AppSentinels
Business Logic Security

AppSentinels API Security Platform

Java-Nodejs Watcher Integration

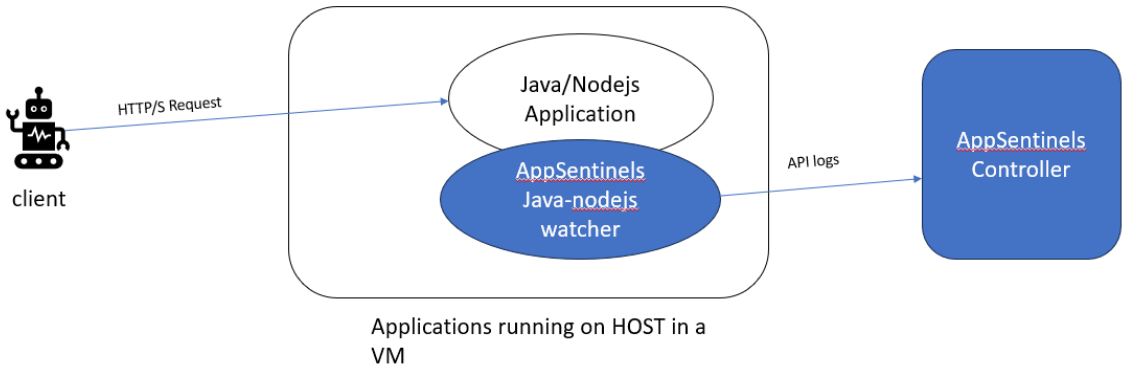
Revision	Date Modified	Author	Comments
1.0	01-Dec-25	Sagar	Initial spec for java-nodejs agent integration
2.0	25-Feb-2026	Sagar	Added windows installation process

Contents

AppSentinels Java-Nodejs watcher

The AppSentinels Java/Nodejs Watcher provides agent based auto-instrumentation using trusted Hypertrace to enable API visibility and monitoring. The approach uses the build-in java/nodejs agent functionality to instrument the java/nodejs application.

Architecture Diagram



AppSentinels Watcher for Java Application:

AppSentinels Java watcher instrument the application automatically to capture the below information:

- Request and response headers
- Request and response bodies

Linux Environment:

How to Install:

To install the AppSentinels Java Watcher, you need to install the RPM package provided. Follow these steps:

- **Download the RPM package**: Obtain the RPM package for the AppSentinels Java Watcher from the Below link:
  - https://downloads.appsentinels.ai/appsentinels-deployment/Java-Nodejs-Watcher/appsentinels-watcher-1.3.6-1.noarch.rpm
- **Install the RPM package:** Use the following command to install the package on your system:
  - sudo rpm -ivh appsentinels-watcher-<version>.<arch>.rpm

How to Configure:

After installing the watcher, you need to configure the reporting endpoint to ensure that the watcher can send data to the AppSentinels controller, In addition to that you can configure the allowedContentTypes, maxbodysize.

```yaml
# Basic Configuration
service_name: appsentinels-watcher
enabled: true  # Enable/disable the agent


# Reporting Configuration
reporting:
  # Please make it https instead of http in case secure connection to controller
  endpoint: http://<Controller IP or FQDN>:<Port Number>  # OTLP gRPC endpoint for traces
  protocol: grpc
  metricReporterType: None
  #HTTPS based config. Disable below 2 lines in case of http based connection to controller
  #secure: true
  #cert_file: <Path of CA cert file which will validate the server certificate in case of self signed cert>

# Data Capture Configuration
dataCapture:
  bodyMaxSizeBytes: 131072  # 128KB - Maximum size of request/response bodies to capture
  httpHeaders:
    request: true  # Enable/disable request headers capture
    response: true  # Enable/disable response headers capture
  httpBody:
    request: true  # Enable/disable request body capture
    response: true  # Enable/disable response body capture
  allowedContentTypes:
    - application/json
    - application/xml
    - text/json
    - text/xml
    - application/x-www-form-urlencoded
    - application/graphql
```

