

MuleSoft.MCIA-Level-1-JPN.v2023-02-08.q60

試験コード : MCIA-Level-1-JPN
試験名称 : MuleSoft Certified Integration Architect - Level 1 (MCIA-Level-1日本語版)
認証ベンダー : MuleSoft
無料問題の数 : 60
バージョン : v2023-02-08
ページの閲覧量 : 395
問題集の閲覧量 : 7821

<https://www.jpnsiken.com/shiken/MuleSoft.MCIA-Level-1-JPN.v2023-02-08.q60.html>

質問: 1

銀行会社は、オンライン ビジネス用の新しい API セットを開発しています。重要な API の 1 つは、システム API であるマスター ルックアップ API です。このマスター ルックアップ API は、永続的なオブジェクト ストアを使用します。この API は、マスター ルックアップ データを提供するために、他のすべての API で使用されます。

マスター ルックアップ API は、キャッシュするマスター データが大量にあるため、それぞれ 0.1 vCore の 2 つの cloudhub ワーカーにデプロイされます。マスター ルックアップ データは、キーと値のペアとして保存されます。キーがキャッシュ内に見つからない場合、キャッシュは更新されます。

パフォーマンス テストを行ったところ、マスター ルックアップ API は、マスター ルックアップ データを取得するためのデータベース クエリの実行により、応答時間が長くなることが観察されました。

このパフォーマンスの問題により、オンライン ビジネスの稼働は保留されており、銀行に財務上の損失をもたらす可能性があります。

統合アーキテクトとして、パフォーマンスの問題を解決するために次のオプションのうちどれを提案しますか？

- A. 追加の Cloudhub ワーカーを追加して、追加の容量を提供します
- B. マスター ルックアップ API のすべての GET エンドポイントに HTTP キャッシュ ポリシーを実装する
- C. マスター ルックアップ API のすべての GET エンドポイントに HTTP キャッシュ ポリシーを実装し、ロックを実装してオブジェクト ストアへのアクセスを同期します。
- D. vCore のサイズを 0.1 vCore から 0.2 vCore にアップグレードします。

正解: C ([コメントを发表する](#))

質問: 2

組織は、サーバル データベース操作と他のいくつかのコネクタとの間のオール オア ナッシング トランザクションをサポートするミュール アプリケーションを設計しています。これにより、コネクタのいずれかに問題が発生した場合にすべてがロールバックされます。トランザクション。

- A. VM
- B. Anypoint MQ
- C. SFTP
- D. オブジェクトストア

正解: **A** ([コメントを发表する](#))

Correct answer is VM VM support Transactional Type. When an exception occur, The transaction rolls back to its original state for reprocessing. This feature is not supported by other connectors.

Here is additional information about Transaction management:

質問: **3**

Mule アプリケーションは、HTTP 要求操作を使用して外部 API を呼び出します。

外部 API は、ステータス コードを適切に使用するために HTTP 仕様に従います。

外部 API からの HTTP 要求操作に 3xx ステータス コードが返される場合、考えられる原因は何ですか？

- A.** 外部 API で要求が受け入れられませんでした
- B.** リクエストは外部 API によって別の URL にリダイレクトされました
- C.** リクエストは外部 API によって受信されませんでした
- D.** リクエストは外部 API によって受け入れられました

正解: ([正解を表示します](#))

3xx HTTP status codes indicate a redirection that the user agent (a web browser or a crawler) needs to take further action when trying to access a particular resource.

質問: **4**

展示を参照してください。

ビジネス プロセスには、HTTP を介して互いに非同期に対話する 2 つの API が含まれます。各 API は Mule アプリケーションとして実装されます。API 1 は最初の HTTP リクエストを受信し、API 2 を (ファイア アンド フォーゲット方式で) 呼び出します。API 2 は、処理が完了すると、API 1 にコールバックして、非同期プロセスの完了を通知します。

各 API は、複数の冗長 Mule ランタイムと個別のロード バランサにデプロイされ、個別のネットワーク ゾーンにデプロイされます。

ネットワーク アーキテクチャで、API 1 と API 2 の間の上記の相互作用を有効にするには、ファイアウォール ルールをどのように構成する必要がありますか？

- A.** 証明書が両方の API で使用されることを承認するには
- B.** 各 API の Mule Runtimes およびネットワーク ゾーンから他の API のロード バランサへの通信を有効にするには
- C.** 両方の API の Mule ランタイム間の直接双方向通信を開く
- D.** 各 API で使用するロードバランサー間の通信を許可する

正解: ([正解を表示します](#))

* If your API implementation involves putting a load balancer in front of your APIkit application, configure the load balancer to redirect URLs that reference the baseUri of the application directly. If the load balancer does not redirect URLs, any calls that reach the load balancer looking for the application do not reach their destination.

* When you receive incoming traffic through the load balancer, the responses will go out the same way. However, traffic that is originating from your instance will not pass through the load balancer. Instead, it is

sent directly from the public IP address of your instance out to the Internet. The ELB is not involved in that scenario.

* The question says "each API is deployed to multiple redundant Mule runtimes", that seems to be a hint for self hosted Mule runtime cluster. Set Inbound allowed for the LB, outbound allowed for runtime to request out.

* Hence correct way is to enable communication from each API's Mule Runtimes and Network zone to the load balancer of the other API. Because communication is asynchronous one

質問: 5

その Mule アプリケーションによって公開されたエンドポイントに Anypoint Platform によって適用される API ポリシーを持つことができる Mule アプリケーションはどれですか？

- A. HTTP/1x 経由でリクエストを受け入れる Mule アプリケーション
- B. TCP 経由で JSON 要求を受け入れるが、応答を提供する必要がない Mule アプリケーション。
- C. WebSocket 経由で JSON 要求を受け入れる Mule アプリケーション
- D. HTTP/2 経由で gRPC リクエストを受け入れる Mule アプリケーション

正解: **A** ([コメントを发表する](#))

* HTTP/1.1 keeps all requests and responses in plain text format.

* HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers. It came into use in 2015, and offers several methods to decrease latency, especially when dealing with mobile platforms and server-intensive graphics and videos

* Currently, Mule application can have API policies only for Mule application that accepts requests over HTTP/1x

質問: 6

ある組織が、REST API を実装する Mule アプリケーションを開発しました。Mule アプリケーションは、顧客がホストする Mule ランタイムのクラスタにデプロイされます。

インバウンド API 要求をクラスタの Mule ランタイム全体に分散するために、顧客が提供する必要がある追加のインフラストラクチャ コンポーネントは何ですか？

- A. メッセージブローカー
- B. HTTP ロード バランサー
- C. データベース
- D. オブジェクトストア

正解: ([正解を表示します](#))

Correct answer is An HTTP Load Balancer.

Key thing to note here is that we are deploying application to customer hosted Mule runtime. This means we will need load balancer to route the requests to different instances of the cluster.

Rest all options are distractors and their requirement depends on project use case.

質問: 7

統合ミュート アプリケーションは、CSV ファイルから行のリストを使用して処理します。CSV ファイルとまったく同じ順序で、各行を CSV ファイルから読み取って検証し、行データを JMS キューに送信する必要があります。

行のいずれかの処理ステップが失敗した場合、その行のログ エントリを書き込む必要がありますが、他の行の処理に影響を与えてはなりません。

上記の要件を実装する際に最も慣用的な (意図した目的に従って使用される) Mute コンポーネントの組み合わせはどれですか？

- A. Scatter-Gather コンポーネント On Error Continue スコープ
- B. 最初の VM コネクタ 正常なスコープ オン エラー スコープの伝播
- C. For Each scope On Error 継続スコープ
- D. Async スコープ On Error Propagate スコープ

正解: ([正解を表示します](#))

* On Error Propagate halts execution and sends error to the client. In this scenario it's mentioned that "processing of other rows must not be affected" so Option B and C are ruled out.

* Scatter gather is used to club multiple responses together before processing. In this scenario, we need sequential processing. So option A is out of choice.

* Correct answer is For Each scope & On Error Continue scope Below requirement can be fulfilled in the below way

1) Using For Each scope , which will send each row from csv file sequentially. each row needs to be sent sequentially as requirement is to send the message in exactly the same way as it is mentioned in the csv file

2) Also other part of requirement is if any processing step for a row fails then it should log an error but should not affect other record processing . This can be achieved using On error Continue scope on these set of activities. so that error will not halt the processing. Also logger needs to be added in error handling section so that it can be logged.

* Attaching diagram for reference. Here it's try scope, but similar would be the case with For Each loop.

質問: 8

システム API EmployeeSAPI を使用して、基礎となる SQL データベースから従業員のデータをフェッチします。

アーキテクトは、データベースによって処理される冗長なトランザクションの数を最小限に抑えるために、従業員の安定した更新がある場合にのみデータベースにクエリを実行するか、キャッシュされた応答を返すようにキャッシュ戦略を設計する必要があります。

アーキテクトはキャッシングの目的を達成するために何をしなければなりませんか？

- A. 1 時間ごとに固定の頻度でスケジューラを使用して、無効化キャッシュ フローをトリガーします オブジェクトストア キャッシュ戦略を使用し、有効期限間隔を 1 時間に設定します
- B. 従業員テーブル呼び出し無効化キャッシュに on table ルールを使用し、前述の新しい従業員データをキャッシュします オブジェクトストア キャッシング戦略を使用し、有効期限間隔を 1 時間に設定します
- C. employees テーブルで On Table Row を使用し、invalidate キャッシュを呼び出します
オブジェクトストアのキャッシュ戦略と有効期限を使用して空にする

D. 1 時間ごとに一定の頻度でスケジューラを使用し、無効化キャッシュ フローをトリガーします。オブジェクトストアのキャッシュ戦略と有効期限を使用して空にします。

正解: ([正解を表示します](#))

質問: 9

顧客は、マッピングされた診断コンテキスト (MDC) とログ変数を使用してログを充実させ、ログにより多くのコンテキストを提供することで追跡を改善したいと考えています。

顧客は、メッセージ処理のスループットを向上させ、待ち時間を短縮することも望んでいます。

Mulesoft 統合アーキテクトとして、お客様がこれらの要件を満たすために何を実装する必要があるかをアドバイスできますか？

A. INFO DEBUG または Trace レベルで同期ログを使用し、log4j2.xml 構成ファイルで [%MDC] を使用してパターン レイアウトを使用し、ログ変数を構成します。

B. 同期ロギングを使用し、log4j2.xml 構成ファイルの [%MDC] でパターン レイアウトを使用してから、ロギング変数を構成します。

C. INFO 以上のレベルで async ロガーを使用し、log4j2.xml 構成ファイルで [%MDC] を使用してパターン レイアウトを使用し、ログ変数を構成します。

D. DEBUG または TRACE と同等のレベルで async ロガーを使用し、パターンを使用します。

log4j2.xml 構成ファイルで [%MDC] を使用してレイアウトし、ログ変数を構成します。

正解: **C** ([コメントを發表する](#))

