PassQueen
a.k.a dumpsqueen.com

GitHub FoundationsExam

GitHub GitHub-Foundations

Version Demo

Total Demo Questions: 7

Total Premium Questions: 74

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Introduction to Git and GitHub	14
Topic 2, Working with GitHub Repositories	11
Topic 3, Collaboration Features	21
Topic 4, Modern Development	14
Topic 5, Project Management	3
Topic 6, Privacy, Security, and Administration	11
Total	74

QUESTION NO: 1

What are some scenarios that can automatically subscribe you to conversations on GitHub?

(Each answer presents a complete solution. Choose three.)

A. Pushing a commit to the default branch

B. Being added as a repo admin

C. Opening a pull request or issue

D. Commenting on a thread

E. Being assigned to an issue or pull request

ANSWER: C D E

Explanation:

Opening a pull request or issue automatically subscribes the author to that conversation. This subscription lets the author receive notifications when people comment, change the issue or pull request, add labels, or otherwise generate notification-worthy activity. GitHub treats the creator as a participant because they initiated the discussion and will commonly need to follow its progress.

Commenting on a thread also automatically subscribes a user to the relevant issue, pull request, or discussion conversation. By participating with a comment, the user indicates an interest in subsequent activity, so GitHub delivers notifications for future updates unless the user later unsubscribes or changes notification settings.

Being assigned to an issue or pull request automatically subscribes the assignee as well. Assignment identifies that person as responsible for taking action or contributing to the work, making notifications essential for keeping track of related discussion and changes. These automatic subscriptions are part of GitHub's participation-based notification behavior; users can subsequently manage or unsubscribe from individual conversations. GitHub documents automatic subscriptions for authoring, commenting on, and being assigned issues or pull requests in its guide to viewing subscriptions and explains related controls in subscribing to and unsubscribing from conversations.

QUESTION NO: 2

How can a user choose to receive ongoing updates about a specific activity on GitHub.com?

A. By automatically watching all repositories you have push access to

B. By upgrading from a free to a paid account

C. By subscribing to notifications for all activity in a repository

D. By customizing the types of notifications you will receive in the future

ANSWER: C

Explanation:

By subscribing to notifications for all activity in a repository is correct because GitHub lets users watch repositories and select an *All Activity* notification setting. This subscription sends notifications for activity occurring in that repository, helping a user stay informed about updates such as issues, pull requests, discussions, releases, and other repository events. Watching is a repository-level choice that a user can enable when the project is relevant to them, rather than requiring account-plan changes or relying solely on participation-based notifications.

GitHub's notification system is designed to let users control the level of repository activity they follow. Selecting all activity is appropriate when someone needs continuous awareness of work across a particular project. The user can later manage the subscription from the repository's Watch menu or from notification settings, including changing the level of updates if their needs change. GitHub documents repository watching and its available notification levels in its guidance on configuring

notifications: Configuring notifications. For the repository watch controls and the meaning of the available watching choices, see Managing subscriptions for watched repositories.

QUESTION NO: 3

Which of the following are included as pre-defined repository roles?

(Each answer presents a complete solution. Choose three.)

A. Security

B. View

C. Triage

D. Maintain

E. Delete

F. Write

ANSWER: C D F

Explanation:

GitHub's predefined repository roles include **Triage**, **Maintain**, and **Write**. These are part of GitHub's standard repository permission model, which lets organizations and repository owners assign an access level that matches a collaborator's responsibilities. Triage is intended for contributors who need to manage the project's collaborative workflow, such as working with issues and pull requests, without being able to push directly to the repository. Maintain is a higher-level role for trusted maintainers who need to manage many repository-level operational settings and workflows while not having the complete administrative authority associated with the Admin role. Write provides the ability to actively contribute to the repository, including pushing commits and working with issues and pull requests. GitHub documents these roles alongside the additional predefined levels Read and Admin; their availability can vary based on repository visibility and whether the repository belongs to an organization. Using these predefined roles supports least-privilege access by assigning only the capabilities needed for each collaborator's work. See GitHub's repository roles for an organization documentation and its overview of repository roles for the role definitions and permissions.

QUESTION NO: 4

Which of the following statements most accurately describes secret gists?

