GitLab Certified Associate Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Table of Contents

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- **Practice answering questions under realistic conditions,**
- **Improve accuracy and speed,**
- **Review explanations to strengthen weak areas, and**
- **Approach the exam with greater confidence.**

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

1. What does the Discussions tab in GitLab allow users to do?

A. View and create pull requests

B. View project settings

C. Make comments inline

D. Manage branch permissions

2. Which feature of GitLab Review Apps facilitates collaboration?

A. Automatic Live Preview

B. Cluster management

C. Package auditing

D. Container orchestration

3. What types of environments can GitLab Review Apps be deployed to?

A. Only Kubernetes

B. Only local servers

C. Kubernetes, Heroku, FTP, and more

D. Only cloud-based platforms

4. What does the integration of the Dependency Proxy add for developers?

A. Enhanced billing features

B. A caching proxy with security

C. Increased server uptime

D. Simplified user interface

5. When using Go and common, which package is suggested for configuring the formatter for Logrus?

A. logutil package from Logrus

B. common's logutil package

C. GoLogger

D. LoggerConfig

6. What is the primary function of issues in GitLab?

A. To manage user permissions

B. To track tasks, enhancements, and bugs

C. To host static websites

D. To connect to external services

7. What is the purpose of the .gitlab-ci.yml file in GitLab CI/CD?

A. It serves as a configuration guide for GitLab Runner.

B. It defines the pipeline stages, jobs, and actions to perform.

C. It contains all project documentation and guidelines.

D. It is used to deploy applications to Kubernetes clusters.

8. Why should the before_script not be used in security job definitions?

A. It may complicate the CI/CD process

B. It can lead to poor project organization

C. Users may rely on it to prepare their projects for scanning

D. It is not supported in GitLab's environment

9. What is a primary benefit of using Canary Deployments in GitLab?

A. They allow for complete application downtime.

B. They enable deploying changes to a small group of users first.

C. They offer version control for Docker images.

D. They simplify user management.

10. What is a Group in GitLab?

A. A single project repository

B. A system for managing user accounts

C. A collection of multiple projects

D. A defined workflow process

Answers

1. C
2. A
3. C
4. B
5. B
6. B
7. B
8. C
9. B
10. C

Explanations

1. What does the Discussions tab in GitLab allow users to do?

A. View and create pull requests

B. View project settings

C. Make comments inline

D. Manage branch permissions

The Discussions tab in GitLab is specifically designed to facilitate communication and collaboration among users on a specific topic related to the project. This feature allows users to make comments inline with the code, documentation, or other project-related content. By enabling discussions within the context of specific lines or sections of the project, team members can provide feedback, ask questions, or share insights directly where they are most relevant. This inline commenting fosters a more cohesive review process and helps maintain a clear and organized conversation about changes or aspects of the project. The other choices focus on features that do not pertain to discussions. Viewing and creating pull requests, for instance, relates to code management rather than dialogue. Project settings management addresses configuration options for the project and is unrelated to discussions. Managing branch permissions deals with access control and security rather than collaborative discussions. Each of these activities serves distinct functions within the GitLab environment, separate from the discussion capabilities.

2. Which feature of GitLab Review Apps facilitates collaboration?

A. Automatic Live Preview

B. Cluster management

C. Package auditing

D. Container orchestration

The feature that facilitates collaboration among team members in GitLab Review Apps is the Automatic Live Preview. When developers create a merge request, Automatic Live Preview allows stakeholders to view and interact with the application as it would appear in production. This immediate visibility helps teams gather feedback quicker and more effectively, enabling better collaboration during the review process. With Automatic Live Preview, reviewers can see the actual changes made in the code without needing to run the application in their local environments. This helps eliminate discrepancies between what the developer sees and what the reviewer experiences. By providing a shared view of the application at various stages of development, it enhances discussions around specific features or bugs, making the review process more efficient and focused. In contrast, cluster management, package auditing, and container orchestration serve other important functions within DevOps workflows, but they do not directly enhance collaboration in the same way that Automatic Live Preview does. Cluster management relates to handling groups of servers, package auditing focuses on security through managing dependencies, and container orchestration pertains to automated management of containerized applications. While these are vital for software deployment and maintenance, they do not specifically enhance real-time collaboration among team members during the review process.

