61. What is externalized configuration in microservices?

Externalized configuration refers to storing config values like DB URLs, ports, or feature flags outside the app. This allows changes without redeploying services.

62. What is the benefit of using Spring Cloud Config for configuration management?

Centralizes all service configurations, supports Git versioning, dynamic updates with @RefreshScope, and integrates with Vault for secure secret storage.

63. What is @RefreshScope in Spring Cloud?

It allows a Spring bean to be refreshed at runtime. When configuration is changed and /actuator/refresh is called, beans marked with @RefreshScope are reloaded.

64. How does Spring Cloud Config Client fetch its configuration?

1. The client contacts the config server.
2. The server fetches from Git/FS/Vault.
3. The client loads the configuration at startup.

65. How do you secure secrets in Spring Cloud Config?

Use Vault for secure token-based secret storage, or encrypt values using Spring Cloud's encryption support (JCE).

66. What is HashiCorp Vault?

Vault is a secure tool to store secrets, API keys, and credentials. It supports access control, audit logging, and dynamic secret generation.

67. How can Spring Boot access secrets from Vault?

Using Spring Cloud Vault configuration:
spring:
cloud:
vault:
uri: http://localhost:8200
token:
generic:
backend: secret

68. What is the difference between Config Server and Vault?

Config Server handles environment configs, usually from Git.
Vault handles secure secret storage.

Config Server = settings; Vault = credentials.

69. Can secrets be versioned?

Yes. Git-based config is inherently versioned. Vault also supports versioned key-value storage and allows rollbacks.

70. How do you update secrets in a running microservice?

Use @RefreshScope and Spring Actuator. POST to /actuator/refresh to reload updated values dynamically.