MCPA-LEVEL-1-MAINTENANCE Q&As

MuleSoft Certified Platform Architect - Level 1 MAINTENANCE

Pass Mulesoft MCPA-LEVEL-1-MAINTENANCE Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/mcpa-level-1-maintenance.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft Official Exam Center

QUESTION 1

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

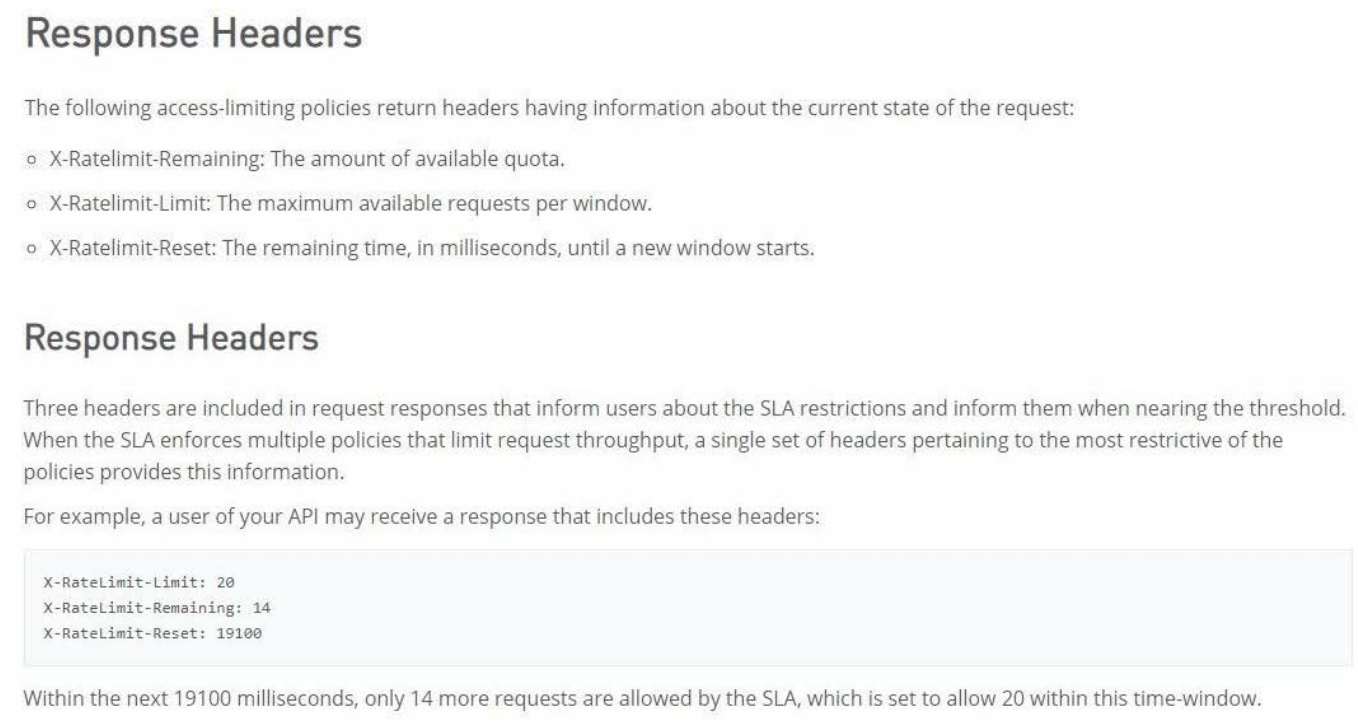
B. By refining the request definitions by adding a remaining Requests query parameter with description, type, and example

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

By refining the response definitions by adding the x-ratelimit-* response headers with description, type, and example
****



References: 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

QUESTION 2

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

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 3

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

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.

