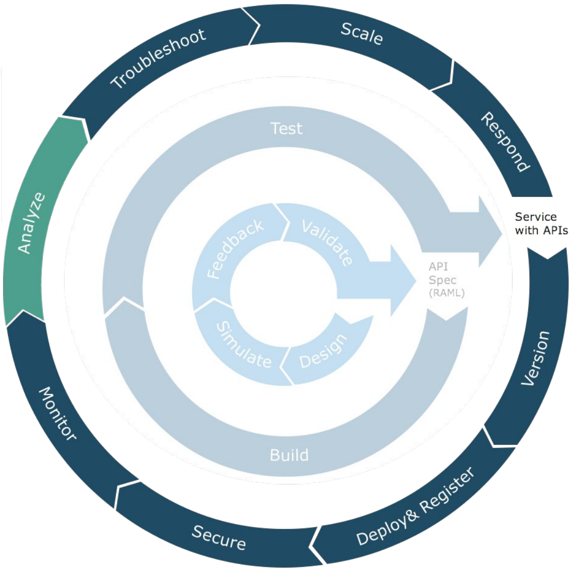
Module 9: Designing Effective Logging and Monitoring

Goal

API Analytics

Anypoint Monitoring

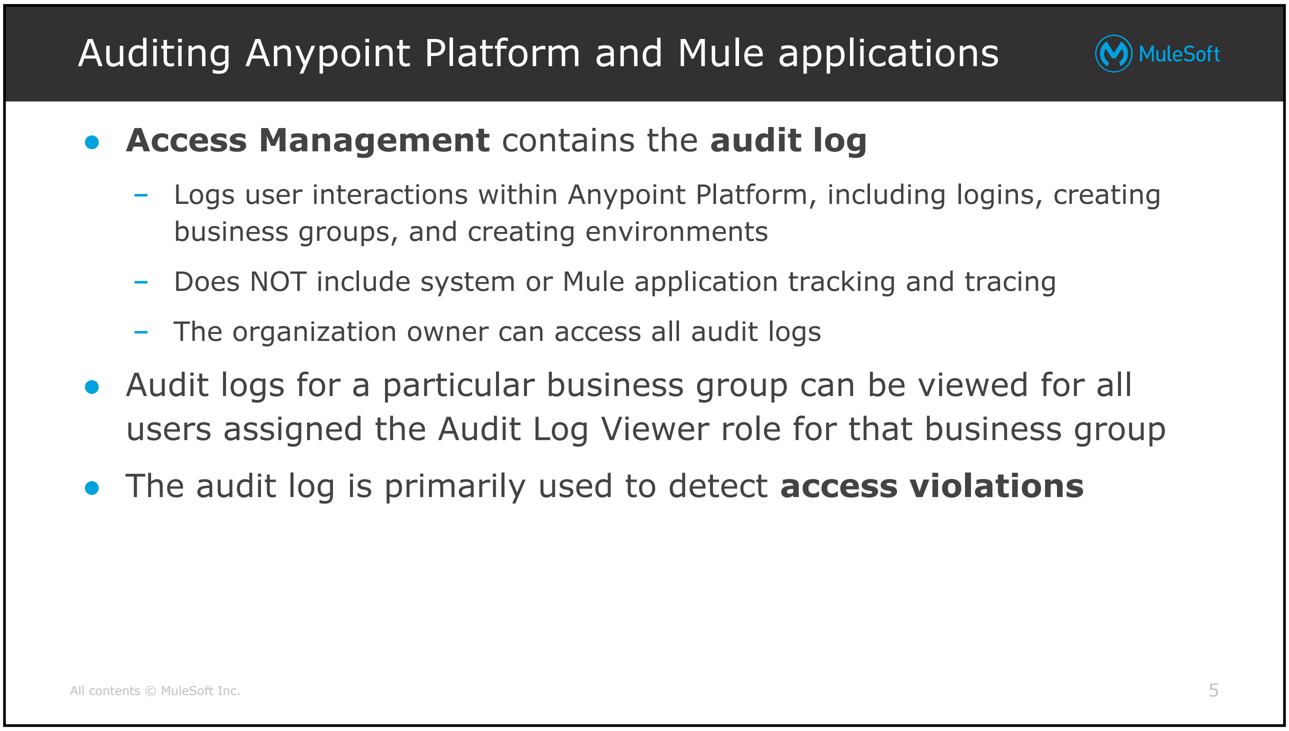
Anypoint Visualizer



At the end of this module, you should be able to

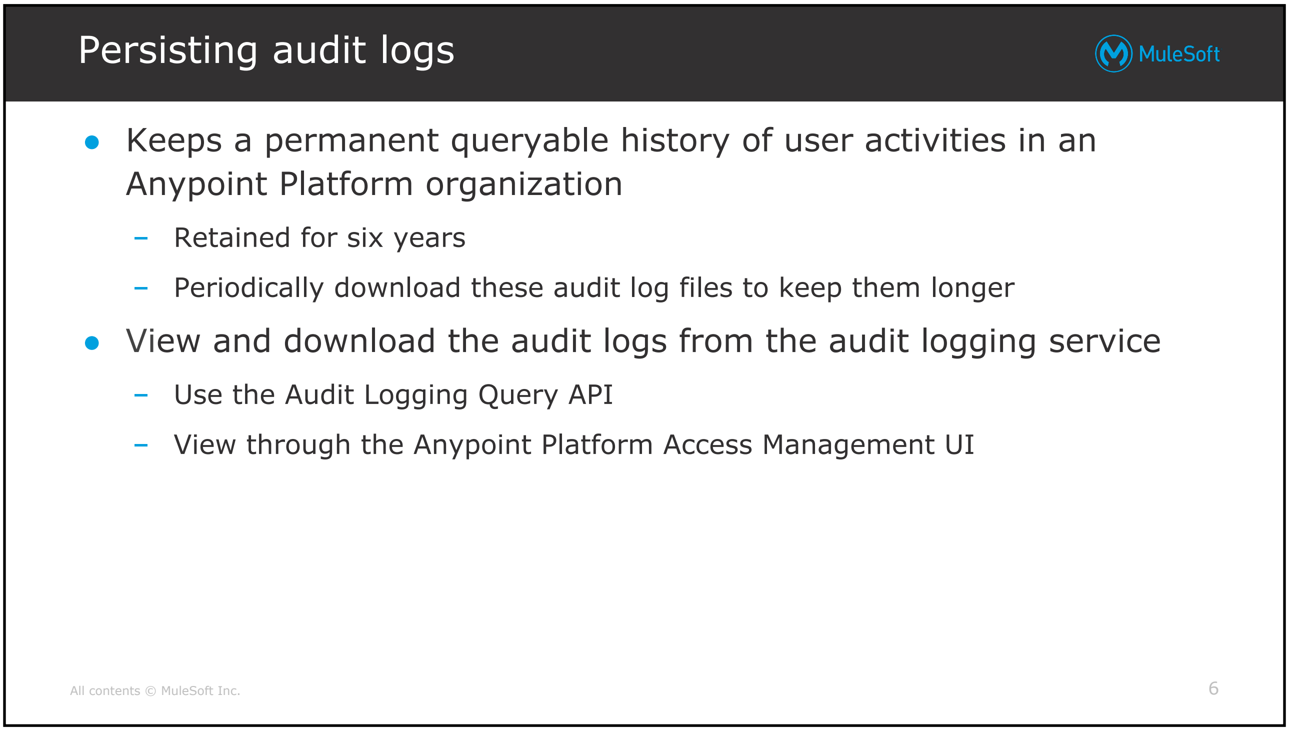
- Identifying auditing options for Mule application
- Identify the logging strategy for a Mule application
- Decide Mule application logging options
- Analyse integration options with third party log management system
- Decide monitoring, alerting, and notification options
- Determine API reporting options

