



DevNet Associate Exam v1.0 (200-901)

試験概要: DevNet Associate Exam v1.0 (DEVASC 200-901) は、DevNet Associate - Developer 認定に関する試験であり、試験時間は 120 分です。この試験では、API の理解と使用、シスコプラットフォームと開発、アプリケーション開発とセキュリティ、およびインフラストラクチャと自動化など、ソフトウェア開発と設計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、Developing Applications and Automating Workflows using Cisco Core 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。

- 15% **1.0 ソフトウェア開発と設計**
 - 1.1 データフォーマットの比較 (XML、JSON、および YAML)
 - 1.2 一般的なデータ・フォーマット (XML、JSON、および YAML) の構文解析および Python データ構造への変換
 - 1.3 テスト駆動開発の概念
 - 1.4 ソフトウェア開発手法の比較 (アジャイル、リーン、ウォーターフォール)
 - 1.5 メソッドや関数、クラス、モジュールにコードを組織化するメリットの説明
 - 1.6 一般的な設計パターン (MVC および Observer) の利点
 - 1.7 バージョン管理の利点
 - 1.8 Git でよく使用されるバージョン管理操作の活用
 - 1.8.a クローン
 - 1.8.b 追加および削除
 - 1.8.c コミット
 - 1.8.d プッシュおよびプル
 - 1.8.e ブランチ
 - 1.8.f コンフリクトのマージおよび処理
 - 1.8.g diff

- 20% **2.0 API の理解と使用**
 - 2.1 API ドキュメントに指定されたタスクを遂行するための REST API 要求の作成
 - 2.2 ウェブフックに関連する一般的な使用パターンの説明
 - 2.3 API を消費する際の制約
 - 2.4 REST API に関連する HTTP の一般的な応答コード
 - 2.5 HTTP 応答コード、要求、および API ドキュメントで指定された問題のトラブルシューティング
 - 2.6 HTTP 応答 (応答コード、ヘッダー、ボディ)

- 2.7 一般的な API 認証メカニズムの利用 (Basic、カスタムトークン、API キー)
 - 2.8 一般的な API のスタイルの比較 (REST、RPC、同期、非同期)
 - 2.9 リクエスト ライブラリを使用して REST API をコールする Python スクリプトの作成
- 15%**
- 3.0 シスコ プラットフォームと開発**
 - 3.1 Cisco SDK を使用した Python スクリプトの作成 (SDK ドキュメントを利用可能)
 - 3.2 シスコのネットワーク管理プラットフォームおよび API の機能の説明 (Meraki、Cisco DNA Center、ACI、Cisco SD-WAN、および NSO)
 - 3.3 シスコのコンピューティング管理プラットフォームおよび API の機能の説明 (UCS Manager、UCS Director、および Intersight)
 - 3.4 シスコのコラボレーション プラットフォームおよび API の機能の説明 (Webex Teams、Webex デバイス、Cisco Unified Communication Manager (AXL/UDS インターフェイス、および Finesse を含む))
 - 3.5 シスコのセキュリティ プラットフォームおよび API の機能の説明 (Firepower、Umbrella、AMP、ISE、および ThreatGrid)
 - 3.6 IOS XE および NX-OS のデバイス レベル API および動的インターフェイスの説明
 - 3.7 指定されたシナリオに最適な DevNet リソースの特定 (Sandbox、Code Exchange、サポート、フォーラム、ラーニング ラボ、および API ドキュメント)
 - 3.8 モデル駆動型プログラマビリティの概念のシスコ環境での適用 (YANG、RESTCONF、および NETCONF)
 - 3.9 一連の要件および所定の API リファレンス ドキュメントに基づいた、以下の特定の操作を実行するコードの作成:
 - 3.9.a ネットワーク デバイスのリストの取得 (Meraki、Cisco DNA Center、ACI、Cisco SD-WAN、または NSO の使用)
 - 3.9.b Webex Teams でのスペース、参加者、メッセージの管理
 - 3.9.c ネットワーク上に見られるクライアントおよびホストのリストの取得 (Meraki または Cisco DNA Center を使用)
- 15%**
- 4.0 アプリケーションの展開とセキュリティ**
 - 4.1 エッジ コンピューティングのメリットの説明
 - 4.2 さまざまなアプリケーション展開モデルの属性 (プライベート クラウド、パブリック クラウド、ハイブリッド クラウド、エッジ)
 - 4.3 以下のアプリケーション展開モデルの属性
 - 4.3.a 仮想マシン
 - 4.3.b ベア メタル
 - 4.3.c コンテナ
 - 4.4 アプリケーション展開における CI/CD パイプラインの構成要素の説明
 - 4.5 ユニット テスト機能を使用する Python プログラムの作成
 - 4.6 Dockerfile の内容の解釈
 - 4.7 ローカルな開発環境における Docker イメージの利用
 - 4.8 機密保護、暗号化 (ストレージおよびトランスポート)、データ ハンドリングに関連するアプリケーション セキュリティの問題の特定
 - 4.9 アプリケーション展開におけるファイアウォール、ロード バランサ、リバース プロキシの機能の説明
 - 4.10 OWASP Top 10 の脅威についての説明 (XSS、SQL インジェクション、および CSRF)

