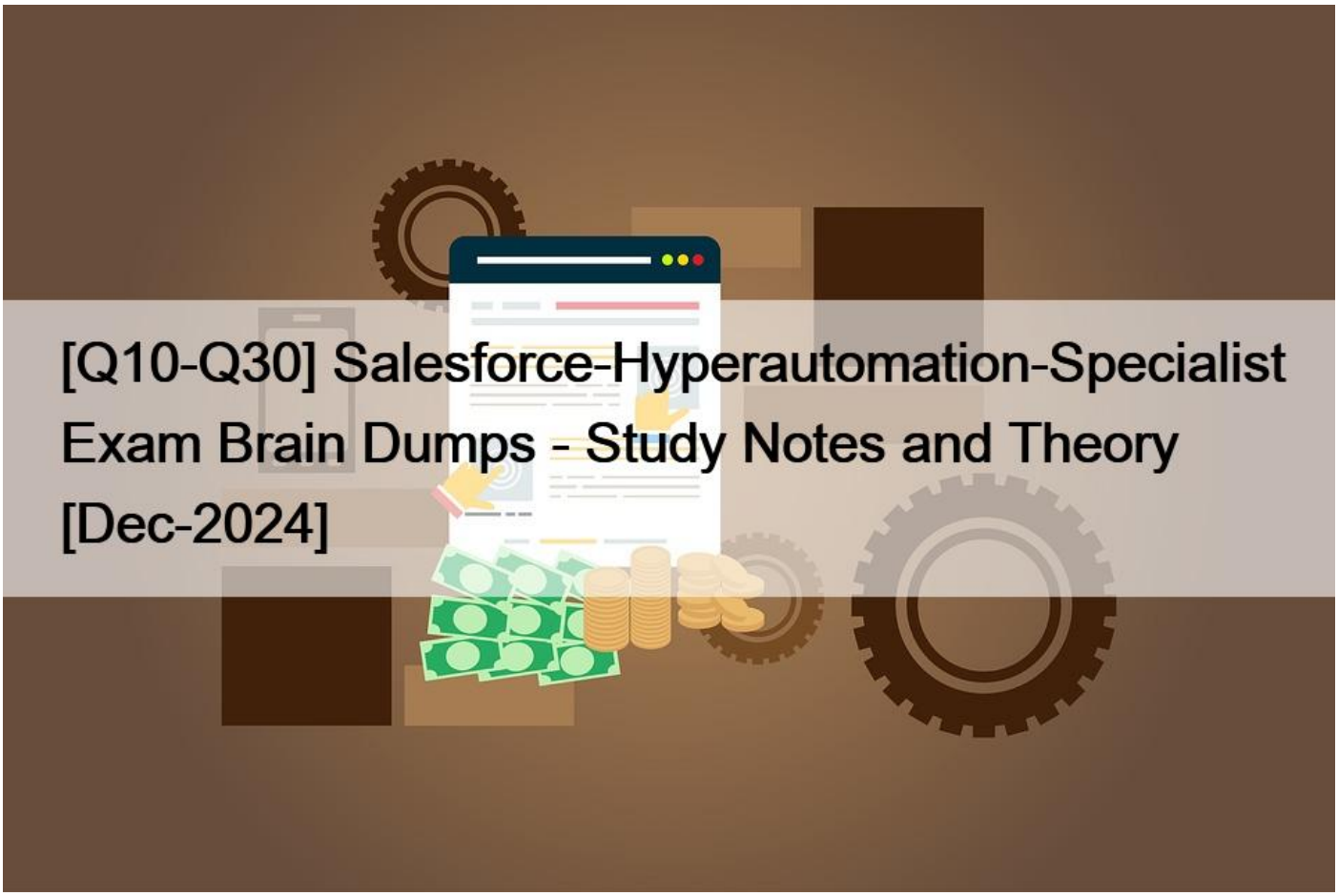
[Q10-Q30 Salesforce-Hyperautomation-Specialist Exam Brain Dumps - Study Notes and Theory [Dec-2024



Salesforce-Hyperautomation-Specialist Exam Brain Dumps - Study Notes and Theory [Dec-2024]
100% Guaranteed Results Salesforce-Hyperautomation-Specialist Unlimited 62 Questions

Salesforce Salesforce-Hyperautomation-Specialist Exam Syllabus Topics:

TopicDetailsTopic 1- Use Salesforce Flow Orchestrator to build parallel, multi-user, multi-step workstreams: This part focuses on combining automated workflows, customizing entry and exit conditions, assigning interactive steps, and managing Flow Orchestration.Topic 2- Use Composer to automate data integrations for hyperautomation: This part focuses on using Composer flows and connectors, HTTP connectors, sandbox to production transitions, flow controls, data transformation, and testing Composer flows.Topic 3- Use Anypoint platform to deliver, and manage APIs in a hyperautomation project: This section focuses on composable building blocks, API-led connectivity, functional design requirements, RAML, Anypoint Platform capabilities, and Mule application deployment options.Topic 4- Use Anypoint Platform to monitor hyperautomation API endpoints: This part covers managing APIs using endpoint configurations and policies and describes Anypoint Monitoring for applications and APIs.Topic 5- Design, build, and manage MuleSoft RPA processes used for hyperautomation: This section deals with evaluating, developing, deploying, triggering, monitoring, and troubleshooting RPA processes in MuleSoft RPA Manager.Topic 6- Salesforce-Hyperautomation-Specialist: This section covers identifying appropriate tools, understanding drawbacks of manual tasks, integration solutions, MuleSoft RPA processes, testing, design patterns, fault handling, reuse scenarios, and development velocity in hyperautomation.

NO.10 Northern Trail Outfitters developed an integration between its two Salesforce orgs using MuleSoft Composer.

Which two actions should be taken before testing the Composer flow? (Choose two.)
* Ensure the flow trigger is connected to a sandbox instance of Salesforce.
* Ensure action steps are connected to a sandbox instance of Salesforce.
* Ensure the credentials to the target production org are still valid.
* Ensure MuleSoft Composer is installed on both the source and target orgs.
* Flow Trigger Connection: Before testing any Composer flow, it is crucial to connect the flow trigger to a sandbox instance of Salesforce. This ensures that testing does not impact the production environment. The sandbox provides a safe space to simulate real-world conditions without the risk of data corruption or unintended actions in the live system.

Reference:

* Action Steps Connection: Similar to the flow trigger, action steps within the Composer flow should also be connected to a sandbox instance. This allows comprehensive testing of the flow’s functionality, ensuring that each step performs as expected without affecting the production data.

* Ensuring Validity of Credentials: While it is important to ensure that credentials to the production org are valid when moving to production, for testing purposes, the emphasis is on sandbox connections. The credentials should be verified to avoid disruptions during testing.

* Installation of MuleSoft Composer: MuleSoft Composer does not need to be installed on both the source and target orgs as it operates independently and connects to these orgs through provided credentials.

NO.11 AnyAirlines needs to select a tool for developing an integration between Salesforce and an ERP system in the cloud. The requirements state that the systems must communicate bidirectionally and as close to real time as possible. The ERP system can be accessed via a SOAP-based web service.

Which tool meets the requirements of this integration?
* Anypoint Studio
* MuleSoft Composer
* Orchestrator
* MuleSoft RPA
Anypoint Studio is the most suitable tool for developing an integration between Salesforce and an ERP system in the cloud, especially when bidirectional communication in near real-time is required. Here’s why:

SOAP-Based Web Services Support:

Anypoint Studio supports SOAP-based web services, which is essential since the ERP system can be accessed via SOAP.

Bidirectional Communication:

Anypoint Studio allows for complex integrations that require bidirectional communication. You can set up flows that handle both inbound and outbound data seamlessly.

Real-Time Integration:

With Anypoint Studio, you can design integrations that process data in near real-time, ensuring timely updates between Salesforce and the ERP system.

Advanced Integration Capabilities:

Anypoint Studio offers advanced capabilities for data transformation, error handling, and connecting various systems, making it ideal for complex integration scenarios.

Reference:

Anypoint Studio Documentation

NO.12 Northern Trail Outfitters (NTO) has a new business channel that requires exposing their existing non-MuleSoft APIs to the public. They do not have an Anypoint Flex Gateway.

The NTO digital channel team wants to leverage Anypoint Platform as its API management tool.

What is the most time-efficient mechanism of securing their backend systems?
* Rewrite the existing APIs using MuleSoft.
* Create a proxy in front of each existing API.
* Expose each endpoint as a basic endpoint.
* Use a basic endpoint with a configured consumer endpoint.

