**Topic: Runtime Manager / Exchange / API Manager – Capabilities**

**Example: Extract, Transform and Load using Batch Processing**

Extract

Extract records from database Orders using ‘On Table Row’ operation of Database Connector which polls for new records added to db by checking against watermark column (OrderID) and ID column.

It returns Java objects of each record sequentially.

Using Batch Processing, the records are transformed and loaded/inserted to database again.

Transform

Firstly, order and list of SKU saved as variables

In batch job, first batch step to query SKU Product names.

Second batch step to Query Order Status.

Third batch step: Insert

**Connector: Anypoint MQ Connector**

Anypoint MQ is an enterprise multi-tenant, cloud messaging queuing service that is fully integrated with Anypoint Platform.

With Anypoint MQ, users can perform advanced asynchronous messaging scenarios such as queueing and pub/sub with fully hosted and managed cloud message queues and exchanges. A service of Anypoint Platform, Anypoint MQ supports environments, business groups, role-based access control (RBAC) to help you deliver seamless customer experiences across channels and integrate devices reliably for Internet of Things (IoT) applications with enterprise-class functionality.

Ref : <https://www.mulesoft.com/resources/cloudhub/what-is-anypoint-messaging-queue>

<https://docs.mulesoft.com/mq/mq-understanding>

<https://docs.mulesoft.com/mq/>

<https://docs.mulesoft.com/mq/mq-exchanges>

When we don’t have enterprise account to create anypoint MQ, we can use AMQP connector.

<https://docs.mulesoft.com/amqp-connector/1.6/>

Anypoint MQ Connector: Cloud Messaging for the Enterprise

The Anypoint MQ connector enables Mule applications to connect to Anypoint MQ, MuleSoft's multi-tenant cloud messaging service for the enterprise. With it, you can publish and receive messages, acknowledge messages, and delete messages from queues and exchanges.

Anypoint MQ supports hybrid use cases, IoT, where you collect data from different devices, and a [REST API](https://docs.mulesoft.com/anypoint-mq/mq-apis) you can use with device applications. You can also use Anypoint MQ with other application frameworks such as node.js to communicate to queues.

**Diff between Anypoint MQ connector vs JMS Connector vs VM connector :**

They all use different queuing/messaging protocols/platforms underneath.

VM is Mule's internal transport for messaging/queueing. The VM transport is for intra-JVM communication between Mule flows. So, that means when you use a VM in your flow, you can communicate between different flows in the application. It can only be used by Mule applications. When creaing a VM queue it can only be accessed by the Mule application that creates it (Cloudhub for example) OR it can be reused by same Mule apps running in a domain project or cluster. No existing broker infrastructure needs to be setup. Supports persistent and transient.

Anypoint MQ is Mulesoft's Cloud Messaging platform. This can be used by other applications - not just Mule. It can also be used across multiple Mule apps regardless of domains or cluster, well suited for Cloudhub applications. No infra setup, all in the cloud. Think Amazon SQS but a lot better and great integration with Anypoint Platform

JMS uses the Java Messaging Service protocol and requires an external JMS broker such as ActiveMQ. Can be used by any application that supports JMS connectivity.

<https://help.mulesoft.com/s/question/0D52T00004naR1C/diff-between-anypoint-mq-connector-vs-jms-connector-vs-vm-connector>

**Security Policy: Client-id enforcement**

Ref: <https://dzone.com/articles/mule-4-client-id-enforcement>

The purpose of the Client ID Enforcement policy is to allow access only to authorized client applications.

The Client Id Enforcement policy is used to restrict access to a protected resource by allowing requests only from registered client applications. The policy ensures that each request, which contains valid client credentials is able to access protected resources.

The client application has to be registered on the AnyPoint platform to generate client credentials (client\_Id  and  client\_secret). After client application registration, all subsequent requests have to pass client\_id and client\_secret, as part of the request while invoking an API.

There are some policies which internally enforce client application credentials. Those are:

* Rate Limiting - SLA-Based Policies.
* OAuth 2.0 Access Token Enforcement.
* [JWT Validation](https://dzone.com/articles/state-of-api-security).

Before a client application is allowed to consume an API protected by a Client ID Enforcement policy, the client application must request access to the API. After an approved contract exists between the client application and the API, every request must include the client application credentials, according to how the policy is configured.

For example, if the policy is configured to expect a client ID and client secret as headers, the application must send those credentials in the request, using the corresponding headers.

**Obtaining Credentials Using HTTP Headers**

#[attributes.headers.['client\_id']]

#[attributes.headers.['client\_secret']]

In this example, the policy is configured to expect two headers: client\_id and client\_secret, with the pair of credentials. The policy is flexible to allow other types of headers also. This is the default configuration for the policy. This can be reflected as traits in API specification.

