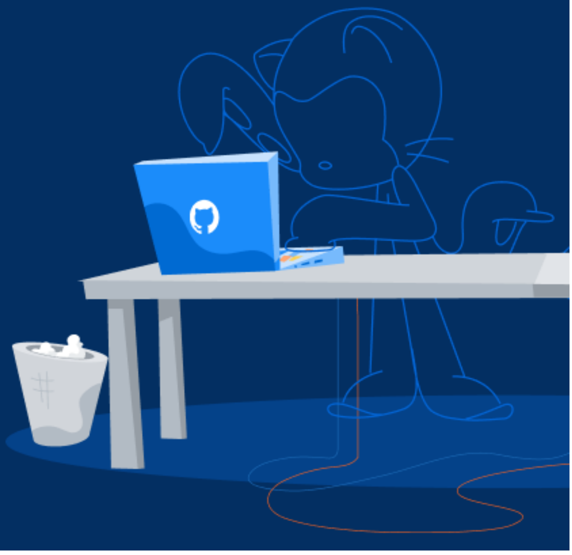
GitHub

Exam GH-100 GitHub Certified Practitioner Exam



Audience Profile

This exam is designed for candidates looking to demonstrate foundational knowledge on the considerations, benefits, and options of GitHub software as a service. This exam also covers basic knowledge of Git concepts and is intended for candidates with both technical (i.e., developers, engineers) or non-technical backgrounds (i.e., project managers, documentation specialists).

Exam content

Multiple choice

Has one correct response and three incorrect responses (distractors)

Multiple response

Has two or more correct responses out of four or more options

Skills and knowledge measured

The exam validates an examinee's ability to:

- Detail and understand foundational Git and GitHub concepts
- Describe the benefits of the GitHub community
- Describe and understand the concepts and components of software development on GitHub
- Describe and identify GitHub distribution and GitHub consumption models
- Describe and understand GitHub security, compliance, privacy, and trust options

Recommended GitHub knowledge

Basic working knowledge of GitHub is required

Recommended General IT knowledge

Candidates should have a basic understanding of IT services and their uses including basic understanding of software development concepts

Objective Domains (functional groups)

The domains listed below are intended to illustrate how we are assessing the skills, knowledge, and abilities. It is not an exhaustive list and is not definitive. Percentages allocates to each domain are approximations.

Domain	% of Examination
Domain 1: Describe foundational Git and GitHub concepts	35%
Domain 2: Describe the benefits of the GitHub community	15%
Domain 3: Describe the components of modern software development practices on GitHub	25%
Domain 4: Describe GitHub distribution and consumption models	5%
Domain 5: Describe security, compliance, privacy, and trust options in GitHub	20%

Domain 1 (35%): Describe foundational Git and GitHub concepts

Describe foundational Git commands and Git version control concepts

- Describe what Git is
- Describe basic Git features like repositories, clones, branches, commits, and remotes
- Describe the different types of version control
- Describe how to solve merge conflicts

Describe core GitHub concepts and features

- Describe what GitHub is
- Describe features of a GitHub repository, such as issues and pull requests
- Describe GitHub organizations and teams
- Describe the GitHub branching workflow
- Describe how code gets pushed to a GitHub repository
- Describe how code can be locally retrieved from a GitHub repository
- Define Markdown
- Describe how Markdown is used across GitHub
- Locate endpoints of the GitHub API

Domain 2 (15%): Describe the benefits of the GitHub community

Describe the benefits of open source software

- Define open source
- Describe the implications/parameters of assigning common open source licenses for your project
- Identify how to sponsor a project

