Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

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

NEW QUESTION 1
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 2
A retail company is using an Order API to accept new orders. The Order API uses a JMS queue to submit orders to a backend order management service. The normal load for orders is being handled using two (2) CloudHub workers, each configured with 0.2 vCore. The CPU load of each CloudHub worker normally runs well below 70%. However, several times during the year the Order API gets four times (4x) the average number of orders. This causes the CloudHub worker CPU load to exceed 90% and the order submission time to exceed 30 seconds. The cause, however, is NOT the backend order management service, which still responds fast enough to meet the response SLA for the Order API. What is the MOST resource-efficient way to configure the Mule application's CloudHub deployment to help the company cope with this performance challenge?

A. Permanently increase the size of each of the two (2) CloudHub workers by at least four times (4x) to one(1) vCore
B. Use a vertical CloudHub autoscaling policy that triggers on CPU utilization greater than 70%
C. Permanently increase the number of CloudHub workers by four times (4x) to eight (8) CloudHub workers
D. Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70%

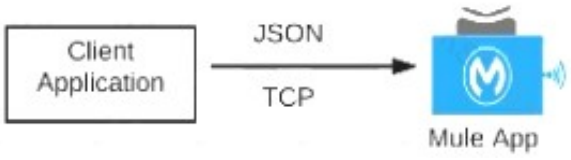
Answer: D

Explanation:

Correct Answer
Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70%
****
The scenario in the question is very clearly stating that the usual traffic in the year is pretty well handled by the existing worker configuration with CPU running well below 70%. The problem occurs only "sometimes" occasionally when there is spike in the number of orders coming in.
So, based on above, We neither need to permanently increase the size of each worker nor need to permanently increase the number of workers. This is unnecessary as other than those "occasional" times the resources are idle and wasted.
We have two options left now. Either to use horizontal Cloudhub autoscaling policy to automatically increase the number of workers or to use vertical Cloudhub autoscaling policy to automatically increase the vCore size of each worker.
Here, we need to take two things into consideration:
* 1. CPU
* 2. Order Submission Rate to JMS Queue
>> From CPU perspective, both the options (horizontal and vertical scaling) solves the issue. Both helps to bring down the usage below 90%.
>> However, If we go with Vertical Scaling, then from Order Submission Rate perspective, as the application is still being load balanced with two workers only, there may not be much improvement in the incoming request processing rate and order submission rate to JMS queue. The throughput would be same as before. Only CPU utilization comes down.
>> But, if we go with Horizontal Scaling, it will spawn new workers and adds extra hand to increase the throughput as more workers are being load balanced now. This way we can address both CPU and Order Submission rate.
Hence, Horizontal CloudHub Autoscaling policy is the right and best answer.

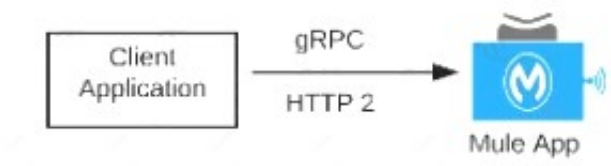
NEW QUESTION 3
What Mule application can have API policies applied by
Anypoint Platform to the endpoint exposed by that Mule application?
A) A Mule application that accepts requests over HTTP/1.x

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



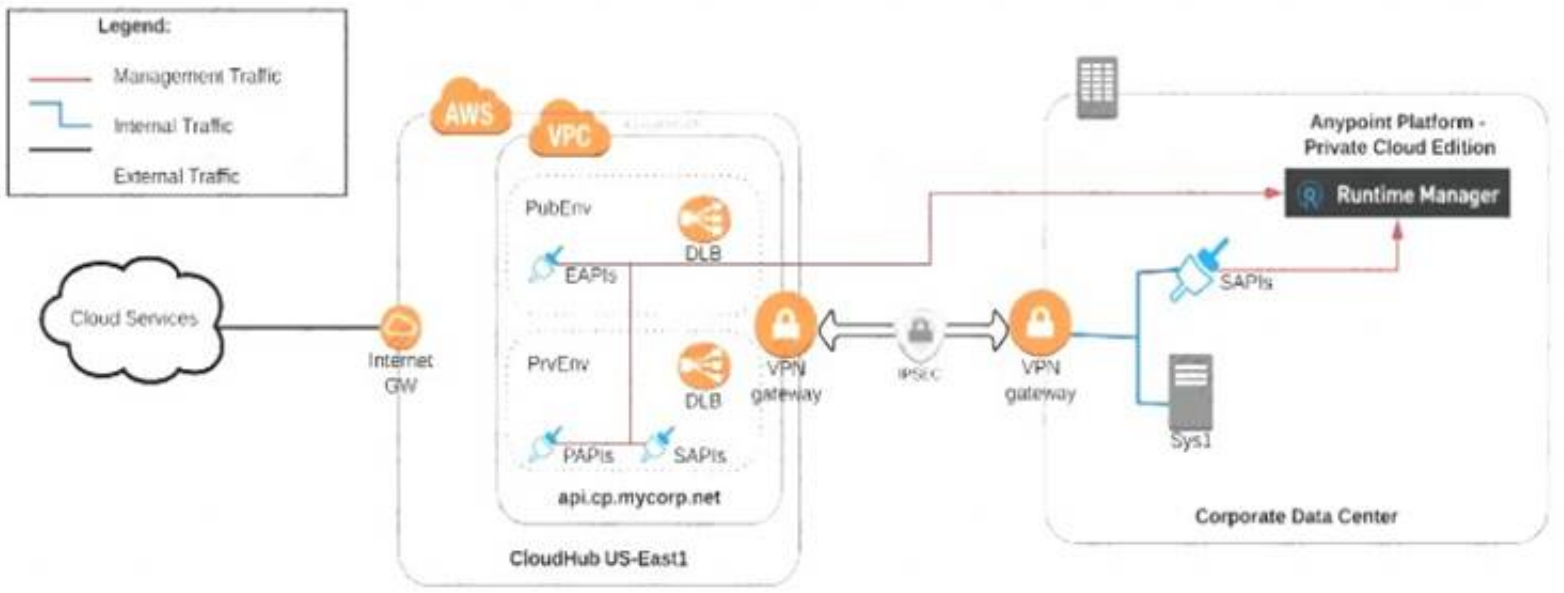
A. Option A
B. Option B
C. Option C
D. Option D

