Number: Certified MuleSoft Developer
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: Certified MuleSoft Developer I

Exam Name: Salesforce Certified MuleSoft Developer I

Exam A

QUESTION 1
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 wifch the backend system

Correct Answer: B
Section:
Explanation:
Model all API resources and methods to closely mimic theoperations of the backend system.. >> There are NO fixed and straight best practices while opting data models for APIs. They are completly 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.

QUESTION 2
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

Correct Answer: A
Section:
Explanation:
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' PoliciesHowever, unfortunately, there is no proper way currently on Anypoint Platform to guard against DOS attacks.

QUESTION 3
An organization makes a strategic decision to move towards an IT operating model that emphasizes consumption of reusable IT assets using modern APIs (as defined by MuleSoft).
What best describes each modern API in relation to this new IT operating model?

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

Correct Answer: B
Section:
Explanation:

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

Bottom of FormTop of Form

QUESTION 4
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?

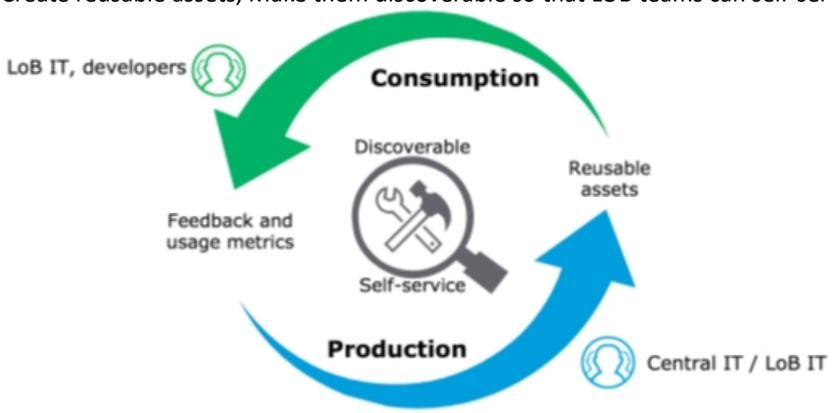
A. OAuth 2.0 access token enforcement
B. Client ID enforcement
C. JSON threat protection
D. IPwhitellst

Correct Answer: D
Section:
Explanation:
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.

QUESTION 5
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 resuable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

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



QUESTION 6
Traffic is routed through an API proxy to an API implementation. The API proxy is managed by API Manager and the API implementation is deployed to a CloudHub VPC using Runtime Manager. API policies have been applied to this API. In this deployment scenario, at what point are the API policies enforced on incoming API client requests?

A. At the API proxy
B. At the API implementation
C. At both the API proxy and the API implementation
D. At a MuleSoft-hosted load balancer

Correct Answer: A
Section:
Explanation:
At the API proxy. >> API Policies can be enforced at two places in Mule platform.>> One - As an Embedded Policy enforcement in the same Mule Runtime where API implementation is running.>> Two - On an API Proxy sitting in front of the Mule Runtime where API implementation is running.>> As the deployment scenario in the question has API Proxy involved, the policies will be enforced at the API Proxy.

QUESTION 7
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

Correct Answer: D
Section:
Explanation:
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 APIsSo, API consumer is the one who needs to request access on the API from Anypoint Exchange

QUESTION 8
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

Correct Answer: C
Section:
Explanation:
https://www.folkstalk.com/2019/11/mulesoft-integration-and-platform.htmlTo invoke OAuth 2.0-protected APIs managed by AnypointPlatform, 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 ManagementOnly 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'https://docs.mulesoft.com/api-manager/2.x/external-oauth-2.0-token-validation-policyhttps://blogs.mulesoft.com/dev/api-dev/api-security-ways-to-authenticate-and-authorize/

QUESTION 9
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

Correct Answer: D
Section:
Explanation:
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/

QUESTION 10
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?

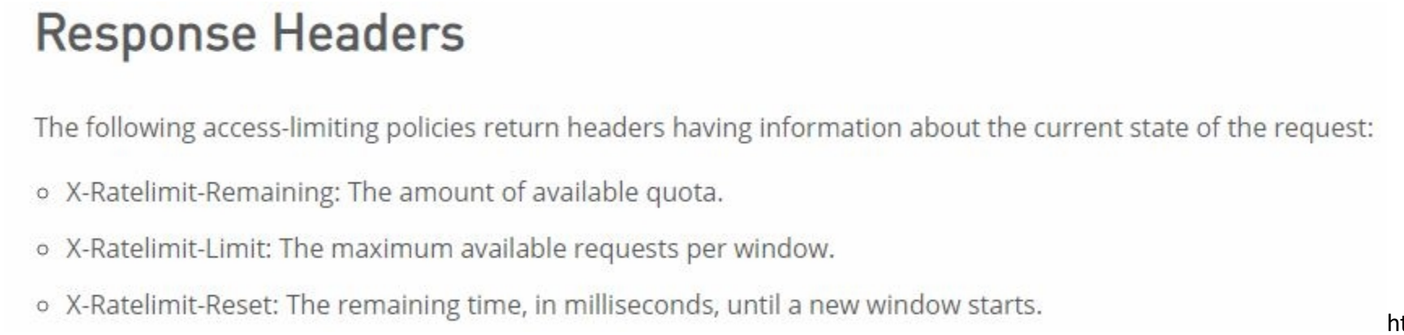
A. Redis distributed cache
B. java.util.WeakHashMap
C. Persistent Object Store
D. File-based storage

Correct Answer: C
Section:
Explanation:
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.

QUESTION 11
How can the application of a rate limiting API policy be accurately reflected in the RAML definition of an API?

A. By refining the resource definitions by adding a description of the rate limiting policy behavior
B. By refining the request definitions by adding a remaining Requests query parameter with description, type, and example
C. By refining the response definitions by adding the out-of-the-box Anypoint Platform rate-limit-enforcement securityScheme with description, type, and example
D. By refining the response definitions by adding the x-ratelimit-* response headers with description, type, and example

Correct Answer: D
Section:
Explanation:
By refining the response definitions by adding thex-ratelimit-*responseheaders with description, type, and example.



https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling#response-headershttps://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-policies#response-headers

QUESTION 12
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

Correct Answer: D
Section:
Explanation:
Apply a JSON threat protection policy to all APIs to detect potentialthreat 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.

QUESTION 13
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

Correct Answer: D
Section:
Explanation:
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.

QUESTION 14
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

Correct Answer: D
Section:
Explanation:
The number of API specifications in RAML or OAS format publishedto 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.

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

Correct Answer: D
Section:
Explanation:
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 connectorConsidering 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.

QUESTION 16
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 implementations and API clients

Correct Answer: D
Section:
Explanation:
When server-side load-balanced TLS mutual authentication isrequired 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 :>> WeCANNOT 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.

QUESTION 17
What do the API invocation metrics provided by Anypoint Platform provide?

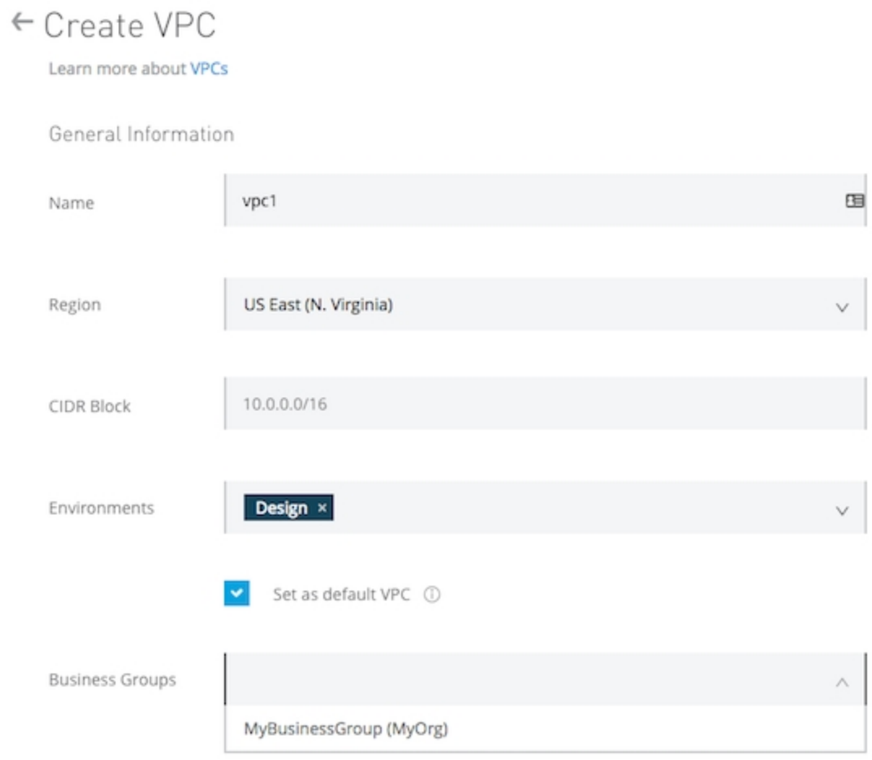
A. ROI metrics from APIs that can be directly shared with business users
B. Measurements of the effectiveness of the application network based on the level of reuse
C. Data on past API invocations to help identify anomalies and usage patterns across various APIs
D. Proactive identification of likely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

Correct Answer: C
Section:
Explanation:
Data on past API invocations to help identify anomalies and usagepatterns 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.

