Free Questions for MCPA-Level-1-Maintenance by dumpssheet

Shared by Vazquez on 12-12-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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?

Options:

A- A customer-hosted load balancer

B- The CloudHub shared load balancer

C- An API proxy

D- Runtime Manager autoscaling

Answer:

B

Explanation:

Correct Answer: 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 2

Question Type: MultipleChoice

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

Options:

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

Answer:

C

Explanation:

Correct Answer:

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.

****

Question 3

Question Type: MultipleChoice

Which of the following best fits the definition of API-led connectivity?

Options:

A- API-led connectivity is not just an architecture or technology but also a way to organize people and processes for efficient IT delivery in the organization

B- API-led connectivity is a 3-layered architecture covering Experience, Process and System layers

C- API-led connectivity is a technology which enabled us to implement Experience, Process and System layer based APIs

Answer:

A

Explanation:

Correct Answer: API-led connectivity is not just an architecture or technology but also a way

****

Question 4

Question Type: MultipleChoice

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?

Options:

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

Answer:

A

Explanation:

Correct Answer: In parallel, invoke the system API deployed to the primary environment and

****

>> 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 requirement

The 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 5

Question Type: MultipleChoice

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

Options:

A- TRUE

B- FALSE

Answer:

A

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 6

Question Type: **MultipleChoice**

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?

Options:

A- At the experience and process tiers

B- At the experience and system tiers

C- At the process and system tiers

D- At the experience, process, and system tiers

Answer:

C

Explanation:

Correct Answer: At the process and system tiers

****

>> Experience Layer APIs are modeled and designed exclusively for the end user's experience. So, the data models of experience layer vary based on the nature and type of such API consumer. For example, Mobile consumers will need light-weight data models to transfer with ease on the wire, where as web-based consumers will need detailed data models to render most of the info on web pages, so on. So, enterprise data models fit for the purpose of canonical models but not of good use for experience APIs.

>> That is why, EDMs should be used extensively in process and system tiers but NOT in experience tier.