

LinuxFoundation.CGOA.v2026-06-05.q23

| | |
|---|----------------------------|
| 試験コード: | CGOA |
| 試験名称: | Certified GitOps Associate |
| 認定資格: | Linux Foundation |
| 無料問題数: | 23 |
| バージョン: | v2026-06-05 |
| アクセス数: | 111 |
| ページビュー数: | 230 |
| https://www.jpnpdf.com/LinuxFoundation.CGOA.v2026-06-05.q23-mondaishu.html | |

最新問題: 1

GitOps 管理ソフトウェア システムには次のどれが含まれますか？

- A. ソフトウェア システムをホストするために使用されるハードウェア インフラストラクチャ。
- B. 管理対象のリソースで構成される 1 つ以上のランタイム環境。
- C. 開発に使用される 1 つ以上のプログラミング言語。
- D. ソフトウェア システムをホストするために使用されるオペレーティング システム。

Answer: B (メッセージを残す)

AGitOps 管理ソフトウェア システムは、リソースが GitOps プラクティスを通じて宣言的に管理される 1 つ以上のランタイム環境として定義されます。

GitOpsで管理されるソフトウェアシステムには、クラスターなどの1つ以上のランタイム環境が含まれており、そこでリソースが管理されます。これらのリソースの望ましい状態はGitで宣言され、継続的に調整されます。」したがって、正しい選択肢はBです。

参考資料:GitOps 用語集 (CNCF GitOps ワーキング グループ)。

最新問題: 2

GitOps の文脈で、DevSecOps の原則がどのように現れ、従来の DevOps ライフサイクルを強化したかを示す 1 つの例は何ですか。

- A. GitOps は、セキュリティ ポリシーのドリフトを検出することで DevSecOps エクスペリエンスを強化します。
- B. GitOps の DevSecOps は、主にデプロイメント後のセキュリティ監査に重点を置いています。
- C. GitOps は DevSecOps を使用して、各デプロイメント段階で手動のセキュリティチェックを実施します。
- D. GitOps では、DevSecOps によってセキュリティ タスクが分離され、セキュリティ チームにのみ割り当てられます。

Answer: A (メッセージを残す)

GitOpsでは、DevSecOpsはセキュリティポリシーをコードとして扱い、Gitに保存することで、セキュリティをGitOpsワークフローに統合します。これにより、セキュリティポリシーの逸脱を自動検出し、アプリケーションやインフラストラクチャのコードと同様に、構成ミスや違反を確実に修正できます。

GitOpsは、セキュリティポリシーをコードとして管理することでDevSecOpsを適用します。これにより、セキュリティ構成の逸脱を検出し、環境のコンプライアンスとセキュリティを維持できます。」したがって、正解はAです。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)、DevSecOps 統合。

最新問題: 3

クラスター内リコンサイラを使用するか、外部リコンサイラを使用するかを決定する際に、どのような要素を考慮する必要がありますか？

- A. Kubernetes のバージョンとネットワーク リソースの可用性。
- B. クラスターのサイズと調整ロジックの複雑さ。
- C. アプリケーションが記述されているプログラミング言語。
- D. 状態ストアの場所とレプリカの数。

Answer: B (メッセージを残す)

GitOpsでは、リコンサイラーが実際の状態と目的の状態が一致することを確認します。リコンサイラーはクラスター内 (in-cluster) またはクラスター外 (external) で実行できます。どちらを選択するかは、主に運用規模とリコンシリエーションロジックの複雑さによって決まります。

リコンサイラの配置を決定する際には、環境の規模、リコンサイラの運用上の複雑さ、パフォーマンス要件などの要素を評価する必要があります。クラスター内リコンサイラは単純な導入で一般的ですが、大規模または複雑なシステムでは外部リコンサイラが選択される場合があります。したがって、最も重要な考慮事項はクラスターのサイズとリコンサイラロジックの複雑さであり、Bが正しいものとなります。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)、GitOps リコンサイラ ガイドライン。

最新問題: 4

DevOps と GitOps の関係を最もよく表す記述は次のどれですか。

- A. DevOps と GitOps は、同じ概念と原則を説明するために使用される互換可能な用語です。
- B. GitOps は、現代の DevOps を実践するための一連の原則です。
- C. DevOps と GitOps は競合する方法論であり、組織はどちらか一方を選択する必要があります。
- D. DevOps と GitOps は互いに関係のない、完全に別の概念です。