QUESTION 18
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

Correct Answer: B
Section:
Explanation:
Traffic between Mule applications deployed to an Anypoint VPC andon-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.

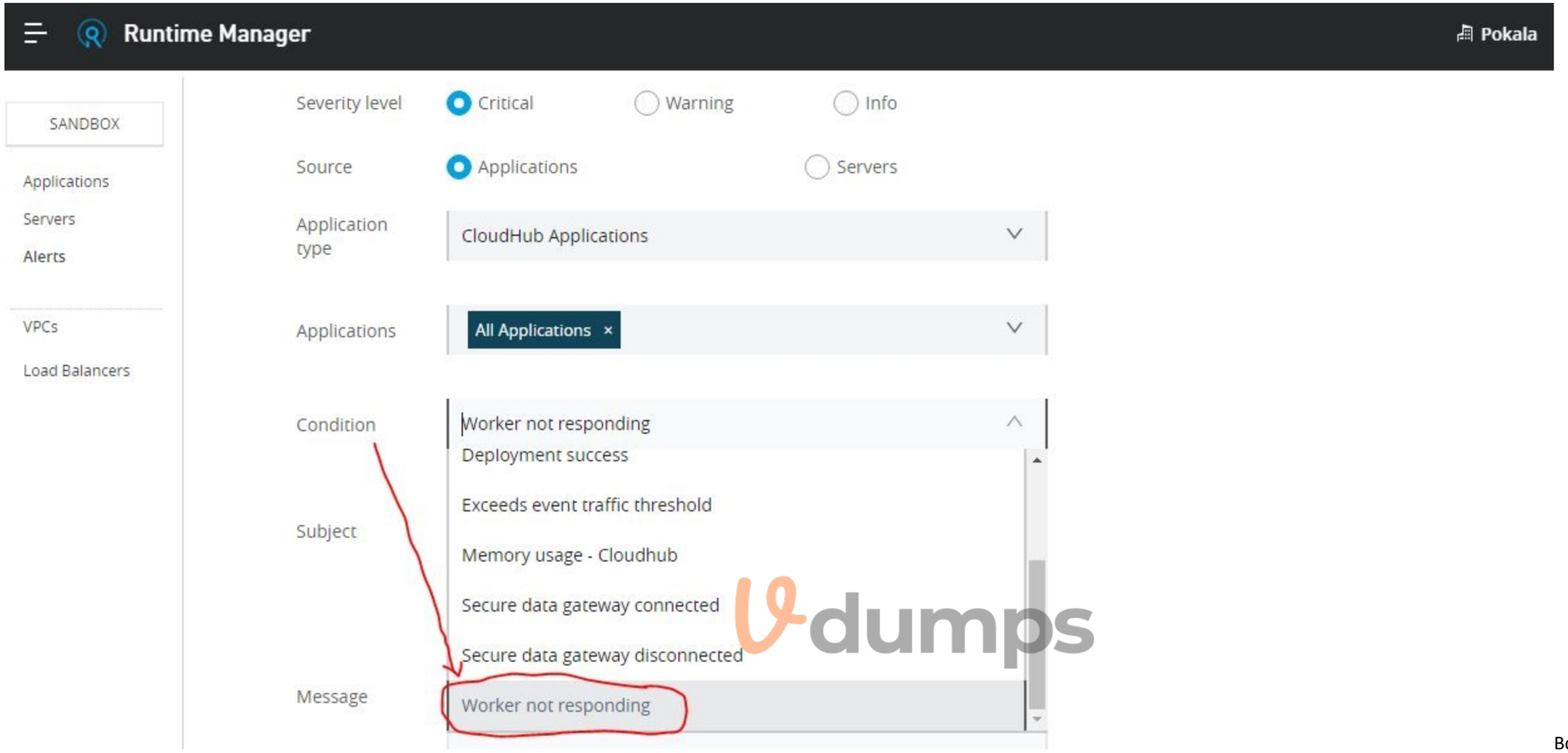


that each CloudHub environment requires a separate Anypoint VPC. Once an Anypoint VPC is created, we can choose a same VPC by multiple environments. However, it is generally a best and recommended practice to always have seperate Anypoint VPCs for Non-Prod and Prod environments.>> We use Anypoint VPN to link the underlying AWS VPC to an on-premises (non AWS) private network. NOT VPC Peering.Only true statement in the given choices is that the traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network.https://docs.mulesoft.com/runtime-manager/vpc-connectivity-methods-concept

QUESTION 19
An API implementation is deployed on a single worker on CloudHub and invoked by external API clients (outside of CloudHub). How can an alert be set up that is guaranteed to trigger AS SOON AS that API implementation stops responding to API invocations?

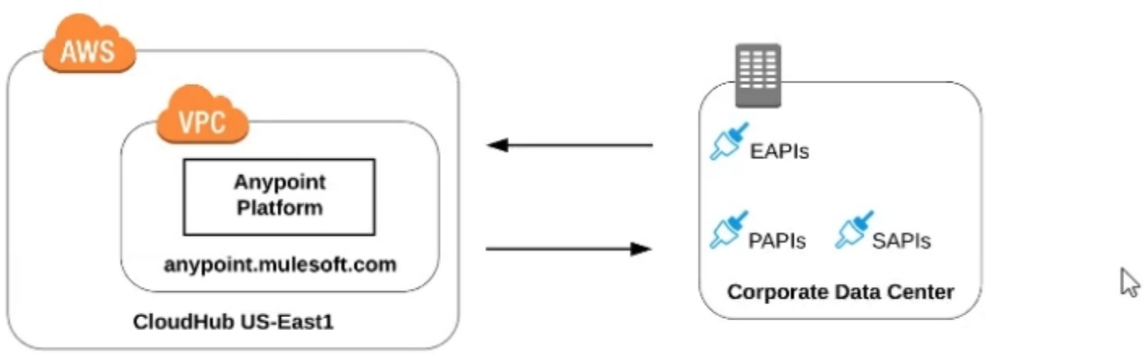
A. Implement a heartbeat/health check within the API and invoke it from outside the Anypoint Platform and alert when the heartbeat does not respond
B. Configure a 'worker not responding' alert in Anypoint Runtime Manager
C. Handle API invocation exceptions within the calling API client and raise an alert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

Correct Answer: B
Section:
Explanation:
Configure a ''Worker not responding'' alert in Anypoint Runtime Manager.. >>All the options eventually helps to generate the alert required when the application stops responding.>>However, handling exceptions within calling API and then raising alert from API client is inappropriate and silly. There could be many API clients invoking the API implementation and it is not ideal to have this setup consistently in all of them. Not a realistic way to do.>>Implementing a health check/ heartbeat with in the API and calling from outside to detmine the health sounds OK but needs extra setup for it and same time there are very good chances of generating false alarms when there are any intermittent network issues between external tool calling the health check API on API implementation. The API implementation itself may not have any issues but due to some other factors some false alarms may go out.>>Creating an alert in API Manager when the API receives no requests within a specified time period would actually generate realistic alerts but even here some false alarms may go out when there are genuinely no requests fromAPI clients.The best and right way to achieve this requirement is to setup an alert on Runtime Manager with a condition 'Worker not responding'. This would generate an alert ASSOONAS the workers become unresponsive.



Bottom of FormTop of Form

QUESTION 20
Refer to the exhibit.

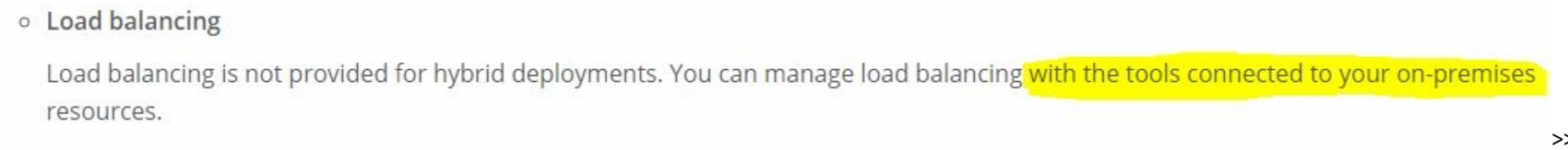


what is true when using customer-hosted Mule runtimes with the MuleSoft-hosted Anypoint Platform control plane (hybrid deployment)?

