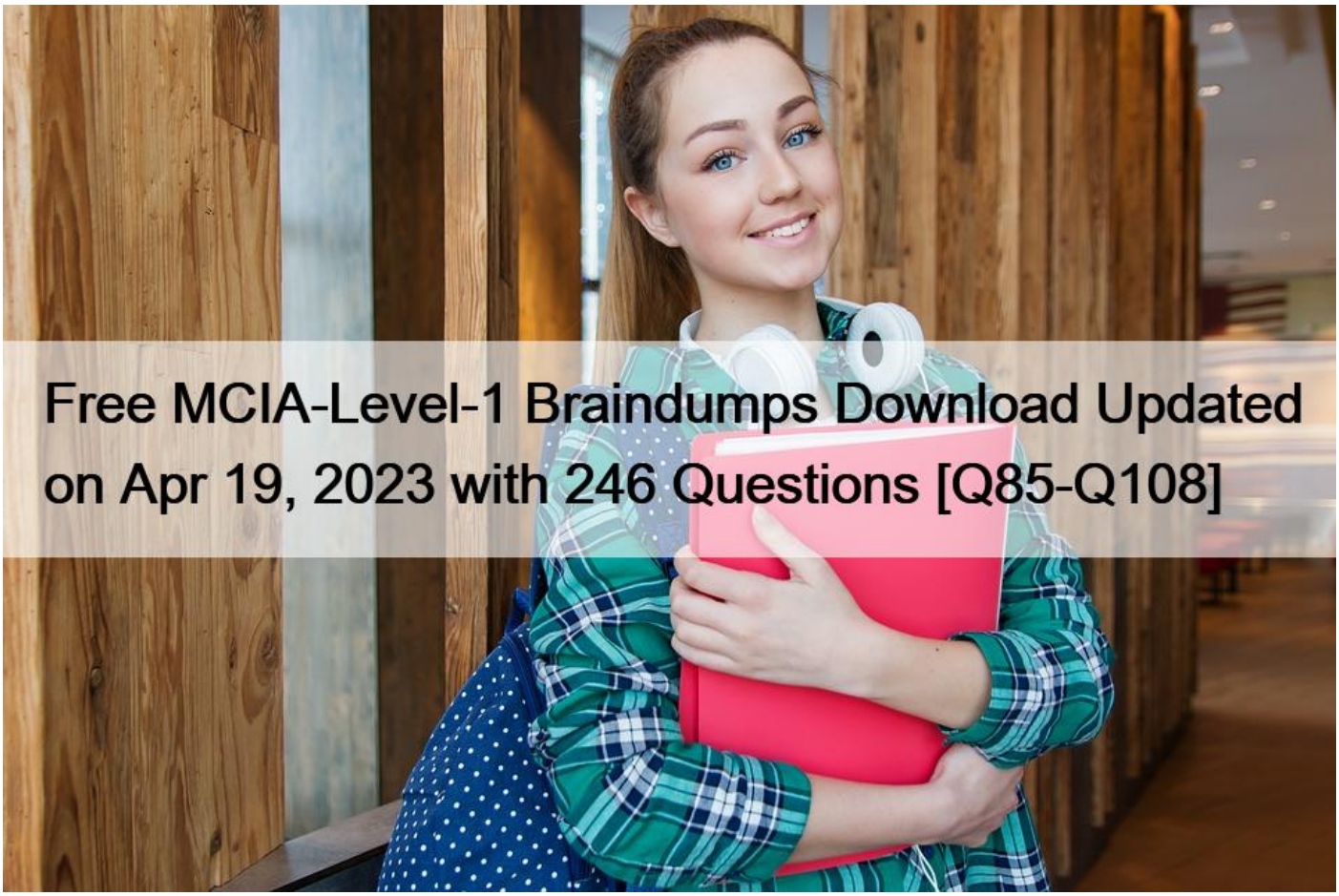
Free MCIA-Level-1 Braindumps Download Updated on Apr 19, 2023 with 246 Questions [Q85-Q108