Answer: B (メッセージを残す)

GitOps は DevOps の代替ではありません。Git を唯一の真実のソースとして使用し、継続的な調整を運用モデルとして使用する、DevOps の原則の進化と実用的な実装です。

GitOpsはDevOpsの基盤の上に構築され、その原則を実践するためのフレームワークを提供します。Gitと宣言的リコンシリエーションを活用してDevOpsの目標を実現します。」したがって、この関係を最もよく表すのはBです。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)。

最新問題: 5

GitOps のコンテキストでは、Continuous とはどのような意味ですか？

- A. 和解は継続しています。
- B. 調整はインスタンス化中にのみ行われます。
- C. 調整は瞬時に行われなければなりません。
- D. 調整は 1 回だけ行われます。

Answer: A (メッセージを残す)

GitOpsの4つのコア原則の一つは、システムが継続的に調整されなければならないということです。これは、調整が一度限りのプロセスやオンデマンドのプロセスではなく、バックグラウンドで継続的に行われ、実際のシステム状態が宣言された望ましい状態と常に一致することを意味します。

GitOpsでは継続的な調整が必要です。ソフトウェアエージェントは実際の状態と望ましい状態を継続的に比較し、差異を自動的に調整します。」したがって、正解はCです。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)、原則 4: 継続的に調整されます。

最新問題: 6

GitOps において、継続的な調整とはどのような意味ですか？

- A. 外部ソースからの最新の変更を Git リポジトリに定期的に更新します。
- B. Git リポジトリの定期的なバックアップを実行します。
- C. 必要に応じてシステムの状態を自動的に比較して調整します。
- D. システムを監視して不正な変更がないか確認し、変更を元に戻します。

Answer: (解答を表示する)

継続的な調整は、GitOpsのもう一つの核となる原則です。これは、ソフトウェアエージェント (オペレーターまたはコントローラー) がグループを実行し、ライブシステムを継続的に監視し、Gitで宣言された望ましい状態と比較することを意味します。乖離 (ドリフト) が見つかった場合、エージェントは宣言された構成と一致するようにシステムを自動的に調整します。

ソフトウェアエージェントは、実際のシステム状態を継続的に監視し、Gitで宣言された望ましい状態と比較します。相違が検出された場合、エージェントは自動的にその差異を調

整し、システムを元の状態に戻します。これにより、GitOpsの特徴である自動化、一貫性、自己修復が実現されます。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)、原則 4: 継続的に調整されます。

最新問題: 7

GitOps デプロイメントに取り組んでおり、複数の環境にまたがる Kubernetes リソースの構成を管理したいと考えています。Kustomize はどのように役立ちますか？

- A. Kustomize は、Terraform/OpenTofu を使用してインフラストラクチャ リソースをデプロイするためのツールです。
- B. Kustomize を使用すると、アプリケーションを Helm チャートとしてパッケージ化して配布できます。
- C. Kustomize は、グラフィカル ユーザー インターフェイス (GUI) を提供することで、Kubernetes リソース マニフェストの作成と管理を支援します。
- D. Kustomize は、パッチ適用を通じて Kubernetes リソース マニフェストをカスタマイズする方法を提供することで、Kubernetes リソース マニフェストの作成と管理を支援します。

Answer: D (メッセージを残す)

KustomizeはKubernetesネイティブの構成管理ツールで、元のYAMLファイルを変更することなくマニフェストをカスタマイズできます。オーバーレイとパッチを使用することで、さまざまな環境に合わせて構成を調整できます。

Kustomize は、パッチとオーバーレイを適用することで Kubernetes マニフェストをカスタマイズする宣言的な方法を提供します。

これにより、マニフェスト ファイルを重複させることなく複数の環境を管理できます。したがって、正解は D です。

参考資料:GitOps ツール (CNCF GitOps ワーキング グループ)、Kustomize。

最新問題: 8

組織でGitOpsを実装し、Gitリポジトリでアプリケーションの「望ましい状態」を設定しました。しかし、デプロイプロセス中に設定エラーが発生しています。このシナリオで推奨されるアクションは何ですか？

- A. アプリケーションに大きな影響を与えない可能性があるため、問題を監視し続けてデプロイメントを続行します。
- B. 開発チームにチケットを発行して、構成ファイルのエラーを修正します。
- C. 構成ファイル内のエラーを調査しながら、デプロイメントを以前の作業バージョンにロールバックします。
- D. 修正を加えて Kubernetes API を呼び出します。

