

Exam Questions MCPA-Level-1

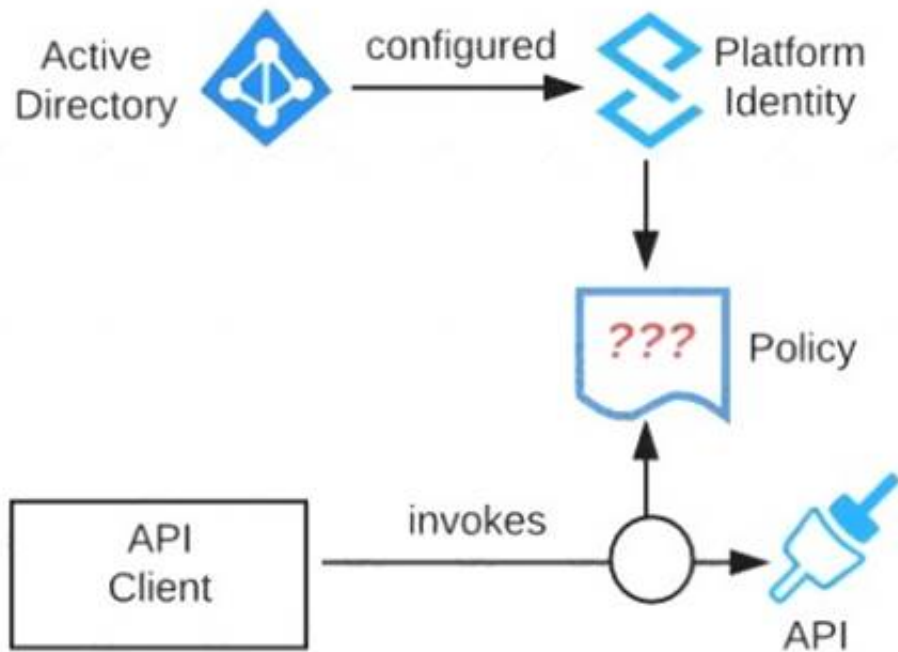
MuleSoft Certified Platform Architect - Level 1

<https://www.2passeasy.com/dumps/MCPA-Level-1/>



NEW QUESTION 1

Refer to the exhibit. An organization is running a Mule standalone runtime and has configured Active Directory as the Anypoint Platform external Identity Provider. The organization does not have budget for other system components.



What policy should be applied to all instances of APIs in the organization to most effectively restrict access to a specific group of internal users?

- A. Apply a basic authentication - LDAP policy; the internal Active Directory will be configured as the LDAP source for authenticating users
- B. Apply a client ID enforcement policy; the specific group of users will configure their client applications to use their specific client credentials
- C. Apply an IP whitelist policy; only the specific users' workstations will be in the whitelist
- D. Apply an OAuth 2.0 access token enforcement policy; the internal Active Directory will be configured as the OAuth server

Answer: A

Explanation:

Correct Answer

Apply a basic authentication - LDAP policy; the internal Active Directory will be configured as the LDAP source for authenticating users.

>> IP Whitelisting does NOT fit for this purpose. Moreover, the users workstations may not necessarily have static IPs in the network.

>> OAuth 2.0 enforcement requires a client provider which isn't in the organizations system components.

>> It is not an effective approach to let every user create separate client credentials and configure those for their usage.

The effective way it to apply a basic authentication - LDAP policy and the internal Active Directory will be configured as the LDAP source for authenticating users.

NEW QUESTION 2

In which layer of API-led connectivity, does the business logic orchestration reside?

- A. System Layer
- B. Experience Layer
- C. Process Layer

Answer: C

Explanation:

Correct Answer

Process Layer

>> Experience layer is dedicated for enrichment of end user experience. This layer is to meet the needs of different API clients/ consumers.

>> System layer is dedicated to APIs which are modular in nature and implement/ expose various individual functionalities of backend systems

>> Process layer is the place where simple or complex business orchestration logic is written by invoking one or many System layer modular APIs

So, Process Layer is the right answer.

NEW QUESTION 3

What best describes the Fully Qualified Domain Names (FQDNs), also known as DNS entries, created when a Mule application is deployed to the CloudHub Shared Worker Cloud?

- A. A fixed number of FQDNs are created, IRRESPECTIVE of the environment and VPC design
- B. The FQDNs are determined by the application name chosen, IRRESPECTIVE of the region
- C. The FQDNs are determined by the application name, but can be modified by an administrator after deployment
- D. The FQDNs are determined by both the application name and the Anypoint Platform organization

Answer: B

Explanation:

Correct Answer

The FQDNs are determined by the application name chosen, IRRESPECTIVE of the region

>> When deploying applications to Shared Worker Cloud, the FQDN are always determined by application name chosen.

>> It does NOT matter what region the app is being deployed to.