質問: 10

CloudHub Dedicated Load Balancer を使用する必要があるのはどのような条件ですか？

A. 同じ Mule アプリケーションの個別のデプロイ間でリージョン間の負荷分散が必要な場合

B. 顧客がホストする Mule ランタイムにデプロイされた API 実装にカスタム DNS 名が必要な場合

C. 複数の CloudHub ワーカーにまたがる API 呼び出しの負荷を分散する必要がある場合

D. API 実装と API クライアント間でサーバー側の負荷分散 TLS 相互認証が必要な場合

正解: **D** ([コメントを發表する](#))

Correct answer is When server-side load-balanced TLS mutual authentication is required between API implementations and API clients CloudHub dedicated load balancers (DLBs) are an optional component of Anypoint Platform that enable you to route external HTTP and HTTPS traffic to multiple Mule applications deployed to CloudHub workers in a Virtual Private Cloud (VPC). Dedicated load balancers enable you to: * Handle load balancing among the different CloudHub workers that run your application. * Define SSL configurations to provide custom certificates and optionally enforce two-way SSL client authentication. * Configure proxy rules that map your applications to custom domains. This enables you to host your applications under a single domain

質問: 11

展示を参照してください。

Mule アプリケーションは、SOAP Web サービスをそのクライアントに公開するように設計されています。

Web サービスがそのクライアントと交換することが期待されるデータ構造を定義するために、Web サービスのインターフェース定義内で通常使用される言語はどれですか？

- A. WSDL
- B. XSD
- C. JSON スキーマ
- D. ラミ

正解: ([正解を表示します](#))

Correct: XSD In this approach to developing a web service, you begin with

Reference: https://www.w3schools.com/xml/schema_intro.asp

質問: 12

どの Anypoint Connector がトランザクションをサポートしていますか？

- A. データベース、JMS、VM
- B. データベース、3MS、HTTP
- C. データベース、JMS、VM、SFTP
- D. データベース、VM、ファイル

正解: A ([コメントを发表する](#))

Below Anypoint Connectors support transactions * JMS - Publish - Consume VM - Publish - Consume * Database - All operations

質問: 13

グローバルな組織が多く、多くの国でデータセンターを運営しています。これらのデータセンター間にはプライベート ネットワーク リンクがあります。これは、すべてのビジネス データ (ただし、メタデータは除く) をこれらのプライベート ネットワーク接続を介して交換する必要があるためです。

この組織は現在、AWS をまったく使用していません。

IT 運用の労力と今後の投資を厳密に最小化するという戦略的決定が行われました。

Anypoint Platform コントロール プレーンとランタイム プレーンのデプロイ オプションのどの組み合わせが、この戦略的ジャーニーの開始時にこの組織に最も適していますか？

- A. MuleSoft がホストする Anypoint Platform コントロール プレーン CloudHub 共有ワーカー クラウドが複数の AWS リージョンにある
- B. Anypoint Platform - Private Cloud Edition 各データセンターでお客様がホストするランタイム プレーン
- C. MuleSoft がホストする Anypoint Platform コントロール プレーン 複数の AWS リージョンで顧客がホストするランタイム プレーン
- D. MuleSoft がホストする Anypoint Platform コントロール プレーン 顧客がホストする各データセンターのランタイム プレーン

正解: D ([コメントを发表する](#))

Correct answer is MuleSoft-hosted Anypoint Platform control plane Customer-hosted runtime plane in each datacenter There are two things to note about the question which can help us figure out correct answer.. * Business data must be exchanged over these private network connections which means we can not use MuleSoft provided Cloudhub option. So we are left with either customer hosted runtime in

external cloud provider or customer hosted runtime in their own premises. As customer does not use AWS at the moment. Hence that don't have the immediate option of using Customer-hosted runtime plane in multiple AWS regions. hence the most suitable option for runtime plane is Customer-hosted runtime plane in each datacenter * Metadata has no limitation to reside in organization premises. Hence for control plane MuleSoft hosted Anypoint platform can be used as a strategic solution.

Hybrid is the best choice to start. Mule hosted Control plane and Customer hosted Runtime to start with. Once they mature in cloud migration, everything can be in Mule hosted.

質問: 14

ある会社は、Mule API をヨーロッパ地域に拡張することを計画しています。現在、すべての新しいアプリケーションは、この命名規則に従って米国リージョンの Cloudhub にデプロイされています
{API 名}-{環境}。たとえば、Orders-SAPI-dev、Orders-SAPI-prod などです。

API 間の通信をブロックするネットワーク制限がないことを考慮して、CloudHub の EU リージョンで実行されている同じ新しい API を適用し、API とヨーロッパのターゲット ユーザーおよびシステムとの間のレイテンシを最小限に抑えるために、どのような戦略を実装する必要がありますか？

- A. すべてのミュール アプリケーションのランタイム マネージャーで地域プロパティをヨーロッパ (eu-de) に設定します。命名規則を {API 名}-{環境}-{地域} に変更し、この変更を消費するアプリケーションとユーザーに通知します。
- B. すべてのミュール アプリケーションの API マネージャーで地域プロパティをヨーロッパ (eu-de) に設定します。命名規則を {API 名}-{環境}-{地域} に変更し、この変更を使用するアプリケーションとユーザーに通知します。
- C. すべてのミュール アプリケーションのランタイム マネージャーで地域プロパティをヨーロッパ (eu-de) に設定します。命名規則を変更する必要はありません。
- D. すべてのミュール アプリケーションの API マネージャーで、リージョン プロパティをヨーロッパ (eu-de) に設定します。命名規則を変更する必要はありません。

正解: A ([コメントを发表する](#))

質問: 15

エンタープライズ アーキテクトとして、Mulesoft Anypoint プラットフォームを使用した新しい統合プロジェクトで正規データ モデルを使用する 2 つの理由は何ですか (2 つの回答を選択してください)。

- A. 一貫したデータ構造をプロセスに揃えるため
- B. 各データ型の正規定義が複数あります
- C. 業界標準のデータ形式を組み込むため
- D. モデルがバックとシステムを分離し、ラバ アプリケーションを変更からサポートするため
- E. 境界のあるコンテキスト内で領域を分離するには

正解: A,E ([コメントを发表する](#))

質問: 16

ある会社は、注文をオフライン処理のためにバックエンド システムに送信することで注文を処理する統合 Mule アプリケーションを設計しています。各注文は HTTP5 POST を介して Mule アプリケーションによって受信され、すぐに確認する必要があります。

承認されると、注文はバックエンド システムに送信されます。バックエンド システムからの拒否により正常に送信できない注文は、手動で (銀行システムの外で) 処理する必要があります。

Mule アプリケーションは、お客様がホストするランタイムにデプロイされ、必要に応じて既存の ActiveMQ ブローカーを使用できます。ActiveMQ ブローカーは、組織のファイアウォール内にあります。バックエンド システムには、小規模なネットワーク接続の問題と長時間の停止の両方が原因で、信頼性が低いという実績があります。

手動注文処理をサポートしながら最小限に抑えながら、バックエンド システムへの注文の自動送信を保証するには、Mule アプリケーション コンポーネントと ActiveMQ キューのどの組み合わせが必要ですか？

- A. システムでコールバックするためのバッチ ジョブ スコープ 長時間の再試行のためのオブジェクト ストア コンポーネントを含む、成功するまでのスコープ。Mule アプリケーションで設定された配信不能オブジェクト ストア
- B. バックエンド システムの呼び出しを支援する 1 つ以上の On Error スコープ 長時間の再試行用の VM コンポーネントを含む Until 成功スコープ クラウド ハブで構成された永続的な配信不能 VM キュー
- C. バックエンド システムの呼び出しを支援する 1 つ以上の on-Error スコープ 1 つ以上の ActiveMQ ロング リトライ キュー CloudHub オブジェクト ストア サービス内の永続的な配信不能オブジェクト ストア 構成
- D. バックエンド システムを呼び出す Until Successful スコープ 1 つ以上の ActiveMQ ロング リトライ キュー 1 つ以上の ActiveMQ デッドレター キュー (手動処理用)

正解: ([正解を表示します](#))

有効的な**MCIA-Level-1-JPN**問題集はJPNTTest.com提供され、**MCIA-Level-1-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**MCIA-Level-1-JPN**試験問題集を提供します。JPNTTest.com MCIA-Level-1-JPN試験問題集はもう更新されました。ここで**MCIA-Level-1-JPN**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/MCIA-Level-1-JPN-mondaishu> **275問、30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 17

VM を使用する Mule アプリケーションが顧客がホストするクラスターまたは複数のクラウドハブ ワーカーにデプロイされる場合、メッセージは Mule エンジンによってどのように消費されますか？

- A. 重複処理を避けるため、プライマリののみ
- B. 決定論的な方法で
- C. 非決定的な方法で
- D. 新しいメッセージごとに XA トランザクションを開始することによって

正解: ([正解を表示します](#))

質問: 18

Mule アプリケーションでは、フローには 2 つの JMS 消費操作が含まれます。これらは、JMS ブローカーに接続し、2 つの JMS 宛先からのメッセージを消費するために使用されます。その後、Mule アプリケーションは 2 つの JMS メッセージを結合します。

JMS ブローカーは高可用性 (HA) を実装しておらず、定期的なメンテナンスのために最大 10 分間のスケジュールされた停止が定期的 발생합니다。

予想される停止から最適に回復できるようにミュール フローを構築する最も慣用的な (意図された目的で使用される) 方法は何ですか？

- A. JMS コネクタの再接続戦略を構成する
- B. 成功するまでのスコープで 2 つの JMS 操作を囲みます。
- C. JMS コネクタのトランザクションを考える
- D. 2 つの JMS 操作を、エラー コンティニュー エラー ハンドラーを使用して Try スコープで囲みます。

正解: ([正解を表示します](#))

When an operation in a Mule application fails to connect to an external server, the default behavior is for the operation to fail immediately and return a connectivity error. You can modify this default behavior by configuring a reconnection strategy for the operation. You can configure a reconnection strategy for an operation either by modifying the operation properties or by modifying the configuration of the global element for the operation. The following are the available reconnection strategies and their behaviors: None Is the default behavior, which immediately returns a connectivity error if the attempt to connect is unsuccessful Standard (reconnect) Sets the number of reconnection attempts and the interval at which to execute them before returning a connectivity error Forever (reconnect-forever) Attempts to reconnect continually at a given interval

質問: 19

ある会社は、展開環境をオンプレミス クラスタからランタイム ファブリック (RTF) クラスタに移行することを計画しています。また、Mule ランタイム インスタンスにデプロイされた Mule アプリケーションが、アプリケーションのレプリカと再起動間でデータを保存および共有できるようにする必要もあります。

これらの要件はどのように満たすことができますか？

- A. RTF クラスタ内のレプリカ間でデータを共有するための Anypoint オブジェクト ストア V2
- B. RTF で永続ゲートウェイを構成する
- C. Mule Object Store を使用して任意のサーバで Persistence Gateway を設定します。
- D. オブジェクト ストア ポッドをクラスタ ノードの 1 つにインストールします。

正解: ([正解を表示します](#))

