

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。また、ソフトウェアとインフラストラクチャのデリバリーパイプラインを構築してデプロイし、本番環境のシステムとサービスを最適化して保守し、サービスの信頼性と配信速度の調整を行います。

セクション 1: DevOps のための Google Cloud 組織のブートストラップ(試験内容の 17% 以下)

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慮します。

- プロジェクトとフォルダ
- 共有ネットワーク
- Identity and Access Management (IAM) のロールと組織レベルのポリシー
- サービス アカウントを作成して管理する

1.2 インフラストラクチャをコードとして管理する。以下のような点を考慮します。

- Infrastructure as Code ツール (例: Cloud Foundation Toolkit、Config Connector、Terraform、Helm)
- Google 推奨のプラクティスと Infrastructure as Code ブループリントを使用してインフラストラクチャを変更する
- 不変のアーキテクチャ

1.3 Google Cloud、ハイブリッド環境、マルチクラウド環境での CI/CD アーキテクチャ スタックの設計。以下のような点を考慮します。

- Cloud Build を使用した CI
- Google Cloud Deploy を使用した CD
- 広く使用されているサードパーティ製ツール (例: Jenkins、Git、ArgoCD、Packer)
- CI / CD ツールのセキュリティ

1.4 複数の環境を管理する (例: ステージング、本番環境)。以下のような点を考慮します。

- 環境の数とその目的の決定

- Google Kubernetes Engine (GKE) と Terraform を使用して各機能ブランチ用の環境を動的に作成する
- Anthos Config Management

セクション 2: サービスの CI/CD パイプラインの構築と実装(試験内容の 23% 以下)

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慮します。

- Artifact Registry を使用したアーティファクト管理
- ハイブリッドおよびマルチクラウド環境へのデプロイ(例: Anthos、GKE)
- CI/CD パイプライントリガー
- パイプラインで新しいアプリケーション バージョンをテストする
- デプロイ プロセスの構成(例: 承認フロー)
- サーバーレス アプリケーションの CI/CD

2.2 CI/CD パイプラインを実装する。以下のような点を考慮します。

- デプロイの監査とトラッキング(例: Artifact Registry、Cloud Build、Google Cloud Deploy、Cloud Audit Logs)
- 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)
- ロールバック戦略
- デプロイに関する問題のトラブルシューティング

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慮します。

- 安全なストレージ方式と鍵のローテーション サービス(例: Cloud Key Management Service、Secret Manager)
- シークレット管理
- ビルドとランタイム シークレットの挿入

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慮します。

- Artifact Registry を使用した脆弱性分析
- Binary Authorization
- 環境ごとの 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の手法をサービスに適用する(試験内容の 23% 以下)

3.1 サービスの変更、速度、信頼性確保の調整。以下のような点を考慮します。

- SLI の検出(可用性、レイテンシなど)
- SLO の定義と SLA の理解
- エラー バジエット
- トイルの自動化
- リスクと信頼性に関する機会費用(例:「9」の数など)

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慮します。

- サービス管理(例: サービス前のオンボーディング チェックリスト、リリース計画、デプロイプラン、デプロイ、メンテナンス、使用停止を使用した新しいサービスの導入)
- キャパシティプランニング(例: 割り当てと上限の管理)
- マネージド インスタンス グループ、Cloud Run、Cloud Functions、GKE を使用した自動スケーリング
- サービスを改善するためのフィードバック ループの実装

3.3 運用のための健全なコミュニケーションとコラボレーションの確保。以下のような点を考慮します。

- 心身の疲労を防ぐ(例: 心身の疲労を防ぐための自動化プロセスを設定する)
- 学習の文化と責任を追及しない文化を育む
- サービスの共同所有を確立してチームのサイロを解消する

3.4 ユーザーに対するインシデントの影響の軽減。以下のような点を考慮します。

- インシデント時のコミュニケーション
- トラフィックをドレイン/リダイレクトする
- 容量の追加

3.5 事後分析の実施。以下のような点を考慮します。

- 根本原因の文書化
- アクション アイテムを作成して優先順位を付ける
- 事後調査の結果を関係者に伝える

セクション 4: サービス モニタリング戦略の実施(試験内容の 21% 以下)

4.1 ログの管理。以下のような点を考慮します。

- Cloud Logging を使用して、Compute Engine、GKE、サーバーレス プラットフォームから構造化ログと非構造化ログを収集する
- Cloud Logging エージェントの構成
- Google Cloud の外部からログを収集する
- Cloud Logging API にアプリケーション ログを直接送信する
- ログレベル (例: 情報、エラー、デバッグ、致命的)
- ログの最適化 (例: 複数行のロギング、例外、サイズ、コスト)

4.2 Cloud Monitoring を使用して指標を管理する。以下のような点を考慮します。

- アプリケーションとプラットフォームの指標の収集と分析
- ネットワーキングとサービス メッシュの指標の収集
- アドホック指標分析に Metric Explorer を使用する
- ログからのカスタム指標の作成

4.3 Cloud Monitoring で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慮します。

- Monitoring のダッシュボードを作成する
- ダッシュボードをフィルタおよび共有する
- アラートの構成
- SLO と SLI に基づくアラート ポリシーの定義
- Terraform を使用したアラート ポリシー定義の自動化
- Google Cloud Managed Service for Prometheus を使用して指標を収集し、モニタリングとアラートを設定する

4.4 Cloud Logging プラットフォームを管理する。以下のような点を考慮します。

- データアクセス ログを有効化する (例: Cloud Audit Logs)
- VPC フローログを有効化する
- Google Cloud Console にログを表示する
- 基本ログフィルタと高度なログフィルタを使用する
- ログの除外とログのエクスポート
- プロジェクトレベルと組織レベルのエクスポート
- ログのエクスポートの管理と表示
- 外部ロギング プラットフォームへログを送信する
- センシティブ データのフィルタリングと秘匿化 (例: 個人を特定できる情報 [PII]、保護医療情報 [PHI])

4.5 ロギングとモニタリングのアクセス制御を実装する。以下のような点を考慮します。

- Cloud Logging で監査ログと VPC フローログへのアクセスを制限する
- Cloud Logging によるエクスポート構成の制限
- Cloud Monitoring で指標とログの書き込みを許可する

セクション 5: サービス パフォーマンスの最適化 (試験内容の 16% 以下)

5.1 サービス パフォーマンスの問題を特定する。以下のような点を考慮します。

- Google Cloud のオペレーション スイートを使用してクラウド リソースの使用率を特定する
- サービス メッシュ テレメトリーの解釈
- コンピューティング リソースに関する問題のトラブルシューティング
- アプリケーションのデプロイ時間とランタイムの問題のトラブルシューティング
- ネットワークの問題のトラブルシューティング (例: VPC フローログ、ファイアウォール ログ、レイテンシ、ネットワークの詳細の表示など)

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慮します。

- アプリケーション インストルメンテーション
- Cloud Logging
- Cloud Trace
- Error Reporting
- Cloud Profiler
- Cloud Monitoring

5.3 リソース使用率と費用を最適化する。以下のような点を考慮します。

- プリエンプティブル/Spot 仮想マシン (VM)
- 確約利用割引 (例: フレキシブル、リソースベースなど)
- 継続利用割引
- ネットワーク ティア
- 適正サイズに関する推奨事項