

MCPA-Level-1 Dumps

MuleSoft Certified Platform Architect - Level 1

<https://www.certleader.com/MCPA-Level-1-dumps.html>



NEW QUESTION 1

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 2

True or False. We should always make sure that the APIs being designed and developed are self-servable even if it needs more man-day effort and resources.

- A. FALSE
- B. TRUE

Answer: B

Explanation:

Correct Answer
TRUE

>> As per MuleSoft proposed IT Operating Model, designing APIs and making sure that they are discoverable and self-servable is VERY VERY IMPORTANT and decides the success of an API and its application network.

NEW QUESTION 3

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.

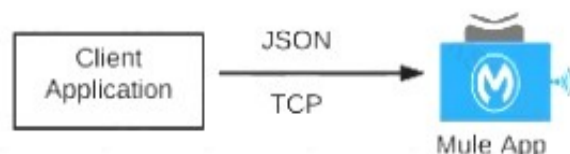
NEW QUESTION 4

What Mule application can have API policies applied by Anypoint Platform to the endpoint exposed by that Mule application?

- A) A Mule application that accepts requests over HTTP/1.x



- B) A Mule application that accepts JSON requests over TCP but is NOT required to provide a response



- C) A Mute application that accepts JSON requests over WebSocket



D) A Mule application that accepts gRPC requests over HTTP/2



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

Answer: A

Explanation:

Correct Answer

Option A

>> Anypoint API Manager and API policies are applicable to all types of HTTP/1.x APIs.

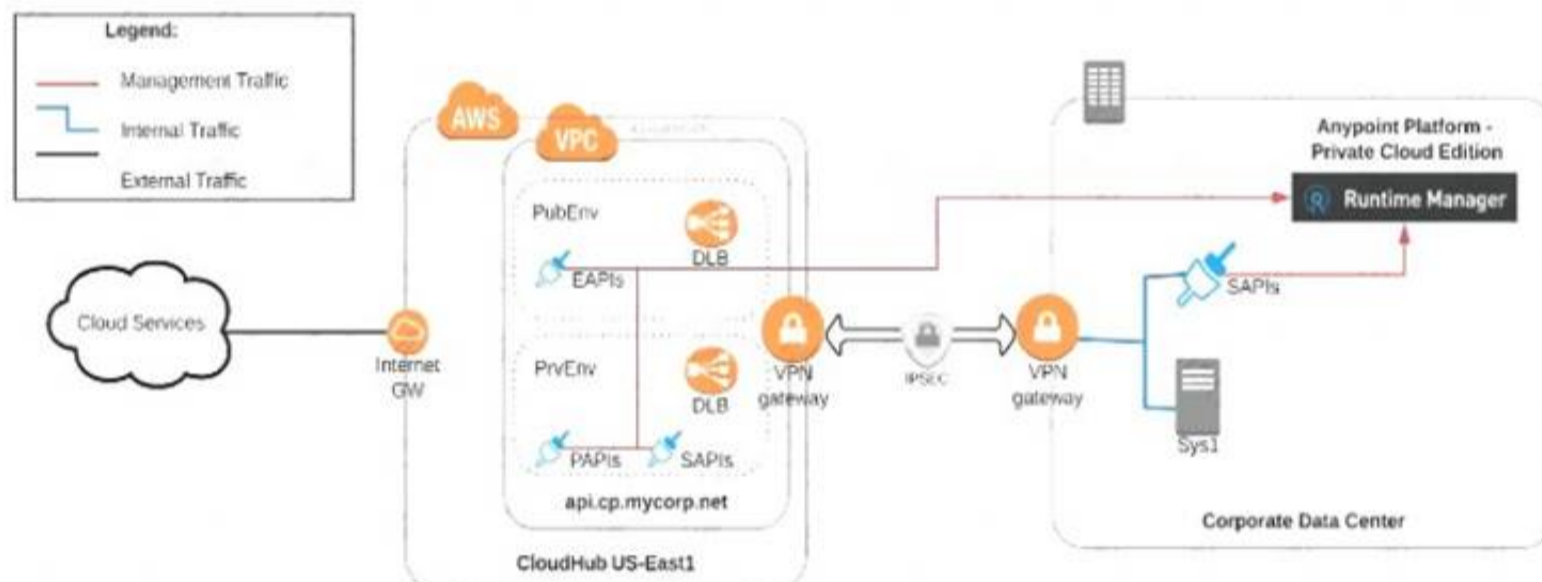
>> They are not applicable to WebSocket APIs, HTTP/2 APIs and gRPC APIs