Answer: A

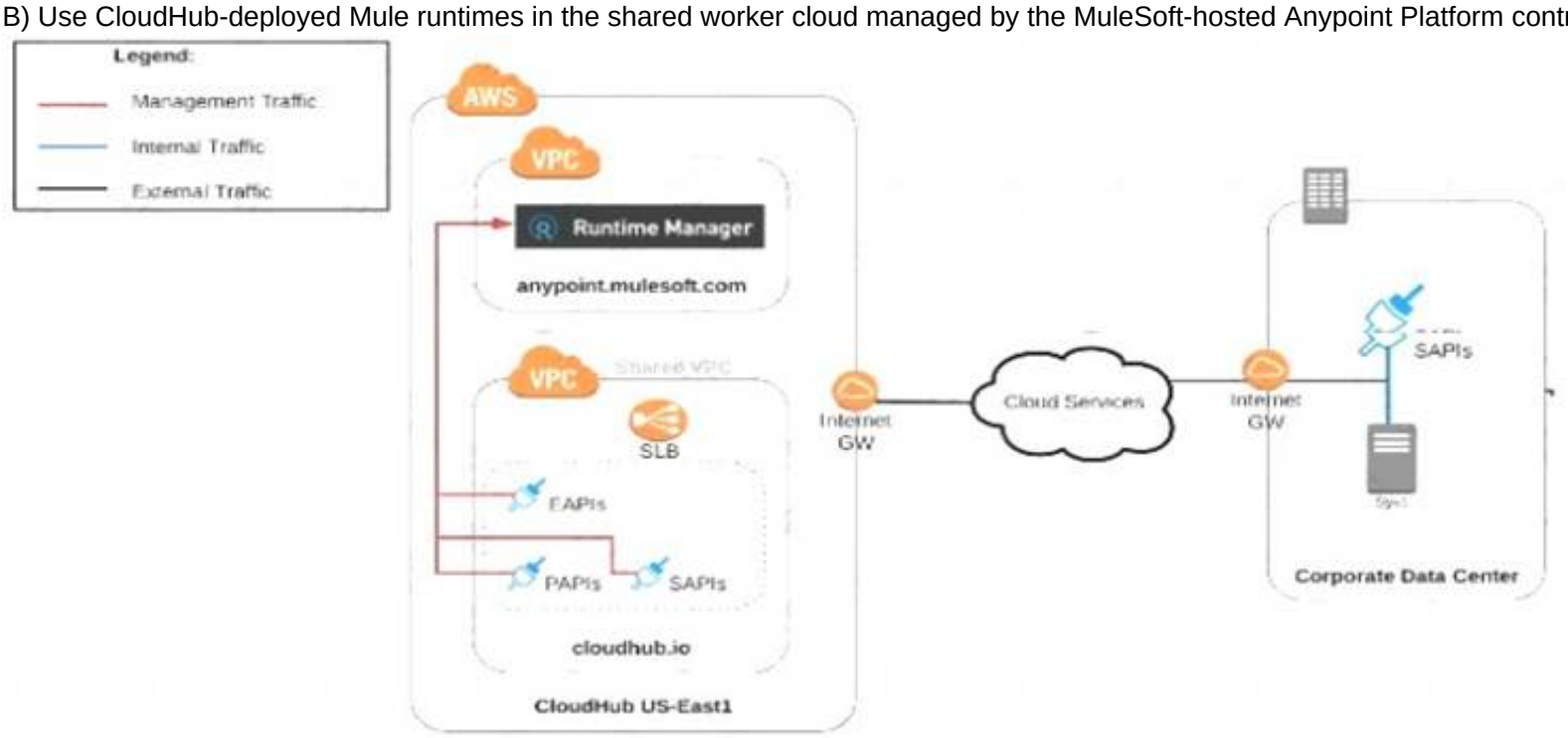
Explanation:

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

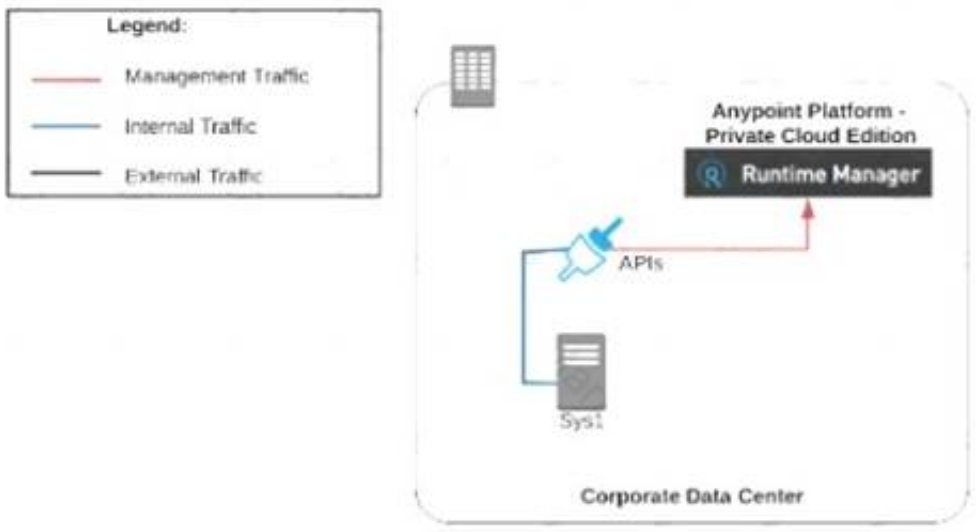
NEW QUESTION 4
An organization uses various cloud-based SaaS systems and multiple on-premises systems. The on-premises systems are an important part of the organization's application network and can only be accessed from within the organization's intranet.
What is the best way to configure and use Anypoint 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



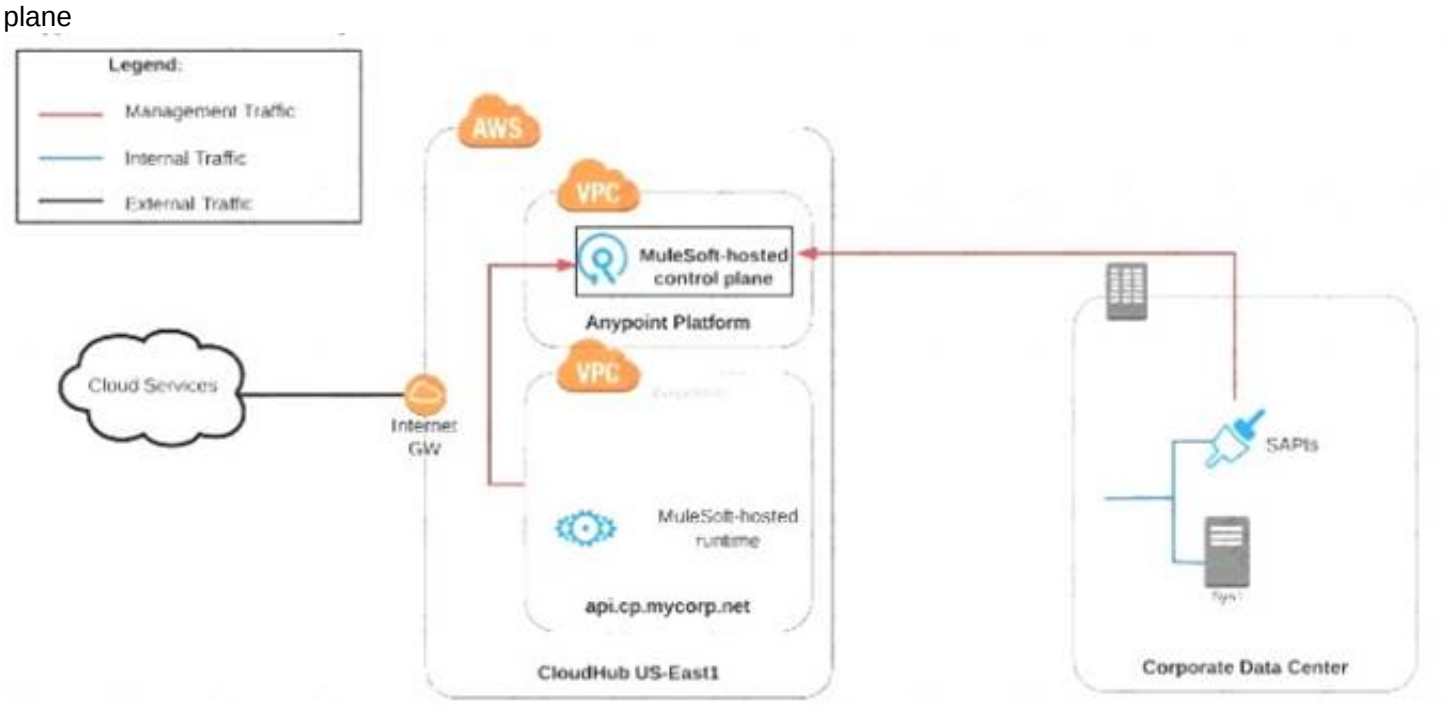
B) Use CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform control plane