Answer: C (メッセージを残す)

GitOpsは不変性と監査可能性を重視しています。Gitに保存された設定にエラーが発生した場合、エラーが修正されるまでの間、システムは最新の正常な状態にロールバックする必要があります。これによりシステムの信頼性が維持され、バージョン管理によるロールバックというGitOpsの原則にも合致しています。

Gitを信頼できる情報源として利用することで、エラーが発生した場合でも、以前のコミットに戻すことでシステムをロールバックできます。これにより、問題のある構成を修正する間の安定性が確保されます。したがって、推奨されるアクションはC: 以前の作業バージョンにロールバックすることです。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)。

最新問題: 9

GitOps プロジェクトで作業しており、次回の予定されているリコンサイルの前にリコンサイルプロセスをトリガーしたいと考えています。推奨される方法は何ですか？

- A. 調整プロセスの間隔時間を調整します。
- B. RBAC を使用して認証し、定期的に調整プロセスを実行する cron ジョブをスケジュールします。
- C. GitOps ツールの CLI コマンドを使用して、クラスター上で調整プロセスを開始するスクリプトを手動で実行します。
- D. Webhook を使用して、Git リポジトリ内のイベントまたは変更に基づいて調整プロセスをトリガーします。

Answer: D ([メッセージを残す](#))

GitOpsでは継続的に調整が行われますが、ツールによっては通常のポーリング間隔よりも早く調整をトリガーできる場合が多くあります。推奨される方法は、GitリポジトリからのWebhookを使用することです。WebhookはGitOpsコントローラーに変更を通知し、即時に調整をトリガーします。

調整ループは望ましい状態と実際の状態を継続的に比較しますが、バージョン管理イベントからのWebhookによって調整をより早くトリガーできるため、変更がタイムリーに適用されます。」したがって、正解はAです。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)、調整および Webhook。

最新問題: 10

GitOps は Desired State ストアにアクセスするどの方法をサポートしていますか？

- A. プッシュメソッドのみ
- B. レプリケーションのみ
- C. 押しても引いてもいない
- D. プルは必須であり、プルに加えてプッシュもオプションで使用できます。

Answer: ([解答を表示する](#))

GitOps では、クラスター内のエージェントが Git から必要な状態を自動的にプルするプルベースのアプローチがデフォルトモデルとして採用されています。ただし、プルベースの

リコンシリエーションを補完するために、プッシュベースのトリガー（例Webhook）をオプションで使用することも可能です。

GitOpsでは、自動化とセキュリティを確保するためにプルベースの調整が必要です。プッシュベースのトリガーはオプションでこれを補完できますが、プルモデルが基本です。」したがって、正解はDです。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)、調整モデル。

最新問題: 11

アプリケーションの構成とソースコードを管理するために、Configuration as Code を利用できる例を1つ選択できますか？

- A. 構成ファイルを編集し、モノリシックアプリケーションのソースコードを手動で同期する手動プロセスを使用します。
- B. スプレッドシートを使用して、モバイルアプリケーションの構成とソースコードを手動で更新および管理します。
- C. Helm チャートを使用して、Kubernetes にデプロイされた Web アプリケーションの構成とコンテナ イメージを定義および管理します。
- D. GUI ベースの構成ツールを使用して、マイクロサービス アーキテクチャのソースコードを視覚的に構成および管理します。

Answer: C (メッセージを残す)

構成をコードとして保存する (Configuration as Code) は、GitOps関連のプラクティスであり、構成をバージョン管理において宣言的な定義として保存します。例えば、Helmチャートを使用すると、Kubernetesにデプロイされたアプリケーションで、コンテナイメージと構成の両方を宣言的に指定できます。

Configuration as Codeにより、KubernetesマニフェストやHelmチャートなどの宣言型アプローチを使用して、チームはバージョン管理システムでアプリケーションとインフラストラクチャの構成を管理できます。これにより、再現性、自動化、監査可能性が確保されます。」したがって、Helmチャートはこのプラクティスの好例であり、Cを正しいものにします。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)、Configuration as Code。

最新問題: 12

GitOps 調整ループは何を保証しますか？

- A. 変更のみを適用しますが、目的の状態の一部であったリソースは削除しません。
- B. 望ましい状態がシステムに瞬時に適用されます。
- C. 現在のシステム状態が目的の状態から逸脱した場合、目的の状態がシステムに適用されます。
- D. マニフェストにエラーがある場合でも、可能な限り適用されるようにします。