A. Anyone with the URL for the gist can view the gist.

B. Secret gists require GitHub Enterprise.

C. Anyone can see the gist from the gist Discover page.

D. Users with assigned access can view the gist.

ANSWER: A

Explanation:

Anyone with the URL for the gist can view the gist. A secret gist is an unlisted GitHub Gist: it is not shown in public search results or on the Gist Discover page, but it is not access-controlled or private in the way that a private repository is. The direct URL acts as the means by which the gist is shared, so a person does not need to be individually granted permission in order to open it. This makes secret gists useful for sharing a code snippet, configuration example, or other text with a limited audience when discoverability is not desired. However, the URL should be treated as shareable access information: anyone the recipient forwards it to can also view the content. A secret gist should therefore not be used to store passwords, API keys, tokens, personal data, or other sensitive information. GitHub also notes that secret gists can be made public later,

while public gists cannot be changed back to secret. For the current definition of public and secret gists and their visibility behavior, see GitHub's Creating gists documentation and the About gists documentation.

QUESTION NO: 5

How can a user create a repository template, and what permissions are required?

A. With Admin permissions, navigate to Repository settings and select Template Repository.

B. With Maintain permissions, navigate to Organization settings, select the repository, and choose Template Repository.

C. With Admin permissions, navigate to Organization settings, select the repository, and choose Template Repository.

D. With Maintain permissions, navigate to Repository settings and select Template Repository.

ANSWER: A

Explanation:

With Admin permissions, navigate to Repository settings and select Template Repository. GitHub requires administrator access to a repository to designate it as a template because this setting changes a repository-level configuration. From the repository's **Settings** page, under the general settings, an administrator can enable the **Template repository** setting and save the change. Once enabled, the repository can be used as a starting point when someone selects **Use this template** to generate a new repository. A repository created from a template begins with the template repository's files and directory structure; GitHub also provides choices for whether all branches are included when creating the new repository. Template repositories are useful for standardizing starter code, project layouts, issue configuration, documentation, and other reusable project assets across teams. The person generating a repository from the template must separately have permission to create a repository in the selected owner account or organization. GitHub's documented procedure identifies repository administrators as the users who can make a repository a template and places this control in the repository settings, as described in Creating a template repository. See also GitHub's overview of creating a repository from a template.

QUESTION NO: 6

Which of the following are advantages of saved replies?

(Each correct answer presents part of the solution. Choose two.)

A. Saved replies are tied to a GitHub user's personal account.

B. Saved replies are allocated at the enterprise level for all users.

C. Saved replies allow you to create a reusable response to issues and pull requests.

D. Saved replies will send auto notifications when a user is tagged to an issue.

ANSWER: A C

Explanation:

Saved replies are personal, reusable comment templates that help people respond efficiently and consistently when working with issues and pull requests. "Saved replies are tied to a GitHub user's personal account" is correct because GitHub stores saved replies in the individual user's settings. This makes the replies available to that user wherever GitHub provides the saved-reply feature, rather than limiting the templates to one particular repository. A user can create, edit, delete, and insert their own saved replies as needed.

"Saved replies allow you to create a reusable response to issues and pull requests" is also correct because reuse is the core purpose of this feature. A saved reply can contain commonly repeated text, including Markdown, and can be inserted into a comment while responding to discussions. This reduces repetitive typing and supports more consistent communication for recurring situations such as requesting additional information, thanking contributors, or acknowledging a report. GitHub documents that saved replies are created and managed from a user's saved replies settings, then inserted when writing comments on issues and pull requests. See Using saved replies and About saved replies.

QUESTION NO: 7

In GitHub, why is it recommended to deploy from your feature branch before merging into the main branch?

A. To directly deploy changes from the main branch without any intermediate testing

B. To speed up the process of merging changes into the main branch

C. To avoid the need for testing changes in production

D. To ensure the changes are verified and validated in a production environment

ANSWER: D

Explanation:

To ensure the changes are verified and validated in a production environment is correct because deploying a feature branch to an environment before it reaches the main branch provides a final, realistic validation step. Teams commonly use preview or staging deployments associated with pull requests to test the exact code proposed for merging. This lets developers and reviewers confirm that the feature works correctly when built, configured, and run as an integrated application, rather than only in a local development setup. It can also expose deployment-specific issues, such as incorrect environment configuration, missing dependencies, or unexpected behavior when the feature interacts with external services.

GitHub Actions supports deployment workflows and environments, allowing a workflow to deploy a particular branch or pull request and record the deployment status. GitHub also provides deployment protection rules, which can require review or successful checks before a deployment proceeds. Validating the feature branch in an appropriate pre-production or production-like environment before merge helps preserve the reliability of the main branch and gives the team confidence that the merged change is ready to release. See GitHub's documentation on using environments for deployment and deploying with GitHub Actions.