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

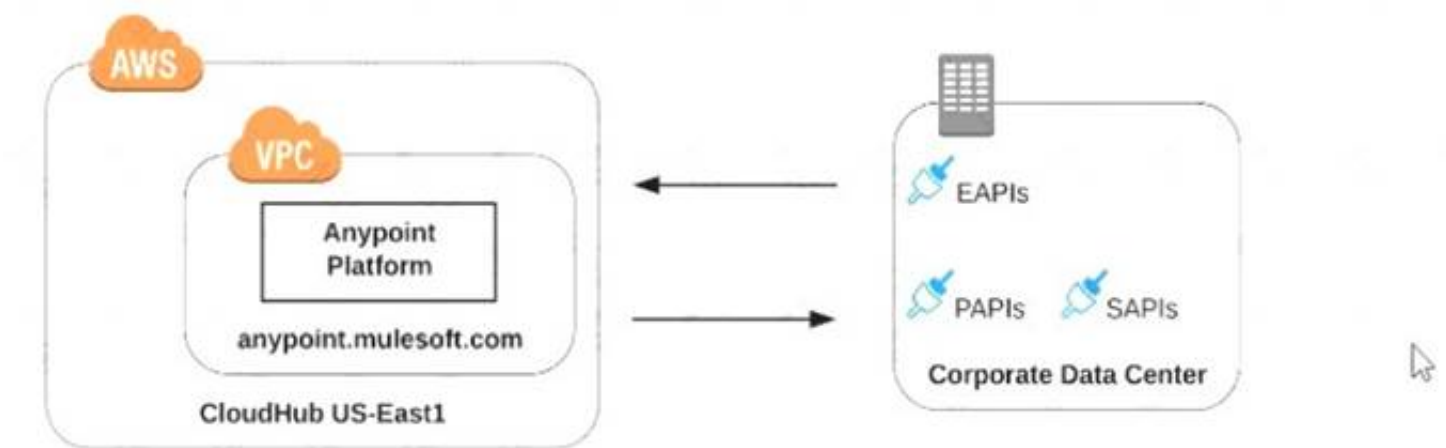


A. Option A
B. Option B
C. Option C
D. Option D

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 5
Refer to the exhibit.



what is true when using customer-hosted Mule runtimes with the MuleSoft-hosted Anypoint Platform control plane (hybrid deployment)?

A. Anypoint Runtime Manager initiates a network connection to a Mule runtime in order to deploy Mule applications
B. The MuleSoft-hosted Shared Load Balancer can be used to load balance API invocations to the Mule runtimes
C. API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane
D. Anypoint Runtime Manager automatically ensures HA in the control plane by creating a new Mule runtime instance in case of a node failure

Answer: C

Explanation:

Correct Answer
API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane.
****
>> We CANNOT use Shared Load balancer to load balance APIs on customer hosted runtimes

- Load balancing

Load balancing is not provided for hybrid deployments. You can manage load balancing with the tools connected to your on-premises resources.

>> For Hybrid deployment models, the on-premises are first connected to Runtime Manager usingRuntime Manager agent. So, the connection is initiated first from On-premises to Runtime Manager. Then all control can be done from Runtime Manager.
>> Anypoint Runtime Manager CANNOT ensure automatic HA. Clusters/Server Groups etc should be configured before hand.
Only TRUE statement in the given choices is, API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane. There are several references below to justify this statement.
References:
https://docs.mulesoft.com/runtime-manager/deployment-strategies#hybrid-deployments https://help.mulesoft.com/s/article/On-Premise-Runtimes-Disconnected-From-US-Control-Plane-June-18th-201 https://help.mulesoft.com/s/article/Runtime-Manager-cannot-manage-On-Prem-Applications-and-Servers-from-https://help.mulesoft.com/s/article/On-premise-Runtimes-Appear-Disconnected-in-Runtime-Manager-May-29th

On-Premise Runtimes Disconnected From US Control Plane - June 18th 2018

Jun 19, 2018 - RCA

Content

Impacted Platforms	Impacted Duration
Anypoint Runtime Manager / On-Prem Runtimes	During this time frame, on-prem runtimes appeared disconnected from the US Anypoint Control Plane: June 18, 2018 10:35 AM PST to June 18, 2018 11:12 AM PST

Incident Description

On-premises applications weren't able to connect to Anypoint Runtime Manager during the length of the incident, which made on-premises runtimes to threw errors in their logs because they received network disconnect messages from the control plane. Other than generating the log as mentioned above entries, on-premises runtimes and applications were not impacted.

==========

Runtime Manager cannot manage On-Prem Applications and Servers from US Control Plane - June 25th 2019

Jul 3, 2019 - RCA

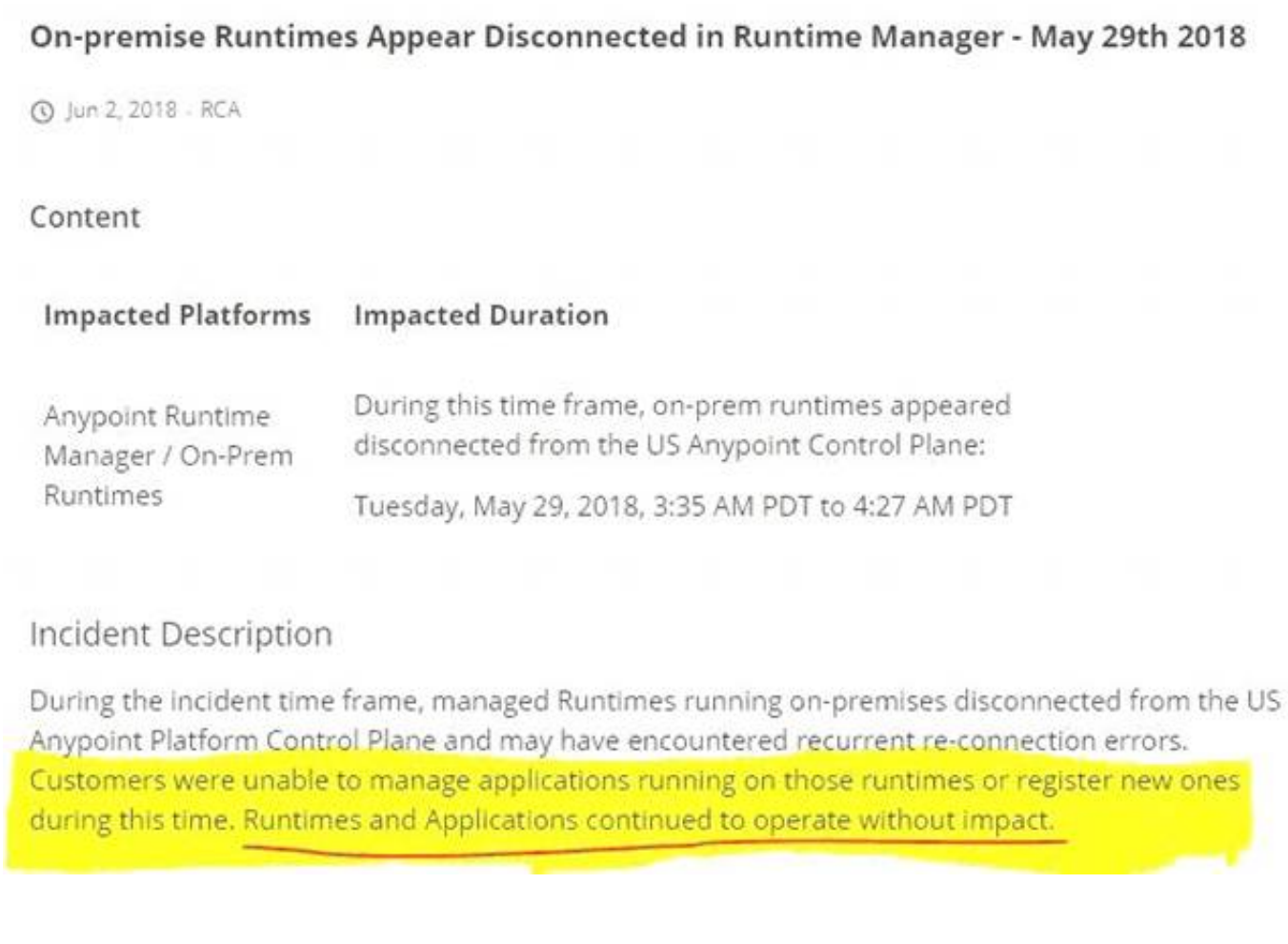
Content

Incident Summary

Between 2:51 p.m. PT June 25th and 12:41 a.m. PT June 26th, customers were not able to manage their On-Prem applications and servers. The availability of running applications and runtimes were not impacted.