Answer: (解答を表示する)

リコンシリエーションループはGitOpsの基本原則です。望ましい状態 (Gitに保存されている状態)と実際の状態 (システム内で実行されている状態)を継続的に比較します。乖離 (ドリフト)が検出されると、リコンサイラーはシステムを自動的に修正し、望ましい状態に合わせます。

調整ループにより、システムが宣言された望ましい状態に継続的に収束することが保証されます。

実際の状態が逸脱するたびに、ループはシステムを調整して目的の状態に合わせます。」したがって、正解はCです。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)。

最新問題: 13

GitOps におけるバージョン管理と不変の原則とはどういう意味ですか？

- A. 構成とインフラストラクチャに対するすべての変更は、実稼働環境で直接行う必要があります。
- B. すべてのソフトウェア バージョンは Git リポジトリに保存する必要があります。
- C. 構成とインフラストラクチャ コードはバージョン管理され、不変の成果物として扱われる必要があります。
- D. 構成とインフラストラクチャ コードは、運用環境で直接変更する必要があります。

Answer: C (メッセージを残す)

GitOpsの4つの基本原則の一つは、「バージョン管理」と「不変性」です。これは、システム全体の望ましい状態をバージョン管理されたGitリポジトリに保存する必要があることを意味します。すべての変更はコミットとして表現され、Gitの不変性により、システムの進化に関する信頼性が高く監査可能な履歴が保証されます。

望ましい状態はバージョン管理システムに保存されます。真実の記録は不変の履歴に保存され、変更は必要に応じて監査および元に戻すことができます。これにより、システムの構成がバージョン管理され、不変かつ追跡可能であることが保証されます。」したがって、構成とインフラストラクチャはバージョン管理され、不変でなければならず、本番環境で直接変更されてはなりません。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)、原則 2: 目的のシステム状態はバージョン管理され、不変として保存されます。

最新問題: 14

どの GitOps ツールにプッシュベースの調整モデルのオプションがありますか？

- A. アルゴCD
- B. フラグ
- C. フラックス
- D. Argoワークフロー

Answer: A (メッセージを残す)

ほとんどのGitOpsツール (Fluxなど)はプルベースのみです。しかし、ArgoCDはプルベースのリコンシリエーション (継続的なモニタリング経由)と、オプションでプッシュベースのモデル (WebhookまたはCIパイプライン経由で変更をトリガー)の両方をサポートしています。

ArgoCDは、コントローラーがリポジトリを監視するプルベースの調整と、Webhookによってトリガーされるオプションのプッシュベースの調整モードの両方をサポートしています。」したがって、正解はA: ArgoCDです。

参考資料:GitOps ツール (CNCF GitOps ワーキング グループ)、調整モデルに関する ArgoCD ドキュメント。

最新問題: 15

GitOps のコンテキストでは、Desired State」とは何を指しますか?

- A. システムまたはアプリケーションがとるべき状態。
- B. 変更が行われる前のシステムまたはアプリケーションの状態。
- C. システムまたはアプリケーションの現在の状態。
- D. すべての変更が行われた後のシステムまたはアプリケーションの状態。

Answer: A ([メッセージを残す](#))

望ましい状態とは、Gitに保存される宣言的な仕様であり、システムの外観と動作を定義します。これは、実際の状態を継続的に調整するための基準点となります。

望ましい状態とは、Gitに保存されるシステムの完全な宣言型仕様です。システムの構成方法を定義し、調整のための信頼できる情報源として機能します。」したがって、正解はAです。

参考資料:GitOps 用語集 (CNCF GitOps ワーキング グループ)。

最新問題: 16

GitOps の文脈では、状態ストアの目的は何ですか?