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

Correct Answer: C
Section:
Explanation:
API implementations can run successfully in customer-hosted Muleruntimes, 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



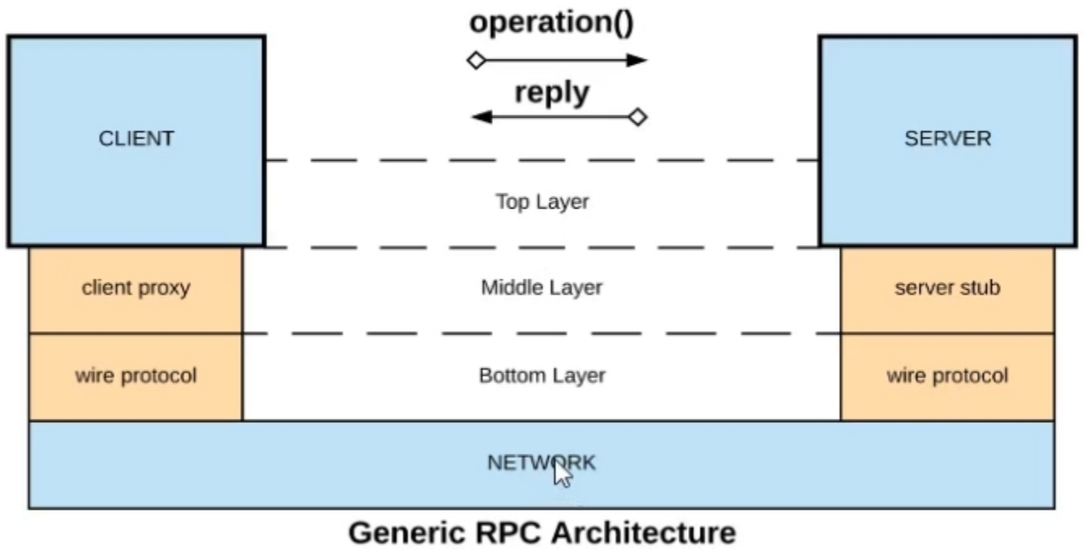
>>For Hybrid deployment models, the on-premises are first connected to Runtime Manager usingRuntime Manager agent.So, the connection is initiated first from On-premises to Runtime Manager. Then all control can be done from Runtime Manager.>>Anypoint Runtime Manager CANNOT ensure automatic HA. Clusters/Server 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.https://docs.mulesoft.com/runtime-manager/deployment-strategies#hybrid-deploymentshttps://help.mulesoft.com/s/article/On-Premise-Runtimes-Disconnected-From-US-Control-Plane-June-18th-2018https://help.mulesoft.com/s/article/Runtime-Manager-cannot-manage-On-Prem-Applications-and-Servers-from-US-Control-Plane-June-25th-2019https://help.mulesoft.com/s/article/On-premise-Runtimes-Appear-Disconnected-in-Runtime-Manager-May-29th-2018==========

QUESTION 21
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

Correct Answer: B
Section:
Explanation:
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

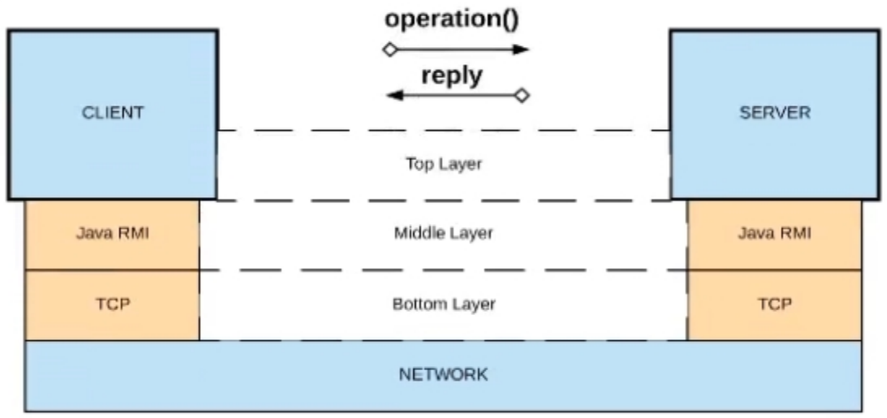
QUESTION 22
Refer to the exhibit.



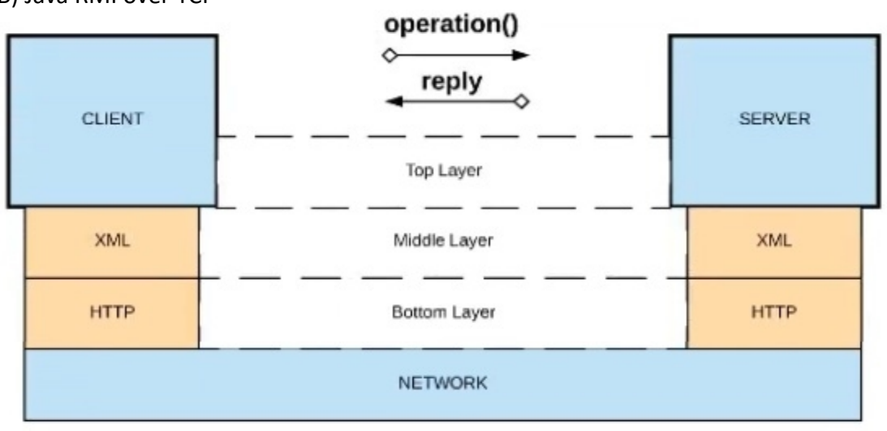
Generic RPC Architecture

What is a valid API in the sense of API-led connectivity and application networks?