To expose existing non-MuleSoft APIs and secure them efficiently using Anypoint Platform, you can create API proxies. This approach provides several benefits:

API Proxy Creation:

Creating a proxy involves setting up an intermediary that forwards requests to the existing backend APIs. This allows you to leverage Anypoint Platform’s API management capabilities without rewriting the existing APIs.

Proxies can be created quickly and configured to apply various security and governance policies.

Security and Management:

By creating a proxy, you can secure the APIs using Anypoint Platform’s features such as rate limiting, authentication, and monitoring.

This method is time-efficient and leverages the robust security features of the Anypoint Platform without significant redevelopment effort.

Reference:

Anypoint Platform API Proxy Documentation

NO.13 Northern Trail Outfitters (NTO) wants to automate a multi-step process that spans several departments.

How do Interactive Steps in Flow Orchestration help NTO involve users at key steps of the process?
* They allow the user to interact directly with external systems through the Salesforce Ul.
* They allow the user to interact with the process in between automated backend steps.
* They enable users to collaborate on specific work items.

* They leverage Al processing to automatically interact with the customer and collect customer data.
Interactive Steps in Flow Orchestration are designed to involve users at specific points within an automated process. Here’s how they help NTO:

User Interaction:

Interactive Steps enable users to engage with the process during key stages. These steps are inserted between automated tasks to require human input or decision-making.

This ensures that critical user actions, such as approvals or data entry, are seamlessly integrated into the flow.

Process Continuity:

Once the user completes the required interaction, the process can automatically proceed to the next step. This creates a cohesive workflow that combines automated and manual tasks efficiently.

Use Case Examples:

Approving a document, entering additional information, or making decisions based on presented data are typical scenarios where Interactive Steps are beneficial.

Reference:

Salesforce Flow Orchestration Documentation

NO.14 A MuleSoft developer at AnyAirlines is tasked with creating a new API for an integration.

According to best practices, what is the first step they need to perform?
* Create a new project in Anypoint Studio.
* Install a standalone Mule runtime on their local machine.
* Create a case in Salesforce.
* Create a RAML definition in Design Center.
* RAML Definition Creation: The first step in creating a new API as per MuleSoft best practices is to create a RAML (RESTful API Modeling Language) definition in the Design Center. This step is critical as it outlines the API’s structure, endpoints, methods, and data types, providing a clear blueprint for subsequent development.

Reference:

* Project Creation in Anypoint Studio: Once the RAML definition is created, the next step would be to generate the API project in Anypoint Studio. This IDE allows developers to implement the API logic as defined in the RAML.

* Mule Runtime Installation: Installing Mule runtime is necessary for running and testing Mule applications locally. However, this step is secondary to defining the API’s structure.

* Case Creation in Salesforce: Creating a case in Salesforce is not relevant to the API development process but may be necessary for support or project management purposes.

NO.15 The customer support team at Northern Trail Outfitters manages and maintains customer service cases using Service Cloud. The team collaborates with other stakeholders such as the sales, product, and technical support teams to resolve cases using Slack.

The team needs to use a MuleSoft Composer flow to automatically trigger when a case is created or modified in Service Cloud with notifications in Slack. Based on these specific case requirements, the team routes the cases to the sales, product, or the technical support team.

What flow component must the customer support team use to route the cases?
* For Each
* If/Else
* Switch/Case
* Swimlane

To route cases based on specific criteria to different teams (sales, product, or technical support) using MuleSoft Composer, the Switch/Case component is the most appropriate choice:

Create a MuleSoft Composer Flow:

Start by creating a flow in MuleSoft Composer that triggers when a case is created or modified in Service Cloud.

Use the Switch/Case Component:

Add a Switch/Case component to the flow. This component allows you to define multiple conditions and route the flow based on these conditions.

Define the different case routing criteria (e.g., case type, priority) within the Switch/Case component. For each case, specify the condition that determines which team the case should be routed to.

Configure Notifications in Slack:

For each case defined in the Switch/Case component, configure the corresponding actions to send notifications to the appropriate Slack channels.

The Switch/Case component enables complex conditional logic, making it ideal for routing cases to different teams based on predefined criteria.

Reference:

MuleSoft Composer Documentation

NO.16 AnyAirlines is developing an RPA process and is implementing testing best practices. They want to take the RPA process through rigorous testing.