****

Reference: https://www.youtube.com/watch?v=U0FpYMnMjmM

QUESTION 4

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

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 5

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

A. The API policy Is defined In Runtime Manager as part of the API deployment to a Mule runtime, and then ONLY applied to the specific API Instance

B. The API policy Is defined In API Manager for a specific API Instance, and then ONLY applied to the specific API instance

C. The API policy Is defined in API Manager and then automatically applied to ALL API instances

D. The API policy is defined in API Manager, and then applied to ALL API instances in the specified environment

Correct Answer: B

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

****

>> Once our API specifications are ready and published to Exchange, we need to visit API Manager and register an API instance for each API. >> API Manager is the place where management of API aspects takes place like addressing

NFRs by enforcing policies on them.

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

>> One instance can have a set of API policies applied and another instance of same API can have different set of policies applied for some other purpose. >> These APIs and their instances are defined PER environment basis. So, one need

to manage them seperately in each environment.

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

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

would need to do that via API Manager only for a cherry picked instance in an environment.

So, based on these facts, right statement in the given choices is - "The API policy is defined in API Manager for a specific API instance, and then ONLY applied to the specific API instance".

Reference: https://docs.mulesoft.com/api-manager/2.x/latest-overview-concept

QUESTION 6

Select the correct Owner-Layer combinations from below options

A. 1. App Developers owns and focuses on Experience Layer APIs

2.

Central IT owns and focuses on Process Layer APIs

3.

LOB IT owns and focuses on System Layer APIs

B. 1. Central IT owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

App Developers owns and focuses on System Layer APIs

C. 1. App Developers owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

Central IT owns and focuses on System Layer APIs

Correct Answer: C

1.

App Developers owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

Central IT owns and focuses on System Layer APIs

References: https://blogs.mulesoft.com/biz/api/experience-api-ownership/ https://blogs.mulesoft.com/biz/api/process-api-ownership/ https://blogs.mulesoft.com/biz/api/system-api-ownership/

QUESTION 7

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

https://www.folkstalk.com/2019/11/mulesoft-integration-and-platform.html To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider

****

>> It is NOT necessary that single sign-on is required to sign in to Anypoint Platform because we are using an external Identity Provider for Client Management >> It is NOT necessary that all APIs managed by Anypoint Platform must be

protected by SAML 2.0 policies because we are using an external Identity Provider for Client Management

>> Not TRUE that the application network must include System APIs that interact with the Identity Provider because we are using an external Identity Provider for Client Management Only TRUE statement in the given options is - "To invoke

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

References:

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

QUESTION 8

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

API Consumer ********** >> API clients are piece of code or programs that use the client credentials of API consumer but does not directly interact with Anypoint Exchange to get the access >> API consumer is the one who should get registered and request access to API and then API client needs to use those client credentials to hit the APIs So, API consumer is the one who needs to request access on the API from Anypoint Exchange