質問: 20

新しいビジネス プロセスを可能にするために、一連の統合 Mule アプリケーション (一部は API を公開) が作成されています。これにより、さまざまな利害関係者が影響を受ける可能性があります。これらの利害関係者は、半技術的なユーザー (JSON や XML などの基本的な統合用語と概念を理解している) と、Mule アプリケーションと API の技術的に熟練した潜在的な消費者の組み合わせです。

Mule アプリケーションと API の構築を担当するプロジェクト チームが、Anypoint Platform とその提供ツールセットを使用してこれらの利害関係者と通信するための効果的な方法は何ですか？

- A. Anypoint Design Center を使用して Mule アプリケーションと API を実装し、さまざまな利害関係者がこれらの Design Center プロジェクトにアクセスできるようにして、協力してフィードバックを提供できるようにします。
- B. API ノートブック (該当する場合) を含む統合設計を詳述するページで Anypoint Exchange エントリを作成し、利害関係者がさまざまなレベルの技術的深さで Mule アプリケーションと API を理解し、操作できるようにします。
- C. Anypoint Exchange を使用して、さまざまな Mule アプリケーションと API を登録し、RAML 定義を関係者と共有して、検出できるようにします。
- D. Mule 統合フロー内で Mule アプリケーションと API に関するドキュメントをインラインでキャプチャし、Anypoint Studio のドキュメントのエクスポート機能を使用して、このドキュメントの HTML バージョンを関係者に提供します。

正解: **B** ([コメントを发表する](#))

As the stakeholders are semitechnical users , preferred option is Create Anypoint Exchange entries with pages elaborating the integration design, including API notebooks (where applicable) to help the stakeholders understand and interact with the Mule applications and APIs at various levels of technical depth

質問: 21

組織は、Mulesoft がホストするコントロール プレーンを使用して管理される一連の顧客がホストする Mule ランタイムを使用します。カスタム コンポーネントやカスタム コーディングなしで、Anypoint Runtime Manager からアラートを生成できる条件は何ですか？

- A. 特定の顧客がホストするサーバー上の Mule ランタイムで、特定の期間に大量のメモリ消費が発生している場合
- B. デプロイされた Mule アプリケーションのいずれかで使用される SSL 証明書の有効期限が近づいている場合
- C. Mule ランタイムにインストールされている Mule ランタイム ライセンスの有効期限が近づいている場合
- D. Mule ランタイムの顧客がホストするサーバーのディスク容量が不足しそうになったとき

正解: ([正解を表示します](#))

Correct answer is When a Mule runtime on a given customer-hosted server is experiencing high memory consumption during certain periods Using Anypoint Monitoring, you can configure two different types of alerts: Basic alerts for servers and Mule apps Limit per organization: Up to 50 basic alerts for users who do not have a Titanium subscription to Anypoint Platform You can set up basic alerts to trigger email notifications when a metric you are measuring passes a specified threshold. You can create basic alerts for the following metrics for servers or Mule apps: For on-premises servers and CloudHub apps: * CPU utilization * Memory utilization * Thread count Advanced alerts for graphs in custom dashboards in Anypoint Monitoring. You must have a Titanium subscription to use this feature. Limit per organization: Up to 20 advanced alerts

質問: 22

ある企業がアプリケーション ネットワークを構築しており、4 つの Mule API を展開しています。1 つはエクスペリエンス API、1 つはプロセス API、2 つはシステム API です。すべての API からのログは、外部のログ集計ツールで集計されます。この会社は、複数の API 実装間で交換されるメッセージを追跡したいと考えています。複数の API 実装にわたって Mule イベント トレースを実装するために使用する必要がある、最も慣用的な (使用目的に基づく) 識別子は何ですか？

- A. Mule イベント ID
- B. Mule 相関 ID
- C. クライアントの IP アドレス
- D. DataWeave UUID

正解: ([正解を表示します](#))

Correct answer is Mule correlation ID By design, Correlation Ids cannot be changed within a flow in Mule 4 applications and can be set only at source. This ID is part of the Event Context and is generated as soon as the message is received by the application. When a HTTP Request is received, the request is inspected for "X-Correlation-Id" header. If "X-Correlation-Id" header is present, HTTP connector uses this as the Correlation Id. If "X-Correlation-Id" header is NOT present, a Correlation Id is randomly generated. For Incoming HTTP Requests: In order to set a custom Correlation Id, the client invoking the HTTP request must set "X-Correlation-Id" header. This will ensure that the Mule Flow uses this Correlation Id. For Outgoing HTTP Requests: You can also propagate the existing Correlation Id to downstream APIs. By default, all outgoing HTTP Requests send "X-Correlation-Id" header. However, you can choose to set a different value to "X-Correlation-Id" header or set "Send Correlation Id" to NEVER.

質問: 23

大手 e コマース企業は、Mulesoft API のオン ランタイム ファブリック (RTF) を使用して、顧客の注文を処理します。クレジットカード情報などの一部の顧客の機密情報も、API ペイロードの一部として存在します。

機密データを元のデータと照合するリスクを最小限に抑え、いつでもどこでも必要なときに元の値に戻すことができるアプローチはどれですか？

- A. マスキングを適用して機密情報を非表示にしてから API を使用する
- B. トークン化形式を作成し、トークン化ポリシーを API Gateway に適用します。
- C. マスキングとトークン化の両方を使用
- D. API Gateway でフィールド レベルの暗号化ポリシーを適用します。
- E. マスキング形式をトークン化解除して元の値を返すマネージャー

正解: ([正解を表示します](#))

質問: 24

展示を参照してください。

ビジネス プロセスには、SFTP を介した外部ベンダーからのファイルの受信が含まれます。ファイルを解析し、そのコンテンツを処理して検証し、最終的にデータベースに永続化する必要があります。ファイル転

送や HTTP POST などの他のメカニズムを使用して同様のファイルを送信するベンダーが増えるにつれて、配信メカニズムは将来変更されることが予想されます。

将来の変更の影響を最小限に抑えるために、これらの要件に合わせて設計する最も効果的な方法は何ですか？

- A. MuleSoft Scatter-Gather と MuleSoft Batch Job を使用して、さまざまなソースからのさまざまなファイルを処理します。
- B. ファイルを受信して MuleSoft バッチ ジョブを使用して処理するプロセス API を作成し、データ保存プロセスをシステム API に委任します。
- C. ファイルを受信し、ファイルに含まれるデータを使用してプロセス API を呼び出す API を作成し、必要に応じて MuleSoft バッチ ジョブおよびその他のシステム API を使用してプロセス API にデータを処理させます。
- D. 複合データ ソースを使用して、さまざまなソースからファイルを取得し、処理のために MuleSoft バッチ ジョブに配信できるようにします。

正解: C ([コメントを发表する](#))

* Scatter-Gather is used for parallel processing, to improve performance. In this scenario, input files are coming from different vendors so mostly at different times. Goal here is to minimize the impact of future change. So scatter Gather is not the correct choice.

* If we use 1 API to receive all files from different Vendors, any new vendor addition will need changes to that 1 API to accommodate new requirements. So Option A and C are also ruled out.

* Correct answer is Create an API that receives the file and invokes a Process API with the data contained in the file, then have the Process API process the data using a MuleSoft Batch Job and other System APIs as needed. Answer to this question lies in the API led connectivity approach.

* API-led connectivity is a methodical way to connect data to applications through a series of reusable and purposeful modern APIs that are each developed to play a specific role - unlock data from systems, compose data into processes, or deliver an experience. System API : System API tier, which provides consistent, managed, and secure access to backend systems. Process APIs : Process APIs take core assets and combines them with some business logic to create a higher level of value. Experience APIs : These are designed specifically for consumption by a specific end-user app or device.

So in case of any future plans , organization can only add experience API on addition of new Vendors, which reuse the already existing process API. It will keep impact minimal.

質問: 25

Mule アプリケーションからの同期ロギングと非同期ロギングの主な違いは何ですか？

- A. 同期ロギングは、単一のロギング スレッドにログ メッセージを書き込みますが、次のイベント プロセッサによって処理される Mule イベントをブロックしません。
- B. 非同期ログは、各 Mule イベントの処理時間を短縮しながら、Mule イベント処理のスループットを向上させることができます。
- C. 非同期ログは、より正確なタイムスタンプでより信頼性の高い監査証跡を生成します
- D. 進行中のトランザクション内の同期ロギングは、現在の Mule イベントを処理するスレッドと同じスレッドにログ メッセージを書き込みます。

正解: ([正解を表示します](#))

Types of logging:

A) Synchronous: The execution of thread that is processing messages is interrupted to wait for the log message to be fully handled before it can continue.

- * The execution of the thread that is processing your message is interrupted to wait for the log message to be fully output before it can continue

- * Performance degrades because of synchronous logging

- * Used when the log is used as an audit trail or when logging ERROR/CRITICAL messages

- * If the logger fails to write to disk, the exception would raise on the same thread that's currently processing the Mule event. If logging is critical for you, then you can rollback the transaction.

B) Asynchronous:

- * The logging operation occurs in a separate thread, so the actual processing of your message won't be delayed to wait for the logging to complete

- * Substantial improvement in throughput and latency of message processing

- * Mule runtime engine (Mule) 4 uses Log4j 2 asynchronous logging by default

- * The disadvantage of asynchronous logging is error handling.

- * If the logger fails to write to disk, the thread doing the processing won't be aware of any issues writing to the disk, so you won't be able to rollback anything. Because the actual writing of the log gets deferred, there's a chance that log messages might never make it to disk and get lost, if Mule were to crash before the buffers are flushed.

----- So Correct answer is: Asynchronous logging can improve Mule event processing throughput while also reducing the processing time for each Mule event

質問: 26

プロセス API の実装は変更する必要があります。API クライアントに対するこの変更の影響を最小限に抑える有効なアプローチは何ですか？

A. 可能な限り、プロセス API の RAML 定義が変更されないように、プロセス API 実装に必要な変更を実装します。

B. 現在のプロセス API の RAML 定義を更新し、更新された RAML 定義へのリンクを送信して API クライアント開発者に通知します。

C. API コンシューマーが新しいプロセス API または API バージョンに移行する準備ができていることを確認するまで、変更を延期します。

D. プロセス API の変更を新しい API 実装に実装し、古い API 実装に HTTP ステータス コード 301 - Moved Permanently を返させて、新しい API 実装を呼び出す必要があることを API クライアントに通知します。

正解: ([正解を表示します](#))

- * Option B shouldn't be used unless extremely needed, if RAML is changed, client needs to accommodate changes. Question is about minimizing impact on Client. So this is not a valid choice.

- * Option C isn't valid as Business can't stop for consumers acknowledgment.

- * Option D again needs Client to accommodate changes and isn't viable option.
- * Best choice is A where RAML definition isn't changed and underlined functionality is changed without any dependency on client and without impacting client.

質問: 27

Mule アプリケーションが JMS コネクタを使用して JMS プロバイダ (メッセージブローカー) と対話する場合のネットワーク接続について正しいのはどれですか?

