**Configure Appender to Splunk in Mule Application log4j configuration**

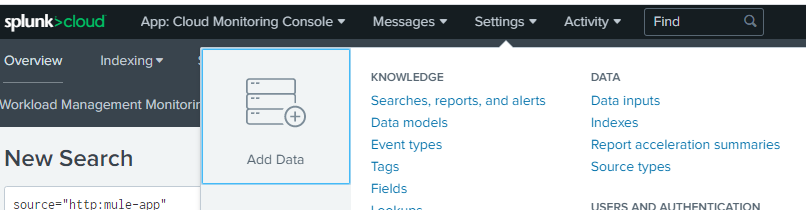
**Introduction:** Splunk is a tool used for logging, analysing, reporting, visualising, monitoring or searching the machine data in real time.

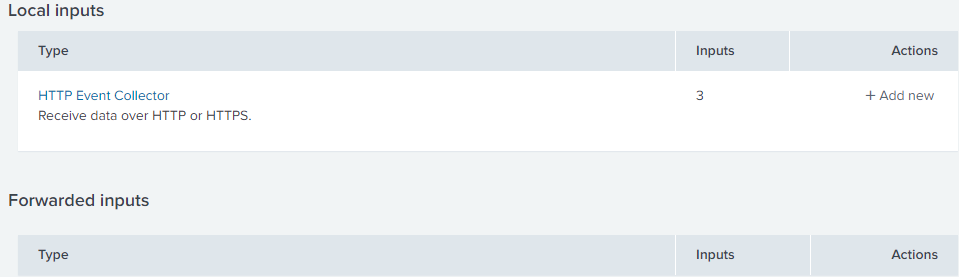
HEC offers an experience on Splunk Cloud deployments that Splunk manages that is similar to the experience on self-service Splunk Cloud. The following exceptions apply, you cannot make changes to configuration files, because Splunk Cloud does not provide that access.

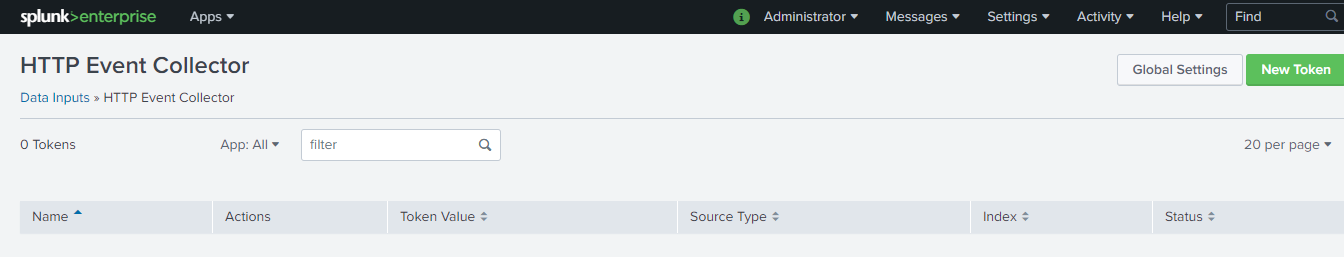
**Creating HTTP Event Collector Token:**

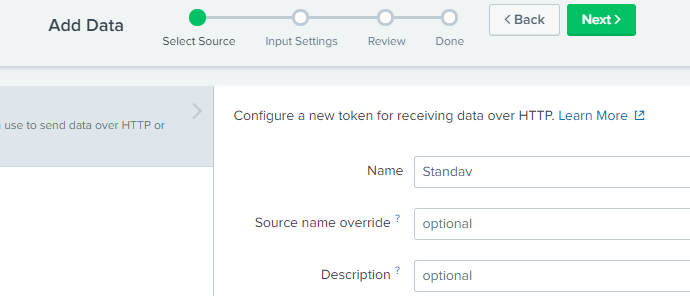
**The HTTP Event Collector (HEC) lets you send data and application events to a Splunk deployment over the HTTP and Secure HTTP (HTTPS) protocols. HEC uses a token-based authentication model.**