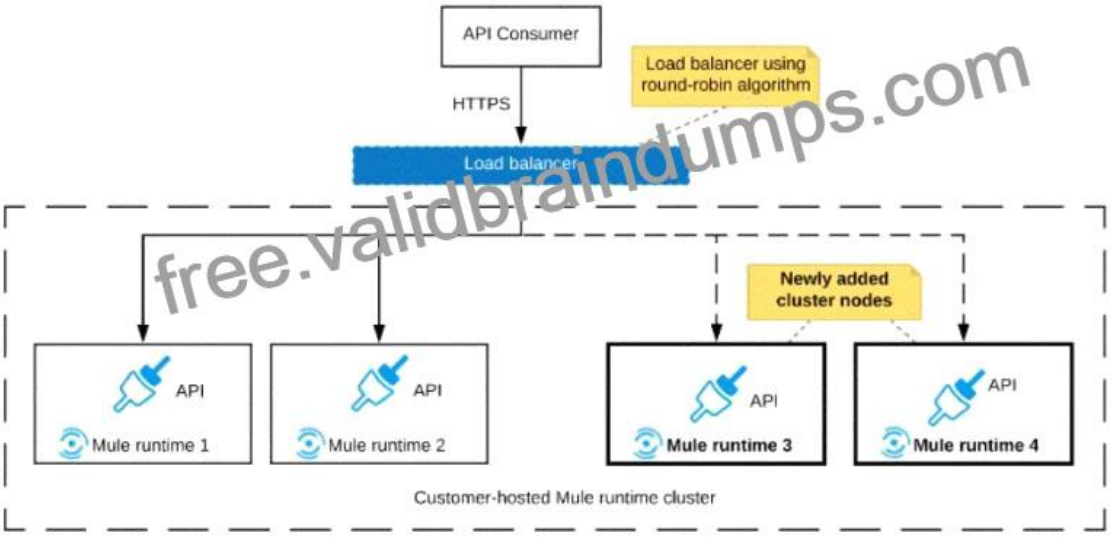


Free MCIA-Level-1 Braindumps Download Updated on Apr 19, 2023 with 246 Questions
MuleSoft MCIA-Level-1 Exam Practice Test Questions

Q85. As a part of project requirement, Java Invoke static connector in a mule 4 application needs to invoke a static method in a dependency jar file. What are two ways to add the dependency to be visible by the connectors class loader?

(Choose two answers)
* In the Java Invoke static connector configuration, configure a path and name of the dependency jar file
* Add the dependency jar file to the java classpath by setting the JVM parameters
* Use Maven command to include the dependency jar file when packaging the application
* Configure the dependency as a shared library in the project POM
* Update mule-artefact.json to export the Java package

Q86. Refer to the exhibit.



An organization uses a 2-node Mute runtime cluster to host one stateless API implementation. The API is accessed over HTTPS through a load balancer that uses round-robin for load distribution.

Two additional nodes have been added to the cluster and the load balancer has been configured to recognize the new nodes with no other change to the load balancer.

What average performance change is guaranteed to happen, assuming all cluster nodes are fully operational?
* 50% reduction in the response time of the API
* 100% increase in the throughput of the API
* 50% reduction In the JVM heap memory consumed by each node
* 50% reduction In the number of requests being received by each node

Q87. An external REST client periodically sends an array of records in a single POST request to a Mule application API endpoint.

The Mule application must validate each record of the request against a JSON schema before sending it to a downstream system in the same order that it was received in the array Record processing will take place inside a router or scope that calls a child flow. The child flow has its own error handling defined. Any validation or communication failures should not prevent further processing of the remaining records.

To best address these requirements what is the most idiomatic(used for it intended purpose) router or scope to used in the parent flow, and what type of error handler should be used in the child flow?
* First Successful router in the parent flow

On Error Continue error handler in the child flow
* For Each scope in the parent flow

On Error Continue error handler in the child flow
* Parallel For Each scope in the parent flow

On Error Propagate error handler in the child flow
* Until Successful router in the parent flow

On Error Propagate error handler in the child flow

Q88. What is required before an API implemented using the components of Anypoint Platform can be managed and governed (by applying API policies) on Anypoint Platform?
* The API must be published to Anypoint Exchange and a corresponding API instance ID must be obtained from API Manager to be used in the API implementation
* The API implementation source code must be committed to a source control management system (such as GitHub)
* A RAML definition of the API must be created in API designer so it can then be published to Anypoint Exchange
* The API must be shared with the potential developers through an API portal so API consumers can interact with the API
Context of the question is about managing and governing mule applications deployed on Anypoint platform.

