**Integration Solutions Knowledge Base**

**Stakeholders for Integration Project**

* Product Owner
* Project Sponsors
* Architects / Developers/ Testing/ Operations/ Admins
* Business Users
* Business Analysts
* Infrastructure Team
* Clients/Consumers
* Project Management Team
* Security Team
* Network Team
* External Application / 3rd Party Apps / Partner Applications
* Content Writers
* CXX Suite
* Data Architects (CDM, EDM - Data Model)
* Legal / Audit Teams
* MuleSoft
  + C4E Team (Center For Enablement)
  + Internal Support Team

**Mulesoft Catalyst**

* <https://knowledgehub.mulesoft.com/>
* Framework/Approach
* 3 Dimensions / 3 Aspects
  + Business Outcomes
  + Technology Delivery
    - Anypoint Platform
    - Projects
  + Organization Enablement
    - C4E
    - Internal Support
    - Trainings

**4+1 Views**

* <https://en.wikipedia.org/wiki/4%2B1_architectural_view_model>

**Anypoint Platform**

* iPaaS Solutions from MuleSoft
* Control Plane
* Runtime Plane

**Control Plane**

* Is a Web Application to control lots of features of Anypoint Platform and Runtime Planes
* Features
  + Design Center
    - API Designer
      * API Specifications using RAML/OAS
      * Editors
        + Text Editor
        + Visual Editor (Point & Click Interface)
    - Flow Designer
      * API Implementation
      * Used Only for Simple, Ad-hoc POC requirements
        + Not for Full-fledged Development

For Full-fledged Development use Anypoint Studio

* + - * Limitations
        + Cannot integrate with External Repos, SCM (Source Control Management)
        + Does not support the following

MUnit

Error Handling

APIKit

ASync Scope

* + Exchange
    - Sharing and Collaboration
    - Private
    - Public / Dev Portal
      * Limited Customization
    - Use Case Scenario
      * Create Dev Portal to support the following
        + FAQs
        + Forums
        + Blogs
        + Service Management
        + Hackathons
        + Filter Assets
        + Customizations
        + Personalization
      * Question: Which Mulesoft product supports these requirements
      * Answer: ACM (API Community Manager)
        + Additional License
        + Uses Salesforce Community/Experience Cloud
  + Data Graph
    - Unify many APIs into a single graph of data, and consume exactly the data you need in a single request.
    - <https://docs.mulesoft.com/datagraph/datagraph-qsg>
  + Management Center
    - Access Management
      * Manage Users, Teams, Business Groups, External IdP, Client Providers, etc
    - API Manager
      * API Management
      * Manage API Security
    - Partner Manager
      * B2B integration
      * <https://docs.mulesoft.com/partner-manager/2.x/>
      * Additional License
    - Runtime Manager
      * Manage Application Deployments, Hybrid Servers, VPC, VPN, DLB, RTF Server, etc
    - MQ
      * Manage Anypoint MQ
        + Creating Queues, Exchange, Clients, etc
    - Visualizer
      * To Identify Application Network
    - Monitoring
      * Identify Performance Metrics and KPI
    - Secrets Manager
      * Used by API Manager, Partner Manager, RTF
      * Manage Certificates, Security Credentials, TLS Config, Key Store, Trust Store
    - Anypoint Security
      * Used by RTF only
      * Features
        + Security Policies
        + Tokenizations

**Deployment Options for Control Plane**

* Managed By Mulesoft
  + US Control Plane
    - anypoint.mulesoft.com
  + EU Control Plane
    - eu1.anypoint.mulesoft.com
* Managed By Customer
  + Customer installs the Control Plane UI in their Data Center
  + PCE
  + Additional License
  + Mulesoft Provides Control Plane UI Software as Dockers and Kubernetes
  + <https://docs.mulesoft.com/private-cloud/3.0/>

**Runtime Plane**

* Is a Collection of Infrastructure and Mule Runtime Software
* Managed By MuleSoft
  + **CloudHub**
    - Uses AWS Services (EC2 Instance, Elastic Load Balancer, S3 Buckets, Route 53, Amazon DynamoDB, AmazonSQS, etc)
    - Mulesoft provides Infra
    - Features
      * Multiple Mule Runtime Versions, Runtime Install/upgrade/patching, Global Deployment, Scalability, Availability Zones, Load Balancer(SLB- Round Robin), Zero-Downtime Redeployments, Auto-Restart
    - Benefits
      * Clients with No Regulatory Requirements
      * Less IT Operations Effort
      * Cost Effective with No Additional Infra setup and Operations requirements
      * Ready to use - Time-to-market
      * New Features Availability
* Managed By Customer
  + Customer Provides Infrastructure
  + Benefits
    - Clients with Regulatory Requirements/Data Residency Requirements
    - Clients having existing Infra Setup and operations Team
    - Control Mule Runtime and JVM Tuning
    - Less Latency to access Data Source
    - Flexibility to deploy in Private Clouds
  + **On-Prem**
    - Manual: Multiple Mule Runtime Versions, Runtime Install/upgrade/patching, Global Deployment, Scalability, Availability Zones, Load Balancer, Zero-Downtime Redeployments, Auto-Restart
    - Install Options
      * Bare Metals (Physical Machines)
      * Virtual Machines
      * Private Cloud Instances (AWS, Azure, GCP, PCF, etc)
      * Docker and Containers
    - Deploy Options
      * Stand-Alone
      * Cluster
        + Hazelcast
  + **RTF - Runtime Fabric**
    - Uses Dockers and Kubernetes
    - Customer-Hosted Solution
    - Features: Multiple Mule Runtime Versions(Images), Runtime Install/upgrade/patching(Images), Zero-Downtime Redeployments, Auto-Restart
    - Manual: RTF Cluster Install/Upgrading, Global Deployment, Scalability, Availability Zones, Load Balancer
    - Types
      * RTF Appliance
        + Customer Provides Infrastructure
        + Mulesoft provides Docker, Kubernetes, RTF Components
      * RTF on Self-Managed Kubernetes (BYOK)
        + Customer Provides Infrastructure,Docker, Kubernetes
        + Mulesoft provides RTF Components
        + Currently Supports EKS(Amazon), AKS(Microsoft), GKE(Google)
        + Future Roadmap: OpenShift, Any Kubernetes
    - RTF supports only Mulesoft Managed Control Plane

**Deployment Options for Control Plane and Runtime Planes**

* Managed By MuleSoft
  + Mulesoft Managed Control Plane
  + Mulesoft Managed Runtime Plane(CloudHub)
* Managed By Customer - PCE (Private Cloud Edition)
  + Customer Managed Control Plane
  + Customer Managed Runtime Plane(On-Prem)(Anywhere except PCF)
    - PCF - Pivotal Cloud Foundry
      * Customer Managed Control Plane
      * Customer Managed Runtime Plane(On-Prem)(PCF)
* Hybrid
  + Mulesoft Managed Control Plane
  + Customer Managed Runtime Plane(Anywhere except PCF)

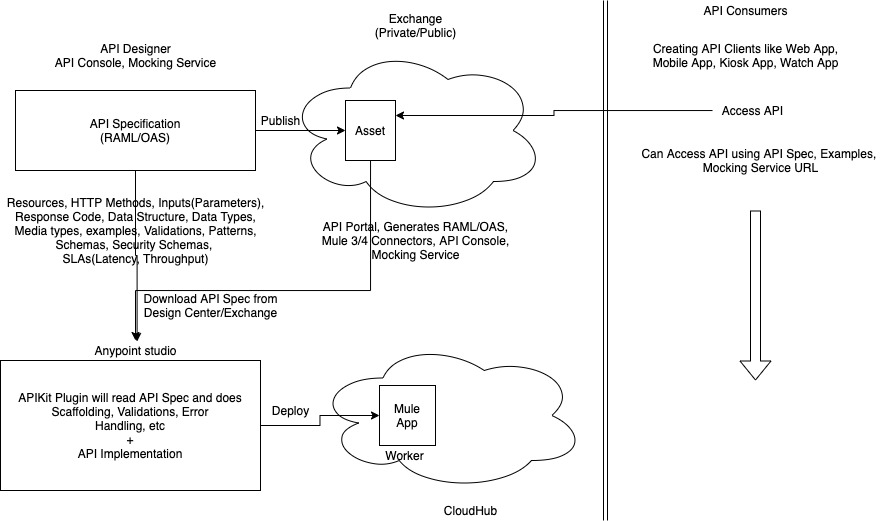
Note:

* CloudHub and RTF supports only Mulesoft Managed Control Plane
* PCF supports only Customer Managed Control Plane

**API Development Lifecycle**

* Design
* Implement
* Manage
  + Deploy
  + Secure
  + Scaling
  + Troubleshoot
  + Monitoring/Analyzing

**Design First Approach / API-Led Development**



**Access Private Mocking Service Endpoint**

<https://help.mulesoft.com/s/article/How-To-Access-The-Exchange-Mocking-Service-Endpoint>

**Behavioral Headers for Mocking Service Only**

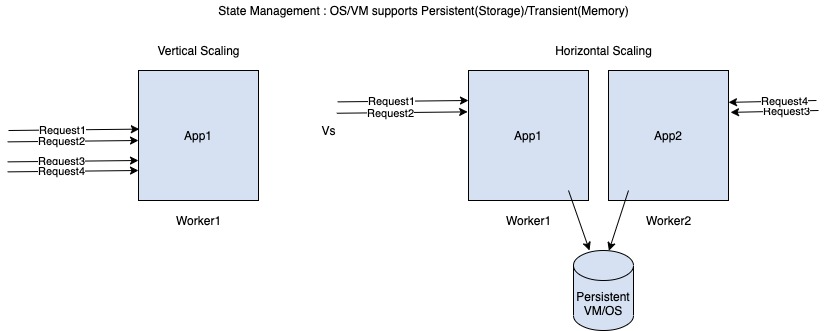
* <https://docs.mulesoft.com/design-center/apid-behavioral-headers>

**Deploying Mule Application to CloudHub**

* Name
  + Globally Unique
* Deployment Target
  + CloudHub
* Application File
  + Mule 3 - \*.zip
  + Mule 4 - \*.jar
* Tabs
  + Runtime
    - Runtime Version
    - Region
      * 1 Region Per App
    - Worker Size
      * Increase/Decrease vCores and MEmory
      * Vertical Scaling
      * Min: 0.1 vCore (500 MB Memory)
      * Max: 16 vCores (32 GB Memory)
      * Depends On
        + Mule Event Object Size Per Request
        + Complexity
        + Performance
        + TPS (Throughputs)
        + Supports State Management using Persistent and/or Transient OS/VM
    - Workers
      * Increase/Decrease the Number of Workers
      * Horizontal Scaling
      * Min: 1 Per App
      * Max: 8 Per App
      * Depends On
        + TPS (Throughputs)
        + Distributed Requests

Load Balanced

* + - * + Performance
        + High Availability
        + Supports State Management using only Persistent OS/VM
  + Properties
  + Insights
  + Logging
  + Static IP