Impacted Platforms	Impact Duration
US-Prod	9 hours and 50 minutes

==========


On-premise Runtimes Appear Disconnected in Runtime Manager - May 29th 2018

Jun 2, 2018 - RCA

Content

Impacted Platforms	Impacted Duration
Anypoint Runtime Manager / On-Prem Runtimes	During this time frame, on-prem runtimes appeared disconnected from the US Anypoint Control Plane: Tuesday, May 29, 2018, 3:35 AM PDT to 4:27 AM PDT

Incident Description

During the incident time frame, managed Runtimes running on-premises disconnected from the US Anypoint Platform Control Plane and may have encountered recurrent re-connection errors. Customers were unable to manage applications running on those runtimes or register new ones during this time. Runtimes and Applications continued to operate without impact.

NEW QUESTION 6
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 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 condition requires using a CloudHub Dedicated Load Balancer?

A. When cross-region load balancing is required between separate deployments of the same Mule application
B. When custom DNS names are required for API implementations deployed to customer-hosted Mule runtimes
C. When API invocations across multiple CloudHub workers must be load balanced
D. When server-side load-balanced TLS mutual authentication is required between API implementationsand API clients

Answer: D

Explanation:
Correct Answer
When server-side load-balanced TLS mutual authentication is required between API implementations and API clients
****
Fact/ Memory Tip: Although there are many benefits of CloudHub Dedicated Load balancer, TWO important things that should come to ones mind for considering

it are:
>> Having URL endpoints with Custom DNS names on CloudHub deployed apps
>> Configuring custom certificates for both HTTPS and Two-way (Mutual) authentication. Coming to the options provided for this question:
>> We CANNOT use DLB to perform cross-region load balancing between separate deployments of the same Mule application.
>> We can have mapping rules to have more than one DLB URL pointing to same Mule app. But vicevera (More than one Mule app having same DLB URL) is NOT POSSIBLE
>> It is true that DLB helps to setup custom DNS names for Cloudhub deployed Mule apps but NOT true for apps deployed to Customer-hosted Mule Runtimes.
>> It is true to that we can load balance API invocations across multiple CloudHub workers using DLB but it is NOT A MUST. We can achieve the same (load balancing) using SLB (Shared Load Balancer) too. We DO NOT necessarily require DLB for achieve it.
So the only right option that fits the scenario and requires us to use DLB is when TLS mutual authentication is required between API implementations and API clients.

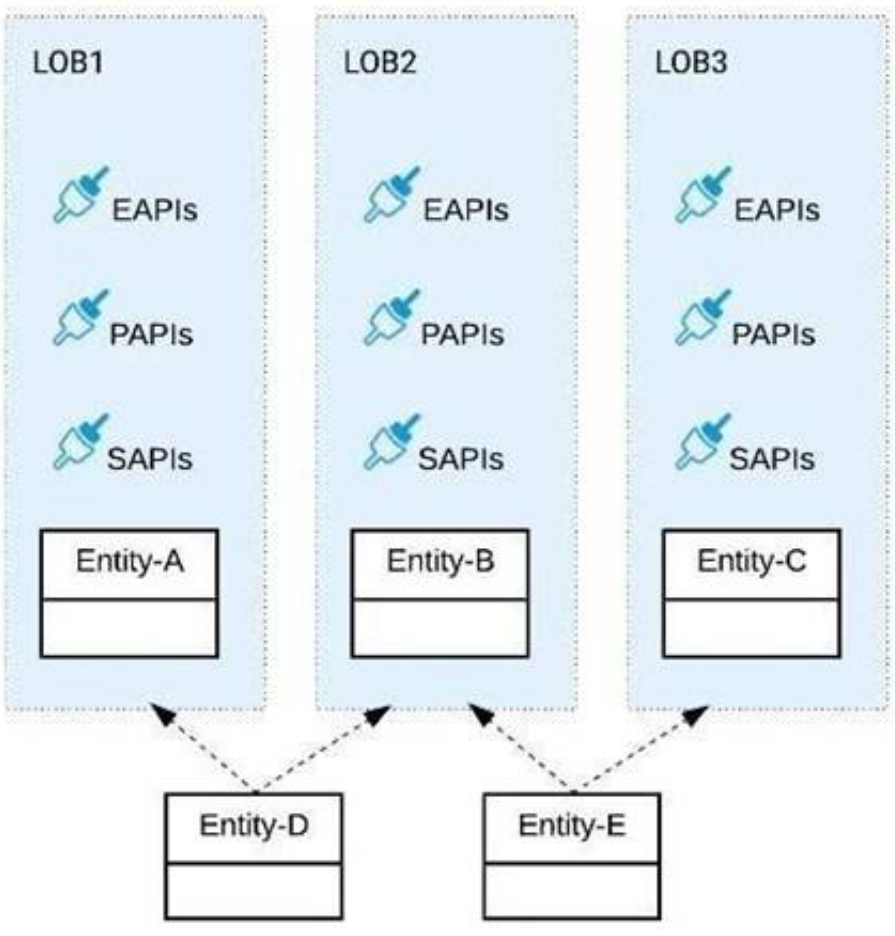
NEW QUESTION 9
Say, there is a legacy CRM system called CRM-Z which is offering below functions:
* 1. Customer creation
* 2. Amend details of an existing customer
* 3. Retrieve details of a customer
* 4. Suspend a customer

A. Implement a system API named customerManagement which has all the functionalities wrapped in it asvarious operations/resources
B. Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns
C. Implement different system APIs named createCustomerInCRMZ, amendCustomerInCRMZ, retrieveCustomerFromCRMZ and suspendCustomerInCRMZ as they are modular and has seperation of concerns

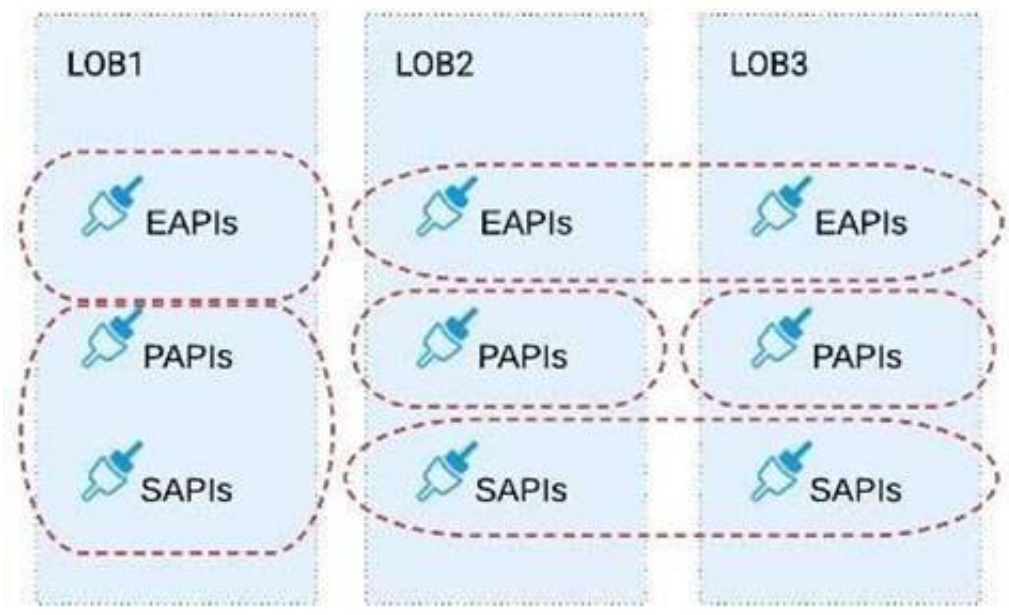
Answer: B

Explanation:
Correct Answer
Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns
****
>> It is quite normal to have a single API and different Verb + Resource combinations. However, this fits well for an Experience API or a Process API but not a best architecture style for System APIs. So, option with just one customerManagement API is not the best choice here.
>> The option with APIs in createCustomerInCRMZ format is next close choice w.r.t modularization and less maintenance but the naming of APIs is directly coupled with the legacy system. A better foreseen approach would be to name your APIs by abstracting the backend system names as it allows seamless replacement/migration of any backend system anytime. So, this is not the correct choice too.
>> createCustomer, amendCustomer, retrieveCustomer and suspendCustomer is the right approach and is the best fit compared to other options as they are both modular and same time got the names decoupled from backend system and it has covered all requirements a System API needs.

