Salesforce

Mule-Dev-201

Salesforce Certified MuleSoft Developer

Exams Boost
Boost Up Your Career

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ Up to Date products, reliable and verified.
✓ Questions and Answers in PDF Format.

Latest Version: 6.0

Question: 1

What is true about automating interactions with Anypoint Platform using tools such as Anypoint Platform REST APIs, Anypoint CU, or the Mule Maven plugin?

A. Access to Anypoint Platform APIs and Anypoint CU can be controlled separately through the roles and permissions in Anypoint Platform, so that specific users can get access to Anypoint CLI white others get access to the platform APIs
B. Anypoint Platform APIs can ONLY automate interactions with CloudHub, while the Mule Maven plugin is required for deployment to customer-hosted Mule runtimes
C. By default, the Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications
D. API policies can be applied to the Anypoint Platform APIs so that ONLY certain LOBs have access to specific functions

Answer: C

Explanation:
Correct Answe r: By default, the Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications
****
>> We CANNOT apply API policies to the Anypoint Platform APIs like we can do on our custom written API instances. So, option suggesting this is FALSE.
>> Anypoint Platform APIs can be used for automating interactions with both CloudHub and customer-hosted Mule runtimes. Not JUST the CloudHub. So, option opposing this is FALSE.
>> Mule Maven plugin is NOT mandatory for deployment to customer-hosted Mule runtimes. It just helps your CI/CD to have smoother automation. But not a compulsory requirement to deploy. So, option opposing this is FALSE.
>> We DO NOT have any such special roles and permissions on the platform to separately control access for some users to have Anypoint CLI and others to have Anypoint Platform APIs. With proper general roles/permissions (API Owner, Cloudhub Admin etc..), one can use any of the options (Anypoint CLI or Platform APIs). So, option suggesting this is FALSE.
Only TRUE statement given in the choices is that - Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications.
Maven is part of Studio or you can use other Maven installation for development.
CLI is convenience only. It is one of many ways how to install app to the runtime.
These are definitely NOT part of anything except your process of deployment or automation.

Question: 2

What Mule application deployment scenario requires using Anypoint Platform Private Cloud Edition or Anypoint Platform for Pivotal Cloud Foundry?

A. When it Is required to make ALL applications highly available across multiple data centers
B. When it is required that ALL APIs are private and NOT exposed to the public cloud
C. When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data
D. When ALL backend systems in the application network are deployed in the organization's intranet

Answer: C

Explanation:
Correct Answe r: When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.
****
We need NOT require to use Anypoint Platform PCE or PCF for the below. So these options are OUT.
>> We can make ALL applications highly available across multiple data centers using CloudHub too.
>> We can use Anypoint VPN and tunneling from CloudHub to connect to ALL backend systems in the application network that are deployed in the organization's intranet.
>> We can use Anypoint VPC and Firewall Rules to make ALL APIs private and NOT exposed to the public cloud.
Only valid reason in the given options that requires to use Anypoint Platform PCE/ PCF is - When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.

Question: 3

What is typically NOT a function of the APIs created within the framework called API-led connectivity?

A. They provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.
B. They allow for innovation at the user Interface level by consuming the underlying assets without being aware of how data Is being extracted from backend systems.
C. They reduce the dependency on the underlying backend systems by helping unlock data from backend systems In a reusable and consumable way.
D. They can compose data from various sources and combine them with orchestration logic to create higher level value.

Answer: A

Explanation:
Correct Answe r: They provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.
****
In API-led connectivity,
>> Experience APIs - allow for innovation at the user interface level by consuming the underlying assets without being aware of how data is being extracted from backend systems.
>> Process APIs - compose data from various sources and combine them with orchestration logic to create higher level value
>> System APIs - reduce the dependency on the underlying backend systems by helping unlock data from backend systems in a reusable and consumable way.
However, they NEVER promise that they provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.
https://dzone.com/articles/api-led-connectivity-with-mule

Question: 4

An organization has implemented a Customer Address API to retrieve customer address information. This API has been deployed to multiple environments and has been configured to enforce client IDs everywhere.
A developer is writing a client application to allow a user to update their address. The developer has found the Customer Address API in Anypoint Exchange and wants to use it in their client application.
What step of gaining access to the API can be performed automatically by Anypoint Platform?

