ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

You have subscribed to GitHub Premium Support, and you need to submit a support ticket. GitHub Premium Support can help you with:

A. writing scripts.
B. installing GitHub Enterprise Server.
C. setting up hardware.
D. integrating with third-party applications.

Answer: B

Explanation:
GitHub Premium Support includes assistance with installing and using GitHub Enterprise Server, ensuring your deployment is configured correctly and any installation issues are resolved.

Question: 2

You need to contact GitHub Premium Support. What are valid reasons for submitting a support ticket? (Each answer presents a complete solution. Choose two.)

A. license renewal
B. hardware setup issues or errors
C. business impact from security issues within your organization
D. outages on GitHub.com affecting core Git functionality

Answer: C, D

Explanation:
Business-impact security issues (for example, a critical vulnerability affecting your organization) are classified as High-priority tickets and are covered under your Premium Support SLA.
Outages on GitHub.com that disrupt core Git or web application functionality trigger Urgent-priority responses under Premium Support's SLA.

Question: 3

Which of the following is a key benefit of using GitHub Marketplace Apps in an enterprise?

A. They guarantee no downtime during enterprise GitHub maintenance windows
B. They often include integrations with external services, reducing the need for custom code
C. Apps eliminate the need for GitHub Actions entirely
D. All apps come pre-approved by GitHub's internal security team

Answer: B

Explanation:
GitHub Marketplace Apps come with built-in integrations to external services - so you can plug in things like CI servers, code-quality scanners, or deployment tools without writing and maintaining custom connectors.

Question: 4

You need to create a support bundle for your GitHub Enterprise Server instance with the hostname ghe.avocado.corp. What command should you use to create a support bundle?

A. ssh -p 122 adming@ghe.avocado.corp -- 'ghe-support-bundle -o' > support-bundle.tgz
B. ssh -p 122 adming@ghe.avocado.corp – 'ghe-diagnostics' > support-bundle.tgz
C. curl -u admin https://ghe.avocado.corp/diagnostics/support-bundle.tgz -o
D. ssh -p 122 adming@ghe.avocado.corp -- 'ghe-config generate-support-bundle' > support- bundle.tgz

Answer: A

Explanation:
Run the ghe-support-bundle command over SSH on your appliance and redirect its output to a file.
For example:
ssh -p 122 admin@ghe.avocado.corp -- 'ghe-support-bundle -o' > support-bundle.tgz
This invokes the built-in support-bundle utility on your GitHub Enterprise Server instance and captures the resulting archive locally.

Question: 5

What do you need to successfully generate a support bundle on a GitHub Enterprise Server?

A. Approval from GitHub Support
B. A custom GitHub Action in the root repo
C. Administrator SSH access to the appliance
D. A GitHub App with read:org permissions

Answer: C

Explanation:
You must have administrator-level SSH access to the GitHub Enterprise Server appliance so you can run the ghe-support-bundle command over SSH and capture the bundle locally.

Question: 6

A financial services company is evaluating GitHub account types. Which of the following is a key distinction between GitHub Enterprise Managed Users and Personal Accounts?

A. Enterprise Managed Users can collaborate across both personal and enterprise repositories.
B. Personal Accounts are owned by users and can be used for both personal and professional work.
C. Personal Accounts provide stricter control over repositories and user activity.
D. Enterprise Managed Users require the organization to manage their own authentication server.

Answer: B

Explanation:
Personal Accounts are owned and controlled by individual users and can serve both their personal projects and professional work, whereas Enterprise Managed Users exist solely within the enterprise context and

cannot be used for personal repositories.

Question: 7

Which THREE of the following accurately describe how the SCIM protocol enhances user management in GitHub Enterprise Cloud? (Choose three.)

A. SCIM synchronizes changes to user attributes from the identity provider to GitHub.
B. SCIM deactivates GitHub accounts when users are deleted from the identity provider.
C. SCIM automatically deletes organization repositories when administrators are removed.
D. SCIM automates user provisioning when new users are added to the identity provider.
E. SCIM generates authentication tokens for accessing GitHub's REST API.
F. SCIM configures repository permissions based on user roles within the organization.

Answer: A, B

Explanation:
SCIM automatically updates a user's account on GitHub whenever their profile attributes change in the identity provider.
When a user is removed or deactivated in the IdP, SCIM deactivates (soft-deprovisions) their GitHub account and disables access.
SCIM provisions new GitHub Enterprise Cloud accounts automatically when users are added in the identity provider.

