# Microservizi

## Pattern “database per service”

## “Backing Service”

### INTRO

### SPRING PROFILE

application.properties

1. localH2
2. mySqlLocal
3. mySqlDocker
4. dockerContainer (specializzare la connessione jdbc secondo lo standard docker run)
5. cloudFoundry

**DEMO**

1. **Esecuzione localH2 (in memory - local Console)**
2. **Esecuzione mySqlLocal -/Env**
3. **Esecuzione mySqlDocker -/Env**
4. **BUILD maven docker local published -/Env**
5. **Esecuzione dockerContainer**

### BINDING BACKING SERVICES IN CLOUD FOUNDRY

manifest.yml

cf create-service ClearDb <plan> <name>

env:

SPRING\_PROFILE\_ACTIVE: <profile\_name>

**DEMO**

1. **Esecuzione cloudFoundry /Env**

### PAAS CONNECTOR

Definizione datasource

@Configuration

@Profile(“cloudFoundry”)

Mettere log print sulla classe che definisce la configuration

**DEMO**

1. **Esecuzione cloudFoundry /Env**

### JAVA CONFIGURATION

Definizione datasource

@Bean

@Profile(“cloudFoundry”)

Definizione di un custom connetion pool

## “Service discovery and registration”

### EUREKA SERVICE REGISTRY

Cosa è Eureka

Implementazione applicazione

Console

**DEMO**

1. **Esecuzione EUREKA**
2. **Console EUREKA**

### SERVER APPLICATION REGISTRATION IN EUREKA

Implementazione applicazione

Maven di deployment

Dettagli degli elementi di registrazione sul file yml

Console di registrazione

**DEMO**

1. **Deployment su Cloud Foundry**
2. **Console EUREKA**

### CLIENT APPLICATION CONSUMING SERVER APPLICATION ON EUREKA

Implementazione applicazione

Dettagli di bind verso il server sul codice

Dettaglio sul file yml

Console Eureka