A. Approve the client application request for the chosen SLA tier
B. Request access to the appropriate API Instances deployed to multiple environments using the client application's credentials
C. Modify the client application to call the API using the client application's credentials
D. Create a new application in Anypoint Exchange for requesting access to the API

Answer: A

Explanation:
Correct Answe r: Approve the client application request for the chosen SLA tier
****
>> Only approving the client application request for the chosen SLA tier can be automated
>> Rest of the provided options are not valid

Question: 5

What is a typical result of using a fine-grained rather than a coarse-grained API deployment model to implement a given business process?

A. A decrease in the number of connections within the application network supporting the business process
B. A higher number of discoverable API-related assets in the application network
C. A better response time for the end user as a result of the APIs being smaller in scope and complexity
D. An overall tower usage of resources because each fine-grained API consumes less resources

Answer: B

Explanation:
Correct Answe r: A higher number of discoverable API-related assets in the application network.
****
>> We do NOT get faster response times in fine-grained approach when compared to coarse-grained approach.
>> In fact, we get faster response times from a network having coarse-grained APIs compared to a network having fine-grained APIs model. The reasons are below.
Fine-grained approach:
1. will have more APIs compared to coarse-grained
2. So, more orchestration needs to be done to achieve a functionality in business process.
3. Which means, lots of API calls to be made. So, more connections will needs to be established.
So, obviously more hops, more network i/o, more number of integration points compared to coarse-grained approach where fewer APIs with bulk functionality embedded in them.
4. That is why, because of all these extra hops and added latencies, fine-grained approach will have bit more response times compared to coarse-grained.
5. Not only added latencies and connections, there will be more resources used up in fine-grained approach due to more number of APIs.
That's why, fine-grained APIs are good in a way to expose more number of resuable assets in your network and make them discoverable. However, needs more maintenance, taking care of integration points, connections, resources with a little compromise w.r.t network hops and response times.

Question: 6

What correctly characterizes unit tests of Mule applications?

A. They test the validity of input and output of source and target systems
B. They must be run in a unit testing environment with dedicated Mule runtimes for the environment
C. They must be triggered by an external client tool or event source
D. They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity

Explanation:
Correct Answe r: They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity.
****

Below TWO are characteristics of Integration Tests but NOT unit tests:
>> They test the validity of input and output of source and target systems.
>> They must be triggered by an external client tool or event source.
It is NOT TRUE that Unit Tests must be run in a unit testing environment with dedicated Mule runtimes for the environment.
MuleSoft offers MUnit for writing Unit Tests and they run in an embedded Mule Runtime without needing any separate/ dedicated Runtimes to execute them. They also do NOT need any external connectivity as MUnit supports mocking via stubs.
https://dzone.com/articles/munit-framework

Question: 7

An organization is deploying their new implementation of the OrderStatus System API to multiple workers in CloudHub. This API fronts the organization's on-premises Order Management System, which is accessed by the API implementation over an IPsec tunnel.
What type of error typically does NOT result in a service outage of the OrderStatus System API?

A. A CloudHub worker fails with an out-of-memory exception
B. API Manager has an extended outage during the initial deployment of the API implementation
C. The AWS region goes offline with a major network failure to the relevant AWS data centers
D. The Order Management System is Inaccessible due to a network outage in the organization's on-premises data center

Explanation:
Correct Answe r: A CloudHub worker fails with an out-of-memory exception.
****

>> An AWS Region itself going down will definitely result in an outage as it does not matter how many workers are assigned to the Mule App as all of those in that region will go down. This is a complete downtime and outage.
>> Extended outage of API manager during initial deployment of API implementation will of course cause issues in proper application startup itself as the API Autodiscovery might fail or API policy templates and polices may not be downloaded to embed at the time of applicaiton startup etc... there are many reasons that could cause issues.
>> A network outage onpremises would of course cause the Order Management System not accessible and it does not matter how many workers are assigned to the app they all will fail and cause outage for sure.

