Create a new Spring Boot project using Spring Initializr.

**Add necessary dependencies:**

Spring Web, Spring Security, Eureka Server, Spring Cloud Gateway, Spring Cloud Config Server, Spring Data JPA, etc.

**Flight Reservation Microservice:**

Develop a microservice to manage flight reservations.

Define REST endpoints for creating, updating, and fetching reservations.

Implement data persistence using Spring Data JPA and a database (e.g., MySQL).

**User Authentication with Spring Security:**

Configure Spring Security for user authentication and authorization.

Implement user registration and login functionality.

Secure the REST endpoints based on user roles.

Eureka Server for Service Discovery:

Create a Eureka Server to act as a service registry.

Register the Flight Reservation microservice with the Eureka Server.

Enable other microservices to discover and communicate with each other.

**API Gateway (Spring Cloud Gateway):**

Develop an API Gateway to handle incoming requests and route them to appropriate microservices.

Implement load balancing and routing rules for different services.

Spring Cloud Config Server:

Set up a Spring Cloud Config Server to manage externalized configuration.

Store configuration files in a Git repository and configure the Config Server to fetch configurations.

Microservices Communication:

Implement communication between microservices using REST APIs or messaging (e.g., RabbitMQ or Apache Kafka).

Handle service-to-service communication securely and efficiently.

**Testing and Validation:**

Write unit tests and integration tests for each microservice.

Test the entire system to ensure proper communication and functionality.

**Security and Authorization:**

Implement OAuth2-based token authentication for securing microservices.

Define roles and permissions to control access to resources.

**Configuring the Config Server:**

Configure the Spring Cloud Config Server to fetch configuration files from the Git repository.

Store environment-specific configurations.

**Documentation and Testing:**

Create documentation for API endpoints, configurations, and deployment instructions.

Test the application thoroughly to ensure all components work seamlessly.

Continuous Integration and Deployment (CI/CD):

Set up a CI/CD pipeline to automate building, testing, and deploying microservices.