Anypoint API Manager (API Manager) is a component of Anypoint Platform that enables you to manage, govern, and secure APIs. It leverages the runtime capabilities of API Gateway and Anypoint Service Mesh, both of which enforce policies, collect and track analytics data, manage proxies, provide encryption and authentication, and manage applications.

Prerequisite of managing an API is that the API must be published to Anypoint Exchange. Hence the correct option in C Mule Ref Doc : https://docs.mulesoft.com/api-manager/2.x/getting-started-proxy

Q89. An organization if struggling frequent plugin version upgrades and external plugin project dependencies. The team wants to minimize the impact on applications by creating best practices that will define a set of default dependencies across all new and in progress projects.

How can these best practices be achieved with the applications having the least amount of responsibility?
* Create a Mule plugin project with all the dependencies and add it as a dependency in each application’s POM.xml file
* Create a mule domain project with all the dependencies define in its POM.xml file and add each application to the domain Project
* Add all dependencies in each application’s POM.xml file
* Create a parent POM of all the required dependencies and reference each in each application’s POM.xml file

Q90. What operation can be performed through a JMX agent enabled in a Mule application?
* View object store entries
* Replay an unsuccessful message
* Set a particular tog4J2 log level to TRACE
* Deploy a Mule application

Q91. In a Mule Application, a flow contains two (2) JMS consume operations that are used to connect to a JMS broker and consume messages from two(2) JMS destination. The Mule application then joins the two JMS messages together.

The JMS broker does not implement high availability (HA) and periodically experiences scheduled outages of upto 10 mins for routine maintenance.

What is the most idiomatic (used for its intented purpose) way to build the mule flow so it can best recover from the expected outages?
* Configure a reconnection strategy for the JMS connector
* Enclose the two(2) JMS operation in an Until Successful scope
* Consider a transaction for the JMS connector
* Enclose the two(2) JMS operations in a Try scope with an Error Continue error handler

Q92. A leading bank implementing new mule API.

The purpose of API to fetch the customer account balances from the backend application and display them on the online platform the online banking platform. The online banking platform will send an array of accounts to Mule API get the account balances.

As a part of the processing the Mule API needs to insert the data into the database for auditing purposes and this process should not have any performance related implications on the account balance retrieval flow How should this requirement be implemented to achieve better throughput?
* Implement the Async scope fetch the data from the backend application and to insert records in the Audit database
* Implement a for each scope to fetch the data from the back-end application and to insert records into the Audit database
* Implement a try-catch scope to fetch the data from the back-end application and use the Async scope to insert records into the Audit database
* Implement parallel for each scope to fetch the data from the backend application and use Async scope to insert the records into the Audit database

Q93. A global organization operates datacenters in many countries. There are private network links between these datacenters because all business data (but NOT metadata) must be exchanged over these private network connections.

The organization does not currently use AWS in any way.

The strategic decision has Just been made to rigorously minimize IT operations effort and investment going forward.

What combination of deployment options of the Anypoint Platform control plane and runtime plane(s) best serves this organization at the start of this strategic journey?
* MuleSoft-hosted Anypoint Platform control plane CloudHub Shared Worker Cloud in multiple AWS regions
* Anypoint Platform – Private Cloud Edition Customer-hosted runtime plane in each datacenter
* MuleSoft-hosted Anypoint Platform control plane Customer-hosted runtime plane in multiple AWS regions
* MuleSoft-hosted Anypoint Platform control plane Customer-hosted runtime plane in each datacenter Hybrid is the best choice to start. Mule hosted Control plane and Customer hosted Runtime to start with. Once they mature in cloud migration, everything can be in Mule hosted.