**Hybrid Deployment Model**

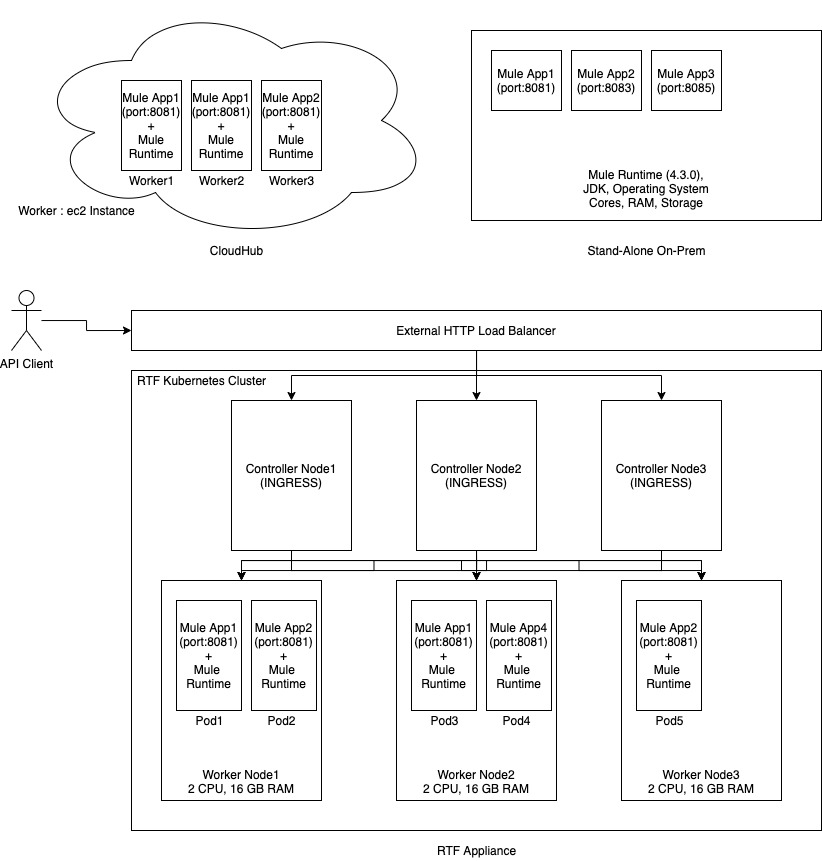
**Connecting Mulesoft Managed Control Plane With Customer-Managed Runtime Plane(On-Prem)**

1. Download Mule EE Distribution File
   1. <https://www.mulesoft.com/lp/dl/mule-esb-enterprise>
2. UnZip
3. (optional) Rename to node1
4. Open Command Prompt @ node1/bin
5. Navigate to Anypoint Platform Control Plane > Run-Time Manager > Sandbox Environment> Servers
   1. Click Add Server Button
      * Copy the command
6. Execute the command in the command prompt
   1. (optional)Replace server-name with node1
   2. Copy the command which is generated
   3. Execute the command in the command prompt @ node1/bin
      * Windows
        + Starts with amc\_setup
      * Mac/Linux
        + Starts with ./amc\_setup
7. After successful execution of the command, goto Runtime Manager and check that Server is listed and status is “Connected”
8. From Command Prompt, now execute mule(windows) or ./mule(Mac/Linux)
9. After successful execution of the command, goto Runtime Manager and check that Server is listed and status is  “Running”

**Deploying Mule Application to On-Prem**

* Name
  + Unique within the Local Server
* Deployment Target
  + <<ServerName>>
* Application File
  + Mule 3 - \*.zip
  + Mule 4 - \*.jar
* Tabs
  + Deployment Target
    - Read Only
  + Properties
  + Flow Monitoring
  + Alerts History
  + Insights
  + Logging

**CloudHub Vs On-Prem(Stand-Alone) Vs RTF Appliance**



**Flow Types**

* Regular Flow
  + Contains Event Source(1) , Event Processor(1+), Error Handling
* Private Flow
  + Is a Regular flow without an Event Source
  + Contains Event Processors, Error Handling
  + Invoke using
    - Flow Reference
    - Dataweave lookup()
    - APIKit Router
* Sub Flow
  + Contains Event Processors only
  + Invoke using
    - Flow Reference
    - APIKit Router
  + Better in Performance
  + Limitations
    - Does not support the following
      * MaxConcurrency
      * Initial State
      * Insights
      * Flow Level Monitoring
      * Flow Level Management (On-Prem)
    - Cannot invoke using Dataweave lookup()

**Invoking External APIs/Services**

* Connector
  + Simplifies Development
  + Shields the complexity by displaying properties for Query Params, URI Params, Headers, Security, etc
  + Easy Version upgrades
  + Types
    - Pre-built
    - Custom
      * Mule SDK
        + <https://docs.mulesoft.com/mule-sdk/1.1/>
      * Publishing API Spec to Exchange
        + REST Connectors

Internally uses Mule SDK with XML

* HTTP Request
  + Invokes REST APIs
  + Can invoke SOAP Services
* Web Service Consumer
  + Invokes SOAP Services
  + Limitations
    - Impacts Performance because it reads WSDL during Design and Runtime
    - Does not support RPC style of SOAP Services
      * Supports only DOC Style of SOAP Services

**Implementing API Spec in Studio**

* APIKit Plugin will download API Spec(RAML/OAS) and does scaffolding, Validations, Error Handling
* Download API Spec into Studio
  + Download RAML From Design Center
    - Downloads API Spec in src/main/resources/api Folder
    - Limitations
      * Only one API Spec can be downloaded
        + If try to download another API spec, it overrides the earlier API Spec
      * Supports only RAML
  + From Exchange
    - Many API Specs can be downloaded
    - Supports both RAML and OAS
    - Right-Click on Project > Manage Dependencies > Manage APIs
      * Click “+” > From Exchange
    - Downloaded as Maven Dependency (pom.xml)
* When downloading multiple API Specs in a single Mule Projects, Need to resolve the following ambiguities
  + Multiple HTTP Listener Configs are created per API Spec
    - Need to delete one with 8082+
  + HTTP Listener paths in Main flow and Console Flow
    - Need to be unique

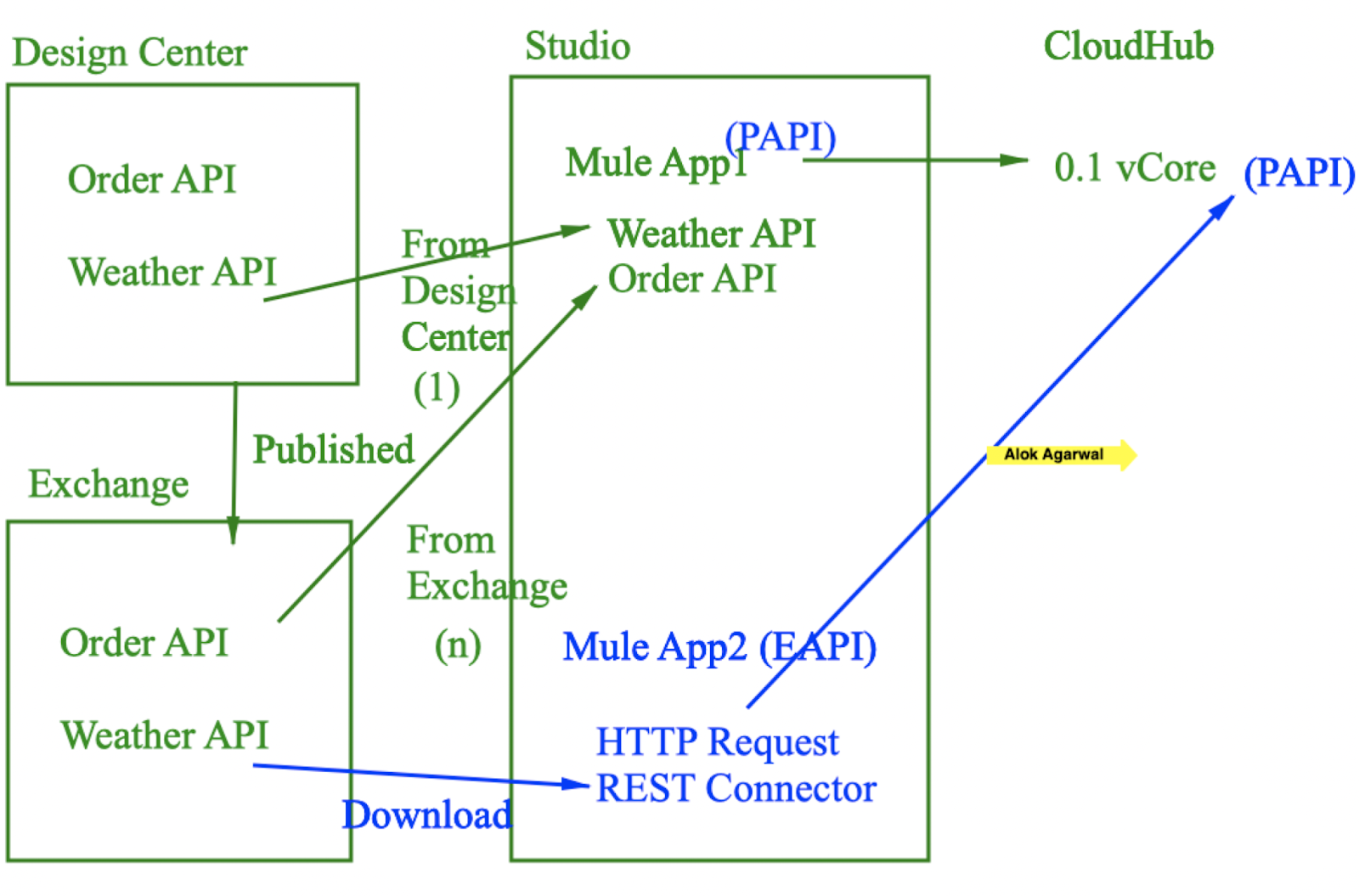
**Invoking Mule APIs**

* Connector
  + Downloaded
    - If visible/accessible in Exchange
      * Mule Palette > Search In Exchange
      * Right-Click on Project > Manage Dependencies > Manage Modules
        + Click “+” > From Exchange
    - If not visible/accessible in Exchange
      * Right-Click on Project > Manage Dependencies > Manage Modules
        + Click “+” > From Maven

Click “Install” next to Install a Local Dependency

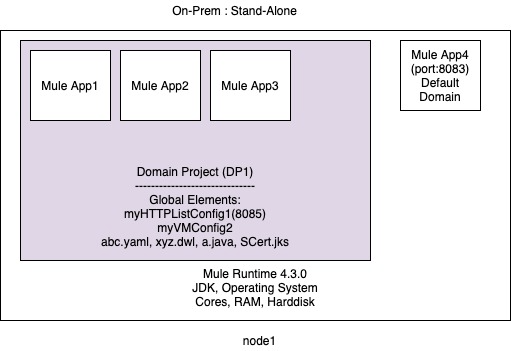
Upload the Connector’s JAR

* + Pros
    - Simplifies Development
    - Shields the complexity by displaying properties for Query Params, URI Params, Headers, Security, etc
    - Easy Version upgrades
    - Automatically picks Resource, HTTP Method, Media-types for Body
    - Supports Data Sense
    - Fail Fast
    - Protocol Property Supports Expression
  + Cons
    - Does not support HTTPS
    - Can NOT control lots of properties like Reconnection Strategy, TLS Config, Response Validator, Cookies, etc
* HTTP Request
  + Pros
    - Can control lots of properties like Reconnection Strategy, TLS Config, Response Validator, Cookies, etc
  + Cons
    - Manually need to pass URI Paramas, Query Params, Headers, HTTP Methods, endpoints, media-type, etc
    - No Fail Fast
    - No Data Sense
    - Protocol Property Does not Support Expression