3. What types of environments can GitLab Review Apps be deployed to?

A. Only Kubernetes

B. Only local servers

C. Kubernetes, Heroku, FTP, and more

D. Only cloud-based platforms

GitLab Review Apps provide the functionality to deploy applications to various environments that facilitate testing and validation of code changes in real-time before merging them into the main branch. This capability allows teams to review changes in an isolated environment, which is essential for Continuous Integration/Continuous Deployment (CI/CD) workflows. The correct choice indicates that Review Apps can be deployed not just to Kubernetes, but also to other platforms like Heroku, FTP, and more. This versatility supports a wide range of deployment strategies across different types of infrastructures, whether they are cloud-based, on-premises, or hybrid. By supporting multiple deployment options, GitLab allows teams to choose the environment that best fits their needs and existing infrastructure. Kubernetes is a popular choice due to its orchestration capabilities and scalability, but limiting deployments to only this platform would restrict the flexibility needed in diverse development environments. Similarly, local servers and cloud-only options would not cover the full range of possible deployment scenarios, making the chosen response the most comprehensive and accurate in reflecting GitLab Review Apps' capabilities.

4. What does the integration of the Dependency Proxy add for developers?

A. Enhanced billing features

B. A caching proxy with security

C. Increased server uptime

D. Simplified user interface

The integration of the Dependency Proxy provides a caching proxy specifically designed for managing dependencies in a more secure and efficient manner. This allows developers to cache and deliver dependencies from popular package registries directly in their GitLab environment. The primary benefit here is that it enhances security by allowing teams to utilize dependencies without exposing their systems to public repositories, reducing the risk of malicious code. Additionally, caching improvements lead to increased speed when dependencies are required for builds, as repeated access to the same resources in a secure environment prevents the need to redownload them, thus optimizing the development workflow. This feature is particularly beneficial in continuous integration and continuous deployment (CI/CD) scenarios, where build times can be significantly reduced due to the effective handling of dependencies. In contrast, the other options do not accurately reflect the primary advantages offered by the Dependency Proxy. Enhanced billing features, increased server uptime, and a simplified user interface do not align with the core capabilities and purpose of the Dependency Proxy within the GitLab ecosystem.

5. When using Go and common, which package is suggested for configuring the formatter for Logrus?

A. logutil package from Logrus

B. common's logutil package

C. GoLogger

D. LoggerConfig

The common's logutil package is suggested for configuring the formatter for Logrus because it is specifically designed to facilitate logging functionalities within the Go ecosystem while leveraging Logrus's powerful logging features. This package typically includes pre-defined configuration options, making it easier for developers to set up and customize their logging without diving deep into the Logrus library's intricacies. Using the common's logutil package not only streamlines logging configuration but also ensures consistency across various projects or components that may utilize logging functionalities. It serves as a wrapper that provides a more user-friendly interface for configuring log formats, log levels, and other logging behaviors, which enhances both readability and maintainability of the code. In contrast, the other choices may not provide the same level of ease or direct integration with Logrus. For instance, while the logutil package from Logrus might seem like a viable option, it does not have the same level of abstraction or configurability that the common's logutil package offers. Similarly, GoLogger and LoggerConfig do not align directly with Logrus's recommended practices or integrations, which are essential for ensuring reliable logging implementations in Go applications.

6. What is the primary function of issues in GitLab?

A. To manage user permissions

B. To track tasks, enhancements, and bugs

C. To host static websites

D. To connect to external services

The primary function of issues in GitLab is to track tasks, enhancements, and bugs. Issues serve as a crucial part of project management within the platform, allowing teams to document and prioritize work effectively. Each issue can contain details such as descriptions, labels, due dates, and comments, facilitating collaboration among team members. Using issues, teams can create a clear workflow for managing project tasks, whether they are new features, ongoing improvements, or bugs that need fixing. This organized approach not only helps in tracking progress but also enhances communication among developers by providing a centralized place for discussions related to specific tasks. By focusing on task management, issues help teams maintain clarity and improve overall productivity throughout the development lifecycle.

