MuleSoft

Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

    All examinations will be up to date.

* 24/7 Quality Support

    We will provide service round the clock.

* 100% Pass Rate

    Our guarantee that you will pass the exam.

* Unique Gurantee

    If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1
In which layer of API-led connectivity, does the business logic orchestration reside?

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

Answer: C

Explanation:

Correct Answer
Process Layer
****
>> Experience layer is dedicated for enrichment of end user experience. This layer is to meet the needs of different API clients/ consumers.
>> System layer is dedicated to APIs which are modular in nature and implement/ expose various individual functionalities of backend systems
>> Process layer is the place where simple or complex business orchestration logic is written by invoking one or many System layer modular APIs
So, Process Layer is the right answer.

NEW QUESTION 2
True or False. We should always make sure that the APIs being designed and developed are self-servable even if it needs more man-day effort and resources.

A. FALSE
B. TRUE

Answer: B

Explanation:

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

NEW QUESTION 3
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

Answer: B

Explanation:

Correct Answer
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.

NEW QUESTION 4
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

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.

NEW QUESTION 5
What is most likely NOT a characteristic of an integration test for a REST API implementation?

A. The test needs all source and/or target systems configured and accessible
B. The test runs immediately after the Mule application has been compiled and packaged
C. The test is triggered by an external HTTP request
D. The test prepares a known request payload and validates the response payload

Answer: B

Explanation:

Correct Answer
The test runs immediately after the Mule application has been compiled and packaged
****
>> Integration tests are the last layer of tests we need to add to be fully covered.
>> These tests actually run against Mule running with your full configuration in place and are tested from external source as they work in PROD.
>> These tests exercise the application as a whole with actual transports enabled. So, external systems are affected when these tests run.
So, these tests do NOT run immediately after the Mule application has been compiled and packaged.
FYI... Unit Tests are the one that run immediately after the Mule application has been compiled and packaged.

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

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

Answer: D

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

NEW QUESTION 7
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 Answer
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.

NEW QUESTION 8
A Mule application exposes an HTTPS endpoint and is deployed to the CloudHub Shared Worker Cloud. All traffic to that Mule application must stay inside the AWS VPC.
To what TCP port do API invocations to that Mule application need to be sent?

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

Answer: D

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

NEW QUESTION 9
An API client calls one method from an existing API implementation. The API implementation is later updated. What change to the API implementation would require the API client's invocation logic to also be updated?

A. When the data type of the response is changed for the method called by the API client
B. When a new method is added to the resource used by the API client
C. When a new required field is added to the method called by the API client
D. When a child method is added to the method called by the API client

Answer: C

Explanation:

Correct Answer
When a new required field is added to the method called by the API client
****
>> Generally, the logic on API clients need to be updated when the API contract breaks.
>> When a new method or a child method is added to an API , the API client does not break as it can still continue to use its existing method. So these two options are out.
>> We are left for two more where "datatype of the response if changed" and "a new required field is added".
>> Changing the datatype of the response does break the API contract. However, the question is insisting on the "invocation" logic and not about the response handling logic. The API client can still invoke the API successfully and receive the response but the response will have a different datatype for some field.
>> Adding a new required field will break the API's invocation contract. When adding a new required field, the API contract breaks the RAML or API spec agreement that the API client/API consumer and API provider has between them. So this requires the API client invocation logic to also be updated.

NEW QUESTION 10
What is true about the technology architecture of Anypoint VPCs?

A. The private IP address range of an Anypoint VPC is automatically chosen by CloudHub
B. Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can staywithin a private network
C. Each CloudHub environment requires a separate Anypoint VPC
D. VPC peering can be used to link the underlying AWS VPC to an on-premises (non AWS) private network

Answer: B

Explanation:
Correct Answer
Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network
****
>> The private IP address range of an Anypoint VPC is NOT automatically chosen by CloudHub. It is chosen by us at the time of creating VPC using thr CIDR blocks.
CIDR Block: The size of the Anypoint VPC in Classless Inter-Domain Routing (CIDR) notation.
For example, if you set it to 10.111.0.0/24, the Anypoint VPC is granted 256 IP addresses from 10.111.0.0 to 10.111.0.255.
Ideally, the CIDR Blocks you choose for the Anypoint VPC come from a private IP space, and should not overlap with any other Anypoint VPC's CIDR Blocks, or any CIDR Blocks in use in your corporate network.

that each CloudHub environment requires a separate Anypoint VPC. Once an Anypoint VPC is created, we can choose a same VPC by multiple environments.
However, it is generally a best and recommended practice to always have seperate Anypoint VPCs for Non-Prod and Prod environments.
>> We use Anypoint VPN to link the underlying AWS VPC to an on-premises (non AWS) private network. NOT VPC Peering.