**Domain Projects**

* Supported only for **On-Prem** Customer-Hosted Runtime Plane
  + Not for CloudHub, RTF
* Used for Sharing Global Elements (Connector Configurations, DWL, Java, Properties, Certificates, etc) with multiple Mule Applications belonging to the same Domain Project
* Can create many Domain Projects
* A single Mule Application can belong to only one Domain Project
* Cannot create Flows in Domain Project
* To Deploy Domain Project
  + Copy .jar file into $MULE\_HOME/domains folder

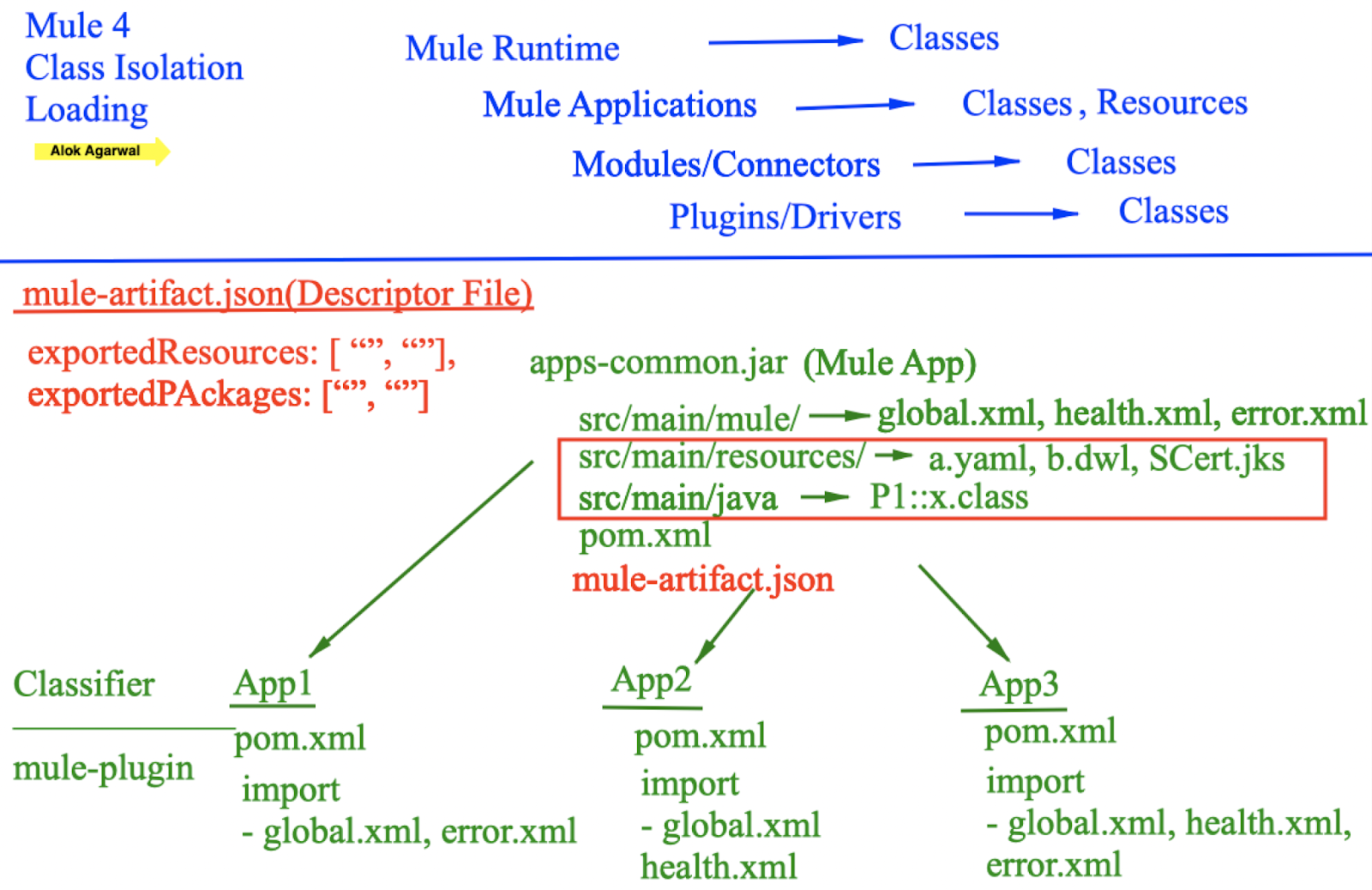


**Mule 3 Vs Mule 4**

* Event Structure
  + Attributes
  + Payload and Attachments
  + Variables
* Mule 3 supports Processing Strategy at Flow Level
* Mule 3 uses SEDA Queues
* To Create Custom Connectors
  + Mule 3 -> DevKit
  + Mule 4 -> Mule SDK
* Mule 3 Runtime Version and Connector Versions are tightly coupled
* Mule 3 have Polling Scope and Mule 4 have Scheduler Component
* Mule 3 uses Dataweave for Transformation and MEL for Expressions, but Mule 4 uses Dataweave for both Transformations and expressions
* Mule 3 Class loading was hierarchical and in Mule 4 Class loading is Isolated

**Class Isolation Loading**

* <https://docs.mulesoft.com/mule-runtime/4.3/about-classloading-isolation>



**Triggering Flow automatically without an external Event**

* Scheduler
* File/FTP/SFTP > On New or Update File
* DB > On Table Row
* Based on Connector
  + Ex: Anypoint MQ, Salesforce, etc

**Routers**

* Choice
* Scatter-Gather
* First Successful
* Round-Robin
* Error Handling
* APIKit Router

**Scenario for Processing Larger-than-memory payloads**

* Mule App1 > Flow1 > HTTP Listener (POST) + SetPayload
* Deployed to 0.1 vCore [ 500 MB Memory + 8GB Storage ]
* Request1 -> Sending 2GB of POST Body
* Question
  + Will Request1 be processed by Mule? Yes/No
* Answer: YES
* How?
  + <https://docs.mulesoft.com/runtime-manager/cloudhub-architecture#cloudhub-workers>
  + File Store Streaming(Default)
    - Stores Incoming Data in [Memory(0.5MB) + Storage(2GB-0.5MB)]

**Memory Management**

* TPS
  + Minimum of
    - Based on Memory
      * Total Available Memory / Memory used Per Request
    - Based on Storage
      * Total Available Storage / Storage used Per Request
* Scenario 1
  + Mule App1 > Flow1 > HTTP Listener (POST) + SetPayload

[512KB In-Memory]

* Deployed to 0.1 vCore [ 500 MB Memory(478MB Available) + 8GB Storage(4GB Available) ]
* Average Payload Size: 2GB [In-Memory(0.5MB) + Storage (2047.5MB)]
* TPS
  + Minimum of
    - Based on Memory
      * Total Available Memory / Memory used Per Request
      * 478 MB / 0.5MB = 956 TPS
    - Based on Storage
      * Total Available Storage / Storage used Per Request
      * 4096 MB / 2047.5 = 2 TPS
* Scenario 2A
  + Mule App1 > Flow1 > HTTP Listener (POST) + SetPayload

[512KB In-Memory]

* Deployed to 0.1 vCore [ 500 MB Memory(478MB Available) + 8GB Storage(4GB Available) ]
* Average Payload Size: 10MB [In-Memory(0.5MB) + Storage (9.5MB)]
* TPS
  + Minimum of
    - Based on Memory
      * Total Available Memory / Memory used Per Request
      * 478 MB / 0.5MB = 956 TPS
    - Based on Storage
      * Total Available Storage / Storage used Per Request
      * 4096 MB / 9.5MB = 431 TPS
* Scenario 2B
  + Mule App1 > Flow1 > HTTP Listener (POST) + SetPayload

[1MB In-Memory]

* Deployed to 0.1 vCore [ 500 MB Memory(478MB Available) + 8GB Storage(4GB Available) ]
* Average Payload Size: 10MB [In-Memory(1 MB) + Storage (9 MB)]
* TPS
  + Minimum of
    - Based on Memory
      * Total Available Memory / Memory used Per Request
      * 478 MB / 1MB = 478 TPS
    - Based on Storage
      * Total Available Storage / Storage used Per Request
      * 4096 MB / 9MB = 455 TPS
* Scenario 3
  + Mule App1 > Flow1 > HTTP Listener (POST) + File Read + SetPayload

[512 KB In-Memory] [ 1MB Memory]

* Deployed to 0.1 vCore [ 500 MB Memory(478MB Available) + 8GB Storage(4GB Available) ]
* Average Payload Size: 50MB [In-Memory(1.5 MB) + Storage (48.5 MB)]
* TPS
  + Minimum of
    - Based on Memory
      * Total Available Memory / Memory used Per Request
      * 478 MB / 1.5MB = 318 TPS
    - Based on Storage
      * Total Available Storage / Storage used Per Request
      * 4096 MB / 48.5MB = 84 TPS

**Thread Management**

* Reactive Engine
  + <https://www.mulesoft.com/lp/whitepaper/api/reactive-programming>
  + Design Patterns
    - Iterator
      * The Iterator pattern gives the consumer the power and control over when to consume data.
    - Observer
      * it gives the publisher power over when to push data
    - Functional Programming, Declarative like SQL
      * Dataweave
* Blocking Vs Non-Blocking Threads
  + Blocking Threads
    - 1 Thread = 1 Request
    - Mule supports in Transactions
    - Single Request will be served by Single thread
  + Non-Blocking Threads (NIO)
    - 100 Thread = 100+ Requests
    - Default in Mule
    - Single Request will be served by multiple threads
* Thread Types
  + Mule Threads
    - CPU\_LITE
      * Take < 10ms
    - CPU\_INTENSIVE
      * Takes > 10ms
    - IO (BLOCKING\_IO)
      * Used by Transactions, Batch Jobs
      * Uses Proactor Patterns
        + <https://en.wikipedia.org/wiki/Proactor_pattern>
      * Example:
        + DB Select

(BLOCKING\_IO+Completion Handler) + DB\_THREAD + (Completion Handler+BLOCKING\_IO)

