4. Security and compliance
└─ **4.4 Security and compliance scanning**
└─ **4.4.3 Github advanced security**

---

1. What features are included in GitHub Advanced Security (GHAS)?
2. How do you enable GHAS for a GitHub repository or organization?
3. What is CodeQL in GHAS and how does it function?
4. How does GHAS handle secret scanning and what are its capabilities?
5. How is dependency scanning managed within GHAS?
6. What is the difference between alerts and security overview in GHAS?
7. How do you enforce GHAS checks in branch protection rules?
8. What are the licensing and billing requirements for GHAS?
9. How does GHAS integrate with CI/CD workflows and GitHub Actions?
10. What are best practices for managing GHAS alerts and findings?

---

1. What features are included in GitHub Advanced Security (GHAS)?
GHAS includes

- CodeQL (static analysis),
- secret scanning,
- and dependency review.

It enhances repo security with

- alerts,
- PR checks,
- and security dashboards.

---

2. How do you enable GHAS for a GitHub repository or organization?
In *repo settings → Security & analysis*, enable features like Code scanning and Secret scanning. Organization admins must assign a GHAS license to the repo.

---

3. What is CodeQL in GHAS and how does it function?
CodeQL scans source code for vulnerabilities using semantic queries. It's configured via *GitHub Actions* and runs on PRs, pushes, or scheduled intervals.

---

4. How does GHAS handle secret scanning and what are its capabilities?
Secret scanning detects credentials (e.g., API keys, tokens) in pushes and PRs. Alerts are triggered, and some partner secrets are automatically revoked.

---

5. How is dependency scanning managed within GHAS?
GHAS shows known CVEs via the dependency graph and triggers *Dependabot* alerts. It identifies vulnerable versions and suggests secure updates.

---

6. What is the difference between alerts and security overview in GHAS?
Alerts show specific issues (e.g., a secret or CVE). Security overview aggregates all alerts, provides trends, and gives org-wide or repo-wide posture insight.

---

7. How do you enforce GHAS checks in branch protection rules?
Enable required status checks (e.g., CodeQL, secret scanning) under *Branch protection rules*. This blocks merging PRs with unresolved security issues.

---

8. What are the licensing and billing requirements for GHAS?
GHAS requires an Enterprise plan with Advanced Security enabled. Billing is per active committer per month for each enabled private repository.

---

9. How does GHAS integrate with CI/CD workflows and GitHub Actions?
GHAS tools (e.g., CodeQL) run as *GitHub Actions* in the CI pipeline. They can fail builds, comment on PRs, and prevent merging based on scan results.

---

10. What are best practices for managing GHAS alerts and findings?
Triage alerts promptly, prioritize critical issues, auto-create issues from alerts, integrate with project boards, and use codeowners for review responsibility.