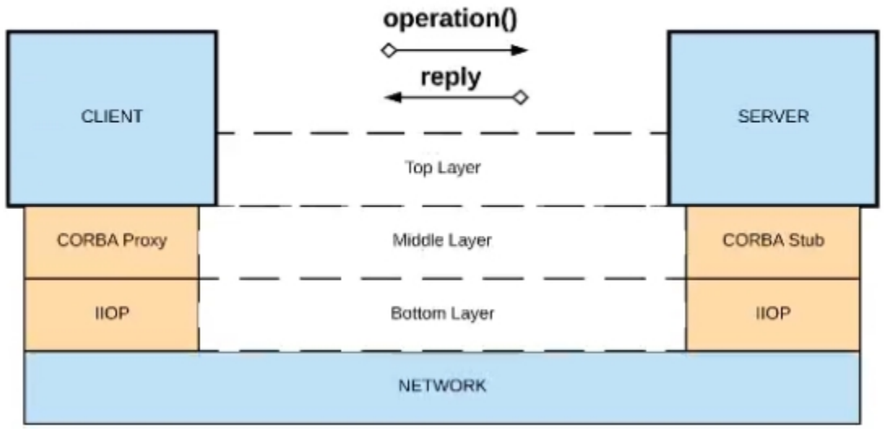
A) Java RMI over TCP



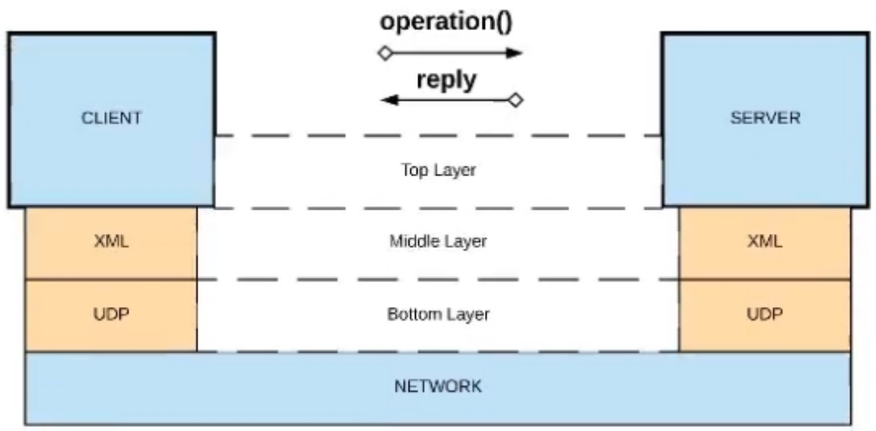
B) Java RMI over TCP



C) CORBA over HOP

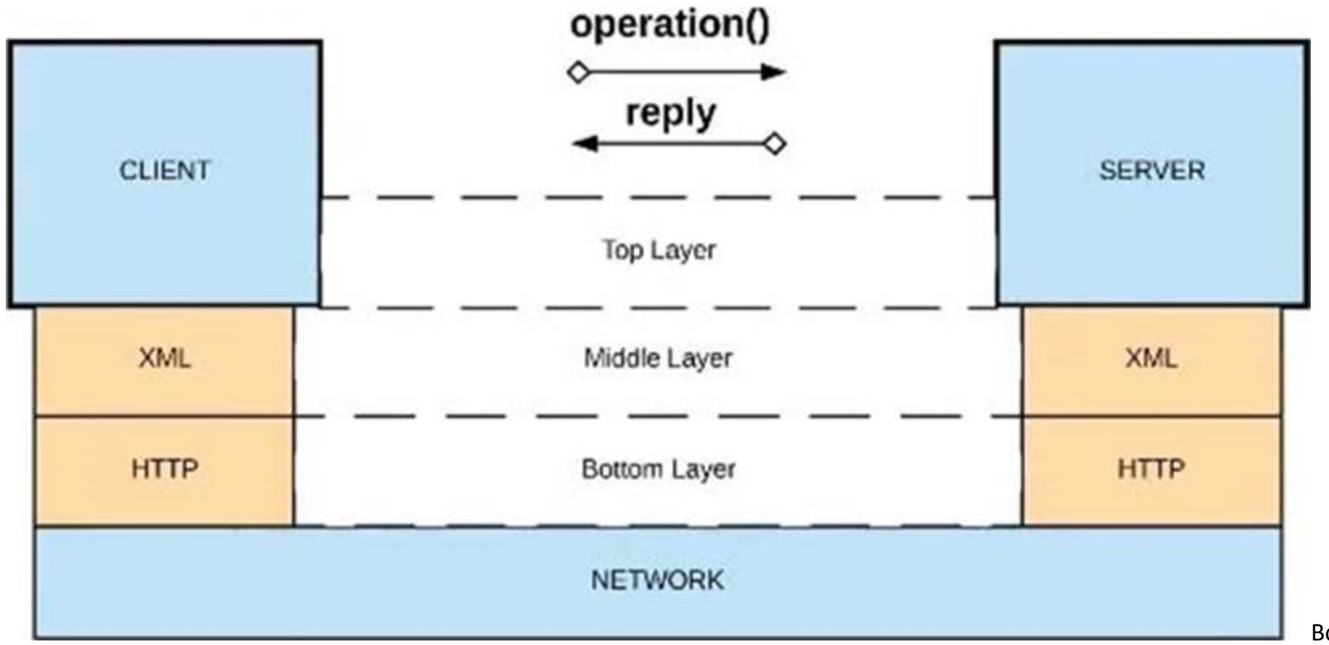


D) XML over UDP



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

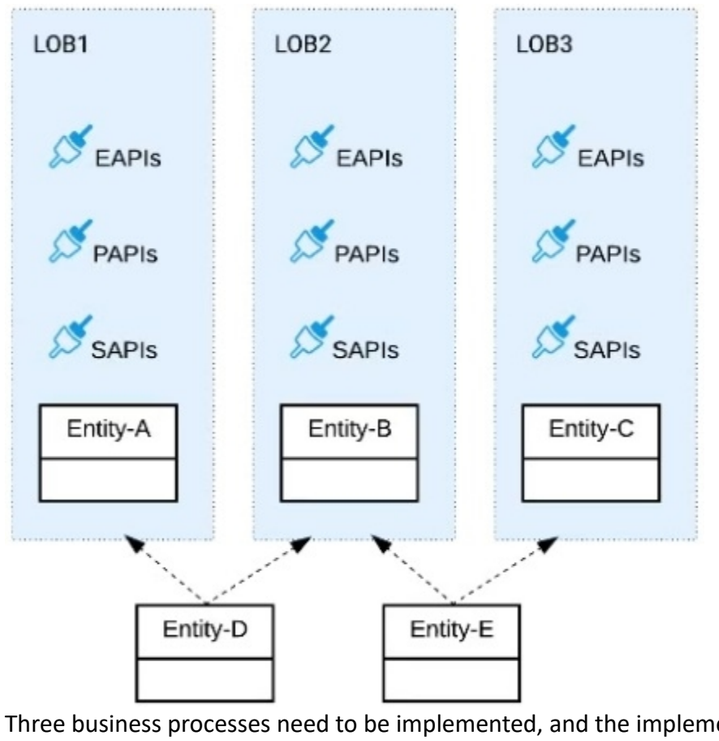
Correct Answer: D
Section:
Explanation:
XML over HTTP. >>API-led connectivity and Application Networks urge to have the APIs on HTTP based protocols for building most effective APIs and networks on top of them.>>The HTTP based APIs allow the platform to apply various varities of policies to address many NFRs>>The HTTP based APIs also allow to implement many standard and effective implementation patterns that adhere to HTTP based w3c rules.



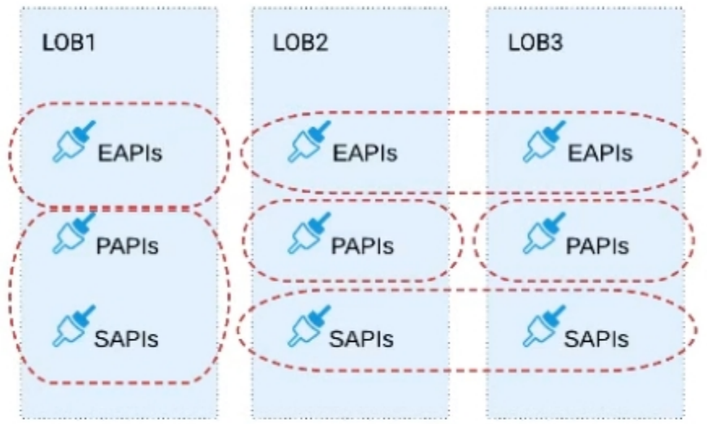
Bottom of FormTop of Form

QUESTION 23

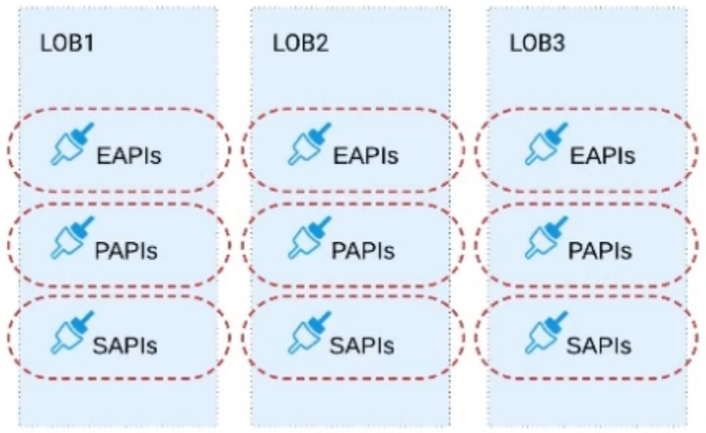
Refer to the exhibit.



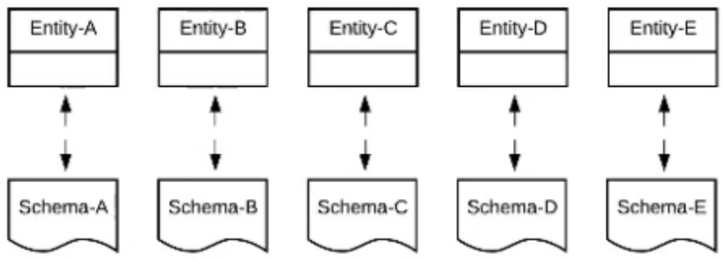
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications.
These processes are owned by separate (siloed) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB has one development team 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



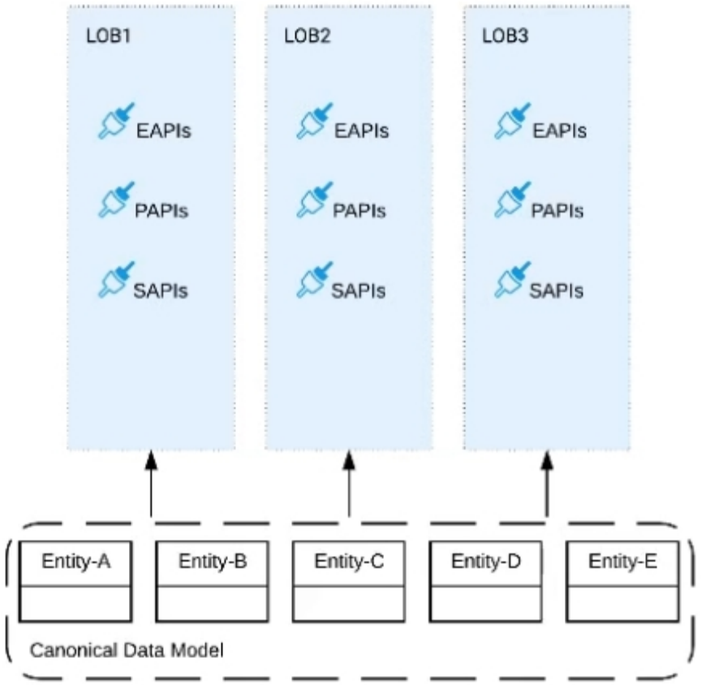
B) Build distinct data models for each API to follow established micro-services and Agile API-centric practices



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

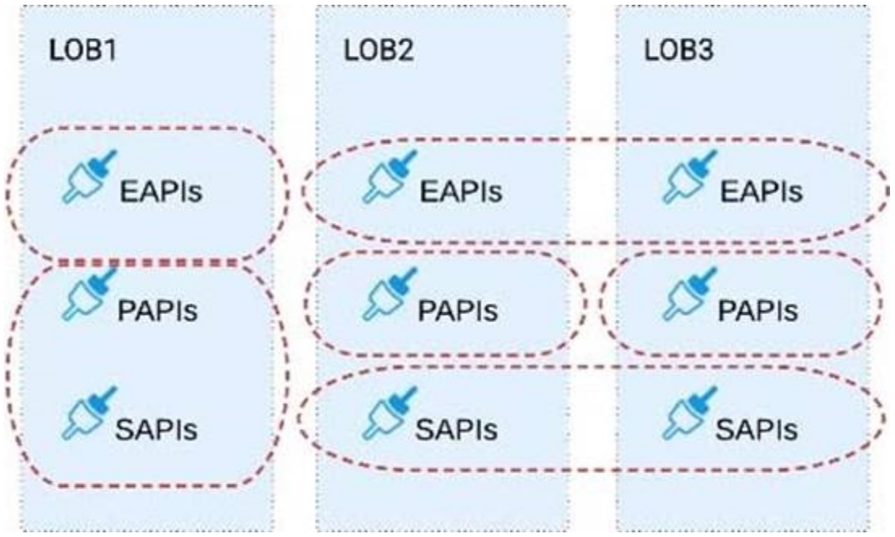


D) Build one centralized Canonical Data Model (Enterprise Data Model) that unifies all the data types from all three business processes, ensuring the data model is consistent and non-redundant



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

Correct Answer: A
Section:
Explanation:
Build several Bounded Context Data Models that align with coherent partsof 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.



