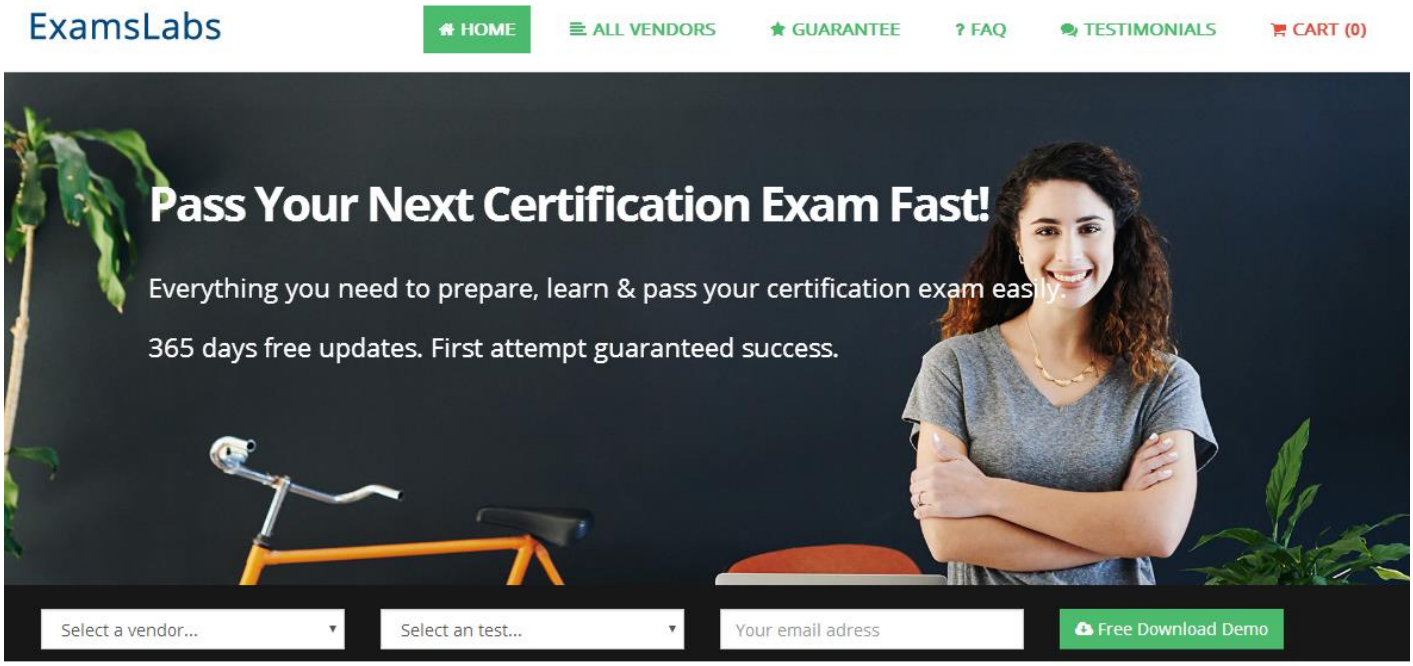
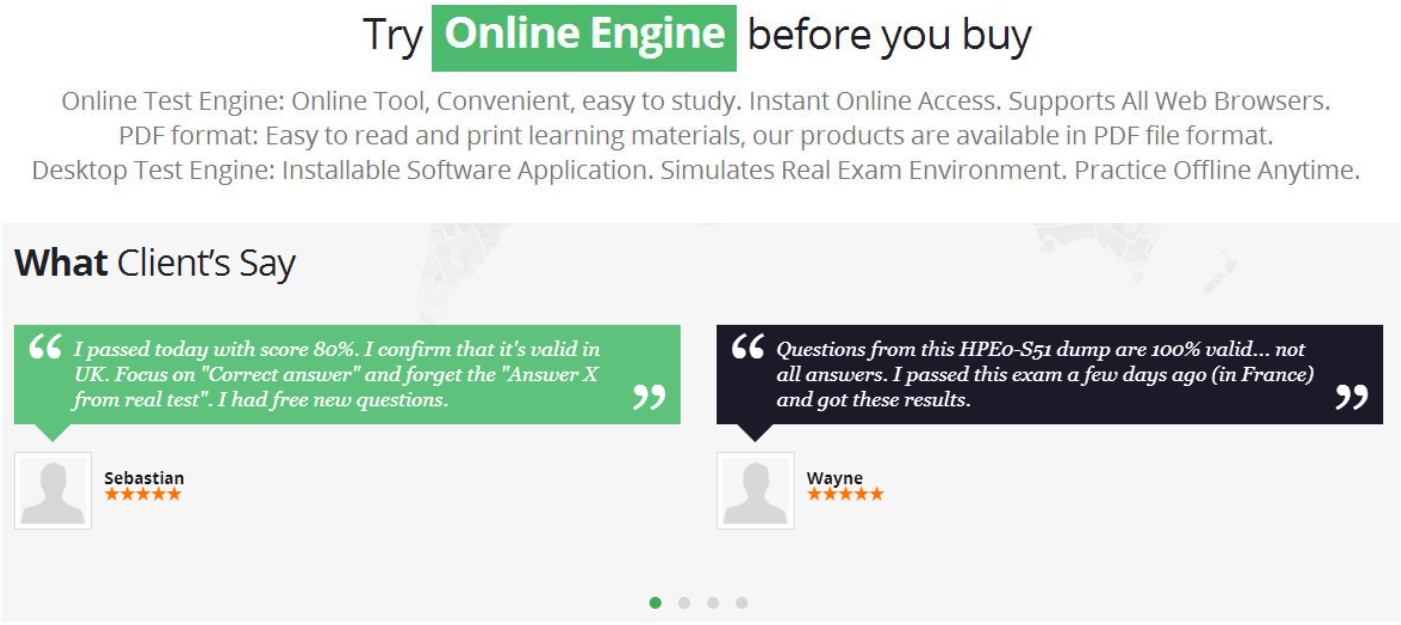
ExamsLabs





