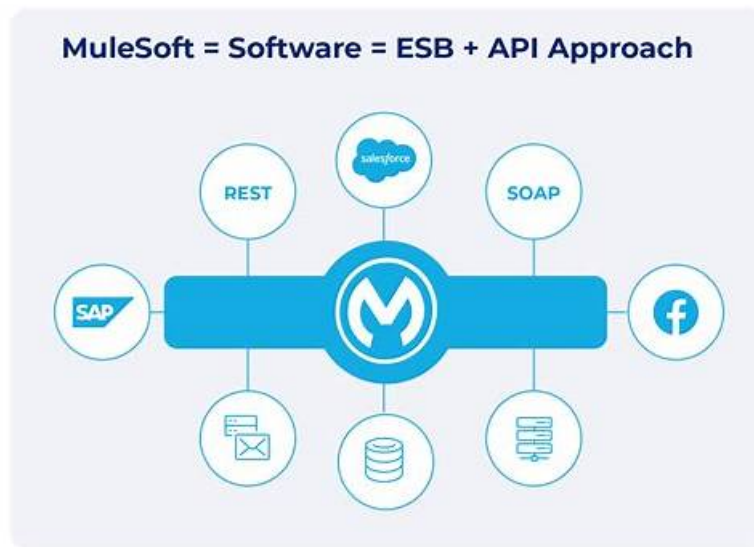


MuleSoft-Integration-Architect-I Testing Engine, MuleSoft-Integration-Architect-I Testengine



Wenn Sie die Salesforce MuleSoft-Integration-Architect-I (Salesforce Certified MuleSoft Integration Architect I) Zertifizierungsprüfung bestehen wollen, hier kann Fast2test Ihr Ziel erreichen. Wir sind uns im Klar, dass Sie die die MuleSoft-Integration-Architect-I Zertifizierungsprüfung wollen. Unser Versprechen sind die wissenschaftliche und qualitativ hochwertige Prüfungsfragen und Antworten zur MuleSoft-Integration- Architect-I Zertifizierungsprüfung.

Salesforce MuleSoft-Integration-Architect-I Prüfungsplan:

Thema	Einzelheiten
Thema 1	Designing Architecture Using Integration Paradigms: This topic focuses on creating high-level integration architectures using various paradigms. It includes API-led connectivity, web APIs and HTTP, event-driven APIs, and message brokers, and designing Mule application using messaging patterns and technologies.
Thema 2	Designing Integration Solutions to Meet Performance Requirements: This topic covers meeting performance and capacity goals, using streaming features, and processing large message sequences.
Thema 3	Designing Integration Solutions to Meet Reliability Requirements: It includes selecting alternatives to traditional transactions, recognizing the purpose of various scopes and strategies, differentiating disaster recovery and high availability, and using local and XA transactions.
Thema 4	Designing Automated Tests for Mule Applications: This topic covers unit test suites, and scenarios for integration and performance testing.
Thema 5	Designing Integration Solutions to Meet Persistence Requirements: It addresses the usage of VM queues and connectors, object stores and services, and stateful components configured with object stores.
Thema 6	Designing and Developing Mule Applications: It includes selecting application properties, using fundamental features, designing with core routers, understanding the Salesforce Connector, and leveraging core connectors.
Thema 7	- Applying DevOps Practices and Operating Integration Solutions: Its sub-topics are related to designing CI - CD pipelines with MuleSoft plugins, automating interactions with Anypoint Platform, designing logging configurations, and identifying Anypoint Monitoring features.

Thema 8	• Designing for the Runtime Plane Technology Architecture: It includes analyzing Mule runtime clusters, designing solutions for CloudHub, choosing Mule runtime domains, leveraging Mule 4 class loader isolation, and understanding the reactive event processing model.
Thema 9	• Initiating Integration Solutions on Anypoint Platform: Summarizing MuleSoft Catalyst and Catalyst Knowledge Hub, differentiating between functional and non-functional requirements, selecting features for designing and managing APIs, and choosing deployment options are its sub-topics.

>> **MuleSoft-Integration-Architect-I Testing Engine** <<

MuleSoft-Integration-Architect-I Testengine, MuleSoft-Integration-Architect-I Fragenpool

Die Prüfungsfragen und Antworten von Fast2test Salesforce MuleSoft-Integration-Architect-I bieten Ihnen alles, was Sie zur Prüfungsvorbereitung brauchen. Für Salesforce MuleSoft-Integration-Architect-I Prüfung können Sie auch Lernhilfe aus anderen Websites oder Büchern finden. Aber Hauptsache ist es, sie müssen logisch verbinden. Unsere Salesforce MuleSoft-Integration-Architect-I Zertifizierungsantworten ermöglichen es Ihnen, mühelos die Prüfung zum ersten Mal zu bestehen. Zugleich können Sie auch viele wertvolle Zeit sparen.

