Google Cloud

Professional Cloud DevOps Engineer

認定試験ガイド

Professional Cloud DevOps Engineer は、Google 推奨の手法とツールを使用して、システム開発ライフサイクル全体にプロセスを実装します。信頼性と配信速度のバランスを取りながら、ソフトウェアとインフラストラクチャの効率的な配信を実現します。また、本番環境のシステムとサービスを最適化して保守します。

セクション 1: Google Cloud 組織のブートストラップと保守（試験内容の約 15%）

1.1 組織の全体的なリソース階層を設計する。以下のような点を考察します。

- プロジェクトとフォルダ
- 共有ネットワーク
- 複数のプロジェクトのモニタリングとロギング
- Identity and Access Management（IAM）のロールと組織レベルのポリシー
- サービス アカウントの作成と管理
- アプリケーション中心のアプローチを使用したリソースの整理（App Hub など）

1.2 インフラストラクチャを管理する。以下のような点を考察します。

- Infrastructure as Code ツール（Cloud Foundation Toolkit、Config Connector、Terraform、Helm など）
- Google 推奨の実践方法とブループリントを使用したインフラストラクチャの変更
- スクリプトを使用した自動化（Python、Go など）

1.3 Google Cloud、ハイブリッド環境、マルチクラウド環境で CI / CD アーキテクチャ スタックを設計する。以下のような点を考察します。

- Cloud Build を使用した継続的インテグレーション（CI）
- Kustomize と Skaffold を含む Cloud Deploy を使用した継続的デリバリー（CD）
- 広く使用されているサードパーティ製ツール（Jenkins、Git、Argo CD、Packer など）
- CI / CD ツールのセキュリティ

1.4 複数の環境を管理する（ステージング、本番環境など）。以下のような点を考察します。

- 環境の数とその目的の決定

- 一時的な環境の管理
- 構成とポリシーの管理
- エンタープライズ全体での Google Kubernetes Engine（GKE）クラスタの管理
- 安全なパッチ適用とアップグレードの手法

1.5 安全なクラウド開発環境を実現する。以下のような点を考察します。

- クラウド開発環境の構成と管理（Cloud Workstations、Cloud Shell など）
- 環境のブートストラップとそれに必要なツール（カスタムイメージ、IDE、Cloud SDK など）
- AI による開発と運用の支援（Cloud Code、Gemini Code Assist など）

セクション **2:** アプリケーションとインフラストラクチャの **CI / CD** パイプラインの構築および実装（試験内容の約 **27%**）

2.1 CI / CD パイプラインを設計、管理する。以下のような点を考察します。

- Artifact Registry を使用したアーティファクト管理
- ハイブリッド クラウドおよびマルチクラウド環境へのデプロイ（GKE Enterprise など）
- CI / CD パイプライン トリガー
- パイプラインでの新しいアプリケーション バージョンのテスト
- デプロイ プロセスの構成（承認フローなど）
- サーバーレス アプリケーションの CI / CD
- CI / CD 手法のインフラストラクチャへの適用（GKE クラスタ、マネージド インスタンス グループ、Cloud Service Mesh の構成など）

2.2 CI / CD パイプラインを実装する。以下のような点を考察します。

- デプロイの監査とトラッキング（Artifact Registry、Cloud Build、Cloud Deploy、Cloud Audit Logs など）
- デプロイ戦略（カナリア、Blue/Green、ローリング、トラフィック分割など）
- デプロイに関する問題のトラブルシューティングと軽減

2.3 CI / CD の構成とシークレットを管理する。以下のような点を考察します。

- 鍵管理（Cloud Key Management Service など）
- シークレット管理（Secret Manager、Certificate Manager など）
- ビルド時とランタイム時のシークレット挿入の比較

2.4 CI / CD デプロイ パイプラインを保護する。以下のような点を考察します。

- Artifact Registry を使用した脆弱性分析
- ソフトウェア サプライ チェーンのセキュリティ（Binary Authorization、ソフトウェア アーティファクトのためのサプライ チェーン レベル [SLSA] フレームワークなど）
- 環境に基づく IAM ポリシー

セクション **3:** サイト信頼性エンジニアリング手法のアプリケーションへの適用（試験内容の約 **23%**）

3.1 サービスの変更、速度、信頼性のバランスを取る。以下のような点を考察します。

- SLI（可用性、レイテンシなど）、SLO、SLA の定義
- エラー バジェット
- リスクと信頼性に関する機会費用（「9」の桁数など）

3.2 サービスのライフサイクルを管理する。以下のような点を考察します。

- サービスの管理（サービス前のオンボーディング チェックリストを使用した新しいサービスの導入、リリース計画、デプロイ計画、デプロイ、メンテナンス、提供終了など）
- キャパシティ プランニング（割り当て、上限など）
- 自動スケーリング（マネージド インスタンス グループ、Cloud Run、GKE など）

3.3 ユーザーに対するインシデントの影響を軽減する。以下のような点を考察します。

- トラフィックのドレイン / リダイレクト
- 容量の追加
- ロールバック戦略

セクション **4:** オブザーバビリティ手法の実装（試験内容の約 **20%**）

4.1. ログを管理する。以下のような点を考察します。

- ログの収集とインポート（Cloud Logging エージェント、Cloud Audit Logs、VPC フローログ、Cloud Service Mesh など）
- ロギングの最適化（フィルタ、サンプリング、除外、費用、ソースに関する考慮事項など）
- ログのエクスポート（監査のための BigQuery、Pub/Sub など）
- ログの保持

- ログの分析
- センシティブ データの処理（個人を特定できる情報 [PII]、保護医療情報 [PHI] など）

4.2 指標を管理する。以下のような点を考察します。

- 指標の収集と分析（アプリケーション、プラットフォーム、ネットワーキング、Cloud Service Mesh、Google Cloud Managed Service for Prometheus、ハイブリッド / マルチクラウドなど）
- ログからのカスタム指標の作成
- Metrics Explorer を使用したアドホック指標分析
- 合成モニターの作成

4.3 ダッシュボードとアラートを管理する。以下のような点を考察します。

- ダッシュボードの管理（作成、フィルタ、共有、ハンドブックなど）
- アラートおよびアラート ポリシーの構成（SLI、SLO、費用管理など）
- 広く使用されているサードパーティのアラート ツール

セクション **5:** パフォーマンスの最適化とトラブルシューティング（試験内容の約 **15%**）

5.1 問題のトラブルシューティングを行う。以下のような点を考察します。

- インフラストラクチャの問題
- アプリケーションの問題
- CI / CD パイプラインの問題
- オブザーバビリティの問題
- パフォーマンスとレイテンシの問題

5.2 Google Cloud にデバッグツールを実装する。以下のような点を考察します。

- アプリケーションへの計測手法の実装
- Cloud Trace
- Error Reporting

5.3 リソースの使用率と費用を最適化する。以下のような点を考察します。

- オブザーバビリティ費用
- Spot 仮想マシン（VM）
- インフラストラクチャの費用計画（確約利用割引、継続利用割引、ネットワーク ティアなど）
- Google Cloud Recommender（費用、セキュリティ、パフォーマンス、管理性、信頼性など）