http://www.examslabs.com/

Latest Study Materials, Valid Dumps - ExamsLabs

Exam : **MuleSoft-Platform-Architect-I**

Title : Salesforce Certified MuleSoft Platform Architect I

Vendor : Salesforce

Version : DEMO

NO.1 Which of the following sequence is correct?

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

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

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

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

Answer: B

Explanation:

Correct Answe r: 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.

NO.2 Which layer in the API-led connectivity focuses on unlocking key systems, legacy systems, data sources etc and exposes the functionality?

A. Experience Layer

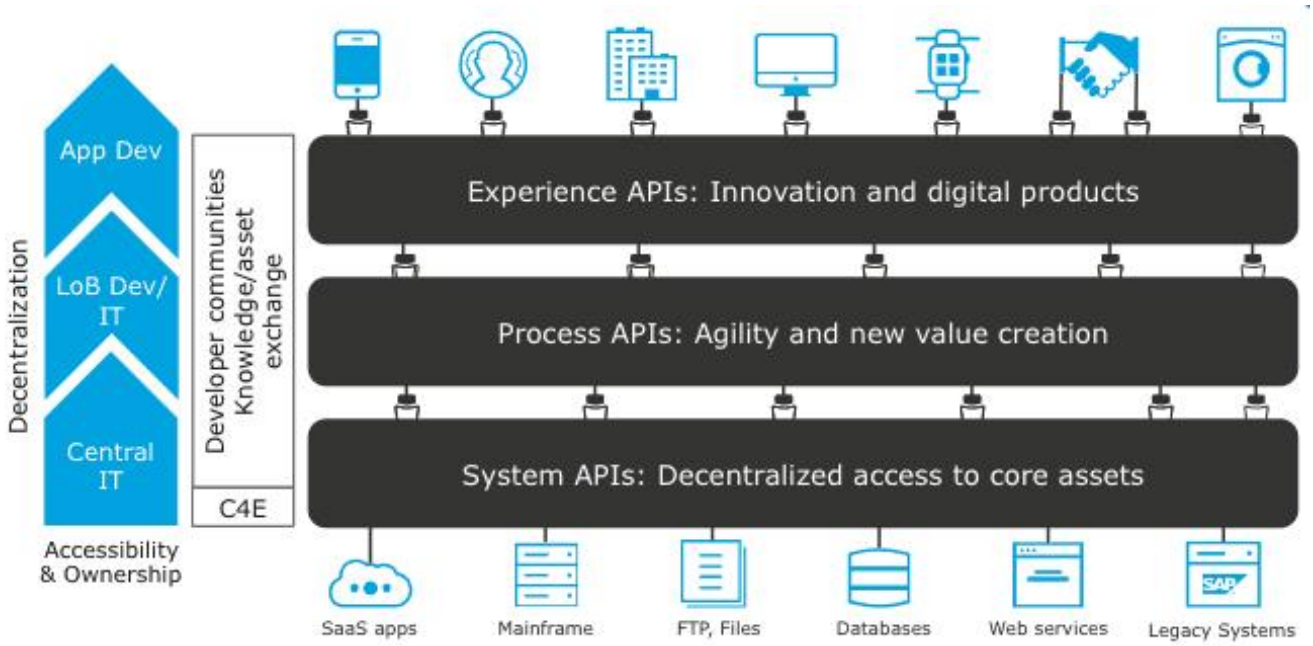
B. Process Layer

C. System Layer

Answer: C

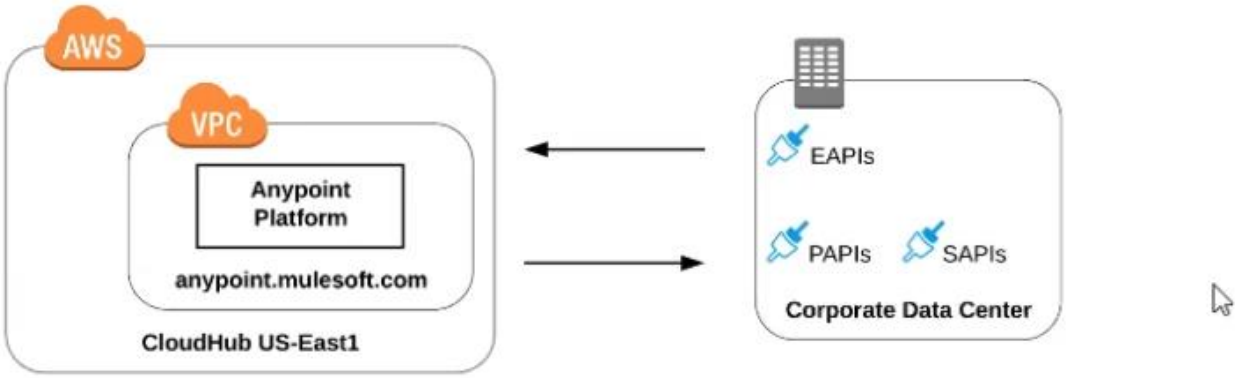
Explanation:

Correct Answe r: 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.

NO.3 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 Answe r: 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

>> For Hybrid deployment models, the on-premises are first connected to Runtime Manager using Runtime 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.

Reference:

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-2018

https://help.mulesoft.com/s/article/Runtime-Manager-cannot-manage-On-Prem-Applications-and-Servers-from-US-Control-Plane-June-25th-2019

https://help.mulesoft.com/s/article/On-premise-Runtimes-Appear-Disconnected-in-Runtime-Manager-May-29th-2018

NO.4 In an organization, the InfoSec team is investigating Anypoint Platform related data traffic. From where does most of the data available to Anypoint Platform for monitoring and alerting originate?

A. From the Mule runtime or the API implementation, depending on the deployment model

B. From various components of Anypoint Platform, such as the Shared Load Balancer, VPC, and Mule runtimes

C. From the Mule runtime or the API Manager, depending on the type of data

D. From the Mule runtime irrespective of the deployment model

Answer: D

Explanation:

Correct Answe r: From the Mule runtime irrespective of the deployment model

****

>> Monitoring and Alerting metrics are always originated from Mule Runtimes irrespective of the deployment model.