>> Although it is fact and true that the generated FQDN will have the region included in it (Ex:

exp-salesorder-api.au-s1.cloudhub.io), it does NOT mean that the same name can be used when deploying to another CloudHub region.
>> Application name should be universally unique irrespective of Region and Organization and solely determines the FQDN for Shared Load Balancers.

NEW QUESTION 4

What best explains the use of auto-discovery in API implementations?

- A. It makes API Manager aware of API implementations and hence enables it to enforce policies
- B. It enables Anypoint Studio to discover API definitions configured in Anypoint Platform
- C. It enables Anypoint Exchange to discover assets and makes them available for reuse
- D. It enables Anypoint Analytics to gain insight into the usage of APIs

Answer: A

Explanation:

Correct Answer

It makes API Manager aware of API implementations and hence enables it to enforce policies.

>> API Autodiscovery is a mechanism that manages an API from API Manager by pairing the deployed application to an API created on the platform.

>> API Management includes tracking, enforcing policies if you apply any, and reporting API analytics.

>> Critical to the Autodiscovery process is identifying the API by providing the API name and version. References:

<https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept> <https://docs.mulesoft.com/api-manager/1.x/api-auto-discovery>

<https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept>

NEW QUESTION 5

A company has created a successful enterprise data model (EDM). The company is committed to building an application network by adopting modern APIs as a core enabler of the company's IT operating model. At what API tiers (experience, process, system) should the company require reusing the EDM when designing modern API data models?

- A. At the experience and process tiers
- B. At the experience and system tiers
- C. At the process and system tiers
- D. At the experience, process, and system tiers

Answer: C

Explanation:

Correct Answer

At the process and system tiers

>> Experience Layer APIs are modeled and designed exclusively for the end user's experience. So, the data models of experience layer vary based on the nature and type of such API consumer. For example, Mobile consumers will need light-weight data models to transfer with ease on the wire, where as web-based consumers will need detailed data models to render most of the info on web pages, so on. So, enterprise data models fit for the purpose of canonical models but not of good use for experience APIs.

>> That is why, EDMs should be used extensively in process and system tiers but NOT in experience tier.

NEW QUESTION 6

What is the most performant out-of-the-box solution in Anypoint Platform to track transaction state in an asynchronously executing long-running process implemented as a Mule application deployed to multiple CloudHub workers?

- A. Redis distributed cache
- B. java.util.WeakHashMap
- C. Persistent Object Store
- D. File-based storage

Answer: C

Explanation:

Correct Answer

Persistent Object Store

>> Redis distributed cache is performant but NOT out-of-the-box solution in Anypoint Platform

>> File-storage is neither performant nor out-of-the-box solution in Anypoint Platform

>> java.util.WeakHashMap needs a completely custom implementation of cache from scratch using Java code and is limited to the JVM where it is running. Which means the state in the cache is not worker aware when running on multiple workers. This type of cache is local to the worker. So, this is neither out-of-the-box nor worker-aware among multiple workers on cloudhub. <https://www.baeldung.com/java-weakhashmap>

>> Persistent Object Store is an out-of-the-box solution provided by Anypoint Platform which is performant as well as worker aware among multiple workers running on CloudHub. <https://docs.mulesoft.com/object-store/>

So, Persistent Object Store is the right answer.

NEW QUESTION 7

Which of the following best fits the definition of API-led connectivity?

- A. API-led connectivity is not just an architecture or technology but also a way to organize people and processes for efficient IT delivery in the organization
- B. API-led connectivity is a 3-layered architecture covering Experience, Process and System layers
- C. API-led connectivity is a technology which enabled us to implement Experience, Process and System layer based APIs

Answer: A

Explanation:

Correct Answer

API-led connectivity is not just an architecture or technology but also a way to organize people and processes for efficient IT delivery in the organization.

NEW QUESTION 8

What are the major benefits of MuleSoft proposed IT Operating Model?

- A. * 1. Decrease the IT delivery gap* 2. Meet various business demands without increasing the IT capacity* 3. Focus on creation of reusable assets first
- B. Upon finishing creation of all the possible assets then inform the LOBs in the organization to start using them
- C. * 1. Decrease the IT delivery gap* 2. Meet various business demands by increasing the IT capacity and forming various IT departments* 3. Make consumption of assets at the rate of production
- D. * 1. Decrease the IT delivery gap* 2. Meet various business demands without increasing the IT capacity* 3. Make consumption of assets at the rate of production

Answer: C

Explanation:

Correct Answer

- * 1. Decrease the IT delivery gap
- * 2. Meet various business demands without increasing the IT capacity
- * 3. Make consumption of assets at the rate of production.

NEW QUESTION 9

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 APIs.
- 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

Answer: A

Explanation:

Correct Answer

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.

NEW QUESTION 10

What is a best practice when building System APIs?