NEW QUESTION 5

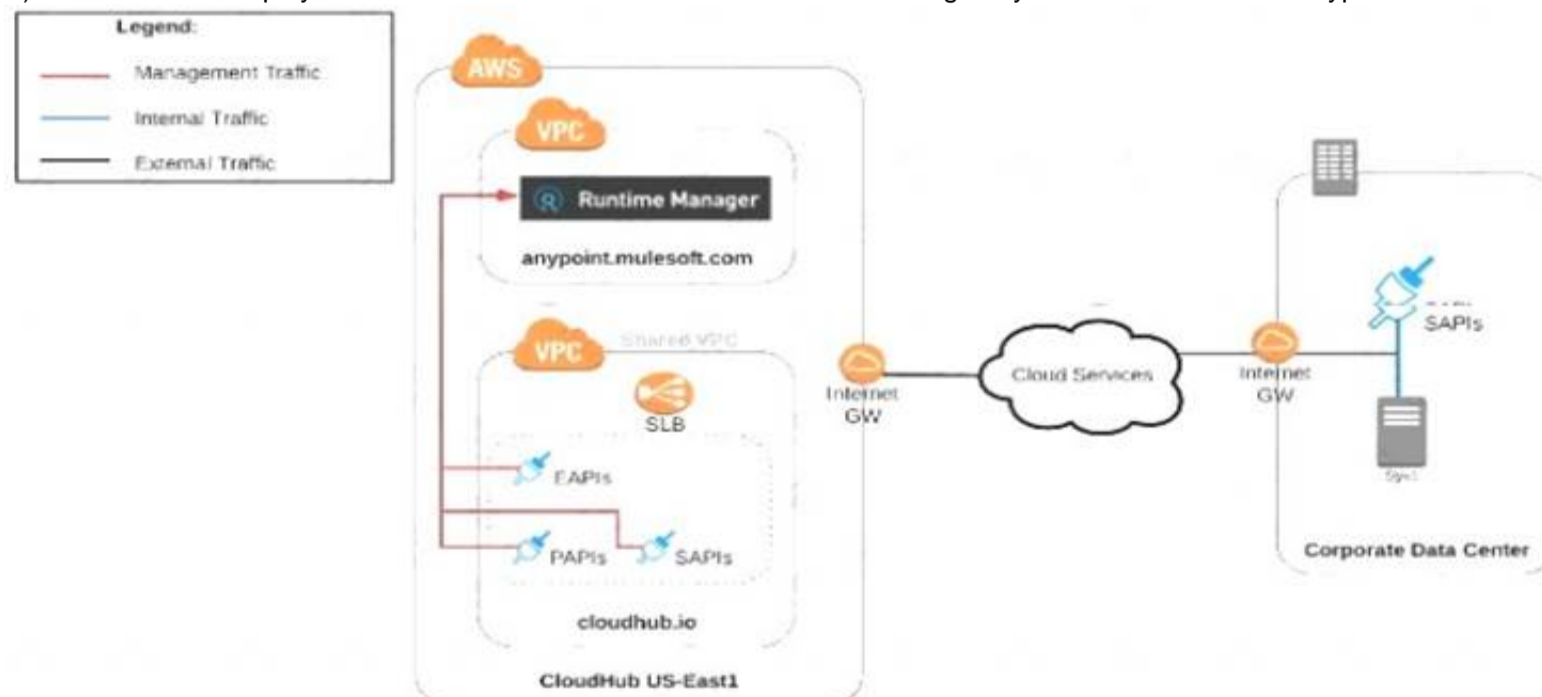
An organization uses various cloud-based SaaS systems and multiple on-premises systems. The on-premises systems are an important part of the organization's application network and can only be accessed from within the organization's intranet.

What is the best way to configure and use Anypoint Platform to support integrations with both the cloud-based SaaS systems and on-premises systems?

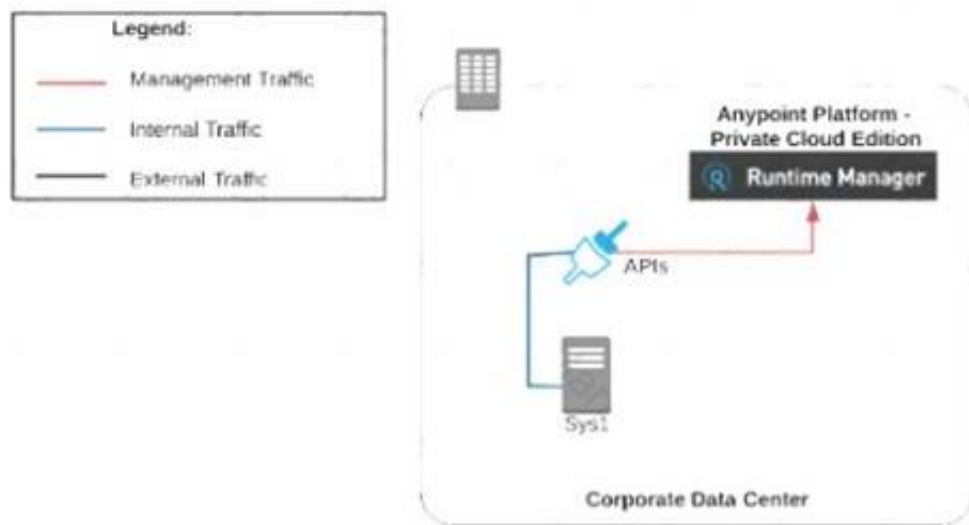
A) Use CloudHub-deployed Mule runtimes in an Anypoint VPC managed by Anypoint Platform Private Cloud Edition control plane