- This config file will be present in below location and is configured by below environmental variable:
    - **HT_CONFIG_FILE=/opt/appsentinels-watcher/config.yaml**

- **Edit the Configuration File:** Open the **config.yaml** file located in the **HT_CONFIG_FILE** environmental variable and configure **reporting.endpoint**

How to activate:

Once the configuration is complete, follow these steps to reload your shell and start the Java application:

- **Reload Your Shell:** This step ensures that any environment variables required to instrument the java application using watcher are updated.
- **Check the Environment Variable**: Ensure **JAVA_TOOL_OPTIONS** environment has been set or appended with the appsentinels watcher configuration file path:
    - JAVA_TOOL_OPTIONS=-javaagent:/opt/appsentinels-watcher/appsentinels-agent.jar=ht.config.file=/opt/appsentinels-watcher/config.yaml
- **Restart Your Java Application**: Restart your Java application in new shell.The AppSentinels Java Watcher will automatically instrument your application for monitoring.

By following these steps, you will have successfully installed and configured the AppSentinels Java Watcher for enhanced API visibility and monitoring.

How to uninstall:

- List the installed appSentinels package name
  - rpm -qa | grep appsentinels
- Remove the appSentinels package
  - sudo rpm -e appsentinels-watcher-1.3.6-1.noarch
- Come out of your current shell and launch a new shell, then restart your application.

Windows Environment:

How to Install:

To install the AppSentinels Java Watcher, you need to update the command line arguments provided during startup of a specific service.

- **Download the ZIP file**: Get the ZIP file for the AppSentinels Java Watcher from the Below link:
  - https://sample-config.appsentinels.ai/appsentinels-deployment/Java-Nodejs-Watcher/appsentinels_java_watcher_win.zip
  - unzip it and place to some location. The zip file contains two files:
    - hypertrace-agent.jar
    - agent-config.yaml
- **Update the command line arguments:** Add below two arguments with the existing arguments
  - **-javaagent:<path of the agent jar file >**
    - Ex: -javaagent:C:\hypertrace-agent\hypertrace-agent.jar
  - **-Dht.config.file=<path of the agent config file>**
    - -Dht.config.file=C:\hypertrace-agent\agent-config.yaml

How to Configure:

After installing the watcher, you need to configure the reporting endpoint to ensure that the watcher can send data to the AppSentinels controller, In addition to that you can configure the allowedContentTypes, maxbodysize.


```yaml
# Basic Configuration
service_name: appsentinels-watcher
enabled: true  # Enable/disable the agent


# Reporting Configuration
reporting:
  # Please make it https instead of http in case secure connection to controller
  endpoint: http://<Controller IP or FQDN>:<Port Number>  # OTLP gRPC endpoint for traces
  protocol: grpc
  metricReporterType: None
  #HTTPS based config. Disable below 2 lines in case of http based connection to controller
  #secure: true
  #cert_file: <Path of CA cert file which will validate the server certificate in case of self signed cert>

# Data Capture Configuration
dataCapture:
  bodyMaxSizeBytes: 131072  # 128KB - Maximum size of request/response bodies to capture
  httpHeaders:
    request: true  # Enable/disable request headers capture
    response: true  # Enable/disable response headers capture
  httpBody:
    request: true  # Enable/disable request body capture
    response: true  # Enable/disable response body capture
  allowedContentTypes:
    - application/json
    - application/xml
    - text/json
    - text/xml
    - application/x-www-form-urlencoded
    - application/graphql
```

- This config file will be present in below location which you have already configured in command line:
    - -Dht.config.file= **<path of the agent config file>**

How to activate:

Once the configuration is complete, follow these steps to reload your shell and start the Java application:

- **Restart Your Java Service**: Restart your Java service. The AppSentinels Java Watcher will automatically instrument your application for monitoring.