Q94. An organization uses a set of customer-hosted Mule runtimes that are managed using the Mulesoft-hosted control plane. What is a condition that can be alerted on from Anypoint Runtime Manager without any custom components or custom coding?
* When a Mule runtime on a given customer-hosted server is experiencing high memory consumption during certain periods
* When a Mule runtime’s customer-hosted server is about to run out of disk space
* When the Mute runtime license installed on a Mule runtime is about to expire
* When an SSL certificate used by one of the deployed Mule applications is about to expire
Correct answer is When a Mule runtime on a given customer-hosted server is experiencing high memory consumption during certain periods Using Anypoint Monitoring, you can configure two different types of alerts: Basic alerts for servers and Mule apps Limit per organization: Up to 50 basic alerts for users who do not have a Titanium subscription to Anypoint Platform You can set up basic alerts to trigger email notifications when a metric you are measuring passes a specified threshold. You can create basic alerts for the following metrics for servers or Mule apps: For on-premises servers and CloudHub apps: * CPU utilization * Memory utilization * Thread count Advanced alerts for graphs in custom dashboards in Anypoint Monitoring. You must have a Titanium subscription to use this feature. Limit per organization: Up to 20 advanced alerts

Q95. A Mule application contains a Batch Job with two Batch Steps (Batch_Step_l and Batch_Step_2). A payload with 1000 records is received by the Batch Job.

How many threads are used by the Batch Job to process records, and how does each Batch Step process records within the Batch Job?
* Each Batch Job uses SEVERAL THREADS for the Batch Steps Each Batch Step instance receives ONE record at a time as the payload, and RECORDS are processed IN PARALLEL within and between the two Batch Steps

* Each Batch Job uses a SINGLE THREAD for all Batch steps Each Batch step instance receives ONE record at a time as the payload, and RECORDS are processed IN ORDER, first through Batch_Step_l and then through Batch_Step_2
* Each Batch Job uses a SINGLE THREAD to process a configured block size of record Each Batch Step instance receives A BLOCK OF records as the payload, and BLOCKS of records are processed IN ORDER
* Each Batch Job uses SEVERAL THREADS for the Batch Steps Each Batch Step instance receives ONE record at a time as the payload, and BATCH STEP INSTANCES execute IN PARALLEL to process records and Batch Steps in ANY order as fast as possible
* Each Batch Job uses SEVERAL THREADS for the Batch Steps

* Each Batch Step instance receives ONE record at a time as the payload. It’s not received in a block, as it does not wait for multiple records to be completed before moving to next batch step. (So Option D is out of choice)

* RECORDS are processed IN PARALLEL within and between the two Batch Steps.

* RECORDS are not processed in order. Let’s say if second record completes batch_step_1 before record 1, then it moves to batch_step_2 before record 1. (So option C and D are out of choice)

* A batch job is the scope element in an application in which Mule processes a message payload as a batch of records. The term batch job is inclusive of all three phases of processing: Load and Dispatch, Process, and On Complete.

* A batch job instance is an occurrence in a Mule application whenever a Mule flow executes a batch job. Mule creates the batch job instance in the Load and Dispatch phase. Every batch job instance is identified internally using a unique String known as batch job instance id.

Q96. An integration Mute application is being designed to process orders by submitting them to a backend system for offline processing. Each order will be received by the Mute application through an HTTPS POST and must be acknowledged immediately. Once acknowledged, the order will be submitted to a backend system. Orders that cannot be successfully submitted due to rejections from the backend system will need to be processed manually (outside the backend system).

The Mule application will be deployed to a customer-hosted runtime and is able to use an existing ActiveMQ broker if needed.

The backend system has a track record of unreliability both due to minor network connectivity issues and longer outages.