Question: 8

When comparing a partner identity provider integration with a non-partner identity management solution for GitHub Enterprise Managed Users, which statement is Correct?

A. The non-partner identity provider integrations can utilize OIDC for authentication.
B. The non-partner identity provider integrations require manual configuration of SAML 2.0 details.
C. The partner identity provider integrations support fewer GitHub-supported authentication methods.
D. The partner identity provider integrations rely on the partner to support the application on the partner IdP.

Answer: B

Explanation:
Non-partner identity provider integrations require you to enter SAML 2.0 configuration details by hand - such as the Sign-on URL, Issuer, and X.509 certificate - whereas partner IdPs supply a preconfigured application integration.

Question: 9

When comparing Group SCIM to Team Sync for identity management in GitHub Enterprise, which statement is Correct?

A. Group SCIM requires less initial configuration than Team Sync.
B. Team Sync supports more identity providers than Group SCIM.
C. Team Sync provides more automated user deprovisioning than Group SCIM.
D. Group SCIM enables centralized user and group management through the IdP.

Answer: D

Explanation:
Group SCIM lets you manage both user accounts and group memberships centrally in your identity provider - automatically provisioning, updating, and deprovisioning users and groups in GitHub - whereas Team Sync only mirrors IdP group membership into existing GitHub teams.

Question: 10

Why is a GitHub App preferred over a PAT for machine authentication?

A. GitHub Apps are required to pass SAML assertions
B. GitHub Apps have time-limited installation tokens with scoped access
C. PATs cannot be used in GitHub Actions
D. PATs support fewer GitHub APIs than Apps

Answer: B

Explanation:
GitHub Apps issue short-lived installation tokens that you scope to only the permissions and repositories your automation needs, reducing blast radius and automatically rotating credentials.

Question: 11

You are planning GitHub account management for a healthcare organization with strict compliance requirements. Which THREE of the following statements accurately describe GitHub Enterprise Managed Users (EMU) accounts? (Choose three.)

A. EMU accounts can be used for both personal and enterprise repositories.
B. EMU accounts are managed through an identity provider such as Azure AD.
C. EMU accounts allow users to create and manage their own credentials.
D. EMU accounts restrict users to enterprise-related activities only
E. EMU accounts are created and managed by individual users.
F. EMU accounts are owned by the organization and cannot be unlinked.

Answer: B, D, F

Explanation:
Enterprise Managed User accounts are provisioned and authenticated exclusively through your identity provider (for example, Azure AD), so the IdP handles their creation, attribute updates, and deprovisioning. Managed user accounts cannot create public content or interact with repositories outside your enterprise; they're confined to private and internal repos within the enterprise.
EMU accounts are owned and controlled by the enterprise (via the IdP) and cannot be converted into or

unlinked as personal accounts outside that enterprise.

Question: 12

A GitHub Enterprise administrator is planning to implement SAML SSO across their company. Which of the following correctly distinguishes enterprise-wide SAML SSO from organization-level SAML SSO?

A. Enterprise-wide SAML SSO requires less initial administrative overhead than organization-level implementation.
B. Enterprise-wide SAML SSO allows different organizations to use different authentication methods.
C. Enterprise-wide SAML SSO immediately removes users who fail to authenticate via the IdP.
D. Enterprise-wide SAML SSO ensures users authenticate through the same IdP across all organizations.

Answer: D

Explanation:
Enterprise-wide SAML SSO enforces a single IdP across all member organizations—its configuration overrides any per-organization SAML settings, so everyone must authenticate through the same provider.

Question: 13

What distinguishes Enterprise Managed Users (EMUs) from standard GitHub accounts?
A. EMUs are fully controlled by an IdP and cannot log in with personal credentials
B. EMUs can only be created using email invites
C. EMUs are managed in GitHub and use GitHub authentication
D. EMUs are only available for GitHub Enterprise Server

Answer: A

Explanation:
EMU accounts are provisioned and authenticated exclusively through your identity provider - users sign in via the IdP and cannot use or manage GitHub-native credentials.

Question: 14

Your organization is implementing team synchronization. Which of the following should you prioritize during the setup process?

A. Disabling the audit log stream
B. Setting an infrequent sync schedule to reduce performance impact
C. Allowing manual updates to team memberships
D. Clearly define how identity provider groups will align with GitHub teams and roles

Answer: D

Explanation:
Before you enable team synchronization, you should clearly define how groups in your identity provider will

map to GitHub teams and roles - ensuring that when the sync runs, users land in the correct teams with the right permissions.