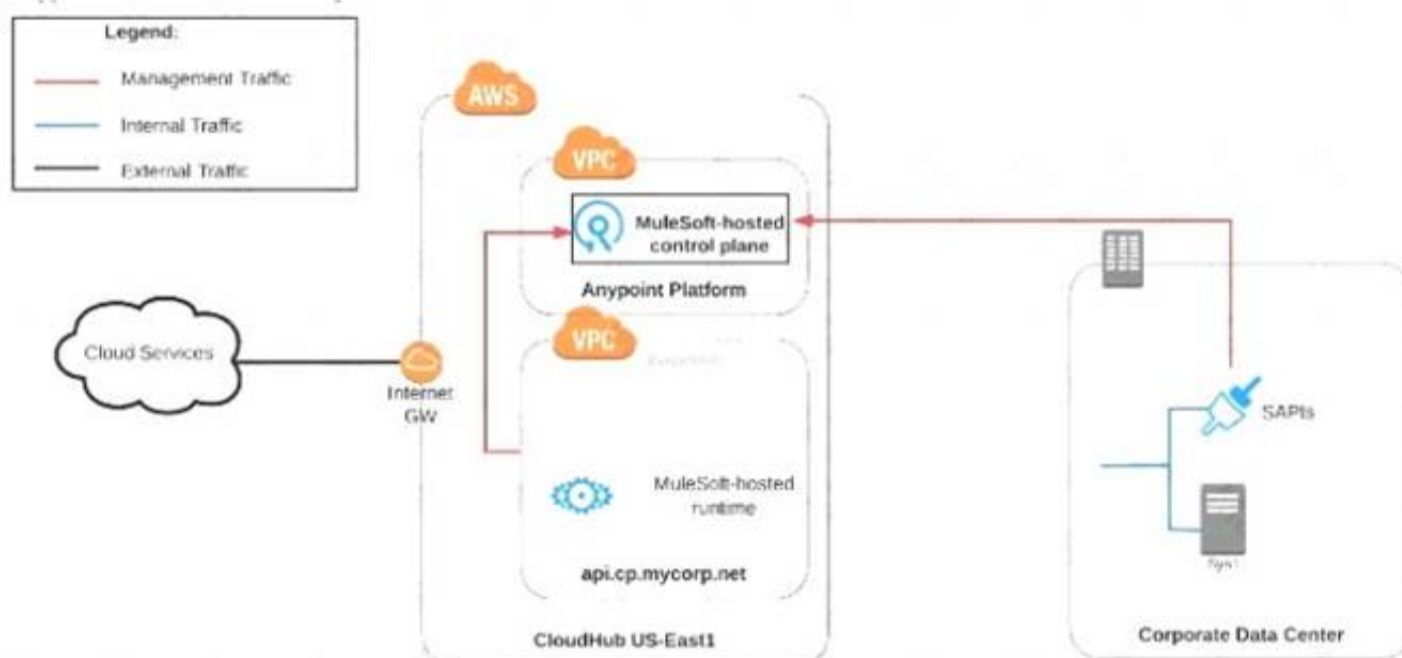
B) Use CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform control plane



C) Use an on-premises installation of Mule runtimes that are completely isolated with NO external network access, managed by the Anypoint Platform Private Cloud Edition control plane



D) Use a combination of Cloud Hub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Anypoint Platform control plane



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

Answer: B

Explanation:

Correct Answer

Use a combination of CloudHub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Platform control plane.

***** Key details to be taken from the given scenario:

>> Organization uses BOTH cloud-based and on-premises systems

>> On-premises systems can only be accessed from within the organization's intranet Let us evaluate the given choices based on above key details:

>> CloudHub-deployed Mule runtimes can ONLY be controlled using MuleSoft-hosted control plane. We CANNOT use Private Cloud Edition's control plane to control CloudHub Mule Runtimes. So, option suggesting this is INVALID

>> Using CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform is completely IRRELEVANT to given scenario and silly choice. So, option suggesting this is INVALID

>> Using an on-premises installation of Mule runtimes that are completely isolated with NO external network access, managed by the Anypoint Platform Private Cloud Edition control plane would work for On-premises integrations. However, with NO external access, integrations cannot be done to SaaS-based apps. Moreover CloudHub-hosted apps are best-fit for integrating with SaaS-based applications. So, option suggesting this is BEST WAY.

The best way to configure and use Anypoint Platform to support these mixed/hybrid integrations is to use a combination of CloudHub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Platform control plane.

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

What is most likely NOT a characteristic of an integration test for a REST API implementation?

- A. The test needs all source and/or target systems configured and accessible
- B. The test runs immediately after the Mule application has been compiled and packaged
- C. The test is triggered by an external HTTP request
- D. The test prepares a known request payload and validates the response payload

Answer: B

Explanation:

Correct Answer

The test runs immediately after the Mule application has been compiled and packaged

>> Integration tests are the last layer of tests we need to add to be fully covered.
>> These tests actually run against Mule running with your full configuration in place and are tested from external source as they work in PROD.
>> These tests exercise the application as a whole with actual transports enabled. So, external systems are affected when these tests run.
So, these tests do NOT run immediately after the Mule application has been compiled and packaged.
FYI... Unit Tests are the one that run immediately after the Mule application has been compiled and packaged.

NEW QUESTION 8

Which of the below, when used together, makes the IT Operational Model effective?

- A. Create reusable assets, Do marketing on the created assets across organization, Arrange time to time LOB reviews to ensure assets are being consumed or not
- B. Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics
- C. Create reusable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

