



Spring Cloud Config provides server-side and client-side support for externalized configuration in a distributed system.

Spring Cloud - Config

(Configuring Services Using Distributed Configuration)

Topics

- Configuration in a Distributed System
- Configuration with spring Cloud Config
 - Using Config client and server
 - Backend stores (Git, SVN Repository)
 - Updating Configuration & @RefreshScope
 - Spring cloud actuator
 - Spring cloud bus
 - Spring cloud monitor
 - Securing Config Server
 - Storing and retrieving sensitive Configuration

What is So different about managing configuration in a cloud-native application?

Spring Cloud Config by Pratap Kumar

Configuration Management

Configuration: Non-distributed vs Distributed



From one or a handful
of configuration files



To ...



Many, many
configuration files

Spring Cloud Config by Pratap Kumar

First Level Solution

- configuration management tooling to the rescue, right?
 - e.g. chef / puppet / Ansible
- it will work , but it is not ideal for the cloud

Configuration Management

Issues with Typical Configuration Management



Deployment-oriented



Push-based is usually
not dynamic enough



Pull-based adds latency
with temporal polling

Spring Cloud Config by Pratap Kumar

A True Solution

Spring Cloud Config by Pratap Kumar

Q: If configuration management tooling doesn't solve our problem, what does?

A: Configuration Server

Configuration Server

- **Application Configuration Server**

- Dedicated, dynamic, centralized, key/value store (may be distributed)
- Authoritative source
- Auditing
- Versioning
- Cryptography support

Tools

- **Managing Application Configuration with Spring Cloud**
- Manage Config with
 - Spring Cloud Consul
 - Spring Cloud Zookeeper
 - Spring Cloud Config
 - Config-Client
 - Config-Server
- **Spring Cloud Config**
 - Spring Cloud Config provides server and client-side support for externalized configuration in a distributed system

Spring Cloud Config by Pratap Kumar

Configuration Management

Integration with Spring Applications

Config Client



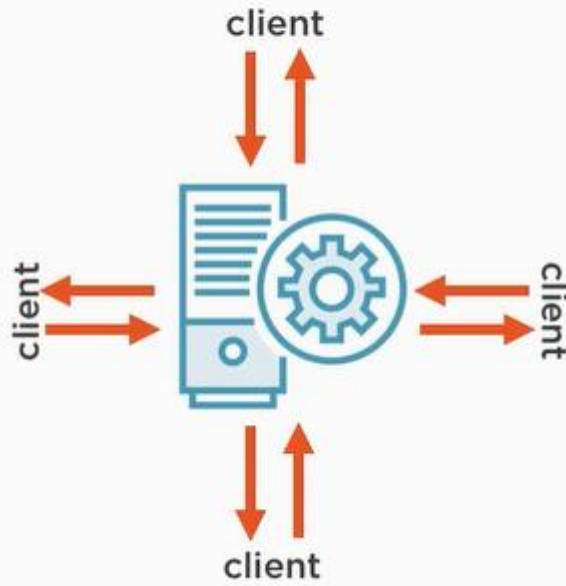
- Embedded in application
- Spring Environment abstraction
 - e.g. @Inject Environment

Config Server



- Standalone (can be embedded)
- Spring PropertySource abstraction
 - e.g. classpath:file.properties

Config Server



Spring Cloud Config Server

- Is another web application
- HTTP REST access
- No Write Support
- Supports various Output formats
 - JSON (default)
 - Properties
 - YAML
- Backend stores
 - Git (default)
 - SVN
 - File System
 - <https://www.logicbig.com/tutorials/spring-framework/spring-cloud/cloud-config-with-file-system-backend.html>
- Configuration Scopes
 - Global Configuration that applies to all applications
 - Application specific configuration
 - Spring Profile specific configuration

Spring Cloud Config by Pratap

Config-server example

Spring Cloud Config by Pratap Kumar

- Pom.xml
- Actuator
- Devtols
- web
- <dependency>
 - <groupId>org.springframework.cloud</groupId>
 - <artifactId>spring-cloud-config-server</artifactId>
- </dependency>
- Eureka discovery (Optional)

Spring Cloud Config Server

Spring Cloud Config by Pratap Kumar

Using Spring Cloud Config Server

Create a folder to store
configuration

(optional) Add a
properties or yaml file
with a named
application

Add properties or yaml
files named
{application}-{profile}

```
git init
```

```
git add  
git commit
```