What idiomatic (used for their intended purposes) combination of Mule application components and ActiveMQ queues are required to ensure automatic submission of orders to the backend system, while minimizing manual order processing?
* An On Error scope Non-persistent VM ActiveMQ Dead Letter Queue for manual processing
* An On Error scope MuleSoft Object Store ActiveMQ Dead Letter Queue for manual processing
* Until Successful component MuleSoft Object Store ActiveMQ is NOT needed or used
* Until Successful component ActiveMQ long retry Queue ActiveMQ Dead Letter Queue for manual processing
Correct answer is using below set of activities Until Successful component ActiveMQ long retry Queue ActiveMQ Dead Letter Queue for manual processing We will see why this is correct answer but before that lets understand few of the concepts which we need to know. Until Successful Scope The Until Successful scope processes messages through its processors until the entire operation succeeds. Until Successful repeatedly retries to process a message that is attempting to complete an activity such as: – Dispatching to outbound endpoints, for example, when calling a remote web service that may have availability issues. – Executing a component method, for example, when executing on a Spring bean that may depend on unreliable resources. – A sub-flow execution, to keep re-executing several actions until they all succeed, – Any other message processor execution, to allow more complex scenarios. How this will help requirement : Using Until Successful Scope we can retry sending the order to backend systems in case of error to avoid manual processing later. Retry values can be configured in Until Successful Scope Apache ActiveMQ It is an open source message broker written in Java together with a full Java Message Service client ActiveMQ has the ability to deliver messages with delays thanks to its scheduler. This functionality is the base for the broker

redelivery plug-in. The redelivery plug-in can intercept dead letter processing and reschedule the failing messages for redelivery. Rather than being delivered to a DLQ, a failing message is scheduled to go to the tail of the original queue and redelivered to a message consumer. How this will help requirement : If backend application is down for a longer duration where Until Successful Scope wont work, then we can make use of ActiveMQ long retry Queue. The redelivery plug-in can intercept dead letter processing and reschedule the failing messages for redelivery. Mule Reference:
https://docs.mulesoft.com/mule-runtime/4.3/migration-core-until-successful

Q97. Mule application A receives a request Anypoint MQ message REQU with a payload containing a variable-length list of request objects. Application A uses the For Each scope to split the list into individual objects and sends each object as a message to an Anypoint MQ queue.

Service S listens on that queue, processes each message independently of all other messages, and sends a response message to a response queue.

Application A listens on that response queue and must in turn create and publish a response Anypoint MQ message RESP with a payload containing the list of responses sent by service S in the same order as the request objects originally sent in REQU.

Assume successful response messages are returned by service S for all request messages.

What is required so that application A can ensure that the length and order of the list of objects in RESP and REQU match, while at the same time maximizing message throughput?
* Use a Scatter-Gather within the For Each scope to ensure response message order Configure the Scatter-Gather with a persistent object store
* Perform all communication involving service S synchronously from within the For Each scope, so objects in RESP are in the exact same order as request objects in REQU
* Use an Async scope within the For Each scope and collect response messages in a second For Each scope in the order In which they arrive, then send RESP using this list of responses
* Keep track of the list length and all object indices in REQU, both in the For Each scope and in all communication involving service Use persistent storage when creating RESP
Correct answer is Perform all communication involving service S synchronously from within the For Each scope, so objects in RESP are in the exact same order as request objects in REQU : Using Anypoint MQ, you can create two types of queues: Standard queue These queues don’t guarantee a specific message order. Standard queues are the best fit for applications in which messages must be delivered quickly. FIFO (first in, first out) queue These queues ensure that your messages arrive in order. FIFO queues are the best fit for applications requiring strict message ordering and exactly-once delivery, but in which message delivery speed is of less importance Use of FIFO queue is no where in the option and also it decreased throughput. Similarly persistent object store is not the preferred solution approach when you maximizing message throughput. This rules out one of the options. Scatter Gather does not support ObjectStore. This rules out one of the options. Standard Anypoint MQ queues don’t guarantee a specific message order hence using another for each block to collect response wont work as requirement here is to ensure the order. Hence considering all the above factors the feasible approach is Perform all communication involving service S synchronously from within the For Each scope, so objects in RESP are in the exact same order as request objects in REQU

Q98. A set of integration Mule applications, some of which expose APIs, are being created to enable a new business process. Various stakeholders may be impacted by this. These stakeholders are a combination of semi-technical users (who understand basic integration terminology and concepts such as JSON and XML) and technically skilled potential consumers of the Mule applications and APIs.