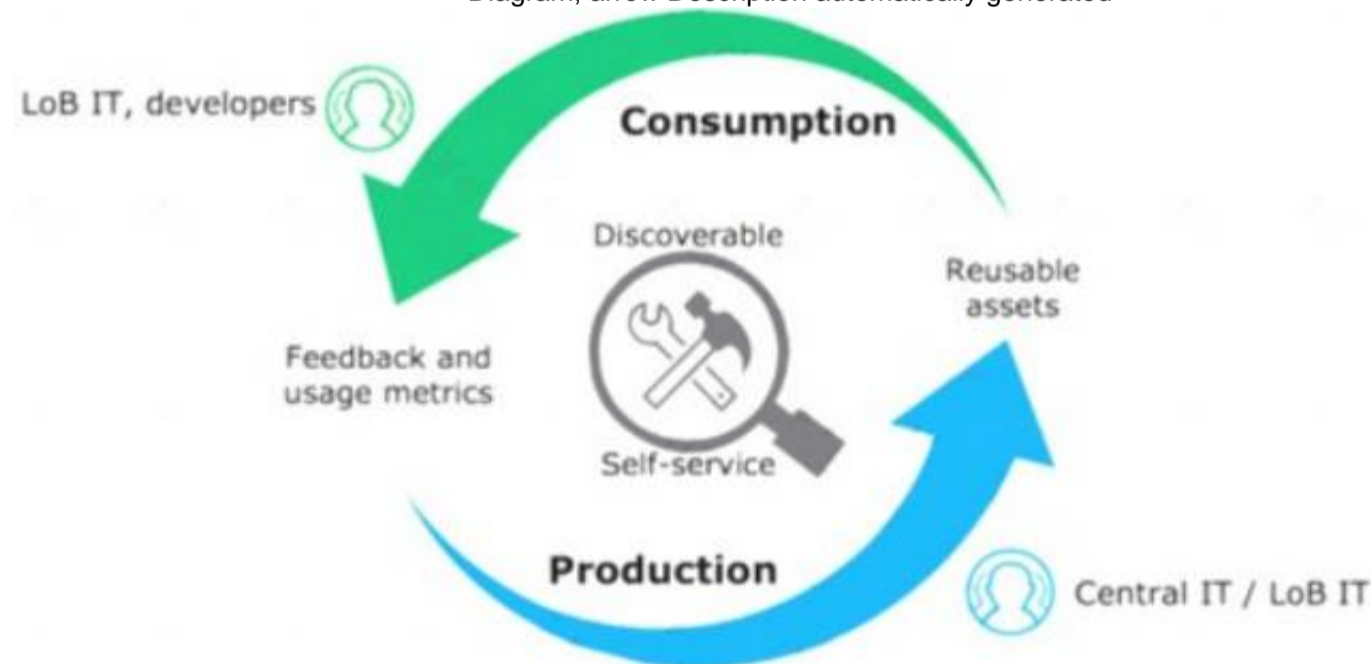
Answer: C

Explanation:

Correct Answer

Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics.

***** Diagram, arrow Description automatically generated



NEW QUESTION 9

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 implementation is deployed to CloudHub.

What conditions can be alerted on using the default Anypoint Platform functionality, where the alert conditions depend on the end-to-end request processing of the API implementation?

- A. When the API is invoked by an unrecognized API client
- B. When a particular API client invokes the API too often within a given time period
- C. When the response time of API invocations exceeds a threshold
- D. When the API receives a very high number of API invocations

Answer: C

Explanation:

Correct Answer

When the response time of API invocations exceeds a threshold

>> Alerts can be setup for all the given options using the default Anypoint Platform functionality

>> However, the question insists on an alert whose conditions depend on the end-to-end request processing of the API implementation.

>> Alert w.r.t "Response Times" is the only one which requires end-to-end request processing of API implementation in order to determine if the threshold is exceeded or not.

NEW QUESTION 10

An organization is implementing a Quote of the Day API that caches today's quote.

What scenario can use the GoudHub Object Store via the Object Store connector to persist the cache's state?

- A. When there are three CloudHub deployments of the API implementation to three separate CloudHub regions that must share the cache state
- B. When there are two CloudHub deployments of the API implementation by two Anypoint Platform business groups to the same CloudHub region that must share the cache state
- C. When there is one deployment of the API implementation to CloudHub and anottV deployment to a customer-hosted Mule runtime that must share the cache state
- D. When there is one CloudHub deployment of the API implementation to three CloudHub workers that must share the cache state

Answer: D

Explanation:

Correct Answer

When there is one CloudHub deployment of the API implementation to three CloudHub workers that must share the cache state.

***** Key details in the scenario:

>> Use the CloudHub Object Store via the Object Store connector Considering above details:

>> CloudHub Object Stores have one-to-one relationship with CloudHub Mule Applications.

>> We CANNOT use an application's CloudHub Object Store to be shared among multiple Mule applications running in different Regions or Business Groups or Customer-hosted Mule Runtimes by using Object Store connector.

>> If it is really necessary and very badly needed, then Anypoint Platform supports a way by allowing access to CloudHub Object Store of another application using Object Store REST API. But NOT using Object Store connector.

So, the only scenario where we can use the CloudHub Object Store via the Object Store connector to persist the cache's state is when there is one CloudHub deployment of the API implementation to multiple CloudHub workers that must share the cache state.

NEW QUESTION 15

An API implementation is updated. When must the RAML definition of the API also be updated?

- A. When the API implementation changes the structure of the request or response messages
- B. When the API implementation changes from interacting with a legacy backend system deployed on-premises to a modern, cloud-based (SaaS) system
- C. When the API implementation is migrated from an older to a newer version of the Mule runtime
- D. When the API implementation is optimized to improve its average response time

Answer: A

Explanation:

Correct Answer

When the API implementation changes the structure of the request or response messages

>> RAML definition usually needs to be touched only when there are changes in the request/response schemas or in any traits on API.

>> It need not be modified for any internal changes in API implementation like performance tuning, backend system migrations etc..

NEW QUESTION 16

What is true about the technology architecture of Anypoint VPCs?

- A. The private IP address range of an Anypoint VPC is automatically chosen by CloudHub
- B. Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network
- C. Each CloudHub environment requires a separate Anypoint VPC
- D. VPC peering can be used to link the underlying AWS VPC to an on-premises (non AWS) private network

Answer: B

Explanation:

Correct Answer

Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network

>> The private IP address range of an Anypoint VPC is NOT automatically chosen by CloudHub. It is chosen by us at the time of creating VPC using the CIDR blocks.

CIDR Block: The size of the Anypoint VPC in Classless Inter-Domain Routing (CIDR) notation.





For example, if you set it to 10.111.0.0/24, the Anypoint VPC is granted 256 IP addresses from 10.111.0.0 to 10.111.0.255.

