



Implementing DevOps Solutions and Practices using Cisco Platforms v1.0 (300-910)

試験概要: Implementing DevOps Solutions and Practices using Cisco Platforms v1.0 (DEVOPS 300-910) は、DevNet Professional 認定に関する試験であり、試験時間は 90 分です。この試験では、シスコプラットフォーム上でのクラウド マイクロサービスおよびインフラストラクチャ プロセスの自動コンフィグレーション、管理、スケーラビリティに関連する DevOps のプラクティスに関する受験者の知識が問われます。本試験の受験対策として、Implementing DevOps Solutions and Practices using Cisco Platforms コースの受講をお勧めします。

次に、この試験の一般的な出題内容を示します。ただし、試験によっては、ここに示されていない関連項目も出題される場合があります。試験内容をより適切に反映し、明確にするために、次のガイドラインは予告なく変更されることがあります。

- 20%** **1.0** **CI/CD パイプライン**
 - 1.1 Jenkins、Drone、Travis CI などのビルドおよび展開ツールの特性と概念についての説明
 - 1.2 指定されたシナリオにおける CI/CD パイプラインを実装するシーケンス、コンポーネント、統合作業の特定
 - 1.3 CI/CD パイプラインを使用した問題のトラブルシューティング(コードベースのエラー、パイプラインの問題、ツールの互換性など)
 - 1.4 指定されたシナリオでの CI/CD パイプラインに統合するテストの特定
 - 1.5 指定されたシナリオでのリリース展開戦略の特定(カナリア、ロールバック、ブルー/グリーン)
 - 1.6 コードの依存関係の管理に関連する問題の診断(API、ツール チェーン、ライブラリなど)

- 15%** **2.0** **アプリケーションのパッケージングと配布**
 - 2.1 アプリケーションをコンテナ化するステップの特定
 - 2.2 複数のマイクロサービス アプリケーションを配備するステップの手順
 - 2.3 技術的やビジネス的要件に基づいたマイクロサービスおよびコンテナのアーキテクチャダイアグラムの評価(セキュリティ、性能、安定性、コスト)
 - 2.4 コンフィグレーション項目、アプリケーション パラメータ、機密データを安全に取り扱うためのプラクティスの特定
 - 2.5 アプリケーションの仕様に対応するための Docker ファイルの構築
 - 2.6 アプリケーションを配備するためのゴールデン イメージの使用についての説明

- 20%** **3.0** **自動化インフラストラクチャ**
 - 3.1 DevOps のプラクティスを既存の組織構造に統合する方法の説明
 - 3.2 Ansible、Puppet、Terraform、Chef などのインフラストラクチャ サービスを自動化するための

- コンフィグレーション管理ツールの使用の説明
 - 3.3 インフラストラクチャ サービスのアプリケーション配備を自動化する Ansible プレイブックの作成
 - 3.4 インフラストラクチャのアプリケーション配備を自動化する Terraform コンフィグレーションの作成
 - 3.5 コードとしてのインフラストラクチャのプラクティスおよび利点の説明
 - 3.6 指定されたシナリオでの CI/CD パイプラインのネットワーク状態の事前検証
 - 3.7 指定されたシナリオでの CI/CD パイプラインのアプリケーション インフラストラクチャの事前検証
 - 3.8 DevOps プラクティスの NetDevOps 用ネットワークへの拡張という概念についての説明
 - 3.9 アプリケーションまたはサービスの拡張に必要な要件(メモリ、ディスク I/O、ネットワーク、CPU など)の特定
- 15%**
- 4.0 クラウドおよびマルチクラウド**
 - 4.1 Kubernetes の概念およびオブジェクトの説明
 - 4.2 Kubernetes クラスタへのアプリケーションの配備
 - 4.3 Kubernetes オブジェクトを活用した配備のビルドによる要件への対応
 - 4.4 Drone コンフィグレーション ファイルの継続的配信用のためのパイプラインの解釈
 - 4.5 Kubernetes 上でアプリケーション配備が成功したかどうかの確認
 - 4.6 複数の環境にアプリケーションを配備する際の手法および考慮事項の説明(複数のクラウドプロバイダー、ハイ アベイラビリティ コンフィグレーション、ディザスタリカバリ コンフィグレーション、およびクラウドの可搬性のテストなど)
 - 4.7 パブリック クラウドを消費する際のコスト追跡および予測プロセスの説明
 - 4.8 パブリック クラウドの反復的消費におけるコードとしてのインフラストラクチャの利点の説明
 - 4.9 クラウド サービス戦略の比較(構築対購入)
- 20%**
- 5.0 ロギング、モニタリング、メトリック**
 - 5.1 性能の問題などアプリケーションのトラブルシューティングやテレメトリ ログ ストリーミングに役立つログおよびメトリックシステムの要素の特定
 - 5.2 アプリケーション用のログ収集およびレポーティング システムの実装
 - 5.2.a 複数の関連アプリケーションのログの集約
 - 5.2.b 検索機能
 - 5.2.c レポーティング機能
 - 5.3 AppDynamics およびアプリケーション パフォーマンス モニタリングを使用した分散アプリケーションのトラブルシューティング
 - 5.4 カオス エンジニアリングの原理についての説明
 - 5.5 API を使用して以下のタスクを実行する Python スクリプトの作成
 - 5.5.a モニタリング ダッシュボードのビルド
 - 5.5.b Webex Teams スペースの通知
 - 5.5.c アラートおよび違反に対する反応
 - 5.5.d 通知の作成
 - 5.5.e ヘルス チェック モニタリング
 - 5.5.f インシデントのオープンとクローズ
 - 5.6 アプリケーションのヘルスおよび性能に対する可視性を提供するために必要な付加的要件の特定

- 5.7 ログイング、モニタリング、メトリックに関連する Kubernetes の機能についての説明
- 5.8 CI/CD パイプライン設計におけるログイング、モニタリング、アラートイングの統合についての説明

- 10% 6.0 セキュリティ**
 - 6.1 CI/CD パイプラインの実稼動およびテスト中に、アプリケーションおよびインフラストラクチャのセキュリティを保護するための手法の特定
 - 6.2 セキュアなソフトウェア開発のライフサイクルを実装する手法の特定