* First Task
  + I have to build a Monolithic Application that consumes Course Catalog, Course, User, and a database.
  + This needs to be done using the SQLite3.

**Monolithic Application**

Course Catalog

Database

Course

User

**Microservice Architecture**

Course Catalog App Service

Course App Service

Rest

**Microservice 2**

**Microservice 1**

Rest

User App Service

**Microservice 3**



* Integrating services with RestTemplate

Course Catalog Service

Course Application Service

* With Spring Cloud Netflix, you can implement Service Discovery!
  + 2 Main Projects:

Spring Cloud Netflix Eureka Server

Spring Cloud Netflix Eureka Client

* + Concept:

**Discovers Service 2**

Discovery Server

Service 1

Registers

Service 2

* + Discover Server is an actively managed registry for service location.
  + Discovery Server monitors deployed services.

This is how service discover works in the microservices world.

Discovers Course App

Discovery Server

Course Catalog

Register

Course App

* Handling failure with Circuit Breaker Pattern
  + We can use the Circuit Breaker Pattern to handle this problem.
  + Spring Could and Netflix Hystrix

If more than 50% of the request fail in a 10 second rolling window, then break the circuit.

Service 2

Service 1

Allow a request every 5 seconds to check if the service is up.

Adding the UserApp Service

Course Catalog Service

Rest

Course App Service

Microservice 1

Microservice 2

Rest

Fetch Course Informaiton from CourseApp and corresponding User information from UserApp and display in the CourseCatalog Service.

User App Service

Microservice 3

Adding the UserApp Service

Course Catalog Service

Rest

Course App Service

Microservice 1

Microservice 2

Rest

Fetch Course Informaiton from CourseApp and corresponding User information from UserApp and display in the CourseCatalog Service.

User App Service

Microservice 3