>> It may seems that some metrics (Runtime Manager) are originated from Mule Runtime and some are (API Invocations/ API Analytics) from API Manager. However, this is realistically NOT TRUE. The reason is, API manager is just a management tool for API instances but all policies upon applying on APIs eventually gets executed on Mule Runtimes only (Either Embedded or API Proxy).

>> Similarly all API Implementations also run on Mule Runtimes.
So, most of the day required for monitoring and alerts are originated fron Mule Runtimes only irrespective of whether the deployment model is MuleSoft-hosted or Customer-hosted or Hybrid.

NO.5 4 Production environment is running on a dedicated Virtual Private Cloud (VPC) on CloudHub 1,0, and the security team guidelines clearly state no traffic on HTTP.
Which two options support these security guidelines?
Choose 2 answers

A. Configure the HTTPS protocol in HTTP listener in the Mule application

B. Create a custom policy to apply to outgoing and incoming HTTP requests to control access to a configured API endpoint

C. Remove the entry from the VPC firewall rule

```
{
  "CIDR Block": "0.0.0.0/0", // (Anywhere)
  "Protocol": "TCP",
  "From port": 8081,
},
{
  "CIDR Block": "10.111.0.0/24", // (Local VPC)
  "Protocol": "TCP",
  "From port": 8091,
}
```

D.
Configure the IP Blocklist policy to control access to a configured API endpoint from either a single IP address or a range of IP addresses.

E.
Add the entry in the VPC firewall rule.

```
{
  "CIDR Block": "0.0.0.0/0", // (Anywhere)
  "Protocol": "TCP",
  "From port": 8081,
},
{
```

Answer: A,C
Explanation:
Security Guidelines Overview:

The production environment is hosted on a dedicated Virtual Private Cloud (VPC) on CloudHub 1.0, with a specific requirement from the security team that no traffic should occur over HTTP. This implies that only secure HTTPS traffic should be permitted, and HTTP access (port 8081, the default HTTP port in Mule applications) should be disabled.
Evaluating the Options:
Option A (Correct Answer): Configuring the HTTPS protocol in the HTTP listener in the Mule application ensures that all traffic is encrypted and occurs over HTTPS (port 8092 by default for HTTPS on Mule applications). This directly aligns with the security guideline to prevent unencrypted HTTP traffic.
Option B: Creating a custom policy for incoming and outgoing HTTP requests could provide some control over access, but it does not enforce the use of HTTPS exclusively. This option does not disable HTTP traffic and, therefore, does not meet the guideline effectively.
Option C (Correct Answer): Removing the entry for HTTP (port 8081) in the VPC firewall rule ensures that HTTP traffic is completely blocked at the firewall level. This prevents any HTTP requests from reaching the application, adding a layer of security that complies with the guidelines.
Option D: The IP Blocklist policy controls access based on IP addresses but does not enforce the use of HTTPS. This policy does not address the specific requirement of preventing HTTP traffic.
Option E: Adding a firewall rule entry for HTTP (port 8081) would enable HTTP traffic, which directly contradicts the security guidelines. Therefore, this option should be avoided.
Conclusion:
Option A and Option C are the correct choices. Configuring the HTTPS protocol in the Mule application's HTTP listener ensures that only HTTPS traffic is allowed, and removing the firewall rule for HTTP (port 8081) blocks any HTTP traffic from reaching the application. Together, these options enforce secure traffic as required by the security guidelines.
Refer to MuleSoft documentation on configuring HTTP listeners and managing VPC firewall rules for further details on implementing these security controls.

