



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Workbench User's Guide

AD Troubleshooting

5/2/2025

Contents

- 1 AD Troubleshooting
 - 1.1 General
 - 1.2 Logs for Troubleshooting
 - 1.3 AD API for Troubleshooting

AD Troubleshooting

General

Important

- Workbench uses the Hostname for component configuration
- Please ensure hostname resolution between Workbench components, including AD and Engage Hosts is accurate and robust
- If the Workbench Hosts have multiple NIC's, please ensure the Hostname resolves to the desired IP Address **prior** to Workbench installation
- Double-check network ports that are used by AD are from a firewall perspective, **open and not already in use** by other applications
- AD Nodes/Hosts require a minimum of 8 CPU cores
- Install the AD components on dedicated hosts - not on the same Nodes/Hosts as the Workbench core components.

Logs for Troubleshooting

AD automatically creates the file *ad_monitoring.log* in the {LOG_PATH} folder configured.

The structure for this log file is using this format:

```
'%(asctime)s | %(levelname)s | %(processName)s | %(message)s')
```

- Time format: 2021-09-20 03:15:46,291
- The default Log_Level is INFO. DEBUG mode can be used to see details about the process executed by AD. Doing that will reduce the performance of some components like streaming consumers and collectors.
- processName tell the AD component that is generating the event

Below a few tips of Log information for troubleshooting:

- AD start: check if AD is running as a primary or additional node

```
2021-09-08 16:16:21,446 | INFO | application_manager | WB-AD starting
2021-09-08 16:16:21,447 | INFO | application_manager | AD compilation time:
210908-192852
2021-09-08 16:16:21,447 | INFO | application_manager | configuration path: configs
2021-09-08 16:16:21,447 | INFO | application_manager | main path: /Installation/path
2021-09-08 16:16:22,172 | INFO | application_manager | App Manager started
2021-09-08 16:16:22,173 | INFO | application_manager | local data storage initialized
2021-09-08 16:16:22,173 | INFO | application_manager | AD --.---.---.--- as primary node
2021-09-08 16:16:22,173 | INFO | application_manager | app_manager class initialized
```

- AD components are started in this order: ad_api, streaming consumer, collector, model_manager, anomaly_detector and alarm_monitoring.

```
2021-09-20 03:03:47,939 | INFO | application_manager | New ad_api process started
with pid 49852
2021-09-20 03:03:47,941 | INFO | ad_api | starting AD API: -----:8182
2021-09-20 03:03:47,943 | INFO | application_manager | New
streaming_consumer_logstash0 process started with pid 49853
2021-09-20 03:03:47,952 | INFO | application_manager | New collector process started
with pid 49854
2021-09-20 03:03:47,953 | INFO | streaming_consumer_logstash0 | Streaming Consumer
initialized
2021-09-20 03:03:47,957 | INFO | collector | AD Collector initialized
2021-09-20 03:03:48,021 | INFO | model_manager | Model Manager initialized
2021-09-20 03:03:48,011 | INFO | application_manager | New model_manager process
started with pid 49855
2021-09-20 03:03:48,036 | INFO | application_manager | New anomaly_detector process
started with pid 49856
2021-09-20 03:03:48,059 | INFO | application_manager | New alarm_monitoring process
started with pid 49857
2021-09-20 03:03:48,063 | INFO | anomaly_detector | Anomaly Analyzer initialized
2021-09-20 03:03:48,072 | INFO | application_manager | modules initialized
2021-09-20 03:03:48,072 | INFO | anomaly_detector | Anomaly Detector initialized
```

2021-09-20 03:03:48,087 | INFO | alarm_monitoring | Alarm Monitoring initialized

- Commons errors detected:
 - Trying to connect with Logstash TCP server: must be confirmed with an Alarm generated by AD.

```
2021-09-20 02:32:44,139 | ERROR | streaming_consumer_logstash0 | error collecting
messages. Traceback (most recent call last): File "core/streaming_consumer.py", line
133, in main SC.streaming_process() File "core/streaming_consumer.py", line 67, in
streaming_process message = self.broker.get_message() File "core/
streaming_consumer.py", line 24, in get_message message = self.socketFile.readline()
File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/socket.py",
line 586, in readinto return self._sock.recv_into(b)socket.timeout: timed out
```

```
error collecting messages. Traceback (most recent call last): File
"streaming_consumer.py", line 131, in main File "streaming_consumer.py", line 60, in
set_broker File "streaming_consumer.py", line 19, in __init__ ConnectionRefusedError:
[Errno 111] Connection refused
```

Important events:

- New source detected
- AD model trained or updated
- New alarm sent
- New anomaly (insight) created
- change detected in AD config file
- restating AD components
- AD component terminated
- Additional AD Nodes:
 - new source added from primary
 - AD model updated from primary
 - Primary node is not responding
 - sending request to update primary node in WB

AD API for Troubleshooting

Additional AD API endpoints were added to monitor AD status. Per default `ad_api` is running on port 8182

- `/ad_api/status`: return the current status for AD an components.
- `/ad_api/get_sources_summary`: return the list of sources (metrics) collected by AD.
- `/ad_api/get_alarms`: return the alarms generated by AD in status open.

- `/ad_api/get_last_error`: return the last error detected in logs
- `/ad_api/get_last_insight`: return basic information about the source of last insight detected.