Got it ✅ Let’s cover **SonarQube** (code quality & security) and **Trivy** (vulnerability scanner) together since they often appear in **DevSecOps interview questions**.

# 🔥 SonarQube – Most Asked Interview Questions & Answers

### 1. **What is SonarQube?**

👉 SonarQube is an open-source tool that continuously inspects code quality, checks for bugs, vulnerabilities, code smells, and ensures coding standards are followed.

### 2. **What are “Code Smells” in SonarQube?**

👉 Code Smells are maintainability issues that don’t break functionality but make the code harder to maintain (e.g., long methods, duplicated code).

### 3. **Difference between Bug, Vulnerability, and Code Smell in SonarQube?**

* **Bug** → Causes wrong behavior in runtime (null pointer, array index out of bounds).
* **Vulnerability** → Security risk (SQL injection, hardcoded credentials).
* **Code Smell** → Maintainability issue (complex functions, duplicated code).

### 4. **What is a Quality Gate in SonarQube?**

👉 A set of conditions (coverage, duplication, critical issues) that code must pass before being merged or deployed. Example: *Fail build if code coverage < 80%*.

### 5. **How does SonarQube integrate with CI/CD?**

👉 Using **Sonar Scanner plugins** in Jenkins, GitHub Actions, GitLab CI, or Azure DevOps. After build/test, SonarQube analyzes code and reports results to Quality Gates.

### 6. **Can SonarQube scan IaC or containers?**

👉 SonarQube mainly scans **codebases** (Java, Python, Go, etc.). For containers & IaC vulnerabilities, tools like **Trivy or Checkov** are used.

### 7. **How do you secure SonarQube itself?**

👉 Use authentication (LDAP/OAuth), HTTPS, role-based access, and ensure SonarQube server is patched.

# 🔥 Trivy – Most Asked Interview Questions & Answers

### 1. **What is Trivy?**

👉 Trivy is an open-source vulnerability scanner by Aqua Security that scans **container images, file systems, IaC (Terraform, Helm, K8s manifests), and Git repos** for vulnerabilities, misconfigurations, and secrets.

### 2. **What can Trivy scan?**

* **Container images** → OS packages & libraries.
* **File system** → Local project dependencies.
* **Infrastructure as Code (IaC)** → Terraform, Helm, Kubernetes YAML.
* **Git repos** → Secrets, vulnerabilities.

### 3. **Difference between Trivy and other scanners (Clair, Anchore, Grype)?**

👉 Trivy is **fast, lightweight, and supports multiple targets** (containers, IaC, secrets). Others like Clair/Anchore focus mainly on container images.

### 4. **How do you use Trivy in CI/CD?**

👉 Example (Jenkins or GitHub Actions pipeline):

```bash
trivy image myapp:latest --exit-code 1 --severity CRITICAL,HIGH
```

➡️ This fails the pipeline if critical/high vulnerabilities exist.

### 5. **What are Trivy’s output formats?**

👉 JSON, Table, and SARIF (for GitHub Security Dashboard).

### 6. **How does Trivy handle false positives?**

👉 Use `.trivyignore` file to exclude known vulnerabilities or CVEs.

### 7. **What’s the difference between Vulnerability Scanning & Misconfiguration Scanning in Trivy?**

* **Vulnerability Scan** → Checks OS & app dependencies for CVEs.
* **Misconfiguration Scan** → Checks IaC (Terraform, K8s) for bad practices (e.g., open security groups, pods running as root).

### 8. **How do you secure Kubernetes with Trivy?**

👉 Use **Trivy Operator** inside the cluster – it continuously scans workloads, reports vulnerabilities/misconfigs as Kubernetes CRDs.

# 🎯 Interview Pro-Tip

If asked *“SonarQube vs Trivy”*:

* **SonarQube** → Focused on **code quality & maintainability**.
* **Trivy** → Focused on **security vulnerabilities & misconfigurations**.
  👉 Both are complementary in a DevSecOps pipeline.

✅ Next Step:
Would you like me to also create a **pipeline flow diagram** showing how **SonarQube + Trivy + Jenkins/Terraform/K8s** work together in DevSecOps? That’s a very common **practical interview question**.

"In our DevSecOps pipeline, we use SonarQube and Trivy together to ensure code quality and security.

When a developer pushes code to Git, Jenkins triggers the pipeline. First, SonarQube scans the source code for bugs, vulnerabilities, and code smells. If the Quality Gate fails (e.g., low coverage, critical vulnerabilities), the pipeline stops.

Next, the application is containerized, and Trivy scans the Docker image for OS-level and library vulnerabilities. We enforce that no image with critical/high CVEs is deployed.

We also integrate Trivy IaC scanning to check Terraform and Kubernetes manifests for misconfigurations like overly permissive security groups or pods running as root.

Finally, only if both SonarQube and Trivy checks pass, we deploy to staging, run functional tests, and then promote to production with approval.

This ensures our pipeline enforces both code quality (SonarQube) and security compliance (Trivy) before production deployment."*