Salesforce Certified MuleSoft Integration Architect I MuleSoft-Integration-Architect-I Prüfungsfragen mit Lösungen (Q256-Q261):

256. Frage

A company is planning to extend its Mule APIs to the Europe region. Currently all new applications are deployed to Cloudhub in the US region following this naming convention

{API name} - {environment}. for example, Orders-SAPI-dev, Orders-SAPI-prod etc.

Considering there is no network restriction to block communications between API's, what strategy should be implemented in order to apply the same new API's running in the EU region of CloudHub as well to minimize latency between API's and target users and systems in Europe?

- A. Set region property to Europe (eu-de) in API manager for all the mule application Change the naming convention to {API name} - {environment} - {region} and communicate this change to the consuming applications and users
- B. Set region property to Europe (eu-de) in API manager for all the mule application No need to change the naming convention
- C. Set region property to Europe (eu-de) in runtime manager for all the mule application No need to change the naming convention
- **D. Set region property to Europe (eu-de) in runtime manager for all the mule application Change the naming convention to {API name} - {environment} - {region} and communicate this change to the consuming applications and users**

Antwort: D

Begründung:

To extend Mule APIs to the Europe region and ensure minimal latency between APIs and target users and systems in Europe, it is important to set the region property correctly and update the naming convention. Here's the detailed approach:

* Setting the Region Property:

* Runtime Manager Configuration: In MuleSoft's Runtime Manager, you can deploy Mule applications to specific regions. For the European deployment, set the region property to Europe (eu-de) for all Mule applications. This ensures that the applications are physically hosted in the European data centers, reducing latency for European users.

* Go to Runtime Manager in Anypoint Platform.

* Select the application to deploy.

* Choose the Region setting and select Europe (eu-de).

* Updating Naming Convention:

* Naming Convention Update: Change the naming convention to include the region in the application names. The new convention should be {API name} - {environment} - {region} (e.g., Orders-SAPI-dev-eu, Orders-SAPI-prod-eu). This helps in distinguishing applications deployed in different regions.

* Update the naming convention during deployment.

* Ensure the new naming convention is documented and communicated to all stakeholders.

* Communicating the Change:

* Stakeholder Communication: Inform all consuming applications and users about the updated naming convention. This ensures that they update their references and configurations accordingly to avoid any disruptions in service.

* Documentation: Update any internal documentation to reflect the new naming convention and deployment regions.

* Benefits:

* Reduced Latency: By deploying the applications in the Europe region, the data proximity to European users will be improved, resulting in lower latency.

* Clear Identification: The updated naming convention helps in easily identifying the environment and region of deployment, making management and troubleshooting easier.

References:

* MuleSoft Documentation on Deploying to CloudHub

* MuleSoft Documentation on Managing Applications in Runtime Manager

257. Frage

The AnyAirline organization's passenger reservations center is designing an integration solution that combines invocations of three different System APIs (bookFlight, bookHotel, and bookCar) in a business transaction. Each System API makes calls to a single database.

The entire business transaction must be rolled back when at least one of the APIs fails.

What is the most idiomatic (used for its intended purpose) way to integrate these APIs in near real-time that provides the best balance of consistency, performance, and reliability?

- A. Implement eXtended Architecture (XA) transactions between the API implementations Coordinate between the API implementations using a Saga pattern Implement caching in each API implementation to improve performance
- **B. Implement local transactions in each API implementation**
Coordinate between the API implementations using a Saga pattern
Apply various compensating actions depending on where a failure occurs
- C. Implement local transactions within each API implementation
Configure each API implementation to also participate in the same eXtended Architecture (XA) transaction Implement caching in each API implementation to improve performance
- D. Implement an eXtended Architecture (XA) transaction manager in a Mule application using a Saga pattern Connect each API implementation with the Mule application using XA transactions Apply various compensating actions depending on where a failure occurs

Antwort: B

258. Frage

An organization has deployed both Mule and non-Mule API implementations to integrate its customer and order management systems. All the APIs are available to REST clients on the public internet.

The organization wants to monitor these APIs by running health checks: for example, to determine if an API can properly accept and process requests. The organization does not have subscriptions to any external monitoring tools and also does not want to extend its IT footprint.

What Anypoint Platform feature provides the most idiomatic (used for its intended purpose) way to monitor the availability of both the Mule and the non-Mule API implementations?