- A. Document the API using an easily consumable asset like a RAML definition
- B. Model all API resources and methods to closely mimic the operations of the backend system
- C. Build an Enterprise Data Model (Canonical Data Model) for each backend system and apply it to System APIs
- D. Expose to API clients all technical details of the API implementation's interaction with the backend system

Answer: B

Explanation:

Correct Answer

Model all API resources and methods to closely mimic the operations of the backend system.

>> There are NO fixed and straight best practices while opting data models for APIs. They are completely contextual and depends on number of factors. Based upon those factors, an enterprise can choose if they have to go with Enterprise Canonical Data Model or Bounded Context Model etc.
>> One should NEVER expose the technical details of API implementation to their API clients. Only the API interface/ RAML is exposed to API clients.
>> It is true that the RAML definitions of APIs should be as detailed as possible and should reflect most of the documentation. However, just that is NOT enough to call your API as best documented API. There should be even more documentation on Anypoint Exchange with API Notebooks etc. to make and create a developer friendly API and repository..
>> The best practice always when creating System APIs is to create their API interfaces by modeling their resources and methods to closely reflect the operations and functionalities of that backend system.

NEW QUESTION 10

An API client calls one method from an existing API implementation. The API implementation is later updated. What change to the API implementation would require the API client's invocation logic to also be updated?

- A. When the data type of the response is changed for the method called by the API client
- B. When a new method is added to the resource used by the API client
- C. When a new required field is added to the method called by the API client
- D. When a child method is added to the method called by the API client

Answer: C

Explanation:

Correct Answer

When a new required field is added to the method called by the API client

>> Generally, the logic on API clients need to be updated when the API contract breaks.

>> When a new method or a child method is added to an API , the API client does not break as it can still continue to use its existing method. So these two options are out.

>> We are left for two more where "datatype of the response if changed" and "a new required field is added".

>> Changing the datatype of the response does break the API contract. However, the question is insisting on the "invocation" logic and not about the response handling logic. The API client can still invoke the API successfully and receive the response but the response will have a different datatype for some field.

>> Adding a new required field will break the API's invocation contract. When adding a new required field, the API contract breaks the RAML or API spec agreement that the API client/API consumer and API provider has between them. So this requires the API client invocation logic to also be updated.

NEW QUESTION 15

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 Answer

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.

NEW QUESTION 19

A set of tests must be performed prior to deploying API implementations to a staging environment. Due to data security and access restrictions, untested APIs cannot be granted access to the backend systems, so instead mocked data must be used for these tests. The amount of available mocked data and its contents is sufficient to entirely test the API implementations with no active connections to the backend systems. What type of tests should be used to incorporate this mocked data?

- A. Integration tests
- B. Performance tests
- C. Functional tests (Blackbox)
- D. Unit tests (Whitebox)

Answer: D

Explanation:

Correct Answer

Unit tests (Whitebox)

NEW QUESTION 22

An API implementation is deployed on a single worker on CloudHub and invoked by external API clients (outside of CloudHub). How can an alert be set up that is guaranteed to trigger AS SOON AS that API implementation stops responding to API invocations?

- A. Implement a heartbeat/health check within the API and invoke it from outside the Anypoint Platform and alert when the heartbeat does not respond
- B. Configure a "worker not responding" alert in Anypoint Runtime Manager
- C. Handle API invocation exceptions within the calling API client and raise an alert from that API client when the API is unavailable
- D. Create an alert for when the API receives no requests within a specified time period

Answer: B

Explanation:

Correct Answer

Configure a "Worker not responding" alert in Anypoint Runtime Manager.

