MCIA-LEVEL-1-MAINTENANCE^{Q&As}

MuleSoft Certified Integration Architect - Level 1 MAINTENANCE

Pass Mulesoft MCIA-LEVEL-1-MAINTENANCE Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/mcia-level-1-maintenance.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft
Official Exam Center

QUESTION 1

A company is using Mulesoft to develop API\\\'s and deploy them to Cloudhub and on premises targets. Recently it has decided to enable Runtime Fabric deployment option as well and infrastructure is set up for this option.

What can be used to deploy Runtime Fabric?

A. AnypointCLI

B. Anypoint platform REST API\\\'s

C. Directly uploading ajar file from the Runtime manager

D. Mule maven plug-in

Correct Answer: D

QUESTION 2

A company is modernizing its legal systems lo accelerate access lo applications and data while supporting the adoption of new technologies. The key to achieving this business goal is unlocking the companies\\\' key systems and dala including microservices miming under Docker and kubernetes containers using apis.

Considering the current aggressive backlog and project delivery requirements the company wants to take a strategic approach in the first phase of its transformation projects by quickly deploying API\\\'s in mule runtime that are able lo scale, connect to on premises systems and migrate as needed.

Which runtime deployment option supports company\\\'s goals?

A. Customer hosted self provisioned runtimes

B. Cloudhub runtimes

C. Runtime fabric on self managed Kubernetes

D. Runtime fabric on Vmware metal

Correct Answer: C

QUESTION 3

The AnyAirline organization\\\'s passenger reservations center is designing an integration solution that combines invocations of three different System APIs (bookFlight, bookHotel, and bookCar) in a business transaction. Each System API

makes calls to a single database.

The entire business transaction must be rolled back when at least one of the APIs fails.

What is the most idiomatic (used for its intended purpose) way to integrate these APIs in near real-time that provides the best balance of consistency, performance, and reliability?

A. Implement eXtended Architecture (XA) transactions between the API implementations Coordinate between the API implementations using a Saga pattern Implement caching in each API implementation to improve performance

B. Implement local transactions within each API implementation Configure each API implementation to also participate in the same eXtended Architecture (XA) transaction Implement caching in each API implementation to improve performance

C. Implement local transactions in each API implementation Coordinate between the API implementations using a Saga pattern Apply various compensating actions depending on where a failure occurs

D. Implement an eXtended Architecture (XA) transaction manager in a Mule application using a Saga pattern Connect each API implementation with the Mule application using XA transactions Apply various compensating actions depending on where a failure occurs

Correct Answer: C

Reference: https://aws.amazon.com/blogs/compute/building-a-serverless-distributed- application-using-a-saga-orchestration-pattern/

QUESTION 4

An organization is designing a mule application to support an all or nothing transaction between serval database operations and some other connectors so that they all roll back if there is a problem with any of the connectors. Besides the database connector , what other connector can be used in the transaction.

A. VM

B. Anypoint MQ

C. SFTP

D. ObjectStore

Correct Answer: A

Correct answer is VM VM support Transactional Type. When an exception occur, The transaction rolls back to its original state for reprocessing. This feature is not supported by other connectors. Here is additional information about Transaction management:

	Shared Load Balancer	Dedicated Load Balancer
VPC	Shared VPC (Mulesoft)	VPC (Customer)
Default Load Balancer	Cloudhub provides Deault Shared Load Balancer available in All Environment	Need to Purchase
Organization Use	Multiple Oragnization	Specific to Organization
Certificate	Mulesoft Certificate	Organization Certificate
TLS Support	Yes	Yes
URL Mapping	Fixed URL Mapping	Customer URL Mapping
Timeout	30 Sec Session Timeout	Custom Timeout
Ports	Public Port {80 : 8081, 443 : 8082}	Private Port {80 : 8091, 443 : 8092}
Fashion	Round Robin	Round Robin
Supports HTTPS Protocol	Yes	Yes
Worker Assignment	No	Yes
IP Blacklisting/ Whitelisting	No	Yes
	https://docs.mulesoft.com/runtime-manager/lb-whitelists	
Configure Custom Domain	No	Yes
Custom Certificate	No	Yes
Rate Limit	Lower Rate Limit and applied According to Region	Higher Rate Limit Threshold
VPC	Anypoint VPC optional	Can't Use DLB without Anypoint VPC

QUESTION 5

What API policy would LEAST likely be applied to a Process API?