QUESTION 9

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

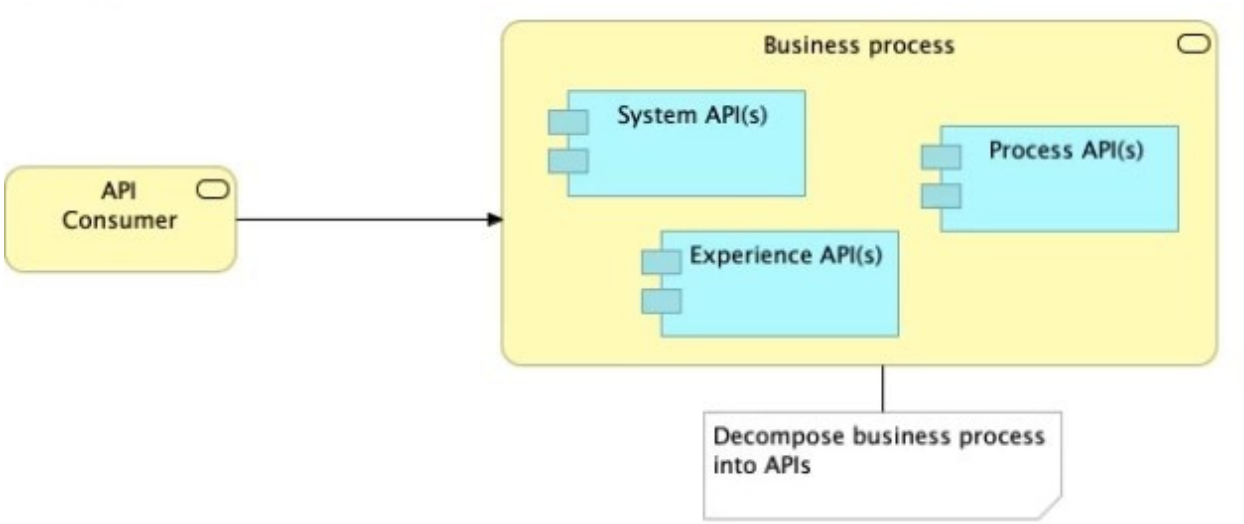
D. IPwhitellst

Correct Answer: D

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.

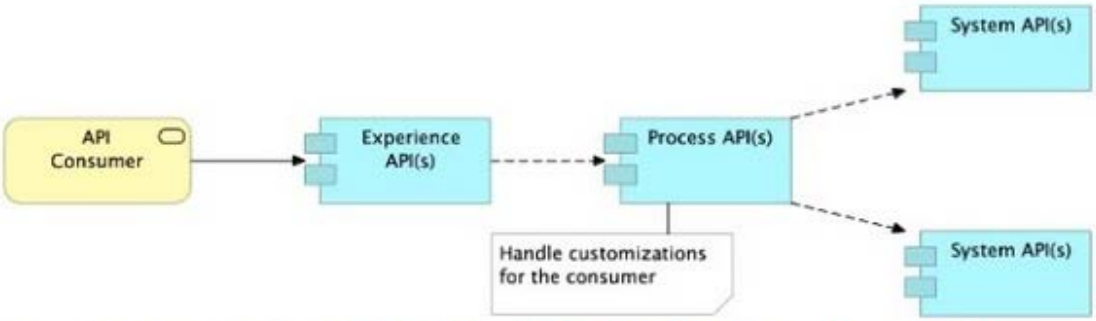
QUESTION 10

Refer to the exhibit.

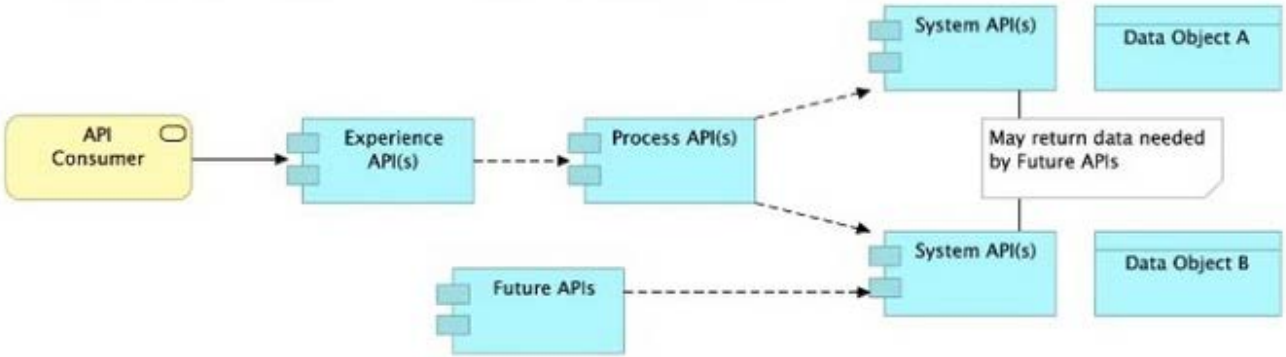


What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?