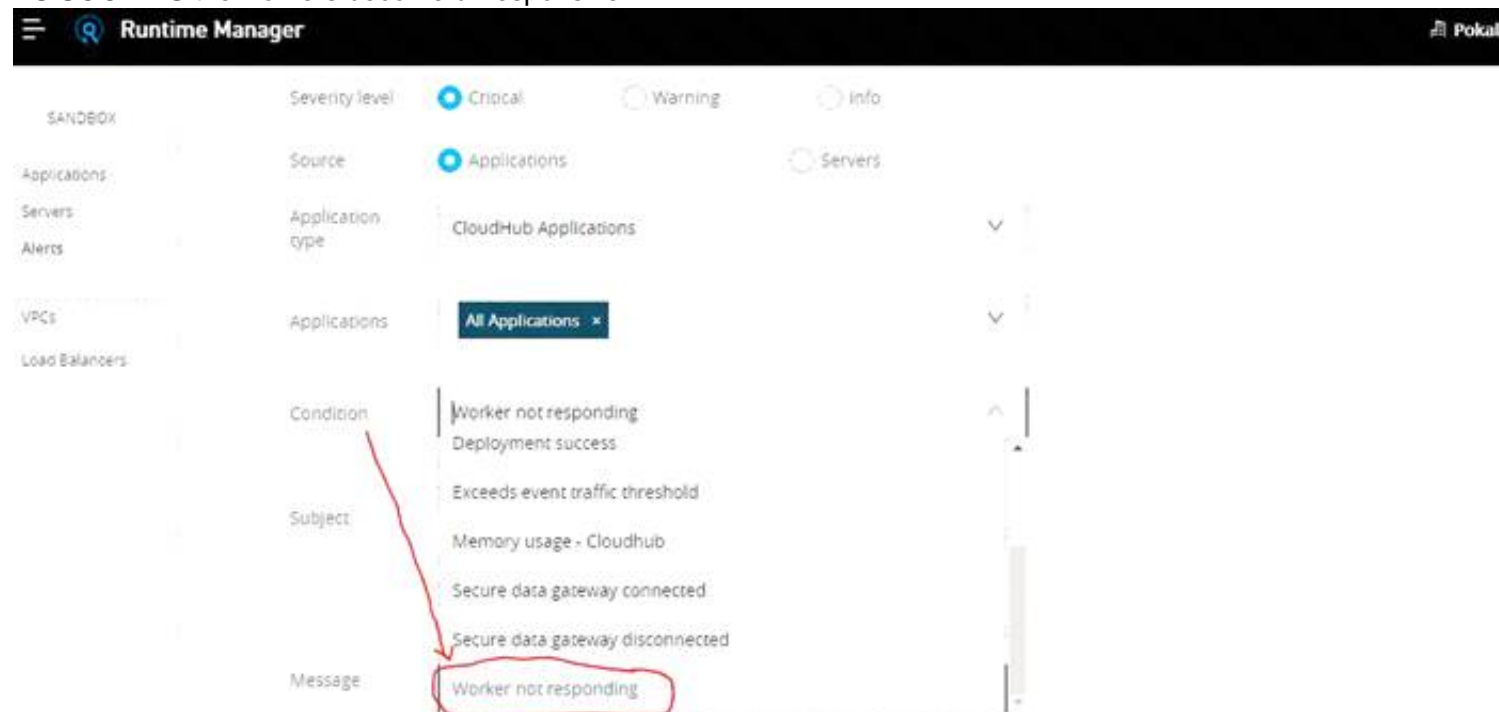
>> All the options eventually helps to generate the alert required when the application stops responding.

>> However, handling exceptions within calling API and then raising alert from API client is inappropriate and silly. There could be many API clients invoking the API implementation and it is not ideal to have this setup consistently in all of them. Not a realistic way to do.

>> Implementing a health check/ heartbeat with in the API and calling from outside to detmine the health sounds OK but needs extra setup for it and same time there are very good chances of generating false alarms when there are any intermittent network issues between external tool calling the health check API on API implementation. The API implementation itself may not have any issues but due to some other factors some false alarms may go out.

>> Creating an alert in API Manager when the API receives no requests within a specified time period would actually generate realistic alerts but even here some false alarms may go out when there are genuinely no requests from API clients.

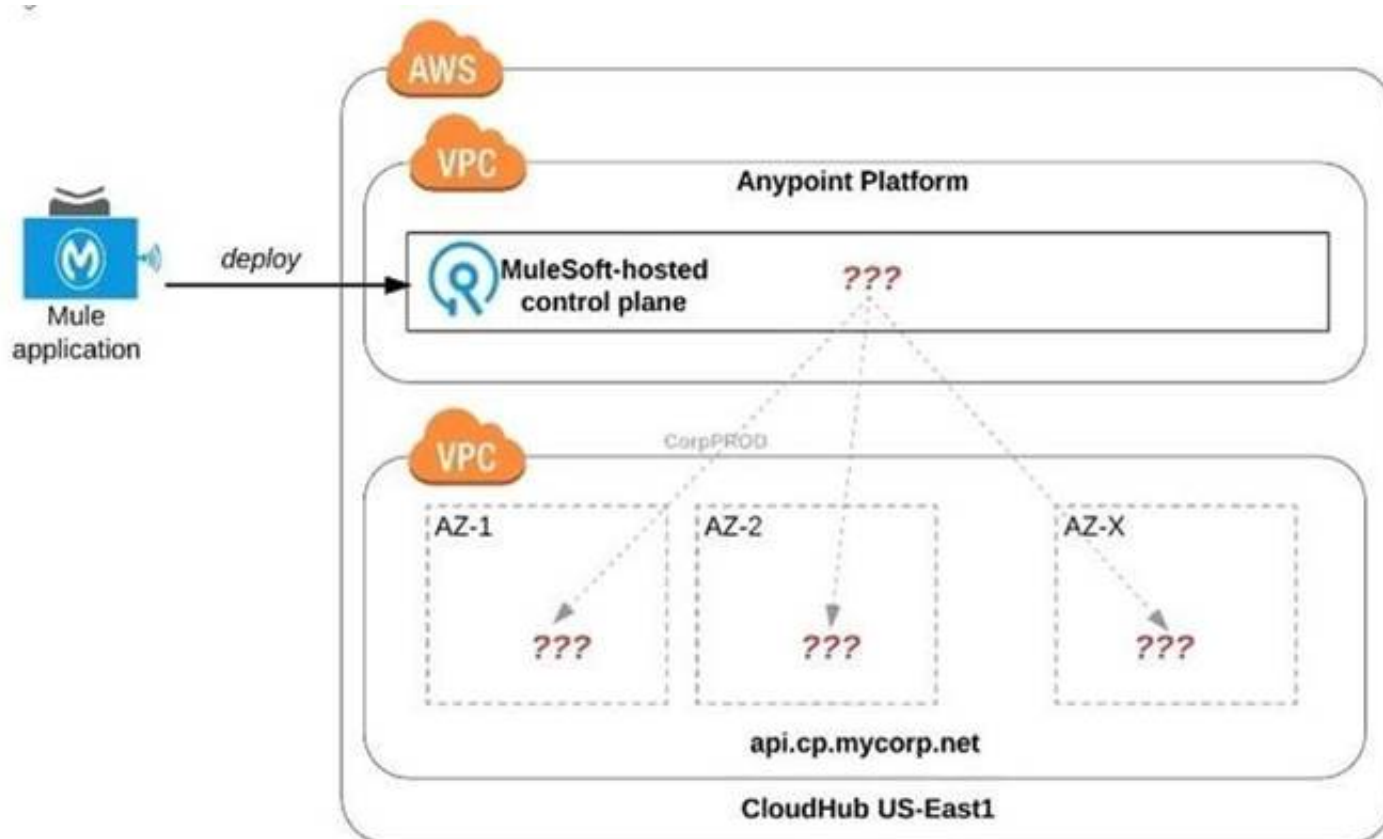
The best and right way to achieve this requirement is to setup an alert on Runtime Manager with a condition "Worker not responding". This would generate an alert AS SOON AS the workers become unresponsive.



