cloud-config-server

```
    <properties>
        <java.version>11</java.version>
        <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-config-server</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
        </dependency>
    </dependencies>
```

```
server:
  port: 9296

spring:
  application:
    name: CONFIG-SERVER
  cloud:
    config:
      server:
        git:
          uri: https://github.com/shabbirdwd53/config-server
          clone-on-start: true
```

```
package com.dailycodebuffer.cloud;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.config.server.EnableConfigServer;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication
@EnableEurekaClient
@EnableConfigServer
public class CloudConfigServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(CloudConfigServerApplication.class, args);
    }
}
```

cloud-gateway

```
<properties>
        <java.version>11</java.version>
        <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-gateway</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-hystrix</artifactId>
        </dependency>
    </dependencies>
```

```
spring:
  cloud:
    config:
      enabled: true
      uri: http://localhost:9296
```

```
server:
  port: 9191

spring:
  application:
    name: API-GATEWAY
  cloud:
    gateway:
      routes:
        - id: USER-SERVICE
          uri: lb://USER-SERVICE
          predicates:
            - Path=/users/**
          filters:
            - name: CircuitBreaker
              args:
                name: USER-SERVICE
                fallbackuri: forward:/userServiceFallBack
        - id: DEPARTMENT-SERVICE
          uri: lb://DEPARTMENT-SERVICE
          predicates:
            - Path=/departments/**
          filters:
            - name: CircuitBreaker
              args:
                name: DEPARTMENT-SERVICE
                fallbackuri: forward:/departmentServiceFallBack


hystrix:
  command:
    fallbackcmd:
      execution:
        isolation:
          thread:
            timeoutInMilliseconds: 4000


management:
  endpoints:
    web:
      exposure:
        include: hystrix.stream
```

```
package com.dailycodebuffer.cloud.gateway;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.netflix.hystrix.EnableHystrix;

@SpringBootApplication
@EnableEurekaClient
@EnableHystrix
public class CloudGatewayApplication {

    public static void main(String[] args) {
        SpringApplication.run(CloudGatewayApplication.class, args);
    }

}
```

```

package com.dailycodebuffer.cloud.gateway;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
public class FallBackMethodController {

    @GetMapping("/userServiceFallBack")
    public String userServiceFallBackMethod() {
        return "User Service is taking longer than Expected." +
                " Please try again later";
    }

    @GetMapping("/departmentServiceFallBack")
    public String departmentServiceFallBackMethod() {
        return "Department Service is taking longer than Expected." +
                " Please try again later";
    }
}
```

department-service

```
<properties>
    <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
</properties>

<artifactId>spring-boot-starter-web</artifactId>
<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
<artifactId>spring-cloud-starter-config</artifactId>
<artifactId>spring-cloud-starter-sleuth</artifactId>
<artifactId>spring-cloud-starter-zipkin</artifactId>
```

```
server:
  port: 9001

spring:
  application:
    name: DEPARTMENT-SERVICE
  zipkin:
    base-url: http://127.0.0.1:9411/
```

```
spring:
  cloud:
    config:
      enabled: true
      uri: http://localhost:9296
```

```
@SpringBootApplication
@EnableEurekaClient
public class DepartmentServiceApplication {
    // Code
}
```

user-service

```
server:
  port: 9002

spring:
  application:
    name: USER-SERVICE
  zipkin:
    base-url: http://127.0.0.1:9411/
```

```
spring:
  cloud:
    config:
      enabled: true
      uri: http://localhost:9296
```

```
package com.dailycodebuffer.user;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.context.annotation.Bean;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
@EnableEurekaClient
public class UserServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(UserServiceApplication.class, args);
    }

    @Bean
    @LoadBalanced
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

```
@Autowired
private RestTemplate restTemplate;

Department department = restTemplate.getForObject(
    "http://DEPARTMENT-SERVICE/depts/"+ user.getDeptId(), Department.class);
```

service-registry

```
<properties>
    <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
</properties>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
```

```
@SpringBootApplication
@EnableEurekaServer
public class ServiceRegistryApplication {
    // ...
}
```

```
server:
  port: 8761

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
```

hystrix-dashboard

```
<properties>
    <spring-cloud.version>Hoxton.SR8</spring-cloud.version>
</properties>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
<artifactId>spring-cloud-starter-netflix-hystrix-dashboard</artifactId>
<artifactId>spring-cloud-starter-config</artifactId>
```

```
@SpringBootApplication
@EnableHystrixDashboard
@EnableEurekaClient
public class HystrixDashboardApplication {
    // ...
}
```

```
server:
  port: 9295

spring:
  application:
    name: HYSTRIX-DASHBOARD

hystrix:
  dashboard:
    proxy-stream-allow-list: "*"
```