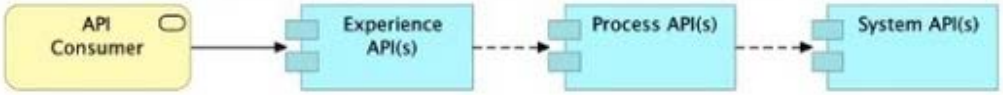
A. Handle customizations for the end-user application at the Process API level rather than the Experience API level



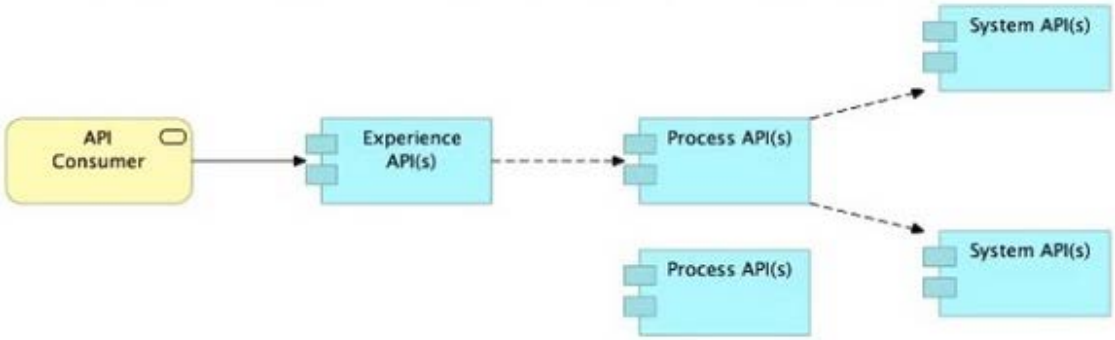
B. Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs



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

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

****

>> All customizations for the end-user application should be handled in "Experience API" only. Not in Process API

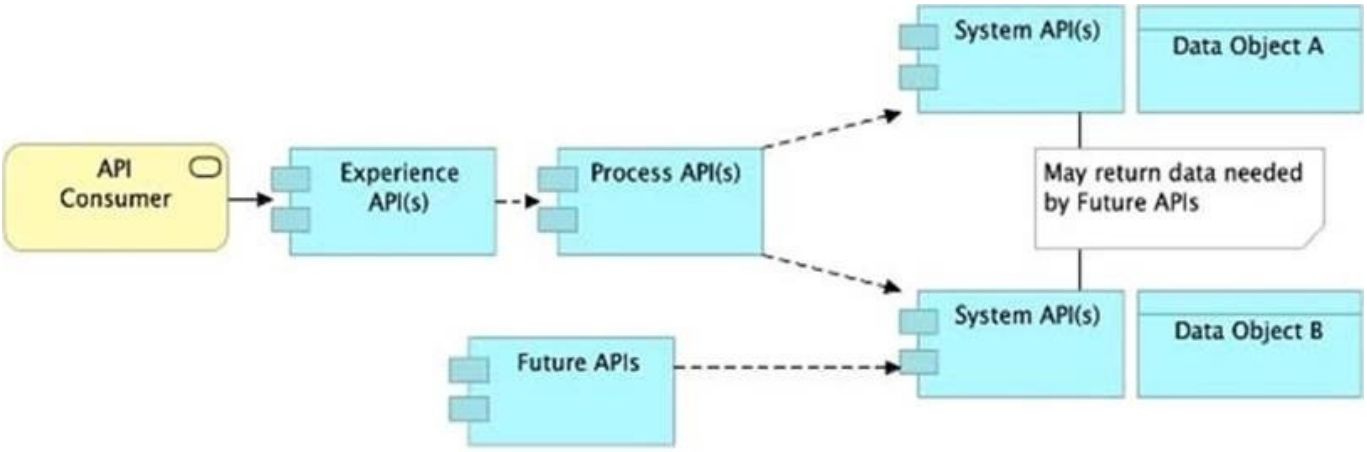
>> We should use tiered approach but NOT always by creating exactly one API for each of the 3 layers. Experience APIs might be one but Process APIs and System APIs are often more than one. System APIs for sure will be more than one

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 11

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

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

****

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

Reference: https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla- based-policies

QUESTION 12

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

A. 3.0.2

B. 4.0.0

C. 3.1.0

D. 3.0.1

Correct Answer: B

4.0.0

****

As per semver.org semantic versioning specification:

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

-MAJOR version when you make incompatible API changes.