QUESTION 24
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

Correct Answer: C
Section:
Explanation:
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.

QUESTION 25
An API has been updated in Anypoint Exchange by its API producer from version 3.1.1 to 3.2.0 following accepted semantic versioning practices and the changes have been communicated via the API's public portal.
The API endpoint does NOT change in the new version.
How should the developer of an API client respond to this change?

A. The update should be identified as a project risk and full regression testing of the functionality that uses this API should be run
B. The API producer should be contacted to understand the change to existing functionality
C. The API producer should be requested to run the old version in parallel with the new one
D. The API client code ONLY needs to be changed if it needs to take advantage of new features

Correct Answer: D
Section:

QUESTION 26

Mule applications that implement a number of REST APIs are deployed to their own subnet that is inaccessible from outside the organization.
External business-partners need to access these APIs, which are only allowed to be invoked from a separate subnet dedicated to partners - called Partner-subnet. This subnet is accessible from the public internet, which allows these external partners to reach it.
Anypoint Platform and Mule runtimes are already deployed in Partner-subnet. These Mule runtimes can already access the APIs.
What is the most resource-efficient solution to comply with these requirements, while having the least impact on other applications that are currently using the APIs?

A. Implement (or generate) an API proxy Mule application for each of the APIs, then deploy the API proxies to the Mule runtimes
B. Redeploy the API implementations to the same servers running the Mule runtimes
C. Add an additional endpoint to each API for partner-enablement consumption
D. Duplicate the APIs as Mule applications, then deploy them to the Mule runtimes

Correct Answer: A
Section:

QUESTION 27
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

Correct Answer: C
Section:
Explanation:
When a pragmatic approach with only limited isolation from thebackend 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

QUESTION 28
What best describes the Fully Qualified Domain Names (FQDNs), also known as DNS entries, created when a Mule application is deployed to the CloudHub Shared Worker Cloud?

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

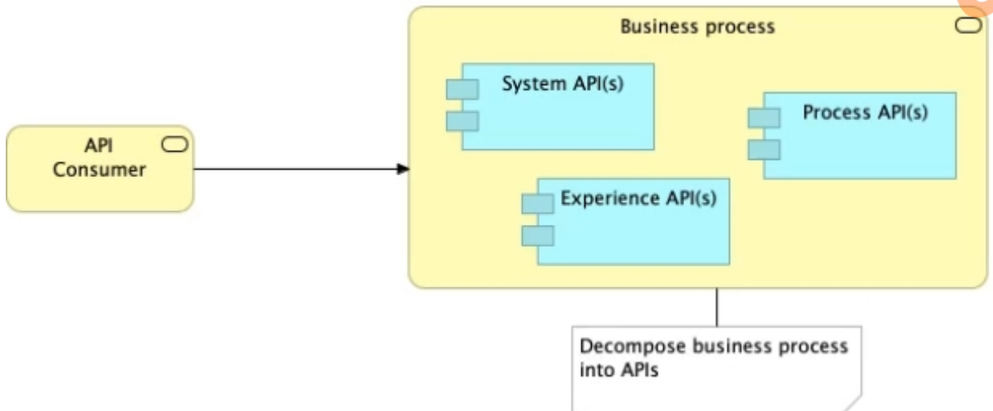
Correct Answer: B
Section:
Explanation:
The FQDNs are determined by the application name chosen,IRRESPECTIVE of the region. >> When deploying applications to Shared Worker Cloud, the FQDN are always determined by application name chosen.>> It does NOT matter what region the app is being deployed to.>> Although it is fact and true that the generated FQDN will have the region included in it (Ex: exp-salesorder-api.au-s1.cloudhub.io), it does NOT mean that the same name can be used when deploying to another CloudHub region.>> Application name should be universally unique irrespective of Region and Organization and solely determines the FQDN for Shared Load Balancers.

QUESTION 29
When using CloudHub with the Shared Load Balancer, what is managed EXCLUSIVELY by the API implementation (the Mule application) and NOT by Anypoint Platform?

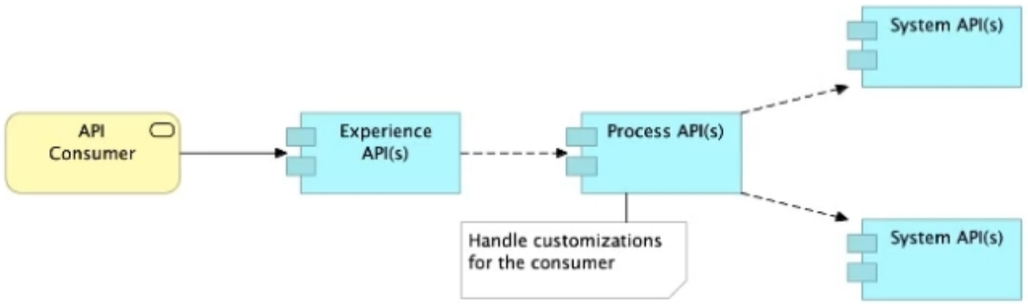
A. The assignment of each HTTP request to a particular CloudHub worker
B. The logging configuration that enables log entries to be visible in Runtime Manager
C. The SSL certificates used by the API implementation to expose HTTPS endpoints
D. The number of DNS entries allocated to the API implementation

Correct Answer: C
Section:
Explanation:
The SSL certificates used by the API implementation to exposeHTTPS 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.

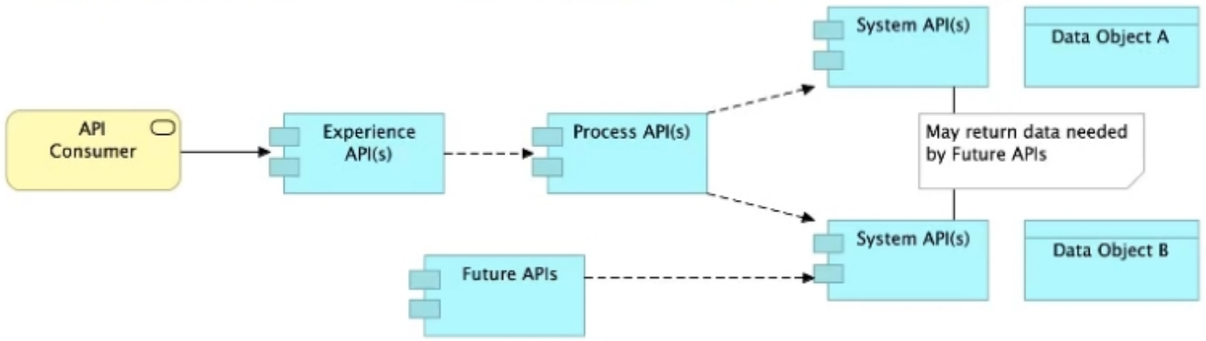
QUESTION 30
Refer to the exhibit.