NO.6 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

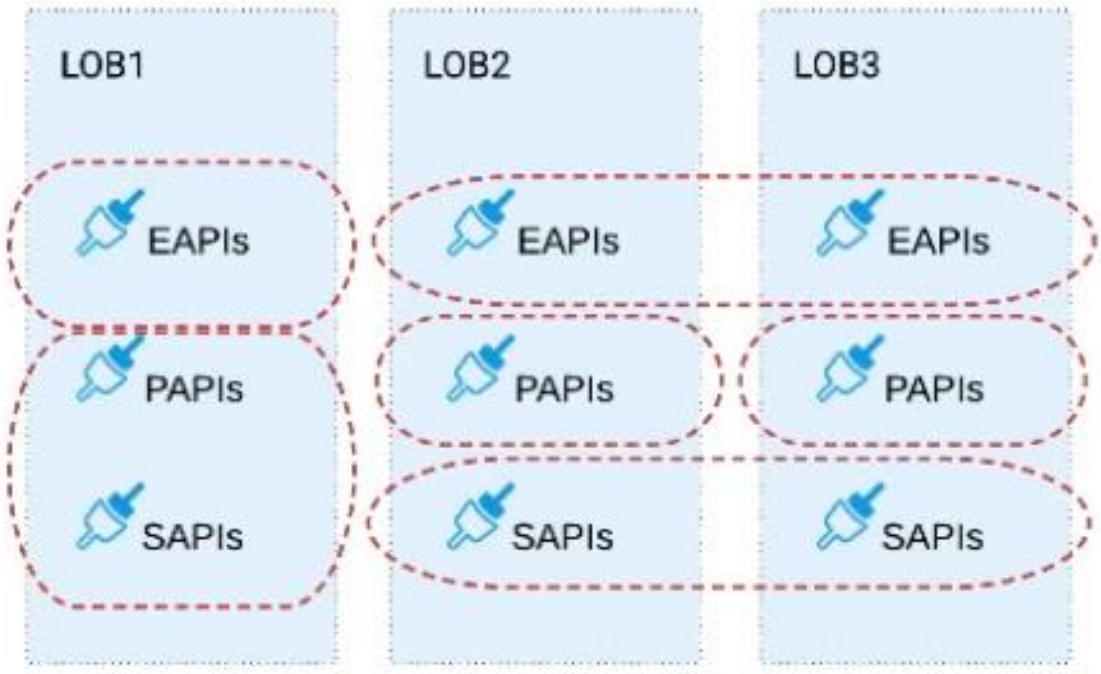
C. The fraction of API implementations deployed manually relative to those deployed using a CI/CD tool

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

Answer: D
Explanation:
Correct Answe r: 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.