- A. GitOps デプロイメント中に一時的な状態データを保存するためのシステムを提供します。
- B. 望ましい状態宣言の変更可能なバージョンを格納するシステムを提供します。
- C. アプリケーションの現在の状態を保存するためのシステムを提供します。
- D. 望ましい状態宣言の不変バージョンを保存するためのシステムを提供します。

Answer: (解答を表示する)

GitOpsにおける状態ストアは、望ましい状態宣言が保存される記録システムです。完全な監査、トレーサビリティ、ロールバック機能を確認するため、状態ストアは不変かつバージョン管理されている必要があります。Git自体は最も一般的な状態ストアです。

望ましい状態は、バージョン管理され不変な状態ストアに保存する必要があります。これにより、追跡可能性が保証され、以前のコミットに戻すことで回復が可能になります。」したがって、正解はDです。

参考資料:GitOps 用語集 (CNCF GitOps ワーキング グループ)。

有効な **CGOA** 問題集は GoShiken.com が提供された合格しやすい CGOA 試験問題集！ GoShiken.com が最新の **CGOA** 試験問題集を提供しています。GoShiken.com CGOA 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CGOA 問題集をゲットする人はこちら: <https://www.goshiken.com/Linux-Foundation/CGOA-mondaishu.html> (**6230%OFF**問題集溶と正解付きで **30%w** 特別割引コード: **Freepdfdumps**)

最新問題: 17

自動的にプル」とは何を意味しますか？

- A. リレーショナル データベースへの GET 要求。
- B. 常に Git プルを参照します。
- C. 新しいコミットについてシステムに通知する Webhook。
- D. 状態ストアから目的の状態にアクセスします。

Answer: ([解答を表示する](#))

GitOpsの自動プル原則とは、ソフトウェアエージェントが状態ストア (Gitなど)に保存された「必要な状態」に継続的にアクセスする方法を指します。エージェントはリポジトリから状態を自動的にプルし、それに応じてシステムを調整します。

ソフトウェアエージェントは、信頼できる情報源 (状態ストア)から必要な状態宣言を自動的に取得し、それに合わせてシステムを継続的に調整します。」したがって、正解は D です。

参考資料:GitOps の原則 (CNCF GitOps ワーキング グループ)。

最新問題: 18

GitOps では、どのオプションが状態ストア管理を説明しますか？

- A. 状態情報をリレーショナル データベースに保存します。
- B. 状態情報を集中データベースに保存します。
- C. 分散ファイルシステムに状態情報を保存します。
- D. バージョン管理システムに状態情報を保存します。

Answer: ([解答を表示する](#))

状態ストアは通常、Gitなどのバージョン管理システム (VCS)を使用して実装されます。これにより、システムの状態が監査可能、バージョン管理され、不変であることが保証されます。データベースや分散ファイルシステムなどの他のシステムは、GitOpsの不変性とバージョン管理の要件を満たしていません。

GitOpsでは、望ましい状態をバージョン管理システムに保存する必要があります。これは状態ストアとして機能し、完全で不変かつ監査可能な変更履歴を提供します。」したがって、正解はD バージョン管理システムです。

参考資料:GitOps 用語集 (CNCF GitOps ワーキング グループ)。

最新問題: 19

Terraform/OpenTofu と Ansible の主な違いは何ですか？

- A. Terraform/OpenTofu は CUE と呼ばれる構成言語を使用しますが、Ansible は HCL を使用します。
- B. Terraform/OpenTofu は各リソースの状態を保存しますが、Ansible は fire-and-forget モードで動作します。
- C. Terraform/OpenTofu は本質的に命令型ですが、Ansible は宣言型です。
- D. Ansible は Golang で書かれています、Terraform/OpenTofu は Python で書かれています。

Answer: ([解答を表示する](#))

Terraform (または OpenTofu) は宣言型モデルを使用し、状態ファイルを維持してリソースの現在の状態を追跡し、変更を計画および調整できるようにします。対照的に、Ansible はより手順的であり、永続的なリソース状態を追跡せずに、タスクをファイア アンド フォーゲット方式で実行します。

「Terraform は管理対象リソースごとに状態を維持し、計画的かつ一貫性のある変更を可能にします。Ansible はリソースの状態を追跡せずにタスクを実行し、ファイア・アンド・フォーゲットモデルで動作します。」したがって、正解は B です。

参考資料:GitOps ツール (CNCF GitOps ワーキング グループ)。

最新問題: 20

和解にとってフィードバック ループが重要なのはなぜですか？

- A. 調整が必要かどうか、同期を部分的に行うか完全に行うかを決定します。
- B. 状態同期ログ情報を分析し、同期を実行します。
- C. 変更が検出された場合にアラートをトリガーし、イベントをログ集約サービスに記録します。
- D. フィードバック ループは調整にとって重要ではありません。

Answer: ([解答を表示する](#))