Bottom of Form Top of Form

NEW QUESTION 27

Refer to the exhibit.



An organization uses one specific CloudHub (AWS) region for all CloudHub deployments.

How are CloudHub workers assigned to availability zones (AZs) when the organization's Mule applications are deployed to CloudHub in that region?

- A. Workers belonging to a given environment are assigned to the same AZ within that region
- B. AZs are selected as part of the Mule application's deployment configuration
- C. Workers are randomly distributed across available AZs within that region
- D. An AZ is randomly selected for a Mule application, and all the Mule application's CloudHub workers are assigned to that one AZ

Answer: D

Explanation:

Correct Answer

Workers are randomly distributed across available AZs within that region.

>> Currently, we only have control to choose which AWS Region to choose but there is no control at all using any configurations or deployment options to decide what Availability Zone (AZ) to assign to what worker.
 >> There are no fixed or implicit rules on platform too w.r.t assignment of AZ to workers based on environment or application.
 >> They are completely assigned in random. However, cloudhub definitely ensures that HA is achieved by assigning the workers to more than one AZ so that all workers are not assigned to same AZ for same application.

NEW QUESTION 31

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

Answer: D

Explanation:

Correct Answer

JSON threat protection

Fact: Technically, there are no restrictions on what policy can be applied in what layer. Any policy can be applied on any layer API. However, context should also be considered properly before blindly applying the policies on APIs.

That is why, this question asked for a policy that would LEAST likely be applied to a Process API. From the given options:

>> All policies except "JSON threat protection" can be applied without hesitation to the APIs in Process tier.

>> JSON threat protection policy ideally fits for experience APIs to prevent suspicious JSON payload coming from external API clients. This covers more of a security aspect by trying to avoid possibly malicious and harmful JSON payloads from external clients calling experience APIs.

As external API clients are NEVER allowed to call Process APIs directly and also these kind of malicious and harmful JSON payloads are always stopped at experience API layer only using this policy, it is LEAST LIKELY that this same policy is again applied on Process Layer API.

NEW QUESTION 35

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.