Identifying auditing options for Mule application



Auditing Anypoint Platform and Mule applications

- **Access Management** contains the **audit log**
  - Logs user interactions within Anypoint Platform, including logins, creating business groups, and creating environments
  - Does NOT include system or Mule application tracking and tracing
  - The organization owner can access all audit logs
- Audit logs for a particular business group can be viewed for all users assigned the Audit Log Viewer role for that business group
- The audit log is primarily used to detect **access violations**



Persisting audit logs

- Keeps a permanent queryable history of user activities in an Anypoint Platform organization
  - Retained for six years
  - Periodically download these audit log files to keep them longer
- View and download the audit logs from the audit logging service
  - Use the Audit Logging Query API
  - View through the Anypoint Platform Access Management UI

Identifying logging options for Mule application

Logging in Mule

- Logs help to **debug and track** processing of Mule applications, such as details from Mule events
- In Mule flows, the Logging component creates log messages with various log levels, such as INFO, WARN, ERROR, and DEBUG
- Log messages can be configured to log to various **appenders**
  - The system console, a file, a database, the CloudHub logging service, or another server or service
- By default, Mule runtimes show the INFO log level, so ignore DEBUG or TRACE level log messages

Mule runtimes and Mule applications use standard slf4j and log4j2 logging

- Supports synchronous or asynchronous logging
- By default, logging in Mule is done **asynchronously**
- Can be configured with standard log4j2.xml at the system and Mule application level

System vs Application logs in the CloudHub runtime planes

System log

- Specific to the Mule runtime
- Configured by log4j2.xml inaccessible to customers
- Contains
  - Log messages about the Mule runtime lifecycle (startup and shutdown)
  - Status messages about Mule application

Application log

- Specific to the Mule application
- Configured by log4j2.xml typically packaged in Mule application
- Contains
  - All log messages generated inside the Mule application
    - Including System.out messages

System vs Application logs in customer-hosted runtime planes

MuleSoft

System log

- Specific to the Mule runtime
- Configured by log4j2.xml inaccessible to customers
- Contains
  - Log messages about the Mule runtime lifecycle (startup and shutdown)
  - Status messages about Mule application **and Mule domain deployments**
  - **System.out messages**

Application log

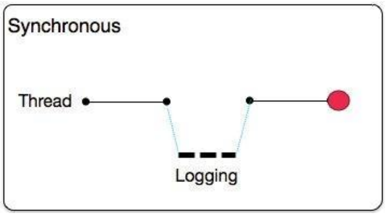
- Specific to the Mule application
- Configured by log4j2.xml typically packaged in Mule application
- Contains
  - All log messages generated inside the Mule application
    - ~~Including System.out messages~~

Synchronous vs asynchronous logging

MuleSoft

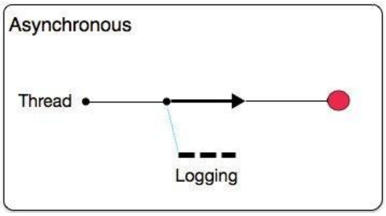
Synchronous

- The execution of the thread that is processing your message is interrupted to wait for the log message to be fully output before it can continue
- Performance degrades because of synchronous logging
- Used when the log is used as an audit trail or when logging ERROR/CRITICAL messages



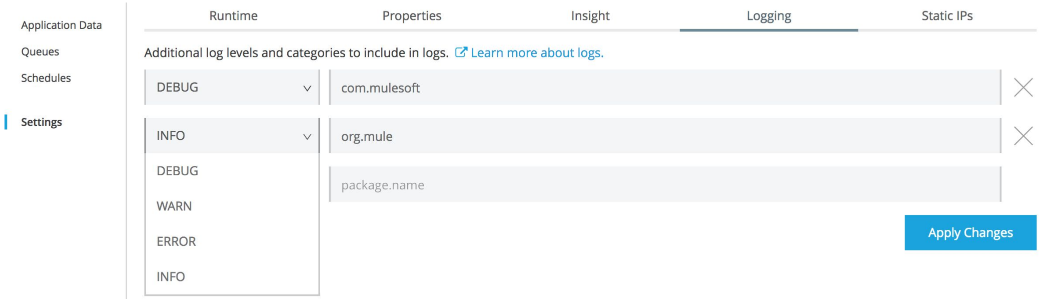
Asynchronous

- The logging operation occurs in a separate thread, so the actual processing of your message won't be delayed to wait for the logging to complete
- Substantial improvement in throughput and latency of message processing
- Log may lost in case of system crash



Configuring custom logging settings for a Mule runtime and its Mule applications

- The default log setting for Mule is asynchronous and at a level greater than or equal to INFO
  - So DEBUG and TRACE level messages are ignored
- Runtime Manager can override the log level for a deployed Mule application without redeployment



Additional log levels and categories to include in logs. Learn more about logs.

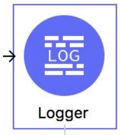
DEBUG com.mulesoft

INFO org.mule

package.name

The Mule Logger component

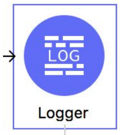
- If nothing is specified in the message attribute of the Logger component, the entire **Mule event** is logged, including all attributes and variables
  - A Logger does **not** log the contents of the **payload**, only its type
- A Logger sets the content of the message and the log level



```
00:23:46.903      11/15/2018      Worker-1      [MuleRuntime].cpuLight.01: [mule-objectstore-prod-am].readHtmlFiles.CPU_LITE
@699e1bae      DEBUG
ChildEventContext { id: 0-c33d5ea0-e8af-11e8-b463-0262da9ac180_296955003_2025473580; correlationId: 0-c33d5ea0-e8af-11e8-b463-
0262da9ac180; flowName: help; componentLocation: help/processors/0; response completed with result.
```

Using log messages to trace messages

- MuleSoft recommends logging messages with a unique id to help to trace the Mule event flow (the transaction) through the system
  - A unique ID could be
    - The correlation id that is automatically set in the Mule event
    - A system generated ID stored in the payload or HTTP request header
    - Another ID value stored in the payload or HTTP request header

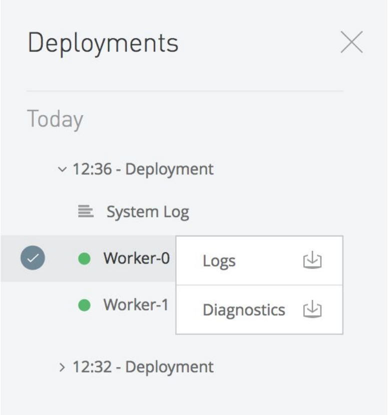


- The correlation id is automatically logged
  - For both CloudHub and customer-hosted Mule runtimes, the Logger component automatically logs the correlationId for DEBUG level messages

