

Add instructor notes  
here.

## **Environment Management with Centralized Configuration**



## Objective

Role of Configuration in microservices

Spring cloud config

Creating a configuration server

Consuming configurations in apps

## Role of Configuration in microservices

1. Removing "Settings" from compiled code
2. Change runtime behaviours
3. Enforce consistency across elastic services
4. Cache values to reduce load on database

## Spring Cloud Config

HTTP access to git or file based configuration .

Steps to create Config Server :-

1. Choose a config source
2. Add config files to the particular source
3. Build the Spring Project
4. Secure the configurations

## Creating Configuration Server –Choosing Source

### Local Files

1. Points to classpath or file system
2. Multiple search locations possible
3. No Audit trail
4. Support labelling
5. Support for placeholders in URI
6. Relies on "native" profile
7. Dev/Test only ,unless set up in reliable shared fashion

### Git based Repository

1. Points to Git repository
2. Multiple search locations possible
3. Full Change history
4. Supports labelling
5. Support for placeholders for URI
6. Multiple profiles possible
7. Local git repo for dev/test highly available file system or service for production

## Creating Configuration Server - Setting up Configuration files

Native Support for YAML, properties files

Can serve out any text file

File name contains application name, optionally profile

Nested folder structure supported

All matching files returned

## Creating Config Server-The Spring Project

1. Use start.spring.io, Spring Tool Suite or chosen IDE to generate scaffolding

2. See POM dependency on spring-cloud-config-server and spring-boot-starter-actuator

3. Add @EnableConfigServer annotation to the class

4. Create application properties (OR YAML) with server port, app name and profile.

## Demo-Creating Configuration Server

Demo



## Consuming Configurations in Apps

Spring Apps use Config servers as property source

Loads values based on app name, Spring profile, label.

Annotate code with @Value attribute

Can consume from non-Spring apps via URL.

Demo