bestehen Sie GH-500 Ihre Prüfung mit unserem Prep GH-500 Ausbildung Material & kostenloser Dowload Torrent

Mit der Entwicklung der IT-Industrie nimmt die Zahl der IT-Lerner seit Jahren immer zu. Das führt zu immer stärkerer Konkurrenzen. Und es ist undenkbar, dass Sie in IT-Industrie von anderen überschritten sind. Deshalb sollen Sie Ihre Fähigkeit ständig erhöhen und Ihre Stärke zu anderen beweisen. Wie können Sie Ihre Fähigkeit zu anderen beweisen? Immer mehr Leute wählen IT-Zertifizierungen, Ihre Fähigkeit zu beweisen. Wollen Sie auch? Kommen Sie zuerst zu Microsoft GH-500 Zertifizierungsprüfung. Das ist die wichtigste Microsoft Prüfung und auch von vielen Unternehmen anerkannt.

Wir ITZert haben reiche Ressourcen und viele entsprechende Prüfungsfragen von Microsoft GH-500 Prüfungen. Und Wir ITZert bieten Ihnen auch die kostlose Demo von Microsoft GH-500 Zertifizierungsprüfungen. Sie können die Prüfungsfragen und Testantworten herunterladen. Wir ITZert bieten echte und umfassende Prüfungsfragen und Testantworten. Mit unseren besonderen Microsoft GH-500 Prüfungsunterlagen können Sie Microsoft GH-500 Prüfungen leicht bestehen. Wir ITZert garantieren 100% Erfolg.

>> GH-500 Originale Fragen <<

GH-500 Unterlage & GH-500 Probesfragen

Schulungsunterlagen zur Microsoft GH-500 Zertifizierungsprüfung von ITZert sind effizient, die von manchen Experten und einigen bestandenen Kandidaten bewiesen sind. Sie sind fast gleich wie die echten GH-500 Prüfungsfragen. Sie können Ihnen dabei helfen, die GH-500 Zertifizierungsprüfung zu bestehen. Wir werden Ihnen alle Ihren bezahlten Summe zurückgeben, entweder Sie die GH-

500 Prüfung nicht bestehen, oder die Testaufgaben von Microsoft GH-500 irgend ein Qualitätsproblem haben. Vertrauen Sie bitte auf ITZert, denn wir werden Ihnen stets begleiten.

Microsoft GH-500 Prüfungsplan:

Thema	Einzelheiten
Thema 1	• Configure and use Dependabot and Dependency Review: Focused on Software Engineers and Vulnerability Management Specialists, this section describes tools for managing vulnerabilities in dependencies. Candidates learn about the dependency graph and how it is generated, the concept and format of the Software Bill of Materials (SBOM), definitions of dependency vulnerabilities, Dependabot alerts and security updates, and Dependency Review functionality. It covers how alerts are generated based on the dependency graph and GitHub Advisory Database, differences between Dependabot and Dependency Review, enabling and configuring these tools in private repositories and organizations, default alert settings, required permissions, creating Dependabot configuration files and rules to auto-dismiss alerts, setting up Dependency Review workflows including license checks and severity thresholds, configuring notifications, identifying vulnerabilities from alerts and pull requests, enabling security updates, and taking remediation actions including testing and merging pull requests.
Thema 2	• Configure and use secret scanning: This domain targets DevOps Engineers and Security Analysts with the skills to configure and manage secret scanning. It includes understanding what secret scanning is and its push protection capability to prevent secret leaks. Candidates differentiate secret scanning availability in public versus private repositories, enable scanning in private repos, and learn how to respond appropriately to alerts. The domain covers alert generation criteria for secrets, user role-based alert visibility and notification, customizing default scanning behavior, assigning alert recipients beyond admins, excluding files from scans, and enabling custom secret scanning within repositories.
Thema 3	• Configure and use Code Scanning with CodeQL: This domain measures skills of Application Security Analysts and DevSecOps Engineers in code scanning using both CodeQL and third-party tools. It covers enabling code scanning, the role of code scanning in the development lifecycle, differences between enabling CodeQL versus third-party analysis, implementing CodeQL in GitHub Actions workflows versus other CI tools, uploading SARIF results, configuring workflow frequency and triggering events, editing workflow templates for active repositories, viewing CodeQL scan results, troubleshooting workflow failures and customizing configurations, analyzing data flows through code, interpreting code scanning alerts with linked documentation, deciding when to dismiss alerts, understanding CodeQL limitations related to compilation and language support, and defining SARIF categories.
Thema 4	• Describe the GHAS security features and functionality: This section of the exam measures skills of Security Engineers and Software Developers and covers understanding the role of GitHub Advanced Security (GHAS) features within the overall security ecosystem. Candidates learn to differentiate security features available automatically for open source projects versus those unlocked when GHAS is paired with GitHub Enterprise Cloud (GHEC) or GitHub Enterprise Server (GHES). The domain includes knowledge of Security Overview dashboards, the distinctions between secret scanning and code scanning, and how secret scanning, code scanning, and Dependabot work together to secure the software development lifecycle. It also covers scenarios contrasting isolated security reviews with integrated security throughout the development lifecycle, how vulnerable dependencies are detected using manifests and vulnerability databases, appropriate responses to alerts, the risks of ignoring alerts, developer responsibilities for alerts, access management for viewing alerts, and the placement of Dependabot alerts in the development process.

