Lesson 1 - Introducing integration solution architectures

- 1.1 Describe the objectives of enterprise integration solution
- 1.2 Summarize how to architect for success with Anypoint Platform
- 1.3 Describe how integration solutions using Anypoint Platform are documented
- 1.4 Start using an architecture template for the course case study

Lesson 2 - Identifying Anypoint Platform components and capabilities

- 2.1 Identify and document the overall design intentions of Anypoint Platform
- 2.2 Summarize Anypoint Platform capabilities and high-level components
- 2.3 Identify and distinguish between Anypoint Platform infrastructure and deployment options
- 2.4 Choose Anypoint Platform components to be used to meet the functional and non-functional requirements of an integration use case

Lesson 3 - Designing integration solutions using Mule applications

- 3.1 Explain the typical uses of Mule components (including connectors, transformers, routers, and error handlers) in Mule applications
- 3.2 Describe the differences between Mule 4 and Mule 3 applications
- 3.3 Choose appropriate Mule application components to design an integration use case
- 3.4 Summarize how class loading isolation is implemented in Mule runtimes

Lesson 4 - Choosing appropriate Mule 4 event processing models

- 4.1 Distinguish between Mule 4 blocking, non-blocking, parallel, and reactive event processing options
- 4.2 Identify the event processing models used in various Mule 4 scopes and components
- 4.3 Describe Mule 4 streaming options and behaviors
- 4.4 Describe the event processing options for JMS and VM connectors
- 4.5 Select appropriate event processing for an integration use case
- 4.6 Design batch, near real-time, and real-time data synchronization integration use cases

Lesson 5 - Choosing appropriate message transformation and routing patterns

- 5.1 Describe reusable ways to transform and process events
- 5.2 Explain how to simplify and decouple complex data mappings using common data models
- 5.3 Design transformations between data models
- 5.4 Choose the best event transformation, data validation, and event routing patterns to an integration use case

Lesson 6 - Designing Mule application testing strategies

- 6.1 Describe possible testing strategies for Mule applications

- 6.2 Document a testing strategy for an integration use case
- 6.3 Define the types of MUnit tests required for an integration use case and document the code coverage
- 6.4 Design integration and performance tests for Mule applications

Lesson 7 - Choosing and developing a deployment strategy

- • 7.1 Distinguish between various Anypoint Platform runtime service models
- 7.2 Distinguish between various Anypoint Platform deployment models
- 7.3 Select the best deployment and runtime service options for an integration use case

Lesson 8 - Designing with appropriate state preservation and management options

- • 8.1 Select the best deployment and runtime service options for an integration use case
- 8.2 Explain how to store Mule application state using Object Store v2
- 8.3 Decide when to manage storage of Mule application state using persistent VM queues
- 8.4 Decide when to use Mule application provided caches to store Mule application state
- 8.5 Design an integration solution to avoid duplicate processing of previous records using Mule connector watermarks
- 8.6 Design the best storage and state management options for an integration use case

Lesson 9 - Designing effective logging and monitoring

- • 9.1 Describe auditing and logging options for Mule applications
- 9.2 Design logging strategies for Mule applications
- 9.3 Choose logging policies for Mule application log files
- 9.4 Describe integration options with third-party log management systems
- 9.5 Specify Anypoint Platform monitoring, alerting, notification, visualization, and reporting options for APIs and integration solutions
- 9.6 Choose the best monitoring, alerting, and notification options for an integration use case

Lesson 10 - Designing an efficient and automated software development lifecycle

- • 10.1 Design to manage property files for Mule applications across different environments
- 10.2 Design to manage Anypoint Platform environments for Mule application deployments
- 10.3 Describe how to implement continuous integration and continuous delivery (CI/CD) for an organization
- 10.4 Describe how to automate deployment and management with Anypoint Platform

Lesson 11 - Designing transaction management in Mule applications

- • 11.1 Identify why and when transactions are supported in Mule applications
- 11.2 Identify resources that participate in transactions in Mule applications

- 11.3 Describe how to manage a transaction using a transaction manager or the Saga pattern
- 11.4 Describe how to demarcate transaction boundaries in Mule applications
- 11.5 Choose the correct transaction type based on the participating resources

Lesson 12 - Designing for reliability goals

- • 12.1 Identify and distinguish between reliability patterns for Mule application and their deployments
- 12.2 Design to effectively balance competing non-functional requirements
- 12.3 Clarify and validate reliability goals for an integration use case
- 12.4 Design Mule applications and their deployments to meet reliability goals
- 12.5 Design to effectively balance reliability goals with other project goals and requirements

Lesson 13 - Designing for high availability goals

- • 13.1 Identify various types of high availability (HA) goals for Mule applications
- 13.2 Identify ways to achieve HA in CloudHub and on-premises deployments
- 13.3 Identify HA aware connectors and their design tradeoffs
- 13.4 Describe how clustering and load balancing work in CloudHub and on-premises deployments
- 13.5 Design to effectively balance HA goals with other project goals and requirements

Lesson 14 - Optimizing the performance of deployed Mule applications

- • 14.1 Clarify performance goals for Mule applications
- 14.2 Identify the need for performance optimization and associated tradeoffs
- 14.3 Describe ways to search for and locate performance bottlenecks
- 14.4 Describe how to design, architect, design, and implement for performance
- 14.5 Describe ways to measure performance
- 14.6 Describe methods and best practices to performance tune Mule applications and Mule runtimes
- 14.7 Design to effectively balance performance goals with reliability and HA goals

Lesson 15 - Designing secure Mule applications and deployments

- • 15.1 Describe Anypoint Platform security concepts and options
- 15.2 Explain how to secure APIs on Anypoint Platform
- 15.3 Differentiate between MuleSoft and customer responsibilities related to Anypoint Platform security
- 15.4 Evaluate security risks for Mule applications
- 15.5 Describe how to secure Mule application properties and data in transit

Lesson 16 - Designing secure network communications between Mule applications

- 16.1 Describe Anypoint Platform network security options and architectures
- 16.2 Identify MuleSoft-owned and customer-owned roles and responsibilities related to Anypoint Platform network security
- 16.3 Describe how to secure Mule applications using Java key stores
- 16.4 Design TLS communication and other network security options for an integration use case
- 16.5 Properly size an Anypoint VPC to support deployment of all expected Mule applications