Question: 15

What makes GitHub Apps a more secure choice for automation over OAuth Apps?

A. GitHub Apps always require two-factor authentication.
B. GitHub Apps can only be installed by organization owners.
C. GitHub Apps are limited to read-only access and cannot write to repositories.
D. GitHub Apps authenticate as an app with fine-grained permissions, not as a user.

Answer: D

Explanation:
GitHub Apps authenticate as themselves with fine-grained, installation-scoped permissions and short-lived tokens - rather than inheriting a user's broad OAuth scopes - minimizing blast radius and aligning with least-privilege principles.

Question: 16

Why would a GitHub App be favored over a machine account for automation tasks?

A. Machine accounts are required for webhook delivery.
B. GitHub Apps provide a higher rate limit ceiling than using a personal access token on a machine account, when they use an install token and are owned by a GitHub Enterprise Cloud licensed enterprise.
C. GitHub Apps are limited to a single repository.
D. Machine accounts are easier to audit than GitHub Apps.

Answer: B

Explanation:
GitHub Apps authenticate with short-lived installation tokens scoped to fine-grained permissions and, when owned by a GitHub Enterprise Cloud organization, enjoy a higher rate limit (15,000 requests/hour) compared to a machine account's personal access token.

Question: 17

When comparing fine-grained Personal Access Tokens (PATs) with classic PATs, which of the following statements is accurate?

A. Fine-grained PATs automatically renew while classic PATs require manual renewal.
B. Fine-grained PATs permissions can be scoped to specific repositories.
C. Classic PATs offer more permission controls than fine-grained PATs.
D. Classic PATs can be restricted to specific organizations, but fine-grained PATs cannot.

Answer: B

Explanation:
Fine-grained personal access tokens let you scope permissions down to individual repositories, whereas classic PATs grant access across every repo the user can reach.

Question: 18

What is the new capability of GitHub's billing dashboard?

A. Automatically removes unused users from billing
B. Enables tracking of GitHub Copilot usage by user
C. Allows self-service plan upgrades
D. Offers real-time Slack alerts for billing

Answer: B

Explanation:
The revamped Billing & Licensing dashboard now includes a dedicated “Copilot” tab that shows peruser seat assignments, usage counts, and estimated costs for your organization’s GitHub Copilot licenses, enabling you to track Copilot consumption by individual users.

Question: 19

What is a key characteristic of GitHub Enterprise Server (GHES) compared to GitHub Enterprise Cloud (GHEC)?

A. GHES is hosted by GitHub and offers automatic scaling, while GHEC requires self-hosting.
B. GHEC offers data residency options in regions that GHES does not support.
C. GHES allows enterprises to have complete control over their hosting environment, including data storage and network security policies.
D. GHES users cannot integrate with external identity providers for authentication.

Answer: C

Explanation:
GitHub Enterprise Server is a self-hosted product you install and manage on your own infrastructure - giving you full control over data storage, network security policies, and the underlying environment.

Question: 20

Your organization wants to reduce costs. Which of the following actions should you take?

A. Grant all users admin permissions
B. Remove all outside collaborators
C. Regularly audit for inactive users
D. Disable SAML SSO for members

Answer: C

Explanation:
Regularly auditing for inactive (dormant) users lets you suspend or remove accounts that aren't consuming seats - freeing up licenses and directly lowering your per-user subscription costs.

Question: 21

How does metered billing work in GitHub Enterprise Cloud with Enterprise Managed Users (EMU)?

A. Billing is based on number of total users in the enterprise
B. Billing is based on owners and members of GitHub organizations
C. Billing is based on total users in the enterprise that are not dormant
D. Billing is based on the number of users created in Azure AD

Answer: A

Explanation:
Billing for GitHub Enterprise Cloud under metered (usage-based) billing is calculated by the total number of Enterprise Managed Users (and other license-consuming accounts) in your enterprise - each EMU consumes a seat and contributes to the monthly bill.

Question: 22

A team member is unable to push to a repository due to a 403-error related to branch protection. What should the GitHub Enterprise administrator do first?