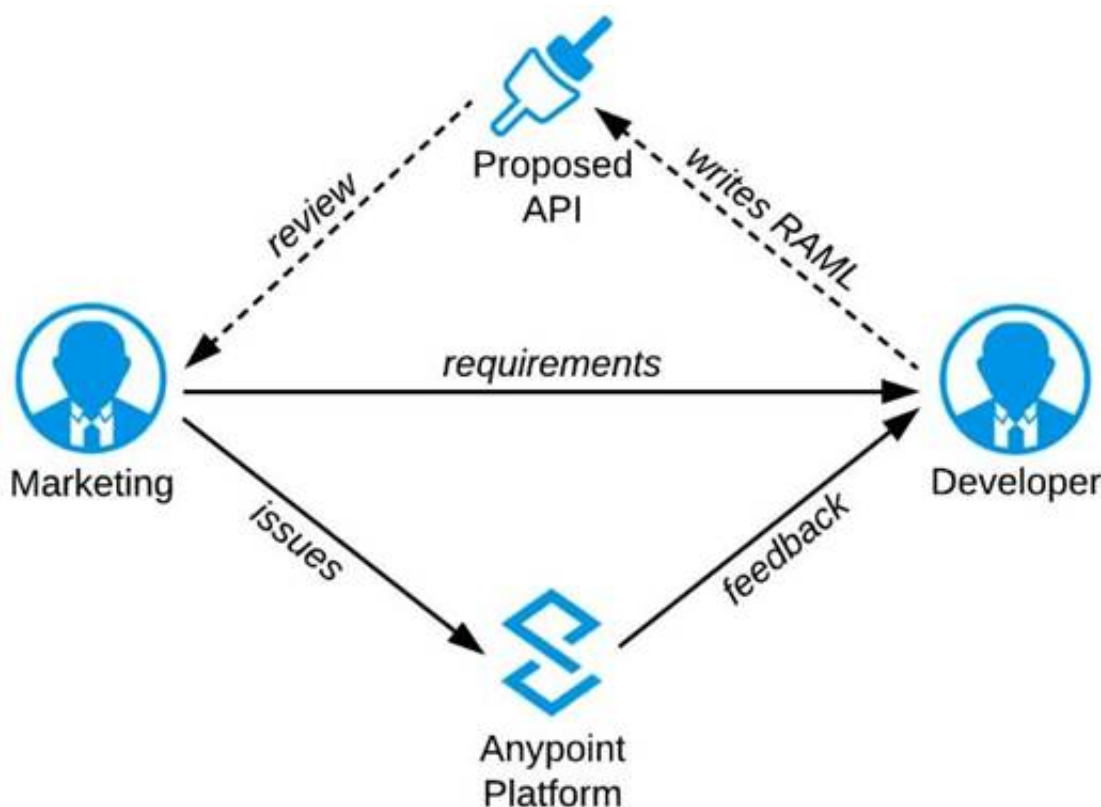
What is the most resource-efficient solution to comply with these requirements, while having the least impact on other applications that are currently using the APIs?

- 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

Answer: A

NEW QUESTION 38

Refer to the exhibit.

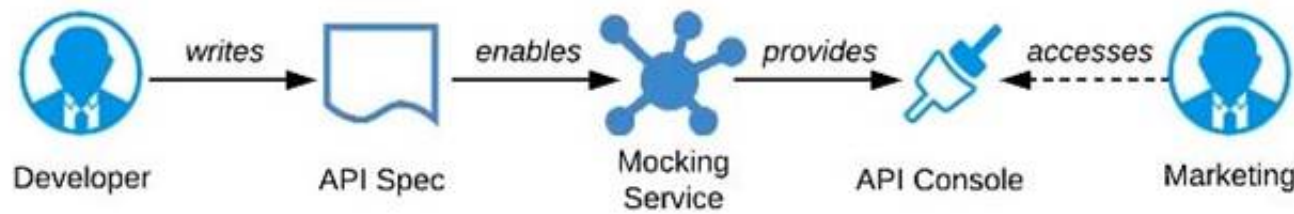


A RAML definition has been proposed for a new Promotions Process API, and has been published to Anypoint Exchange.

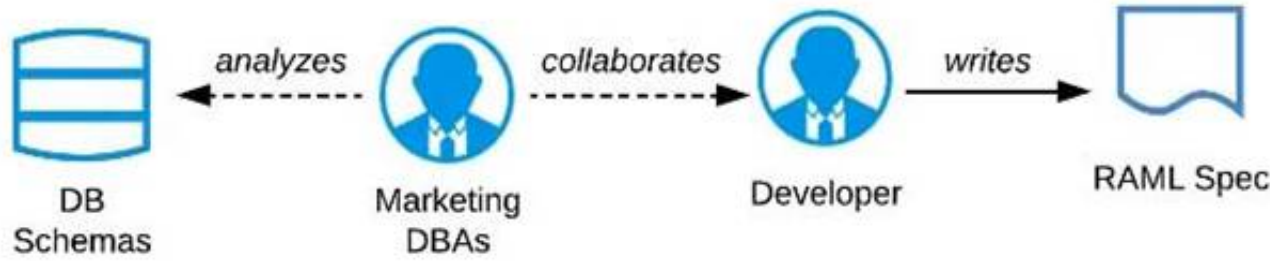
The Marketing Department, who will be an important consumer of the Promotions API, has important requirements and expectations that must be met.

What is the most effective way to use Anypoint Platform features to involve the Marketing Department in this early API design phase?

