GitHub

学習ガイド
GitHub 管理

総合的な学習ガイドで GitHub Administration 認定試験に備えましょう。GitHub 管理試験に向けてしっかりと準備して合格の可能性を高めるために、必須のリソースと学習アクティビティを厳選しました。

対象読者のプロフィール

この試験は、GitHub Enterprise 管理で中級レベルの経験があるシステム管理者、ソフトウェア開発者、アプリケーション管理者、IT プロフェッショナル向けに作成されています。

試験範囲

認定試験の試験範囲は、「ドメイン」または「試験ドメイン」と呼ばれることが多く、認定試験で出題される具体的な知識、スキル、トピックを定義する構造化された概要またはフレームワークです。試験の出題傾向や、学習と準備が必要な部分についての明確なロードマップを受験者に提供します。

この学習ガイドで提供されるドメインは、GitHub 管理試験がカバーするトピックカテゴリーと、各ドメインでの学習目標について理解を深めることを目的としています。

ドメインの内訳	ドメイン別の割合
ドメイン 1: ユーザーと主要な関係者向けの GitHub Enterprise をサポートする	15%
ドメイン 2: ユーザー ID と GitHub 認証を管理する	20%
ドメイン 3: GitHub のデプロイ、配布、ライセンスの方法を説明する	5%
ドメイン 4: メンバーシップに基づいてアクセスと権限を管理する	20%
ドメイン 5: 安全なソフトウェア 開発を可能にしてコンプライアンスを確保する	15%
ドメイン 6: GitHub Actions を管理する	20%
ドメイン 7: GitHub Packages を管理する	5%

合格のための推奨事項とベストプラクティス

Github 管理試験に合格する可能性を高めるには、GitHub 管理の基本的な経験、知識、管理能力の基盤固めから始めることが不可欠です。この認定試験の推奨されるラーニング パスでは、学習コンテンツを徹底的に学習した後に、実践的な演習やコンテンツに取り組み、知識を磨き、試験に備えることができます。

コンテンツ リソース

GitHub 管理試験における各ドメインの学習目標をカバーする推奨コンテンツとして、GitHub と共同で次のリソースが作成されました。Microsoft Learn と LinkedIn ラーニングは、どちらも認定試験のための完全なラーニング パスを提供しますが、学習の進め方は異なります。

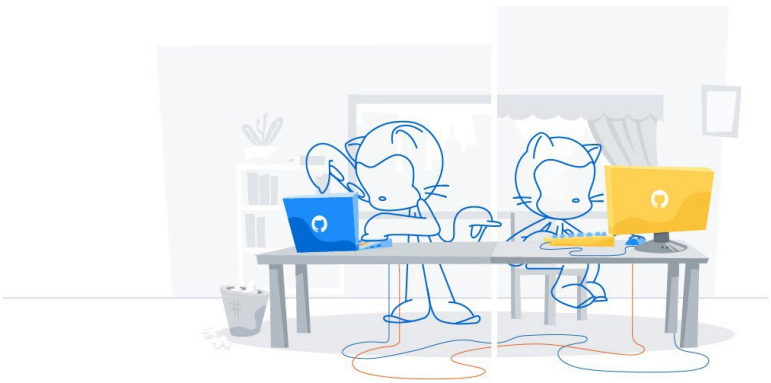
Microsoft Learn

MS Learn での GitHub 管理 ラーニング パスは、GitHub Actions 試験に備えるために作成された強力な学習モジュールのコレクションを提供します。GitHub 管理者として、組織の要求と開発者のワークフローをサポートし、安全で安定した GitHub 環境を維持する必要があります。次のモジュールでは、GitHub プラットフォーム上で管理者として利用できる、さまざまなオプションとカスタマイズの概要を説明します。ユーザー ID と認証、デプロイオプション、アクセスと権限の管理、安全なソフトウェア開発、コンプライアンスの確保に関する詳しい情報が得られます。

LinkedIn ラーニング

LinkedIn Learning

GitHub Administration 認定試験向けに制作されたビデオコースのシリーズを紹介する LinkedIn ラーニングの GitHub Administration 認定試験に向けた準備ラーニング パスをご体験ください。GitHub 管理者としての役割は、組織の独自のニーズと開発者のワークフローに完全に一致する、堅牢で安全かつ高効率な GitHub 環境を維持する上で極めて重要です。GitHub 管理者として利用できる豊富なオプションとカスタマイズを詳しく学ぶには、業界の専門家が指導する厳選モジュールをお調べください。ソフトウェア開発とコンプライアンスに不可欠なユーザー ID 管理、認証、デプロイ戦略、アクセス制御、セキュリティ対策に関する貴重な知識が得られます。