**Traits** is like function and is used to define common attributes for HTTP method (GET, PUT, POST, PATCH, DELETE, etc) such as whether or not they are filterable, searchable, or pageable

**Obtaining Credentials Using HTTP Query Parameters**

#[attributes.queryParams.'client\_id']

#[attributes.queryParams.'client\_secret']

The requester must send the two specified query parameters with the request. Although this is a supported configuration, it poses possible security risks. The recommended method is to use headers.

**Obtaining Credentials Using HTTP Request Payload**

#[payload.client\_id]

#[payload.client\_secret]

Although you can configure the policy to obtain the credentials from the request payload, this option is not recommended because it is harder to reflect in the API specification.

**Steps to enforce cliend-id policy**

1. Design API specification using RAML. Define client credentials as traits, required as headers.
2. Publish it in Anypoint Exchange.
3. Create an API instance with a basic endpoint or proxy endpoint, using published API from Exchange.
4. Not API instance ID.
5. In the policies section, Click on Add new policy.
6. Add Client-id enforcement policy.
7. Keep default configurations for extracting credentials.
8. Create mule project in Anypoint Studio by importing API from Exchange.
9. Automatically mule flows will be created based on API specification with APIKIT router.
10. If we test the app at this point, it will throw an error as the app needs client-id and client-secret as headers based on traits specified in RAML. The request will be served with any client credentials provided because the Client-id policy is not yet added to this mule project.
11. To add the policy, we need to set global elements. Global elements -> create and search for API Autodiscovery. Enter API Id and save it.
12. After deploying this application, we will get an error with any random values of client credentials. To get genuine credentials, we need to ‘Request access’ to the PAI from the Exchange in Anypoint Platform.
13. Create an client application in the Request Access Window, which will give client-id and client-secret for that particular client application.
14. Now, by sending a request to the application with these client credentials, we can get a response.

**Design Pattern: Splitter pattern**

The **Splitter** is an EIP pattern that describes a mechanism for breaking composite messages into parts in order that they can be processed individually. Mule objects associated with splitter EIP – For Each scope, Parallel for each, batch.

The For Each scope splits a payload into elements and processes them one by one through the components that you place in the scope. For Each does not modify the payload, while the Parallel For Each outputs a collection of the output messages from each iteration.

Batch Processing

Mule allows you to process messages in batches.

Within an application, you can initiate a Batch Job scope, which splits messages into individual records, performs actions upon each record, and then reports on the results and potentially pushes the processed output to other systems or queues.

Batch jobs allow for the description of a reliable process that automatically splits up source data and stores it into persistent queues, which makes it possible to process large data sets while providing reliability. In the event that the application is redeployed or Mule crashes, the job execution is able to resume at the point it stopped.

**Topic for issues from Mulesoft Help Center: XML transformation errors**

**Additional Activity**

1. **What is control plane and runtime plane?**

<https://blogs.mulesoft.com/dev/anypoint-platform-dev/deployment-options-anypoint-platform/>

<https://docs.mulesoft.com/eu-control-plane/>

**Control plane** is referred to an environment that provided necessary tools and services to design, reuse, implement and manage integrations and APIs.

The Anypoint Platform control plane provides a set of cloud services that simplify the design, reuse, and management of integrations and APIs. These cloud services are available in a multi-tenant environment of the Anypoint Platform in the U.S. and [EU regions](https://docs.mulesoft.com/eu-control-plane/). MuleSoft also offers a single-tenant environment via [Private Cloud Edition](https://docs.mulesoft.com/private-cloud/latest/), which allows customers to deploy the entire Anypoint Platform on-premise or on private clouds, such as AWS.

**Available control plane options:**

**US Cloud:**

* multi-tenant
* provides extensive list of features and services for design, reuse, security and management of integrations and API.
* industry security standards: offers out-of-the-box compliance for ISO 27001, SOC 2, PCI DSS, and HIPAA.

**EU Cloud:**

* Multi-tenant configuration
* GDPR compliant- General Data Protection Regulation
* ensures all personal data of EU citizens is stored and processed within the EU boundaries,

**Government Cloud:**

* FedRAMP compliant : Federal Risk and Authorization Management Program (FedRAMP) is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services.
* Designed specifically for US government
* includes hundreds of enhancements in access control, auditing, and encryption.
* provides continuous security monitoring with process improvements in threat detection and response, patch and vulnerability management, personnel, and training.

**Private Cloud Edition:**

* single-tenant version of Anypoint Platform
* offers full control of the integration lifecycle within your own data centers
* We have control over data storage and processing
* How? Install control plane appliance onto your own data servers, such as on-prem or AWS
* leverages Docker and Kubernetes technologies to provide built-in high availability and scalability of all management features

The best hosting option for your Anypoint Platform environment depends on your security, data privacy, and legal requirements. Different hosting options enable you to control the location where your and your customer’s data is stored, processed, and managed. The data remains segregated within the physical location required by your companies IT policies and requirements.