Ideally, the CIDR Blocks you choose for the Anypoint VPC come from a private IP space, and should not overlap with any other Anypoint VPC's CIDR Blocks, or any CIDR Blocks in use in your corporate network.

← Create VPC

Learn more about VPCs

General Information

Name	vpc1	
Region	US East (N. Virginia)	▼
CIDR Block	10.0.0.0/16	
Environments	Design x	▼
	 Set as default VPC 	
Business Groups	MyBusinessGroup (MyOrg) 	

that each CloudHub environment requires a separate Anypoint VPC. Once an Anypoint VPC is created, we can choose a same VPC by multiple environments. However, it is generally a best and recommended practice to always have separate Anypoint VPCs for Non-Prod and Prod environments.

>> We use Anypoint VPN to link the underlying AWS VPC to an on-premises (non AWS) private network. NOT VPC Peering.

NEW QUESTION 19

When must an API implementation be deployed to an Anypoint VPC?

- A. When the API Implementation must invoke publicly exposed services that are deployed outside of CloudHub in a customer- managed AWS instance
- B. When the API implementation must be accessible within a subnet of a restricted customer-hosted network that does not allow public access
- C. When the API implementation must be deployed to a production AWS VPC using the Mule Maven plugin
- D. When the API Implementation must write to a persistent Object Store

Answer: A

NEW QUESTION 22

An API experiences a high rate of client requests (TPS) with small message payloads. How can usage limits be imposed on the API based on the type of client application?

- A. Use an SLA-based rate limiting policy and assign a client application to a matching SLA tier based on its type
- B. Use a spike control policy that limits the number of requests for each client application type
- C. Use a cross-origin resource sharing (CORS) policy to limit resource sharing between client applications, configured by the client application type
- D. Use a rate limiting policy and a client ID enforcement policy, each configured by the client application type

Answer: A

Explanation:

Correct Answer

Use an SLA-based rate limiting policy and assign a client application to a matching SLA tier based on its type.

>> SLA tiers will come into play whenever any limits to be imposed on APIs based on client type

NEW QUESTION 26

When using CloudHub with the Shared Load Balancer, what is managed EXCLUSIVELY by the API implementation (the Mule application) and NOT by Anypoint Platform?

- A. The assignment of each HTTP request to a particular CloudHub worker
- B. The logging configuration that enables log entries to be visible in Runtime Manager
- C. The SSL certificates used by the API implementation to expose HTTPS endpoints

D. The number of DNS entries allocated to the API implementation

Answer: C

Explanation:

Correct Answer

The SSL certificates used by the API implementation to expose HTTPS endpoints

>> The assignment of each HTTP request to a particular CloudHub worker is taken care by Anypoint Platform itself. We need not manage it explicitly in the API implementation and in fact we CANNOT manage it in the API implementation.

>> The logging configuration that enables log entries to be visible in Runtime Manager is ALWAYS managed in the API implementation and NOT just for SLB. So this is not something we do EXCLUSIVELY when using SLB.

>> We DO NOT manage the number of DNS entries allocated to the API implementation inside the code. Anypoint Platform takes care of this.

It is the SSL certificates used by the API implementation to expose HTTPS endpoints that is to be managed EXCLUSIVELY by the API implementation. Anypoint Platform does NOT do this when using SLBs.

NEW QUESTION 27

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 Answer

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.

NEW QUESTION 29

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 another instance of it on top of same API implementation. So, it does NO GOOD by using clone of 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 30

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

Answer: A

Explanation:

Correct Answer

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.

NEW QUESTION 34

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 37

When designing an upstream API and its implementation, the development team has been advised to NOT set timeouts when invoking a downstream API, because that downstream API has no SLA that can be relied upon. This is the only downstream API dependency of that upstream API. Assume the downstream API runs uninterrupted without crashing. What is the impact of this advice?

- A. An SLA for the upstream API CANNOT be provided
- B. The invocation of the downstream API will run to completion without timing out
- C. A default timeout of 500 ms will automatically be applied by the Mule runtime in which the upstream API implementation executes
- D. A load-dependent timeout of less than 1000 ms will be applied by the Mule runtime in which the downstream API implementation executes

Answer: A

Explanation:

Correct Answer
An SLA for the upstream API CANNOT be provided.

>> First thing first, the default HTTP response timeout for HTTP connector is 10000 ms (10 seconds). NOT 500 ms.
>> Mule runtime does NOT apply any such "load-dependent" timeouts. There is no such behavior currently in Mule.
>> As there is default 10000 ms time out for HTTP connector, we CANNOT always guarantee that the invocation of the downstream API will run to completion without timing out due to its unreliable SLA times. If the response time crosses 10 seconds then the request may time out.
The main impact due to this is that a proper SLA for the upstream API CANNOT be provided.

NEW QUESTION 38

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

Answer: D

Explanation:

Correct Answer
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>

NEW QUESTION 43

What is a key performance indicator (KPI) that measures the success of a typical C4E that is immediately apparent in responses from the Anypoint Platform APIs?

- A. The number of production outage incidents reported in the last 24 hours
- B. The number of API implementations that have a publicly accessible HTTP endpoint and are being managed by Anypoint Platform
- C. The fraction of API implementations deployed manually relative to those deployed using a CI/CD tool
- D. The number of API specifications in RAML or OAS format published to Anypoint Exchange

Answer: D

Explanation:

Correct Answer

The number of API specifications in RAML or OAS format published to Anypoint Exchange

>> The success of C4E always depends on their contribution to the number of reusable assets that they have helped to build and publish to Anypoint Exchange.
>> It is NOT due to any factors w.r.t # of outages, Manual vs CI/CD deployments or Publicly accessible HTTP endpoints
>> Anypoint Platform APIs helps us to quickly run and get the number of published RAML/OAS assets to Anypoint Exchange. This clearly depicts how successful a C4E team is based on number of returned assets in the response.