A. Custom circuit breaker

B. Client ID enforcement

C. Rate limiting

D. JSON threat protection

Correct Answer: D

Key to this question lies in the fact that Process API are not meant to be accessed directly by clients. Lets analyze options one by one. Client ID enforcement : This is applied at process API level generally to ensure that identity of API clients is always known and available for API-based analytics Rate Limiting : This policy is applied on Process Level API to secure API\'s against degradation of service that can happen in case load received is more than it can handle Custom circuit breaker : This is also quite useful feature on process level API\'s as it saves the API client the wasted time and effort of invoking a failing API. JSON threat protection : This policy is not required at Process API and rather implemented as Experience API\'s. This policy is used to safeguard application from malicious attacks by injecting malicious code in JSON object. As ideally Process API\'s are never called from external world , this policy is never used on Process API\'s Hence correct answer is JSON threat protection MuleSoft Documentation Reference : https://docs.mulesoft.com/api-manager/2.x/policy-mule3-json-threat

---

QUESTION 6

Which of the below requirements prevent the usage of Anypoint MQ in a company\'s network? (Choose two answers)
A. single message payload can be up to 15 MB

B. payloads must be encrypted

C. the message broker must be hosted on premises

D. support for point-to-point messaging

E. ability for a third party outside the company\'s network to consume events from the queue

Correct Answer: CD

QUESTION 7

Customer has deployed mule applications to different customer hosted mule run times. Mule applications are managed from Anypoint platform.

What needs to be configured to monitor these Mule applications from Anypoint monitoring and what sends monitoring data to Anypoint monitoring?

A. Enable monitoring of individual applications from runtime manager application settings Runtime manager agent sends monitoring data from the mule applications to Anypoint monitoring

B. Install runtime manager agent on each mule runtime Runtime manager agent since monitoring data from the mule applications to Anypoint monitoring

C. Anypoint monitoring agent on each mule runtime Anypoint monitoring agent sends monitoring data from the mule applications to Anypoint monitoring

D. By default, Anypoint monitoring agent will be installed on each Mule run time Anypoint Monitoring agent automatically sends monitoring data from the Mule applications to Anypoint monitoring

Correct Answer: C

QUESTION 8

Mule application is deployed to Customer Hosted Runtime. Asynchronous logging was implemented to improved throughput of the system. But it was observed over the period of time that few of the important exception log messages which were used to rollback transactions are not working as expected causing huge loss to the Organization. Organization wants to avoid these losses. Application also has constraints due to which they cant compromise on throughput much. What is the possible option in this case?

A. Logging needs to be changed from asynchronous to synchronous

B. External log appender needs to be used in this case

C. Persistent memory storage should be used in such scenarios

D. Mixed configuration of asynchronous or synchronous loggers should be used to log exceptions via synchronous way

Correct Answer: D

Correct approach is to use Mixed configuration of asynchronous or synchronous loggers shoud be used to log exceptions via synchronous way Asynchronous logging poses a performance-reliability trade-off. You may lose some messages if Mule crashes before the logging buffers flush to the disk. In this case, consider that you can have a mixed configuration of asynchronous or synchronous loggers in your app. Best practice is to use asynchronous logging over synchronous with a minimum logging level of WARN for a production application. In some cases, enable INFO logging level when you need to confirm events such as successful policy installation or to perform troubleshooting. Configure your logging strategy by editing your application\\'s src/main/resources/log4j2.xml file

---

QUESTION 9

An API implementation is being developed to expose data from a production database via HTTP requests. The API implementation executes a database SELECT statement that is dynamically created based upon data received from each incoming HTTP request. The developers are planning to use various types of testing to make sure the Mule application works as expected, can handle specific workloads, and behaves correctly from an API consumer perspective. What type of testing would typically mock the results from each SELECT statement rather than actually execute it in the production database?

A. Unit testing (white box)

B. Integration testing

C. Functional testing (black box)

D. Performance testing

Correct Answer: A

In Unit testing instead of using actual backends, stubs are used for the backend services. This ensures that developers are not blocked and have no dependency on other systems. In Unit testing instead of using actual backends, stubs are used for the backend services. This ensures that developers are not blocked and have no dependency on other systems. Below are the typical characteristics of unit testing. -- Unit tests do not require deployment into any special environment, such as a staging environment -- Unit tests san be run from within an embedded Mule runtime -- Unit tests can/should be implemented using MUnit -- For read-only interactions to any dependencies (such as other APIs): allowed

to invoke production endpoints -- For write interactions: developers must implement mocks using MUnit -- Require knowledge of the implementation details of the API implementation under test

QUESTION 10

An airline is architecting an API connectivity project to integrate its flight data into an online aggregation website. The interface must allow for secure communication high-performance and asynchronous message exchange. What are suitable interface technologies for this integration assuming that Mulesoft fully supports these technologies and that Anypoint connectors exist for these interfaces?