- 4.11 Bash コマンドの利用(ファイル管理、ディレクトリのナビゲーション、環境変数)
- 4.12 DevOps の実践原則の特定

- 20% **5.0 インフラストラクチャと自動化**
 - 5.1 インフラストラクチャの自動化におけるモデル駆動型プログラマビリティの価値の説明
 - 5.2 コントローラレベルとデバイスレベルの管理の比較
 - 5.3 ネットワークシミュレーションおよびテスト ツール(VIRL、pyATS など)の使用と役割についての説明
 - 5.4 インフラストラクチャの自動化における CI/CD パイプラインの構成要素とメリットの説明
 - 5.5 コードとしてのインフラストラクチャの原理の説明
 - 5.6 Ansible、Puppet、Chef、Cisco NSO などの自動化ツールの機能の説明
 - 5.7 Cisco API(ACI、Meraki、Cisco DNA Center、RESTCONF など)を使用した Python スクリプトによって自動化される作業フローの特定
 - 5.8 Ansible プレイブックによって自動化される作業フローの特定(マネジメント パッケージ、サービス、基本的なサービス コンフィグレーション、開始/停止に関連するユーザ管理)
 - 5.9 bash スクリプトによって自動化される作業フローの特定(ファイル管理、アプリのインストール、ユーザ管理、ディレクトリのナビゲーション)
 - 5.10 RESTCONF または NETCONF クエリの結果の解釈
 - 5.11 基本的な YANG モデルの解釈
 - 5.12 unified diff の解釈
 - 5.13 コードレビューのプロセスの原理とメリットの説明
 - 5.14 API コールを含むシーケンス図の解釈

- 15% **6.0 ネットワークの基礎**
 - 6.1 MAC アドレスおよび VLAN の目的および使用法の説明
 - 6.2 IP アドレス、経路、サブネット マスクおよびプレフィックス、ゲートウェイの目的および使用法の説明
 - 6.3 一般的なネットワークの構成要素(スイッチ、ルータ、ファイアウォール、ロード バランサなど)の機能の説明
 - 6.4 基本的なネットワークトポロジ図およびその構成要素(スイッチ、ルータ、ファイアウォール、ロードバランサ、ポート値など)の解釈
 - 6.5 ネットワーク デバイスの管理、データ、コントロールプレーンの機能の説明
 - 6.6 以下の IP サービスの機能の説明:DHCP、DNS、NAT、SNMP、NTP
 - 6.7 一般的なプロトコル ポートの値の認識(SSH、Telnet、HTTP、HTTPS、NETCONF など)
 - 6.8 アプリケーションの接続に関する問題の原因の特定(NAT の問題、トランスポート ポートのブロック、プロキシ、および VPN)
 - 6.9 ネットワークの制約によるアプリケーションへの影響についての説明