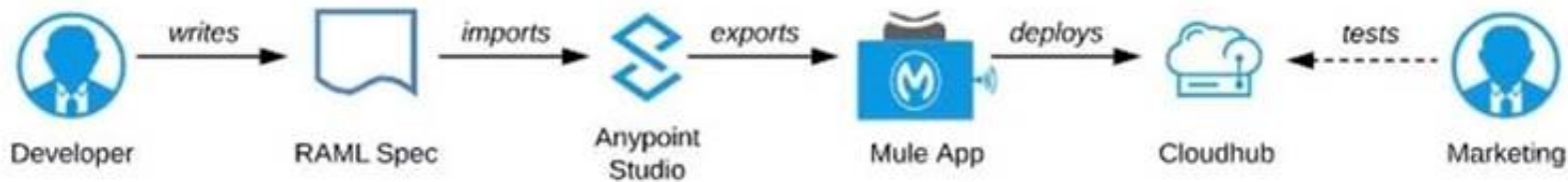
- A) Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console



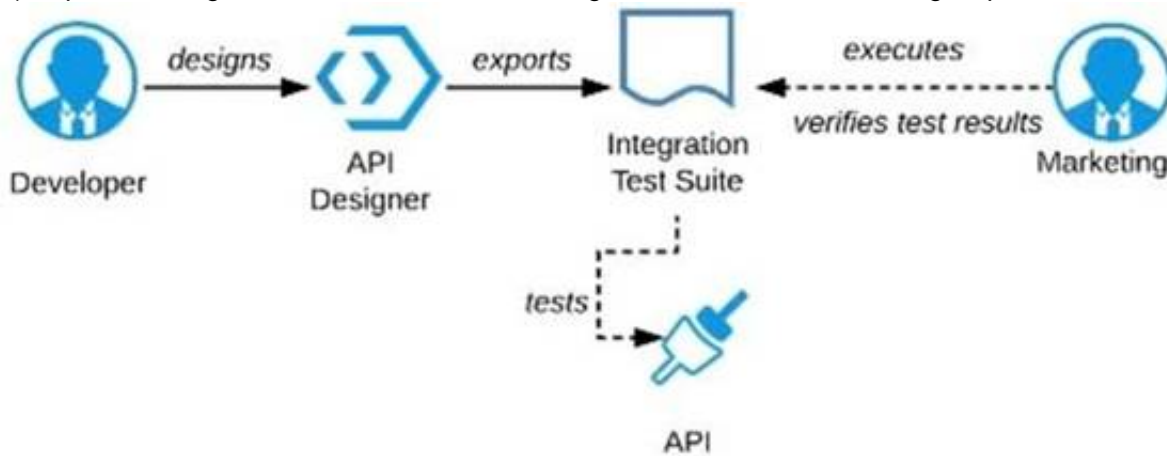
B) Organize a design workshop with the DBAs of the Marketing Department in which the database schema of the Marketing IT systems is translated into RAML



C) Use Anypoint Studio to Implement the API as a Mule application, then deploy that API implementation to CloudHub and ask the Marketing Department to interact with it



D) Export an integration test suite from API designer and have the Marketing Department execute the tests In that suite to ensure they pass



- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

Explanation:

Correct Answer

Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console.

***** As per MuleSoft's IT Operating Model:

>> API consumers need NOT wait until the full API implementation is ready.

>> NO technical test-suites needs to be shared with end users to interact with APIs.

>> Anypoint Platform offers a mocking capability on all the published API specifications to Anypoint Exchange which also will be rich in documentation covering all details of API functionalities and working nature.

>> No needs of arranging days of workshops with end users for feedback.

API consumers can use Anypoint Exchange features on the platform and interact with the API using its mocking feature. The feedback can be shared quickly on the same to incorporate any changes.

NEW QUESTION 42

A REST API is being designed to implement a Mule application.

What standard interface definition language can be used to define REST APIs?

- A. Web Service Definition Language(WSDL)
- B. OpenAPI Specification (OAS)
- C. YAML
- D. AsyncAPI Specification

Answer: B

NEW QUESTION 43

Version 3.0.1 of a REST API implementation represents time values in PST time using ISO 8601 hh:mm:ss format. The API implementation needs to be changed to instead represent time values in CEST time using ISO 8601 hh:mm:ss format. When following the semver.org semantic versioning specification, what version should be assigned to the updated API implementation?

- A. 3.0.2
- B. 4.0.0
- C. 3.1.0
- D. 3.0.1

Answer: B

Explanation:

Correct Answer 4.0.0

***** As per semver.org semantic versioning specification:

Given a version number MAJOR.MINOR.PATCH, increment the:

- MAJOR version when you make incompatible API changes.
- MINOR version when you add functionality in a backwards compatible manner.
- PATCH version when you make backwards compatible bug fixes.

As per the scenario given in the question, the API implementation is completely changing its behavior. Although the format of the time is still being maintained as hh:mm:ss and there is no change in schema w.r.t format, the API will start functioning different after this change as the times are going to come completely different.

Example: Before the change, say, time is going as 09:00:00 representing the PST. Now on, after the change, the same time will go as 18:00:00 as Central European Summer Time is 9 hours ahead of Pacific Time.