- A.** JMS メッセージの送信を完了するには、JMS コネクタは JMS メッセージ受信者とのネットワーク接続を確立する必要があります。
- B.** メッセージを Mule アプリケーションに受信するために、JMS プロバイダは JMS コネクタへのネットワーク接続を開始し、この接続に沿ってメッセージをプッシュします。
- C.** JMS コネクタは、JMS プロバイダーによって決定されたプロトコルを介した JMS メッセージの送受信の両方をサポートします。
- D.** JMS コネクタは AMQP プロトコルを使用して、さまざまなタイプの JMS プロバイダへの接続を移植可能に確立できます。

正解: ([正解を表示します](#))

* To send message or receive JMS (Java Message Service) message no separate network connection need to be established. So option A, C and D are ruled out.

Correct: The JMS connector supports both sending and receiving of JMS

* JMS Connector enables sending and receiving messages to queues and topics for any message service that implements the JMS specification.

* JMS is a widely used API for message-oriented middleware.

* It enables the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous.

MuleSoft Doc Reference: <https://docs.mulesoft.com/jms-connector/1.7/>

質問: 28

Mule アプリケーションは、APIkit for SOAP を使用して SOAP Web サービスを実装します。Mule アプリケーションは、テスト環境で CloudHub ワーカーにデプロイされています。

統合テスト チームは、SOAP クライアントを使用して統合テストを実行したいと考えています。統合テストを実行するには、統合チームは SOAP Web サービスのインターフェース定義を取得する必要があります。

統合テスト チームが、SOAP クライアントとの統合テストを実行するために、デプロイされた SOAP Web サービスのインターフェース定義を取得するための最も慣用的な (本来の目的に使用される) 方法はどれですか?

- A.** API Manager から OpenAPI 仕様ファイルを取得します
- B.** Runtime Manager から XML ファイルを取得します。
- C.** デプロイされた Mule アプリケーションから WSDL ファイルを取得します
- D.** デプロイされた Mule アプリケーションから RAML ファイルを取得します。

正解: ([正解を表示します](#))

質問: 29

組織は、m-unit を使用してアプリケーションのテストスイートを構築しています。統合アーキテクトは、スタジオでテストレコーダーを使用して処理フローを記録し、キャプチャイベントに基づいて単体テストを構成することを推奨しています。

A. フローのテストは、内部で発生した Mule エラーで作成できません

フローまたは着信イベントに既に存在する

B. 並列プロセスから生じるモック値は可能であり、テストで続くプロセスの実行には影響しません

C. 記録されたフローの実行は正常に終了しますが、結果は失敗します

アプリケーションが強制終了されたため、目的地に到達できません

D. レコーダーは、ForEach プロセッサの前または内部でメッセージをスモークすることをサポートします

E. レコーダーは、テストされたデータの構造が反復内で変更されるループをサポートします。

正解: ([正解を表示します](#))

質問: 30

Order マイクロサービスと Fulfillment マイクロサービスは、メッセージベースの統合 (API 呼び出しではありません) を介してディエントと通信するように設計されています。

Order マイクロサービスは、履行される注文の詳細を含む Order メッセージ (一種のコマンドメッセージ) を発行します。その意図は、Order メッセージが 1 つの Mule アプリケーション (Fulfillment マイクロサービス) によってのみ消費されることです。

Fulfillment マイクロサービスは、Order メッセージを消費し、そこに記述されている注文を履行してから、OrderFulfilled メッセージ (一種のイベントメッセージ) を発行します。各 OrderFulfilled メッセージは、関心のある Mule アプリケーションで使用できます。Order マイクロサービスは、そのような Mule アプリケーションの 1 つです。

このシナリオで最も適切なメッセージブローカーとメッセージ送信先の選択はどれですか？

A. 注文メッセージが Anypoint MQ 交換に送信される OrderFulfilled メッセージが Anypoint MQ キューに送信される 両方のマイクロサービスがメッセージブローカーとして Anypoint MQ と対話するため、両方のマイクロサービスの負荷をサポートするように拡張する必要があります

B. 注文メッセージは JMS キューに送信されます。OrderFulfilled メッセージが JMS トピックに送信される 両方のマイクロサービスが同じ JMS プロバイダー (メッセージブローカー) インスタンスとやり取りするため、両方のマイクロサービスの負荷をサポートするようにスケーリングする必要があります

C. 注文メッセージはフルフィルメント マイクロサービスに直接送信されます。OrderFulfilled メッセージは Order マイクロサービスに直接送信されます。Order マイクロサービスは 1 つの AMQP 互換メッセージブローカーとやり取りし、Fulfillment マイクロサービスは別の AMQP 互換メッセージブローカーとやり取りします。各マイクロサービス

D. 注文メッセージは JMS キューに送信されます。OrderFulfilled メッセージが JMS トピックに送信される Order マイクロサービスは 1 つの JMS プロバイダー (メッセージブローカー) と対話し、Fulfillment マイクロサービスは別の JMS プロバイダーと対話するため、両方のメッセージブローカーを選択してスケーリングし、各マイクロサービスの負荷を最適にサポートできます。

正解: ([正解を表示します](#))

* If you need to scale a JMS provider/ message broker, - add nodes to scale it horizontally or - add memory to scale it vertically

* Cons of adding another JMS provider/ message broker: - adds cost. - adds complexity to use two JMS brokers - adds Operational overhead if we use two brokers, say, ActiveMQ and IBM MQ

* So Two options that mention to use two brokers are not best choice. * It's mentioned that "The Fulfillment microservice consumes Order messages, fulfills the order described therein, and then publishes an OrderFulfilled message. Each OrderFulfilled message can be consumed by any interested Mule application."

- When you publish a message on a topic, it goes to all the subscribers who are interested - so zero to many subscribers will receive a copy of the message.

- When you send a message on a queue, it will be received by exactly one consumer. * As we need multiple consumers to consume the message below option is not valid choice: "Order messages are sent to an Anypoint MQ exchange. OrderFulfilled messages are sent to an Anypoint MQ queue.

Both microservices interact with Anypoint MQ as the message broker, which must therefore scale to support the load of both microservices" * Order messages are only consumed by one Mule application, the Fulfillment microservice, so we will publish it on queue and OrderFulfilled message can be consumed by any interested Mule application so it need to be published on Topic using same broker.

* Correct:

質問: 31

組織は、顧客および注文管理システムを統合するために、Mule API 実装と非 Mule API 実装の両方をデプロイしました。パブリック インターネット上の REST クライアントは、すべての API を利用できます。組織は、ヘルス チェックを実行してこれらの API を監視したいと考えています。たとえば、API がリクエストを適切に受け入れて処理できるかどうかを判断します。この組織は、外部監視ツールへのサブスクリプションを持っておらず、IT フットプリントを拡張することも望んでいません。

Mule API 実装と非 Mule API 実装の両方の可用性を監視する最も慣用的な (本来の目的で使用される) 方法を提供する Anypoint Platform 機能はどれですか？

A. API マネージャー

B. API 機能の監視

C. Anypoint Visualizer

D. ランタイム マネージャー

正解: ([正解を表示します](#))

有効的な**MCIA-Level-1-JPN**問題集はJPNTTest.com提供され、**MCIA-Level-1-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**MCIA-Level-1-JPN**試験問題集を提供します。JPNTTest.com **MCIA-Level-1-JPN**試験問題集はもう更新されました。ここで**MCIA-Level-1-JPN**問題集のテストエンジ

ンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/MCIA-Level-1-JPN-mondaishu>
275問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 32

製品のオーケストレーションを実行するために、ミュール アプリケーションが設計されています。Mule アプリケーションは、在庫 API と製品販売履歴 API からの応答を最小の待機時間で結合する必要があります。

全体的なレイテンシを最小限に抑えるため。Mule アプリケーションで各 API 要求を呼び出す最も慣用的な (本来の目的に使用される) 設計はどれですか?

- A. Dataweave reduce operator から個別のルックアップ コールで各 API リクエストを呼び出します。
- B. Scatter-Gather の個別のルートで各 API リクエストを呼び出します
- C. Parallel For Each スコープの個別のルートで各 API 要求を呼び出します
- D. 個別の Async スコープで各 API リクエストを呼び出します

正解: ([正解を表示します](#))

Scatter-Gather sends a request message to multiple targets concurrently. It collects the responses from all routes, and aggregates them into a single message.

質問: 33

ある航空会社は、フライト データをオンライン集計 Web サイトに統合する API 接続プロジェクトを設計しています。インターフェイスは、安全な通信、高性能および非同期メッセージ交換を可能にする必要があります。

Mulesoft がこれらのテクノロジーを完全にサポートし、これらのインターフェイス用の Anypoint コネクタが存在すると仮定すると、この統合に適したインターフェイス テクノロジーはどれですか?

- A. CSV over FTP YAM L over TLS JSON over HTTPS
- B. SOAP over HTTPS HOP over TLS gRPC over HTTPS
- C. XML over ActiveMQ XML over SFTP XML/REST over HTTPS
- D. HTTPS 経由の AsyncAPI

HTTPS を介した RabbitMQ JSON/REST を使用した AMQP

正解: D ([コメントを发表する](#))

質問: 34

組織は、注文をオフライン処理のためにバックエンド システムに送信することで注文を処理する統合 Mule アプリケーションを設計しています。各注文は、HTTPS POST を介して Mule アプリケーションによって受信され、すぐに確認する必要があります。承認されると、注文はバックエンド システムに送信されます。バックエンド システムからの拒否により正常に送信できない注文は、(バックエンド システム外で) 手動で処理する必要があります。

Mule アプリケーションは、顧客がホストするランタイムにデプロイされ、必要に応じて既存の ActiveMQ ブローカーを使用できます。ActiveMQ ブローカーは、組織のファイアウォール内にあります。バックエンド システムには、小規模なネットワーク接続の問題と長時間の停止の両方が原因で、信頼性が低いという実績があります。

Mule アプリケーション コンポーネントと ActiveMQ キューの (本来の目的に使用される) どの慣用的な組み合わせが、バックエンド システムへの注文の自動送信を保証すると同時に、手動注文処理をサポートしながら最小限に抑えるために必要ですか？

A. バックエンド システムを呼び出すための Batch Job スコープ

長時間の再試行用の Object Store コンポーネントを含む Until Successful スコープ Mule アプリケーションで設定された配信不能オブジェクト ストア

B. バックエンド システムの呼び出しを支援する 1 つ以上の On Error スコープ

1 つ以上の ActiveMQ 長期再試行キュー

CloudHub オブジェクト ストア サービスで構成された永続的な配信不能オブジェクト ストア

C. バックエンド システムの呼び出しを支援する 1 つ以上の On Error スコープ

長時間の再試行のための VM コンポーネントを含む、成功するまでのスコープ

CloudHub で構成された永続的な配信不能 VM キュー

D. バックエンド システムを呼び出すまでのスコープ

1 つ以上の ActiveMQ 長期再試行キュー

手動処理用の 1 つ以上の ActiveMQ 配信不能キュー

正解: ([正解を表示します](#))

質問: 35

展示を参照してください。

組織は、組織のオンプレミス システムに接続する Mule アプリケーションの非実稼働デプロイメント用に Anypoint VPC のサイジングを行っています。これは約に適用されます。60 ミュール アプリケーション。各アプリケーションは、2 つの CloudHub i ワーカーにデプロイされます。組織には現在、この VPC を共有する 3 つの非実稼働環境 (DEV、SIT、および UAT) があります。VPC の AWS リージョンには 2 つの AZ があります。

この組織には、本番環境に自動的にデプロイする前に、すべての非本番環境を通じて各アプリケーションを自動的に進行させる、非常に成熟した DevOps アプローチがあります。このプロセスにより、CloudHub の通常のゼロ ダウンタイム デプロイ機能を使用して、1 時間に複数の Mule アプリケーションがデプロイされます。

使用可能なプライベート IP アドレス範囲が最小になる、この VPC の CIDR ブロックは何ですか？

A. 10.0.0.0/26 (64 IPS)

B. 10.0.0.0/25 (128 IP)

C. 10.0.0.0/24 (256 IP)

D. 10.0.0.0/22 (1024 IP)

正解: **D** ([コメントを发表する](#))

Mule applications are deployed in CloudHub workers and each worker is assigned with a dedicated IP * For zero downtime deployment, each worker in CloudHub needs additional IP addresses * A few IPs in a VPC are reserved for infrastructure (generally 2 IPs) * The IP addresses are usually in a private range with a subnet block specifier, such as 10.0.0.1/24 * The smallest CIDR network subnet block you can assign for your VPC is /24 (256 IP addresses) (60*3 env * 2 worker per application) + 50% of (total) for zero downtime = 540 In this case correct answer is 10.0.0.0/22 as this provided 1024 IP's .