* + Connector Based Threads
    - Ex: HTTP, DB, JMS, etc
* Each Mule component uses a specific type of Mule Thread
* Mule SDK
  + Need to specify which Mule Thread the custom connector uses
    - <https://docs.mulesoft.com/mule-sdk/1.1/non-blocking-operations#specify_execution_type>
    - <https://docs.mulesoft.com/mule-sdk/1.1/non-blocking-operations#execution-types>
    - Optional
  + If Execution Type is not specified, Mule will try to infer/detect the Thread Type
    - Uses BLOCKING\_IO, if Thread.sleep(), lock.lock() is used
    - Else, CPU\_LITE will be used always
    - CPU\_INTENSIVE is never inferred/detected
    - <https://docs.mulesoft.com/mule-sdk/1.1/non-blocking-operations#inferring-execution-types-automatically>
* Thread Strategy
  + DEDICATED
    - Upto Mule Version 4.2
    - Thread counts are fixed for CPU\_LITE, CPU\_INTENSIVE, BLOCKING\_IO
    - Example
      * 2 CORE CPU and 1GB Memory
        + CPU\_LITE=4
        + CPU\_INTENSIVE=4

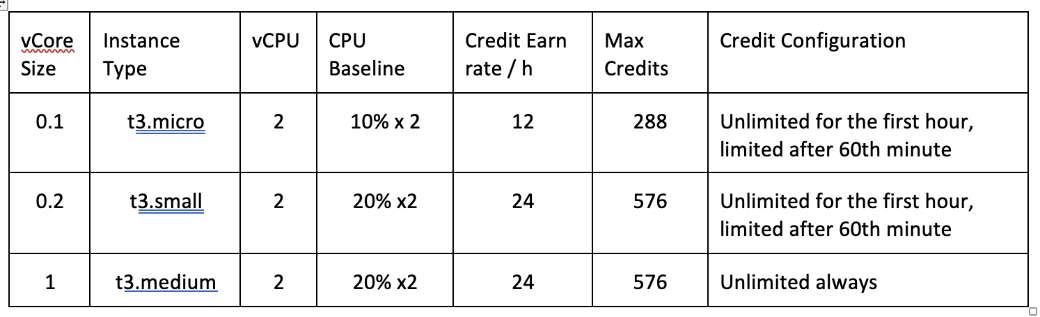
Work Queue= 4

* + - * + BLOCKING\_IO=151
  + UBERPOOL
    - Added from Mule Runtime 4.3+
    - Default from 4.3+
    - Thread counts are flexible/dynamic for CPU\_LITE, CPU\_INTENSIVE, BLOCKING\_IO
    - Example
      * 2 CORE CPU and 1GB Memory
        + UBERPOOL = 160  [threadKeepAlive=30000]

CPU\_LITE

CPU\_INTENSIVE

BLOCKING\_IO

* Thread Strategy and Thread count can be changed
  + Server Level
    - On-Prem
      * $MULE\_HOME/conf/scheduler-pools.conf
      * <https://docs.mulesoft.com/mule-runtime/4.3/execution-engine#configuration>
  + Application Level (On-Prem/CloudHub/RTF)
    - Global Elements > Scheduler Pools configuration
    - <https://docs.mulesoft.com/mule-runtime/4.3/execution-engine#configuration-at-the-application-level>
  + Considerations
    - <https://docs.mulesoft.com/mule-runtime/4.3/execution-engine#considerations-2>
* CloudHub Workers
  + For 0.1, 0.2, 1 vCore it internally uses 2 Cores in Calculation of threads
  + 0.1 vCore can use 10% of 2 CPU Cores
  + 0.2 vCore can use 20% of 2 CPU Cores
  + 1 vCore can use 20% of 2 CPU Cores
  + 

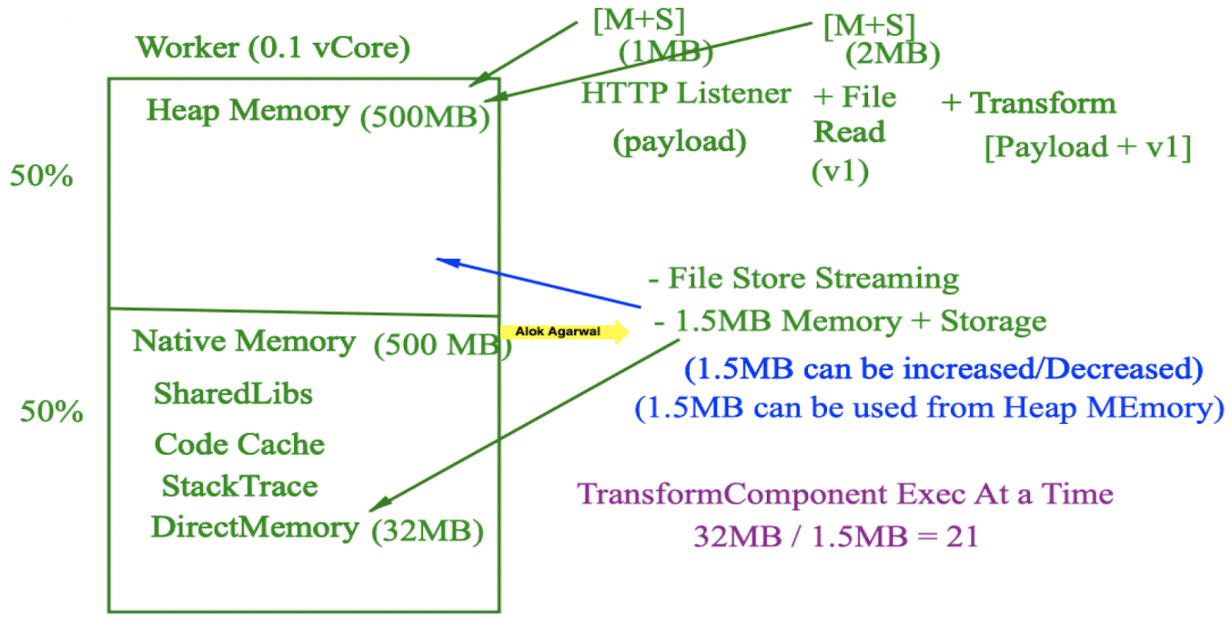
* Back Pressure
  + Automatic
    - For an incoming request Mule Runtime will check for the availability of Threads, Memory, Storage and rejects the request if short of resources with 503-Server Busy Error
  + Manual
    - Flow > MaxConcurrency
    - Number of Requests per Flow concurrently supported
    - Scenario
      * Mule App1 > Flow1 > MaxConcurrency=1
      * Deployed to 2 Workers
      * Question
        + Is MaxConcurrency per Flow or Per Worker?
      * Answer
        + Per Worker

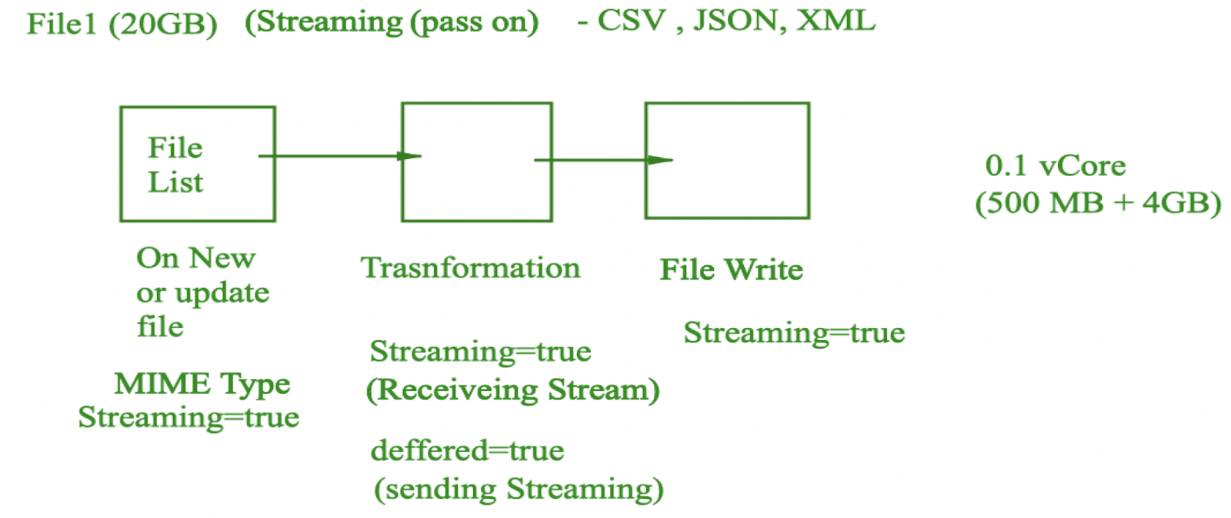
**Streaming**

* Types
  + Binary Streaming
    - Ex: HTTP List, File Read
  + Object Streaming
    - Ex: DB, Salesforce
* Strategy
  + Repeatable File-Store Stream/Iterable
    - Data is Stored in Memory + Storage [Serialization/DeSerialization]
    - Default
    - Configurable Memory Storage details
    - Process larger-than-memory payloads
    - Reliable
    - Can tune for Performance
    - Effect performance because of Serialization/DeSerialization
    - Non-Consumable Payloads
  + Repeatable In-Memory Stream/Iterable
    - Data is Stored in Memory only
    - Configurable Memory Storage and upper limits details
    - Better Performance
    - Non-Consumable Payloads
  + Non Repeatable Stream
    - Data is Stored in Memory only
    - NO Configurable Memory Storage
    - Consumable Payloads
      * Payload can be used only once
      * Payload will become null after a single use
    - Better Performance
      * Stores data in memory
      * After single use, it deletes the data

**Transform Message Component**

* Uses File-Store Streaming internally during Transformation process
* For Transformation it uses Direct Memory + Storage
* <https://help.mulesoft.com/s/article/Application-Deployed-to-a-Fractional-vCore-Worker-in-CloudHub-Throws-java-lang-OutOfMemoryError-Direct-buffer-memory>
* Streaming in Data weave
  + <https://docs.mulesoft.com/mule-runtime/4.3/dataweave-streaming#stream_output>
  + To enable Input Streaming in Data weave
    - Streaming=true
  + To enable output Streaming in data  weave
    - deferred=true
* <https://docs.mulesoft.com/mule-runtime/4.3/dataweave-formats-xml>



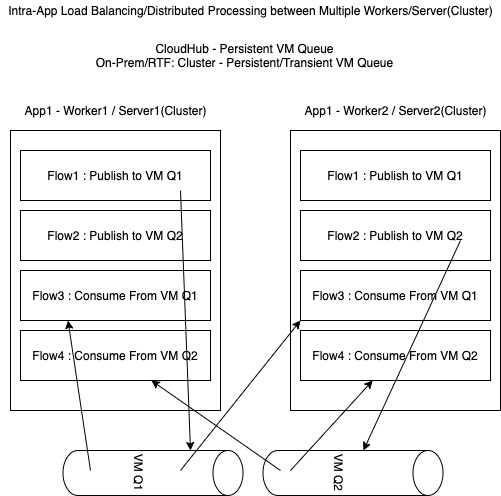


**Asynchronous Processing**

* Async Scope
* Queues
  + JMS
  + Anypoint MQ
  + Kafka
  + VM Queue

**VM Queues**

* Integral part of Mule
* Supports Persistent and Transient VM Queue
  + Default: Transient
* Pros
  + No Additional Infrastructure
  + Supports Inter-App Communication using Domain Projects in On-Prem Runtime Plane
  + Supports Intra-App Worker/Server(Cluster) Load Balancing/Distributed Processing