A. Remove the user from the team and re-add them
B. Check the user's permissions and rulesets applied to the branch
C. Raise a GitHub Support request for permissions issues
D. Revert the branch to an earlier state

Answer: B

Explanation:
The administrator should first review the user's repository role and the branch protection rules applied to that branch. A 403 error on push almost always indicates that the user either lacks the necessary write permissions or is not listed among the actors authorized by the branch protection settings.

Question: 23

Which of the following is true about outside collaborators in a GitHub organization?

A. They are granted explicit access to specific repositories.
B. They inherit organization-wide policies, such as SSO requirements.
C. They have access to all private repositories by default.
D. They appear in the organization's internal member list.

Answer: A

Explanation:

Outside collaborators aren't organization members; instead, they're granted explicit access - at read, write, or admin level - to only the repositories you choose.

Question: 24

Which of the following is a benefit of creating a new GitHub organization?

A. Automatic inheritance of policies from other organizations.
B. Reduced administrative overhead.
C. Clear separation of reggs, projects, teams, billing, and organization-specific policies.
D. Simplified collaboration across all organizations.

Answer: C

Explanation:

Creating a new organization gives you a dedicated container for your shared work, letting you isolate repositories, projects, teams, billing settings, and policy configurations on an organization-by- organization basis.

Question: 25

Which of the following is the responsibility of an Organization Owner in GitHub? (Choose three.)

A. View and manage organization billing information.
B. Create repositories without approval from other members.
C. Manage organization settings, such as configuration and default permissions.
D. Access repositories only if explicitly granted by a team maintainer.

Answer: A, B, C

Explanation:

Organization owners can view and edit billing information for the organization.

Organization owners may create new repositories in the organization without needing approval from other members.

Organization owners have full administrative control over organization settings, including configuring default repository permissions.

Question: 26

Which of the following actions can a user with Write permissions perform in a GitHub repository?

A. Manage repository settings, such as labels and GitHub Pages.
B. Push code to non-protected branches.
C. Configure branch protection rules.

D. Delete the repository.

Answer: B

Explanation:

Users granted Write permission can push commits to non-protected branches, allowing them to update code without needing administrative rights.

Question: 27

Which of the following is a key benefit of setting default read permissions across organizations?

A. Suits environments where all users need write access.
B. Improves collaboration by allowing users to modify content directly.
C. Increases efficiency in content creation and updates.
D. Enhances security by minimizing unintended modifications.

Answer: D

Explanation:

Enforcing a default of Read for organization members ensures they can view content without the ability to push changes, reducing the risk of accidental or unauthorized modifications.

Question: 28

Which of the following is the responsibility of a Team Maintainer in a GitHub organization? (Choose two.)

A. Modifying organization-wide settings.
B. Managing nested sub-teams.
C. Adding or removing team members.
D. Deleting repositories assigned to the team.

Answer: B, C

Explanation:

Team maintainers can manage nested sub-teams - requesting to add or change parent/child teams within the organization's hierarchy.

Team maintainers have permission to add and remove members from their team, controlling day-today team membership.

Question: 29

You are managing a repository in your organization's GitHub account. A team member asks you to confirm who has access to the repository and their permission levels. Which tool should you use to review and manage repository access?

A. GitHub Pages Settings.

B. GitHub Actions Logs.
C. Repository Settings > Manage Access.
D. Branch Protection Rules.

Answer: C

Explanation:
Use the Repository Settings → Manage Access page to view all users and teams with access and their assigned permission levels.

Question: 30

When a user becomes a member of multiple GitHub organizations, which THREE of the following are important considerations for administrators? (Choose three.)

A. The user will automatically have the same role across all organizations.
B. The user's repository access and/or team membership needs to be managed separately for each organization.
C. The user will need to authorize credentials separately for each SAML-enabled organization.
D. The user will have different permission levels in each organization.
E. The user's profile information becomes private to non-organization members.
F. The user's personal repositories will become accessible to all organizations.

Answer: B, C, D

Explanation:
A user’s repository access and team memberships are scoped to each organization, so admins must configure permissions separately per org.
When an organization enforces SAML SSO, each member must authorize their personal access tokens or SSH keys for that org, requiring separate approval for each SAML-enabled organization
Roles and permission levels (owner, member, billing manager, repository roles, etc.) are assigned on a per-organization basis, so a user often has different permissions in different organizations.

Question: 31

A token was used to access an organization's resource via API. What fields in the audit log help determine who used it?

