



Accelerate Your Certification with Linux Foundation CGOA Practice Questions

Last chance to prepare smart! Get your hands on free Linux Foundation Certified GitOps Associate Exam PDF questions. Study real CGOA dumps with verified answers and fast-track your certification success with [PrepBolt](https://prepbolt.com) CGOA exam pdf questions and answers.

Thank you for Downloading CGOA exam PDF Demo

<https://prepbolt.com/CGOA.html>

QUESTIONS & ANSWERS
DEMO VERSION
(LIMITED CONTENT)

Question 1

Question Type: MultipleChoice

What is the main difference between Terraform/OpenTofu and Ansible?

Options:

- A- Terraform/OpenTofu uses a configuration language called CUE, while Ansible uses HCL.
- B- Terraform/OpenTofu stores the state of each resource, while Ansible works in a fire-and-forget mode.
- C- Terraform/OpenTofu is imperative in nature, while Ansible is declarative.
- D- Ansible is written in Golang, while Terraform/OpenTofu is written in Python.

Answer:

B

Explanation:

Terraform (or OpenTofu) uses a declarative model and maintains a state file to track the current status of resources, enabling it to plan and reconcile changes. Ansible, by contrast, is more procedural and executes tasks in a fire-and-forget manner, without tracking persistent resource state.

"Terraform maintains state for each managed resource, enabling planned, consistent changes. Ansible executes tasks without tracking resource state, working in a fire-and-forget model."

Thus, the correct answer is B.

=====

Question 2

Question Type: MultipleChoice

Can you choose one example where Configuration as Code may be utilized to manage an application's configuration and source code?

Options:

- A- Using a manual process of editing configuration files and manually syncing the source code of a monolithic application.
- B- Using a spreadsheet to manually update and manage the configuration and source code of a mobile application.
- C- Using a Helm chart to define and manage the configuration and container image of a web application deployed on Kubernetes.
- D- Using a GUI-based configuration tool to visually configure and manage the source code of a microservices architecture.

Answer:

C

Explanation:

Configuration as Code is a GitOps-related practice where configurations are stored as declarative definitions in version control. Helm charts, for example, allow applications deployed on Kubernetes to have both their container images and configuration specified declaratively.

"Configuration as Code enables teams to manage application and infrastructure configuration in version control systems, using declarative approaches such as Kubernetes manifests or Helm charts. This ensures repeatability, automation, and auditability."

Thus, Helm charts are a prime example of this practice, making C correct.

=====

Question 3

Question Type: MultipleChoice

Would you implement DevSecOps culture in CI/CD pipelines?

Options:

- A- No, because CI/CD systems are totally secure by design.
- B- No, DevSecOps focuses only on security outside of CI/CD pipelines.
- C- No, because CI/CD systems are incompatible with the DevSecOps culture.
- D- Yes, DevSecOps is essential for CI/CD pipelines. It helps build software without any security issues.

Answer:

D

Explanation:

DevSecOps integrates security into the DevOps culture and CI/CD pipelines. Instead of treating security as an afterthought, DevSecOps ensures security checks, policies, and automated validations are embedded throughout the CI/CD process.

"DevSecOps emphasizes shifting security left, integrating security testing, validation, and compliance into every stage of the CI/CD pipeline. This ensures vulnerabilities are detected early and software is delivered securely."

Thus, D is correct: DevSecOps culture is essential for CI/CD pipelines to ensure security is baked into software delivery.

=====

Question 4

Question Type: MultipleChoice

A GitOps-managed Software System includes which of the following?

Options:

- A- Hardware infrastructure used for hosting the software system.
- B- One or more runtime environments consisting of resources under management.
- C- One or more programming languages used for development.
- D- Operating systems used for hosting the software system.

Answer:

B

Explanation:

A GitOps-managed software system is defined as one or more runtime environments whose resources are managed declaratively via GitOps practices.

"A GitOps-managed software system includes one or more runtime environments, such as clusters, where resources are under management. The desired state of these resources is declared in Git and

reconciled continuously."