By following these steps, you will have successfully installed and configured the AppSentinels Java Watcher for enhanced API visibility and monitoring.

How to uninstall:

- Remove these two command line arguments
    - **-javaagent:<path of the agent jar file >**
    - **-Dht.config.file=<path of the agent config file>**
- Restart the service

AppSentinels Watcher for Nodejs Application:

AppSentinels Nodejs watcher instrument the application automatically to capture the below information:

- Request and response headers

- Request and response bodies

How to install:

Installation of AppSentinels Nodejs watcher is same as above mentioned in java watcher. Please follow the same for installation

How to configure:

To configure the AppSentinels Nodejs watcher Please follow the above steps mentioned for Java watcher

How to activate:

Once the configuration is complete, follow these steps to reload your shell and start the Nodejs application:

- **Reload Your Shell:** This step ensures that any environment variables required to instrument the Nodejs application using watcher are updated.
- **Check the Environment Variable**: Ensure **NODE_OPTIONS** environment has been set or appended with the appsentinels watcher configuration file path:
  - NODE_OPTIONS=-r @hypertrace/nodejsagent
- **Restart Your Nodejs Application**: Restart your Nodejs application in new shell. The AppSentinels Nodejs Watcher will automatically instrument your application for monitoring.

By following these steps, you will have successfully installed and configured the AppSentinels Nodejs Watcher for enhanced API visibility and monitoring.

How to uninstall:

- List the installed appSentinels package name
  - rpm -qa | grep appsentinels
- Remove the appSentinels package
  - sudo rpm -e appsentinels-watcher-1.3.6-1.noarch
- Come out of your current shell and launch a new shell, then restart your application.

AppSentinels Watcher for container-based application:

- So, for any container-based environment either it is manged or unmanaged, to deploy the auto-instrumentation agent you have to somehow attach the agent jar file and config file with application deployment file.
- The below procedure is given for AWS ECS FARGATE environment:
  - Add below environment variable as part of "**environment**" section in task definition:
    - { "name": "JAVA_TOOL_OPTIONS", "value": "-javaagent:/otel-auto-instrumentation/hypertrace-agent.jar" },
      { "name": "HT_CONFIG_FILE", "value": "/otel-auto-instrumentation/config.yaml" },
      { "name": "HT_SERVICE_NAME", "value": "<service name>" },
      { "name": "HT_REPORTING_ENDPOINT", "value": "http://<controller IP>:9009" },
      { "name": "HT_REPORTING_TRACE_REPORTER_TYPE", "value": "OTLP" },
      { "name": "HT_REPORTING_SECURE", "value": "false" },


```
{ "name": "HT_DATA_CAPTURE_HTTP_HEADERS_REQUEST",  "value": "true" },
{ "name": "HT_DATA_CAPTURE_HTTP_HEADERS_RESPONSE", "value": "true" },
{ "name": "HT_DATA_CAPTURE_HTTP_BODY_REQUEST",    "value": "true" },
{ "name": "HT_DATA_CAPTURE_HTTP_BODY_RESPONSE",   "value": "true" },
{ "name": "HT_DATA_CAPTURE_RPC_METADATA_REQUEST",  "value": "true" },
{ "name": "HT_DATA_CAPTURE_RPC_METADATA_RESPONSE", "value": "true" },
{ "name": "HT_DATA_CAPTURE_RPC_BODY_REQUEST",     "value": "true" },
{ "name": "HT_DATA_CAPTURE_RPC_BODY_RESPONSE",    "value": "true" },
{ "name": "HT_DATA_CAPTURE_BODY_MAX_SIZE_BYTES",  "value": "131072" },
{ "name": "HT_ENABLED",   "value": "true" },
{ "name": "HT_ENVIRONMENT",   "value": "production" },
{ "name": "HT_RESOURCE_ATTRIBUTES",  "value":
"deployment.environment=production,cloud.provider=aws,cloud.platform=aws_ecs
" }
```