* 
* Cons
  + Supports only 1:1(Publish-Consume Model) only
    - Does not support 1:M (Publish-Subscribe)
  + No Manual Acknowledgment
  + No Fancy UI
  + Non-Mule Application cannot access VM Queues
  + VM queues are specific to a Mule Application
    - Exception: Domain Project(On-Prem)
* Persistent Vs Transient
  + Persistent
    - CloudHub
      * VM Queue is stored in AmazonSQS
      * VM Queue Messages are shared by same Mule App deployed to Multiple Workers
      * “Max Outstanding Messages” ignored
      * Developer’s configuration of Persistent/Transient VM Queue type is overridden by “Persistent Queue” property during Application deployment via Runtime Manager
      * VM Message Info can be viewed in Runtime Manager > Application > Queues
      * Limitations
        + Message Size <= 10MB
        + Retained for 4 Days only
    - Customer-Hosted (On-Prem/RTF)
      * Stand-Alone
        + Stores in Server’s Storage and Local
      * Cluster
        + Developer’s configuration of Persistent/Transient VM Queue type is overridden by Cluster’s Data Grid configuration of In-Memory
        + VM Queue Messages are shared by same Mule App deployed to Multiple Servers in the Cluster
  + Transient
    - CloudHub
      * VM Queue is stored in Worker’s Memory
      * VM Queue Messages are NOT shared by same Mule App deployed to Multiple Workers
      * Developer’s configuration of Persistent/Transient VM Queue type is overridden by “Persistent Queue” property during Application deployment via Runtime Manager
      * VM Message Info can NOT be viewed in Runtime Manager > Application > Queues
    - Customer-Hosted (On-Prem/RTF)
      * Stand-Alone
        + Stores in Server’s Memory and Local
      * Cluster
        + Developer’s configuration of Persistent/Transient VM Queue type is overridden by Cluster’s Data Grid configuration of In-Memory
        + VM Queue Messages are shared by same Mule App deployed to Multiple Servers in the Cluster

**Serialization**

* Mule uses Java Serialization and KRYO Serialization
* KRYO Serialization
  + By default, used only for File-Store Streaming and Batch Processing
  + Better Performance
    - <https://docs.mulesoft.com/mule-runtime/3.9/improving-performance-with-the-kryo-serializer#performance-using-kryo>
  + Configure KRYO as default for Mule Application
    - <https://docs.mulesoft.com/mule-runtime/3.9/improving-performance-with-the-kryo-serializer#configuring-the-kryo-serializer>
  + Configure Custom Serialization
    - <https://docs.mulesoft.com/mule-runtime/3.9/improving-performance-with-the-kryo-serializer#configuring-a-custom-serializer>
  + Considerations and Limitations
    - <https://docs.mulesoft.com/mule-runtime/3.9/improving-performance-with-the-kryo-serializer#limitations-and-considerations>

**Processing Records**

* Sequential Processing
  + For Each Scope
    - Batch Size: To Group records for Bulk Processing
      * Default: 1
  + Dataweave map()
* Parallel Processing
  + Parallel For Each
    - Needs XML changes for studio < 7.6 Version
    - <https://docs.mulesoft.com/mule-runtime/4.3/parallel-foreach-scope>
  + Batch Job
    - Is Asynchronous
    - Executed by BLOCKING\_IO Thread only
    - Batch jobs are never distributed
      * Needs to start and complete the entire Batch processing on the same Server/Worker
    - Always Internally uses Persistent VM Queues
      * Stand-Alone
        + VM Queue is created in Server’s Storage
      * Cluster
        + VM Queue is created in Server’s Storage
        + Not in Data Grid(In-Memory)
      * CloudHub
        + Depends on “Persistent Queues” Check box in Runtime Manager

If Checked, VM Queue is created in AmazonSQS

If Not Checked, VM Queue is created in Worker’s Storage

* + - Number of Threads used is minimum of
      * Number of Records/Block Size
      * Max Concurrency
      * 2 \* Cores
    - Variables inside Batch are Record Specific
    - Sections
      * Process Records
        + Batch Steps

Sections

Processor

Aggregators

Aggregator Size

Streaming

Both are Mutually exclusive and one is required

Accept Expression

Accept Policy

* + - * On Complete
        + Returns BatchJobResults Object
    - Max Failed Records
      * 0 - Stop on Error
      * N - Stop on (N+1) Error
      * -1 - Don’t Stop on Error
    - Batch Block Size
      * Number of records processed by a single Thread
      * Number of Threads = Number of Records / Batch Block Size
        + Example: 1000 Records/ 100 = 10 Threads
* Bulk Processing
  + Depends on connectors
    - Ex: DB, Salesforce, etc

**For-Each Vs Parallel For-Each**

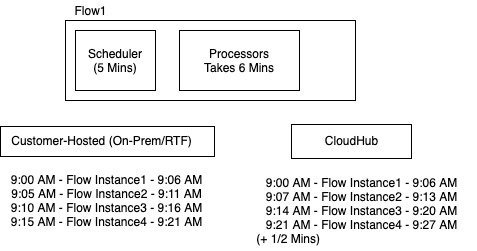
* Only One Thread executes For Each
  + Multiple Threads are executed for Parallel For Each
* Variables used inside For Each are available after ForEach
  + Not in the case of Parallel For Each
* Payload after For Each is original Payload
  + In Parallel For Each payload is aggregated in the same sequence as it was before
* By Default on Error For Each scope stops
  + In Parallel For Each, it throws MULE:COMPOSITE Error

**Parallel For-Each Vs Batch**

* Parallel For-Each is Synchronous and Batch is Asynchronous
* Predefined sections in Batch for Handling Errors, Processing, Aggregation, On Complete
* Can customize Number of Threads based on incoming Records and Batch Block Size
* Batch does not give any output payload. Parallel For Each provides the output of Processed Records in the same order
* Batch uses Persistent VM Queues to store the records to be processed. Parallel For Each does not.

**Scheduler Component**

* Behaves differently in Customer-Hosted and CloudHub Runtime Planes
* Scheduler runs only on Primary Node in a Cluster



**Transformations**

* Dataweave
  + Java!Package::Classname::method()
* Java
* XML > XSLT
* Scripting Engines
  + Groovy
  + JRuby
  + Jython
  + JavaScript

**Validations:**

* Dataweave
  + Can Use Validation::isEmail()
* XML Validation
* JSON Validation
* Java Validation
* APIKit Validation
* HTTP Response Validation
* Validation Module
  + Can be used as Dataweave Module also

**Sequential Processing**

* For Each Scope
* Dataweave map()
* Flow > MaxConcurrency=1

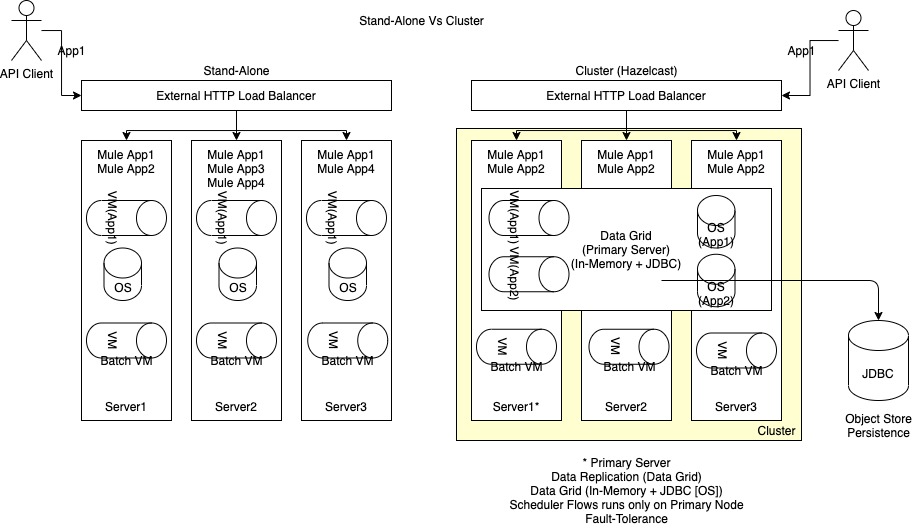
**Parallel Processing**

* Batch Job
* Parallel For-Each
* Scatter-Gather
* Async Scope
* Queues

**CDM (Common Data Model / Canonical Data Model)**

* Union of all Fields from multiple Integrated Apps
* Fields are typically FlexFields / AnyType and optional
* Simplifies/Reduces Transformations
  + Requires only 2 \* N Transformations
    - N=Number of Apps Integrated
* Complex CDM Building process
  + Need Data Architects
* Reusability
* Provides Common Language

**Stand-Alone Vs Cluster**



**Testing Strategies**

* White-Box / Unit Testing
  + MUnit
    - Test suites
      * Test Cases
        + Created by

Automatically

Manually

Recording (Studio 7.5 +)

Record Test for this Flow

Automatically creates a test case using normal testing scenarios using the Recording feature

* + - * + Sections

Behavior

Spy

Mocking

Set Events, set Payloads

Execution

Set Events, set Payloads

Flow Reference

HTTP Requests

Validation

Assertion

Verify Call

Before Test/After Test/ Before Suite/ After Suite

* + - * + Run Test

Manually in Studio

Run Test Case

Run Test Suite

Maven MUnit Plugin

* + - * + MUnit and MUnit Tools Modules
        + Coverage Reports
        + Tags
        + MUnit Examples

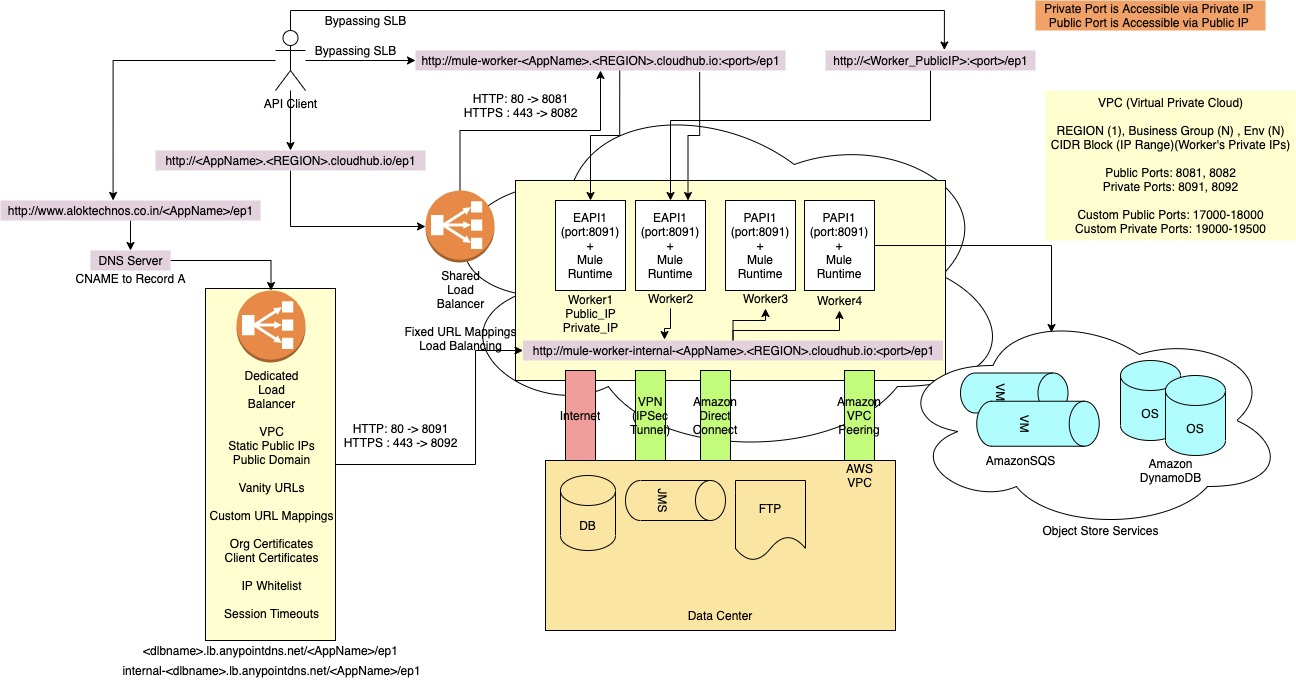
<https://docs.mulesoft.com/munit/latest/munit-cookbook>