NEW QUESTION 47

What CANNOT be effectively enforced using an API policy in Anypoint Platform?

- A. Guarding against Denial of Service attacks
- B. Maintaining tamper-proof credentials between APIs
- C. Logging HTTP requests and responses
- D. Backend system overloading

Answer: A

Explanation:

Correct Answer

Guarding against Denial of Service attacks

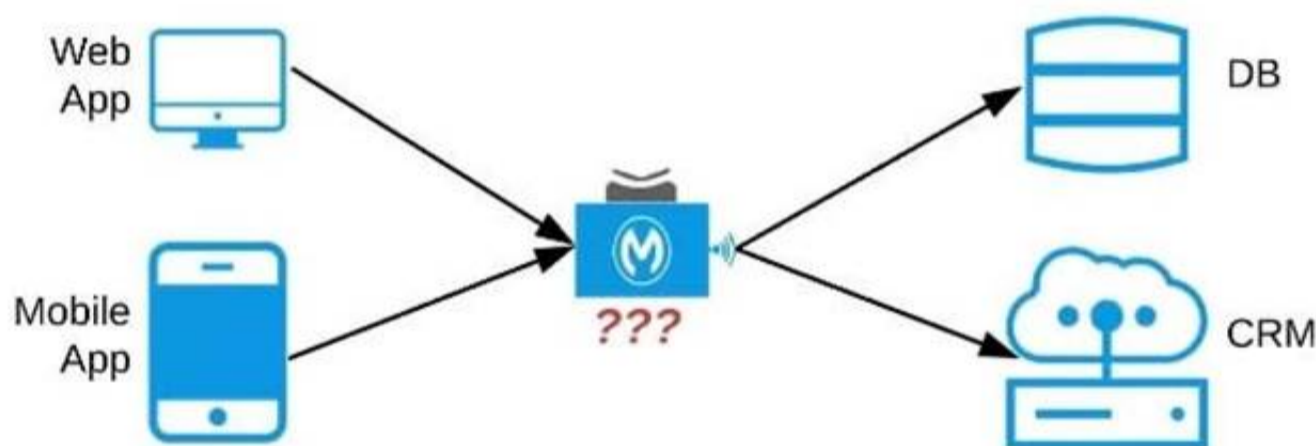
>> Backend system overloading can be handled by enforcing "Spike Control Policy"
>> Logging HTTP requests and responses can be done by enforcing "Message Logging Policy"
>> Credentials can be tamper-proofed using "Security" and "Compliance" Policies
However, unfortunately, there is no proper way currently on Anypoint Platform to guard against DOS attacks.

NEW QUESTION 48

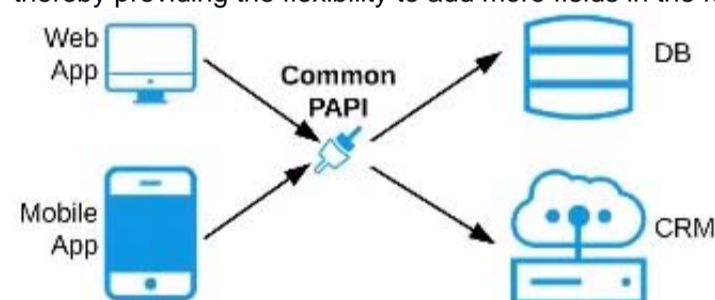
Refer to the exhibit. An organization needs to enable access to their customer data from both a mobile app and a web application, which each need access to common fields as well as certain unique fields.

The data is available partially in a database and partially in a 3rd-party CRM system.

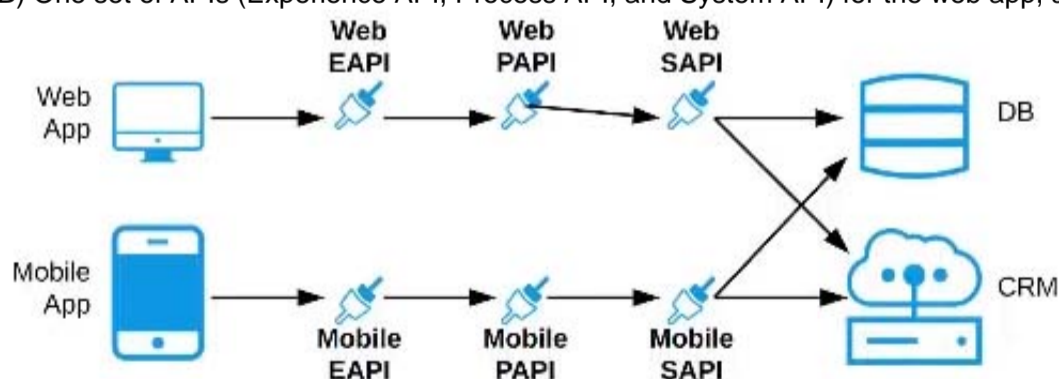
What APIs should be created to best fit these design requirements?