What Is an effective way for the project team responsible for the Mule applications and APIs being built to communicate with these stakeholders using Anypoint Platform and its supplied toolset?
* Use Anypoint Design Center to implement the Mule applications and APIs and give the various stakeholders access to these Design Center projects, so they can collaborate and provide feedback

* Create Anypoint Exchange entries with pages elaborating the integration design, including API notebooks (where applicable) to help the stakeholders understand and interact with the Mule applications and APIs at various levels of technical depth
* Use Anypoint Exchange to register the various Mule applications and APIs and share the RAML definitions with the stakeholders, so they can be discovered
* Capture documentation about the Mule applications and APIs inline within the Mule integration flows and use Anypoint Studio’s Export Documentation feature to provide an HTML version of this documentation to the stakeholders
As the stakeholders are semitechnical users , preferred option is Create Anypoint Exchange entries with pages elaborating the integration design, including API notebooks (where applicable) to help the stakeholders understand and interact with the Mule applications and APIs at various levels of technical depth

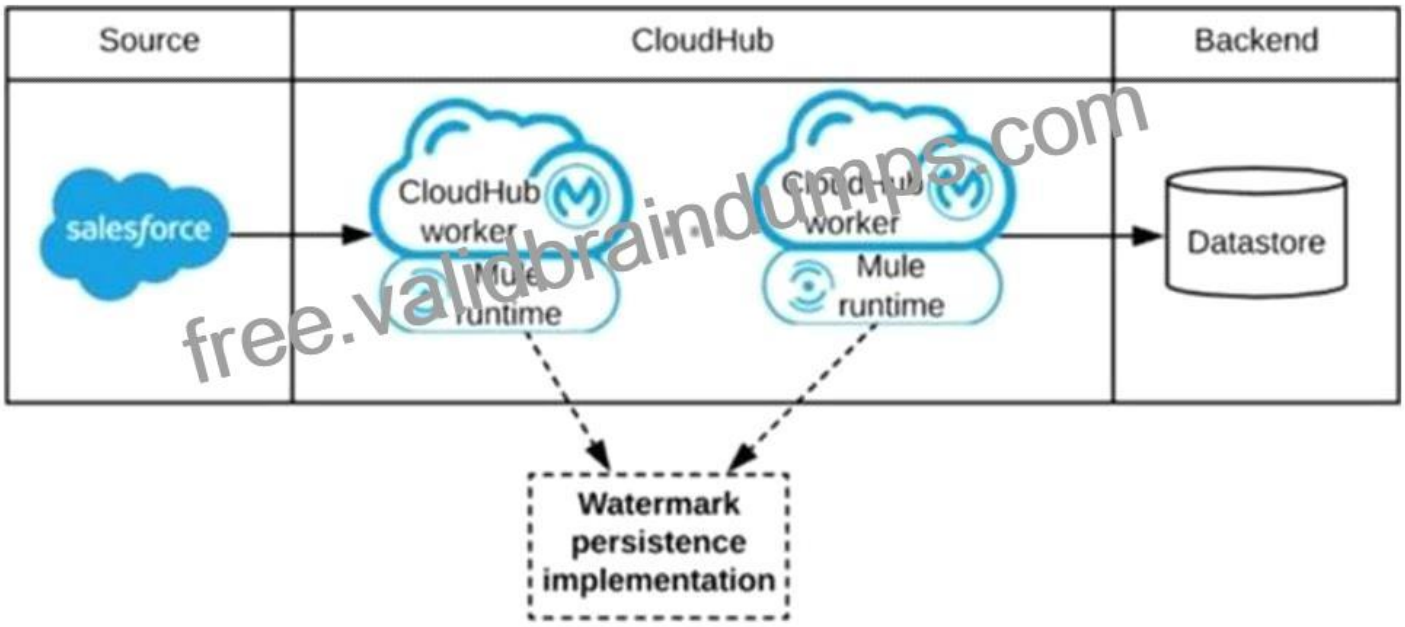
Q99. What aspects of a CI/CD pipeline for Mule applications can be automated using MuleSoft-provided Maven plugins?
* Compile, package, unit test, validate unit test coverage, deploy
* Compile, package, unit test, deploy, integration test (Incorrect)
* Compile, package, unit test, deploy, create associated API instances in API Manager
* Import from API designer, compile, package, unit test, deploy, publish to Anypoint Exchange
Correct answer is “Compile, package, unit test, validate unit test coverage, deploy” Explanation: : Anypoint Platform supports continuous integration and continuous delivery using industry standard tools Mule Maven Plugin The Mule Maven plugin can automate building, packaging and deployment of Mule applications from source projects Using the Mule Maven plugin, you can automate your Mule application deployment to CloudHub, to Anypoint Runtime Fabric, or on-premises, using any of the following deployment strategies * CloudHub deployment * Runtime Fabric deployment * Runtime Manager REST API deployment * Runtime Manager agent deployment MUnit Maven Plugin The MUnit Maven plugin can automate test execution, and ties in with the Mule Maven plugin. It provides a full suite of integration and unit test capabilities, and is fully integrated with Maven and Surefire for integration with your continuous deployment environment. Since MUnit 2.x, the coverage report goal is integrated with the maven reporting section. Coverage Reports are generated during Maven’s site lifecycle, during the coverage-report goal. One of the features of MUnit Coverage is to fail the build if a certain coverage level is not reached. MUnit is not used for integration testing Also publishing to Anypoint Exchange or to create associated API instances in API Manager is not a part of CICD pipeline which can ne achieved using mulesoft provided maven plugin Architecture mentioned in the question can be diagrammatically put as below. Persistent Object Store is the correct answer .

