Introduction to Messaging and RabbitMQ (Duration: 2 hours)

* Understanding the fundamentals of messaging and its role in distributed systems, including decoupling components and enabling asynchronous communication.
* Introduction to RabbitMQ, an open-source message broker that implements the Advanced Message Queuing Protocol (AMQP).
* Exploring key features of RabbitMQ, such as message queuing, routing, and delivery guarantees.
* Installation and setup of RabbitMQ on different platforms, including Windows, macOS, and Linux.
* Overview of the RabbitMQ architecture, including exchanges, queues, and bindings.
* Exploring various messaging patterns and use cases where RabbitMQ can be utilized, such as task queues, event-driven systems, and publish-subscribe.

RabbitMQ Basics (Duration: 6 hours)

* Working with RabbitMQ exchanges, which act as message routing agents.
* Understanding different exchange types: direct, topic, headers, and fanout, and their use cases.
* Creating and managing RabbitMQ queues to store and deliver messages.
* Establishing bindings between exchanges and queues to define message routing rules.
* Implementing the publish-subscribe pattern in RabbitMQ, allowing multiple consumers to receive messages from a single producer.
* Direct and topic-based message routing in RabbitMQ, based on routing keys and patterns.
* Implementing message acknowledgment to ensure reliable message processing.
* Handling failure scenarios, such as message redelivery and dead-letter exchanges.

Part 3: Advanced RabbitMQ Concepts (Duration: 8 hours)

* Understanding message durability and persistence in RabbitMQ, ensuring message persistence even in case of server restarts or failures.
* Exploring additional exchange types, such as headers and fanout exchanges, and their use cases in message distribution.
* Configuring dead letter exchanges and implementing message retry mechanisms to handle failed message processing.
* Implementing message routing based on headers and routing keys, allowing for more complex message routing scenarios.

Exploring RabbitMQ plugins and extensions to extend RabbitMQ's functionality, such as the management plugin for monitoring and administration.

* Implementing message compression and encryption techniques to ensure data security and optimize message transmission.
* Handling RabbitMQ message flow control and managing backpressure to prevent overwhelming consumers or producers.

RabbitMQ in Real-world Applications (Duration: 12 hours)

* Implementing RabbitMQ in task queue scenarios for distributing and processing background jobs.
* Building event-driven systems with RabbitMQ to enable communication and coordination between different services or microservices.
* Integrating RabbitMQ with popular frameworks and libraries, such as Spring Boot, .NET, or Node.js.
* Implementing RabbitMQ for reliable message-based communication in distributed systems.
* Designing scalable and fault-tolerant architectures using RabbitMQ clusters and high availability configurations.
* Optimizing RabbitMQ performance and scalability through techniques like message batching and connection pooling.
* Monitoring and troubleshooting RabbitMQ deployments, including monitoring message throughput, queue lengths, and consumer lag.
* Implementing RabbitMQ security measures, such as authentication, authorization, and SSL/TLS encryption.