Other IP's are insufficient.

質問: 36

展示を参照してください。

Mule アプリケーションは、マルチノード Mule ランタイム クラスタにデプロイされます。Mule アプリケーションは、クラスタ レプリカ間で競合するコンシューマ パターンを使用して、JMS キューから JMS メッセージを受信します。受信した各 JMS メッセージを処理するために、次の手順がフローで実行されます。

ステップ 1: JMS 関連 ID ヘッダーは、受信した JMS メッセージから読み取られます。

ステップ 2: Mule アプリケーションは、HTTPS を介してべき等 SOAP Web サービスを呼び出し、JMS 関連 ID を SOAP 要求の 1 つのパラメータとして渡します。

ステップ 3: SOAP Web サービスからの応答も、同じ JMS 関連 ID を返します。

ステップ 4: SOAP Web サービスから受け取った JMS 関連 ID は、ステップ 1 で受け取った JMS 関連 ID と同一であることが検証されます。

ステップ 5: Mule アプリケーションは応答 JMS メッセージを作成し、JMS 関連 ID メッセージ ヘッダーを検証済みの JMS 関連 ID に設定し、そのメッセージを応答 JMS キューにパブリッシュします。

Mule アプリケーションは、ステップ 1 とステップ 3 で受け取った JMS 関連 ID 値をどこに保存する必要がありますか？ これにより、ステップ 4 の検証を実行できると同時に、Mule アプリケーション全体の可用性、フォールトトレラント、パフォーマンス、保守性を高めることができますか？

- A. 両方の関連 ID 値を永続オブジェクトストアに格納する必要があります。
- B. 両方の関連 ID 値を非永続オブジェクトストアに格納する必要があります
- C. ステップ 1 の関連 ID 値は永続オブジェクトストアに保存する必要があります ステップ 3 の関連 ID 値は Mule イベント変数/属性として保存する必要があります
- D. 両方の関連 ID 値を Mule イベント変数/属性として保存する必要があります

正解: **C** ([コメントを发表する](#))

* If we store Correlation id value in step 1 as Mule event variables/attributes, the values will be cleared after server restart and we want system to be fault tolerant.

* The Correlation ID value in Step 1 should be stored in a persistent object store.

* We don't need to store Correlation ID value in Step 3 to persistent object store. We can store it but as we also need to make application performant. We can avoid this step of accessing persistent object store.

* Accessing persistent object stores slow down the performance as persistent object stores are by default stored in shared file systems.

* As the SOAP service is idempotent in nature. In case of any failures , using this Correlation ID saved in first step we can make call to SOAP service and validate the Correlation ID.

Top of Form

Additional Information:

* Competing Consumers are multiple consumers that are all created to receive messages from a single Point-to-Point Channel. When the channel delivers a message, any of the consumers could potentially receive it. The messaging system's implementation determines which consumer actually receives the

message, but in effect the consumers compete with each other to be the receiver. Once a consumer receives a message, it can delegate to the rest of its application to help process the message.

* In case you are unaware about term idempotent re is more info:

Idempotent operations means their result will always same no matter how many times these operations are invoked.

Bottom of Form

質問: 37

API は単体テスト済みで、統合テストの準備ができています。API は、すべての環境で Client ID Enforcement ポリシーによって管理されます。

ステージング環境で API の統合テストを開始する前に、テスト チームは何をする必要がありますか？

- A. API ポータルにアクセスし、ステージング環境で API ポータルによって提供されるクライアント ID とクライアント シークレットを使用して API ノートブックを作成する必要があります。
- B. ステージング環境で API インスタンスへのアクセスを要求し、API のテストに使用するクライアント ID とクライアント シークレットを取得する必要があります。
- C. ステージング環境で API の API バージョン所有者として割り当てられる必要があります。
- D. ステージング環境へのアクセスを要求し、その環境を API のテストに使用するためのクライアント ID とクライアント シークレットを取得する必要があります。

正解: ([正解を表示します](#))

* It's mentioned that the API is governed by a Client ID Enforcement policy in all environments.

* Client ID Enforcement policy allows only authorized applications to access the deployed API implementation.

* Each authorized application is configured with credentials: client_id and client_secret.

* At runtime, authorized applications provide the credentials with each request to the API implementation.

MuleSoft Reference: <https://docs.mulesoft.com/api-manager/2.x/policy-mule3-client-id-based-policies>

質問: 38

組織は、CloudHub にデプロイされる Mule アプリケーションを作成しています。Mule アプリケーションには、データベース ユーザのパスワードを格納する dbPassword という名前のプロパティがあります。組織のセキュリティ基準では、Runtime Manager の [プロパティ] タブで値を設定した後、dbPassword プロパティをすべての Anypoint Platform ユーザーから非表示にする必要があることが示されています。Runtime Manager で dbPassword プロパティ値を非表示にするのに役立つ Mule アプリケーションの設定はどれですか？

- A. pom.xml ファイルの secureProperties セクションに dbPassword プロパティを追加します。
- B. mule-artifact.json ファイルの secureProperties セクションに dbPassword プロパティを追加します。
- C. secure::dbPassword をプロパティ プレースホルダー名として使用し、クリアテキスト (暗号化されていない) 値をセキュア プロパティ プレースホルダー ファイルに格納します。
- D. secure::dbPassword をプロパティ プレースホルダー名として使用し、プロパティの暗号化された値をセキュア プロパティ プレースホルダー ファイルに格納します。

正解: D ([コメントを发表する](#))

質問: 39

Mule アプリケーションは、次のことを行うように設計されています。

ステップ 1: JMS キューから SalesOrder メッセージを読み取ります。各 SalesOrder は、ヘッダーと SalesOrderLineItem のリストで構成されます。

ステップ 2: SalesOrder ヘッダーと各 SalesOrderLineItem を RDBMS の異なるテーブルに挿入します。

ステップ 3: SalesOrder ヘッダーと、そのすべての SalesOrderLineItems の価格の合計を別の RDBMS のテーブルに挿入します。

SalesOrder メッセージが失われることはなく、両方の RDBMS のすべての SalesOrder 関連情報の一貫性が常に確保されている必要があります。

これらの要件に対応する設計上の選択 (トランザクションの選択を含む) とステップの順序は?

A. 1) JMS メッセージを読み取ります (XA トランザクションではありません)。

2) 1 つの DB トランザクションで両方の DB 挿入を実行する

3) JMS メッセージを確認する

B. 1) JMS メッセージを読み取ります (XA トランザクションではありません)。

2) SEPARATE DB トランザクションで EACH DB 挿入を実行する

3) JMS メッセージを確認する

C. 1) XA トランザクションで JMS メッセージを読み取る

2) 同じ XA トランザクションで、両方の DB 挿入を実行しますが、JMS メッセージを確認しません。

D. 1) JMS メッセージを読み取り、確認します (XA トランザクションではありません)。

2) NEW XA トランザクションで、両方の DB 挿入を実行する

正解: ([正解を表示します](#))

* Option A says "Perform EACH DB insert in a SEPARATE DB transaction". In this case if first DB insert is successful and second one fails then first insert won't be rolled back causing inconsistency. This option is ruled out.

* Option D says Perform BOTH DB inserts in ONE DB transaction.

Rule of thumb is when one or more DB connections are required we must use XA transaction as local transactions support only one resource. So this option is also ruled out.

* Option B acknowledges the before DB processing, so message is removed from the queue. In case of system failure at later point, message can't be retrieved.

* Option C is Valid: Though it says "do not ack JMS message", message will be auto acknowledged at the end of transaction. Here is how we can ensure all components are part of XA transaction:

<https://docs.mulesoft.com/jms-connector/1.7/jms-transactions> Additional Information about transactions:

* XA Transactions - You can use an XA transaction to group together a series of operations from multiple transactional resources, such as JMS, VM or JDBC resources, into a single, very reliable, global transaction.

* The XA (eXtended Architecture) standard is an X/Open group standard which specifies the interface between a global transaction manager and local transactional resource managers.

The XA protocol defines a 2-phase commit protocol which can be used to more reliably coordinate and sequence a series of "all or nothing" operations across multiple servers, even servers of different types

* Use JMS ack if

- Acknowledgment should occur eventually, perhaps asynchronously
- The performance of the message receipt is paramount
- The message processing is idempotent
- For the choreography portion of the SAGA pattern

* Use JMS transactions

- For all other times in the integration you want to perform an atomic unit of work
- When the unit of work comprises more than the receipt of a single message
- To simply and unify the programming model (begin/commit/rollback)

質問: 40

ある企業は、新しいテクノロジーの採用をサポートしながら、アプリケーションとデータへのアクセスを高速化するために法制度を最新化しています。このビジネス目標を達成するための鍵は、API を使用する Docker および kubernetes コンテナの下で模倣するマイクロサービスを含む、企業の主要なシステムとデータのロックを解除することです。

現在の積極的なバックログとプロジェクト配信の要件を考慮して、同社は、必要に応じてスケール、オンプレミス システムへの接続、および移行が可能なミュール ランタイムに API を迅速に展開することにより、変革プロジェクトの第 1 フェーズで戦略的アプローチを採用したいと考えています。

会社の目標をサポートするランタイム展開オプションはどれですか？

- A. VMware メタルのランタイム ファブリック
- B. セルフマネージド Kubernetes 上のランタイム ファブリック
- C. お客様がホストするセルフ プロビジョニング ランタイム
- D. Cloudhub ランタイム

正解: ([正解を表示します](#))

質問: 41

ある保険会社は、2 つのベンダーから在庫の詳細を取得するために MuleSoft API を実装しています。ネットワークの問題により、ベンダー アプリケーションへの呼び出しが断続的にタイムアウトになります。しかし、トランザクションは再処理時に成功します。この要件を実装する最もパフォーマンスの高い方法は何か？