A. The token's permissions and the geographic region of access
B. The token expiration date
C. The GitHub Actions runner name
D. The token ID, requesting IP address, and associated user

Answer: D

Explanation:
The audit log records the token’s identifier (the hashed_token value), the source IP address of the request, and the actor (the user or app) associated with that token, allowing you to trace exactly who used it.

Question: 32

What will happen if Dependabot discovers a vulnerable transitive dependency in a repository?

A. It creates a pull request to update the direct dependency to a version that resolves the vulnerability.
B. It opens a pull request to update the affected package directly, regardless of version compatibility.
C. It automatically removes the package from the repository.
D. It sends an email to the repository owner but does not alter code.

Answer: A

Explanation:
Dependabot will automatically open a pull request that updates the direct dependency to a version which, in turn, resolves (or removes) the vulnerable transitive dependency—ensuring the fix is applied via your declared dependencies.

Question: 33

Which GitHub feature is responsible for tracking dependencies and known vulnerabilities in those dependencies from an advisory database?

A. Repository Insights
B. Dependency Graph
D. Security Policy
D. CodeQL

Answer: B

Explanation:
The Dependency Graph continuously analyzes your repository's manifest and lock files to build an inventory of direct and transitive dependencies and flags any that match entries in the GitHub Advisory Database, surfacing known vulnerabilities.

Question: 34

Which events from the audit log are exposed by the GraphQL API? Each answer presents a complete solution. (Choose three.)

A. changes in permissions
B. promoting users to administrators
C. pushes to repositories
D. changes to permissions of a GitHub App
E. cloning of repositories

Answer: A, B, D

Explanation:
The GraphQL Audit Log API surfaces entries whenever repository or organization permissions are changed

("Changes permissions").
It records when users are elevated to administrative roles ("Promotes users to admin").
It logs alterations to a GitHub App's granted permissions ("Changes permissions of a GitHub App").

Question: 35

When a token is used to perform actions across different GitHub resources, how is this reflected in audit logs?

A. Each API action made with the token generates a separate audit log entry
B. Only the first repository accessed is recorded
C. GitHub creates a ZIP archive of all token activity
D. The audit log stores only the token name and not its actions

Answer: A

Explanation:
Each API call authenticated with a token generates its own audit-log event, so you'll see a distinct entry for every action performed across different resources, each annotated with the token's hashed ID, actor, and source IP.

Question: 36

Which practice helps avoid service disruption when consuming GitHub APIs at scale?

A. Designing your application to work within GitHub's rate limits
B. Using multiple tokens to bypass limits
C. Caching all API responses permanently
D. Ignoring secondary rate limits

Answer: A

Explanation:
Designing your integration to stay within GitHub's documented rate limits—by batching requests, using conditional requests, handling 429 responses with back-off, and monitoring the X-RateLimit-* headers - ensures you won't be temporarily throttled or cut off when you hit secondary limits.

Question: 37

How does GitHub handle secrets found via secret scanning in a public repository?

A. It alerts the service provider (e.g., AWS, Stripe).
B. It immediately blocks the commit to protect the secret.
C. It deletes the secret from the repository automatically.
D. It notifies the admin via webhook.

Answer: A

Explanation:

When secret scanning detects a supported credential in a public repository, GitHub notifies the issuing service provider so they can revoke or rotate the exposed secret.

Question: 38

Our organization is updating its enterprise policies. Which of the following steps should you take to ensure alignment with security requirements?

A. Maintain clear documentation of existing policies and policy changes.

B. Implement the new enterprise policies across the organization first and then consult with the security team to identify- any necessary adjustments or retrofits

C. Implement changes without consulting stakeholders.

D. Regularly assess and adjust policies based on evolving risks.

Answer: A, B

Question: 39

Which of the following correctly describes the difference between controlling actions at the enterprise level versus the organization level in GitHub?

A. Enterprise policies and organization policies are independent, with organization policies taking precedence for repositories within the organization.

B. Enterprise policies configure mandatory settings for organizations.

C. Enterprise policies apply only to public repositories, while organization policies apply to public, internal, and private repositories.

D. Enterprise policies can block specific actions, while organization policies can only enable or disable actions entirely.

Answer: B

Explanation:

Enterprise policies let you define and enforce mandatory settings across all member organizations - organization-level policies then operate within the options that the enterprise policy exposes.

Question: 40

What is the potential consequence of enabling multiple rulesets that apply to the same branch in a repository?

