

コースコード：MS-AZ-400T00
税込価格：330,000円 (税抜価格：300,000円)
日数：4日間

ここに注目!!

MCP試験「AZ-400:Microsoft Azure DevOpsソリューション」受験予定の方に最適です。

受講対象者

このトレーニングはこのような方におすすめです。

DevOps プロセスの実装、もしくは Microsoft Azure DevOps Solutions
認定試験合格に関心がある、DevOps に関連するエンジニア

前提条件/前提知識

このコースを受講する前に受講者が習得しておく必要がある知識およびスキルは次のとおりです。

- ・ PaaS、SaaS、IaaS の実装に関する理解を含む、クラウドコンピューティングの概念を有すること
- ・ Azure 管理と Azure 開発の両方で、これらの領域の少なくとも 1 つで実績のある専門知識があること
- ・ Azure、バージョン管理、アジャイルなソフトウェア開発、およびコアソフトウェア開発の原則に関する基本的な知識を有すること
- ・ ソフトウェアを開発提供する組織でのソフトウェア開発・運用経験があること

目的

このコースを修了すると次のことができるようになります。

- ・ Git リポジトリやブランチの管理ができる
- ・ Azure Pipelines と GitHub Actions を使用した、DevOps ワークフローのパイプライン構成、エージェント管理、コンテナ構築戦略を立案できる
- ・ エンタープライズレベルのデプロイパイプライン構成、自動化された品質保証システム、スケーラブルな DevOps エクセレンスのための高度な承認ワークフローを実装できる
- ・ パイプラインアーキテクチャ内の自動リリースゲート、シークレット、シークレット管理を活用できる
- ・ Infrastructure as Code (IaC) の原則とベストプラクティスについて説明できる
- ・ DevSecOpsプラクティスを使用して、ソフトウェア開発ライフサイクル全体にセキュリティを実装できる
- ・ Azure Artifacts や GitHub Packages を活用した、依存関係の管理ができる
- ・ DevOps サイクル全体で継続的なフィードバックプラクティスを実装できる

アウトライン

Enterprise DevOps の開発



DevOps の導入

GitHub プロジェクトと Azure Boards を使用してアジャイルを計画する

ブランチ戦略とワークフローの設計と実装

Azure Repos で pull request を使用して共同作業する

Git フックを探索する

内部ソースの育成を計画する

リポジトリの管理と構成

技術的負債の特定

Azure Pipelines と GitHub Actions を使用して CI を実装する

Azure Pipelines を探索する

Azure Pipeline のエージェントとプールを管理する

パイプラインとコンカレンシーについて説明する

パイプライン戦略を設計して実装する

Azure Pipelines との統合

GitHub Actions の導入

GitHub Actions との継続的インテグレーションについて学習する

コンテナのビルド戦略を設計する

リリース戦略の設計と実装

リリースパイプラインを作成する

リリース戦略に関する推奨事項を確認する

環境の構成とプロビジョニング

タスクとテンプレートの管理とモジュール化

正常性の検査を自動化する



デプロイ パターンの概要

青緑色のデプロイと機能の切り替えを実装する

カナリア リリースとダーク ローンチを実装する

A/B テストとプログレッシブ 露出展開を実装する

ID 管理システムとの統合

アプリケーション構成データの管理

Azure と DSC を使用したコードとしてのインフラストラクチャの管理

コードとしてのインフラストラクチャと構成の管理について調べる

Azure Resource Manager テンプレートを使用して Azure リソースを作成する

Azure CLI を使用して Azure リソースを作成する

DevOps を使用して Azure Automation を探索する

Desired State Configuration (DSC) の実装

Bicep を実装する

セキュリティを実装し、コンプライアンスのコード ベースを検証する

Secure DevOps の概要

オープンソース ソフトウェアを実装する

ソフトウェアコンポジション分析

セキュリティの監視とガバナンス

依存関係管理戦略を設計して実装する

パッケージの依存関係を調べる

パッケージ管理について

成果物の移行、統合、セキュリティ保護



バージョン管理戦略を実装する

GitHub Packages の概要

継続的フィードバックを実装する

使用状況とフローを追跡するためのツールを実装する

監視ダッシュボードと状態ダッシュボードを開発する

チーム内でナレッジを共有する

アプリケーション分析を自動化するプロセスを設計する

アラート、ブレイムレスレトロスペクティブ
(誰も責めることのないふりかえり)、およびジャストカルチャを管理する