(optional) Setup remote
git repository and
git push

Spring Cloud Config Server

Application.java

```
@SpringBootApplication
@EnableConfigServer ←
public class Application {

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

Tip: add eureka client dependencies, service-url configuration, and @EnableDiscoveryClient to make the config server discoverable!

Spring Cloud Config by Pratap Kumar

Using Spring Cloud Config Server

application.properties

```
server.port=8888
spring.cloud.config.server.git.uri=<uri_to_git_repo>
```

Spring Cloud Config Server : REST Endpoints

Spring Cloud Config by Pratap Kumar

REST Endpoint Parameters

`{application}`

maps to
`spring.application.name`
on client

`{profile}`

maps to
`spring.profiles.active`
on client

`{label}`

server side feature to
refer to set of config
files by name

Spring Cloud Config Server : REST Endpoints

Spring Cloud Config by Pratap Kumar

REST Endpoints



Endpoint

GET `/ {application} / {profile} [/ {label}]`



Example

- `/myapp/dev/master`
- `/myapp/prod/v2`
- `/myapp/default`

Spring Cloud Config Server : REST Endpoints

Spring Cloud Config by Pratap Kumar

REST Endpoints

Format



GET `/ {application}-{profile}.(yaml | properties)`



Example

- `/myapp-dev.yaml`
- `/myapp-prod.properties`
- `/myapp-default.properties`

Spring Cloud Config Server : REST Endpoints

Spring Cloud Config by Pratap Kumar

REST Endpoints



Endpoint

GET `/ {label} / {application} - {profile} . (yaml | properties)`



Example

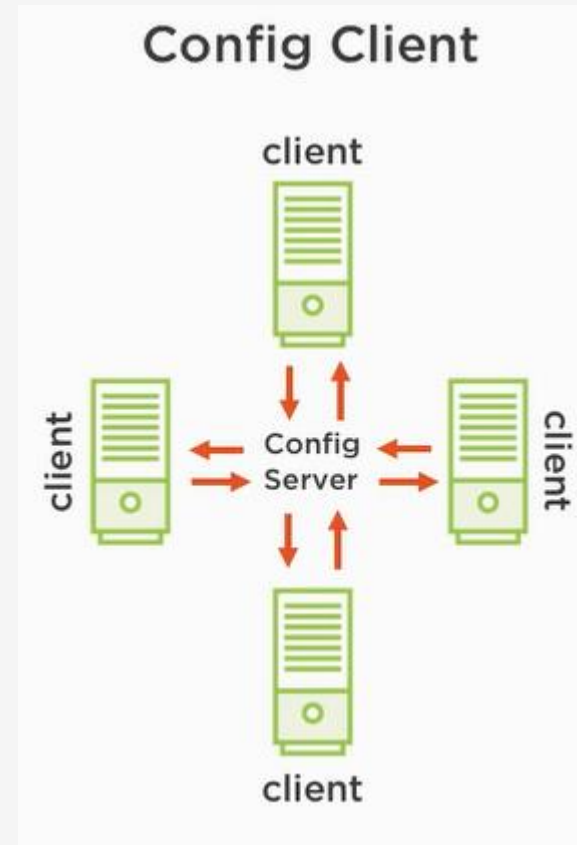
- `/master/myapp-dev.yaml`
- `/v2/myapp-prod.properties`
- `/master/myapp-default.properties`

Spring Cloud Config Client

Spring Cloud Config by Pratap Kumar

Spring Cloud Config Client

Spring Cloud Config by Pratap Kumar



Bootstrap & Fetch App Configuration

Spring Cloud Config Client

Fetching Configuration: Application Startup



Spring Cloud Config by Pratap Kumar

Spring Cloud Config Client

Bootstrapping with
`bootstrap.properties` or `bootstrap.yml`



Config first

Specify the location of the config server



Discovery first

Discover the location of the config server

Using Spring Cloud Config Client

Spring Cloud Config by Pratap Kumar

Using Spring Cloud Config Client

pom.xml

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

Using Spring Cloud Config Client

Using Spring Cloud Config Client: Config First

bootstrap.properties

```
spring.application.name=<your_app_name>  
spring.cloud.config.uri=http://localhost:8888/
```

OR

bootstrap.yml

```
spring:  
  application:  
    name: <your_app_name>  
  cloud:  
    config:  
      uri: http://localhost:8888/
```

Spring Cloud Config by Pratap Kumar

Using Spring Cloud Config Client

Spring Cloud Config by Pratap Kumar

Using Spring Cloud Config Client: Discovery First

bootstrap.properties

```
spring.application.name=<your_app_name>  
spring.cloud.config.discovery.enabled=true
```

OR

bootstrap.yml