* + - * + To display the Test Coverage Report using **maven** make the following changes in pom.xml

<https://docs.mulesoft.com/munit/2.2/munit-maven-plugin#specify-runtime-product>

* Functional / Integration / Black-box  Testing
  + User Interface
    - Control Plane UI > Monitoring > Functional Monitoring
  + CLI
    - BAT CLI (Blackbox Automated Test)
      * <https://docs.mulesoft.com/api-functional-monitoring/bat-install-task>
      * Based on BDD
      * Uses DWL (Dataweave)
      * Functional Monitoring internally uses BAT CLI only
* Performance / Load Testing
  + Monitoring, Visualizer , Analytics, Dashboards
* Other Testings - External Tools
  + Regression
  + UAT
  + Security
  + Regression
  + Resilience
  + Destructive

**CloudHub Architecture**



* SLB performs Round-Robin Load Balancing and maps
  + http://<appname>.<REGION>.cloudhub.io to http://mule-worker-<appname>.<REGION>.cloudhub.io:8081
  + https://<appname>.<REGION>.cloudhub.io to https://mule-worker-<appname>.<REGION>.cloudhub.io:8082
* To use ports other than 8081/8082, use VPC
* mule-worker-<appname>.<REGION>.cloudhub.io:<port> can be used from the public internet  to access Application deployed to any public ports including 8081/8082
  + API Consumer/Client should do load balancing
* <worker’s\_IPAddress>:<port> can be used from the public internet  to access Application deployed to any public ports including 8081/8082
  + API Consumer/Client should do load balancing
* Applications running on private ports within the VPC cannot be accessed via the public internet. Can be accessed only within the VPC
* Applications running on a private port within the VPC can be accessed using mule-worker-internal-<appname>.<REGION>.cloudhub.io:<private\_port>
  + HTTP Request and Mule REST Connectors perform Load Balancing

**Shared Load Balancer Vs Dedicated Load Balancer:**

* Common
  + Round-Robin Load Balancing
  + .jar <= 200MB
  + Fixed Port Mappins
    - SLB : HTTP - 80 -> 8081 , HTTPS - 443 -> 8082
    - DLB : HTTP - 80 -> 8091 , HTTPS - 443 -> 8092
* Shared Load Balancer (SLB)
  + Shared by Multiple Organizations
  + Performance is slow
  + Maps HTTP:80 and HTTPS:443 to HTTP:8081 and HTTPS:8082 respectively
  + Cannot provide Organization specific Certificates and uses a common Mulesoft CloudHub’s Certificate
  + Uses Shared VPC
  + Fixed URL - appname.REGION.cloudhub.io
* Dedicated Load Balancer (DLB)
  + Dedicated to your Organization
  + Performance is better
  + Write your own mapping rules
  + Can provide Organization and Client specific Certificates
  + Add Whitelisted CIDRs
  + Uses Org specific VPC
  + Can specify Org specific Timeouts
  + Vanity URLs
  + Public Domain: <dlbname>.lb.anypointdns.net
  + Private/Internal Domain: internal-<dlbname>.lb.anypointdns.net

**State Management**

* Pre-Built
  + Object Store
  + VM Queues
* External
  + DB
  + File/FTP
  + External Services

**Object Store**

* Key/Value pair Storage
* Support Persistent and Transient(Non-Persistent) Object Store
  + Default: Persistent
* Integral part of Mule
* Specific to a Mule Application
  + Exception: Domain Project, REST APIs
* Use Cases
  + Watermark
  + OAuth Tokens
  + Idempotency
  + Caching
  + Totals
  + Counters
* Object Store Types
  + CloudHub
    - OSv1
      * Available only in US Region
      * 100,000 Keys Per App
      * Value <= 1MB Per Key
      * Value <= 1GB Per App
    - OSv2
      * Available in the same region as worker
      * No Key Limit
      * Value <= 10MB Per Key (After Base 64 Encoding)
      * 10 TPS per App for Free License
        + For Additional license you can get 100 TPS per App
    - Depends on “Use Object store v2” checkbox During Application Deployment using Runtime Manager
  + Customer-Hosted (On-Prem/RTF)
    - Mule Object Store
* Connect to Object Store
  + Object Store Connector
    - Connect to OSv1, OSv2, Mule Object Store
  + Object Store REST APIs
    - Connect to only OSv2
* Persistent Vs Transient
  + Persistent
    - CloudHub
      * OS is stored in Amazon DynamoDB
      * OS Data are shared by same Mule App deployed to Multiple Workers
      * “Max Entries” is ignored
      * OS Info can be viewed in Runtime Manager > Application > Object Store
    - Customer-Hosted
      * On-Prem
        + Stand-Alone

Stores in Server’s Storage and Local

* + - * + Cluster

Developer’s configuration of Persistent/Transient OS type is overridden by Cluster’s Data Grid configuration of In-Memory/JDBC

OS Data are shared by same Mule App deployed to Multiple Servers in the Cluster

* + - * RTF
        + No Persistence Gateway

Non-Clustered

Pod’s Disk and Local

Behaves as Stand-Alone

Clustered(Hazelcast)

Stored in Data Grid(In-Memory)

* + - * + Persistence Gateway

Can configure PostgreSQL to store Object store data Permanently

OS Data is permanently stores in PostgresQL

* + Transient
    - CloudHub
      * OS is stored in Worker’s Memory
      * OS Data are NOT shared by same Mule App deployed to Multiple Workers
      * OS Info can NOT be viewed in Runtime Manager > Application > Object Store
    - Customer-Hosted
      * On-Prem
        + Stand-Alone

Stores in Server’s Memory and Local

* + - * + Cluster

Developer’s configuration of Persistent/Transient OS type is overridden by Cluster’s Data Grid configuration of In-Memory/JDBC

OS Data are shared by same Mule App deployed to Multiple Servers in the Cluster

* + - * RTF
        + No Persistence Gateway

Non-Clustered

Pod’s Memory and Local

Behaves as Stand-Alone

Clustered(Hazelcast)

Stored in Data Grid(In-Memory)

* + - * + Persistence Gateway

Can configure PostgreSQL to store Object store data Permanently

OS Data is permanently stores in PostgresQL

* Object Store Connector Configuration Types
  + Default Object Store
    - Type: Persistent
    - TTL: 30 Days
    - Max Entries: No Limit
  + Global Object Store Configuration
    - Partition Name
    - Type: Persistent/Transient
    - Max Entries
    - Entry TTL
    - Entry TTL Unit
    - Expiration Interval
      * To execute Expiry Process (Cleaning Algorithm)
    - Expiration Interval Unit
  + Private/Inline Object Store Configurations
* Object Store REST APIs
  + Only for OSv2
  + Sharing of Object Store Data is possible between Mule Applications
  + Non-Mule Applications can also access OSv2
  + APIs
    - Object Store v2
    - Object Store v2 stats
  + To access these APIs, client need to pass UserName/Password, BEARER Token, Org ID, Env ID, App’s Object Store Specific Client\_id and Client\_secret

**Store Profiles**

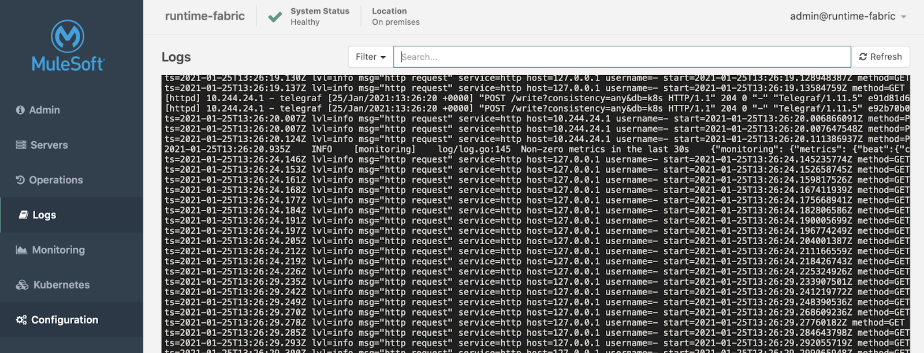
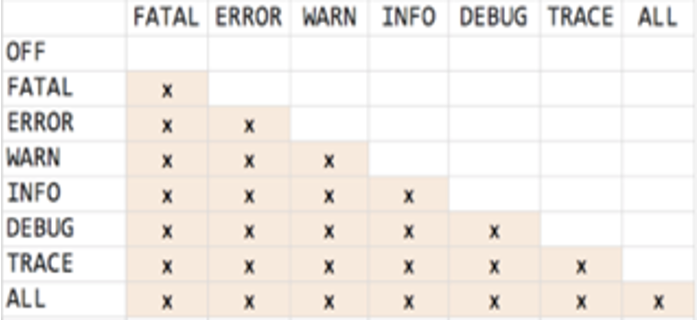
* Are applicable only to Customer-Hosted Clusters[On-Prem/RTF]
* Store profile overrides Cluster Properties/Rules
* Value
  + Reliable
    - Default
    - Follows Cluster Properties/Rules
  + Performant
    - Uses Local Queues, Local Object store
    - Does not use Data Grid and No data Replications
    - Does not follow Cluster Properties/Rules
* Setting Store Profile
  + Application
    - Global Elements > Global Configuration > Configuration > HA Profile
    - Ex:
      * <cluster:cluster-config >