NEW QUESTION 10
When must an API implementation be deployed to an Anypoint VPC?

A. When the API Implementation must invoke publicly exposed services that are deployed outside of CloudHub in a customer- managed AWS instance
B. When the API implementation must be accessible within a subnet of a restricted customer-hosted network that does not allow public access

C. When the API implementation must be deployed to a production AWS VPC using the Mule Maven plugin
D. When the API Implementation must write to a persistent Object Store

Answer: A

NEW QUESTION 14
A set of tests must be performed prior to deploying API implementations to a staging environment. Due to data security and access restrictions, untested APIs cannot be granted access to the backend systems, so instead mocked data must be used for these tests. The amount of available mocked data and its contents is sufficient to entirely test the API implementations with no active connections to the backend systems. What type of tests should be used to incorporate this mocked data?

A. Integration tests
B. Performance tests
C. Functional tests (Blackbox)
D. Unit tests (Whitebox)

Answer: D

Explanation:

Correct Answer
Unit tests (Whitebox)
****

NEW QUESTION 15
Question 10: Skipped
An API implementation returns three X-RateLimit-* HTTP response headers to a requesting API client. What type of information do these response headers indicate to the API client?

A. The error codes that result from throttling
B. A correlation ID that should be sent in the next request
C. The HTTP response size
D. The remaining capacity allowed by the API implementation

Answer: D

Explanation:

Correct Answer
The remaining capacity allowed by the API implementation.
****
>> Reference:
https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-policies#response-headers

NEW QUESTION 16
What correctly characterizes unit tests of Mule applications?

A. They test the validity of input and output of source and target systems

B. They must be run in a unit testing environment with dedicated Mule runtimes for the environment
C. They must be triggered by an external client tool or event source
D. They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity

Answer: D

Explanation:
Correct Answer
They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity.
****
Below TWO are characteristics of Integration Tests but NOT unit tests:
>> They test the validity of input and output of source and target systems.
>> They must be triggered by an external client tool or event source.
It is NOT TRUE that Unit Tests must be run in a unit testing environment with dedicated Mule runtimes for the environment.
MuleSoft offers MUnit for writing Unit Tests and they run in an embedded Mule Runtime without needing any separate/ dedicated Runtimes to execute them. They also do NOT need any external connectivity as MUnit supports mocking via stubs.
https://dzone.com/articles/munit-framework

NEW QUESTION 18
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

Answer: B

Explanation:

Correct Answer
Set a timeout of 100ms; that leaves 400ms for other two downstream 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.

NEW QUESTION 21
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

Answer: C

Explanation:
https://www.folkstalk.com/2019/11/mulesoft-integration-and-platform.html
Correct Answer
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/

NEW QUESTION 23
What should be ensured before sharing an API through a public Anypoint Exchange portal?

A. The visibility level of the API instances of that API that need to be publicly accessible should be set to public visibility
B. The users needing access to the API should be added to the appropriate role in Anypoint Platform
C. The API should be functional with at least an initial implementation deployed and accessible for users to interact with
D. The API should be secured using one of the supported authentication/authorization mechanisms to ensure that data is not compromised

Answer: A

Explanation:

Correct Answer
The visibility level of the API instances of that API that need to be publicly accessible should be set to public visibility.
****

NEW QUESTION 24
What is typically NOT a function of the APIs created within the framework called API-led connectivity?

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

Answer: A

Explanation:

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

NEW QUESTION 29
Refer to the exhibit.

A RAML definition has been proposed for a new Promotions Process API, and has been published to
Anypoint Exchange.
The Marketing Department, who will be an important consumer of the Promotions API, has important requirements and expectations that must be met.
What is the most effective way to use Anypoint Platform features to involve the Marketing Department in this early API design phase?
A) Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console

B) Organize a design workshop with the DBAs of the Marketing Department in which the database schema of the Marketing IT systems is translated into RAML

C) Use Anypoint Studio to Implement the API as a Mule application, then deploy that API implementation to CloudHub and ask the Marketing Department to interact with it

D) Export an integration test suite from API designer and have the Marketing Department execute the tests In that suite to ensure they pass

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

Answer: A

Explanation:
Correct Answer
Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console.
*** As per MuleSoft's IT Operating Model:
>> API consumers need NOT wait until the full API implementation is ready.
>> NO technical test-suites needs to be shared with end users to interact with APIs.
>> Anypoint Platform offers a mocking capability on all the published API specifications to Anypoint Exchange which also will be rich in documentation covering all details of API functionalities and working nature.
>> No needs of arranging days of workshops with end users for feedback.
API consumers can use Anypoint Exchange features on the platform and interact with the API using its mocking feature. The feedback can be shared quickly on the same to incorporate any changes.