```
spring:  
  application:  
    name: <your_app_name>  
  cloud:  
    discovery:  
      enabled: true
```

** Note: don't forget to add eureka client dependencies, service-url configuration, and @EnableDiscoveryClient*

spring.cloud.config.discovery.service-id=config-server

Config-client-app example

- Pom.xml
- Config-client
- Eureka-Discovery
- Actuator
- Devtools
- web

Spring Cloud Config by Pratap Kumar

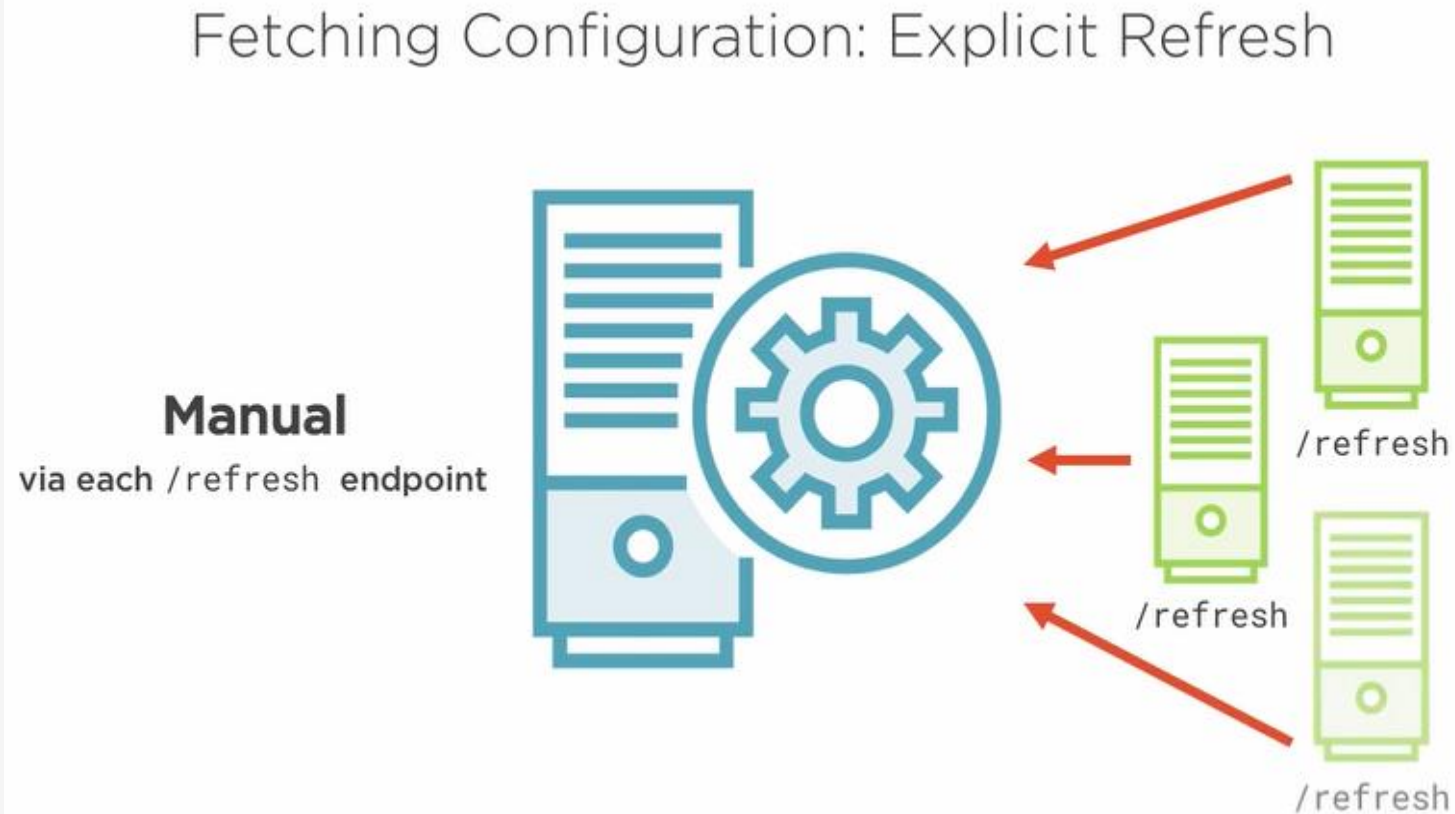
Updating Configuration at Runtime

- With this you can Refresh your @ConfigurationProperties at Runtime.
- Update the Logging Level
- Without Restarting the Application
- **Step 1 (Configuration changes)**
 - Edit & Save Configuration file(s)
 - Commit and / or Push Changes to VCS
 - Git add .
 - Git commit -m "made some configuration change"
 - Git push origin head

Spring Cloud Config by Pratap Kumar

Updating Configuration at Runtime

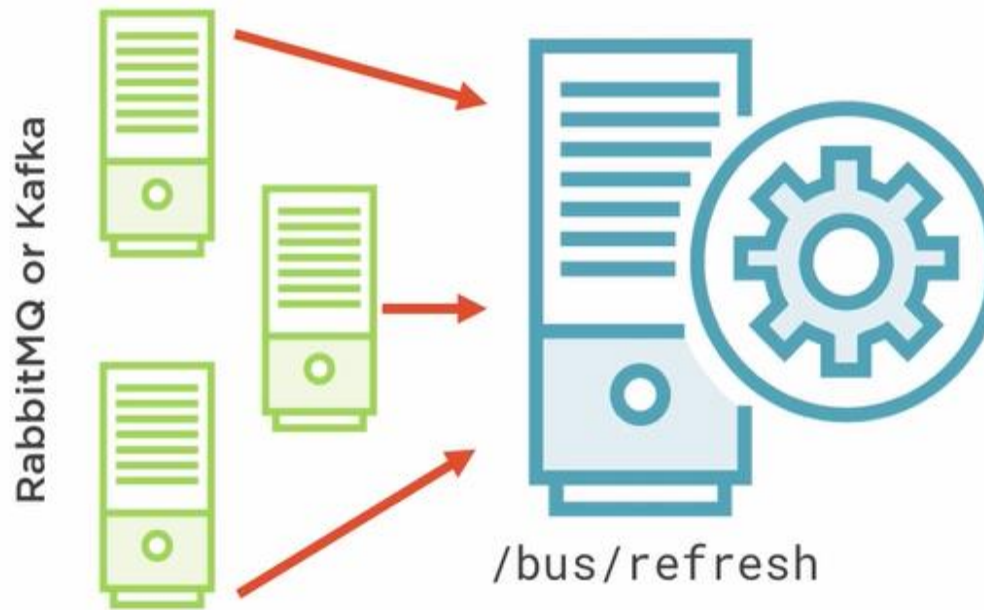
- **Step 2 : Notify Application(s) to Refresh Configuration**
 - Manually using /refresh endpoint of spring boot actuator



