

DUMPS ARENA

Salesforce Certified MuleSoft Integration Associate

Salesforce MuleSoft-Integration-Associate

Version Demo

Total Demo Questions: 6

Total Premium Questions: 62

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Integration Fundamentals	27
Topic 2, API-led Connectivity	7
Topic 3, Anypoint Platform	21
Topic 4, Application Networks	2
Topic 5, MuleSoft Catalyst	5
Total	62

QUESTION NO: 1

According to MuleSoft what is a major distinguishing characteristic of an application network in relation to the integration of systems, data, and devices?

- A. It uses a well-organized monolithic approach with standards
- B. It is built for change and self-service
- C. It leverages well-accepted internet standards like HTTP and JSON
- D. It uses CI/CD automation for real-time project delivery

ANSWER: B

Explanation:

It is built for change and self-service is correct because MuleSoft defines an application network as an approach to enterprise architecture that enables organizations to connect applications, data, and devices through reusable APIs. Its distinguishing value is not simply making an initial integration work; it is creating a network of capabilities that can be discovered, reused, composed, and adapted as business requirements evolve. APIs expose systems in manageable layers, allowing teams to add new channels, automate processes, or support new consumer experiences with less duplication and less dependency on one central integration team.

Self-service is fundamental because API consumers can discover and use governed, reusable assets through a shared platform such as Anypoint Exchange. This lets teams build integrations and digital experiences more independently while applying consistent security, governance, and lifecycle practices. Being built for change follows from the same API-led design: changes can be isolated behind API contracts and reusable building blocks, helping the organization respond more quickly to new requirements. MuleSoft describes application networks and API-led connectivity as strategies for making enterprise capabilities reusable and for accelerating delivery across the organization. See [MuleSoft: What is an application network?](#) and [MuleSoft API-led connectivity documentation](#).

QUESTION NO: 2

What are two reasons why a typical MuleSoft customer favors a MuleSoft-hosted Anypoint Platform runtime plane over a customer-hosted runtime for its Mule application deployments? (Choose two.)

- A. Reduced time-to-market for the first application
- B. Reduced application latency
- C. Reduced IT operations effort
- D. increased application isolation
- E. Increased application throughput

ANSWER: A C

Explanation:

Reduced time-to-market for the first application is a key benefit of choosing a MuleSoft-hosted runtime plane such as CloudHub. The customer can provision a managed deployment environment through Anypoint Platform rather than first designing, procuring, configuring, securing, and validating its own runtime infrastructure. This lets teams focus earlier on building integrations, configuring application properties, and deploying Mule applications. MuleSoft describes CloudHub as a fully managed integration platform as a service, with deployment and operational capabilities integrated into Anypoint Platform.

Reduced IT operations effort is also a primary reason to favor a MuleSoft-hosted runtime. MuleSoft operates the underlying platform components and provides managed capabilities for runtime deployment, monitoring, logging, scaling configuration, and platform maintenance. Although customers remain responsible for their applications, integration design, and appropriate configuration, they do not need to perform the same level of infrastructure administration required for a customer-hosted

runtime plane. This operational model can free IT operations teams to concentrate on application delivery and governance. See the [CloudHub 2.0 documentation](#) and MuleSoft's [deployment strategies documentation](#).

QUESTION NO: 3

Which component of AnypointPlatform belongs to the platform control plane"?

- A. Runtime Replica
- B. Anypoint Connectors
- C. API Manager
- D. Runtime Fabric

ANSWER: C

Explanation:

API Manager is part of the Anypoint Platform control plane because it provides the centralized management capabilities used to govern APIs across an organization. From API Manager, teams can apply and manage API policies, configure client access, establish service-level agreements, view API analytics, and manage API instances and deployments. These functions are administrative and governance-oriented: they define how APIs should be secured, exposed, monitored, and consumed, rather than executing the API's integration logic itself.

The control plane is the centrally managed layer of Anypoint Platform that supplies tools and services for designing, governing, monitoring, and operating integrations. API Manager is a key service in this layer because it gives administrators and API owners a unified interface for lifecycle and policy management. This centralized approach allows governance rules and access controls to be consistently managed across environments and runtime deployments. MuleSoft's API Manager documentation describes its role in managing APIs and applying policies, while the Anypoint Platform architecture overview distinguishes platform management services from runtime execution infrastructure. See [MuleSoft API Manager documentation](#) and [MuleSoft Runtime Manager documentation](#).

QUESTION NO: 4

What are two reasons why a typical MuleSoft customer favors a MuleSoft-hosted AnypointPlatform runtime plane over a customer-hosted runtime for its Mule application deployments? (Choose two.)

- A. Reduced time-to-market for the first application
- B. Reduced application latency
- C. Reduced IT operations effort
- D. increased application isolation
- E. Increased application throughput

ANSWER: A C

Explanation:

Reduced time-to-market for the first application is a key benefit of a MuleSoft-hosted runtime plane because the hosting environment is already provided and managed as part of the Anypoint Platform service. Teams can deploy a Mule application without first procuring infrastructure, installing and configuring runtime-hosting components, or establishing the operational processes required for a self-managed environment. This enables an integration team to begin delivering APIs and integrations sooner.

Reduced IT operations effort is also a central reason to choose MuleSoft-hosted deployment. With CloudHub, MuleSoft operates the underlying hosting platform and provides managed capabilities for application deployment, monitoring, availability, and platform maintenance. Customer teams retain responsibility for their applications, configurations, and

integrations, but avoid much of the infrastructure administration associated with operating a customer-hosted runtime plane. This lets IT operations and integration teams focus more of their capacity on delivering, governing, and improving integrations rather than maintaining the runtime infrastructure. MuleSoft documents CloudHub as its fully managed integration platform as a service, including its role in deploying and managing Mule applications. See [CloudHub documentation](#) and [CloudHub 2.0 documentation](#).

QUESTION NO: 5

An organization is implementing an employee-onboarding capability using API-led connectivity. The solution must retrieve employee data from an HR system and coordinate account creation in payroll and identity-management systems. A separate mobile application requires a response tailored to its user interface.

Which two API design decisions are appropriate? (Choose two.)

- A. Create a System API that exposes the HR system through a stable interface
- B. Create a Process API that orchestrates the onboarding steps across systems
- C. Allow the mobile application to access the HR system directly
- D. Implement payroll and identity orchestration separately in each Experience API
- E. Use an Experience API as the system-of-record interface for the HR system

ANSWER: A B

Explanation:

A System API should encapsulate access to the HR system and hide its connection details and underlying data model from consumers. A Process API should orchestrate the reusable onboarding logic across the HR, payroll, and identity-management systems.

An Experience API can tailor data for a particular channel, such as the mobile application, but it should not replace the reusable process orchestration. Direct channel access to the HR system would create unnecessary coupling.

QUESTION NO: 6

An organization is exposing product availability from its ERP system for use by a mobile shopping application, a partner portal, and an internal sales application.

The organization wants to prevent the consuming applications from depending directly on the ERP data model and connection details. Which API-led connectivity layer should expose the ERP system in this scenario?

- A. System API
- B. Process API
- C. Experience API
- D. API proxy

ANSWER: A

Explanation:

A **System API** is the appropriate layer because it provides a stable, reusable interface to a system of record such as an ERP system. The System API encapsulates the ERP-specific connectivity, authentication, data model, and technical complexity so that downstream consumers do not need to interact with the ERP directly.

A Process API can combine or apply business logic across one or more System APIs, while an Experience API is tailored to the needs of a particular channel or consumer experience. Neither is the primary boundary for directly unlocking the ERP system.