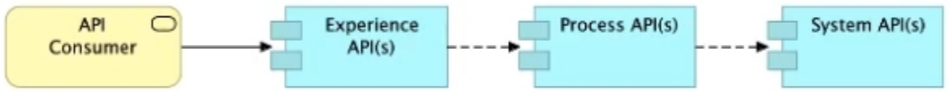
What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?
A) Handle customizations for the end-user application at the Process API level rather than the Experience API level



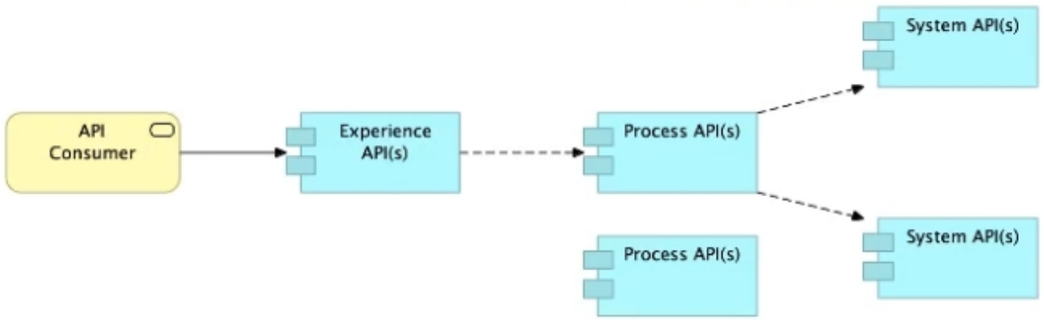
B) Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs



C) Always use a tiered approach by creating exactly one API for each of the 3 layers (Experience, Process and System APIs)



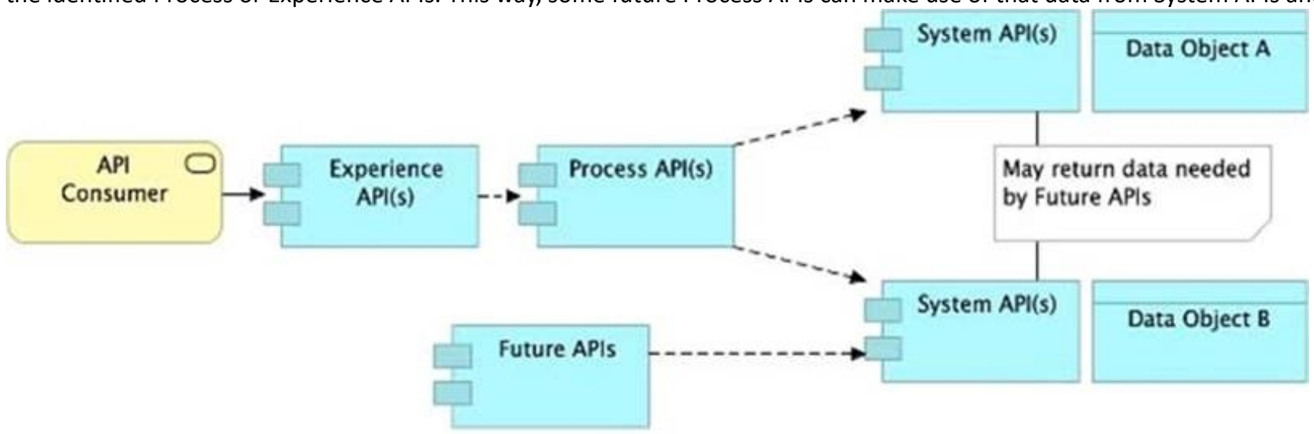
D) Use a Process API to orchestrate calls to multiple System APIs, but NOT to other Process APIs



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

Correct Answer: B
Section:
Explanation:
Allow System APIs to return data that is NOT currently required bythe identified Process or Experience APIs.. >> All customizations for the end-user application should be handled in 'Experience API' only. Not in Process API>> We should use tiered approach but NOT always by creating exactly one API for each of the 3 layers. Experience APIs might be one but Process APIs and System APIs are often more than one. System APIs for sure will be more

than one all the time as they are the smallest modular APIs built in front of end systems.>> Process APIs can call System APIs as well as other Process APIs. There is no such anti-design pattern in API-Led connectivity saying Process APIs should not call other Process APIs.So, the right answer in the given set of options that makes sense as per API-Led connectivity principles is to allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs. This way, some future Process APIs can make use of that data from System APIs and we need NOT touch the System layer APIs again and again.



QUESTION 31
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

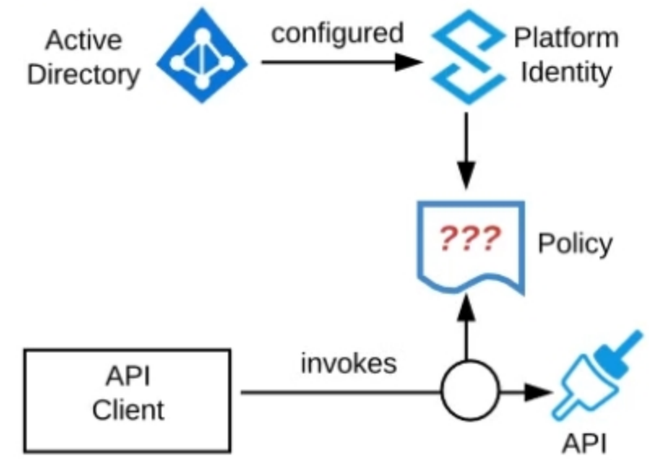
Correct Answer: A
Section:
Explanation:
Use an SLA-based rate limiting policy and assign a clientapplication to a matching SLA tier based on its type.. >> SLA tiers will come into play whenever any limits to be imposed on APIs based on client type

QUESTION 32
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

Correct Answer: C
Section:
Explanation:
Create API Notebooks and Include them in the relevant Anypoint exchangeentries. >>API Notebooks are the one on Anypoint Platform that enable us to provide code-centric API documentationBottom of FormTop of Form

QUESTION 33
Refer to the exhibit. An organization is running a Mule standalone runtime and has configured Active Directory as the Anypoint Platform external Identity Provider. The organization does not have budget for other system components.



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

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

Correct Answer: A
Section:
Explanation:
Apply a basic authentication - LDAP policy; the internal ActiveDirectory will be configured as the LDAP source for authenticating users.. >> IP Whitelisting does NOT fit for this purpose. Moreover, the users workstations may not necessarily have static IPs in the network.>> OAuth 2.0 enforcement requires a client provider which isn't in the organizations system components.>> It is not an effective approach to let every user create separate client credentials and configure those for their usage.The effective way it to apply a basic authentication - LDAP policy and the internal Active Directory will be configured as the LDAP source for authenticating users.

QUESTION 34
A new upstream API Is being designed to offer an SLA of 500 ms median and 800 ms maximum (99th percentile) response time. The corresponding API implementation needs to sequentially invoke 3 downstream APIs of very similar complexity.
The first of these downstream APIs offers the following SLA for its response time: median: 100 ms, 80th percentile: 500 ms, 95th percentile: 1000 ms.
If possible, how can a timeout be set in the upstream API for the invocation of the first downstream API to meet the new upstream API's desired SLA?

A. Set a timeout of 50 ms; this times out more invocations of that API but gives additional room for retries
B. Set a timeout of 100 ms; that leaves 400 ms for the other two downstream APIs to complete
C. No timeout is possible to meet the upstream API's desired SLA; a different SLA must be negotiated with the first downstream API or invoke an alternative API
D. Do not set a timeout; the Invocation of this API Is mandatory and so we must wait until it responds

Correct Answer: B
Section:
Explanation:
Set a timeout of 100ms; that leaves 400ms for other twodownstream APIs to complete. Key details to take from the given scenario:>> Upstream API's designed SLA is 500ms (median). Lets ignore maximum SLA response times.>> This API calls 3 downstream APIs sequentially and all these are of similar complexity.>> The first downstream API is offering median SLA of 100ms, 80th percentile: 500ms; 95th percentile: 1000ms.Based on the above details:>> We can rule out the option which is suggesting to set 50ms timeout. Because, if the median SLA itself being offered is 100ms then most of the calls are going to timeout and time gets wasted in retried them and eventually gets exhausted with all retries. Even if some retries gets successful, the remaining time wont leave enough room for 2nd and 3rd downstream APIs to respond within time.>> The option suggesting to NOT set a timeout as the invocation of this API is mandatory and so we must wait until it responds is silly. As not setting time out would go against the good implementation pattern and moreover if the first API is not responding within its offered median SLA 100ms then most probably it would either respond in 500ms (80th percentile) or 1000ms (95th percentile). In BOTH cases, getting a successful response from 1st downstream API does NO GOOD

because already by this time the Upstream API SLA of 500 ms is breached. There is no time left to call 2nd and 3rd downstream APIs.>> It is NOT true that no timeout is possible to meet the upstream APIs desired SLA.As 1st downstream API is offering its median SLA of 100ms, it means MOST of the time we would get the responses within that time. So, setting a timeout of 100ms would be ideal for MOST calls as it leaves enough room of 400ms for remaining 2 downstream API calls.

QUESTION 35
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

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

QUESTION 36
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

Correct Answer: A
Section:
Explanation:
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..

QUESTION 37
What are the major benefits of MuleSoft proposed IT Operating Model?

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

Correct Answer: C
Section:
Explanation:
1. Decrease the IT delivery gap2. Meet various business demands without increasing the IT capacity3. Make consumption of assets at the rate of production..

QUESTION 38
A Mule application exposes an HTTPS endpoint and is deployed to three CloudHub workers that do not use static IP addresses. The Mule application expects a high volume of client requests in short time periods. What is the most cost-effective infrastructure component that should be used to serve the high volume of client requests?