During these tests, where do RPA process test plans execute?
* On a configured RPA Bot
* In RPA Manager
* In RPA Builder
* In an RPA process runtime

During testing of an RPA process, test plans are executed on a configured RPA Bot. This allows you to simulate real-world scenarios and ensure the RPA process works correctly under various conditions:

On a Configured RPA Bot:

RPA Bots are configured to execute the automated tasks defined in the RPA process. By running test plans on these bots, you can

verify the functionality and performance of the RPA process.

This approach ensures that the RPA process is thoroughly tested in an environment that closely mirrors production conditions.

Reference:

MuleSoft RPA Documentation

NO.17 A Salesforce flow needs to connect to external APIs provided by Northern Trail Outfitters (NTO) and AnyAirlines to retrieve data.

Which three steps should be taken to connect to the external APIs? (Choose three.)
* Use an Action element to call and consume the appropriate API in the Salesforce flow.
* Create External Services in Salesforce for NTO and AnyAirlines.
* Create Named Credentials in Anypoint for NTO and AnyAirlines.
* Use a Virtual service to call and consume the appropriate API in the Salesforce flow.
* Create Named Credentials in Salesforce for NTO and AnyAirlines.

To connect a Salesforce flow to external APIs, follow these steps:

Create Named Credentials in Salesforce:

Named Credentials in Salesforce simplify the authentication process by storing the URL and authentication details needed to call the external service. Set up Named Credentials for both NTO and AnyAirlines APIs.

Navigate to Salesforce Setup > Named Credentials > New Named Credential.

Enter the details for NTO API, including the endpoint URL and authentication settings.

Repeat the process for the AnyAirlines API.

Create External Services in Salesforce:

External Services allow you to register the external APIs with Salesforce so that they can be invoked from a flow.

Navigate to Salesforce Setup > External Services > New External Service.

Use the OpenAPI/Swagger or RAML specification files provided by NTO and AnyAirlines to register their APIs.

This step involves importing the API specifications and configuring the services, which will be used in the flow.

Use an Action Element in Salesforce Flow:

Within Salesforce Flow, use the Action element to call the external services that were registered.

This involves adding an Action to your flow and selecting the appropriate External Service action that corresponds to the API method you want to invoke.

Reference:

Salesforce External Services Documentation

Salesforce Named Credentials Documentation

NO.18 AnyAirlines implements a credit card program that requires customer applications to go through a review process before approval. They want to develop a series of hyperautomation solutions that will integrate to process the applications and enter the customer’s information into a legacy system once approved.

They want to complete the following components:

An Einstein bot that will initiate the credit card application and create a record of an existing Salesforce Custom Object A Salesforce flow that marks the credit card application as approved in Salesforce An RPA process that interacts with multiple applications and websites A simple MuleSoft Composer flow that triggers if a credit card application is approved and then invokes an RPA process Which component will likely require the most effort to complete?
* A simple MuleSoft Composer flow that triggers if a credit card application is approved and then invokes an RPA process
* A Salesforce flow that marks the credit card application as approved in Salesforce
* An RPA process that interacts with multiple applications and websites
* An Einstein bot that will initiate the credit card application and create a record of an existing Salesforce Custom Object
Developing an RPA process that interacts with multiple applications and websites typically requires the most effort due to several factors:

Complexity of Interaction:

RPA processes involve simulating human actions to interact with different user interfaces. This includes navigating web pages, filling out forms, and clicking buttons, which can be complex and time-consuming to script and test.

Integration Challenges:

The RPA process must handle different applications and websites, each with unique behaviors and potential for errors. Ensuring reliable and consistent interaction across these systems requires thorough testing and potentially custom handling for each system.

Maintenance and Updates:

RPA processes need to be maintained and updated as the applications or websites they interact with change. This ongoing effort can be significant compared to other components.

Reference:

MuleSoft RPA Documentation

NO.19 AnyAirlines is developing an RPA process to extract information from a legacy system. To capture the manual workflow, they leverage RPA Recorder.

Which two best practices should they be aware of when working with the autogenerated workflow code? (Choose two.)
* All autocaptured information is for documentation purposes only.
* Some autogenerated code must be replaced with more robust or specialized action steps.
* The autogenerated workflows may contain sensitive information that must be removed.
* All keystrokes and mouse clicks in the autogenerated code must be disabled before deploying to production.
When developing an RPA process using RPA Recorder, it is essential to be mindful of the following best practices concerning the autogenerated workflow code:

Replace Autogenerated Code:

Robustness: Some of the autogenerated code may not be optimized for robustness or specific use cases. It is often necessary to review and replace parts of the autogenerated workflow with more robust or specialized action steps to ensure reliability and accuracy.