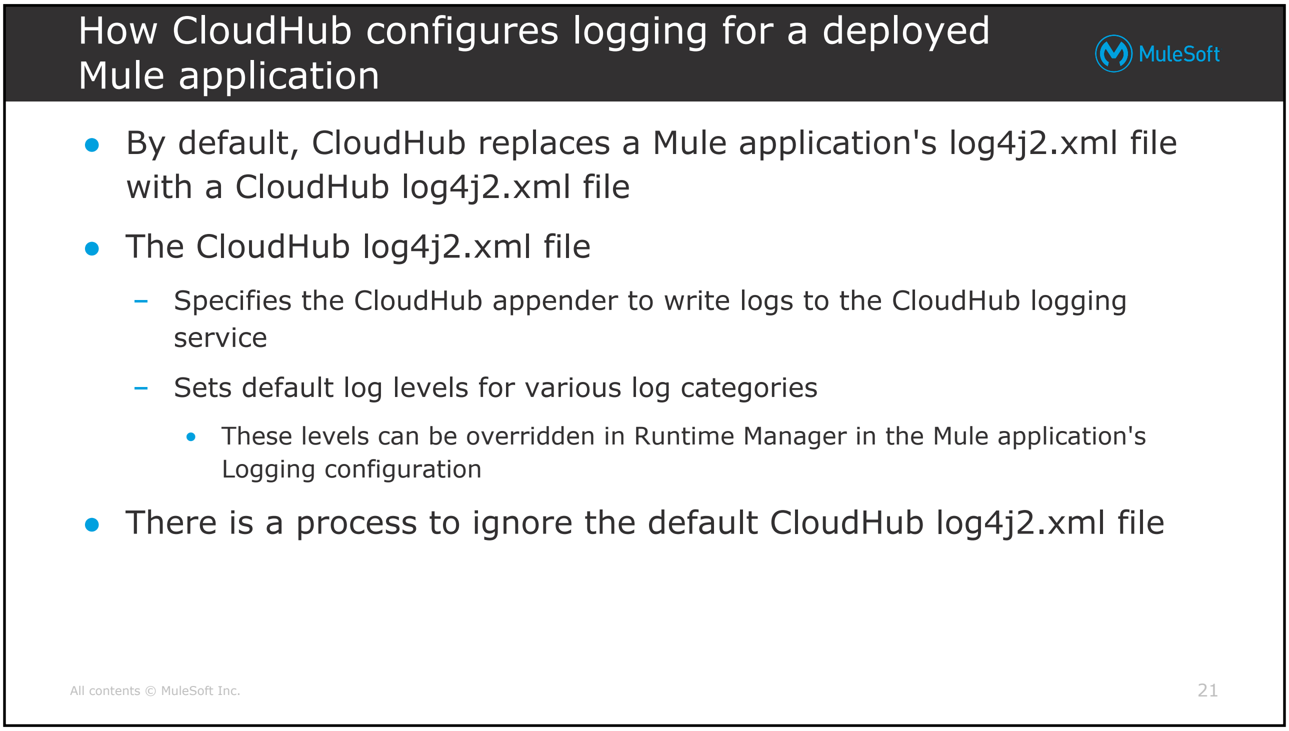
```
00:23:46.903    11/15/2018    Worker-1    [MuleRuntime].cpuLight.01: [mule-objectstore-prod-am].readHtmlFiles.CPU_LITE
@699e1bae    DEBUG
ChildEventContext { id: 0-c33d5ea0-e8af-11e8-b463-0262da9ac180_296955003_2025473580; correlationId: 0-c33d5ea0-e8af-11e8-b463-
0262da9ac180; flowName: help; componentLocation: help/processors/0; response completed with result.
```

Log retention in CloudHub

- CloudHub truncates log messages
  - Up to 100 MB per Mule application and per worker
  - At most 30 days,



Choosing between custom logging options



How CloudHub configures logging for a deployed Mule application

- By default, CloudHub replaces a Mule application's log4j2.xml file with a CloudHub log4j2.xml file
- The CloudHub log4j2.xml file
  - Specifies the CloudHub appender to write logs to the CloudHub logging service
  - Sets default log levels for various log categories
    - These levels can be overridden in Runtime Manager in the Mule application's Logging configuration
- There is a process to ignore the default CloudHub log4j2.xml file

Sending CloudHub logs to external logging systems using a **custom log appender**

- In CloudHub, you can disable the CloudHub provided Mule application log4j2 file
  - Allows integrating Mule application logs with custom or third-party log management systems
  - Only available on request via the support portal
  - Once enabled, the Mule application's log4j2.xml file is used
    - Can send/export application logs to other log4j2 appenders, such as a custom logging system
    - The system logs cannot be exported to other appenders
  - MuleSoft is not responsible for lost logging data due to misconfiguration of your own log4j appender
    - Mule application logs may not appear in Runtime Manager anymore and may no longer be available for download

Combining a CloudHub log appender with custom log appenders

- The CloudHub appender can also be configured inside a Mule application
  - Then the CloudHub appender will also be used if the default CloudHub log4j2.xml file is disabled for the Mule application deployment
  - Then Mule application logs will still appear in Runtime Manager and can be downloaded
  - See the docs for configuring both the **CloudHub log appender and custom log appenders at the same time**
    - Reference: Custom-CloudHub-log4j2-file.xml in the Module 9 student files

CloudHub can integrate with third party log management systems with a **custom aggregator**

- As another option, you can create a **custom aggregator application**
  - A custom application to fetch logs using CloudHub APIs and then send the logs to an external system
  - Can be done by creating an application with Mule or any other language
  - The custom aggregator application must programmatically recognize or discover new Mule runtimes (nodes) used by Mule application deployments
    - For example, when a new customer-hosted Mule runtime is added to a server group or cluster

How to integrate with third-party log management systems from customer-hosted runtime planes

- Use the Splunk or ELK plugins in Runtime Manager
  - Uses the Runtime Manager agent to exchange logs and analytics data with Splunk or ELK
  - Plugins to other third-party tool can be developed
- Fetch the logs from the third-party server side
  - Configured from the specific logging system server side to take/read the log files directly from the Mule server
  - For example, using the Splunk Universal Forwarder
- Add log4j-specific system appenders to the Mule runtime
  - Specific system appender libraries are placed in ${MULE_HOME}/lib/user

Comparing logging retention and storage limits for various runtime planes

- CloudHub logging limits the log retention period and log size
- Customer-hosted runtime can provide large storage space for logs
  - But is still limited by the available storage of the host or VM

Comparing the ability to externalize logs for various runtime planes

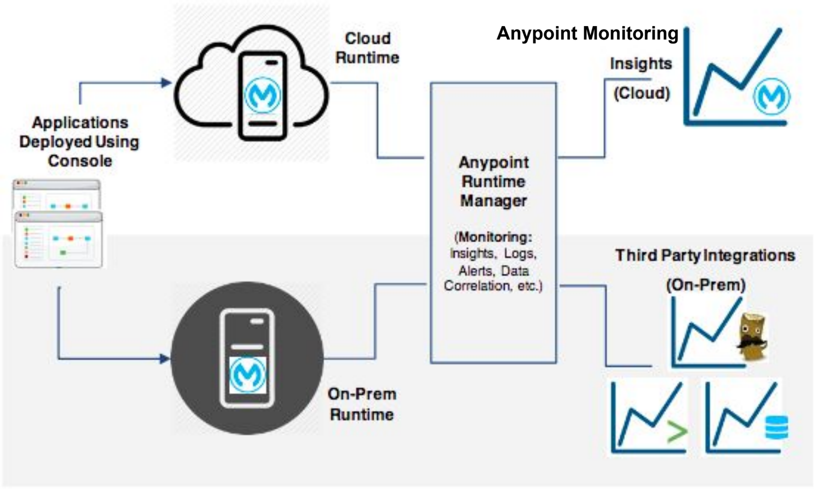
- In Mule applications deployed to CloudHub, the Mule application log messages can be sent to external log management systems
  - But the Mule application **log may be lost if the external log system fails**
  - Or if network connectivity is lost between CloudHub and the external log system
- CloudHub system log messages **cannot** be sent to external log management system without installing custom CH logging configuration through support
- Customer-hosted runtime can **send system and application log** to external log management system

Reflection questions

- What scenarios would require some or all Mule application logs to be externalized?
- What scenarios would not allow storing Mule application logs in CloudHub, and what are the tradeoffs resulting from this decision?
- What is the tradeoff when Mule application logs are only stored in CloudHub?