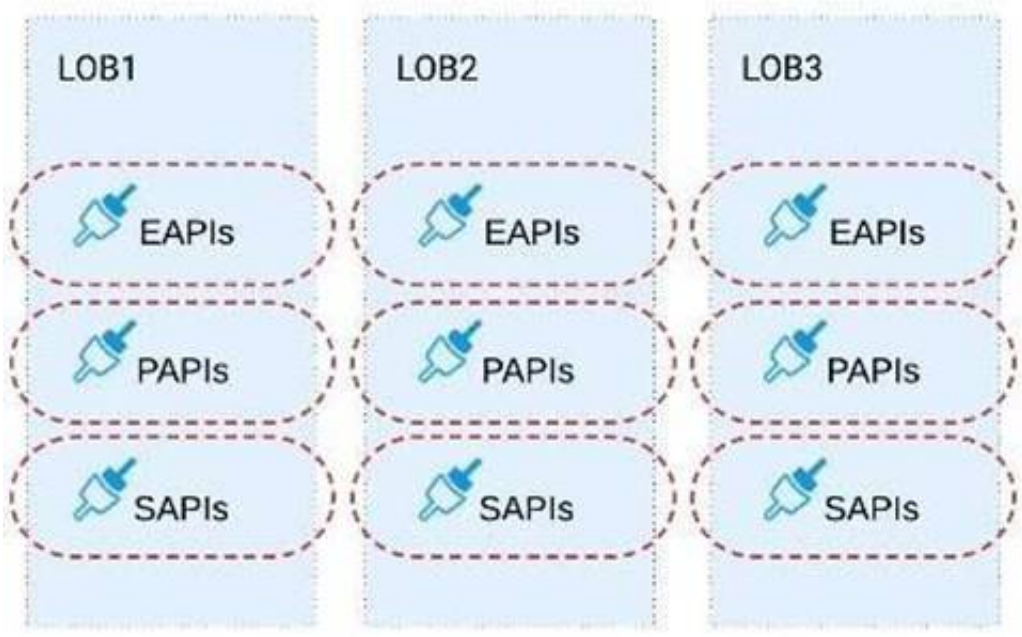
NEW QUESTION 10
Refer to the exhibit.



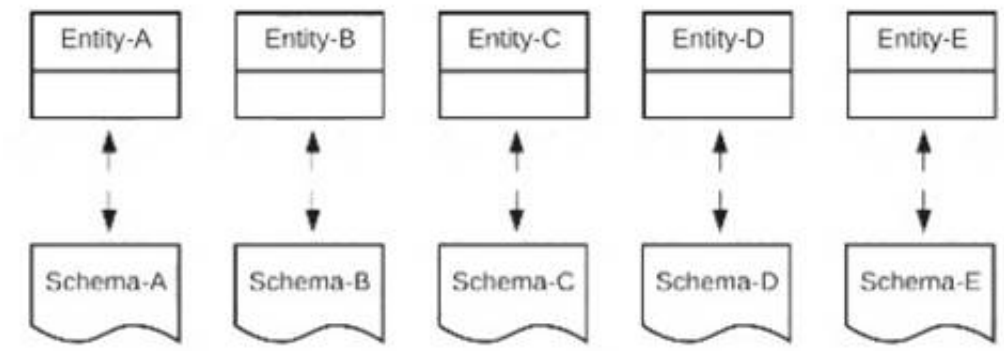
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications.
These processes are owned by separate (siloed) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB has one development team and their own budget
In this organizational context, what is the most effective approach to choose the API data models for the APIs that will implement these business processes with minimal redundancy of the data models?
A) Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities



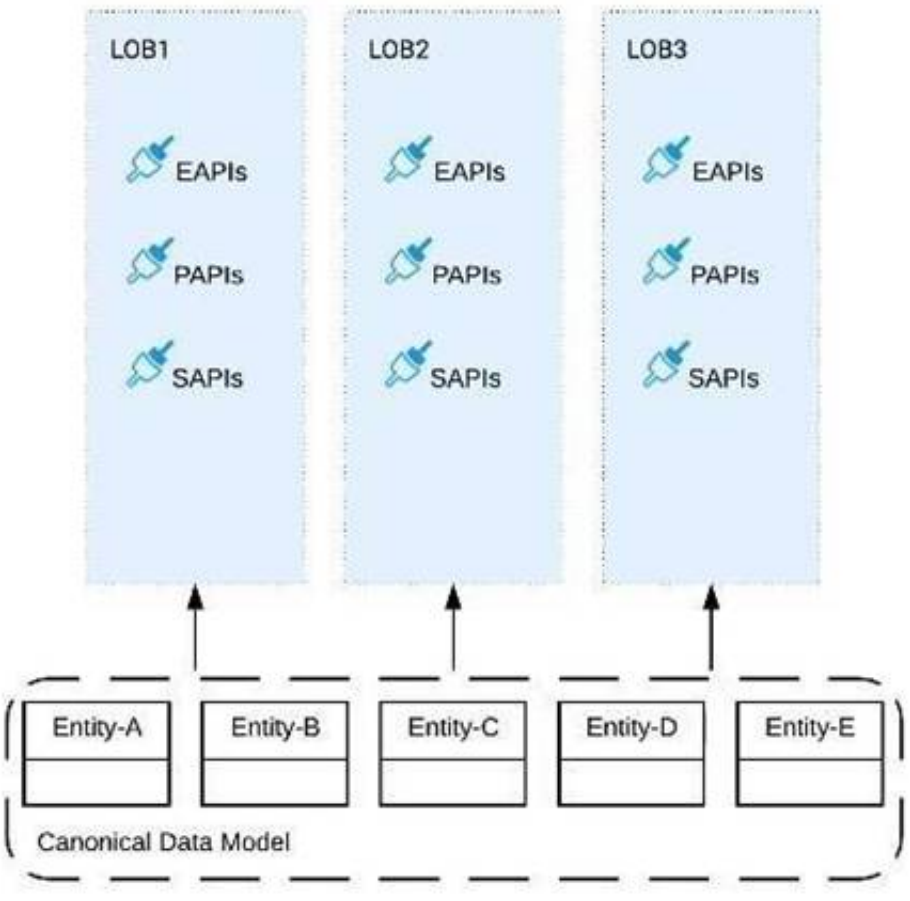
B) Build distinct data models for each API to follow established micro-services and Agile API-centric practices



C) Build all API data models using XML schema to drive consistency and reuse across the organization



D) Build one centralized Canonical Data Model (Enterprise Data Model) that unifies all the data types from all three business processes, ensuring the data model is consistent and non-redundant



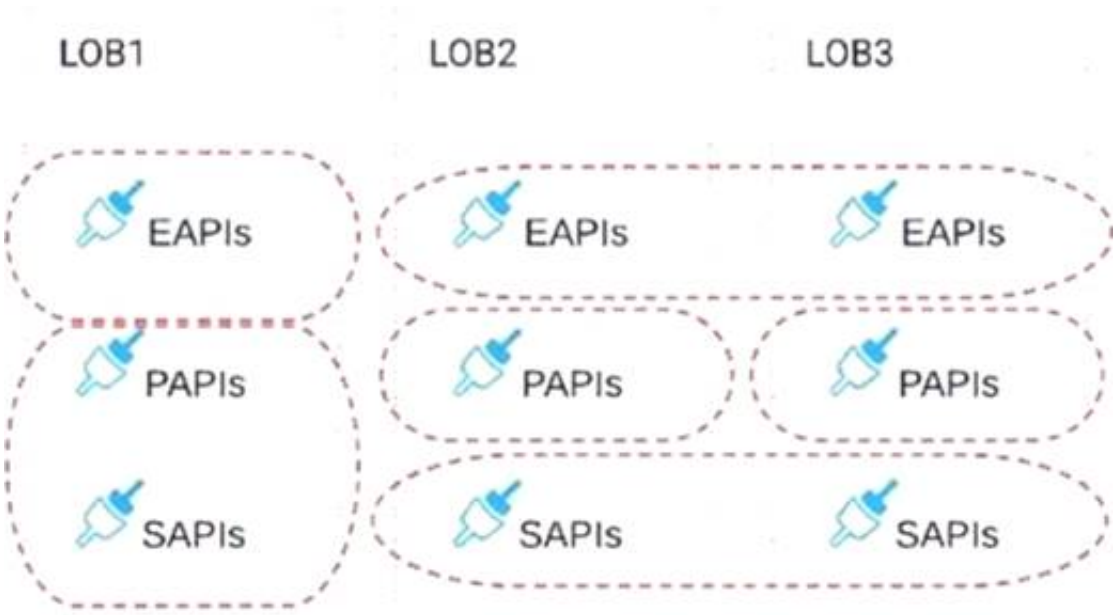
A. Option A

B. Option B
C. Option C
D. Option D

Answer: A

Explanation:

Correct Answer
Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.
****
>> The options w.r.t building API data models using XML schema/ Agile API-centric practices are irrelevant to the scenario given in the question. So these two are INVALID.
>> Building EDM (Enterprise Data Model) is not feasible or right fit for this scenario as the teams and LOBs work in silo and they all have different initiatives, budget etc.. Building EDM needs intensive coordination among all the team which evidently seems not possible in this scenario.
So, the right fit for this scenario is to build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.