<https://docs.mulesoft.com/general/intro-platform-hosting>

The Anypoint Platform **runtime** **plane** is where applications are deployed, and also where the Mule runtime engine and other application-related services, such as Anypoint Connectors run. The runtime engine includes Anypoint Security edge policies and tokenization, MQ, Object Store, and Connectors.

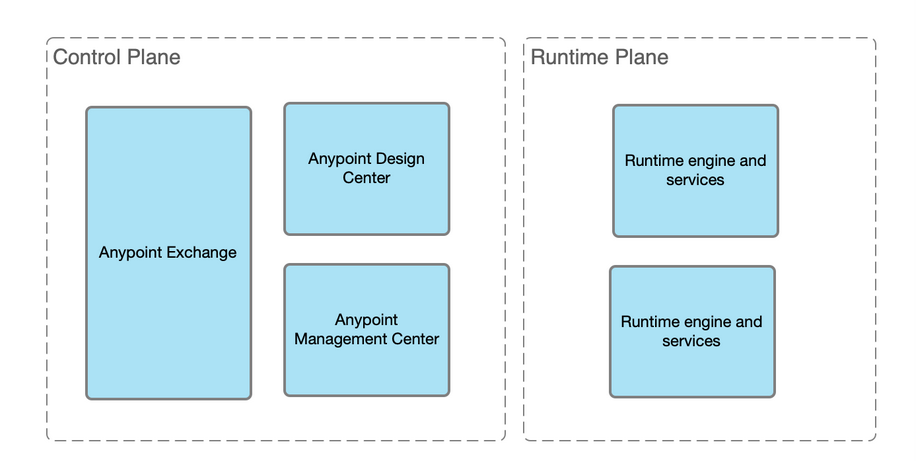
To deploy your applications and APIs via runtime plane, there are three options: CloudHub, Anypoint Runtime Fabric, or on–premises Mule runtime engine instances. Additionally, you can operate a hybrid environment to include both CloudHub and on-premises functionality.

**CloudHub:** CloudHub allows Anypoint Platform customers to host and manage Mule runtimes in the MuleSoft cloud. Enterprise-grade capabilities include 99.99% uptime, one-click scalability, and automatic updates.

**Runtime Fabric:** Using Anypoint Runtime Fabric deploy Mule runtimes within your own data centers, whether it’s in a private IaaS (Microsoft Azure or AWS) or on-premises infrastructure. Runtime Fabric provides horizontal scalability and zero downtime redeployments while keeping runtimes within your full control.

**On-premises:** Use standalone Mule instances when you want to host and manage your own infrastructure.

<https://docs.mulesoft.com/general/intro-platform-hosting>



* Control plane

The components of the Anypoint Platform architecture that you use to design, deploy, and manage APIs and Mule applications. The control plane contains product features and components that are part of the Anypoint Design Center, Anypoint Management Center, and Anypoint Exchange.

* Runtime plane

The components of the Anypoint Platform architecture to which your APIs and Mule applications are deployed and made available to your users. The runtime plane includes the Mule runtime server and supporting services.

**Runtime plane options:**

**CloudHub**

- provides high availability, clustering and failover of your APIs and Mule applications and performs load balancing for them

- can only be managed by cloud control plane in the same region

**Runtime Fabric**

**-** Runtime Fabric is a container service that enables you to run Mule applications and API gateways within a data center or third-party cloud environment that you control and manage.

- We can install Runtime Fabric on a set of physical servers, virtual machines, or within Amazon Web Services and Microsoft Azure.

- Runtime Fabric comes bundled with technology such as Docker and Kubernetes, which offer benefits such as high availability, failover, clustering, and load balancing.

- can be managed only by Cloud control planes (US cloud, EU cloud).

- MuleSoft Government Cloud and Anypoint Platform PCE do not support Runtime Fabric.

**Stand-alone runtime**

* The standalone option enables you to host Mule runtime engine server and related services in an environment that you manage.
* Using standalone runtimes, the Mule runtime server can run on a physical server, a virtual machine, or within a third-party cloud installation like Amazon Web Services or Microsoft Azure.
* When using standalone runtimes, you are also responsible for providing the framework for high availability, failover, clustering and load balancing.
* can manage standalone runtimes by using cloud control planes (US Cloud, EU Cloud, MuleSoft Government Cloud) or a customer-hosted control plane (Anypoint Platform PCE).

**2. Key capabilities of Runtime manger, exchange, API manager?**

**3. How do API's in runtime plane communicate with control plane?**

**References during session**

<https://docs.mulesoft.com/object-store/osv2-faq>

<https://docs.mulesoft.com/object-store/osv2-faq#can-one-mule-app-access-the-object-store-of-another-mule-app>

<https://blogs.mulesoft.com/dev/mule-dev/how-poll-scope-and-watermark/>

<https://help.mulesoft.com/s/article/Scheduler-getting-Disabled>