Spring Cloud Config Server: Production Readiness Checklist

To ensure that your Spring Cloud Config Server setup is acceptable for a production environment, especially when handling sensitive values like passwords, tokens, or API keys, follow these best practices:

1. Use HTTPS End-to-End

- All communication (Client <-> Config Server <-> Git/Vault) must use HTTPS.
- Use trusted TLS certificates.

2. Enforce Authentication and Authorization

- Secure Config Server with Basic Auth, OAuth2, or mTLS.
- Use firewalls or IP allowlists.

3. Limit What Clients Can Access

- Use service-specific config files to prevent cross-access.

4. Encrypt Sensitive Properties on Server Only

- Use {cipher} values.
- Decrypt only on the server using a keystore.
- Never hardcode the encryption key.

5. Restrict Actuator Access

- Disable or secure /actuator/env, /configprops, and /beans.

6. Sanitize Logs and Actuator Output

- Use keys-to-sanitize to mask sensitive values in logs.

7. Use Vault or a Secrets Manager

- For high-security, use Vault or cloud secrets managers.
- Supports secret rotation, auditing, and access policies.

8. Enable Audit Logging & Monitoring

- Log all config requests.
- Monitor with tools like ELK, Grafana, or Datadog.

Spring Cloud Config Server: Production Readiness Checklist

9. Use Configuration Profiles & Branches

- Separate dev, test, and prod configs.

10. Harden the Infrastructure

- Deploy in secure private networks.
- Use API gateways or service mesh.
- Apply firewall rules and disable /encrypt & /decrypt in prod.

Production-Ready Checklist:

Feature	Required?	Status
HTTPS everywhere	Must	Yes
Authentication (Basic/OAuth2/mTLS)	Must	Yes
Secrets encrypted with {cipher}	Must	Yes
Clients get only plain values	Expected	Yes
Vault or secrets manager	Recommended	Yes
Logs/actuator sanitized	Must	Yes
Config repo access secured	Must	Yes
Infra hardened (VPC/firewall)	Must	Yes

Receiving decrypted config on clients is acceptable in production if:

- Transport is secure (HTTPS)
- Access is locked down and authenticated
- Secrets are encrypted at rest
- Logs are sanitized
- Clients access only their config
- (Preferably) secrets are managed via Vault