A. AsyncAPI over HTTPS AMQP with RabbitMQ JSON/REST over HTTPS

B. XML over ActiveMQ XML over SFTP XML/REST over HTTPS

C. CSV over FTP YAM L over TLS JSON over HTTPS

D. SOAP over HTTPS HOP over TLS gRPC over HTTPS

Correct Answer: A

QUESTION 11

An organization has strict unit test requirement that mandate every mule application must have an MUnit test suit with a test case defined for each flow and a minimum test coverage of 80%.

A developer is building Munit test suit for a newly developed mule application that sends API request to an external rest API.

What is the effective approach for successfully executing the Munit tests of this new application while still achieving the required test coverage for the Munit tests?

A. Invoke the external endpoint of the rest API from the mule floors

B. Mark the rest API invocations in the Munits and then call the mocking service flow that simulates standard responses from the REST API

C. Mock the rest API invocation in the Munits and return a mock response for those invocations

D. Create a mocking service flow to simulate standard responses from the rest API and then configure the mule flows to call the marking service flow

Correct Answer: C

QUESTION 12

An API implementation is being designed that must invoke an Order API which is known to repeatedly experience downtime. For this reason a fallback API is to be called when the Order API is unavailable. What approach to designing invocation of the fallback API provides the best resilience?

A. Redirect client requests through an HTTP 303 temporary redirect status code to the fallback API whenever the Order API is unavailable

B. Set an option in the HTTP Requester component that invokes the order API to instead invoke a fallback API whenever an HTTP 4XX or 5XX response status code is received from Order API

C. Create a separate entry for the order API in API manager and then invoke this API as a fallback API if the primary Order API is unavailable

D. Search Anypoint Exchange for a suitable existing fallback API and them implement invocations to their fallback API in addition to the Order API

Correct Answer: A

*

 Resilience testing is a type of software testing that observes how applications act under stress. It\\'s meant to ensure the product\\'s ability to perform in chaotic conditions without a loss of core functions or data; it ensures a quick recovery after unforeseen, uncontrollable events.

*

 In case an API invocation fails -- even after a certain number of retries -- it might be adequate to invoke a different API as a fallback. A fallback API, by definition, will never be ideal for the purpose of the API client, otherwise it would be the primary API.

*

 Here are some examples for fallback APIs:

-

An old, deprecated version of the same API.

-

An alternative endpoint of the same API and version (e.g. API in another CloudHub region).

-

An API doing more than required, and therefore not as performant as the primary API.

- An API doing less than required and therefore forcing the API Client to offer a degraded service, which is still better than no service at all.

*

 API clients implemented as Mule applications offer the `Until Successful Scope and Exception\\' strategies at their disposal, which together allow configuring fallback actions such as a fallback API invocation.