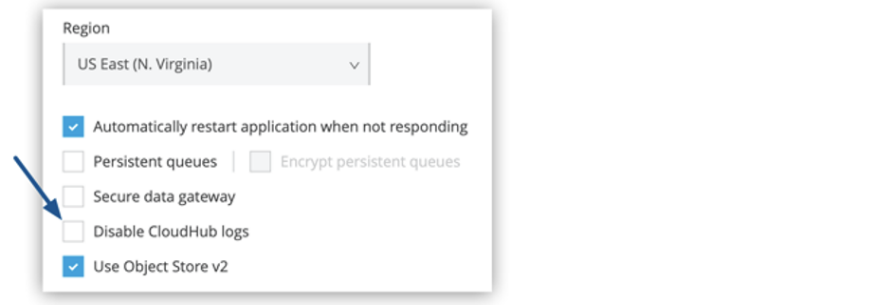
<cluster:performance-store-profile />

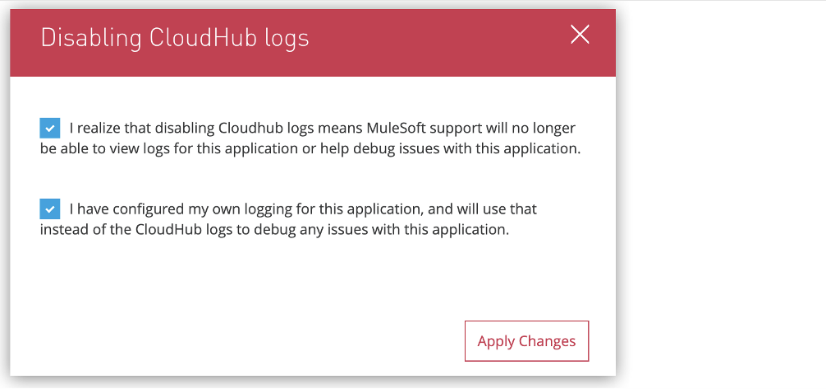
</cluster:cluster-config>

* Server
  + mule.cluster.storeprofile=performance
    - $MULE\_HOME/conf/wrapper.conf
    - $MULE\_HOME/.mule/mule-cluster.properties

**Logging**

* Audit Logs
  + Is used to identify Access Violations
  + Access Management > Audit Logs
  + Business Group Aware
  + Required Audit Log Viewer permission
  + Retained for 6 years
  + Download
    - User Interface
    - Audit Log Query REST API
* System Logs
  + CloudHub
    - Runtime Manager > Application > System Logs
    - Download
      * Mule Logs (User Interface)
      * CloudHub API
    - Retention is 30 Days or 100 MB
  + On-Prem
    - $MULE\_HOME/logs/mule\_ee.log
  + RTF
    - OpsCenter UI
    - 
* Application Logs
  + CloudHub
    - Runtime Manager > Application > Worker-N Logs
    - Download
      * Logs (User Interface)
      * CloudHub API
    - Retention is 30 Days or 100 MB
  + On-Prem
    - $MULE\_HOME/logs/<AppName>.log
  + RTF
    - OpsCenter UI
* Log Levels
  + Default: INFO
  + 
* Logging Tab
  + Enable DEBUG Levels dynamically
  + No Redeployments/ No Restarts
* Logger Component
  + Is Asynchronous
  + Bypassing streaming and loads complete(Used in logger) data in Memory
* Externalize Logs
  + Push
    - On-Prem
      * Runtime Manager > Server > Settings > Plugins
        + Configure SPLUNK and ELK
      * Add Appenders to Application’s log4j2.xml
    - CloudHub
      * Add Appenders to Application’s log4j2.xml
      * Application’s log4j2.xml is overridden by CloudHub log4j2.xml during deployment
      * There is a process to disable CloudHub log4j2.xml overriding
        + During Deployment check “Disable CloudHub logs” checkbox





* From now, CloudHub will not record any logs for this Application
* To store CloudHub logs along with Application’s log4j2.xml Appenders, Add CLOUDHUB LOG APPENDER to log4j2.xml
  + <https://docs.mulesoft.com/runtime-manager/custom-log-appender#create-your-log4j-configuration>
* Pull
  + On-Prem
    - Can install background process to pull logs
      * Ex: SPLUNK UNIVERSAL FORWARDER, etc
  + CloudHub
    - Using CloudHub REST APIs

**Monitoring**

* Available for All Runtime Planes
  + CloudHub
  + On-Prem
    - Need to perform some install(Monitoring Agent) and configuration to enable monitoring feature
    - Monitoring > Settings > Hybrid Tab
  + RTF
* 3-5% Performance hit
* Features
  + Built-In Dashboards
    - Titanium Features
      * Flows Tab
      * Connector Tab
      * Custom Metrics Tab
      * Application Network Tab
  + Custom Dashboards
    - Advanced Alerts
      * Titanium Feature
  + Reports
    - Titanium Feature
    - High-Level Text/Tabular Reports
    - Eagle-Eye view reports
  + Alerts
    - Basic
      * Monitoring
      * Runtime Manager
        + Applications

CloudHub / Hybrid

* + - * + Server/Server Groups/Cluster
      * API Manager
    - Advanced
      * Created in Custom Dashboards
      * Titanium Feature
  + Custom Metrics
    - Titanium Feature
  + Functional Monitoring
  + Log Management (Titanium Feature)
    - Log Search
      * Aggregated Logs from all Mule APIs
    - Log Points
      * Codeless logging
    - Raw Data

**Properties**

* Supports .yaml  and .properties file
* Types:
  + Configuration Properties
    - Clear Text Values
  + Secured Properties
    - <https://docs.mulesoft.com/mule-runtime/4.3/secure-configuration-properties>
    - Encrypted Values
      * Previous Release
        + java -jar secure-properties-tool.jar string encrypt Blowfish CBC secure12345 mule
      * Latest release: 04/17/2020
        + java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC secure12345 mule
    - Encrypted values must be placed in ! [ “” ]
    - Global Elements > Secure Configuration
    - Use using ${secure::<property>}
  + Hidden Properties / Masked Properties
    - Hiding the Property value in Runtime Manager> Properties tab
    - Displaying \*\*\*\*\*\*\*\* in Runtime Manager > Properties Tab
    - Properties need to be mentioned in mule-artifact.json
      * mule-artifact.json

{

  "minMuleVersion": "4.3.0",

  "secureProperties": ["encrypt.key","db.password"]

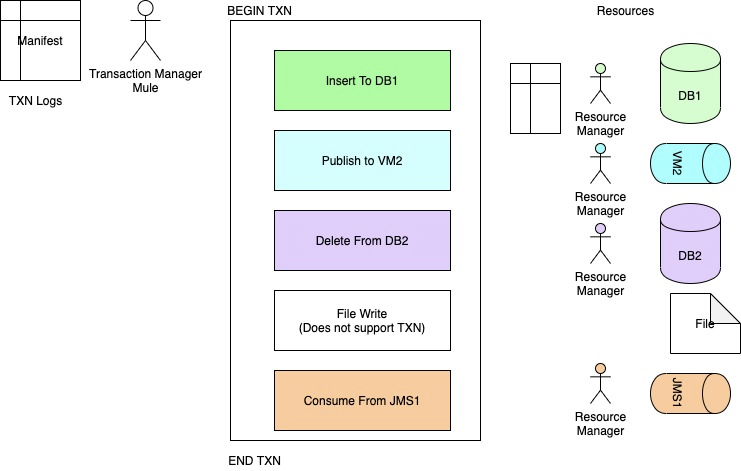
}

**Automation Options**

* Maven
  + Mule Plugin and MUnit plugin
  + Maven Phases/Goals
    - Compile
    - Test
    - Package
    - Install
    - Deploy
  + Executing later phases automatically trigger earlier phases
  + Deploying to All Runtime Planes using Maven
    - <https://docs.mulesoft.com/mule-runtime/4.3/deploying>
* Anypoint Platform CLI
  + <https://docs.mulesoft.com/runtime-manager/anypoint-platform-cli>
  + Connected using Username/Password
  + Internally uses Anypoint Platform REST APIs
* Anypoint Platform REST APIs
  + Connect using Username/Password, BEARER Token, Org ID, Env ID
  + List of APIs
    - <https://anypoint.mulesoft.com/exchange/portals/anypoint-platform/>
  + Article for accessing APIs
    - <https://help.mulesoft.com/s/article/How-to-get-Cloudhub-application-information-using-Anypoint-Platform-API?r=8&ui-force-components-controllers-recordGlobalValueProvider.RecordGvp.getRecord=1>
  + Anypoint Platform Control Plane UI internally uses Anypoint Platform REST APIs
  + If your Organization is using External Identity Provider then to access Anypoint Platform CLI and/or REST APIs, use the following info
    - <https://help.mulesoft.com/s/article/How-to-generate-your-Authorization-Bearer-token-for-Anypoint-Platform>
    - <https://docs.mulesoft.com/runtime-manager/anypoint-platform-cli#bearer-token>
    - <https://docs.mulesoft.com/access-management/saml-bearer-token>

**Transaction Management**

* By default Transactions are disabled
* Executed by BLOCKING\_IO Mule Threads
* Uses Blocking threads (Single thread Per Request)
* Supported Connectors and Operations
  + Database
    - All Operations
  + JMS
    - On New Message
    - Publish
    - Consume
  + VM
    - Listener
    - Publish
    - Consume
* 2PC / Global Transaction / XA Transaction



* Components which can BEGIN Transactions
  + Database
    - On Table Row
  + JMS
    - On New Message
  + VM
    - Listener
  + TRY Scope
* Transaction Action
  + None
    - Default
  + INDIFFERENT (Try Scope)
  + ALWAYS\_BEGIN
* Transaction Type
  + Local
    - Single Resource
    - Better Performance
    - Does not Support Nested Transaction
  + XA (eXtended Architecture)
    - Multiple Resources
    - Uses 2PC
    - Need a Transaction Manager
      * Enable Transaction Manager in Mule Application
        + Global Elements > Bitronix Transaction Manager
    - Supports Nested Transaction
* Bitronix Transaction Manager
  + <https://docs.mulesoft.com/mule-runtime/4.3/using-bitronix-to-manage-transactions>
* Commits if
  + Flow completed successfully
  + Try Block completed successfully
  + On Error Continue is used
* Rollback if
  + No Error Handling
  + On Error Propagate is Used
* SAGA Patterns
  + Managing Transaction Manually
  + Patterns
    - Event/Choreography pattern
    - Command/Orchestration pattern
* Behavior of Routers and Scopes in Transactions
  + <https://docs.mulesoft.com/mule-runtime/4.3/transaction-management#tx_scopes_routers>
  + Async Scope and Batch Jobs are **not** part of Transactions and execute outside of the transaction
  + Scatter-Gather and Parallel For-Each executed sequentially because of single BLOCKING\_IO thread

**Reliable Integration Solutions**

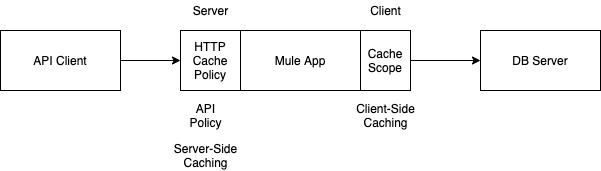
* Queues
  + VM
    - Persistent
  + External Queues
* State Management (Persistent)
  + Object Store
  + VM
  + External Services(DB, FTP, etc)
* Cluster
  + Supports Fault Tolerance
* Until Successful
  + Throws RETRY\_EXHAUSTED Error
  + Retry must be done for Idempotent Methods
    - Safe Methods (Cacheable)
      * GET, HEAD, OPTIONS
    - UnSafe Methods
      * PUT, DELETE
* Transaction Management
* Redelivery Policy
  + Throws REDELIVERY\_EXHAUSTED Error
* Reconnection Strategy
  + Throws RETRY\_EXHAUSTED Error
* Error Handling
* File Store Streaming
* Testing
* Reliability Pattern
  + Acquisition flow
  + Business/Logic/Transaction/Process Flow
  + Dispatcher Flow