7. What is the purpose of the .gitlab-ci.yml file in GitLab CI/CD?

A. It serves as a configuration guide for GitLab Runner.

B. It defines the pipeline stages, jobs, and actions to perform.

C. It contains all project documentation and guidelines.

D. It is used to deploy applications to Kubernetes clusters.

The .gitlab-ci.yml file plays a crucial role in GitLab CI/CD by defining the pipeline stages, jobs, and the specific actions to execute during the CI/CD process. This file acts as the blueprint for automation in the continuous integration and continuous deployment workflows. Within it, you can specify single or multiple jobs that detail how to build, test, and deploy your applications, as well as configure various stages in the pipeline, such as build, test, and deploy. By structuring the CI/CD pipeline in this manner, teams can automate their development processes efficiently, ensuring that code changes are tested and deployed consistently and reliably. This not only enhances the quality of the code but also speeds up the release cycles, making it essential for DevOps practices. Other choices refer to different aspects: while the GitLab Runner does rely on configurations, it is not specifically outlined in the .gitlab-ci.yml. Project documentation can be maintained separately and is not the primary purpose of this file, and deploying applications to Kubernetes is a specific action that can be defined within the .gitlab-ci.yml but is not the overarching purpose of the file itself.

8. Why should the before_script not be used in security job definitions?

A. It may complicate the CI/CD process

B. It can lead to poor project organization

C. Users may rely on it to prepare their projects for scanning

D. It is not supported in GitLab's environment

The rationale behind avoiding the use of before_script in security job definitions centers on the potential for users to depend on it for preparing their projects for scanning. The before_script is typically designed for setting up environments or installing dependencies that are required for running the main job scripts in a CI/CD pipeline. However, for security jobs specifically, relying on before_script can create issues, as it might lead to inconsistency in security posture and scanning processes. Security jobs need to execute standardized scans that reflect the actual project state without the influence of additional setup processes that may vary between runs. If users start depending on before_script to configure or prepare their code for scanning, this could yield unreliable results. It may imply varying environments or executions, ultimately leading to a false sense of security regarding their code's safety. In contrast, security jobs should be executed in a clean and controlled environment to ensure that they accurately reflect the project's security status without any pre-processing that could alter the results. This reinforces the premise that reliance on before_script in this context should be avoided to maintain the integrity and reliability of security scans.

9. What is a primary benefit of using Canary Deployments in GitLab?

A. They allow for complete application downtime.

B. They enable deploying changes to a small group of users first.

C. They offer version control for Docker images.

D. They simplify user management.

Using Canary Deployments in GitLab offers the significant benefit of enabling changes to be deployed to a small group of users initially. This approach allows teams to monitor performance and user feedback before rolling out the changes to a larger audience. By exposing only a subset of users to the new features or updates at first, teams can mitigate risks associated with deploying untested code in a live environment. This staged rollout gives developers the opportunity to catch and address any unforeseen issues or bugs, and it can help in gathering valuable insights on how the new changes are performing. This process is essential in maintaining application stability and ensuring a smooth transition for all users. In contrast, the other options do not accurately describe the benefits of Canary Deployments. Complete application downtime would be detrimental to user experience and service availability; version control for Docker images relates more to the management of containerized applications rather than deployment strategies; and simplifying user management is not a core feature associated with the concept of Canary Deployments. Hence, the focus on targeted user experiences makes Canary Deployments particularly valuable in modern development practices.

10. What is a Group in GitLab?

A. A single project repository

B. A system for managing user accounts

C. A collection of multiple projects

D. A defined workflow process

A Group in GitLab refers to a collection of multiple projects. This organizational structure allows users to group related projects together, enabling better management and collaboration. By creating a Group, teams can manage permissions and access controls for all projects within that Group simultaneously, rather than setting them individually for each project. This grouping can simplify workflows when projects share common goals or need to collaborate closely, making it easier for team members to navigate and contribute to their related projects. Additionally, any CI/CD settings, issues, boards, and milestones set at the Group level can apply to all included projects, enhancing consistency across the projects contained within the Group. This understanding helps distinguish Groups from other entities in GitLab, such as projects, which are individual repositories, or workflows, which define processes but do not encompass multiple projects as Groups do. As a result, recognizing a Group as a collective of projects highlights its role in organizational efficiency and collaboration in GitLab.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://gitlabcertifiedassoc.examzify.com

We wish you the very best on your exam journey. You've got this!