- A. ラウンド ロビン スコープを実装して、2 つの異なるルートで 2 つのベンダー アプリケーションを呼び出します。Try-Catch スコープを使用して、各ルートでタイムアウト エラーの再試行メカニズムを実装します。
- B. For-Each スコープを実装して、2 つのベンダー アプリケーションを呼び出します。
タイムアウトエラーの再試行メカニズムを実装するために、スコープが成功するまで使用します
- C. Choice スコープを実装して、2 つの異なるルートで 2 つのベンダー アプリケーションを呼び出します。try-catch スコープを使用して、各ルートでタイムアウト エラーの再試行メカニズムを実装します。
- D. 2 つのベンダーを呼び出すためのスキッター ギャザー スコープを実装します。

2 つの異なるルートでのアプリケーション

再試行メカニズムを実装するには、Until-Successful スコープを使用します。

各ルートのタイムアウト エラーの場合

正解: ([正解を表示します](#))

質問: 42

新しいミュール API を実装する大手銀行。

API の目的は、バックエンド アプリケーションから顧客の口座残高を取得し、それらをオンライン プラットフォームのオンラインバンキング プラットフォームに表示することです。オンラインバンキング プラットフォームは、一連の口座を Mule API に送信し、口座残高を取得します。

処理の一環として、Mule API は監査目的でデータをデータベースに挿入する必要があります。このプロセスは、アカウント残高の取得フローにパフォーマンス関連の影響を与えるべきではありません。スループットを向上させるには、この要件をどのように実装する必要がありますか？

- A. Async スコープを実装して、バックエンド アプリケーションからデータを取得し、Audit データベースにレコードを挿入します。
- B. バックエンド アプリケーションからデータを取得するためにスコープごとに並列を実装し、Async スコープを使用してレコードを Audit データベースに挿入します。
- C. バックエンド アプリケーションからデータをフェッチするための try-catch スコープを実装し、Async スコープを使用してレコードを Audit データベースに挿入します。
- D. for each スコープを実装して、バックエンド アプリケーションからデータをフェッチし、Audit データベースにレコードを挿入します。

正解: ([正解を表示します](#))

質問: 43

企業は、さまざまなクラウドハブ ワーカーにさまざまなパブリック API とプライベート API を実装する複数のミュール アプリケーションをデプロイしました。これらの API は、利害関係者によって定義された高可用性と信頼性 SLA に沿ったものでなければならぬ重要なアプリケーションです。

運用チームが停止通知を受信できるように、API の可用性 (稼働状況または準備状況) を監視するにはどうすればよいですか？

- A. API マネージャーでアラートの失敗条件を構成する
- B. 任意のポイント機能監視テスト API の機能動作を使用
- C. Anypoint 監視から個々のアプリケーションの監視を有効にします
- D. ランタイム マネージャーで障害条件のアラートを構成する

正解: ([正解を表示します](#))

質問: 44

機械設備に予知保全を実装するために、ACME Tractors は何千もの IoT センサーをインストールしました。これらのセンサーは、各機械資産のデータを JMS メッセージのシーケンスとしてほぼリアルタイムで、JMS サーバー上の SENSOR_DATA という名前の JMS キューに送信します。Mule アプリケーションには、JMS サーバーの SENSOR_DATA JMS キューから着信メッセージを受信するように設定された JMS リスナー操作が含まれています。Mule アプリケーションは、受信した各 JMS メッセージを保持し、対応する Mule イベントの変換済みバージョンを機械装置のバックエンド システムに送信します。

Mule アプリケーションは、顧客がホストするマルチノードの Mule ランタイム クラスタにデプロイされます。通常の状態では、各 JMS メッセージは 1 回だけ処理する必要があります。

JMS キューのパフォーマンスと同時メッセージ処理を最大化するには、JMS リスナーをどのように構成する必要がありますか？

A. numberOfConsumers を 1 より大きい値に設定します。

primaryNodeOnly を false に設定

B. numberOfConsumers を 1 より大きい値に設定します。

primaryNodeOnly を true に設定

C. numberOfConsumers = 1 を設定します。

primaryNodeOnly を true に設定

D. numberOfConsumers = 1 を設定します。

primaryNodeOnly を false に設定

正解: ([正解を表示します](#))

質問: 45

統合 Mule アプリケーションは、顧客がホストするマルチノード Mule 4 ランタイム ダスターにデプロイされます。Mule アプリケーションは、JMS コネクタの Listener 操作を使用して、JMS キューから着信メッセージを受信します。

メッセージは Mule アプリケーションによってどのように消費されますか？

A. JMS プロバイダーの構成に応じて、すべてのメッセージがプライマリ クラスタ ノードのみによって消費されるか、またはすべてのメッセージがすべてのクラスタ ノードによって消費されます。

B. リスナー操作の構成に関係なく、すべてのメッセージがすべてのクラスタ ノードによって消費されます。

C. リスナー操作の構成に応じて、すべてのメッセージがプライマリ クラスタ ノードのみによって消費されるか、または各メッセージがいずれかのクラスタ ノードによって消費されます。

D. リスナー操作の構成に関係なく、すべてのメッセージはプライマリ クラスタ ノードのみによって消費されます。

正解: ([正解を表示します](#))

Correct answer is Depending on the Listener operation configuration, either all messages are consumed by ONLY the primary cluster node or else EACH message is consumed by ANY ONE cluster node For applications running in clusters, you have to keep in mind the concept of primary node and how the connector will behave. When running in a cluster, the JMS listener default behavior will be to receive messages only in the primary node, no matter what kind of destination you are consuming from. In case of consuming messages from a Queue, you'll want to change this configuration to receive messages in all the nodes of the cluster, not just the primary.

This can be done with the primaryNodeOnly parameter:

```
<jms:listener config-ref="config" destination="${inputQueue}" primaryNodeOnly="false"/>
```

質問: 46

API 実装に含まれるバックエンド システムの 1 つは、パーティクル クライアントが作成できるリクエスト数にレート制限を適用します。

バックエンド システムと API 実装の両方が、ステージング環境を含むいくつかの非運用環境と特定の運用環境にデプロイされます。バックエンド システムのレート制限は、すべての非実稼働環境に適用されます。

ただし、本番環境にはレート制限がありません。

非実稼働ステージング環境で API 実装のパフォーマンス テストを実施する費用対効果の高い方法はどれですか？

A. MUnit を使用して、バックエンド システムからの標準応答をシミュレートします。

次に、パフォーマンス テストを実施して、システム内の他のボトルネックを特定します。

B. ステージング環境のバックエンド システムの場所をバイパスし、典型的なバックエンド システムの応答を複製するモッキング サービスを呼び出す API 実装内のロジックを含めます。次に、この API 実装を使用してパフォーマンス テストを実施します。

C. レート制限のあるバックエンド システムに対して、ステージング環境で縮小されたパフォーマンス テストを実施します。次に、パフォーマンス結果をフル プロダクション スケールにアップスケールします。

D. バックエンド システムの

生産性能特性

次に、モックを使用するように API 実装を構成します。

サービスを提供し、性能テストを実施する

正解: ([正解を表示します](#))

有効的な**MCIA-Level-1-JPN**問題集はJPNTTest.com提供され、**MCIA-Level-1-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**MCIA-Level-1-JPN**試験問題集を提供します。JPNTTest.com MCIA-Level-1-JPN試験問題集はもう更新されました。ここで**MCIA-Level-1-JPN**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/MCIA-Level-1-JPN-mondaishu> **275問、30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 47

ある組織は、顧客がホストする 4 つのノードの Mule ランタイム クラスタを使用して、1 つのステートレス API 実装をホストします。API は、負荷分散にラウンドロビンを使用するロード バランサーを介して HTTPS 経由でアクセスされます。クラスター内の各ノードは、現在の要求数の 4 倍を受け入れることができるようにサイズ設定されています。

クラスタ内の 2 つのノードで停電が発生し、使用できなくなりました。ロード バランサーは停止を指示し、使用できない 2 つのノードがそれ以上 HTTP 要求を受信しないようにブロックします。

残りのクラスタ ノードが完全に動作していると仮定した場合、パフォーマンスに関連する平均的な結果として、どのような結果が保証されますか？

A. API の平均応答時間が 100% 増加

B. API のスループットが 50% 低下

C. 残りの各ノードが受け取るリクエスト数が 100% 増加

D. 残りの各ノードで消費される JVM ヒープ メモリが 50% 増加します。

正解: **C** ([コメントを发表する](#))

* "100% increase in the throughput of the API" might look correct, as the number of requests processed per second might increase, but is it guaranteed to increase by 100%? Using 4 nodes will definitely increase throughput of system. But it is cant be precisely said if there would be 100% increase in throughput as it depends on many other factors. Also it is nowhere mentioned in the description that all nodes have same CPU/memory assigned. The question is about the guaranteed behavior * Increasing number of nodes will have no impact on response time as we are scaling application horizontally and not vertically. Similarly there is no change in JVM heap memory usage. * So Correct answer is 50% reduction in the number of requests being received by each node This is because of the two reasons. 1) API is mentioned as stateless 2) Load Balancer is used

質問: 48

新しいアップストリーム API は、中央値 500 ミリ秒、最大 800 ミリ秒 (99 パーセンタイル) の応答時間の SLA を提供するように設計されています。対応する API 実装は、非常に類似した複雑さの 3 つのダウンストリーム API を順番に呼び出す必要があります。これらのダウンストリーム API の最初のものは、応答時間について次の SLA を提供します: 中央値: 100 ミリ秒、80 パーセンタイル: 500 ミリ秒、95 パーセンタイル: 1000 ミリ秒。可能であれば、新しいアップストリーム API の望ましい SLA を満たすために、最初のダウンストリーム API の呼び出しに対してアップストリーム API でタイムアウトを設定するにはどうすればよいですか?

A. タイムアウトを 100 ミリ秒に設定します。他の 2 つのダウンストリーム API が完了するまでに 400 ミリ秒かかります

B. タイムアウトを設定しません。この API の呼び出しは必須であるため、応答するまで待つ必要があります

C. タイムアウトを 50 ミリ秒に設定します。これにより、その API のより多くの呼び出しがタイムアウトになりますが、再試行のための追加の余地が与えられます

D. アップストリーム API の望ましい SLA を満たすためのタイムアウトはありません。別の SLA を最初のダウンストリーム API とネゴシエートするか、別の API を呼び出す必要があります

正解: ([正解を表示します](#))

Before we answer this question , we need to understand what median (50th percentile) and 80th percentile means. If the 50th percentile (median) of a response time is 500ms that means that 50% of my transactions are either as fast or faster than 500ms.

If the 90th percentile of the same transaction is at 1000ms it means that 90% are as fast or faster and only 10% are slower. Now as per upstream SLA , 99th percentile is 800 ms which means 99% of the incoming requests should have response time less than or equal to 800 ms. But as per one of the backend API , their 95th percentile is 1000 ms which means that backend API will take 1000 ms or less than that for 95% of. requests. As there are three API invocation from upstream API , we can not conclude a timeout that can be set to meet the desired SLA as backend SLA's do not support it.

Let see why other answers are not correct.