Describe how to contribute to an open source repository in GitHub

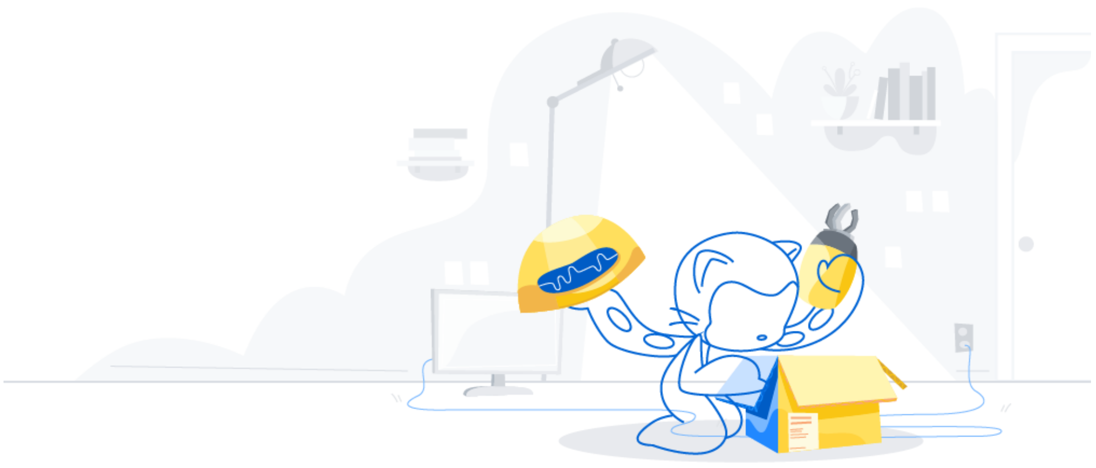
- Describe where to find open source projects
- Identify how to contribute to an open source repository
- Describe how to communicate with project maintainers

Describe how to apply the benefits of open source within private organizations

- Describe the various files used for open source documentation
- Define InnerSource
- Describe tenets of InnerSource communities
- Describe the components of a good contributing guideline
- Describe a discoverable repository
- Describe when to use issue templates and pull request templates

Identify how to use community-created apps, workflows, and actions in a project

- Define GitHub Apps, their use cases, and where to find them
- Describe how to find a community-created action in a repository
- Describe how to find a GitHub Actions workflow within a repository



Domain 3 (25%): Describe the components of modern software development practices on GitHub

Describe code review principles

- Define a pull request and its benefits
- Describe how to create a pull request
- Describe how to communicate on pull requests
- Define a pull request review and types of pull request reviews (comment, approve, request changes)

Describe the code-to-cloud process using GitHub

- Describe GitHub Actions
- Define components of GitHub Actions workflows, including events that trigger workflows, workflow files, workflow runs, and jobs
- Define an action
- Describe GitHub packages and when to use them
- Define a runner and types of runners available for GitHub Actions
- Describe how to use encrypted secrets

Describe Continuous Integration (CI)

- Define continuous integration (CI) and its benefits
- Define continuous delivery (CD) and its benefits

Domain 4 (5%): Describe GitHub distribution and consumption models

Describe how GitHub is deployed and distributed

- Describe GitHub Enterprise Cloud
- Describe GitHub Enterprise Server
- Describe GitHub Private Instances

Describe pricing, support and privacy options for GitHub products

- Describe pricing for GitHub products, including GitHub Actions
- Describe privacy expectations for GitHub products like personal repositories, organizations, and team membership

Domain 5 (20%): Describe security, compliance, privacy, and trust options in GitHub

Define code review controls and approval processes

- Choose when to use protected branches
- Define required pull request reviews
- Describe required status checks
- Describe code owners

Describe access control

- Describe multi-factor authentication (MFA)
- Describe repository collaborators
- Describe organization and team membership
- Define enterprise account

Describe security and compliance concepts with GitHub

- Define the use case for audit logs
- Describe allowed IP list

Describe identify protection and management options

- Describe how an identity provider can manage the identities of GitHub users and applications (through SSO and SAML)
- Describe how SSH keys are used for accessing GitHub repositories
- Describe personal access tokens (PAT)

Describe scrubbing of sensitive data from GitHub repositories

- Describe how to remove sensitive data from a Git repository (for example, git filter-branch or BFG repo cleaner)
- Describe the steps required to remove sensitive data from GitHub

Describe the security features of a GitHub repository

- Describe how to detect and fix outdated dependencies with security vulnerabilities
- Describe security vulnerability alerts
- Describe automated code scanning
- Describe automated security updates
- Describe secret scanning
- Describe a security policy