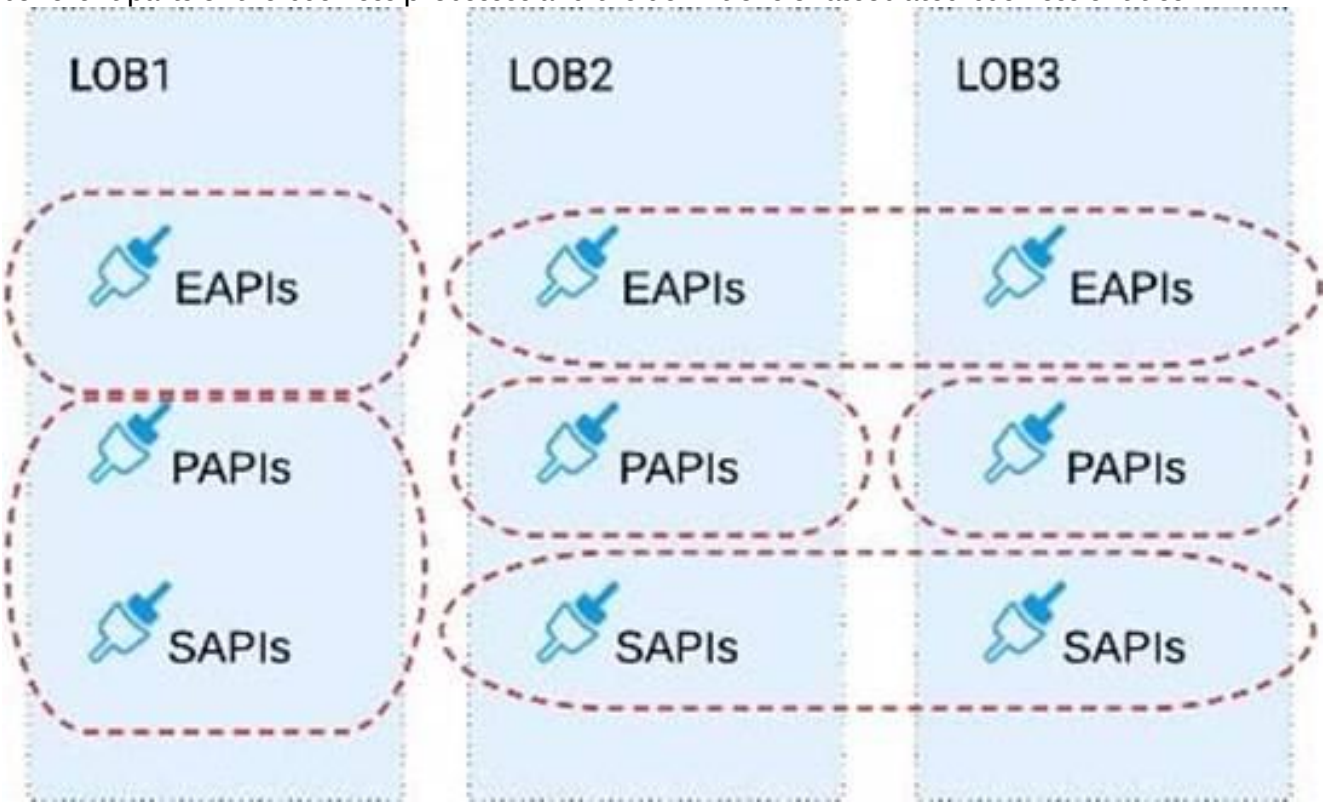
NO.7 Refer to the exhibit.
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications.
These processes are owned by separate (siloed) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB 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 Answe r: 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.



