MCPA-LEVEL-1-MAINTENANCE Q&As

MuleSoft Certified Platform Architect - Level 1 MAINTENANCE

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

QUESTION 1

A company uses a hybrid Anypoint Platform deployment model that combines the EU control plane with customer-hosted Mule runtimes. After successfully testing a Mule API implementation in the Staging environment, the Mule API implementation is set with environment-specific properties and must be promoted to the Production environment. What is a way that MuleSoft recommends to configure the Mule API implementation and automate its promotion to the Production environment?

A. Bundle properties files for each environment into the Mule API implementation\\'s deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIsB.

B. Modify the Mule API implementation\\'s properties in the API Manager Properties tab, then promote the Mule API implementation to the Production environment using API Manager

C. Modify the Mule API implementation\\'s properties in Anypoint Exchange, then promote the Mule API implementation to the Production environment using Runtime Manager D. Use an API policy to change properties in the Mule API implementation deployed to the Staging environment and another API policy to deploy the Mule API implementation to the Production environment

Correct Answer: A

Bundle properties files for each environment into the Mule API implementation\\'s deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs *** >> Anypoint Exchange is for asset discovery and documentation. It has got no provision to modify the properties of Mule API implementations at all. >> API Manager is for managing API instances, their contracts, policies and SLAs. It has also got no provision to modify the properties of API implementations. >> API policies are to address Non-functional requirements of APIs and has again got no provision to modify the properties of API implementations. So, the right way and recommended way to do this as part of development practice is to bundle properties files for each environment into the Mule API implementation and just point and refer to respective file per environment.

QUESTION 2

Mule applications that implement a number of REST APIs are deployed to their own subnet that is inaccessible from outside the organization.

External business-partners need to access these APIs, which are only allowed to be invoked from a separate subnet dedicated to partners - called Partner-subnet. This subnet is accessible from the public internet, which allows these external partners to reach it.

Anypoint Platform and Mule runtimes are already deployed in Partner-subnet. These Mule runtimes can already access the APIs.

What is the most resource-efficient solution to comply with these requirements, while having the least impact on other applications that are currently using the APIs?

A. Implement (or generate) an API proxy Mule application for each of the APIs, then deploy the API proxies to the Mule runtimes

B. Redeploy the API implementations to the same servers running the Mule runtimes

C. Add an additional endpoint to each API for partner-enablement consumption

D. Duplicate the APIs as Mule applications, then deploy them to the Mule runtimes

Correct Answer: A

QUESTION 3

Traffic is routed through an API proxy to an API implementation. The API proxy is managed by API Manager and the API implementation is deployed to a CloudHub VPC using Runtime Manager. API policies have been applied to this API. In this deployment scenario, at what point are the API policies enforced on incoming API client requests?

A. At the API proxy

B. At the API implementation

C. At both the API proxy and the API implementation

D. At a MuleSoft-hosted load balancer

Correct Answer: A

At the API proxy

****

>> API Policies can be enforced at two places in Mule platform. >> One - As an Embedded Policy enforcement in the same Mule Runtime where API implementation is running.

>> Two - On an API Proxy sitting in front of the Mule Runtime where API implementation is running.

>> As the deployment scenario in the question has API Proxy involved, the policies will be enforced at the API Proxy.

QUESTION 4

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

Correct Answer: C

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 5

Which of the following sequence is correct?

A. API Client implementes logic to call an API >> API Consumer requests access to API >> API Implementation routes the request to >> API

B. API Consumer requests access to API >> API Client implementes logic to call an API >> API routes the request to >> API Implementation

C. API Consumer implementes logic to call an API >> API Client requests access to API >> API Implementation routes the request to >> API

D. API Client implementes logic to call an API >> API Consumer requests access to API >> API routes the request to >> API Implementation

Correct Answer: B

API Consumer requests access to API >> API Client implementes logic to call an API >> API routes the request to >> API Implementation *** >> API consumer does not implement any logic to invoke APIs. It is just a role. So, the option stating "API Consumer implementes logic to call an API" is INVALID. >> API Implementation does not route any requests. It is a final piece of logic where functionality of target systems is exposed. So, the requests should be routed to the API implementation by some other entity. So, the options stating "API Implementation routes the request to >> API" is INVALID >> The statements in one of the options are correct but sequence is wrong. The sequence is given as "API Client implementes logic to call an API >> API Consumer requests access to API >> API routes the request to >> API Implementation". Here, the statements in the options are VALID but sequence is WRONG. >> Right option and sequence is the one where API consumer first requests access to API on Anypoint Exchange and obtains client credentials. API client then writes logic to call an API by using the access client credentials requested by API consumer and the requests will be routed to API implementation via the API which is managed by API Manager.

Latest MCPA-LEVEL-1-MAINTENANCE Dumps	MCPA-LEVEL-1-MAINTENANCE Practice Test	MCPA-LEVEL-1-MAINTENANCE Braindumps