A. Only organization-level rulesets are enforced over repository-level ones
B. All applicable rulesets will be evaluated, and their combined rules enforced
C. Only the most recently created ruleset will be enforced
D. Rulesets will override each other, leading to unpredictable behavior

Answer: B

Explanation:
If you enable multiple rulesets that target the same branch, GitHub will evaluate every matching ruleset and enforce the aggregate of their rules - so all constraints from all applicable rulesets apply.

Question: 41

In a GitHub repository using Dependabot, which of the following best describes the purpose of the .github/dependabot.yml file?

A. It configures scheduling, package ecosystems, and target directories for update checks.
B. It lists commit SHAs to exclude from automatic pull requests.
C. It enables GitHub to scan for secrets in dependency files.
D. It encrypts dependency versions before storing them in the repo.

Answer: A

Explanation:
The .github/dependabot.yml file defines Dependabot's package-ecosystem, the directories to inspect, and the update schedule (daily/weekly/monthly), controlling when and where Dependabot checks for new versions.

Question: 42

What is the key benefit of using a GitHub security advisory within a repository?

A. It automatically reverts commits that introduced the vulnerability.
B. It allows maintainers to privately disclose, discuss, and publish vulnerabilities.
C. It flags all forks of the repository as vulnerable.
D. It prevents users from cloning the repository until issues are resolved.

Answer: B

Explanation:
GitHub security advisories let maintainers privately disclose, discuss fixes, and then publish vulnerabilities in a controlled manner within the repository.

Question: 43

How does GitHub support compliance requirements for enterprises?

A. GitHub provides configurable controls such as an audit log, SAML authentication, and enterprise

rulesets.

B. GitHub disables all external collaboration features.

C. GitHub only allows those with repository owner (admin) permissions to write changes to repositories.

D. GitHub automatically encrypts user passwords in plaintext for quick access.

Answer: A

Explanation:
GitHub Enterprise gives you a suite of configurable controls - like a comprehensive audit log, enforced SAML single sign-on, and enterprise-level rulesets - that you can tailor and enforce to meet your organization's compliance mandates.

Question: 44

You discover that a secret (e.g., a token or password) was accidentally committed to a GitHub repository. What is the first step you should take to mitigate the risk?

A. Contact GitHub Support to remove the secret from all forks and clones of the repository.

B. Revoke and/or rotate the secret to render it unusable, then assess whether history rewriting is necessary.

C. Rewrite the repository history using git filter-repo or BFG Repo-Cleaner to remove the secret from all commits.

D. Delete the repository and create a new one to ensure the secret is no longer accessible.

Answer: B

Explanation:
The immediate priority is to revoke or rotate the exposed credential so it can no longer be used; once it's invalidated, you can safely proceed with history-rewriting or other cleanup steps.

Question: 45

Why would someone choose to configure a security policy?

A. To communicate corporate security and compliance policies for end users on a private repository. B. To provide information on an open source repository for open source collaborators and researchers that may need to report and disclose sensitive security findings to maintainers securely.

C. To prevent anyone from pushing to the repository without approval.

D. To define which open source packages are permitted for use as part of that repository.

Answer: B

Explanation:
A security policy (the SECURITY.md file) lets maintainers of an open source repository provide clear, private instructions for collaborators and external researchers on how to report and disclose security vulnerabilities responsibly.

Question: 46

How is CodeQL different from other static analysis tools?

A It removes insecure code automatically

B. It allows querying of code semantics using a database-like language.

C. It only works for open-source projects.

D. It runs analysis only after a security breach.

Answer: B

Explanation:

CodeQL differs from traditional static analysis tools by ingesting your code into a queryable database and letting you write QL queries - its own database-style language - to express semantic checks and find patterns across the codebase.

Question: 47

Your enterprise has multiple organizations, and you want to ensure consistent security policies across all teams. Which feature should you use?

A. Outside collaborators for all repositories.

B. Organization-specific teams with custom policies.

C. Enterprise-level teams with inherited enterprise policies.

D. Assigning admin permissions to all team members.

Answer: C

Explanation:

By using enterprise-level teams with inherited enterprise policies, you can group members across all your organizations and enforce the same security settings globally - ensuring every team abides by the enterprise's mandatory policies.

Question: 48

What benefit does GitHub Advanced Security provide?