-

MINOR version when you add functionality in a backwards compatible manner.

-

PATCH version when you make backwards compatible bug fixes. As per the scenario given in the question, the API implementation is completely changing its behavior. Although the format of the time is still being maintained as hh:mm:ss

and there is no change in schema w.r.t format, the API will start functioning different after this change as the times are going to come completely different. Example: Before the change, say, time is going as 09:00:00 representing the PST.

Now on, after the change, the same time will go as 18:00:00 as Central European Summer Time is 9 hours ahead of Pacific Time.

>> This may lead to some uncertain behavior on API clients depending on how they are handling the times in the API response. All the API clients need to be informed that the API functionality is going to change and will return in CEST

format. So, this considered as a MAJOR change and the version of API for this new change would be 4.0.0

QUESTION 13

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

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

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

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

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

Correct Answer: C

When development is already organized into several independent initiatives or groups

****

>> It would require lot of process effort in an organization to have a single C4E team coordinating with multiple already organized development teams which are into several independent initiatives. A single C4E works well with different teams

having at least a common initiative. So, in this scenario, federated C4E works well instead of centralized C4E.

QUESTION 14

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

A. FALSE

B. TRUE

Correct Answer: B

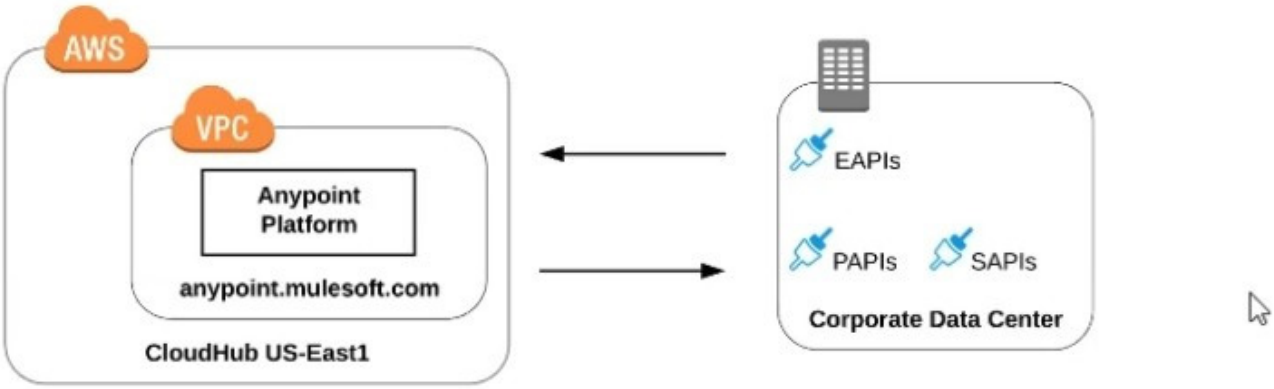
TRUE

****

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

QUESTION 15

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

API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane.

****

>> We CANNOT use Shared Load balancer to load balance APIs on customer hosted runtimes

- Load balancing

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

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

References: https://docs.mulesoft.com/runtime-manager/deployment-strategies#hybrid-deployments https://help.mulesoft.com/s/article/On-Premise-Runtimes-Disconnected-From-US-Control- Plane-June-18th-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-