>> This may lead to some uncertain behavior on API clients depending on how they are handling the times in the API response. All the API clients need to be informed that the API functionality is going to change and will return in CEST format. So, this considered as a MAJOR change and the version of API for this new change would be 4.0.0

NEW QUESTION 44

An API has been updated in Anypoint exchange by its API producer from version 3.1.1 to 3.2.0 following accepted semantic versioning practices and the changes have been communicated via the APIs public portal. The API endpoint does NOT change in the new version. How should the developer of an API client respond to this change?

- A. The API producer should be requested to run the old version in parallel with the new one
- B. The API producer should be contacted to understand the change to existing functionality
- C. The API client code only needs to be changed if it needs to take advantage of the new features
- D. The API clients need to update the code on their side and need to do full regression

Answer: C

NEW QUESTION 48

An organization makes a strategic decision to move towards an IT operating model that emphasizes consumption of reusable IT assets using modern APIs (as defined by MuleSoft).

What best describes each modern API in relation to this new IT operating model?

- A. Each modern API has its own software development lifecycle, which reduces the need for documentation and automation
- B. Each modern API must be treated like a product and designed for a particular target audience (for instance, mobile app developers)
- C. Each modern API must be easy to consume, so should avoid complex authentication mechanisms such as SAML or JWT D
- D. Each modern API must be REST and HTTP based

Answer: B

Explanation:

Correct Answers

* 1. Each modern API must be treated like a product and designed for a particular target audience (for instance mobile app developers)

Bottom of Form Top of Form

NEW QUESTION 53

The implementation of a Process API must change.

What is a valid approach that minimizes the impact of this change on API clients?

- A. Update the RAML definition of the current Process API and notify API client developers by sending them links to the updated RAML definition
- B. Postpone changes until API consumers acknowledge they are ready to migrate to a new Process API or API version
- C. Implement required changes to the Process API implementation so that whenever possible, the Process API's RAML definition remains unchanged
- D. Implement the Process API changes in a new API implementation, and have the old API implementation return an HTTP status code 301 - Moved Permanently to inform API clients they should be calling the new API implementation

Answer: C

Explanation:

Correct Answer

Implement required changes to the Process API implementation so that, whenever possible, the Process API's RAML definition remains unchanged.

***** Key requirement in the question is:

>> Approach that minimizes the impact of this change on API clients Based on above:

>> Updating the RAML definition would possibly impact the API clients if the changes require any thing mandatory from client side. So, one should try to avoid doing that until really necessary.

>> Implementing the changes as a completely different API and then redirectly the clients with 3xx status code is really upsetting design and heavily impacts the API clients.

>> Organisations and IT cannot simply postpone the changes required until all API consumers acknowledge they are ready to migrate to a new Process API or API version. This is unrealistic and not possible.

The best way to handle the changes always is to implement required changes to the API implementations so that, whenever possible, the API's RAML definition remains unchanged.

NEW QUESTION 55

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 Answer

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 need 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 reusable 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.

NEW QUESTION 57

An organization has created an API-led architecture that uses various API layers to integrate mobile clients with a backend system. The backend system consists of a number of specialized components and can be accessed via a REST API. The process and experience APIs share the same bounded-context model that is different from the backend data model. What additional canonical models, bounded-context models, or anti-corruption layers are best added to this architecture to help process data consumed from the backend system?

- A. Create a bounded-context model for every layer and overlap them when the boundary contexts overlap, letting API developers know about the differences between upstream and downstream data models
- B. Create a canonical model that combines the backend and API-led models to simplify and unify data models, and minimize data transformations.
- C. Create a bounded-context model for the system layer to closely match the backend data model, and add an anti-corruption layer to let the different bounded contexts cooperate across the system and process layers
- D. Create an anti-corruption layer for every API to perform transformation for every data model to match each other, and let data simply travel between APIs to avoid the complexity and overhead of building canonical models

Answer: C

Explanation:

Correct Answer

Create a bounded-context model for the system layer to closely match the backend data model, and add an anti-corruption layer to let the different bounded contexts cooperate across the system and process layers

>> Canonical models are not an option here as the organization has already put in efforts and created bounded-context models for Experience and Process APIs.

>> Anti-corruption layers for ALL APIs is unnecessary and invalid because it is mentioned that experience and process APIs share same bounded-context model.

It is just the System layer APIs that need to choose their approach now.

>> So, having an anti-corruption layer just between the process and system layers will work well. Also to speed up the approach, system APIs can mimic the backend system data model.

NEW QUESTION 61

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual MCPA-Level-1 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the MCPA-Level-1 Product From:

<https://www.2passeasy.com/dumps/MCPA-Level-1/>

Money Back Guarantee

MCPA-Level-1 Practice Exam Features:

- * MCPA-Level-1 Questions and Answers Updated Frequently
- * MCPA-Level-1 Practice Questions Verified by Expert Senior Certified Staff
- * MCPA-Level-1 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * MCPA-Level-1 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year