Choosing monitoring, alerting, and notification options

MuleSoft monitoring landscape



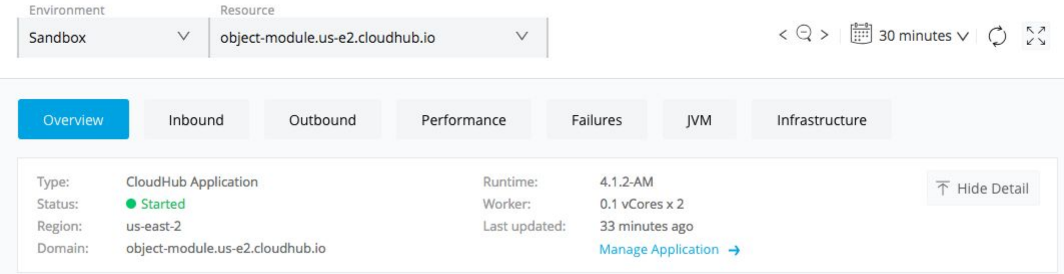
Monitoring applications

- Anypoint Platform involves different types of capabilities when dealing with monitoring
  - Anypoint Monitoring
  - Anypoint Analytics
  - Runtime Manager Dashboard stats
    - CPU, memory, mule messages
  - Business events Insight
    - Default
    - Custom

Monitoring Mule applications with the Anypoint Monitoring dashboard

Monitoring applications using the Anypoint Monitoring dashboard

- Gives insight into application usage and performance
  - https://docs.mulesoft.com/monitoring/dashboards
- Helps to view dashboards (both overview and custom)
- Can be accessed by Organization Administrators and API Version Owners/API Creators



Environment: Sandbox

Resource: object-module.us-e2.cloudhub.io

Type: CloudHub Application

Status: Started

Region: us-east-2

Domain: object-module.us-e2.cloudhub.io

Runtime: 4.1.2-AM

Worker: 0.1 vCores x 2

Last updated: 33 minutes ago

Monitoring applications using the Anypoint Monitoring **built-in dashboard**

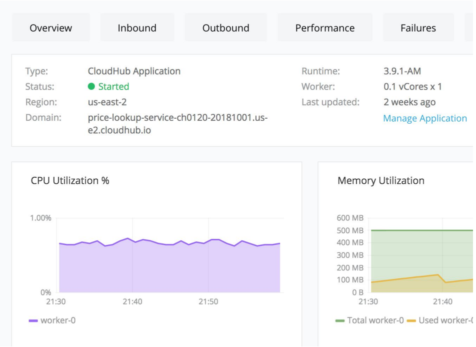
- Default build-in dashboard in Anypoint Monitoring contains a set of time series charts to collect various metrics
  - Inbound and Outbound events
  - Other performance metrics
  - API functional monitoring
- Complete details of each dashboard are mentioned at https://docs.mulesoft.com/monitoring/dashboards

Built-in dashboard **traffic** metrics

- Inbound
  - Total inbound requests, Avg. response time inbound, Total inbound requests by endpoint, Avg. response time inbound by endpoint, Total inbound requests failed
- Outbound
  - Total outbound requests, Avg. response time outbound, Total outbound requests by endpoint, Avg. response time outbound by endpoint, Total outbound requests failed
- Performance
  - Avg. response time inbound, Avg. response time outbound, Avg. response time inbound by endpoint, Avg. response time outbound by endpoint
- Failures
  - Total inbound requests failed, Total outbound requests failed, Total inbound requests failed by endpoint, Total outbound requests failed by endpoint

Built-in dashboard **infrastructure** metrics

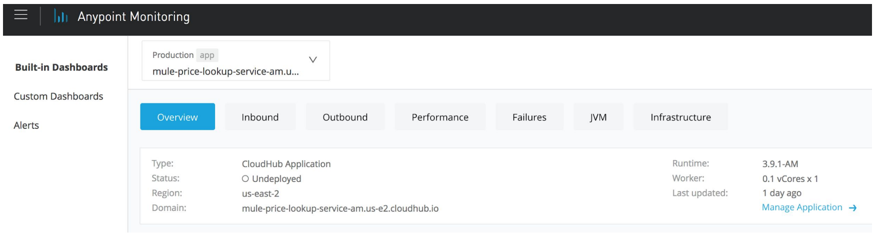
- JVM
  - GC count, Memory stats (heap and meta), Thread count
- Infrastructure
  - CPU, Memory utilization, Thread count, Total system memory and total system processor



Type: CloudHub Application
Status: ● Started
Region: us-east-2
Domain: price-lookup-service-ch0120-20181001.us-e2.cloudhub.io
Runtime: 3.9.1-AM
Worker: 0.1 vCores x 1
Last updated: 2 weeks ago
Manage Application

Exercise 9-1: Explore the Anypoint Monitoring dashboard

- Login to anypoint.mulesoft.com and go to Anypoint Monitoring
- Select environment and resource for viewing monitoring details
- Review an resource usage and performance using below Anypoint Monitoring dashboards
  - Inbound
  - Outbound
  - Performance
  - Failures
  - JVM
  - Infrastructure



Production app
mule-price-lookup-service-am.u...

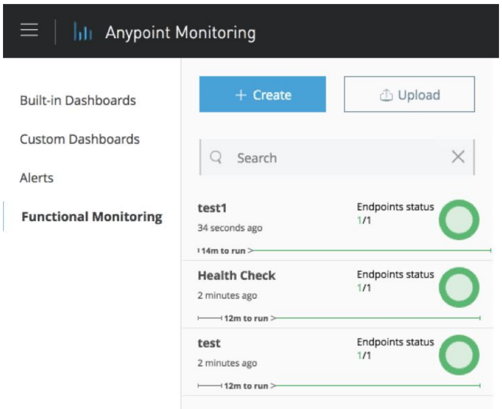
Type: CloudHub Application
Status: ○ Undeployed
Region: us-east-2
Domain: mule-price-lookup-service-am.us-e2.cloudhub.io
Runtime: 3.9.1-AM
Worker: 0.1 vCores x 1
Last updated: 1 day ago
Manage Application →

Reflection questions

- What type of information is not available in the Anypoint Monitoring dashboard?
- What are the use cases and requirements for this missing information?

