# Instrumentation with AppDynamics Hybrid java agent

## Architecture

A diagram of a company

Description automatically generated

## Lab

In this lab we will configure the hybrid mode in the AppDynamics agent. This will make the agent send information to AppDynamics and to Cisco Observability Platform at the same time. Including Logs!!

The first step is to download AppDynamics Hybrid Agent from the link below.

<https://accounts.appdynamics.com/downloads>

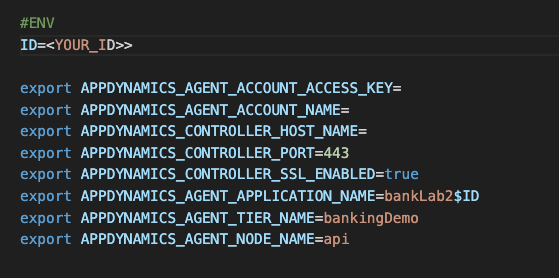
Unzip the agent and name the folder *appd\_agent* insidethe *lab2\_appd\_hybrid\_agent* folder.

This a useful approach when you already have lots of agents deployed and what to expand your observability to other platforms. In addition to that it enables log analytics

Download the executable file from the link below and place it in the *lab2\_appd\_hybrid\_agent* folder.

<https://github.com/lof000/otel-cco-labs/releases/download/v1.0/banking-2.1.0.jar>

Edit the run.sh file and add the information to connect to your AppDynamics Controller. Also, remember to replace YOUR\_ID with your unique id.



Now let’s start the application instrumented with AppDynamics agent in its default config.

Start app.

|  |
| --- |
| ./run.sh |

A screenshot of a computer screen

Description automatically generated

Put some load.

|  |
| --- |
| ./load.sh |

Check AppDynamics.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

So far it is business as usual. Now we will change the agent configuration to enable the hybrid mode.

Stop the app.

|  |
| --- |
| CTRL + C |

## Enable OTEL features in the AppDynamics java agent.

The AppDynamics agent needs to send the data to a OTEL Collector, and the collector sends them to Cisco Observability Platform. To accomplish that we need to prepare the collector configuration.

Edit the *otel-collector-config.yaml* leaving it like the sample below. The information to connect to Cisco Observability Platform can be found in the file you download here.

A screenshot of a computer

Description automatically generated

receivers:

otlp:

protocols:

grpc:

http:

exporters:

logging:

verbosity: detailed

jaeger:

endpoint: jaeger:14250

tls:

insecure: true

otlphttp:

auth:

authenticator: oauth2client

traces\_endpoint: https://<tenant\_host>/data/v1/trace

logs\_endpoint: https://<tenant\_host>/data/v1/logs

processors:

batch: #### Optional for trace batching for AppDynamics Cloud

send\_batch\_max\_size: 1000

send\_batch\_size: 1000

timeout: 10s

extensions: #### Mandatory for AppDynamics Cloud

oauth2client:

client\_id: xxxx

client\_secret: xxxx

token\_url: https://tenant\_host>auth/xxxxx/default/oauth2/token

service:

extensions: #### Mandatory for AppD Cloud

- oauth2client

pipelines:

traces:

receivers: [otlp]

processors: [batch]

exporters: [logging,jaeger,otlphttp]

metrics:

receivers: [otlp]

exporters: [logging]

logs:

receivers: [otlp]

exporters: [logging,otlphttp]

Observe that the otlphttp exporter is a little different from the previous lab. And now we also have a new extension called outh2client. This is because now the collector will send the information do Cisco Observability Platform.

Start the collector.

|  |
| --- |
| docker-compose up -d |

Now we will change the agent configuration to enable OTEL communication. Remember to inform your unique id.

Edit run.sh and the OTEL variables.

#ENV VARIABLES FOR OTEL

export OTEL\_EXPORTER\_OTLP\_ENDPOINT=http://localhost:4317

export OTEL\_RESOURCE\_ATTRIBUTES="service.name=banking,service.namespace=bankLab2$ID"

#APM AGENT VARIABLES

export JAVA\_TOOL\_OPTIONS="-javaagent:appd\_agent/javaagent.jar"

java -Dappdynamics.opentelemetry.enabled=true -jar banking-2.1.0.jar

Run the app.

|  |
| --- |
| ./run.sh |

Put some load.

|  |
| --- |
| ./load.sh |

Check the application in AppDynamics.

A screenshot of a computer

Description automatically generated

Check the same application in CCO.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated