



SETUP & BOM

```
<dep>spring-cloud-dependencies BOM</dep>
spring-cloud-starter-config
spring-cloud-starter-netflix-eureka-client
spring-cloud-starter-gateway
Match Spring Cloud release train to
Spring Boot version
```

CONFIG SERVER

```
@EnableConfigServer
@SpringBootApplication
spring.cloud.config.server.git.uri=https://...
spring.cloud.config.server.git.search-paths=config
GET /{application}/{profile} returns
merged config
Config stored in git repo as
{app}-{profile}.yml
```

CONFIG CLIENT

```
spring.config.import=optional:configserver:
spring.cloud.config.uri=http://config-server:8888
spring.cloud.config.fail-fast=true
@RefreshScope on beans to reload on
/actuator/refresh
spring.cloud.bus (Kafka/RabbitMQ) for
broadcast refresh
```

SERVICE DISCOVERY (EUREKA)

```
Server: @EnableEurekaServer
eureka.instance.hostname=eureka-server
Client: @EnableEurekaClient (or auto
if dep present)
eureka.client.serviceUrl.defaultZone=http://...
spring.application.name=my-service (=
service ID)
Clients fetch registry every 30s by
default
```

API GATEWAY

```
spring.cloud.gateway.routes:
- id: user-service
uri: lb://USER-SERVICE
predicates:
- Path=/api/users/**
filters:
- StripPrefix=1
lb:// = client-side load balancing
```

CIRCUIT BREAKER (RESILIENCE4J)

```
@CircuitBreaker(name="userSvc",
fallbackMethod="fallback")
@Retry(name="userSvc")
@TimeLimiter(name="userSvc")
resilience4j.circuitbreaker.instances.userSvc
.slidingWindowSize=10
.failureRateThreshold=50
```

FEIGN CLIENT

```
@EnableFeignClients
@FeignClient(name="user-service")
interface UserClient {
@GetMapping("/users/{id}")
User getUser(@PathVariable Long id);
}
Integrates with Eureka and
Resilience4j
```

LOAD BALANCING

```
<dep>spring-cloud-starter-loadbalancer</dep>
@LoadBalanced RestTemplate template
template.getForObject("http://USER-SERVICE/",
User.class)
WebClient.Builder with @LoadBalanced
Round-robin by default; pluggable
strategy
```

DISTRIBUTED TRACING

```
<dep>spring-cloud-starter-sleuth</dep>
<dep>zipkin-brave-spring-cloud-sleuth</dep>
MDC populated with traceId, spanId
spring.zipkin.base-url=http://zipkin:9411
spring.sleuth.sampler.probability=1.0
```

SPRING CLOUD BUS

```
<dep>spring-cloud-starter-bus-kafka</dep>
POST /actuator/busrefresh triggers all
clients
Propagates config refresh via
message broker
spring.cloud.bus.enabled=true
Requires Kafka or RabbitMQ broker
```

RETRY & RATE LIMITING

```
@Retry(name="user", fallbackMethod="fb")
resilience4j.retry.instances.user
.maxAttempts=3
.waitDuration=1s
Rate limiter: RequestRateLimiter
filter in Gateway
redis-rate-limiter.replenishRate=10
```

COMMON GOTCHAS

Release train version mismatch
causes NoSuchMethod errors
Feign client needs
@EnableFeignClients on main class
Config Server needs to start before
clients (health check)
Ribbon deprecated use
spring-cloud-starter-loadbalancer
Sleuth auto-adds trace headers;
verify sampling rate in prod