Thema 5	• Describe GitHub Advanced Security best practices, results, and how to take corrective measures: This section evaluates skills of Security Managers and Development Team Leads in effectively handling GHAS results and applying best practices. It includes using Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE) identifiers to describe alerts and suggest remediation, decision-making processes for closing or dismissing alerts including documentation and data-based decisions, understanding default CodeQL query suites, how CodeQL analyzes compiled versus interpreted languages, the roles and responsibilities of development and security teams in workflows, adjusting severity thresholds for code scanning pull request status checks, prioritizing secret scanning remediation with filters, enforcing CodeQL and Dependency Review workflows via repository rulesets, and configuring code scanning, secret scanning, and dependency analysis to detect and remediate vulnerabilities earlier in the development lifecycle, such as during pull requests or by enabling push protection.

Microsoft GitHub Advanced Security GH-500 Prüfungsfragen mit Lösungen (Q25-Q30):

25. Frage
How would you build your code within the CodeQL analysis workflow? (Each answer presents a complete solution. Choose two.)

- A. Use jobs.analyze.runs-on.
- B. Use CodeQL's init action.
- C. Implement custom build steps.
- D. Upload compiled binaries.
- E. Use CodeQL's autobuild action.
- F. Ignore paths.

Antwort: C,E

Begründung:
Comprehensive and Detailed Explanation:
When setting up CodeQL analysis for compiled languages, there are two primary methods to build your code:
GitHub Docs
Autobuild: CodeQL attempts to automatically build your codebase using the most likely build method. This is suitable for standard build processes.
GitHub Docs
Custom Build Steps: For complex or non-standard build processes, you can implement custom build steps by specifying explicit build commands in your workflow. This provides greater control over the build process.
GitHub Docs
The init action initializes the CodeQL analysis but does not build the code. The jobs.analyze.runs-on specifies the operating system for the runner but is not directly related to building the code. Uploading compiled binaries is not a method supported by CodeQL for analysis.

26. Frage
You are a maintainer of a repository and Dependabot notifies you of a vulnerability. Where could the vulnerability have been disclosed? (Each answer presents part of the solution. Choose two.)

- A. In security advisories reported on GitHub
- B. In manifest and lock files
- C. In the dependency graph
- D. In the National Vulnerability Database

Antwort: A,D

Begründung:
Comprehensive and Detailed Explanation:
Dependabot alerts are generated based on data from various sources:
National Vulnerability Database (NVD): A comprehensive repository of known vulnerabilities, which GitHub integrates into its advisory database.
GitHub Docs

Security Advisories Reported on GitHub: GitHub allows maintainers and security researchers to report and discuss vulnerabilities, which are then included in the advisory database.
The dependency graph and manifest/lock files are tools used by GitHub to determine which dependencies are present in a repository but are not sources of vulnerability disclosures themselves.

27. Frage
Where can you view code scanning results from CodeQL analysis?

- A. The repository's code scanning alerts
- B. At Security advisories
- C. A CodeQL query pack
- D. A CodeQL database

Antwort: A

Begründung:
All results from CodeQL analysis appear under the repository's code scanning alerts tab. This section is part of the Security tab and provides a list of all current, fixed, and dismissed alerts found by CodeQL.
A CodeQL database is used internally during scanning but does not display results. Query packs contain rules, not results. Security advisories are for published vulnerabilities, not per-repo findings.

28. Frage
Which CodeQL query suite provides queries of lower severity than the default query suite?

- A. github/codeql-go/ql/src@main
- B. github/codeql/cpp/ql/src@main
- C. security-extended

Antwort: C

Begründung:
The security-extended query suite includes additional CodeQL queries that detect lower severity issues than those in the default security-and-quality suite.
It's often used when projects want broader visibility into code hygiene and potential weak spots beyond critical vulnerabilities.
The other options listed are paths to language packs, not query suites themselves.

29. Frage
Which of the following is the best way to prevent developers from adding secrets to the repository?

- A. Make the repository public
- B. Create a CODEOWNERS file
- C. Enable push protection
- D. Configure a security manager

Antwort: C

Begründung:
The best proactive control is push protection. It scans for secrets during a git push and blocks the commit before it enters the repository.
Other options (like CODEOWNERS or security managers) help with oversight but do not prevent secret leaks.
Making a repo public would increase the risk, not reduce it.

30. Frage
.....

das Bestehen der Microsoft GH-500 Zertifizierungsprüfung gewinnen, der Ihnen bessere Entwicklung im IT-Bereich gewährleisten kann. Das Alles bedürft Ihres Vertrauens: Sie müssen auf ITZert vertrauen und Sie müssen zudem auf die Schulungsunterlagen zur Microsoft GH-500 Zertifizierungsprüfung vertrauen. Inhalt unserer Lehrmaterialien ist absolut echt und zuversichtlich. Darüber hinaus beträgt unsere Bestehensrate der Microsoft GH-500 Zertifizierungsprüfung 100%.

GH-500 Unterlage: https://www.itzert.com/GH-500_valid-braindumps.html