Monitoring applications using Anypoint Functional Monitoring

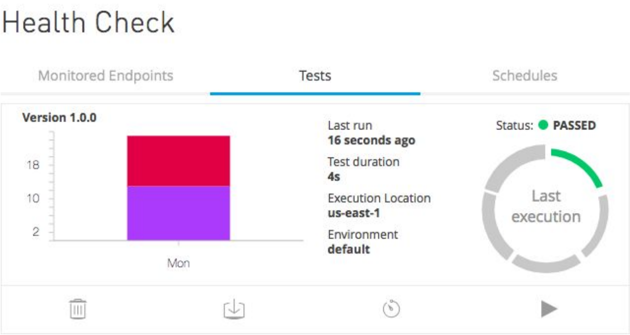
- Enables developers and operators to perform consistent testing of the functional behavior and performance of an API



test1 — 34 seconds ago — Endpoints status 1/1 — 14m to run

Health Check — 2 minutes ago — Endpoints status 1/1 — 12m to run

test — 2 minutes ago — Endpoints status 1/1 — 12m to run



Last run: 16 seconds ago

Test duration: 4s

Execution Location: us-east-1

Environment: default

Status: ● PASSED

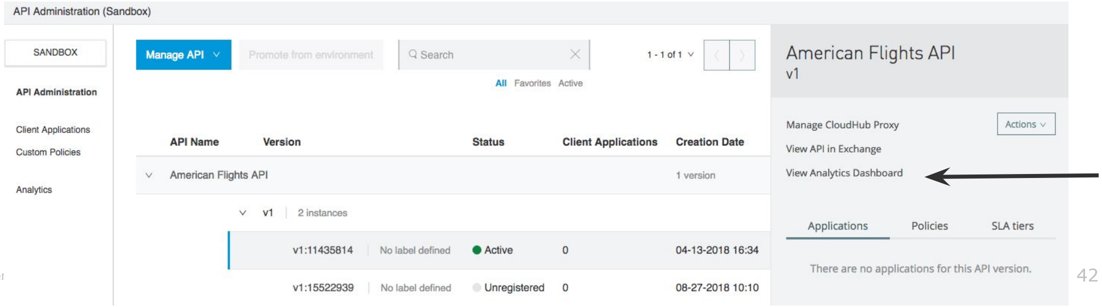
How API testing is performed

- Write tests manually and then schedule them with the Black box Automated Testing (BAT) CLI
  - Test cases are based on the DataWeave language and follows the “given-when-then” approach to describe behavior in terms of conditions and expected outcomes
- Create monitors in the **Functional Monitoring** section of Anypoint Monitoring

Monitoring APIs using the Anypoint Analytics dashboard

Monitoring APIS using the Anypoint Analytics Dashboard

- Gives insight into API usage and performance
  - http://anypoint.mulesoft.com/analytics
- Helps to view dashboards (both overview and custom), create and manage charts and reports
- Can be accessed by Organization Administrators and API Version Owners/API Creators



API Administration (Sandbox)

SANDBOX

API Administration

Client Applications

Custom Policies

Analytics

Manage API ⌄ | Promote from environment | Search | 1 - 1 of 1

All Favorites Active

API Name	Version		Status	Client Applications	Creation Date
American Flights API					1 version
	v1	2 instances			
	v1:11435814	No label defined	● Active	0	04-13-2018 16:34
	v1:15522939	No label defined	○ Unregistered	0	08-27-2018 10:10

American Flights API
v1

Manage CloudHub Proxy | Actions ⌄

View API in Exchange

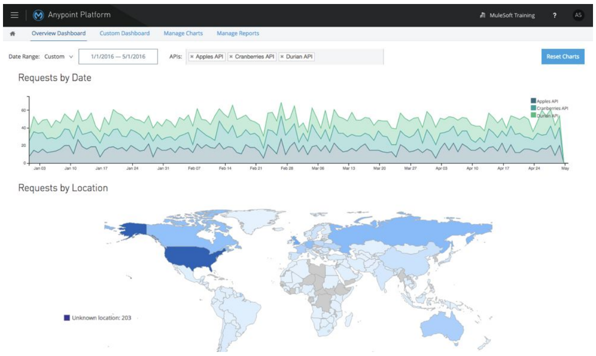
View Analytics Dashboard

Applications | Policies | SLA tiers

There are no applications for this API version.

Viewing API analytics

- The default dashboard contains a set of charts
  - Requests by date
    - Line chart representing number of requests
  - Requests by location
    - Map chart showing the number of requests for each country of origin
  - Requests by application
    - Bar chart showing the number of requests from each of the top five registered applications
  - Requests by platform
    - Ring chart showing the number of requests broken down by platform



Date Range: Custom ⌄ 1/1/2016 — 5/1/2016

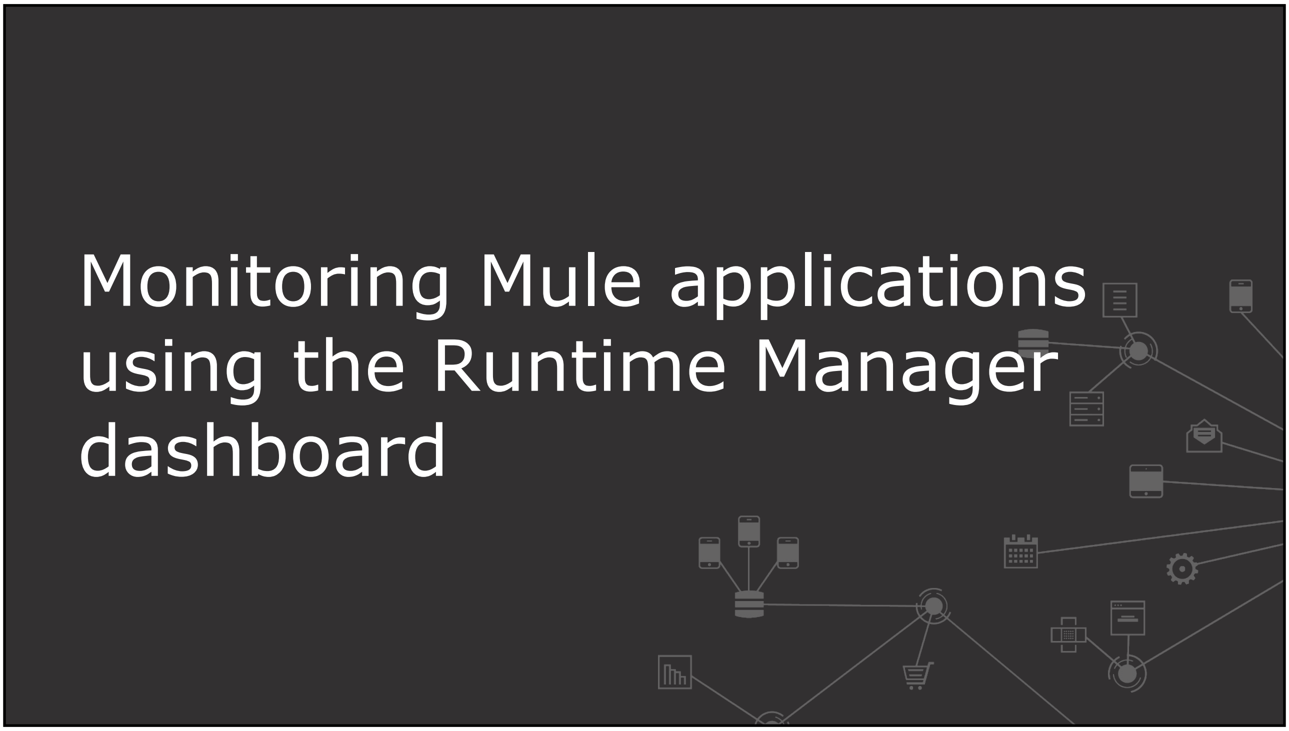
APIs: × Apples API × Cranberries API × Durian API