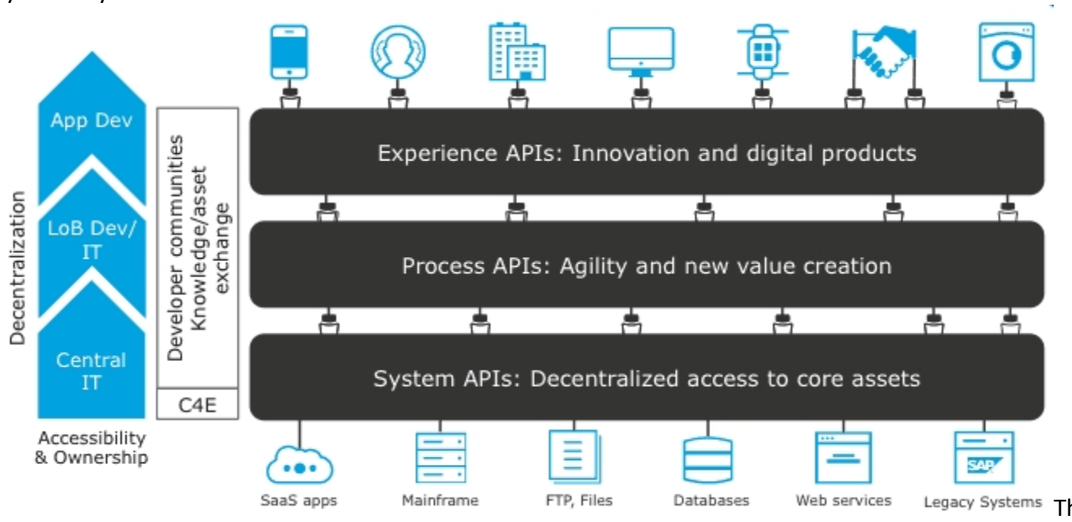
A. A customer-hosted load balancer
B. The CloudHub shared load balancer
C. An API proxy
D. Runtime Manager autoscaling

Correct Answer: B
Section:
Explanation:
The CloudHub shared load balancer. The scenario in this question can be split as below:>> There are 3 CloudHub workers (So, there are already good number of workers to handle high volume of requests)>> The workers are not using static IP addresses (So, one CANNOT use customer load-balancing solutions without static IPs)>> Looking for most cost-effective component to load balance the client requests among the workers.Based on the above details given in the scenario:>> Runtime autoscaling is NOT at all cost-effective as it incurs extra cost. Most over, there are already 3 workers running which is a good number.>> We cannot go for a customer-hosted load balancer as it is also NOT most cost-effective (needs custom load balancer to maintain and licensing) and same time the Mule App is not having Static IP Addresses which limits from going with custom load balancing.>> An API Proxy is irrelevant there as it has no role to play w.r.t handling high volumes or load balancing.So, the only right option to go with and fits the purpose of scenario being most cost-effective is - using a CloudHub Shared Load Balancer.

QUESTION 39
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

Correct Answer: C
Section:
Explanation:
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.

QUESTION 40
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

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

QUESTION 41
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.

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

QUESTION 42
Due to a limitation in the backend system, a system API can only handle up to 500 requests per second. What is the best type of API policy to apply to the system API to avoid overloading the backend system?

A. Rate limiting
B. HTTP caching
C. Rate limiting - SLA based
D. Spike control

Correct Answer: D
Section:
Explanation:
Spike control. >> First things first, HTTP Caching policy is for purposes different than avoiding the backend system from overloading. So this is OUT.>> Rate Limiting and Throttling/ Spike Control policies are designed to limit API access, but have different intentions.>> Rate limiting protects an API by applying a hard limit on its access.>> Throttling/ Spike Control shapes API access by smoothing spikes in traffic.That is why, Spike Control is the right

option.

QUESTION 43
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

Correct Answer: C
Section:
Explanation:
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.

QUESTION 44
The application network is recomposable: it is built for change because it 'bends but does not break'

A. TRUE
B. FALSE

Correct Answer: A
Section:
Explanation:
.
>> Application Network is a disposable architecture.
>> Which means, it can be altered without disturbing entire architecture and its components.
>> It bends as per requirements or design changes but does not break

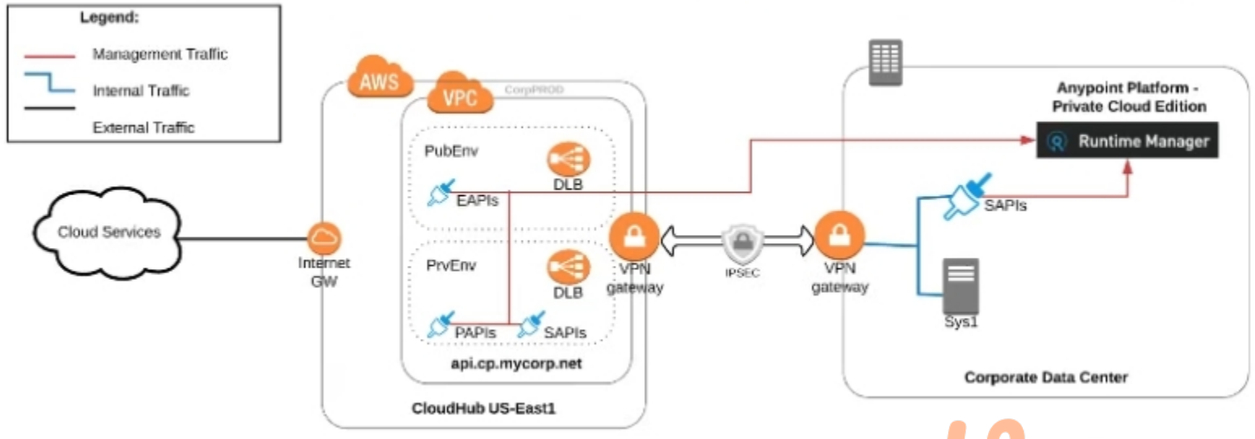
QUESTION 45
A system API has a guaranteed SLA of 100 ms per request. The system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. An upstream process API invokes the system API and the main goal of this process API is to respond to client requests in the least possible time. In what order should the system APIs be invoked, and what changes should be made in order to speed up the response time for requests from the process API?

A. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response
B. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment using a scatter-gather configured with a timeout, and then merge the responses
C. Invoke the system API deployed to the primary environment, and if it fails, invoke the system API deployed to the DR environment
D. Invoke ONLY the system API deployed to the primary environment, and add timeout and retry logic to avoid intermittent failures

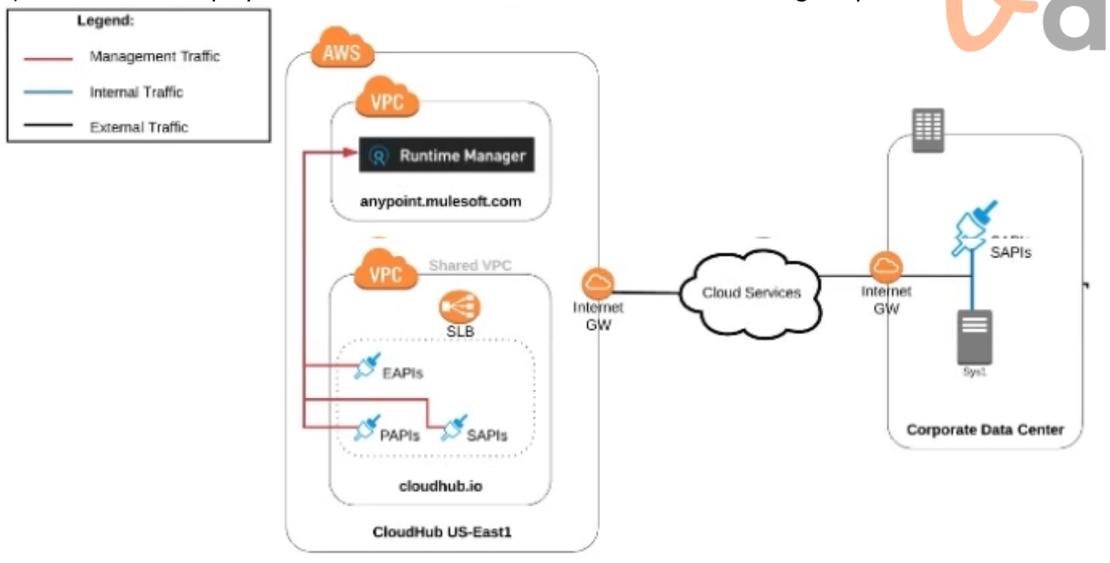
Correct Answer: A
Section:
Explanation:
In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response.. >> The API requirement in the given scenario is to respond in least possible time.>> The option that is suggesting to first try the API in primary environment and then fallback to API in DR environment would result in successful response but NOT in least possible time. So, this is NOT a right choice of implementation for given requirement.>> Another option that is suggesting to ONLY invoke API in primary environment and to add timeout and retries may also result in successful response upon retries but

NOT in least possible time. So, this is also NOT a right choice of implementation for given requirement.>> One more option that is suggesting to invoke API in primary environment and API in DR environment in parallel using Scatter-Gather would result in wrong API response as it would return merged results and moreover, Scatter-Gather does things in parallel which is true but still completes its scope only on finishing all routes inside it. So again, NOT a right choice of implementation for given requirementThe Correct choice is to invoke the API in primary environment and the API in DR environment parallelly, and using ONLY the first response received from one of them.