- Add mount path in "**mountPoints**" where hypertrace-agent.jar file and config.yaml file will be stored:
  - ```
    {
            "sourceVolume": "hypertrace-agent",
            "containerPath": "/otel-auto-instrumentation",
            "readOnly": true
    }
    ```
- Add the Dependency for the application on hypertrace init container in "**dependsOn**":
  - ```
    {
            "containerName": "hypertrace-init",
            "condition": "COMPLETE"
    }
    ```
- Add hypertrace container definition in "**containerDefinitions**" section:
  - ```
    {
      "name": "hypertrace-init",
      "image": "488922454646.dkr.ecr.ap-south-1.amazonaws.com/hypertrace-
    init:latest",
      "essential": false,
      "mountPoints": [
        {
          "sourceVolume": "hypertrace-agent",
          "containerPath": "/otel-auto-instrumentation",
          "readOnly": false
        }
      ],
      "logConfiguration": {
        .
        .
      }
    }
    ```

NB: For any other environment image is available in appsentinels/appsentinels-hypertrace-init:latest

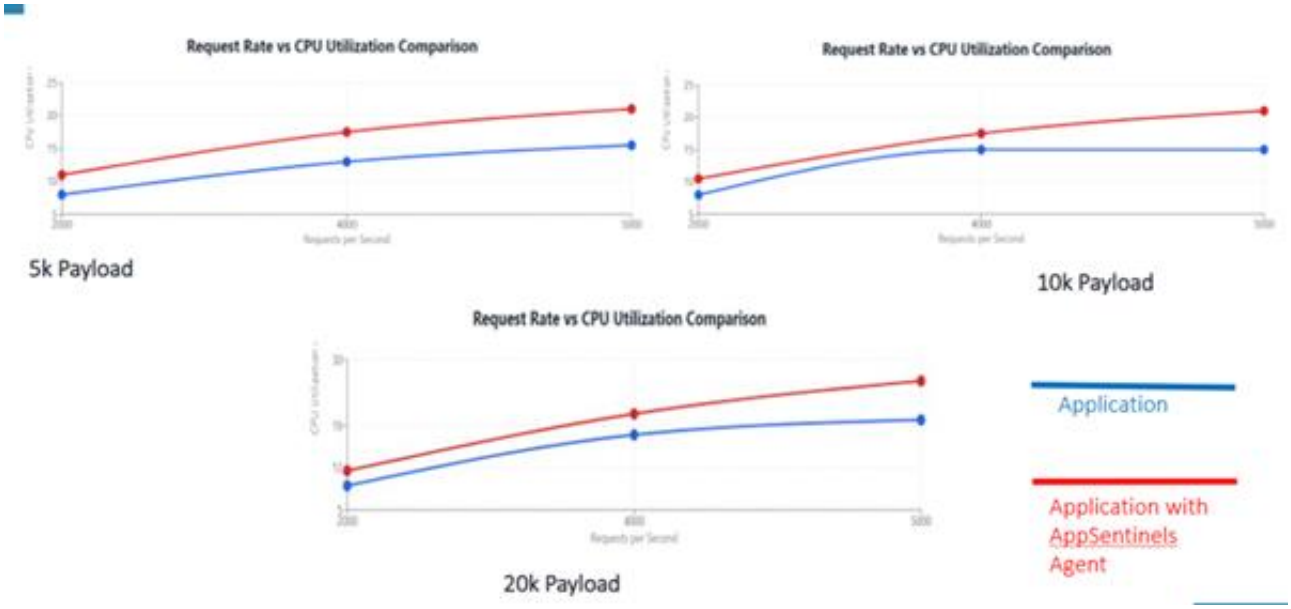

    ```

- Add Volumes in "**volumes**" where hypertrace-agent.jar and config.yaml file will be mounted
  - {
    "name": "hypertrace-agent",
    "host": {}
    }