Specialization: Customizing the workflow to fit the specific requirements of the process can improve performance and handle exceptions better.

Remove Sensitive Information:

Sensitive Data: Autogenerated workflows might capture sensitive information such as usernames, passwords, or other confidential data. It is crucial to identify and remove or mask this information before deploying the RPA process to production to maintain security and compliance.

Compliance: Ensuring that sensitive information is handled appropriately helps in adhering to data protection regulations and organizational policies.

Reference:

MuleSoft RPA Documentation

NO.20 A non-technical employee from AnyAirlines creates a hyperautomation solution. The solution needs to meet the following criteria:

The process needs to begin when a record is created in Salesforce.

Then, it needs to pass data to a pre-existing RPA process which includes a User Task for data integrity purposes.

The output of the RPA process needs to be used to create a record in NetSuite.

According to best practices, how should this automated process be structured?
* 1. A MuleSoft Composer flow triggers on the creation of the record and calls the RPA process.

2. Then, the same flow uses the response to create a record in NetSuite.

B 1. A Salesforce flow triggers on the creation of the record and calls too RPA process.

2. Then, the same flow uses the response to create a record in NetSuite.
* 1. A MuleSoft Composer flow triggers on the creation of the record and calls the RPA process.

2. Then, a second MuleSoft Composer flow triggers when the RPA process is completed and creates a record in NetSuite.
* 1. A Salesforce flow triggers on the creation of the record and makes an outbound request to a MuleSoft Composer flow.

2. Then, the same MuleSoft Composer flow calls the RPA process and uses the result to create a record in NetSuite.
To structure the automated process to meet the given criteria, the following approach is recommended:

MuleSoft Composer Flow Triggers on Record Creation:

Use MuleSoft Composer to create a flow that is triggered when a new record is created in Salesforce. This is done by setting up a

trigger event in MuleSoft Composer that listens for new record creation events in Salesforce.

Call the RPA Process:

Once the flow is triggered, it should call the pre-existing RPA process. MuleSoft Composer can invoke MuleSoft RPA bots, and you can pass the necessary data from the Salesforce record to the RPA process.

The RPA process will include the User Task for data integrity purposes.

Use the RPA Process Output to Create a Record in NetSuite:

After the RPA process completes, the MuleSoft Composer flow can capture the output from the RPA process.

The same MuleSoft Composer flow will then use this output to create a record in NetSuite, ensuring a seamless data transfer and process automation.

Reference:

MuleSoft Composer Documentation

MuleSoft RPA Documentation

NO.21 Northern Trail Outfitters (NTO) uses Flow Orchestration to automate quote development. The “Review Quote” work item is performed by their team of technical writers but can be fulfilled by any technical writer on the team.

How can NTO ensure the “Review Quote” work item is assigned to the correct Salesforce user?
* Use backend steps to automate work item assignment to the next available technical writer.
* Create a Group for the team of Salesforce Users and assign the work item to the group.
* Use MuleSoft RPAto review the document and submit it for approval if no issues are found.
* Create a user collection variable and assign the work item to the user collection.

To ensure the “Review Quote” work item is assigned to the correct Salesforce user within the team of technical writers, the following approach can be used:

Creating a Group:

Create a Group in Salesforce that includes all the technical writers who are eligible to perform the “Review Quote” work item. This group acts as a collective resource pool.

Assigning the Work Item to the Group:

When the “Review Quote” work item is created in Flow Orchestration, assign it to the group rather than an individual user. Salesforce will then allow any available technical writer within the group to pick up and complete the task.

Ensuring Flexibility and Availability:

This method ensures that the work item can be completed by any technical writer in the team, providing flexibility and improving the chances of timely completion by utilizing the group’s collective availability.

Reference:

Salesforce Flow Orchestration Documentation

NO.22 For a MuleSoft Composer flow, errors can be noted in its Flow Details page.

What other way can MuleSoft Composer send notifications when errors occur?
* It posts to a configured Chatter profile.
* It generates a notification in the flow.
* It sends a message to a configured Slack channel.
* It sends a notification to the configured email address.