NEW QUESTION 10
A code-centric API documentation environment should allow API consumers to investigate and execute API client source code that demonstrates invoking one or more APIs as part of representative scenarios.
What is the most effective way to provide this type of code-centric API documentation environment using Anypoint Platform?

A. Enable mocking services for each of the relevant APIs and expose them via their Anypoint Exchange entry
B. Ensure the APIs are well documented through their Anypoint Exchange entries and API Consoles and share these pages with all API consumers
C. Create API Notebooks and include them in the relevant Anypoint Exchange entries
D. Make relevant APIs discoverable via an Anypoint Exchange entry

Answer: C

Explanation:

Correct Answer
Create API Notebooks and Include them in the relevant Anypoint exchange entries
****
>> API Notebooks are the one on Anypoint Platform that enable us to provide code-centric API documentation

NEW QUESTION 15
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 17
A Mule application exposes an HTTPS endpoint and is deployed to the CloudHub Shared Worker Cloud. All traffic to that Mule application must stay inside the AWS VPC.
To what TCP port do API invocations to that Mule application need to be sent?

A. 443
B. 8081
C. 8091
D. 8082

Answer: D

Explanation:
Correct Answer 8082
****
>> 8091 and 8092 ports are to be used when keeping your HTTP and HTTPS app private to the LOCAL VPC respectively.
>> Above TWO ports are not for Shared AWS VPC/ Shared Worker Cloud.
>> 8081 is to be used when exposing your HTTP endpoint app to the internet through Shared LB
>> 8082 is to be used when exposing your HTTPS endpoint app to the internet through Shared LB So, API invocations should be sent to port 8082 when calling this HTTPS based app.
References:
https://docs.mulesoft.com/runtime-manager/cloudhub-networking-guide https://help.mulesoft.com/s/article/Configure-Cloudhub-Application-to-Send-a-HTTPS-Request-Directly-to-An
https://help.mulesoft.com/s/question/0D52T00004mXXULSA4/multiple-http-listerners-on-cloudhub-one-with-p

NEW QUESTION 19
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 21
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 deployedon-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

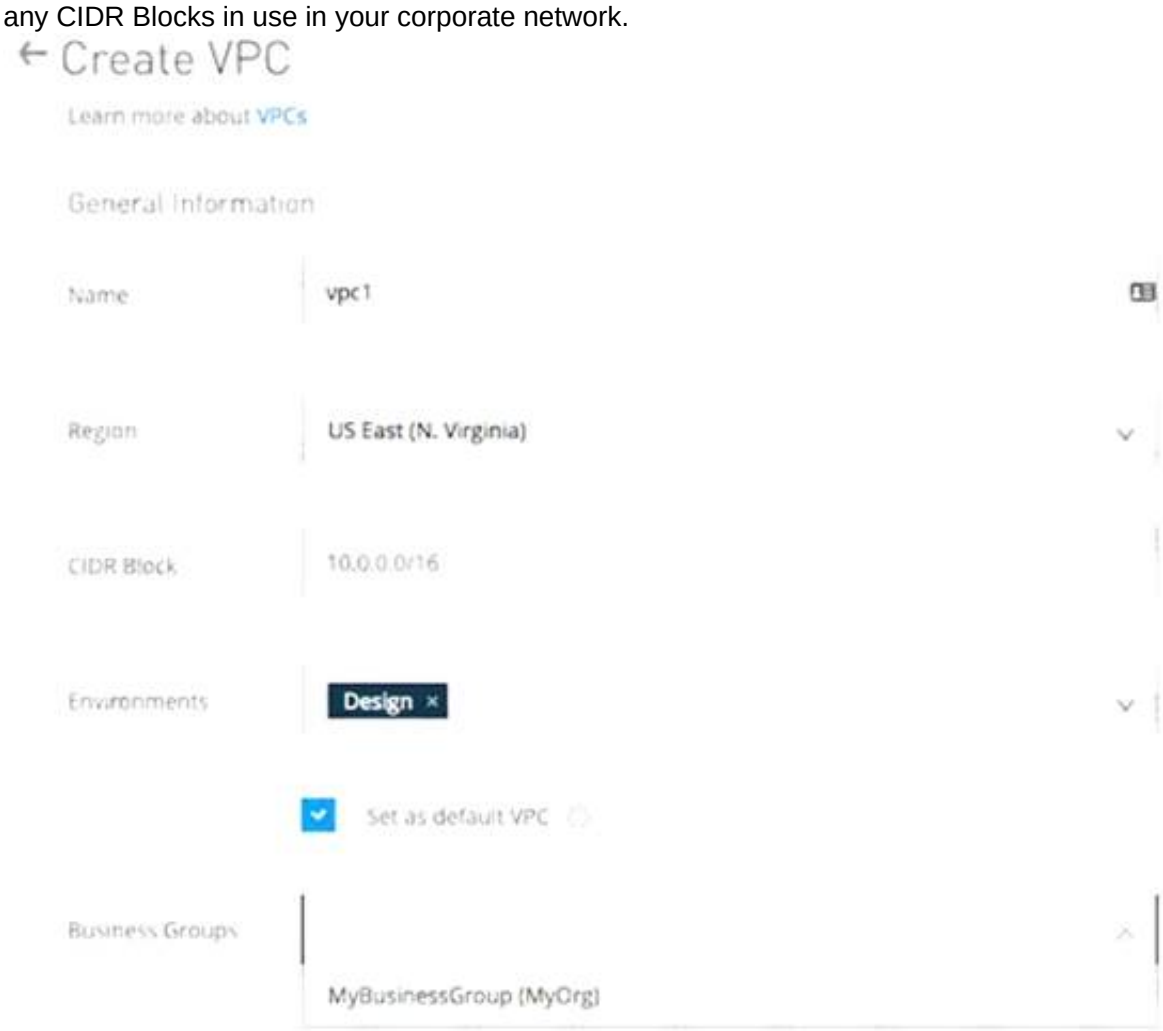
Explanation:
Correct Answer
When the API implementation changes the structure of the request or response messages
****
>> RAML definition usually needs to be touched only when there are changes in the request/response schemas or in any traits on API.
>> It need not be modified for any internal changes in API implementation like performance tuning, backend system migrations etc..

NEW QUESTION 24
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 staywithin 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 thr 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.



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 seperate 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 29
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 34
An API experiences a high rate of client requests (TPS) vwth small message paytoads. 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 36
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 39
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 43
When could the API data model of a System API reasonably mimic the data model exposed by the corresponding backend system, with minimal improvements over the backend system's data model?