NEW QUESTION 30
A REST API is being designed to implement a Mule application.
What standard interface definition language can be used to define REST APIs?

A. Web Service Definition Language(WSDL)
B. OpenAPI Specification (OAS)
C. YAML
D. AsyncAPI Specification

Answer: B

NEW QUESTION 32
What Anypoint Connectors support transactions?

A. Database, JMS, VM
B. Database, 3MS, HTTP
C. Database, JMS, VM, SFTP
D. Database, VM, File

Answer: A

NEW QUESTION 36
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 APIs 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 API producer should be requested to run the old version in parallel with the new one
B. The API producer should be contacted to understand the change to existing functionality
C. The API client code only needs to be changed if it needs to take advantage of the new features
D. The API clients need to update the code on their side and need to do full regression

Answer: C

NEW QUESTION 40
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

Answer: B

Explanation:

Correct Answers
* 1. Each modern API must be treated like a product and designed for a particular target audience (for instance mobile app developers)
****
Bottom of Form Top of Form

NEW QUESTION 45
What is a typical result of using a fine-grained rather than a coarse-grained API deployment model to implement a given business process?

A. A decrease in the number of connections within the application network supporting the business process
B. A higher number of discoverable API-related assets in the application network
C. A better response time for the end user as a result of the APIs being smaller in scope and complexity
D. An overall tower usage of resources because each fine-grained API consumes less resources

Answer: B

Explanation:

Correct Answer
A higher number of discoverable API-related assets in the application network.
****
>> We do NOT get faster response times in fine-grained approach when compared to coarse-grained approach.
>> In fact, we get faster response times from a network having coarse-grained APIs compared to a network having fine-grained APIs model. The reasons are below.
Fine-grained approach:
* 1. will have more APIs compared to coarse-grained
* 2. So, more orchestration needs to be done to achieve a functionality in business process.
* 3. Which means, lots of API calls to be made. So, more connections will needs to be established. So, obviously more hops, more network i/o, more number of integration points compared to coarse-grained approach where fewer APIs with bulk functionality embedded in them.
* 4. That is why, because of all these extra hops and added latencies, fine-grained approach will have bit more response times compared to coarse-grained.
* 5. Not only added latencies and connections, there will be more resources used up in fine-grained approach due to more number of APIs.
That's why, fine-grained APIs are good in a way to expose more number of resuable assets in your network and make them discoverable. However, needs more maintenance, taking care of integration points, connections, resources with a little compromise w.r.t network hops and response times.

NEW QUESTION 46
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

Answer: B

Explanation:

Correct Answer
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".

NEW QUESTION 47
Refer to the exhibit.

A developer is building a client application to invoke an API deployed to the STAGING environment that is governed by a client ID enforcement policy.
What is required to successfully invoke the API?

A. The client ID and secret for the Anypoint Platform account owning the API in the STAGING environment
B. The client ID and secret for the Anypoint Platform account's STAGING environment
C. The client ID and secret obtained from Anypoint Exchange for the API instance in the STAGING environment
D. A valid OAuth token obtained from Anypoint Platform and its associated client ID and secret

Answer: C

Explanation:
Correct Answer
The client ID and secret obtained from Anypoint Exchange for the API instance in the STAGING environment
****
>> We CANNOT use the client ID and secret of Anypoint Platform account or any individual environments for accessing the APIs
>> As the type of policy that is enforced on the API in question is "Client ID Enforcment Policy", OAuth token based access won't work.
Right way to access the API is to use the client ID and secret obtained from Anypoint Exchange for the API instance in a particular environment we want to work on.
References:
Managing API instance Contracts on API Manager https://docs.mulesoft.com/api-manager/1.x/request-access-to-api-task https://docs.mulesoft.com/exchange/to-request-access https://docs.mulesoft.com/api-manager/2.x/policy-mule3-client-id-based-policies

NEW QUESTION 50
......

Relate Links

100% Pass Your MCPA-Level-1 Exam with Exambible Prep Materials

https://www.exambible.com/MCPA-Level-1-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/