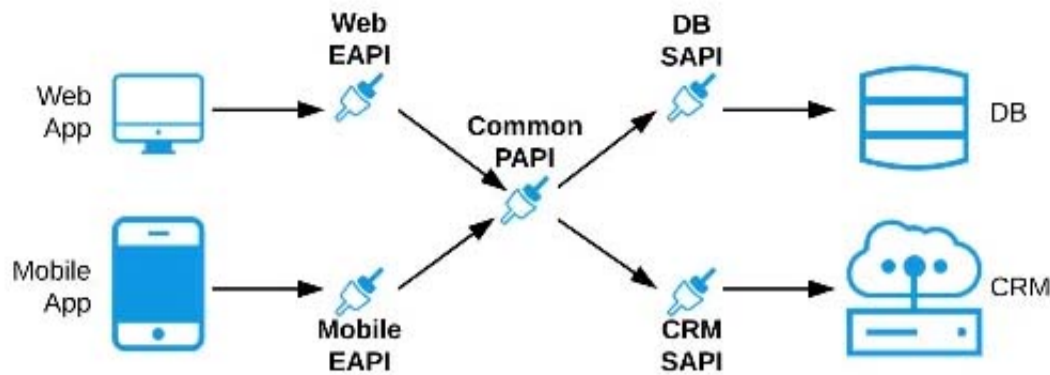
A) A Process API that contains the data required by both the web and mobile apps, allowing these applications to invoke it directly and access the data they need thereby providing the flexibility to add more fields in the future without needing API changes



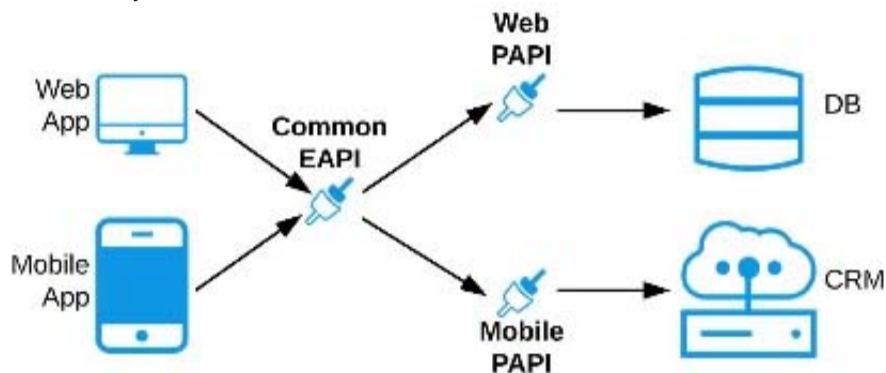
B) One set of APIs (Experience API, Process API, and System API) for the web app, and another set for the mobile app



C) Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system



D) A common Experience API used by both the web and mobile apps, but separate Process APIs for the web and mobile apps that interact with the database and the CRM System



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

Answer: C

Explanation:

Correct Answer

Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system

***** As per MuleSoft's API-led connectivity:

- >> Experience APIs should be built as per each consumer needs and their experience.
- >> Process APIs should contain all the orchestration logic to achieve the business functionality.
- >> System APIs should be built for each backend system to unlock their data.

NEW QUESTION 51

What is a key requirement when using an external Identity Provider for Client Management in Anypoint Platform?

- A. Single sign-on is required to sign in to Anypoint Platform
- B. The application network must include System APIs that interact with the Identity Provider
- C. To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider
- D. APIs managed by Anypoint Platform must be protected by SAML 2.0 policies

Answer: C

Explanation:

<https://www.folkstalk.com/2019/11/mulesoft-integration-and-platform.html>

Correct Answer

To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider

- >> It is NOT necessary that single sign-on is required to sign in to Anypoint Platform because we are using an external Identity Provider for Client Management
- >> It is NOT necessary that all APIs managed by Anypoint Platform must be protected by SAML 2.0 policies because we are using an external Identity Provider for Client Management
- >> Not TRUE that the application network must include System APIs that interact with the Identity Provider because we are using an external Identity Provider for Client Management
- Only TRUE statement in the given options is - "To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider"

References:

<https://docs.mulesoft.com/api-manager/2.x/external-oauth-2.0-token-validation-policy> <https://blogs.mulesoft.com/dev/api-dev/api-security-ways-to-authenticate-and-authorize/>

NEW QUESTION 52

Once an API Implementation is ready and the API is registered on API Manager, who should request the access to the API on Anypoint Exchange?

- A. None
- B. Both
- C. API Client
- D. API Consumer

Answer: D

Explanation:

Correct Answer

API Consumer

>> API clients are piece of code or programs that use the client credentials of API consumer but does not directly interact with Anypoint Exchange to get the access

>> API consumer is the one who should get registered and request access to API and then API client needs to use those client credentials to hit the APIs

So, API consumer is the one who needs to request access on the API from Anypoint Exchange

NEW QUESTION 56

A System API is designed to retrieve data from a backend system that has scalability challenges. What API policy can best safeguard the backend system?

A. IPwhitelist

B. SLA-based rate limiting

C. Auth 2 token enforcement

D. Client ID enforcement

Answer: B

Explanation:

Correct Answer

SLA-based rate limiting

>> Client Id enforcement policy is a "Compliance" related NFR and does not help in maintaining the "Quality of Service (QoS)". It CANNOT and NOT meant for protecting the backend systems from scalability challenges.

>> IP Whitelisting and OAuth 2.0 token enforcement are "Security" related NFRs and again does not help in maintaining the "Quality of Service (QoS)". They CANNOT and are NOT meant for protecting the backend systems from scalability challenges.

Rate Limiting, Rate Limiting-SLA, Throttling, Spike Control are the policies that are "Quality of Service (QOS)" related NFRs and are meant to help in protecting the backend systems from getting overloaded.

<https://dzone.com/articles/how-to-secure-apis>

NEW QUESTION 59

An organization has several APIs that accept JSON data over HTTP POST. The APIs are all publicly available and are associated with several mobile applications and web applications.

