

```
require("appdynamics").profile({
  controllerHostName: 'ControllerURL',
  controllerPort: ControllerPort,
  controllerSslEnabled: true, // Set to true if controllerPort is SSL
  accountName: 'accountName',
  accountAccessKey: 'accountAccessKey', // required
```

```
    applicationName: 'Hello_World_NodeJS', // Change the name of application
    tierName: 'Hello_World_NodeJS_Tier', // Change the name of tier
    nodeName: 'Hello_World_NodeJS_Node', //// Change the name of node
  });
```

### 3단계: 애플리케이션 로직 추가

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

## 다음을 확인합니다.

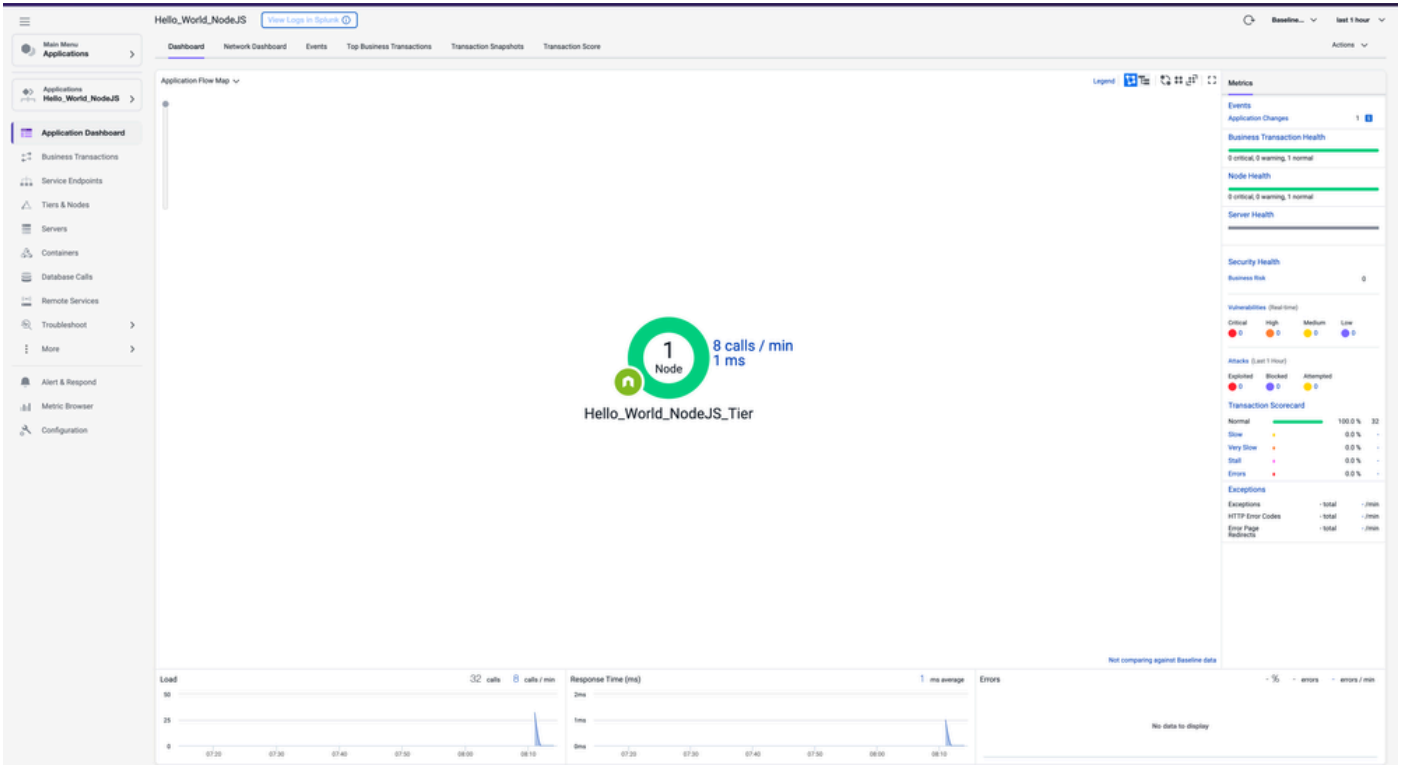
### 1단계: 앱 시작

```
node app.js
```

- 응용 프로그램에 부하 적용

### 2단계: AppDynamics 대시보드로 이동

- AppDynamics 컨트롤러에 로그인
- Applications(애플리케이션) > Hello\_World\_NodeJS로 이동합니다.
- 5분 이내에 메트릭 및 플로우 맵 검색



## 문제 해결

문제	가능한 원인	해결
앱이 대시보드에 표시되지 않음	앱 코드 이전에 에이전트가 초기화되지 않았습니다.	require('appdynamics')가 첫 줄인지 확인합니다.
에이전트 설치 실패	사용 권한이 없거나 인터넷 또는 호환되지 않는 OS	높은 권한으로 실행하거나 지원되는 플랫폼을 사용하는지 확인
401 액세스 오류	계정 액세스 키 또는 계정 이름이 잘못되었습니다.	AppDynamics 컨트롤러에서 자격 증명 다시 확인

## 관련 정보

[AppDynamics 설명서](#)

[Node.js 에이전트 설치](#)

[GitHub Node.js 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.
- 노드 에이전트 로그: /tmp/appd/\*/logs/(생성된 경우)의 에이전트 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.