* Mule Object Stores: An object store is a facility for storing objects in or across Mule applications. Mule uses object stores to persist data for eventual retrieval.

Mule provides two types of object stores:

1) In-memory store – stores objects in local Mule runtime memory. Objects are lost on shutdown of the Mule runtime. So we cant use in memory store in our scenario as we want to share watermark within all cloudhub workers

2) Persistent store – Mule persists data when an object store is explicitly configured to be persistent. Hence this watermark will be available even any of the worker goes down



Q100. An organization uses a four(4) node customer hosted Mule runtime cluster to host one(1) stateless api implementation. The API is accessed over HTTPS through a load balancer that uses round-robin for load distribution. Each node in the cluster has been sized to be able to accept four(4) times the current number of requests.

Two(2) nodes in the cluster experience a power outage and are no longer available. The load balancer directs the outage and blocks the two unavailable the nodes from receiving further HTTP requests.

What performance-related consequence is guaranteed to happen to average, assuming the remaining cluster nodes are fully operational?
* 100% increase in the average response time of the API
* 50% reduction in the throughput of the API
* 100% increase in the number of requests received by each remaining node
* 50% increase in the JVM heap memory consumed by each remaining node

Q101. Anypoint Exchange is required to maintain the source code of some of the assets committed to it, such as Connectors, Templates, and API specifications.

What is the best way to use an organization’s source-code management (SCM) system in this context?
* Organizations should continue to use an SCM system of their choice, in addition to keeping source code for these asset types in Anypoint Exchange, thereby enabling parallel development, branching, and merging
* Organizations need to use Anypoint Exchange as the main SCM system to centralize versioning and avoid code duplication
* Organizations can continue to use an SCM system of their choice for branching and merging, as long as they follow the branching and merging strategy enforced by Anypoint Exchange
* Organizations need to point Anypoint Exchange to their SCM system so Anypoint Exchange can pull source code when requested by developers and provide it to Anypoint Studio
* Organization should continue to use SCM system of their choice, in addition to keeping source code for these asset types in Anypoint Exchange, thereby enabling parallel development, branching.

* Reason is that Anypoint exchange is not full fledged version repositories like GitHub.

* But at same time it is tightly coupled with Mule assets

Q102. What is an example of data confidentiality?
* Signing a file digitally and sending it using a file transfer mechanism
* Encrypting a file containing personally identifiable information (PV)
* Providing a server’s private key to a client for secure decryption of data during a two-way SSL handshake
* De-masking a person’s Social Security number while inserting it into a database

Q103. An organization will deploy Mule applications to Cloudhub, Business requirements mandate that all application logs be stored ONLY in an external splunk consolidated logging service and NOT in Cloudhub.