Thus, the correct option is B.

=====

Question 5

Question Type: MultipleChoice

When using Kustomize, how are resources, configurations, and customizations commonly organized?

Options:

- A- By specifying all resources inline in the customization file.
- B- In separate configuration files for each resource.
- C- In a single configuration file.
- D- Using a combination of folder directories and referenced folder/file paths.

Answer:

D

Explanation:

Kustomize is a GitOps tool for managing Kubernetes configurations declaratively. It uses a folder structure with configuration files and a kustomization.yaml file that references resources and overlays. This enables customization without modifying the base manifests.

"Kustomize allows customization of Kubernetes manifests by organizing resources in directories and referencing them through file paths in a kustomization file. This directory-based approach supports overlays, reusability, and modular configuration."

Thus, the correct answer is D.

=====

Question 6

Question Type: MultipleChoice

A GitOps project wants to leverage both ArgoCD and Flux for a deployment. Can ArgoCD and Flux be used in conjunction?

Options:

- A- ArgoCD and Flux cannot be used together as they have conflicting functionalities.
- B- If you modify their source code, ArgoCD and Flux can only be used together.
- C- ArgoCD and Flux can be used together, leveraging a drop-in extension for ArgoCD, ensuring that both reconciliation engines do not conflict.
- D- ArgoCD and Flux cannot be used together as they are designed for different types of deployments.

Answer:

C

Explanation:

ArgoCD and Flux are the two primary CNCF GitOps tools. While both are reconciliation engines, they can be used together carefully if configured properly to avoid conflicts. For example, Flux can be used to manage configuration sources, while ArgoCD handles application-level delivery. Extensions and integration points allow them to complement each other.

"ArgoCD and Flux implement the GitOps reconciliation principle. Though they provide overlapping functionality, they can be integrated by carefully managing their scope. For instance, Flux can manage sources and Helm charts, while ArgoCD handles higher-level deployments. Extensions exist to allow cooperation without conflict."

Thus, the correct answer is C.

=====

Question 7

Question Type: MultipleChoice

In GitOps, what is the process of ensuring the actual state of a system matches its Desired State called?

Options:

- A- Reconciliation
- B- Webhooks
- C- Monitoring
- D- Manual Intervention

Answer:

A

Explanation:

The process of keeping the actual state in sync with the desired state is called Reconciliation. GitOps controllers (e.g., ArgoCD, Flux) continuously reconcile system resources to match what is declared in Git.

"Reconciliation is the process by which agents compare the actual system state to the desired state and automatically make changes to converge them."

Thus, the correct answer is A: Reconciliation.

=====

Question 8

Question Type: MultipleChoice

You are implementing GitOps in your organization and have configured the Desired State of your applications in a Git repository. However, during the deployment process, you encounter an error in the configuration. What is the recommended action in this scenario?

Options:

- A- Continue to monitor the issue and proceed with the deployment, as it may not significantly impact the application.
- B- Raise a ticket with the development team to fix the error in the configuration file.
- C- Roll back the deployment to the previous working version while investigating the error in the configuration file.
- D- Make a call to the Kubernetes API with the correction.

Answer:

C

Explanation:

GitOps emphasizes immutability and auditability. If an error occurs in the configuration stored in Git, the system should be rolled back to the last known good state while the error is fixed. This preserves system reliability and aligns with the GitOps principle of rollback through version control.

"With Git as the source of truth, if an error is introduced, the system can be rolled back by reverting to a previous commit. This ensures stability while the faulty configuration is corrected."

Thus, the recommended action is C: Roll back to the previous working version.

=====

Thank You for trying CGOA PDF Demo

To try our CGOA practice exam software visit link below

<https://prepbolt.com/CGOA.html>

Start Your CGOA Preparation

Use Coupon “**SAVE50**” for extra 50% discount on the purchase of Practice Test Software. Test your CGOA preparation with actual exam questions.