NO.8 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 Answe r: 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.

NO.9 Several times a week, an API implementation shows several thousand requests per minute in an Anypoint Monitoring dashboard, Between these bursts, the dashboard shows between two and five requests per minute. The API implementation is running on Anypoint Runtime Fabric with two non-clustered replicas, reserved vCPU 1.0 and vCPU Limit 2.0.
An API consumer has complained about slow response time, and the dashboard shows the 99 percentile is greater than 120 seconds at the time of the complaint. It also shows greater than 90% CPU usage during these time periods.
In manual tests in the QA environment, the API consumer has consistently reproduced the slow response time and high CPU usage, and there were no other API requests at this time. In a brainstorming session, the engineering team has created several proposals to reduce the response time for requests.
Which proposal should be pursued first?
A. Increase the vCPU resources of the API implementation
B. Modify the API client to split the problematic request into smaller, less-demanding requests
C. Increase the number of replicas of the API implementation
D. Throttle the APT client to reduce the number of requests per minute
Answer: A
Explanation:
Scenario Analysis:
The API implementation is experiencing high CPU usage (over 90%) during bursts of requests, which correlates with slow response times, as indicated by a 99th percentile response time greater than 120 seconds.
The API implementation is running on Anypoint Runtime Fabric with two non-clustered replicas and has a reserved vCPU of 1.0 and a vCPU limit of 2.0.
The high CPU usage during bursts suggests that the current resources may not be sufficient to handle peak loads.
Evaluating the Options:
Option A (Correct Answer): Increasing the vCPU resources for each replica would provide more processing power to handle high traffic volumes, potentially reducing the response time during spikes. Since the CPU usage is consistently high during bursts, this option directly addresses the resource bottleneck.
Option B: Modifying the API client to split requests may reduce individual request load but could be complex to implement on the client side and may not fully address the high CPU issue.
Option C: Increasing the number of replicas could help distribute the load; however, with a high CPU load on each replica, adding more replicas without increasing CPU resources may not fully resolve the

problem.
Option D: Throttling the client would reduce the number of requests, but this may not be acceptable if the client needs to maintain a high request rate. It also does not directly address the CPU limitations of the API implementation.
Conclusion:
Option A is the best choice as it addresses the root cause of high CPU usage by increasing the vCPU allocation, allowing the API to handle more requests efficiently. This should be pursued first before considering other options.
Refer to MuleSoft's documentation on Runtime Fabric and vCPU resource allocation for more details on optimizing API performance in high-demand environments.