# Appendix

## System CPU Utilization

- Below is the comparative view of system CPU utilization




**System CPU Utilization (Tested in AppSentinels)**

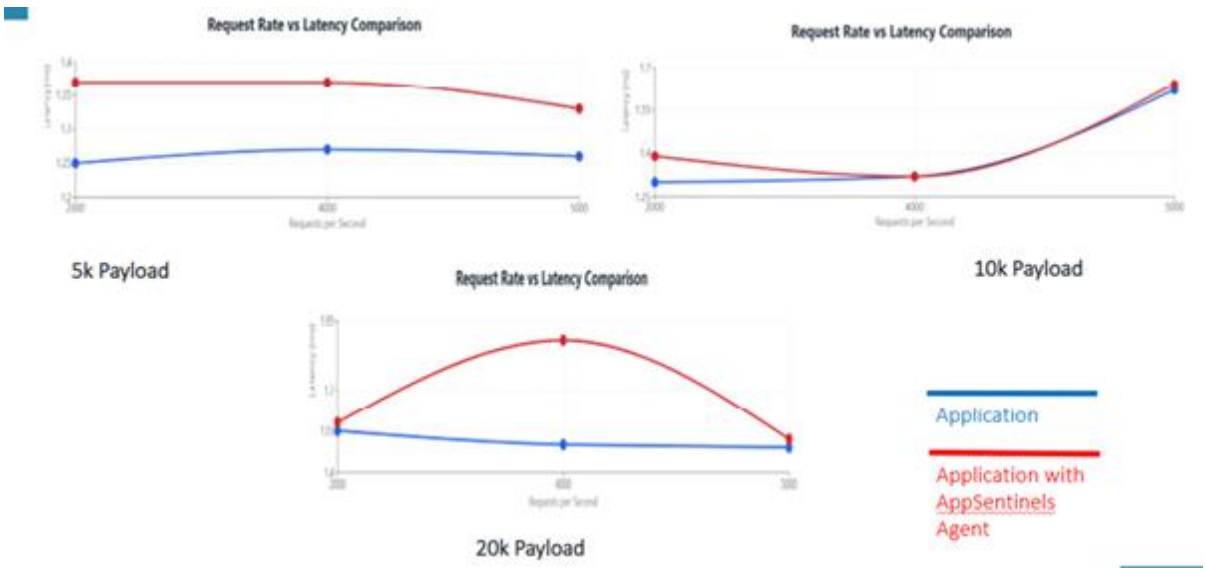
## System Memory Utilization

- Below is the comparative view of system Memory utilization


**System memory Utilization (Tested in AppSentinels)**

## API Latency

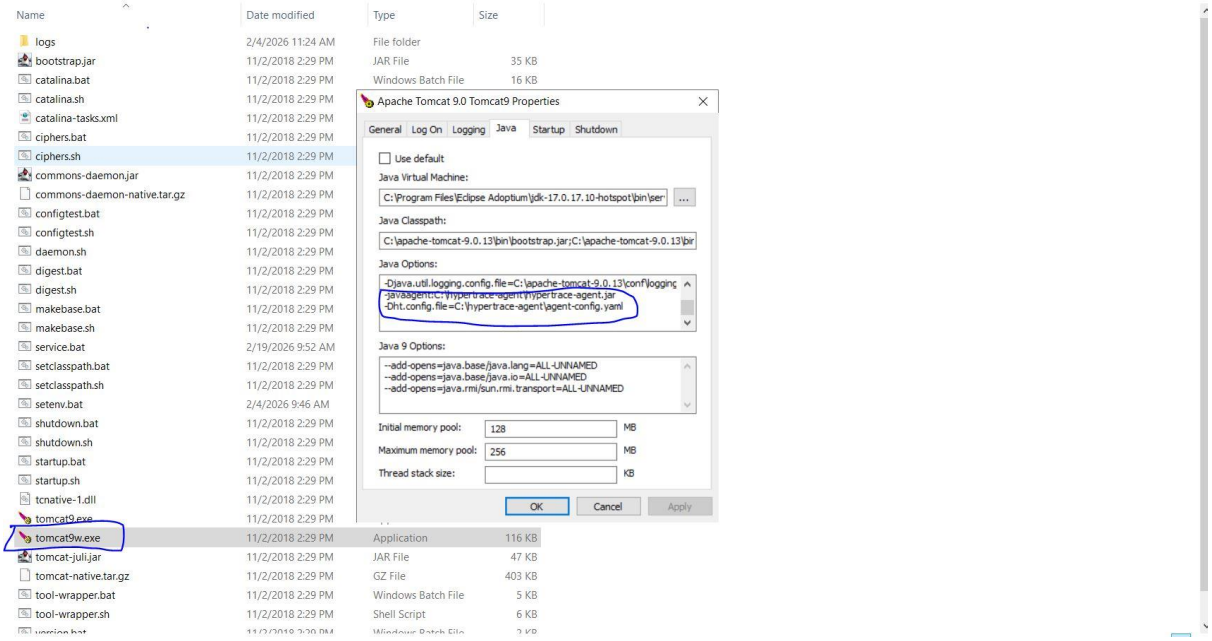
- Below is the comparative view of API latency




**API Latency (Tested in AppSentinels)**

## Reference:

- To integrate with tomcat service in windows Please do below changes:
  - Open the **Tomcat Service Configuration** GUI (tomcat9w.exe)
  - Navigate to the **Java** tab
  - In the **Java Options** text box, add the following two arguments:
    - -javaagent**:<path of the agent jar file>**
    - -Dht.config.file=**<path of the agent config file>**
  - Click **Apply** → **OK**.
  - Restart the Tomcat service.




- If you have different bat file to start the tomcat service, and that bat file internally calling **catalina.bat** file please add these below two lines in **catalina.bat** file
  - set "CATALINA_OPTS=%CATALINA_OPTS% -javaagent:<**path of the agent jar file**>"
  - set "CATALINA_OPTS=%CATALINA_OPTS% -Dht.config.file=<**path of the agent config file**>"