フィードバックループはGitOpsの調整において非常に重要です。システムの実際の状態を継続的に監視し、望ましい状態と比較します。このループは、調整が必要なタイミングと、完全同期か部分同期かを判断します。

「調整におけるフィードバックループは、実際の状態を継続的に監視します。調整が必要かどうかを判断し、宣言された望ましい状態に合わせるために部分同期を実行するか完全同期を実行するかを通知します。」したがって、正解はAです。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)、調整フィードバック ループ。

最新問題: 21

望ましい状態に宣言型構成を使用する利点はどれですか？

- A. 宣言型構成により、ローカルでコードをより効率的に実行し、実行中のシステムに必要な変更を加えることができます。
- B. 実際の状態を調整するために広く採用されているコミュニティ ツールを使用すると、カスタム命令型スクリプトを維持するよりも作業が少なくなります。
- C. 宣言型構成を使用すると、ステップバイステップのプロセスを通じてアプリケーションをガイドする動的なスクリプトを組み込むことができます。
- D. 宣言型構成を使用すると、カスタム コード内で複雑な if/else ロジックを指定できます。

Answer: B (メッセージを残す)

宣言的な構成は、システムがどのように見えるかを記述するものであり、それをどのように実現するかを記述するものではありません。これにより、標準的な調整ツール ArgoCD や Flux など) を使用してシステムを自動的に管理できるようになり、命令型スクリプトの作成と保守の負担が軽減されます。

宣言型構成により、特注のスクリプトではなく汎用調整ツールでシステムを管理できるようになり、運用オーバーヘッドが削減され、信頼性が向上します。」したがって、正解は B です。

参考資料: GitOps の原則 (CNCF GitOps ワーキング グループ)、宣言型システム。

最新問題: 22

Infrastructure as Code (IaC) とは何ですか？

- A. 物理ハードウェアのみを使用してインフラストラクチャリソースを管理するアプローチ
- B. グラフィカルユーザーインターフェースを通じてインフラストラクチャリソースを管理するための方法論
- C. コマンドラインを使用してインフラストラクチャリソースを管理する手動プロセス
- D. 機械可読な定義ファイルを通じてインフラストラクチャリソースを管理およびプロビジョニングするプログラミングアプローチ

Answer: D (メッセージを残す)

Infrastructure as Code (IaC) は、GitOps の基本的なプラクティスです。手動プロセスや GUI ではなく、宣言型の機械可読ファイルを通じてインフラストラクチャを管理およびプロビジョニングします。IaC は、環境全体にわたる一貫性、自動化、そして再現性を保証します。

Infrastructure as Code は、バージョン管理に保存されたコードファイルを通じてインフラストラクチャを定義および管理します。これにより、インフラストラクチャのプロビジョニングにおける自動化、再現性、および不変性が実現されます。」したがって、Dis は正解です。

参考資料: GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)。

最新問題: 23

GitOpsプロジェクトに取り組んでおり、設定ファイルが標準化された形式で記述されていることを確認したいと考えています。これを確実に行うための推奨される方法は何ですか？

- A. チームメンバーに、互いの構成ファイルのフォーマットの問題を手動で確認するよう依頼します。
- B. フォーマットの問題を無視し、構成ファイルの機能に重点を置きます。
- C. リンター ツールを使用して、構成ファイルのフォーマットの問題を自動的にチェックします。
- D. 各構成ファイルを手動で確認し、フォーマットの問題がないか確認します。

Answer: ([解答を表示する](#))

GitOps の実践において、設定ファイルの一貫性、標準化、品質を維持するためには、Lint ツールが不可欠です。自動化された Lint ツールは、YAML マニフェスト、Helm チャート、Kustomize オーバーレイのフォーマットやベストプラクティスを検証できます。

「リンターは、宣言型の設定ファイルにおけるフォーマットと品質ルールの適用を自動化します。標準化された構造を確保し、GitOpsワークフローにおけるエラーを削減します。」したがって、正解はCです。

参考資料:GitOps 関連のプラクティス (CNCF GitOps ワーキング グループ)。

Valid CGOA Dumps shared by GoShiken.com for Helping Passing CGOA Exam!
GoShiken.com now offer the **newest CGOA exam dumps**, the GoShiken.com CGOA exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com CGOA dumps with Test Engine here:
<https://www.goshiken.com/Linux-Foundation/CGOA-mondaishu.html> (62 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)