Updating Configuration at Runtime

- **Step 2 : Notify Application(s) to Refresh Configuration**
 - Both Manual and Automated using `/bus /refresh` with spring cloud Bus

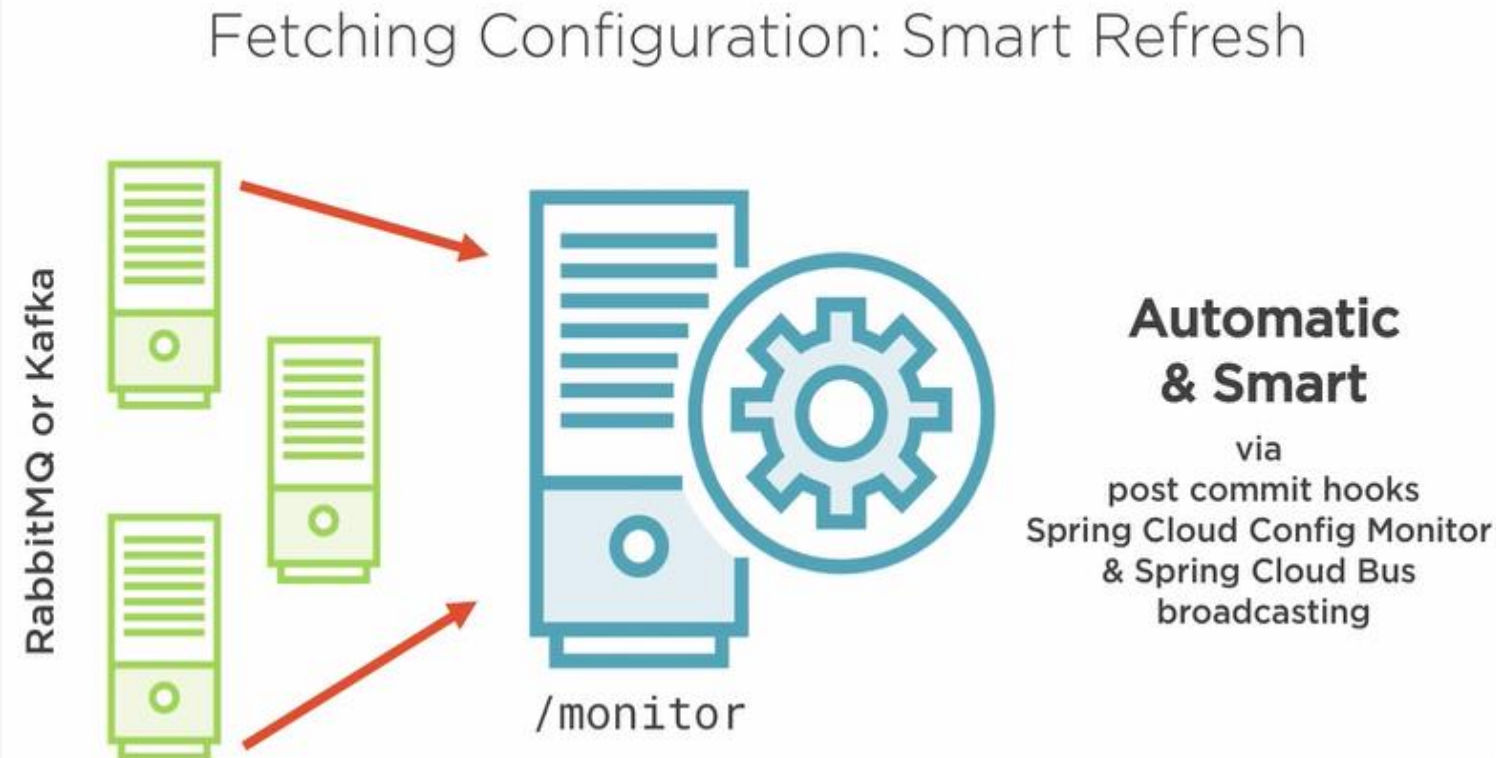
Fetching Configuration: Dynamic Push Refresh



Automatic
via Spring Cloud Bus
broadcasting

Updating Configuration at Runtime

- **Step 2 : Notify Application(s) to Refresh Configuration**
 - VCS + /monitor with spring-cloud-config-monitor and Spring-cloud-bus



Spring Cloud Config by Pratap Kumar

Updating Configuration at Runtime

Spring Cloud Config by Pratap Kumar

- **Step 3: Celebrate**
 - Making Configuration updates on the fly Without Restarting.
 - Updating all your apps at once or Automatically
 - Audit log of all your changes

Refreshing Configuration

Spring Cloud Config by Pratap Kumar

- **What is Covered and What's not ?**
- **Covered**
 - @ConfigurationProperties
 - All logging levels defined by logging.level.* are updated
- **Not Covered**
 - Any @Bean or @Value that only gets its configuration upon Initialization

Refreshing Configuration