```
rem Add tomcat-juli.jar to classpath
rem tomcat-juli.jar can be over-ridden per instance
if not exist "%CATALINA_BASE%\bin\tomcat-juli.jar" goto juliClasspathHome
set "CLASSPATH=%CLASSPATH%;%CATALINA_BASE%\bin\tomcat-juli.jar"
goto juliClasspathDone
:juliClasspathHome
set "CLASSPATH=%CLASSPATH%;%CATALINA_HOME%\bin\tomcat-juli.jar"
:juliClasspathDone

if not "%JSSE_OPTS%" == "" goto gotJsseOpts
set JSSE_OPTS="-Djdk.tls.ephemeralDHKeySize=2048"
:gotJsseOpts
set "JAVA_OPTS=%JAVA_OPTS% %JSSE_OPTS%"

rem Register custom URL handlers
rem Do this here so custom URL handles (specifically 'war:...') can be used in the security policy
set "JAVA_OPTS=%JAVA_OPTS% -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"

rem Hypertrace Agent configuration
set "CATALINA_OPTS=%CATALINA_OPTS% -javaagent:C:\hypertrace-agent\hypertrace-agent.jar"
set "CATALINA_OPTS=%CATALINA_OPTS% -Dht.config.file=C:\hypertrace-agent\agent-config.yaml"

if not "%LOGGING_CONFIG%" == "" goto noJuliConfig
set LOGGING_CONFIG=-Dnop
if not exist "%CATALINA_BASE%\conf\logging.properties" goto noJuliConfig
set LOGGING_CONFIG=-Djava.util.logging.config.file="%CATALINA_BASE%\conf\logging.properties"
:noJuliConfig
```

