AZURE APPLICATION INSIGHT

Application Insights is basically used for vital application telemetry data right out of the box; including usage, exceptions, requests, performance, and logs.

Monitor web apps—whether written in Java, Ruby, Python, PHP, Node.JS, or other languages—using open source SDKs. Install the Status Monitor on your existing Azure App Services and virtual machines through the Azure portal to get performance monitoring without needing to update and redeploy your application.

Visual Studio Application Insights is an extensible analytics service that monitors your live web application. With it you can detect and diagnose performance issues, and understand what users actually do with your app. It’s designed for developers, to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud.

It provides benefits as:

* Exceptions and performance diagnostics.
* Interactive data analysis.
* Azure diagnostics.
* Proactive detection.

**Steps for generating log over Application Insight:**

**Step 1)** The first step is go to Microsoft azure then search for application insight and create new one.

**Step 2)** Once its created then note down its instrumentation id & Name of your application. Now write down these in your application.properties .

**Step 3)** Now create an application in visual studio.

**Step 4)** In Pom.xml add following dependencies.

a) For using azure insight in a pplication:

<dependency>

<groupId>com.microsoft.azure</groupId>

<artifactId>applicationinsights-spring-boot-starter</a rtifactId>

<version>1.1.1</version>

</dependency>

<dependency>

            <groupId>com.microsoft.azure</groupId>

            <artifactId>applicationinsights-logging-logback</artifactId>

            <version>2.6.0</version>

        </dependency>

b) For Output in JSON fromat:

<dependency>

            <groupId>net.logstash.logback</groupId>

            <artifactId>logstash-logback-encoder</artifactId>

            <version>6.3</version>

        </dependency>

**Step 5)** Add another xml file specifically consisting of Log properties i.e. it will control of how the representation and behavior of logs.

<?xml version="1.0" encoding="UTF-8"?>

<configuration>

    <include resource="org/springframework/boot/logging/logback/defaults.xml"/>

    <appender name="jsonConsoleAppender" class="ch.qos.logback.core.ConsoleAppender">

        <encoder class="net.logs tash.logback.encoder.LogstashEncoder">

            <fieldNames>

                <levelValue>[ignore]</levelValue>

                <version>[ignore]</version>

            </fieldNames>

        </encoder>

    </appender>

    <appender name="aiAppender"

        class="com.microsoft.applicationinsights.logback.ApplicationInsightsAppender">

    </appender>

    <root level="debug">

        <appender-ref ref="jsonConsoleAppender"/>

        <appender-ref ref="aiAppender" />

    </root>

    <logger name="cssfm.copo" level="INFO"/>

</configuration>

**Step 6)** Now we have to add few changes in code like,

    private static final Logger log = LoggerFactory.getLogger(CommentController.class);

And use

* logger.info("This is an {} message.","info");
* logger.warn("This is a warn message.");
* logger.error("This is an error message.");
* logger.debug("This is a debug message.");