A. When there is an existing Enterprise Data Model widely used across the organization
B. When the System API can be assigned to a bounded context with a corresponding data model
C. When a pragmatic approach with only limited isolation from the backend system is deemed appropriate
D. When the corresponding backend system is expected to be replaced in the near future

Answer: C

Explanation:

Correct Answer
When a pragmatic approach with only limited isolation from the backend system is deemed appropriate.
********** General guidance w.r.t choosing Data Models:
>> If an Enterprise Data Model is in use then the API data model of System APIs should make use of data types from that Enterprise Data Model and the corresponding API implementation should translate between these data types from the Enterprise Data Model and the native data model of the backend system.
>> If no Enterprise Data Model is in use then each System API should be assigned to a Bounded Context, the API data model of System APIs should make use of data types from the corresponding Bounded Context Data Model and the corresponding API implementation should translate between these data types from the Bounded Context Data Model and the native data model of the backend system. In this scenario, the data types in the Bounded Context Data Model are defined purely in terms of their business characteristics and are typically not related to the native data model of the backend system. In other words, the translation effort may be significant.
>> If no Enterprise Data Model is in use, and the definition of a clean Bounded Context Data Model is considered too much effort, then the API data model of System APIs should make use of data types that approximately mirror those from the backend system, same semantics and naming as backend system, lightly sanitized, expose all fields needed for the given System API's functionality, but not significantly more and making good use of REST conventions.
The latter approach, i.e., exposing in System APIs an API data model that basically mirrors that of the backend system, does not provide satisfactory isolation from backend systems through the System API tier on its own. In particular, it will typically not be possible to "swap out" a backend system without significantly changing all System APIs in front of that backend system and therefore the API implementations of all Process APIs that depend on those System APIs! This is so because it is not desirable to prolong the life of a previous backend system's data model in the form of the API data model of System APIs that now front a new backend system. The API data models of System APIs following this approach must therefore change when the backend system is replaced.
On the other hand:
>> It is a very pragmatic approach that adds comparatively little overhead over accessing the backend system directly
>> Isolates API clients from intricacies of the backend system outside the data model (protocol, authentication, connection pooling, network address, …)
>> Allows the usual API policies to be applied to System APIs
>> Makes the API data model for interacting with the backend system explicit and visible, by exposing it in the RAML definitions of the System APIs
>> Further isolation from the backend system data model does occur in the API implementations of the Process API tier

NEW QUESTION 44
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 toad-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 47
What do the API invocation metrics provided by Anypoint Platform provide?

A. ROI metrics from APIs that can be directly shared with business users
B. Measurements of the effectiveness of the application network based on the level of reuse
C. Data on past API invocations to help identify anomalies and usage patterns across various APIs
D. Proactive identification of likely future policy violations that exceed a given threat threshold

Answer: C

Explanation:

Correct Answer
Data on past API invocations to help identify anomalies and usage patterns across various APIs
****
API Invocation metrics provided by Anypoint Platform:
>> Does NOT provide any Return Of Investment (ROI) related information. So the option suggesting it is OUT.
>> Does NOT provide any information w.r.t how APIs are reused, whether there is effective usage of APIs or not etc...
>> Does NOT prodive any prediction information as such to help us proactively identify any future policy violations.
So, the kind of data/information we can get from such metrics is on past API invocations to help identify anomalies and usage patterns across various APIs.

NEW QUESTION 50
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 55
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 60
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 65
An organization wants to make sure only known partners can invoke the organization's APIs. To achieve this security goal, the organization wants to enforce a Client ID Enforcement policy in API Manager so that only registered partner applications can invoke the organization's APIs. In what type of API implementation does MuleSoft recommend adding an API proxy to enforce the Client ID Enforcement policy, rather than embedding the policy directly in the application's JVM?

A. A Mule 3 application using APIkit
B. A Mule 3 or Mule 4 application modified with custom Java code
C. A Mule 4 application with an API specification
D. A Non-Mule application

Answer: D

Explanation:
Correct Answer
A Non-Mule application
****
>> All type of Mule applications (Mule 3/ Mule 4/ with APIkit/ with Custom Java Code etc) running on Mule Runtimes support the Embedded Policy Enforcement on them.
>> The only option that cannot have or does not support embedded policy enforcement and must have API Proxy is for Non-Mule Applications.
So, Non-Mule application is the right answer.

NEW QUESTION 69
What API policy would be LEAST LIKELY used when designing an Experience API that is intended to work with a consumer mobile phone or tablet application?

A. OAuth 2.0 access token enforcement
B. Client ID enforcement
C. JSON threat protection
D. IPwhitellst

Answer: D

Explanation:
Correct Answer
IP whitelist
****
>> OAuth 2.0 access token and Client ID enforcement policies are VERY common to apply on Experience APIs as API consumers need to register and access the APIs using one of these mechanisms
>> JSON threat protection is also VERY common policy to apply on Experience APIs to prevent bad or suspicious payloads hitting the API implementations.
>> IP whitelisting policy is usually very common in Process and System APIs to only whitelist the IP range inside the local VPC. But also applied occassionally on some experience APIs where the End User/ API Consumers are FIXED.
>> When we know the API consumers upfront who are going to access certain Experience APIs, then we can request for static IPs from such consumers and whitelist them to prevent anyone else hitting the API.
However, the experience API given in the question/ scenario is intended to work with a consumer mobile phone or tablet application. Which means, there is no way we can know all possible IPs that are to be whitelisted as mobile phones and tablets can so many in number and any device in the city/state/country/globe.
So, It is very LEAST LIKELY to apply IP Whitelisting on such Experience APIs whose consumers are typically Mobile Phones or Tablets.

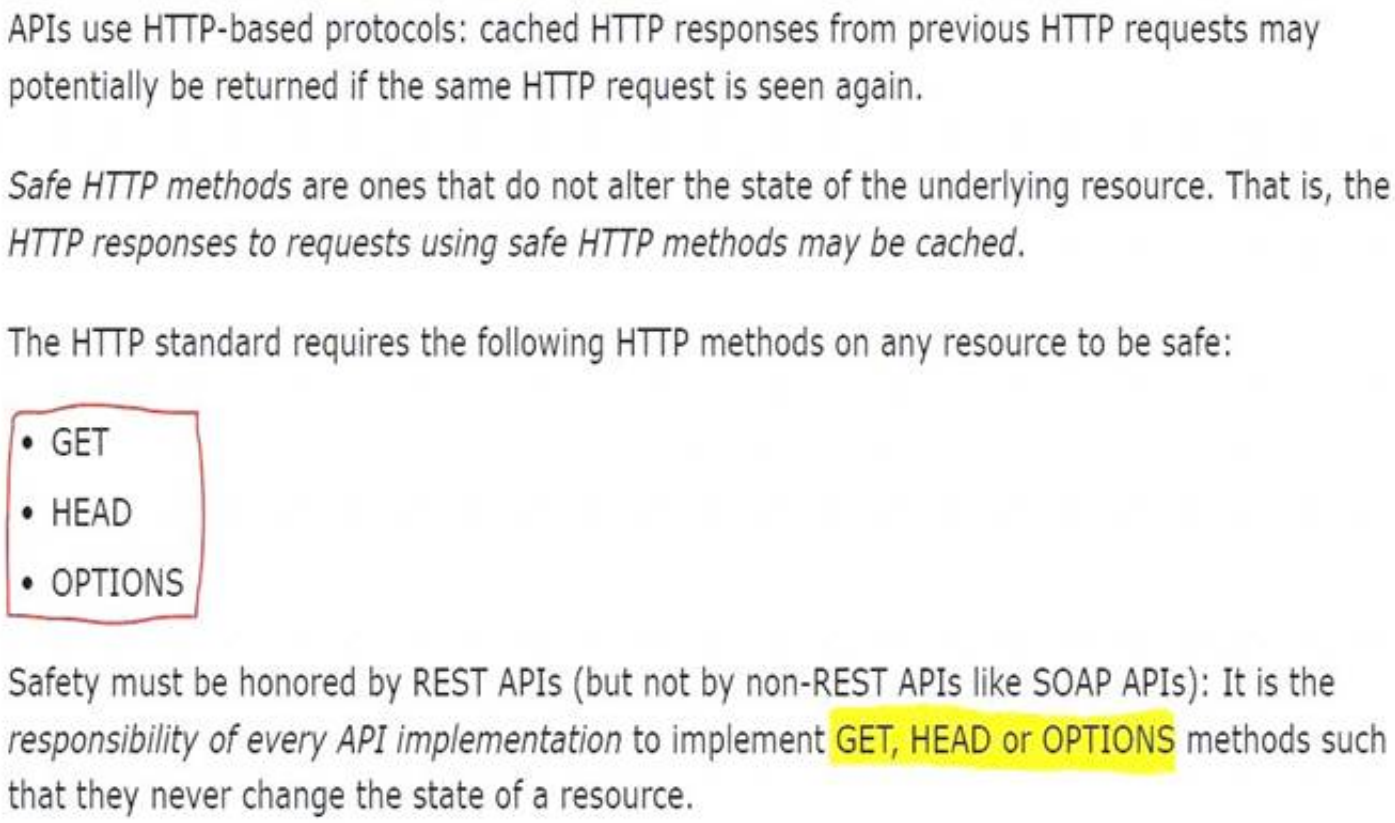
NEW QUESTION 73
The responses to some HTTP requests can be cached depending on the HTTP verb used in the request. According to the HTTP specification, for what HTTP verbs is this safe to do?