The organization does NOT want to use any authentication or compliance policies for these APIs, but at the same time, is worried that some bad actor could send payloads that could somehow compromise the applications or servers running the API implementations.

What out-of-the-box Anypoint Platform policy can address exposure to this threat?

A. Shut out bad actors by using HTTPS mutual authentication for all API invocations

B. Apply an IP blacklist policy to all APIs; the blacklist will Include all bad actors

C. Apply a Header injection and removal policy that detects the malicious data before it is used

D. Apply a JSON threat protection policy to all APIs to detect potential threat vectors

Answer: D

Explanation:

Correct Answer

Apply a JSON threat protection policy to all APIs to detect potential threat vectors

>> Usually, if the APIs are designed and developed for specific consumers (known consumers/customers) then we would IP Whitelist the same to ensure that traffic only comes from them.

>> However, as this scenario states that the APIs are publicly available and being used by so many mobile and web applications, it is NOT possible to identify and blacklist all possible bad actors.

>> So, JSON threat protection policy is the best chance to prevent any bad JSON payloads from such bad actors.

NEW QUESTION 61

A company has started to create an application network and is now planning to implement a Center for Enablement (C4E) organizational model. What key factor would lead the company to decide upon a federated rather than a centralized C4E?

A. When there are a large number of existing common assets shared by development teams

B. When various teams responsible for creating APIs are new to integration and hence need extensive training

C. When development is already organized into several independent initiatives or groups

D. When the majority of the applications in the application network are cloud based

Answer: C

Explanation:

Correct Answer

When development is already organized into several independent initiatives or groups

>> It would require lot of process effort in an organization to have a single C4E team coordinating with multiple already organized development teams which are into several independent initiatives. A single C4E works well with different teams having at least a common initiative. So, in this scenario, federated C4E works well instead of centralized C4E.

NEW QUESTION 65

A company requires Mule applications deployed to CloudHub to be isolated between non-production and production environments. This is so Mule applications deployed to non-production environments can only access backend systems running in their customer-hosted non-production environment, and so Mule applications deployed to production environments can only access backend systems running in their customer-hosted production environment. How does MuleSoft

recommend modifying Mule applications, configuring environments, or changing infrastructure to support this type of per-environment isolation between Mule applications and backend systems?

- A. Modify properties of Mule applications deployed to the production Anypoint Platform environments to prevent access from non-production Mule applications
- B. Configure firewall rules in the infrastructure inside each customer-hosted environment so that only IP addresses from the corresponding Anypoint Platform environments are allowed to communicate with corresponding backend systems
- C. Create non-production and production environments in different Anypoint Platform business groups
- D. Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments

Answer: D

Explanation:

Correct Answer

Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments.

>> Creating different Business Groups does NOT make any difference w.r.t accessing the non-prod and prod customer-hosted environments. Still they will be accessing from both Business Groups unless process network restrictions are put in place.

>> We need to modify or couple the Mule Application Implementations with the environment. In fact, we should never implements application coupled with environments by binding them in the properties. Only basic things like endpoint URL etc should be bundled in properties but not environment level access restrictions.

>> IP addresses on CloudHub are dynamic until unless a special static addresses are assigned. So it is not possible to setup firewall rules in customer-hosted infrastrcture. More over, even if static IP addresses are assigned, there could be 100s of applications running on cloudhub and setting up rules for all of them would be a hectic task, non-maintainable and definitely got a good practice.

>> Thbeest practice recommended

by MulesoftIn(fact any cloud provider), is to have your Anypoint VPCs

seperated for Prod and Non-Prod and perform the VPC peering or VPN tunneling for these Anypoint VPCs to respective Prod and Non-Prod customer-hosted environment networks.

NEW QUESTION 69

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 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.

NEW QUESTION 72

What is true about where an API policy is defined in Anypoint Platform and how it is then applied to API instances?

- A. The API policy Is defined In Runtime Manager as part of the API deployment to a Mule runtime, and then ONLY applied to the specific API Instance
- B. The API policy Is defined In API Manager for a specific API Instance, and then ONLY applied to the specific API instance
- C. The API policy Is defined in API Manager and then automatically applied to ALL API instances
- D. The API policy is defined in API Manager, and then applied to ALL API instances in the specified environment

Answer: B

Explanation:

Correct Answer

The API policy is defined in API Manager for a specific API instance, and then ONLY applied to the specific API instance.

>> Once our API specifications are ready and published to Exchange, we need to visit API Manager and register an API instance for each API.

>> API Manager is the place where management of API aspects takes place like addressing NFRs by enforcing policies on them.

>> We can create multiple instances for a same API and manage them differently for different purposes.

>> One instance can have a set of API policies applied and another instance of same API can have different set of policies applied for some other purpose.

>> These APIs and their instances are defined PER environment basis. So, one need to manage them seperately in each environment.

>> We can ensure that same configuration of API instances (SLAs, Policies etc..) gets promoted when promoting to higher environments using platform feature. But this is optional only. Still one can change them per environment basis if they have to.

>> Runtime Manager is the place to manage API Implementations and their Mule Runtimes but NOT APIs itself. Though API policies gets executed in Mule

Runtimes, We CANNOT enforce API policies in Runtime Manager. We would need to do that via API Manager only for a cherry picked instance in an environment. So, based on these facts, right statement in the given choices is - "The API policy is defined in API Manager for a specific API instance, and then ONLY applied to the specific API instance".

NEW QUESTION 76

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

Answer: B

Explanation:

Correct Answer

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.

NEW QUESTION 78

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your MCPA-Level-1 Exam with Our Prep Materials Via below:

<https://www.certleader.com/MCPA-Level-1-dumps.html>