- **A. Anypoint Visualizer**
- B. API Manager
- C. API Functional Monitoring
- D. Runtime Manager

Antwort: A

Begründung:

Reference: <https://docs.mulesoft.com/visualizer/>

259. Frage

A Mule application is deployed to a cluster of two(2) customer-hosted Mule runtimes. Currently the node name Alice is the primary node and node named bob is the secondary node. The mule application has a flow that polls a directory on a file system for new

files.

The primary node Alice fails for an hour and then restarted.

After the Alice node completely restarts, from what node are the files polled, and what node is now the primary node for the cluster?

- A. Files are polled from Bob node
Alice is now the primary node
- B. Files are polled from Alice node
Alice is now the primary node
- **C. Files are polled from Bob node
Bob is now the primary node**
- D. Files are polled from Alice node
Bob is now the primary node

Antwort: C

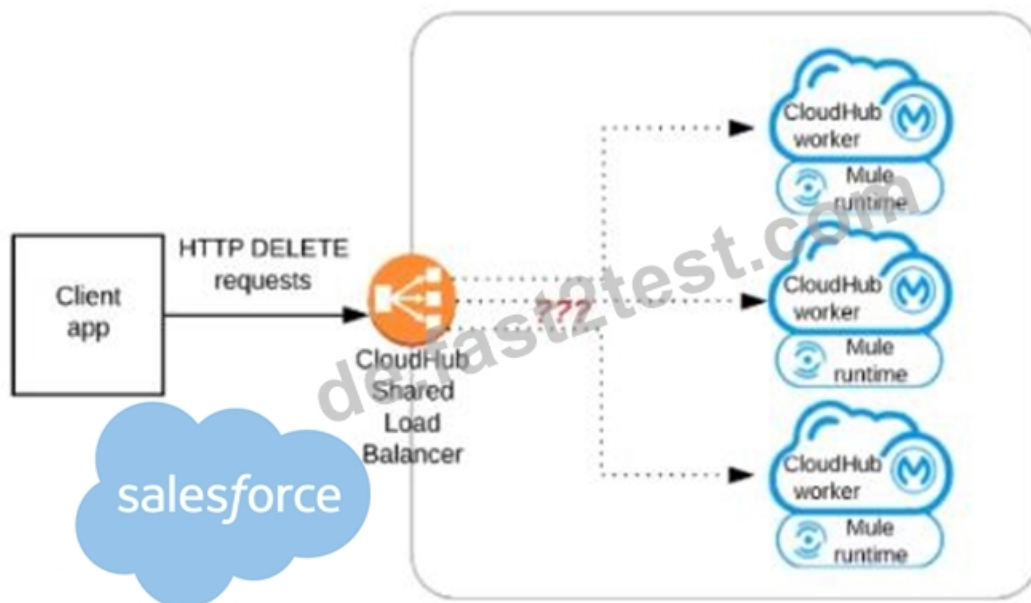
Begründung:

* Mule High Availability Clustering provides basic failover capability for Mule. * When the primary Mule Runtime becomes unavailable, for example, because of a fatal JVM or hardware failure or it's taken offline for maintenance, a backup Mule Runtime immediately becomes the primary node and resumes processing where the failed instance left off. * After a system administrator recovers a failed Mule Runtime server and puts it back online, that server automatically becomes the backup node. In this case, Alice, once up, will become backup -----

----- Reference: <https://docs.mulesoft.com/mule-runtime/4.3/hadr-guide> So correct choice is : Files are polled from Bob node Bob is now the primary node

260. Frage

Refer to the exhibit.



A Mule application has an HTTP Listener that accepts HTTP DELETE requests. This Mule application is deployed to three CloudHub workers under the control of the CloudHub Shared Load Balancer.

A web client makes a sequence of requests to the Mule application's public URL.

How is this sequence of web client requests distributed among the HTTP Listeners running in the three CloudHub workers?

- **A. Each request is routed to ONE ARBITRARY CloudHub worker out of ALL three CloudHub workers**
- B. Each request is routed to the PRIMARY CloudHub worker in the PRIMARY Availability Zone (AZ)
- C. Each request is routed (scattered) to ALL three CloudHub workers at the same time
- D. Each request is routed to ONE ARBITRARY CloudHub worker in the PRIMARY Availability Zone (AZ)

Antwort: A

Begründung:

Correct behavior is Each request is routed to ONE ARBITRARY CloudHub worker out of ALL three CloudHub workers

• • • • •

MuleSoft-Integration-Architect-I Testengine: <https://de.fast2test.com/MuleSoft-Integration-Architect-I-premium-file.html>

- [illegible]