NO.10 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

Answer: D
Explanation:
Correct Answe r: By refining the response definitions by adding the x-ratelimit-* response headers with description, type, and example
****
Reference:
https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling#response-headers
https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-policies#response-headers

NO.11 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

Answer: B
Explanation:
Correct Answe r: 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.

NO.12 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 Answe r: 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 Reference:
Bottom of Form
Top of Form

NO.13 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 Answe r: 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

NO.14 A company uses a hybrid Anypoint Platform deployment model that combines the EU control plane with customer-hosted Mule runtimes. After successfully testing a Mule API implementation in the Staging environment, the Mule API implementation is set with environment-specific properties and must be promoted to the Production environment. What is a way that MuleSoft recommends to configure the Mule API implementation and automate its promotion to the Production environment?

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

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

C. Modify the Mule API implementation's properties in Anypoint Exchange, then promote the Mule API implementation to the Production environment using Runtime Manager

D. Use an API policy to change properties in the Mule API implementation deployed to the Staging environment and another API policy to deploy the Mule API implementation to the Production environment

Answer: A

Explanation:

Correct Answe r: Bundle properties files for each environment into the Mule API implementation's deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs

****

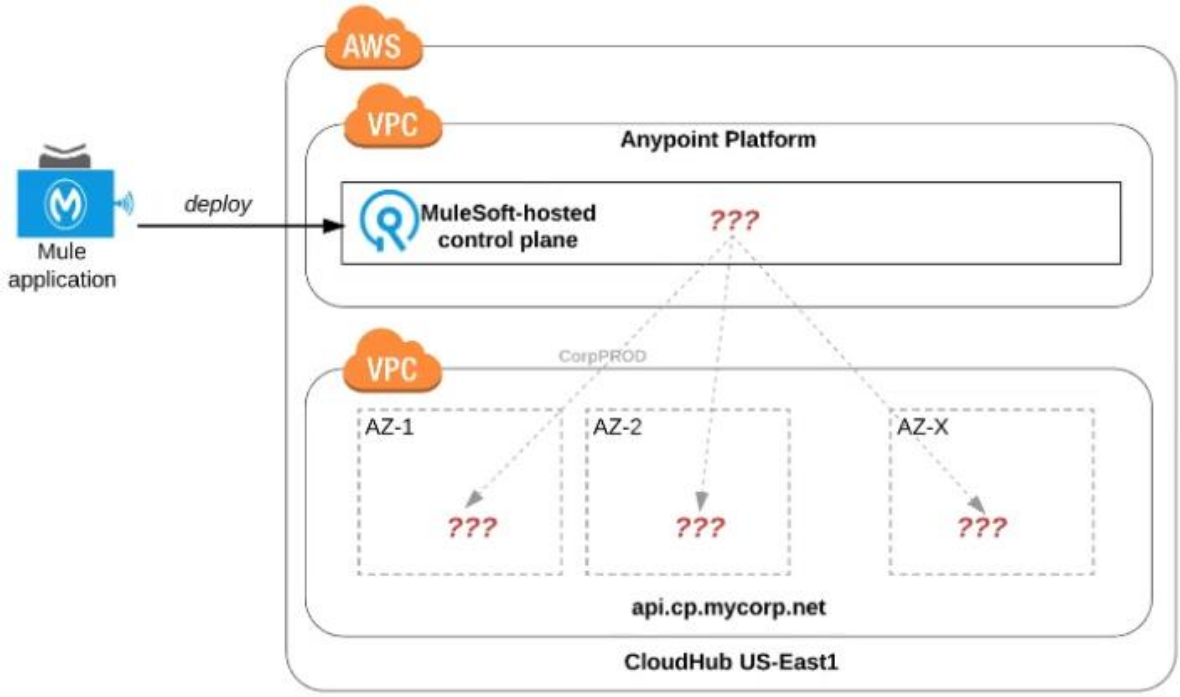
>> Anypoint Exchange is for asset discovery and documentation. It has got no provision to modify the properties of Mule API implementations at all.