A. helps organization administrators analyze and configure permissions to the least privilege required

B. helps developers improve and maintain the security and quality of code

C. helps enterprise administrators improve and maintain network security for their GitHub Enterprise Server instances

D. helps organization administrators manage security tokens

Answer: B

Explanation:

GitHub Advanced Security equips developers with built-in code scanning (CodeQL), secret scanning, dependency review, and other AppSec tools - helping them find, fix, and prevent security vulnerabilities while maintaining code quality.

Question: 49

Which Git operation is not included in the Git activity audit log?

A. Delete branch
B. Fetch
C. Push
D. Clone

Answer: A

Explanation:
Delete branch operations aren't tracked as Git-activity events; the Git activity audit log only records Git events such as clone, fetch (pull), and push.

Question: 50

You are an administrator and need to enforce a policy on forking private and internal repositories. Which options are available for configuring the policy at the enterprise level? (Each answer presents a complete solution. Choose three.)

A. Allow organization owners to administer the setting at the organization level.
B. Allow people who have access to private and internal repositories to fork these repositories.
C. Allow specific people or teams to fork private and internal repositories.
D. Disallow repository owners from administering the setting at the repository level.
E. Disallow forking of private and internal repositories.

Answer: A, B, E

Explanation:
You can configure the enterprise policy to allow organization owners to administer the forking setting at the organization level, giving them control over how repos fork within their orgs.
You can choose to allow any user who already has access to a private or internal repo to fork it.
You can also set the policy to never allow forking of private or internal repositories across all organizations.

Question: 51

What additional capability does secret scanning offer for private repositories on GitHub Enterprise Cloud?

A. Allows custom pattern definitions for internal secret formats.
B. Disables any code that contains a secret.
C. Rewrites history to remove secrets.
D. Revokes GitHub access tokens automatically.

Answer: A

Explanation:

Secret scanning in private repositories on GitHub Enterprise Cloud lets you define and use custom regular-expression patterns - so you can detect internal or proprietary secret formats beyond the default partner-provided types.

Question: 52

What is the first step when sensitive data is accidentally pushed to a public GitHub repository?

A. Revoke any exposed credentials immediately
B. Force push a commit removing the data
C. Open an issue to inform users
D. Delete the repository

Answer: A

Explanation:

Revoke and/or rotate the exposed credentials immediately so they can no longer be used - this is the critical first step before you undertake any history-rewriting or cleanup.

Question: 53

How does Dependabot determine which security update PRs to open?

A. It waits for manual triage of all CVEs.
B. It uses the dependency graph and Dependabot alerts to open PRs for patched versions.
C. It reads the GitHub Issues and automatically suggests fixes.
D. It compares your codebase to the GitHub Trending list.

Answer: B

Explanation:

Dependabot relies on your repository's enabled Dependency Graph and Dependabot Alerts to identify vulnerable dependencies; it then automatically opens pull requests to update to the patched versions that resolve those alerts.

Question: 54

Which of the following GitHub token types supports fine-grained repository permissions AND is recommended for CI/CD automation?

A. Personal Access Tokens (PATs)
B. GitHub App Installation Access Tokens
C. Device Tokens
D. OAuth tokens

Answer: B

Explanation:
GitHub App Installation Access Tokens are privileged to the exact permissions you grant the App - down to individual repositories - and rotate automatically, making them the recommended choice for CI/CD automation workflows that demand least-privilege, fine-grained access.

Question: 55

Which of the following accurately contrasts a GitHub App and a GitHub Action?

A. GitHub Apps can only be used inside .github/workflows
B. GitHub Actions are limited to reading repository content only
C. GitHub Apps run only on GitHub-provided virtual machines, while GitHub Actions run only on customer-hosted machines
D. GitHub Actions can only be used to respond to events within a single repository while GitHub Apps can respond to events from multiple repositories

Answer: D

Explanation:
GitHub Actions workflows are defined and triggered within a single repository's context, whereas GitHub Apps are installed at the organization or user level and can subscribe to events across multiple repositories.

Question: 56

Which of the following are valid ways to pass data to a reusable workflow in a separate repository?

A. Use environment variables to pass data directly to the reusable workflow.
B. Define inputs in the reusable workflow and pass values from the calling workflow.
C. Define the secrets in the caller repository and call the reusable workflow using the 'secrets' keyword.
D. Define the secrets in the reusable workflow's repository and reference the secret using the 'secrets' context.

Answer: B, C

Explanation:
You declare named inputs in the reusable workflow's on.workflow_call block and then pass values from the caller using the with keyword, allowing the called workflow to consume those parameters.
You define required secrets in the caller repository and supply them to the reusable workflow via the secrets keyword in the workflow-call step, ensuring sensitive values are securely passed.