MuleSoft Composer provides a way to handle errors and notify users when something goes wrong in a flow. Aside from viewing errors on the Flow Details page, MuleSoft Composer can also send notifications to alert users about the errors.

Flow Error Handling: When an error occurs in a MuleSoft Composer flow, the error is logged and visible on the Flow Details page.

Email Notifications: MuleSoft Composer can be configured to send notifications to a specified email address. This allows users to be promptly informed of any issues without having to constantly monitor the Flow Details page.

Configuration: This can be set up in the MuleSoft Composer settings, where an email address can be configured to receive these notifications.

NO.23 A MuleSoft developer at AnyAirlines wants to retrieve customer data from an external system.

Before designing a new integration, what should they use to determine if the integration exists and can be reused?
* Design Center
* Anypoint Studio
* Anypoint Exchange
* MuleSoft Composer

To determine if an integration exists and can be reused, the MuleSoft developer should use Anypoint Exchange:

Anypoint Exchange:

Anypoint Exchange is a repository where developers can publish, share, and discover reusable assets such as APIs, connectors, templates, and examples. It serves as a centralized location for all reusable components within the MuleSoft ecosystem.

By searching Anypoint Exchange, the developer can find existing integrations or assets that might fulfill the requirements for retrieving customer data, avoiding the need to design and develop a new integration from scratch.

Reference:

Anypoint Exchange Documentation

NO.24 Northern Trail Outfitters wants to create an automation which runs on a fixed schedule to enter sales data into NetSuite running as a process in the background. The business product owner chose MuleSoft Composer as the tool for this task.

The Salesforce admin wants to advise the product owner about how the MuleSoft Composer scheduling functionality works.

Which two options are available for use as the time mechanism within MuleSoft Composer? (Choose two.)
* Schedule based on a formula
* Every 30 days
* Every 30 minutes

* Every 5 minutes

NO.25 Which API policy can be applied to limit the number of requests an individual client can make to an API?
* Client ID Enforcement
* Spike Control
* Rate limiting – SLA-Based
* OAuth 2.0 access token enforcement

The Rate Limiting – SLA-Based policy in Anypoint Platform is designed to control the number of requests an individual client can make to an API. This policy is highly configurable and allows you to set specific limits based on service level agreements (SLAs).

Rate Limiting – SLA-Based:

This policy helps protect APIs from being overwhelmed by too many requests by enforcing a limit on the number of requests a client can make within a specified time frame.

You can define different rate limits for different tiers of clients, ensuring fair usage and protecting backend services.

Reference:

Anypoint Platform Rate Limiting Documentation

NO.26 Northern Trail Outfitters (NTO) is building a hyperautomation solution using Salesforce and MuleSoft. Their Salesforce admin needs to automate a comprehensive, multi-step process that a single user will execute after a case record is created.

How should the Salesforce Flow solution be structured to meet this requirement?
* An autolaunched flow that will process user inputs and conditional logic to automate the process in Salesforce
* A single flow Orchestration that uses Stages and Steps to organize automated actions and process user inputs
* A screen flow to process user inputs and an autolaunched flow to process backend steps automatically
* A parent flow with subflows to help organize automated actions and generate reusable components

To address the comprehensive, multi-step process automation requirement at Northern Trail Outfitters (NTO), a single flow orchestration that uses Stages and Steps is the best solution.

Flow Orchestration in Salesforce:

Stages and Steps: Flow Orchestration allows Salesforce admins to build sophisticated automations by structuring the flow into Stages (representing different parts of the process) and Steps (individual actions within each Stage).

User Inputs and Automated Actions: By leveraging Stages and Steps, Salesforce Flow Orchestration can handle both user inputs and backend automated steps seamlessly, ensuring the entire process is automated and organized efficiently.

Error Handling and Conditional Logic: It also allows for conditional logic and error handling, ensuring that the flow can adapt to various scenarios that may arise during the automation process.

Comprehensive Process Automation:

Single User Execution: Given that the requirement specifies that a single user will execute the process after a case record is created, Flow Orchestration is ideal as it can manage the end-to-end process in a structured manner, without requiring multiple flows or complex configurations.

Reference:

Salesforce documentation on Flow Orchestration provides detailed insights on how to design and implement such solutions.