```
@Configuration
public class SomeConfiguration
{
    @Bean
    public FooService fooService(FooProperties properties) {
        return new FooService(properties.getConfigValue());
    }
}
```

Example: @Bean Will Not See New Config Value After a Refresh

1. Configuration updates are made
 - Note that FooProperties is a @ConfigurationProperties class
2. POST to /refresh
3. Result: FooService will still contain the **OLD** configuration value
 - Only gets configuration during initialization

Refreshing Configuration

```
@Configuration
public class SomeConfiguration
{
    @Value("${some.config.value}")
    String configValue;

    @Bean
    public FooService fooService() {
        return new FooService(configValue);
    }
}
```

Example: @Value Will Not See New Config Value After a Refresh

1. Configuration updates are made
2. POST to /refresh
3. Result: FooService will still contain the **OLD** configuration value
 - Only gets configuration during initialization

Refreshing Configuration

Spring Cloud Config by Pratap Kumar

Q: How do I refresh a `@Bean` or `@Value` that only gets its configuration during initialization?

A: `@RefreshScope`

Refreshing Configuration

```
@Configuration
public class SomeConfiguration
{
    @Bean
    @RefreshScope
    public FooService fooService(FooProperties properties) {
        return new FooService(properties.getConfigValue());
    }
}
```

Example: Utilizing @RefreshScope

1. Add the @RefreshScope annotation to the @Bean
2. POST to /refresh
3. Result: FooService will now contain the **NEW** configuration value!
 - @RefreshScope tells Spring to please reinitialize this @Bean

Encrypting & Decrypting Configuration

Spring Cloud Config by Pratap Kumar

Res

- https://www.youtube.com/watch?v=tVRzlh_73ws

Spring Cloud Config by Pratap Kumar