**For creating token, navigate to Settings → Data inputs → HTTP Event Collector**

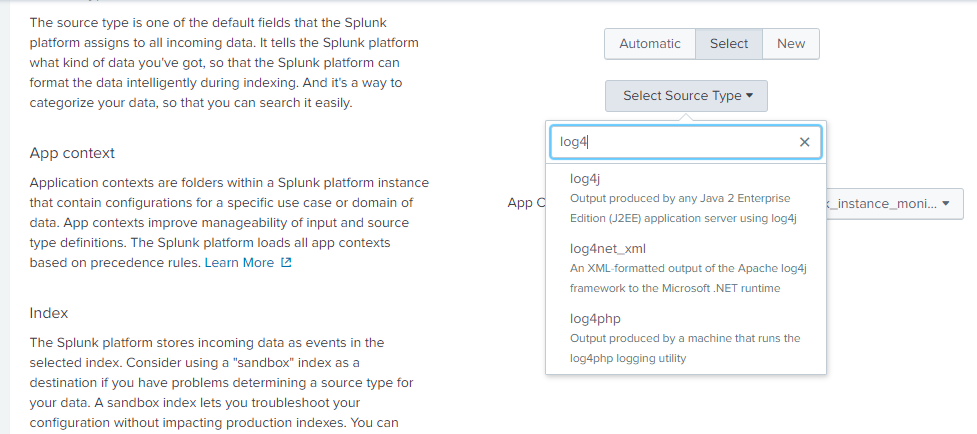




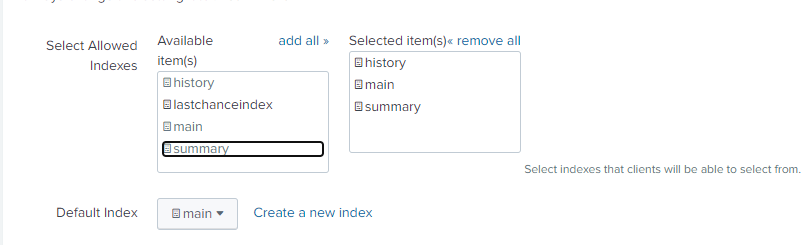


Once you click on New Token button, it will navigate to form, and start filling details like Name and click on Next at top of web page for filling further details.

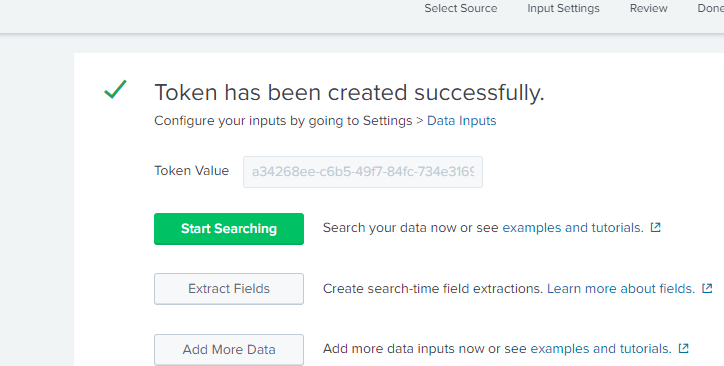
On the next screen you can select **SourceType**as **log4j**as we will be using log4j as source for sending data to Splunk from MuleSoft application.



Now configure the index and make sure main index as Default. And then Click Next 🡪 Submit



Once you clicked on **Submit**button, it will create a token and that we will be using for authentication purpose.



Here in Anypoint studio configure Log4j Appender to send log data to Splunk instance.

**Procedure:** One can configure log4j2.xml file inside application configuration on src/main/resources and use specific Splunk appender. below are the configuration details for Splunk appender approach.

**Splunk appender**

<SplunkHttp name=*"Splunk"*

url=*"https://ped-p-fn3wj.splunkcloud.com:8088/"*

token=*"460fedc3-702b-487d-afaa-7997ccdce793"*

batch\_size\_count=*"1"*

disableCertificateValidation=*"true"*>

<PatternLayout pattern=*"[%d{MM-dd HH:mm:ss}] %-5p %c{1} [%t]: %m%n"* />

</SplunkHttp>

The standard form for the URI in managed Splunk Cloud is as follows:

**<protocol>://http-inputs-<host>:<port>/<endpoint>**

**While using SplunkHttp, you need to add the following dependency in pom.xml of your application.**

**<dependency>**

**<groupId>com.splunk.logging</groupId>**

**<artifactId>splunk-library-javalogging</artifactId>**

**<version>1.7.3</version>**

**</dependency>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**<repository>**

**<id>splunk-artifactory</id>**

**<name>Splunk Releases</name>**

**<url>http://splunk.jfrog.io/splunk/ext-releases-local</url>**

**</repository>**

**You may need to add below two additional dependencies.**

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>2.10.0</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-api</artifactId>

<version>2.10.0</version>

</dependency>

**Note:** We need to define loggers in log4j2.xml.

make sure **AppenderRef**is match with name of **SplunkHttp** Appender.

<AsyncRoot level=”INFO”>   <AppenderRef ref=”Splunk” /></AsyncRoot>

**Enable Custom Logging for CloudHub Application:**

When you disable the default CloudHub application logs, then only the system logs are available. For application worker logs, please check your own application’s logging system.



You can add more loggers like Log4J2CloudhubLogAppender into your log4j2.xml to enable logging on the cloudhub log console of your application.

<Log4J2CloudhubLogAppender name="CLOUDHUB"

addressProvider="com.mulesoft.ch.logging.DefaultAggregatorAddressProvider"

applicationContext="com.mulesoft.ch.logging.DefaultApplicationContext"

appendRetryIntervalMs="${sys:logging.appendRetryInterval}"

appendMaxAttempts="${sys:logging.appendMaxAttempts}"

batchSendIntervalMs="${sys:logging.batchSendInterval}"

batchMaxRecords="${sys:logging.batchMaxRecords}"

memBufferMaxSize="${sys:logging.memBufferMaxSize}"

journalMaxWriteBatchSize="${sys:logging.journalMaxBatchSize}"

journalMaxFileSize="${sys:logging.journalMaxFileSize}"

clientMaxPacketSize="${sys:logging.clientMaxPacketSize}"

clientConnectTimeoutMs="${sys:logging.clientConnectTimeout}"

clientSocketTimeoutMs="${sys:logging.clientSocketTimeout}"

serverAddressPollIntervalMs="${sys:logging.serverAddressPollInterval}"

serverHeartbeatSendIntervalMs="${sys:logging.serverHeartbeatSendIntervalMs}"

statisticsPrintIntervalMs="${sys:logging.statisticsPrintIntervalMs}">

<PatternLayout pattern="[%d{MM-dd HH:mm:ss}] %-5p %c{1} [%t]: %m%n"/>

</Log4J2CloudhubLogAppender>

make sure **AppenderRef**is match with name of **cloudhub** Appender.

<AsyncRoot level=”INFO”>   <AppenderRef ref=”cloudhub” /></AsyncRoot>

**Below is full log4j2.xml which can be used for your application for enabling custom logging**

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

<Configuration status=*"INFO"* name=*"cloudhub"*

packages=*"com.mulesoft.ch.logging.appender,com.splunk.logging,org.apache.logging.log4j"*>

<Appenders>

<Log4J2CloudhubLogAppender name=*"CLOUDHUB"*

addressProvider=*"com.mulesoft.ch.logging.DefaultAggregatorAddressProvider"*

applicationContext=*"com.mulesoft.ch.logging.DefaultApplicationContext"*

appendRetryIntervalMs=*"${sys:logging.appendRetryInterval}"*

appendMaxAttempts=*"${sys:logging.appendMaxAttempts}"*

batchSendIntervalMs=*"${sys:logging.batchSendInterval}"*

batchMaxRecords=*"${sys:logging.batchMaxRecords}"* memBufferMaxSize=*"${sys:logging.memBufferMaxSize}"*