Exercise 9-2: Explore the Anypoint Analytics Dashboard

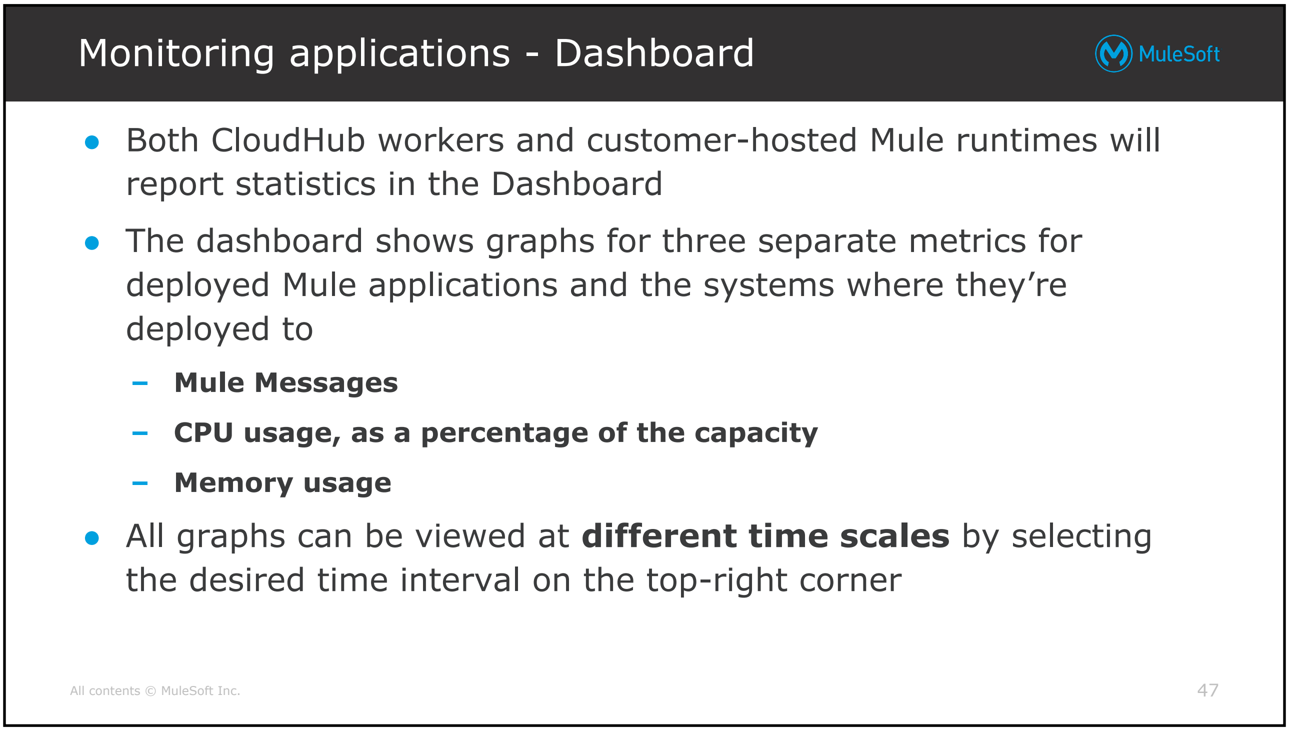
- Login to anypoint.mulesoft.com and go to API Manager
- Navigate to Anypoint Analytics
- Review an API’s usage and performance using the API Analytics Dashboard
- Explore the Overview Dashboard

Custom API dashboards

- The **custom dashboard** in Anypoint Analytics contains a set of charts for a single API or for all APIs
- Each chart displays various API characteristics
  - Requests size
    - Line chart representing size of requests in KBs
  - Requests
    - Line chart representing number of requests over a period
  - Response size
    - Line chart representing size of response in KBs
  - Response time
    - Line chart representing response time in ms



Monitoring Mule applications using the Runtime Manager dashboard



Monitoring applications - Dashboard

MuleSoft

- Both CloudHub workers and customer-hosted Mule runtimes will report statistics in the Dashboard
- The dashboard shows graphs for three separate metrics for deployed Mule applications and the systems where they’re deployed to
  - **Mule Messages**
  - **CPU usage, as a percentage of the capacity**
  - **Memory usage**
- All graphs can be viewed at **different time scales** by selecting the desired time interval on the top-right corner

Monitoring multiple Mule runtimes - Dashboard

- If your Mule application runs on **multiple workers** at a time, they will be charted as different curves on the same graphs, differentiated by different colors
- If your Mule application runs on a cluster or server group, the aggregated metrics of the entire set of servers included will be charted as a single plot line

Exercise 9-3: Access the Runtime Manager dashboard for a Mule application

- Login to anypoint.mulesoft.com and go to Runtime Manager
- Go to American Flights API version v1
- Look at the Dashboard for the Mule application
- Observe the numbers of metrics available in the dashboard for Mule applications and the servers to which the Mule application is deployed
- Verify the metrics for the Mule application's multiple workers for various time spans

Reflection questions

- What are the differences and similarities between the API Manager dashboard and the Runtime Manager dashboard?
- What information is missing in either of these dashboards, and what scenarios would require this missing information?

Configuring alerts in Runtime Manager

Monitoring applications with alerts

- **Alerts** can be configure in Runtime Manager
  - Alerts are set up for applications or servers as source at various severity such as critical, warning and info
  - Different event types are available for different event sources (Mule applications vs. servers)
- These can be triggered by Mule application or server conditions
- Alerts can send notifications to email addresses or to Anypoint Platform users within Runtime Manager
- Custom notifications can be generated by a Mule application using the CloudHub Connector, which can generate custom alerts in Runtime Manager
- Note: These alerts are distinct from API Manager alerts and Monitoring Center alerts

Alert conditions related to Mule applications

CH Mule application alerts

- CPU Usage
- Memory Usage
- Custom Notification alert
- Exceeds event traffic threshold
- Secure data gateway disconnected
- Secure data gateway connected
- Worker not responding
- Deployment success
- Deployment failure

On-prem Mule application alerts

- Number of errors
- Number of mule messages
- Response time
- Application Deployment success
- Application Deployment failure
- Application undeployed

Alert conditions related to Mule runtimes

Server alerts

- Server up
- Server disconnected
- New server registered
- Agent's version changed
- Runtime's version changed
- Server deleted
- CPU Usage
- Memory Usage
- Server Load Average
- Server Thread Count

Server Group alerts

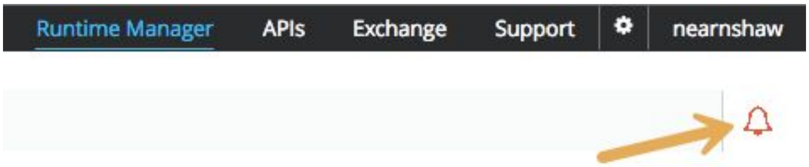
- Server added to a Server Group
- Server removed from a Server Group
- Server added to a Server Group
- Server removed from a Server Group
- Server group is up
- Server group is partially up (some servers are not running!)
- Server group is down
- A server group's node came up
- A server group's node went down