A. PUT, POST, DELETE
B. GET, HEAD, POST
C. GET, PUT, OPTIONS
D. GET, OPTIONS, HEAD

Answer: D

Explanation:

Correct Answer
GET, OPTIONS, HEAD



APIs use HTTP-based protocols: cached HTTP responses from previous HTTP requests may potentially be returned if the same HTTP request is seen again.

Safe HTTP methods are ones that do not alter the state of the underlying resource. That is, the *HTTP responses to requests using safe HTTP methods may be cached*.

The HTTP standard requires the following HTTP methods on any resource to be safe:

- GET
- HEAD
- OPTIONS

Safety must be honored by REST APIs (but not by non-REST APIs like SOAP APIs): It is the *responsibility of every API implementation* to implement GET, HEAD or OPTIONS methods such that they never change the state of a resource.

http://restcookbook.com/HTTP%20Methods/idempotency/

NEW QUESTION 76
What is true about automating interactions with Anypoint Platform using tools such as Anypoint Platform REST APIs, Anypoint CU, or the Mule Maven plugin?

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

Answer: C

Explanation:

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

NEW QUESTION 80
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 81
What is typically NOT a function of the APIs created within the framework called API-led connectivity?

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

Answer: A

Explanation:

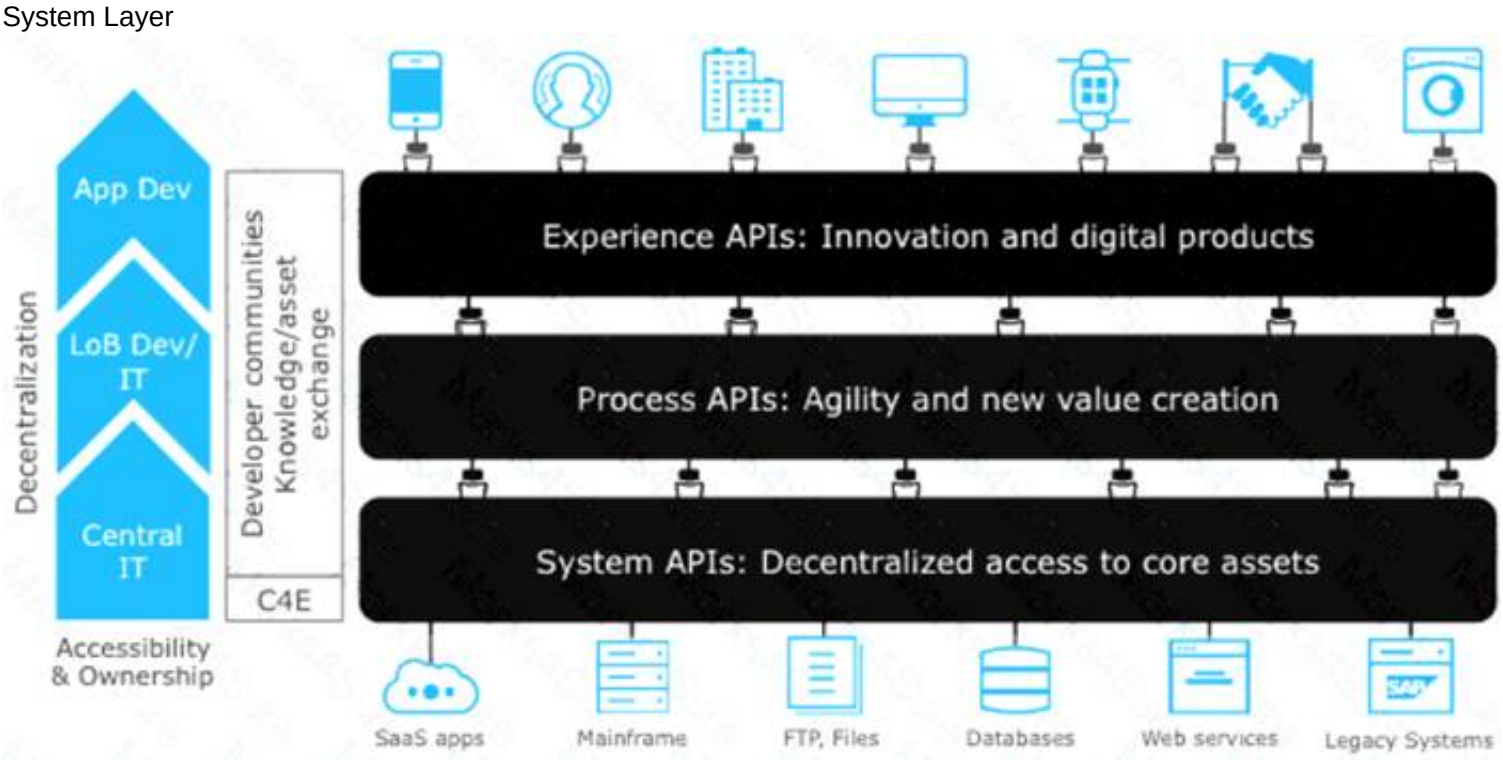
Correct Answer
They provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.
********** In API-led connectivity,
>> Experience APIs - allow for innovation at the user interface level by consuming the underlying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

NEW QUESTION 86
Which layer in the API-led connectivity focuses on unlocking key systems, legacy systems, data sources etc and exposes the functionality?

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

Answer: C

Explanation:
Correct Answer
System Layer



The APIs used in an API-led approach to connectivity fall into three categories:
System APIs – these usually access the core systems of record and provide a means of insulating the user from the complexity or any changes to the underlying systems. Once built, many users, can access data without any need to learn the underlying systems and can reuse these APIs in multiple projects.
Process APIs – These APIs interact with and shape data within a single system or across systems (breaking down data silos) and are created here without a dependence on the source systems from which that data originates, as well as the target channels through which that data is delivered.
Experience APIs – Experience APIs are the means by which data can be reconfigured so that it is most easily consumed by its intended audience, all from a common data source, rather than setting up separate
point-to-point integrations for each channel. An Experience API is usually created with API-first design principles where the API is designed for the specific user experience in mind.

NEW QUESTION 87
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 88
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 enforement 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 90
An Anypoint Platform organization has been configured with an external identity provider (IdP) for identity
management and client management. What credentials or token must be provided to Anypoint CLI to execute commands against the Anypoint Platform APIs?

A. The credentials provided by the IdP for identity management
B. The credentials provided by the IdP for client management
C. An OAuth 2.0 token generated using the credentials provided by the IdP for client management
D. An OAuth 2.0 token generated using the credentials provided by the IdP for identity management

Answer: A

Explanation:

Correct Answer
The credentials provided by the IdP for identity management
****

NEW QUESTION 91
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 92
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 93
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 95
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 97
An organization has implemented a Customer Address API to retrieve customer address information. This API has been deployed to multiple environments and has been configured to enforce client IDs everywhere.
A developer is writing a client application to allow a user to update their address. The developer has found the Customer Address API in Anypoint Exchange and wants to use it in their client application.
What step of gaining access to the API can be performed automatically by Anypoint Platform?

A. Approve the client application request for the chosen SLA tier

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

Answer: A

Explanation:

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

NEW QUESTION 100
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 104
.....

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