- 1) Do not set a timeout --> This can potentially violate SLA's of upstream API
- 2) Set a timeout of 100 ms; ---> This will not work as backend API has 100 ms as median meaning only 50% requests will be answered in this time and we will get timeout for 50% of the requests. Important thing to note here is, All APIs need to be executed sequentially, so if you get timeout in first API, there is no use of going to second and third API. As a service provider you wouldn't want to keep 50% of your consumers dissatisfied. So not the best option to go with.
- *To quote an example: Let's assume you have built an API to update customer contact details.
- First API is fetching customer number based on login credentials
 - Second API is fetching Info in 1 table and returning unique key
 - Third API, using unique key provided in second API as primary key, updating remaining details
- * Now consider, if API times out in first API and can't fetch customer number, in this case, it's useless to call API 2 and 3 and that is why question mentions specifically that all APIs need to be executed sequentially.
- 3) Set a timeout of 50 ms --> Again not possible due to the same reason as above Hence correct answer is 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

質問: 49

統合 Mule アプリケーションは、注文をオフライン処理のためにバックエンド システムに送信することで注文を処理するように設計されています。各注文は、HTTPS POST を介して Mule アプリケーションによって受信され、すぐに確認する必要があります。承認されると、注文はバックエンド システムに送信されます。バックエンド システムからの拒否により正常に送信できない注文は、手動で (バックエンド システム外で) 処理する必要があります。

Mule アプリケーションは、顧客がホストするランタイムにデプロイされ、必要に応じて既存の ActiveMQ ブローカーを使用できます。

バックエンド システムには、小規模なネットワーク接続の問題と長時間の停止の両方が原因で、信頼性が低いという実績があります。

手動の注文処理を最小限に抑えながら、バックエンド システムへの注文の自動送信を確実にするために必要な Mule アプリケーション コンポーネントと ActiveMQ キューの慣用的な (意図した目的に使用される) 組み合わせはどれですか？

- A. 手動処理用の On Error スコープ 非永続 VM ActiveMQ Dead Letter Queue
- B. 手動処理用の On Error スコープ MuleSoft オブジェクトストア ActiveMQ デッド レター キュー
- C. コンポーネントが成功するまで MuleSoft Object Store ActiveMQ は不要または使用されません
- D. コンポーネントが成功するまで ActiveMQ ロング リトライ キュー ActiveMQ デッド レター 手動処理用キュー

正解: D ([コメントを发表する](#))

Correct answer is using below set of activities Until Successful component ActiveMQ long retry Queue ActiveMQ Dead Letter Queue for manual processing We will see why this is correct answer but before that lets understand few of the concepts which we need to know. Until Successful Scope The Until Successful scope processes messages through its processors until the entire operation succeeds. Until Successful

repeatedly retries to process a message that is attempting to complete an activity such as: - Dispatching to outbound endpoints, for example, when calling a remote web service that may have availability issues. - Executing a component method, for example, when executing on a Spring bean that may depend on unreliable resources. - A sub-flow execution, to keep re-executing several actions until they all succeed, - Any other message processor execution, to allow more complex scenarios. How this will help requirement : Using Until Successful Scope we can retry sending the order to backend systems in case of error to avoid manual processing later. Retry values can be configured in Until Successful Scope Apache ActiveMQ It is an open source message broker written in Java together with a full Java Message Service client ActiveMQ has the ability to deliver messages with delays thanks to its scheduler. This functionality is the base for the broker redelivery plug-in. The redelivery plug-in can intercept dead letter processing and reschedule the failing messages for redelivery. Rather than being delivered to a DLQ, a failing message is scheduled to go to the tail of the original queue and redelivered to a message consumer. How this will help requirement : If backend application is down for a longer duration where Until Successful Scope wont work, then we can make use of ActiveMQ long retry Queue. The redelivery plug-in can intercept dead letter processing and reschedule the failing messages for redelivery. Mule Reference: <https://docs.mulesoft.com/mule-runtime/4.3/migration-core-until-successful>

質問: 50

2 つの要件を満たすように設計されたミュール アプリケーション

a) ファイルの処理は、VM イベントのロード バランシングのために VM 中間キューを使用して、FTPS サーバーからバックエンド データベースへ同期的に行われます。FTPS ファイルの場合、アプリケーションが Cloudhub ワーカーにデプロイされている場合、ファイル进行处理するため、およびバッチ ジョブ スコープのために VM キューをどのように構成する必要がありますか？

A. FTPS ファイル処理にクラウド ハブの永続キューを使用する

バッチ ジョブ スコープの VM キューを構成する必要はありません。これは、デフォルトで VM キューイングにワーカーのディスクを使用するためです。

B. FTPS ファイル処理に VM コネクタの永続キューを使用する バッチ ジョブ スコープの VM キューを無効にする

C. FTPS ファイル処理にクラウド ハブの永続的な VM キューを使用する

バッチ ジョブ スコープの VM キューを無効にする

D. FTPS ファイル処理にクラウド ハブの永続的な VM キューを使用する

バッチ ジョブ スコープの VM キューを構成する必要はありません。これは、デフォルトで VM キューイングにワーカーの JVM メモリを使用するためです。

正解: **C** ([コメントを发表する](#))

質問: 51

着信 JMS メッセージを listen する JMS コネクタを含む XA トランザクションが構成されています。XA トランザクションのタイムアウト属性の意味と、タイムアウトの期限が切れた後はどうなりますか？

A. トランザクションのコミットから Mule フローの完了までに許容される時間 タイムアウト後、フロー処理はエラーをトリガーします

B. 同じ JMS 接続で JMS メッセージを受信するまでに許容される時間 タイムアウト後、新しい JMS 接続が確立される

C. トランザクションが明示的に終了されずに経過できる時間 タイムアウト後、トランザクションは強制的にロールバックされます

D. 状態 JMS コンシューマ スレッドが破棄されるまでの許容時間 タイムアウト後、新しい JMS コンシューマ スレッドが作成されます。

正解: ([正解を表示します](#))

* Setting a transaction timeout for the Bitronix transaction manager

* Set the transaction timeout either

- In wrapper.conf

- In CloudHub in the Properties tab of the Mule application deployment

* The default is 60 secs. It is defined as

mule.bitronix.transactiontimeout = 120

* This property defines the timeout for each transaction created for this manager.

If the transaction has not terminated before the timeout expires it will be automatically rolled back.

----- Additional Info

around Transaction Management:

Bitronix is available as the XA transaction manager for Mule applications

* To use Bitronix, declare it as a global configuration element in the Mule application

<bti:transaction-manager />

* Each Mule runtime can have only one instance of a Bitronix transaction manager, which is shared by all Mule applications

* For customer-hosted deployments, define the XA transaction manager in a Mule domain

- Then share this global element among all Mule applications in the Mule runtime

質問: 52

HTTP リスナーのキー ストアとトラスト ストアの両方を構成する必要があるのは何ですか?

A. HTTP クライアントとの TLS 相互 (双方向) 認証のサポート

B. サブドメインと API リソース エンドポイントの両方へのリクエストの暗号化

fhTtPs://aDi.customer.com/ および https://customer.com/api)

C. すべての HTTP クライアントの HTTP 要求と HTTP 応答本文の両方の暗号化

D. すべての HTTP クライアントの HTTP リクエスト ヘッダーと HTTP リクエスト ボディの両方の暗号化

正解: ([正解を表示します](#))

1 way SSL : The server presents its certificate to the client and the client adds it to its list of trusted certificate. And so, the client can talk to the server.

2-way SSL: The same principle but both ways. i.e. both the client and the server has to establish trust between themselves using a trusted certificate. In this way of a digital handshake, the server needs to present a certificate to authenticate itself to client and client has to present its certificate to server.

* TLS is a cryptographic protocol that provides communications security for your Mule app.

* TLS offers many different ways of exchanging keys for authentication, encrypting data, and guaranteeing message integrity. Keystores and Truststores. Truststore and keystore contents differ depending on whether they are used for clients or servers:

For servers: the truststore contains certificates of the trusted clients, the keystore contains the private and public key of the server. For clients: the truststore contains certificates of the trusted servers, the keystore contains the private and public key of the client.

Adding both a keystore and a truststore to the configuration implements two-way TLS authentication also known as mutual authentication.

* in this case, correct answer is Support for TLS mutual (two-way) authentication with HTTP clients.

質問: 53

アップストリーム API とその実装を設計するとき、開発チームは、ダウンストリーム API を呼び出すときにタイムアウトを設定しないようにアドバイスされています。ダウンストリーム API には、信頼できる SLA がないためです。これは、そのアップストリーム API の唯一のダウンストリーム API 依存関係です。ダウンストリーム API がクラッシュすることなく中断なく実行されると仮定します。このアドバイスはどのような影響を与えますか？

- A. ダウンストリーム API の呼び出しは、タイムアウトせずに最後まで実行されます。
- B. アップストリーム API の SLA は提供できません。
- C. 500 ミリ秒のデフォルトのタイムアウトは、アップストリーム API 実装が実行される Mule ランタイムによって自動的に適用されます。
- D. ダウンストリーム API 実装が実行される Mule ランタイムによって、1000 ミリ秒未満の負荷依存タイムアウトが適用されます。

正解: ([正解を表示します](#))

An SLA for the upstream API CANNOT be provided.

質問: 54

Mule アプリケーション A は、要求オブジェクトの可変長リストを含むペイロードを含む要求 Anypoint MQ メッセージ REQU を受信します。アプリケーション A は For Each スコープを使用してリストを個々のオブジェクトに分割し、各オブジェクトをメッセージとして Anypoint MQ キューに送信します。サービス S はそのキューをリッスンし、各メッセージを他のすべてのメッセージとは独立して処理し、応答メッセージを応答キューに送信します。

アプリケーション A はその応答キューをリッスンし、次に、REQU で最初に送信された要求オブジェクトと同じ順序でサービス S によって送信された応答のリストを含むペイロードを使用して、応答 Anypoint MQ メッセージ RESP を作成および発行する必要があります。

すべての要求メッセージに対して、サービス S によって正常な応答メッセージが返されたとします。

アプリケーション A がメッセージスループットを最大化しながら、RESP と REQU のオブジェクトリストの長さの順序を一致させるには何が必要ですか？

- A. For Each スコープ内で Scatter-Gather を使用して、応答メッセージの順序を確保する。永続オブジェクトストアを使用して Scatter-Gather を構成する。

- B.** サービス S に関連するすべての通信を For Each スコープ内から同期的に実行するため、RESP のオブジェクトは REQU の要求オブジェクトとまったく同じ順序になります。
- C.** For Each スコープ内で Async スコープを使用し、2 番目の For Each スコープで応答メッセージを到着順に収集し、この応答リストを使用して RESP を送信します。
- D.** For Each スコープとサービスに関連するすべての通信の両方で、リストの長さと REQU のすべてのオブジェクト インデックスを追跡する RESP の作成時に永続ストレージを使用する

正解: **D** ([コメントを发表する](#))