Cluster alerts

- Cluster Created
- Cluster Deleted
- Server added to a Cluster
- Server removed from a Cluster
- Cluster is up
- Cluster is down
- A cluster's node came up
- A cluster's node went down
- Cluster presents visibility issues

Monitoring applications - Notification

- Notifications give you the ability to give visibility into business related events inside your application
- Notifications are accessible from the Runtime Manager console
- If the notification is sent after an exception, it attaches the exception.message and exception.stacktrace as custom properties of the notification



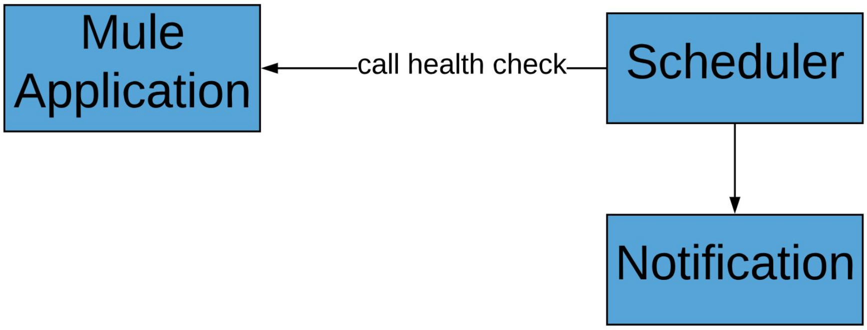
Reflection questions

- How are alerts defined and what types of alerts are possible?
- What types of alerts are missing, and which scenarios require these missing alert types?
- How can you add new alerts to a Mule application in various runtime planes?
- How are event notifications exchanged and handled?
- How can alert notifications be shared with other monitoring systems?

Other Anypoint Platform monitoring and visualization options

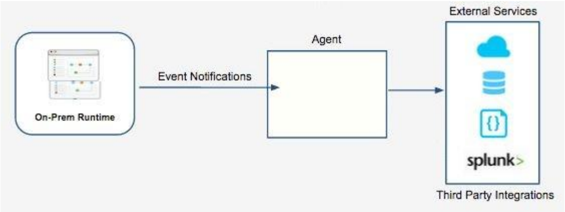
Monitoring applications - using custom component

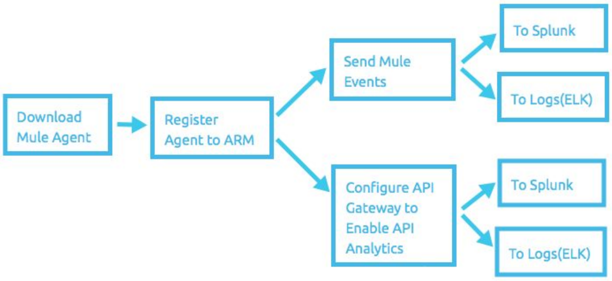
- Any Mule application can define custom health check endpoints, typically accessible via HTTP/S
- An external application or a Functional Monitor can then invoke these health check endpoints at regular intervals



Monitoring applications - using Splunk and ELK

- Applications that you deploy on-premises or to your own cloud servers can be integrated into third-party analytics applications
- **The Mule Runtime agent** enables integration and sends analytics/event notifications to your third party analytics tool





Reporting for API

MuleSoft

- Anypoint platform has reporting capability to gather analytics for APIs in CSV and JSON format
- Report provides key details such as
  - Application name
  - API name and version
  - Response time
  - Status code
  - Violated policy
- Runs on demand
- Can download sample report

Monitoring Mule applications using business events and Anypoint Platform Insight

Monitoring applications with business events

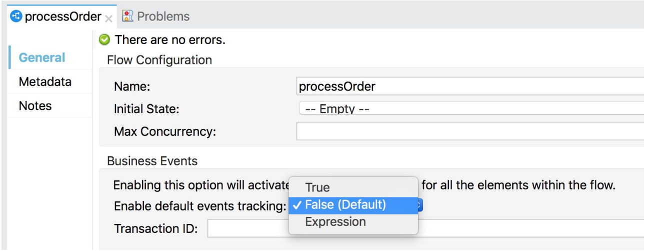
- Mule applications can collect **business events** information as each Mule event transitions through a flow
  - A Mule event's transition across a flow is also called a **transaction**
- Business events are designed to collect key performance indicators (KPIs) and store them in a MuleSoft provided online service
  - Is a convenient alternate way to log KPIs and to troubleshoot issues
- Business events are buffered in the Mule runtime's Runtime Manager agent, then synchronized with the **online** multi-tenant Anypoint Platform Business Events service
- **This process may degrade network or Mule application performance**

How and when to use business events

- When you want to store information in a MuleSoft provided multi-tenant online storage service
- Usually for coarse-grained integration related KPIs relating to entire interactions rather than specific Mule applications
  - At the end of a long chain of Mule applications, count if there were any errors or not writing to various systems
  - To confirm a performance guarantee, count the number of records in an entire batch job
- Then aggregate KPIs business events to get counts to confirm quality SLAs and generate alerts if needed, or perform other auditing

Tracking default business events

- **Default business events** can be enabled or disabled per flow
  - Default business events track and store every Mule event state through every event processor in the flow
  - Can be used even if default business event tracking is disabled
  - Can also be used to replay a Mule event from the flow's event source
- Some event processors and scopes can also enable or disable business events
  - For example, a Choice router



There are no errors.

Flow Configuration

Name: processOrder

Initial State: -- Empty --

Max Concurrency:

Business Events

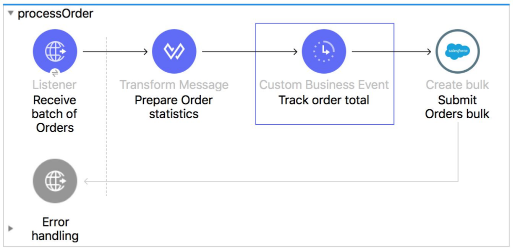
Enabling this option will activate for all the elements within the flow.

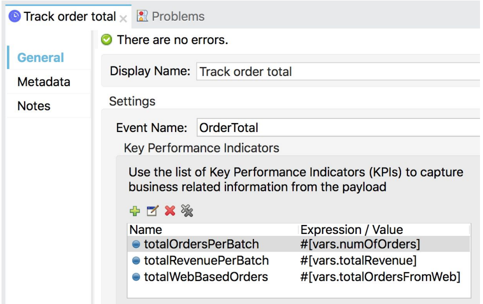
Enable default events tracking: ✓ False (Default)

Transaction ID:

Tracking custom business events

- A **Custom Business Event** component can generate additional business events at a particular place in a flow
  - Contains a list of KPIs
  - Can be added anywhere in the flow
  - Is an alternative to using the Logging component so that the Mule application can store KPIs outside the Mule application logs





There are no errors.