journalMaxWriteBatchSize=*"${sys:logging.journalMaxBatchSize}"*

journalMaxFileSize=*"${sys:logging.journalMaxFileSize}"*

clientMaxPacketSize=*"${sys:logging.clientMaxPacketSize}"*

clientConnectTimeoutMs=*"${sys:logging.clientConnectTimeout}"*

clientSocketTimeoutMs=*"${sys:logging.clientSocketTimeout}"*

serverAddressPollIntervalMs=*"${sys:logging.serverAddressPollInterval}"*

serverHeartbeatSendIntervalMs=*"${sys:logging.serverHeartbeatSendIntervalMs}"*

statisticsPrintIntervalMs=*"${sys:logging.statisticsPrintIntervalMs}"*>

<PatternLayout pattern=*"[%d{MM-dd HH:mm:ss}] %-5p %c{1} [%t]: %m%n"* />

</Log4J2CloudhubLogAppender>

<SplunkHttp name=*"Splunk"*

url=*"https://prd-p-fn3wj.splunkcloud.com:8088/"*

token=*"538feb69-9f16-4bd6-a0e1-ec7d0f71242b"*

batch\_size\_count=*"1"*

disableCertificateValidation=*"true"*>

<PatternLayout pattern=*"[%d{MM-dd HH:mm:ss}] %-5p %c{1} [%t]: %m%n"* />

</SplunkHttp>

</Appenders>

<Loggers>

<AsyncLogger

name=*"org.mule.runtime.core.internal.processor.LoggerMessageProcessor"*

level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft.agent"* level=*"INFO"* />

<AsyncRoot level=*"INFO"*>

<AppenderRef ref=*"Splunk"* />

<AppenderRef ref=*"CLOUDHUB"* />

</AsyncRoot>

<AsyncLogger name=*"com.gigaspaces"* level=*"ERROR"* />

<AsyncLogger name=*"com.j\_spaces"* level=*"ERROR"* />

<AsyncLogger name=*"com.sun.jini"* level=*"ERROR"* />

<AsyncLogger name=*"net.jini"* level=*"ERROR"* />

<AsyncLogger name=*"org.apache"* level=*"WARN"* />

<AsyncLogger name=*"org.apache.cxf"* level=*"WARN"* />

<AsyncLogger name=*"org.springframework.beans.factory"*

level=*"WARN"* />

<AsyncLogger name=*"org.mule"* level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft"* level=*"INFO"* />

<AsyncLogger name=*"org.jetel"* level=*"WARN"* />

<AsyncLogger name=*"Tracking"* level=*"WARN"* />

<AsyncLogger name=*"org.mule"* level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.extensions.jms"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.service.http.impl.service.HttpMessageLogger"*

level=*"INFO"* />

<AsyncLogger name=*"org.mule.extension.salesforce"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.extension.ftp"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.extension.sftp"* level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft.extension.ftps"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.modules.sap"* level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft.extension.mq"* level=*"INFO"* />

<AsyncLogger name=*"com.mulesoft.mq"* level=*"INFO"* />

<AsyncLogger name=*"org.mule.extension.db"* level=*"INFO"* />

<AsyncLogger name=*"httpclient.wire"* level=*"DEBUG"* />

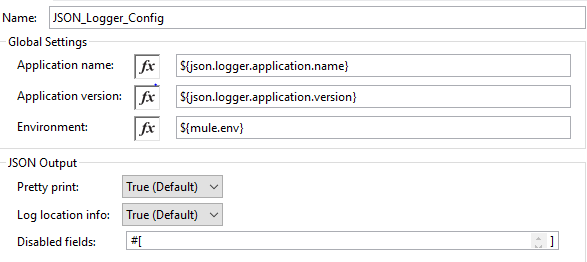
<AsyncLogger name=*"org.mule.transport.email"* level=*"DEBUG"* />

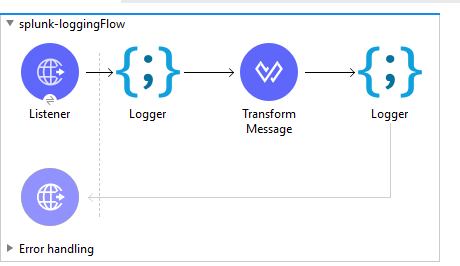
</Loggers>

</Configuration>

**JSON Logger:**

JSON Logger is a component for logging the information, warning, errors. It logs the data in the form of JSON. It is recommended to use JSON Logger for your application and it logs the data.





* Now deploy the API to cloud hub and you can see the logs in splunk cloud

Now, you can verify the logs in Splunk, and it will be in JSON format.



Conclusion:

Splunk is a very useful and powerful tool for logging, analysing, reporting, searching, and visualizing the events and data. MuleSoft provide capability to easily integrate Splunk using Anypoint Studio