RuntimeManager-May-29th-2018

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

Jun 19, 2018 . RCA

Content

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

Incident Description

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

==========

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

Jul 3, 2019 . RCA

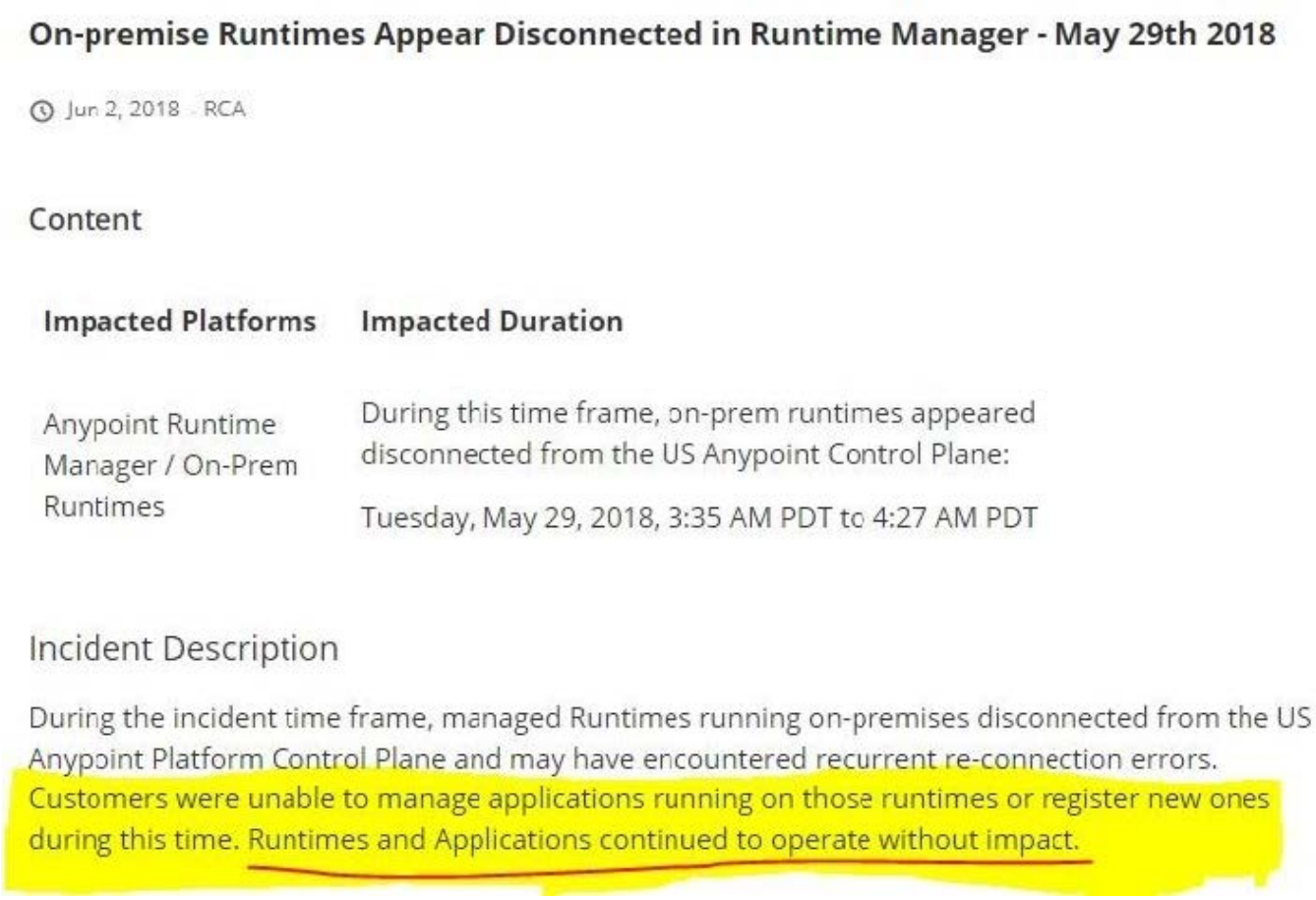
Content

Incident Summary

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

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

==========



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

Jun 2, 2018 · RCA

Content

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

Incident Description

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