Display Name: Track order total

Settings

Event Name: OrderTotal

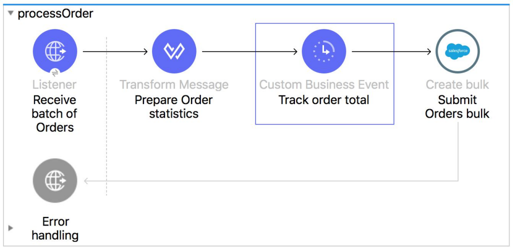
Key Performance Indicators

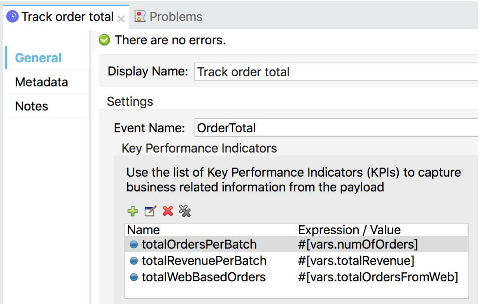
Use the list of Key Performance Indicators (KPIs) to capture business related information from the payload

Name	Expression / Value
totalOrdersPerBatch	#[vars.numOfOrders]
totalRevenuePerBatch	#[vars.totalRevenue]
totalWebBasedOrders	#[vars.totalOrdersFromWeb]

What can be stored in a custom business event?

- A custom business event can store the result of any DataWeave expression
- However, the design intent of business events is to only store small sized, usually coarse-grained, KPIs
  - Such as execution times, errors, result completion or failure notification
  - Usually should not include the entire event payload





There are no errors.

Display Name: Track order total

Settings

Event Name: OrderTotal

Key Performance Indicators

Use the list of Key Performance Indicators (KPIs) to capture business related information from the payload

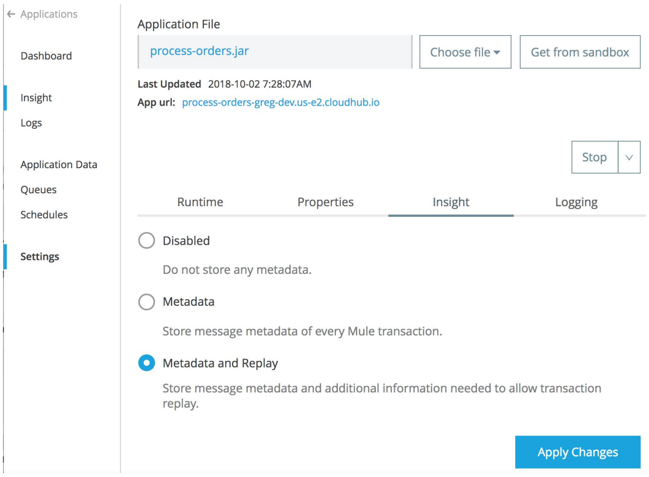
Name	Expression / Value
totalOrdersPerBatch	#[vars.numOfOrders]
totalRevenuePerBatch	#[vars.totalRevenue]
totalWebBasedOrders	#[vars.totalOrdersFromWeb]

Monitoring a Mule application's business events using Insight

- **Insight** can also be used as a troubleshooting tool that gives you in-depth visibility into business transactions and Mule events
  - Through the inspection of Business Events
- Insight helps to
  - Search transactions
  - Provide information about transactions
    - Status such as success/failure, processing time
  - Find and recover from any errors that occurred during message processing and replay your transactions instantly
- Using Insight, you can monitor business events at runtime
  - To analyze the root cause of failures, isolate performance bottlenecks, and test for compliance to company procedures

Enabling default business events for a Mule application

- Enabled in the Runtime Manger Insight tab
  - In Insight it is called Metadata
- For Mule applications deployed to CloudHub, event replay from each flow's event source can also be enabled
  - Every Mule event is then streamed from the Mule runtime to the online business events storage service
  - Should only be used briefly for troubleshooting purposes
- Default business events tracking will be enabled for every flow and component
  - Even if it is currently configured a disabled in the Mule application



Application File

process-orders.jar

Last Updated 2018-10-02 7:28:07AM

App url: process-orders-greg-dev.us-e2.cloudhub.io

○ Disabled

Do not store any metadata.

○ Metadata

Store message metadata of every Mule transaction.

● Metadata and Replay

Store message metadata and additional information needed to allow transaction replay.

Insight performance impact and limitations

- Enabling Insight has a **large performance impact** when processing application data
  - Typically Insight should not be enabled for extended periods (or ever) in production environments
- Replay only works for flows that have inbound endpoints that are deployed to CloudHub workers
- Batch processing is not supported
- The Anypoint Platform - Private Cloud Edition doesn’t currently support Anypoint Insight

Reflection questions

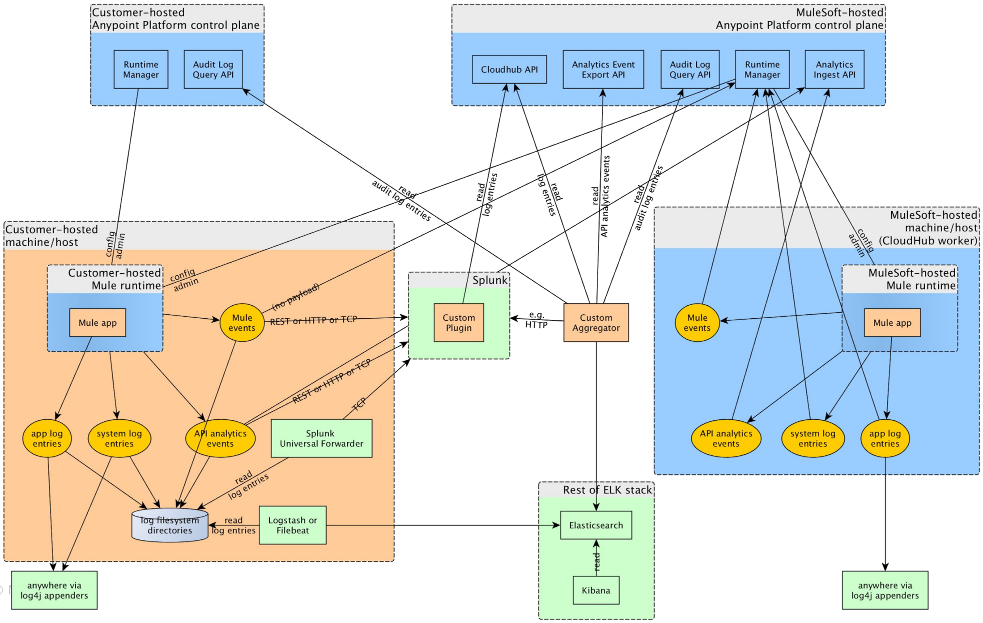
- What use cases would require using Business Events and Anypoint Insight?
- What are the tradeoffs of using Insight in a production environment?
- What are some alternatives to using Business Events, and what are the tradeoffs?
- Can Business Events be exchanged with another log management system such as Splunk or ELK?
- Can Business Events be written to a log file?

Summary

Summary

- Logger processor logs the entire Mule event, including all attributes, as well as flow variables except content of payload
- By default, logging in Mule is done asynchronously
- Anypoint platform provides Anypoint Monitoring, Analytics, Insights, Dashboards and Flow metrics for analyzing and monitoring applications/servers for performance and debugging
- Anypoint Monitoring is next generation tool from MuleSoft for analyzing and monitoring applications/servers for performance and debugging

Overview of log integrations



References

- CloudHub Custom Log Appender
  - https://docs.mulesoft.com/runtime-manager/custom-log-appender
- Splunk (Universal Forwarder):
  - http://docs.splunk.com/Documentation/Forwarder/7.1.1/Forwarder/Abouttheuniversalforwarder
- Log4j specific-system **appenders**
  - Splunk Logging Java: http://dev.splunk.com/view/splunk-logging-java/SP-CAAAE3P#_Maven
  - TCP/Socket appender: http://dev.splunk.com/view/splunk-logging-java/SP-CAAAE3R