*

 All HTTP response status codes within the 3xx category are considered redirection messages. These codes indicate to the user agent (i.e. your web browser) that an additional action is required in order to complete the request and access the desired resource
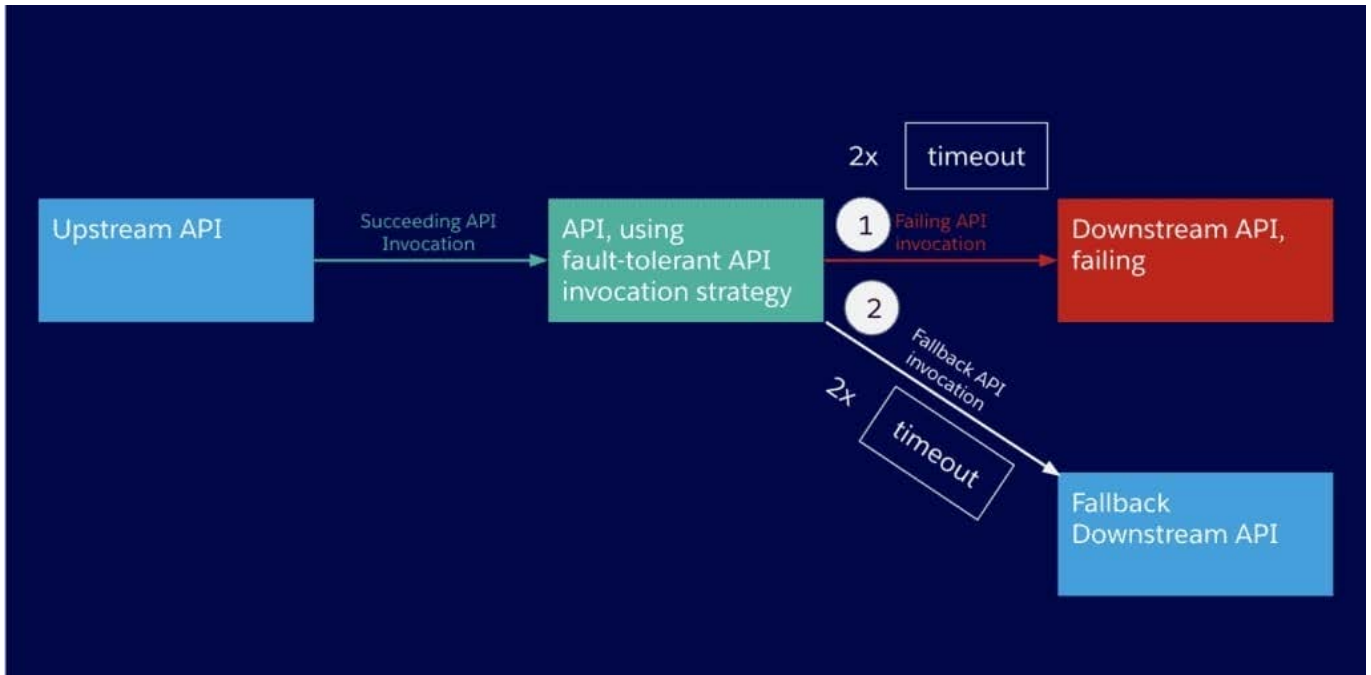
Hence correct answer is Redirect client requests through an HTTP 303 temporary redirect status code to the fallback API whenever the Order API is unavailable

QUESTION 13

An insurance company is implementing a MuleSoft API to get inventory details from the two vendors. Due to network issues, the invocations to vendor applications are getting timed- out intermittently. But the transactions are successful upon reprocessing

What is the most performant way of implementing this requirement?

A. Implement a scatter-gather scope to invoke the two vendor applications on two different route Use the Until-Successful scope to implement the retry mechanism for timeout errors on each route

B. Implement a Choice scope to invoke the two vendor applications on two different route Use the try-catch scope to implement the retry mechanism for timeout errors on each route

C. Implement a For-Each scope to invoke the two vendor applications Use until successful scope to implement the retry mechanism for the timeout errors

D. Implement Round-Robin scope to invoke the two vendor applications on two different routes Use the Try-Catch scope to implement retry mechanism for timeout errors on each route

Correct Answer: A

QUESTION 14

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)

A. In the Java Invoke static connector configuration, configure a path and name of the dependency jar file

B. Add the dependency jar file to the java classpath by setting the JVM parameters

C. Use Maven command to include the dependency jar file when packaging the application

D. Configure the dependency as a shared library in the project POM

E. Update mule-artefact.json to export the Java package

Correct Answer: BD

QUESTION 15

A Mule application is running on a customer-hosted Mule runtime in an organization\\\'s network. The Mule application acts as a producer of asynchronous Mule events. Each Mule event must be broadcast to all interested external consumers

outside the Mule application. The Mule events should be published in a way that is guaranteed in normal situations and also minimizes duplicate delivery in less frequent failure scenarios.

The organizational firewall is configured to only allow outbound traffic on ports 80 and 443. Some external event consumers are within the organizational network, while others are located outside the firewall.

What Anypoint Platform service is most idiomatic (used for its intended purpose) for publishing these Mule events to all external consumers while addressing the desired reliability goals?

A. CloudHub VM queues

B. Anypoint MQ

C. Anypoint Exchange

D. CloudHub Shared Load Balancer

Correct Answer: B

Set the Anypoint MQ connector operation to publish or consume messages, or to accept (ACK) or not accept (NACK) a message. Reference: https://docs.mulesoft.com/mq/