Correct answer is Perform all communication involving service S synchronously from within the For Each scope, so objects in RESP are in the exact same order as request objects in REQU : Using Anypoint MQ, you can create two types of queues: Standard queue These queues don't guarantee a specific message order. Standard queues are the best fit for applications in which messages must be delivered quickly. FIFO (first in, first out) queue These queues ensure that your messages arrive in order. FIFO queues are the best fit for applications requiring strict message ordering and exactly-once delivery, but in which message delivery speed is of less importance Use of FIFO queue is no where in the option and also it decreased throughput. Similarly persistent object store is not the preferred solution approach when you maximizing message throughput. This rules out one of the options. Scatter Gather does not support ObjectStore. This rules out one of the options. Standard Anypoint MQ queues don't guarantee a specific message order hence using another for each block to collect response wont work as requirement here is to ensure the order. Hence considering all the above factors the feasible approach is Perform all communication involving service S synchronously from within the For Each scope, so objects in RESP are in the exact same order as request objects in REQU

質問: **55**

組織は、金融取引データをレガシ システムからデータ ウェアハウス (DWH) にレプリケートするための統合ソリューションを設計しています。

DWH には、CSV ファイルとして配信される金融取引の毎日のスナップショットが含まれている必要があります。1 日の取引量は数千万レコードを超えており、人気のあるショッピング期間中は取引量が大幅に急増します。

組織の現在の要件を満たす統合ソリューションの最も適切な統合スタイルは何ですか？

- A.** イベント駆動型アーキテクチャ
- B.** マイクロサービス アーキテクチャ
- C.** API 主導の接続
- D.** バッチ トリガー ETL

正解: ([正解を表示します](#))

Correct answer is Batch-triggered ETL Within a Mule application, batch processing provides a construct for asynchronously processing larger-than-memory data sets that are split into individual records. Batch jobs allow for the description of a reliable process that automatically splits up source data and stores it into persistent queues, which makes it possible to process large data sets while providing reliability. In the event that the application is redeployed or Mule crashes, the job execution is able to resume at the point it stopped.

質問: 56

展示を参照してください。

Anypoint Platform は、プラットフォームの機能に対するロールベースのアクセス制御 (RBAC) をサポートしています。組織は、Anypoint Platform を使用した ID 管理用の外部 ID プロバイダーを構成しました。Anypoint Platform コントロール プレーンから常に制御する必要があり、外部 ID プロバイダを介して制御できない RBAC の側面はどれですか？

- A. ユーザーが属する Anypoint Platform 内のビジネス グループを制御する
- B. ロールへの Anypoint Platform 権限の割り当て
- C. ユーザーへの Anypoint Platform ロールの割り当て
- D. 組織で勤務しなくなったユーザーの Anypoint Platform へのアクセス権を削除する

正解: ([正解を表示します](#))

* By default, Anypoint Platform performs its own user management

- For user management, one external IdP can be integrated with the Anypoint Platform organization (note: not at business group level)

- Permissions and access control are still enforced inside Anypoint Platform and CANNOT be controlled via the external Identity Provider * As the Anypoint Platform organization administrator, you can configure identity management in Anypoint Platform to set up users for single sign-on (SSO). * You can map users in a federated organization's group to a role which also gives the flexibility of controlling the business group within Anypoint Platform to which the user belongs to. Also user can nbe removed from external identity management system when they no longer work for the organization. So they wont be able to authenticate using SSO to login to Anypoint Platform. * Using external identity we can no change permissions of a particular role in Mulesoft Anypoint platform.

* So Correct answer is Assigning Anypoint Platform permissions to a role

質問: 57

MuleSoft が提供する Maven プラグインを使用して、Mule アプリケーションの CI/CD パイプラインのどの側面を自動化できますか？

- A. コンパイル、パッケージ化、単体テスト、単体テスト カバレッジの検証、デプロイ
- B. コンパイル、パッケージ、単体テスト、デプロイ、統合テスト
- C. コンパイル、パッケージ化、単体テスト、展開、API Manager B で関連する API インスタンスを作成 API デザイナーからインポート、コンパイル、パッケージ化、単体テスト、展開、Am/point Exchange に公開

正解: A ([コメントを発表する](#))

質問: 58

API クライアントは、デフォルト設定を使用する HTTP リクエスト操作を含む Mule アプリケーションとして実装されます。HTTP リクエスト操作は、標準の HTTP ステータス コード規則に従う外部 API を呼び出します。これにより、HTTP リクエスト操作は 4xx ステータス コードを返します。このステータス コードの応答の考えられる原因は何ですか？

- A. Mule アプリケーションの送信 HTTP 要求操作から受信した HTTP 要求を処理するときに、外部 API 実装内でエラーが発生しました
- B. 外部 API は、API 実装が別の外部エンドポイントに移動したことを報告しました
- C. HTTP 応答を外部 API から受信した後、Mule アプリケーションの HTTP 要求操作で解釈できません。
- D. 外部 API は、Mule アプリケーションの送信 HTTP 要求操作から受信した HTTP 要求でエラーを報告しました

正解: ([正解を表示します](#))

Correct choice is: "The external API reported an error with the HTTP request that was received from the outbound HTTP Request operation of the Mule application" Understanding HTTP 4XX Client Error Response Codes : A 4XX Error is an error that arises in cases where there is a problem with the user's request, and not with the server.

Such cases usually arise when a user's access to a webpage is restricted, the user misspells the URL, or when a webpage is nonexistent or removed from the public's view.

In short, it is an error that occurs because of a mismatch between what a user is trying to access, and its availability to the user - either because the user does not have the right to access it, or because what the user is trying to access simply does not exist. Some of the examples of 4XX errors are

400 Bad Request The server could not understand the request due to invalid syntax. 401 Unauthorized Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response. 403 Forbidden The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource. Unlike 401, the client's identity is known to the server. 404 Not Found The server can not find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to hide the existence of a resource from an unauthorized client. This response code is probably the most famous one due to its frequent occurrence on the web. 405 Method Not Allowed The request method is known by the server but has been disabled and cannot be used. For example, an API may forbid DELETE-ing a resource. The two mandatory methods, GET and HEAD, must never be disabled and should not return this error code. 406 Not Acceptable This response is sent when the web server, after performing server-driven content negotiation, doesn't find any content that conforms to the criteria given by the user agent. The external API reported that the API implementation has moved to a different external endpoint cannot be the correct answer as in this situation 301 Moved Permanently The URL of the requested resource has been changed permanently. The new URL is given in the response.

In Lay man's term the scenario would be: API CLIENT -> MuleSoft API - HTTP request "Hey, API.. process this" -> External API API CLIENT <- MuleSoft API - http response "I'm sorry Client.. something is wrong with that request" <- (4XX) External API

Anypoint Analytics を使用して、カスタム チャートで視覚化するために使用できる API 呼び出しに関するメトリクスは何ですか？

- A. リクエスト サイズ、リクエスト HTTP 動詞、応答時間
- B. リクエストサイズ、リクエスト数、JDBC Select操作結果セットサイズ
- C. リクエストサイズ、リクエスト数、レスポンスサイズ、レスポンスタイム
- D. リクエストサイズ、リクエスト数、JDBC Select 操作応答時間

正解: ([正解を表示します](#))

Correct answer is Request size, number of requests, response size, response time Analytics API Analytics can provide insight into how your APIs are being used and how they are performing. From API Manager, you can access the Analytics dashboard, create a custom dashboard, create and manage charts, and create reports. From API Manager, you can get following types of analytics: - API viewing analytics - API events analytics - Charted metrics in API Manager It can be accessed using:

<http://anypoint.mulesoft.com/analytics> API Analytics provides a summary in chart form of requests, top apps, and latency for a particular duration.

The custom dashboard in Anypoint Analytics contains a set of charts for a single API or for all APIs Each chart displays various API characteristics

- Requests size: Line chart representing size of requests in KBs
- Requests : Line chart representing number of requests over a period
- Response size : Line chart representing size of response in KBs
- Response time :Line chart representing response time in ms

* To check this, You can go to API Manager > Analytics > Custom Dashboard > Edit Dashboard > Create Chart > Metric

Reference:

Additional Information:

The default dashboard contains a set of charts

- Requests by date: Line chart representing number of requests
- Requests by location: Map chart showing the number of requests for each country of origin
- Requests by application: Bar chart showing the number of requests from each of the top five registered applications
- Requests by platform: Ring chart showing the number of requests broken down by platform

質問: 60

組織は、モバイル アプリと Web アプリケーションの両方から顧客データにアクセスできるようにする必要があります。それぞれが共通のフィールドと特定の固有のフィールドにアクセスする必要があります。データの一部はデータベースで、一部はサードパーティの CRM システムで利用できます。これらの設計要件に最も適合するには、どの API を作成する必要がありますか？

A. Web アプリとモバイル アプリの両方で必要なデータを含むプロセス API であり、これらのアプリケーションが直接呼び出して必要なデータにアクセスできるようにすることで、API の変更を必要とせず将来的にフィールドを追加する柔軟性を提供します。

B. Web アプリ用の 1 つの API セット (エクスペリエンス API、プロセス API、およびシステム API) と、モバイル アプリ用の別のセット。

C. モバイル アプリと Web アプリ用に別々のエクスペリエンス API を使用しますが、データベースと CRM システム用に作成された別々のシステム API を呼び出す共通のプロセス API です。

D. Web アプリとモバイル アプリの両方で使用される共通のエクスペリエンス API ですが、データベースと CRM システムとやり取りする Web アプリとモバイル アプリの個別の Process API です。

正解: C ([コメントを发表する](#))

Lets analyze the situation in regards to the different options available
Option : A common Experience API but separate Process APIs
Analysis : This solution will not work because having common experience layer will not help the purpose as mobile and web applications will have different set of requirements which cannot be fulfilled by single experience layer API
Option : Common Process API
Analysis : This solution will not work because creating a common process API will impose limitations in terms of flexibility to customize API;s as per the requirements of different applications. It is not a recommended approach.
Option : Separate set of API's for both the applications
Analysis : This goes against the principle of Anypoint API-led connectivity approach which promotes creating reusable assets. This solution may work but this is not efficient solution and creates duplicity of code.

Hence the correct answer is: Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system

Lets analyze the situation in regards to the different options available
Option : A common Experience API but separate Process APIs
Analysis : This solution will not work because having common experience layer will not help the purpose as mobile and web applications will have different set of requirements which cannot be fulfilled by single experience layer API
Option : Common Process API
Analysis : This solution will not work because creating a common process API will impose limitations in terms of flexibility to customize API;s as per the requirements of different applications. It is not a recommended approach.
Option : Separate set of API's for both the applications
Analysis : This goes against the principle of Anypoint API-led connectivity approach which promotes creating reusable assets. This solution may work but this is not efficient solution and creates duplicity of code.

Hence the correct answer is: Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system

有効的な**MCIA-Level-1-JPN**問題集はJPNTTest.com提供され、**MCIA-Level-1-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**MCIA-Level-1-JPN**試験問題集を提供します。JPNTTest.com MCIA-Level-1-JPN試験問題集はもう更新されました。ここで**MCIA-Level-1-JPN**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/MCIA-Level-1-JPN-mondaishu>
275問、30%ディスカウント、特別な割引コード: JPNshiken」