



Pivotal®

Spring Cloud Bus: Distributed Config Updates

Background

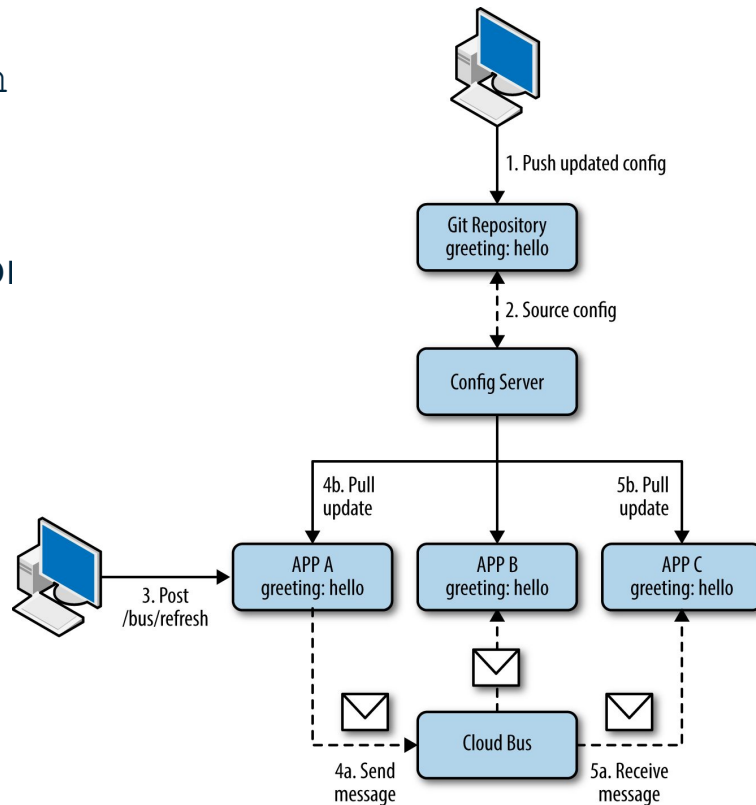
- Spring Cloud Configuration client relies on a pull-based mechanism that is invoked through an Actuator refresh message
- Manually invoking for all instances does not scale
- Need a way to provide for distributed application instance updates

Spring Cloud Bus

- Multiple Instances need to be refreshed
- Disposable instances, cannot track instances manually
- Using `/actuator/refresh` require automated solution :
 - To track live instances
 - Invoke all instances on an app config refresh event
- Solution:
 - Leverages a message bus
 - Leverage general `/actuator/bus-refresh` endpoint
 - Receiving instance broadcasts refresh message to all other instances
 - Use filters to isolate refreshes to specific apps and/or instances

Spring Cloud Config Server + Spring Cloud Bus

- The `/actuator/bus-refresh` endpoint provides a destination parameter to support specifying which applications and application instances to target



Spring Cloud Config and Distributed Config Benefits

- Separate rate of change of development, vs configuration
 - Versioning
 - Staged configuration changes
- Distributed configuration scales

Spring Cloud Config and Distributed Config Trade-Offs

- Possibility of configuration drift during scale up
- Race conditions
- Development should consider design impacts
- Operations should consider windows for config updates