The only option that does NOT result in a service outage is if a cloudhub worker fails with an out-of-memory exception. Even if a worker fails and goes down, there are still other workers to handle the requests and keep the API UP and Running. So, this is the right answer.

Question: 8

An Order API must be designed that contains significant amounts of integration logic and involves the invocation of the Product API.
The power relationship between Order API and Product API is one of "Customer/Supplier", because the Product API is used heavily throughout the organization and is developed by a dedicated development team located in the office of the CTO.
What strategy should be used to deal with the API data model of the Product API within the Order API?

A. Convince the development team of the Product API to adopt the API data model of the Order API such that the integration logic of the Order API can work with one consistent internal data model
B. Work with the API data types of the Product API directly when implementing the integration logic of the Order API such that the Order API uses the same (unchanged) data types as the Product API
C. Implement an anti-corruption layer in the Order API that transforms the Product API data model into internal data types of the Order API
D. Start an organization-wide data modeling initiative that will result in an Enterprise Data Model that will then be used in both the Product API and the Order API

Answer: C

Explanation:
Correct Answe r: Convince the development team of the product API to adopt the API data model of the Order API such that integration logic of the Order API can work with one consistent internal data model
****
Key details to note from the given scenario:
>> Power relationship between Order API and Product API is customer/supplier
So, as per below rules of "Power Relationships", the caller (in this case Order API) would request for features to the called (Product API team) and the Product API team would need to accomodate those requests.

Question: 9

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 the invocation of the fallback API provides the best resilience?

A. Search Anypoint Exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the Order API
B. 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
C. Redirect client requests through an HTTP 307 Temporary Redirect status code to the fallback API whenever the Order API is unavailable
D. 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 returned from the Order API

Answer: A

Explanation:
Correct Answe r: Search Anypoint exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the order API
****
>> It is not ideal and good approach, until unless there is a pre-approved agreement with the API clients that they will receive a HTTP 3xx temporary redirect status code and they have to implement fallback logic their side to call another API.
>> Creating separate entry of same Order API in API manager would just create an another instance of it on top of same API implementation. So, it does NO GOOD by using clone od same API as a fallback API. Fallback API should be ideally a different API implementation that is not same as primary one.
>> There is NO option currently provided by Anypoint HTTP Connector that allows us to invoke a fallback API when we receive certain HTTP status codes in response.
The only statement TRUE in the given options is to Search Anypoint exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the order API.

Question: 10

How are an API implementation, API client, and API consumer combined to invoke and process an API?

A. The API consumer creates an API implementation, which receives API invocations from an API such that they are processed for an API client
B. The API client creates an API consumer, which receives API invocations from an API such that they are processed for an API implementation
C. The ApI consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation
D. The ApI client creates an API consumer, which sends API invocations to an API such that they are processed by an API implementation

Explanation:

Correct Answe r: The API consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation
****

Terminology:

>> API Client - It is a piece of code or program the is written to invoke an API

>> API Consumer - An owner/entity who owns the API Client. API Consumers write API clients.

>> API - The provider of the API functionality. Typically an API Instance on API Manager where they are managed and operated.

>> API Implementation - The actual piece of code written by API provider where the functionality of the API is implemented. Typically, these are Mule Applications running on Runtime Manager.

Thank You for Trying Our Product

For More Information – **Visit link below:**

https://www.examsboost.com/

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ **90 Days Free Updates**
- ✓ **Money Back Pass Guarantee**
- ✓ **Instant Download or Email Attachment**
- ✓ **24/7 Live Chat Support**
- ✓ **PDF file could be used at any Platform**
- ✓ **50,000 Happy Customer**