**High Available Integration Solutions**

* Horizontal Scaling
  + CloudHub
    - Multiple Workers (Same Region)
  + On-Prem
    - Multiple Stand-Alone
    - Cluster
  + RTF
    - Multiple Pods (Replicas)
* Load Balancer
* Server Groups
  + Is a JUST a **Logical** group of Multiple Stand-Alone Servers for common Application deployments
  + Only for On-Prem
  + Server Groups are at Control Plane Level only
    - Not at Runtime Plane
* Cluster (Hazelcast)
  + Is a Physical Group of Servers
  + Only available for Customer-Hosted(On-Prem/RTF)
  + Data Grid for Data sharing
    - Data Grid is In-Memory(Default) + JDBC (Object Store Persistence)
  + Data Grid is replicated across all Servers
  + Active-Active
  + Nodes are aware of each other
  + Nodes can be added/removed based on Load Requirements
  + Quorum Management (Quorum size)
    - Limit the number of servers where Data replication occurs
      * Improves Performance
    - Minimum number of servers required in a cluster to be operational
  + Used for Fault Tolerance
  + Requires External HTTP Load Balancer to load balance HTTP Request
  + Creating Cluster
    - User Interface
    - Manual
      * <https://docs.mulesoft.com/mule-runtime/4.3/creating-and-managing-a-cluster-manually#creating-a-cluster-manually>
  + Primary Node
    - Auto Selected
    - Scheduler always runs only on Primary Node
    - “Primary Node Only” Property
      * Only works in Cluster
      * The listener will run only on Primary Node
      * For JMS Listener this checkbox is selected by default
        + Uncheck it to consume Messages parallely on all servers in a cluster

**Performant Integration Solutions**

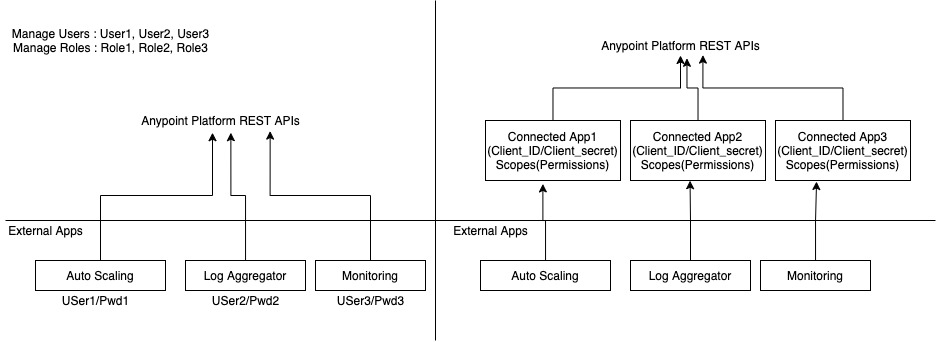
* Parallel Processing
  + Parallel For-Each
  + Batch
    - Batch Aggrgators
  + Scatter-Gather
* Asynchronous Processing
  + Async Scope
  + Queue
    - Transient VM Queue
* Scaling
  + Horizontal
  + Vertical
  + Auto Scaling
    - Available for CloudHub for ELA Customer Only
* Cluster
* Load Balancing
* Use less variable
  + Use Remove Variables
* Streaming
  + In-Memory
  + Non Repeatable Stream
* Pre-built State Management options
  + VM Queues
  + Object Store
* Proper Logging using INFO Level
* Round-Robin
  + Client side Load Balancing
* Bulk Operations
* Cache
  + Client Side Cache
    - Cache Scope
  + Server Side Cache
    - HTTP Cache Policy



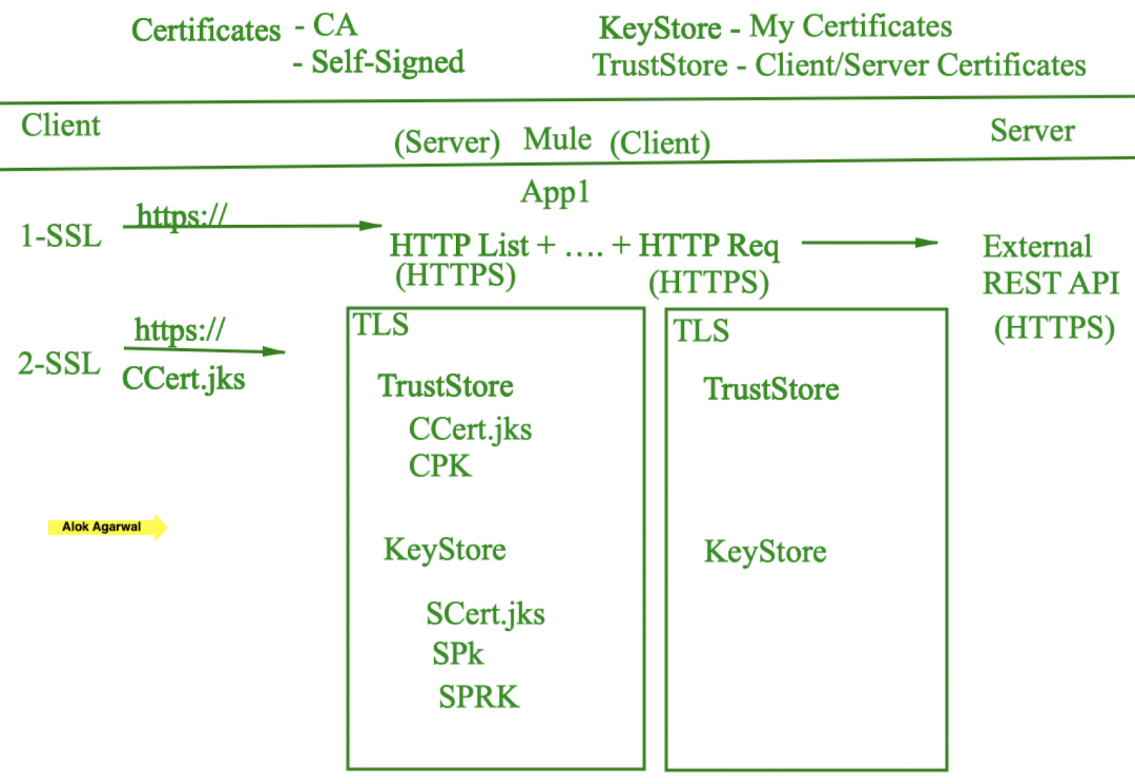
* Use Transform Message Component for Dataweave
  + Internally uses DirectMemory
* JMS Connector
  + Acknowledgement : DUPS\_OK, IMMEDIATE
  + Uncheck “Use Primary Node only”
  + Internally uses Cache for Producer and Consumer Objects
    - Default
  + Number of Consumers
  + Avoid Selector during Receiving Messages
* Redelivery Policy
* Validations
* Dedicated Load Balancer
* HTTP Vs HTTPS
  + In VPC, Use Private Ports over HTTP
* Connector’s Connection Pools
  + Ex: DB Connection Pool
* Compression Module
  + Support ZIP, GZIP
* TLS v1.2
* Use Latest Version of Components, Connectors
* Flow > MaxConcurrency=1
  + Can use More Memory to store the data, hence improve performance
* On-Prem
  + can do Thread Tuning using scheduler-pools.conf
  + Can do JVM Tuning using wrapper.conf
* KRYO Serialization

**Secure Integration Solutions**

* Access Management
  + Users
    - Manage Internal and External Users
  + Teams
    - <https://docs.mulesoft.com/access-management/teams>
  + Business Groups
    - Master Organization
      * Child business Groups
      * Environments
      * Roles
  + Multi-Factor Authentication
    - <https://docs.mulesoft.com/access-management/multi-factor-authentication>
  + Identity Provider
    - Default: Anypoint Platform
    - Authenticate Username/PAssword
    - Used in the following scenarios
      * Anypoint Platform Control Plane UI
      * Anypoint Platform REST APIs
      * Anypoint Platform CLI
      * Maven
      * Anypoint Studio
    - Can integrate **only one** External IDP for Authentication at **Master Org** Level
    - Options to connect IDP
      * OpenId Connect
      * SAML 2.0
    - <https://docs.mulesoft.com/access-management/external-identity>
  + Client Providers
    - Used to Authorize API Client requests for APIs (Experience API, Process API, System APIs)
    - Used for OAuth 2.0 token validations
    - Multiple Client Providers can be registered
    - Connect using
      * OpenAM
      * PingFederate
      * OpenID connect DCR
    - <https://docs.mulesoft.com/access-management/managing-api-clients>
  + Audit Logs
  + Connected Apps
    - <https://docs.mulesoft.com/access-management/connected-apps-overview>



* External Access
  + Used to share One Master Organization’s Private Asset with other Master Organization in a private way
  + <https://docs.mulesoft.com/access-management/external-organization-access>
* API Manager
  + Proxy Apps
  + Policies
    - Prebuilt
    - Custom
* Mule Application
  + Properties
    - Secure
    - Hidden/Masked
  + SSL
    - TLS
  + Crypto Module
    - Dw::crypto
  + Validations
    - Schema and Value Validation
  + Standardize Error Handling
  + Periodic Scans and Audit on the Codes
  + Do not expose too much of Data in
    - API Console Flow
    - Heart Beat Flow
    - Error Handling
    - Logging
  + Use Approved Libraries and Java code



* Secret Manager
  + Used by RTF, API Manager, Partner Manager
  + Is a Secret Vault
  + Features
    - TrustStore
    - KeyStore
    - TLS Context
* Anypoint Security
  + It is used only by RTF
  + Edge-Level / Perimeter-Level Security
  + Features
    - Security Policies (First Line of Defense)
      * Denial Of Service
      * HTTP Limits
      * IP Allowlist
      * WAF (Web Application firewall)
    - Tokenization
      * Policies
        + Tokenization
        + Detokenization
      * Tokenization Service
* VPC
  + Create a VPC
    - Name
    - Region
      * Only One
    - CIDR Block
      * Example
        + 10.0.0.0/16 (MAX)

2 Power (32-16) = 65536 IP Address

* + - * + 10.0.0.0/24 (MIN)

2 Power (32-24) = 256 IP Address

* + - * Max: /16
      * Min: /24
      * Used for Worker’s Private IPs
      * Mulesoft does not charge for higher blocks
      * Internally (For networking/Infrastructure) upto 6 IPs are reserved
    - Environments
    - Business Groups
    - Firewall Rules
  + Sizing VPC
    - Scenario
      * 30 Mule Application
      * 4 Environments
      * 3 Workers Each
      * Zero-Downtime Redeployments
      * Future-Ready
    - Question
      * How many IP Address are required
    - Calculation
      * 30 x 4 x 3 = 360 IPs
      * 360 x 2 = 720 IPs
    - Options
      * A) 360 IPs
      * B) 1024 IPs
      * C) 2048 IPs
      * D) 256 IPs
* DLB
* VPN

**Important Topics for Certification**