Question: 57

An organization wants to share a single API key required for their Actions workflows. They need to restrict its

use to only a subset of repositories. Where should they configure the secrets to minimize maintenance?

A. Repository secrets

B. Environment secrets
C. Organization secrets
D. Development environment secrets

Answer: C

Explanation:
By defining the API key as an organization secret, you centralize management and can grant access only to the subset of repositories you choose - eliminating per-repo duplication while enforcing the desired scope.

Question: 58

Which feature is unique to self-hosted runners?

A. Execute scripts before and after a job
B. Dynamic scaling
C. Automatic updates to the operating system
D. GPU support

Answer: A

Explanation:
Self-hosted runners support custom pre- and post-job scripts via runner hooks, letting you run arbitrary scripts before a job starts and after it finishes - capabilities not available on GitHub-hosted runners.

Question: 59

What is the effect of enforcing a policy that restricts GitHub Actions to only those created by the enterprise?

A. Marketplace actions are allowed only with SSO enabled
B. Actions can only be triggered by organization members
C. Only actions created within the enterprise are allowed
D. All public actions are allowed

Answer: C

Explanation:
When you enforce the “Allow enterprise actions and reusable workflows” policy, GitHub will block all workflows from using actions or reusable workflows that aren’t defined in a repository within your enterprise - so only actions created inside your enterprise are allowed.

Question: 60

You want to ensure a secret is automatically available to only workflows in internal and private repositories in the organization. Where do you configure the required access policy?

A. Actions policies
B. Runner groups
C. Rulesets
D. Organization secret

Answer: D

Explanation:
You set the access policy on the Organization Secret itself - configuring its visibility so it's scoped automatically to only internal and private repositories.

Question: 61

What needs to be done to ensure that only specific repositories can access the runners in an organization runner group?

A. Use GitHub's meta API to configure access.
B. Add a label to the runner group.
C. Configure repository access in the runner group settings.
D. Configure the Actions Policies to "Only selected repositories".

Answer: C

Explanation:
In the organization's runner group settings, switch the access from "All repositories" to "Selected repositories" and then explicitly choose which repos may use those runners.

Question: 62

You are using GitHub-hosted runners and need to securely deploy to an internal system. The security team requires that these runners use IP address ranges that would not be shared with other companies. Which of the following approaches would meet their requirements?

A. GitHub-hosted larger runners with Azure private networking
B. GitHub-hosted standard runners, using the IP addresses provided in "actions" from https://api.github.com/meta
C. GitHub-hosted standard runners, using the IP addresses provided in "api" from https://api.github.com/meta
D. GitHub-hosted larger runners with static IP addresses

Answer: D

Explanation:
GitHub's larger runners let you reserve dedicated static IP addresses for your workflows - so you can allow-list those IPs in your firewall and be sure they aren't shared with any other tenant.

Question: 63

Which factor affects GitHub Actions pricing for GitHub-hosted runners on GitHub Enterprise Cloud?

A. Number of workflows defined in .github/workflows/
B. Number of contributors to the repository Explanation:Incorrect. Contributor count does not impact billing for Actions
C. Total number of repositories using Actions
D. Operating system used in the runner environment

Answer: D

Explanation:
GitHub Actions billing for GitHub-hosted runners is based on the number of minutes consumed and the operating system of the runner - Linux, Windows, and macOS each have different per-minute rates.

Question: 64

You need GitHub to automatically notify a third-party service any time a new repository is created. You want to avoid writing custom code. The vendor has told you that they have a tool in the GitHub Marketplace. Which type of tool do you need?

A. GitHub App
B. GitHub Copilot Extension
C. GitHub Models
D. GitHub Action

Answer: A

Explanation:
You need a GitHub App. Marketplace integrations that listen for events like repository.created and send notifications are delivered as GitHub Apps, since they can subscribe to organization-level webhooks without you writing custom code.

Question: 65

Which product's usage is not included in GitHub Enterprise Cloud's monthly metered billing report?

A. Git LFS bandwidth
B. GitHub Actions minutes
C. GitHub Discussions engagement

D. GitHub Packages storage

Answer: C

Explanation:

GitHub Discussions engagement isn't a metered product and doesn't appear in the "Product billing" list, so its usage isn't included in the monthly metered billing report.