In order to most easily store Mule application logs ONLY in Splunk, how must Mule application logging be configured in Runtime Manager, and where should the log4j2 splunk appender be defined?
* Keep the default logging configuration in RuntimeManager

Define the splunk appender in ONE global log4j.xml file that is uploaded once to Runtime Manager to support at Mule application deployments.
* Disable Cloudhub logging in Runtime Manager

Define the splunk appender in EACH Mule application’s log4j2.xml file
* Disable Cloudhub logging in Runtime Manager

Define the splunk appender in ONE global log4j.xml file that is uploaded once to Runtime Manger to support at Mule application deployments.
* Keep thedefault logging configuration in Runtime Manager

Define the Splunk appender in EACH Mule application log4j2.xml file

Q104. An integration Mule application is deployed to a customer-hosted multi-node Mule 4 runtime cluster. The Mule application uses a Listener operation of a JMS connector to receive incoming messages from a JMS queue.

How are the messages consumed by the Mule application?
* Regardless of the Listener operation configuration, all messages are consumed by ONLY the primary cluster node
* Depending on the JMS provider’s configuration, either all messages are consumed by ONLY the primary cluster node or else ALL messages are consumed by ALL cluster nodes
* Regardless of the Listener operation configuration, all messages are consumed by ALL cluster nodes
* Depending on the Listener operation configuration, either all messages are consumed by ONLY the primary cluster node or else EACH message is consumed by ANY ONE cluster node

Q105. An application load balancer routes requests to a RESTful web API secured by Anypoint Flex Gateway.

Which protocol is involved in the communication between the load balancer and the Gateway?
* SFTP
* HTTPS
* LDAP
* SMTP

Q106. A project team uses RAML specifications to document API functional requirements and deliver API definitions. As per the current legal requirement, all designed API definitions to be augmented with an additional non-functional requirement to protect the

services from a high rate of requests according to define service level agreements.

Assuming that the project is following Mulesoft API governance and policies, how should the project team convey the necessary non-functional requirement to stakeholders?
* Create proxies in API manager for the non functional requirement and publish to exchange
* Add all non functional requirements as comments to RAML specification and publish to exchange
* Create various SLA’s in API manager for the non functional requirement and publish to exchange
* Update API definitions with the fragment for the appropriate policy and publish to exchange

Q107. What requires configuration of both a key store and a trust store for an HTTP Listener?
* Support for TLS mutual (two-way) authentication with HTTP clients
* Encryption of requests to both subdomains and API resource endpoints fhttPs://aDi.customer.com/ and https://customer.com/api)
* Encryption of both HTTP request and HTTP response bodies for all HTTP clients
* Encryption of both HTTP request header and HTTP request body for all HTTP clients

Q108. An Order microservice and a Fulfillment microservice are being designed to communicate with their dients through message-based integration (and NOT through API invocations).

The Order microservice publishes an Order message (a kind of command message) containing the details of an order to be fulfilled. The intention is that Order messages are only consumed by one Mute application, the Fulfillment microservice.

The Fulfilment microservice consumes Order messages, fulfills the order described therein, and then publishes an OrderFulfilted message (a kind of event message). Each OrderFulfilted message can be consumed by any interested Mule application, and the Order microservice is one such Mute application.

What is the most appropriate choice of message broker(s) and message destination(s) in this scenario?
* O Order messages are sent to an Anypoint MQ exchange

OrderFulfilted messages are sent to an Anypoint MQ queue

Both microservices interact with Anypoint MQ as the message broker, which must therefore scale to support the toad of both microservices
* Older messages are sent directly to the Fulfillment microservices

OrderFulfilled messages are sent directly to the Order microservice

The Order microserviceInteracts with one AMQP-compatible message broker and the Fulfillment microservice Interactswith a different AMQP-compatible message broker, so that both message brokers can be chosen and scaled to best support the toad each microservice
* Order messages are sent to a JMS queueOrderFulfilled messages are sent to a JMS topicBoth microservices Interact with the same JMS provider (message broker) Instance, which must therefore scale to support the load of both microservices
* Order messages are sent to a JMS queueOrderFulfilled messages are sent to a JMS topic The Order microservice Interacts with one JMS provider (message broker) and the Fulfillment microservice interacts with a different JMS provider, so that both message brokers can be chosen and scaled to best support the load of each microservice

Updated Verified MCIA-Level-1 dumps Q&As - Pass Guarantee or Full Refund:
https://www.validbraindumps.com/MCIA-Level-1-exam-prep.html]