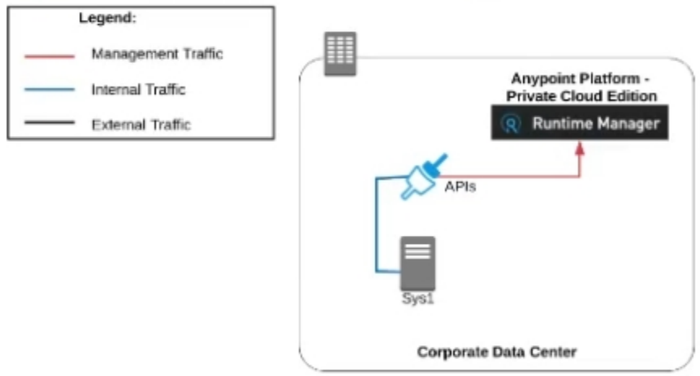
QUESTION 46
An organization uses various cloud-based SaaS systems and multiple on-premises systems. The on-premises systems are an important part of the organization's application network and can only be accessed from within the organization's intranet.
What is the best way to configure and use Anypoint 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



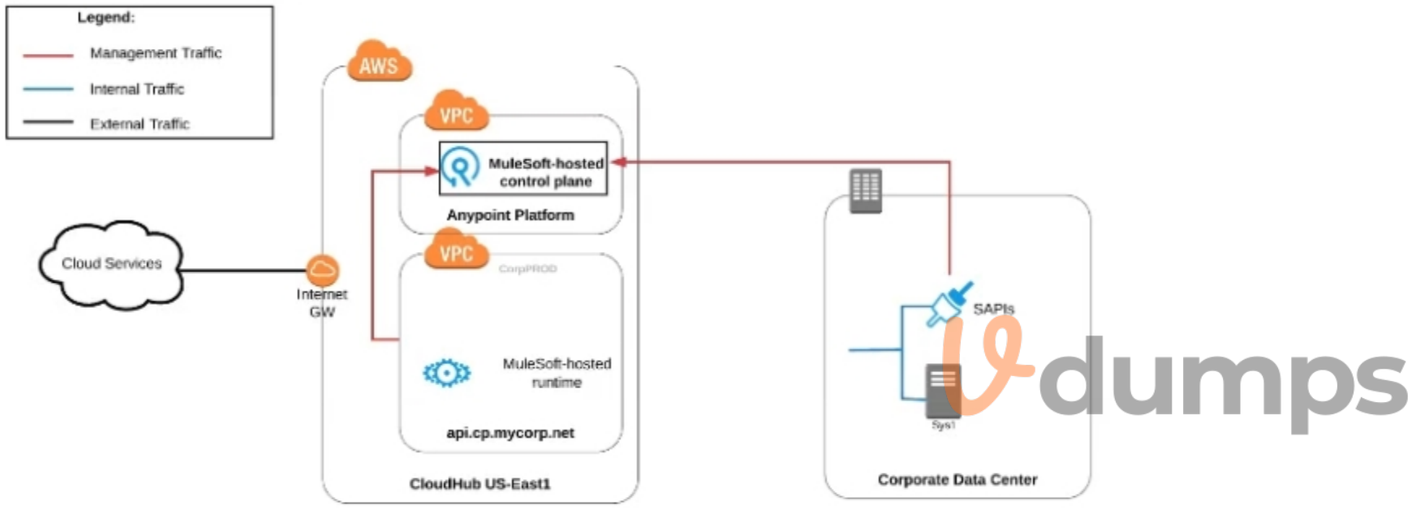
B) Use CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform control plane



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

Correct Answer: B
Section:
Explanation:
Use a combination of CloudHub-deployed and manually provisionedon-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 intranetLet 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.

QUESTION 47
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

Correct Answer: A
Section:

QUESTION 48
What is true about API implementations when dealing with legal regulations that require all data processing to be performed within a certain jurisdiction (such as in the USA or the EU)?

A. They must avoid using the Object Store as it depends on services deployed ONLY to the US East region
B. They must use a Jurisdiction-local external messaging system such as Active MQ rather than Anypoint MQ
C. They must te deployed to Anypoint Platform runtime planes that are managed by Anypoint Platform control planes, with both planes in the same Jurisdiction
D. They must ensure ALL data is encrypted both in transit and at rest

Correct Answer: C
Section:
Explanation:
They must be deployed to Anypoint Platform runtime planes thatare managed by Anypoint Platform control planes, with both planes in the same Jurisdiction.. >> As per legal regulations, all data processing to be performed within a certain jurisdiction. Meaning, the data in USA should reside within USA and should not go out. Same way, the data in EU should reside within EU and should not go out.>> So, just encrypting the data in transit and at rest does not help to be compliant with the rules. We need to make sure that data does not go out too.>> The data that we are talking here is not just about the messages that are published to Anypoint MQ. It includes the apps running, transaction states, application logs, events, metric info and any other metadata. So, just replacing Anypoint MQ with a locally hosted ActiveMQ does NOT help.>> The data that we are talking here is not just about the key/value pairs that are stored in Object Store. It includes the messages published, apps running, transaction states, application logs, events, metric info and any other metadata. So, just avoiding using Object Store does NOT help.>> The only option left and also the right option in the given choices is to deploy application on runtime and control planes that are both within the jurisdiction.

QUESTION 49
Which of the following sequence is correct?

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

Correct Answer: B
Section:
Explanation:
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.

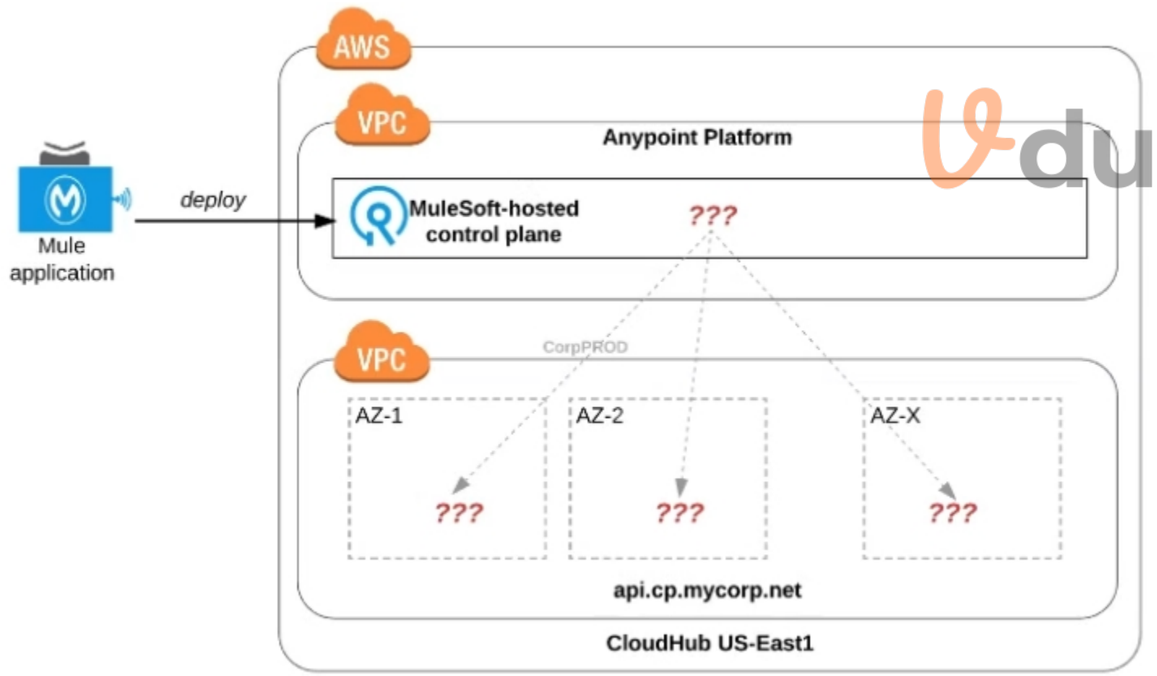
QUESTION 50

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

Correct Answer: C
Section:
Explanation:
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.

QUESTION 51
Refer to the exhibit.



An organization uses one specific CloudHub (AWS) region for all CloudHub deployments.
How are CloudHub workers assigned to availability zones (AZs) when the organization's Mule applications are deployed to CloudHub in that region?

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

Correct Answer: D
Section:
Explanation:
Workers are randomly distributed across available AZs within that region.. >>Currently, we only have control to choose which AWS Region to choose but there is no control at all using any configurations or deployment options to decide what Availability Zone (AZ) to assign to what worker.>>There areNOfixed or implicit rules on platform too w.r.t assignment of AZ to workers based on environment or application.>>They are completely assigned inrandom. However, cloudhub definitely ensures that HA is achieved by assigning the workers to more than on AZ so that all workers are not assigned to same AZ for same application.Bottom of FormTop of Form