NO.27 Which MuleSoft deployment strategy consists of the control plane and runtime plan hosted by the client?
* IPrivate Cloud Edition
* CloudHub
* Runtime Fabric
* Hybrid

A hybrid deployment strategy in MuleSoft involves hosting the control plane (Anypoint Platform management and design tools) in the cloud, while the runtime plane (where Mule applications run) is hosted by the client, either on-premises or in their own private cloud:

Hybrid Deployment:

The control plane is managed by MuleSoft and provides centralized management, monitoring, and deployment capabilities.

The runtime plane is hosted by the client, providing flexibility and control over where and how the Mule applications are executed, whether on-premises or in a private cloud environment.

Benefits:

This approach combines the advantages of cloud-based management with the control and customization available in on-premises or private cloud deployments, making it suitable for organizations with specific hosting and compliance requirements.

Reference:

MuleSoft Hybrid Deployment Documentation

NO.28 Any Airlines is developing a new integration and wants built-in automated testing.

Which tool must be used to satisfy this requirement?
* MuleSoft RPA
* MuleSoft Composer
* Flow Orchestration
* Anypoint Platform

To implement built-in automated testing for new integrations at Any Airlines, the Anypoint Platform is the appropriate tool.

Anypoint Platform Capabilities:

Automated Testing: Anypoint Platform includes various tools such as MUnit for automated testing of Mule applications. MUnit allows developers to create, design, and run tests natively within Anypoint Studio.

Test Automation Features: It supports comprehensive testing features including unit tests, integration tests, and mock services to ensure robust and reliable integrations.

Continuous Integration and Deployment: Anypoint Platform can be integrated with CI/CD pipelines, allowing automated tests to run as part of the deployment process, ensuring that any new code changes do not break existing functionality.

Why Not Other Options:

MuleSoft RPA: Primarily used for automating repetitive manual tasks, not for testing integrations.

MuleSoft Composer: Focuses on low-code integrations and automation, not specifically designed for automated testing.

Flow Orchestration: While useful for process automation within Salesforce, it does not provide the testing capabilities required for MuleSoft integrations.

Reference:

For detailed information on automated testing with Anypoint Platform and MUnit, refer to the official MuleSoft documentation

NO.29 Northern Trail Outfitters evaluates multiple standards for the exit criteria of a stage in their Flow Orchestration. Based on their criteria, they want the flow to go down one of three paths.

How should this be built in Flow Orchestration to meet this requirement?
* Use the evaluation flow to determine the exit criteria for the current stage. Then, use a separate evaluation flow to determine the entry criteria for each of the three paths.
* Have the evaluation flow return a number variable, and use a decision element to determine which path to execute.
* Create two evaluation flows, and execute the second evaluation flow if the first evaluation flow returns false.
* Evaluate the criteria for the first two paths in an evaluation flow. Then, use the default path functionality of the decision element for the third path.

To implement branching logic based on multiple criteria in Flow Orchestration, you can use the following approach:

Evaluation Flow Returns a Number Variable:

Create an evaluation flow that assesses the exit criteria for the current stage and returns a number variable indicating which path to take (e.g., 1, 2, or 3).

Decision Element:

Use a decision element in Flow Orchestration to evaluate the number variable returned by the evaluation flow. Based on the value of the variable, the decision element will determine which path to execute next.

This approach allows for clear and maintainable branching logic, ensuring that the flow can proceed down one of three paths based on the defined criteria.

Reference:

Salesforce Flow Orchestration Documentation

NO.30 Northern Trail Outfitters must create a near real-time inventory API that can be used within its retail POS systems, across its mobile and online stores, and by its strategic B2B e-commerce partners. The API must provide accurate and up-to-date product inventory levels. The data currently resides in both SAP and NetSuite.

According to best practices, which hyperautomation tool should be used to build this solution?
* Salesforce Flow
* MuleSoft Composer
* MuleSoft RPA
* Anypoint Platform

To create a near real-time inventory API that integrates data from SAP and NetSuite and can be used across various platforms and partners, the Anypoint Platform is the most suitable tool:

Anypoint Platform:

Anypoint Platform provides comprehensive integration capabilities, including real-time data processing, API management, and connectivity to various systems like SAP and NetSuite.

It supports building robust, scalable APIs that can handle near real-time data synchronization, ensuring accurate and up-to-date inventory levels across multiple channels.

Best Practices:

Using Anypoint Platform, you can design and manage APIs with fine-grained control over security, performance, and monitoring, adhering to best practices for enterprise integration.

Reference:

Anypoint Platform Documentation