>> API Manager is for managing API instances, their contracts, policies and SLAs. It has also got no provision to modify the properties of API implementations.

>> API policies are to address Non-functional requirements of APIs and has again got no provision to modify the properties of API implementations.

So, the right way and recommended way to do this as part of development practice is to bundle properties files for each environment into the Mule API implementation and just point and refer to respective file per environment.

NO.15 Refer to the exhibit.



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

A. Workers belonging to a given environment are assigned to the same AZ within that region

B. AZs are selected as part of the Mule application's deployment configuration

C. Workers are randomly distributed across available AZs within that region

D. An AZ is randomly selected for a Mule application, and all the Mule application's CloudHub workers are assigned to that one AZ

Answer: D

Explanation:

Correct Answe r: Workers are randomly distributed across available AZs within that region.

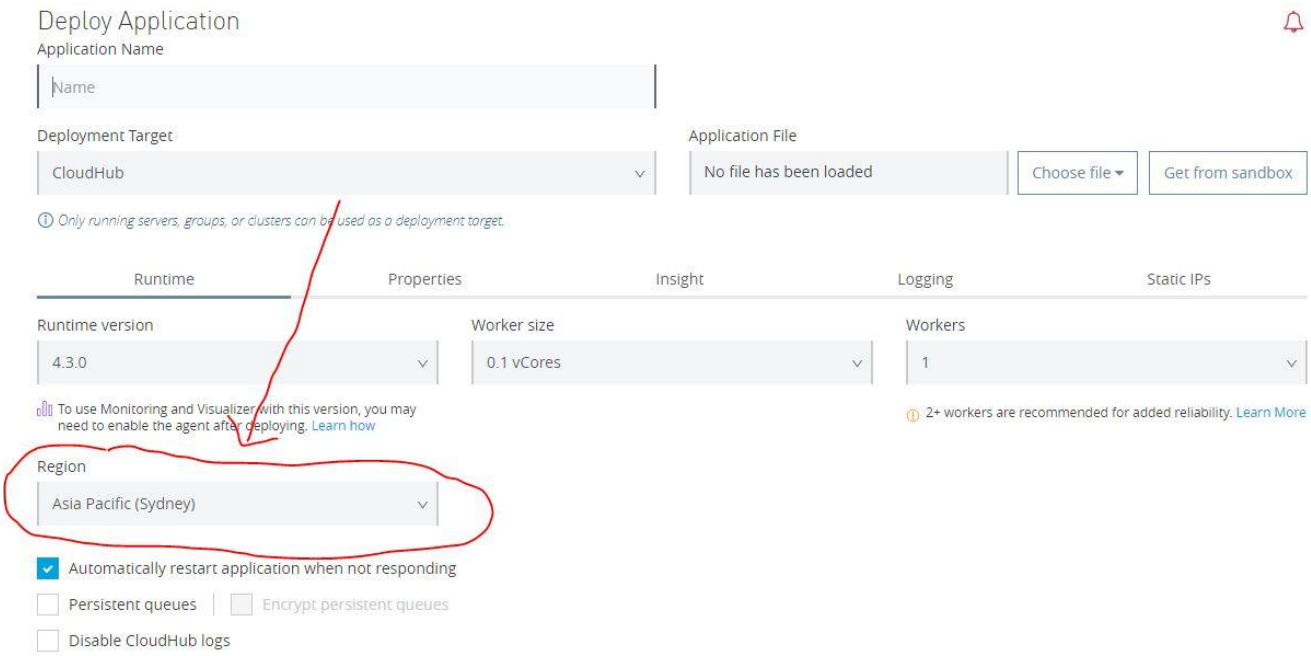
****

>> Currently, we only have control to choose which AWS Region to choose but there is no control at all using any configurations or deployment options to decide what Availability Zone (AZ) to assign to what worker.

>> There are NO fixed or implicit rules on platform too w.r.t assignment of AZ to workers based on environment or application.

>> They are completely assigned in random. However, cloudhub definitely ensures that HA is achieved by assigning the workers to more than on AZ so that all workers are not assigned to same AZ for same application.

Reference:



Bottom of Form

Top of Form