ドメイン 1: ユーザーと主要な関係者向けの GitHub Enterprise をサポートする

ユーザーと主要な関係者向けの GitHub Enterprise をサポートする
管理者が解決できる問題と GitHub Support が必要な問題を区別する
サポート バンドルと診断ファイルの生成方法を説明する
Enterprise において GitHub の製品とサービスがどのように使用されているかを説明し、十分に活用されていない機能、使用中のインテグレーション、最もアクティブなチーム、リポジトリを確認します。
コードコラボレーション (フォークとプルまたはブランチング)、ブランチング、ブランチ保護ルール、コード所有者、コード レビュー、自動化、リリース戦略を含む、開発者ワークフローの標準を推奨します。
Enterprise におけるツール エコシステムを説明する
Enterprise の CI/CD 戦略を説明する
Enterprise 内のチームにツール導入とワークフローを推奨する方法について論じる
GitHub API を使用して、監査ログのクエリや保存など、ユーザーインターフェイスで管理者の機能を拡張する方法を説明する
GitHub Marketplace で特定のニーズ向けのアセットを見つける (例: Marketplace でAzure パイプライン GitHub アプリケーションを検索、インストール、設定してコードをデプロイする)
GitHub アプリケーションとアクションを比較する (例: それぞれの権限、構築方法、使用方法)
GitHub Marketplace のアプリケーションとアクションを使用するメリットとリスクをリストアップする

ドメイン 2: ユーザー ID と GitHub 認証を管理する

ユーザー ID と GitHub 認証を管理する
Enterprise アカウント内の個々の Organization に対して SAML シングル サインオン (SSO) を有効にする場合とすべての Organization に対して有効にする場合の影響をリストアップする
Enterprise アカウントを使用する 1 つの Organization と複数の Organization に対して SAML SSO を有効にして強制する手順をリストアップします。
Organization に対して 2 要素認証 (2FA) を要求する方法を説明する
サポートされている ID プロバイダーの選択方法を説明する
GitHub での ID 管理と認可の仕組みを説明する
インスタンス、1 つの Organization、複数の Organization におけるユーザーのメンバーシップの重要性をリストアップする

認証と認可の体系を説明する (特に、ユーザーがシステムにアクセスする方法、GitHub 内の特定箇所にアクセスする権限をユーザーに付与する方法)
サポートされているクロスドメイン ID 管理システム (SCIM) プロバイダー (Azure、Okta、カスタム) をリストアップする
SCIM プロトコルの仕組みとそれを GitHub がサポートする方法を説明する
チームの同期の仕組みを説明する
チームの同期と SCI を比較する

ドメイン 3: GitHub のデプロイ、配布、ライセンスの方法を説明する

GHES、GHEC、GHAE の機能を比較する
GitHub Enterprise Cloud (GHEC) について説明する
GitHub Enterprise Server (GHES) について説明する
GitHub AE について説明する

シートライセンス、GitHub Actions、GitHub Packages を含む製品の請求方法を区別する
GitHub Actions の料金について説明する
複数の Organization アカウントを利用する場合の料金とサポートについて説明する
特定の Organization のライセンス利用状況の統計情報を確認する方法を説明する
マシン アカウントと周辺サービスのライセンス利用状況の統計情報を確認する方法を説明する
レポートに基づいて従量課金制製品の消費量を説明する (例: GitHub Actions の利用時間数または GitHub Packages のストレージ)

ドメイン 4: メンバーシップに基づいてアクセスと権限を管理する

Enterprise 権限とポリシーについて説明する
1 つの Organization をデプロイする場合と複数の Organization をデプロイする場合のメリットとコストを説明する
Organization 間でデフォルトの読み取り権限を設定する方法とデフォルトの書き込み権限を設定する方法を説明する
AD を介したチームの同期について説明する
保守性および複数の Organization とアクセス権に対するスクリプトの記述を説明する
企業の信頼性と管理の立場に沿って Enterprise ポリシーと Organization の権限を調整する方法を説明する

Enterprise 権限とポリシーについて説明する
GitHub Organization を定義する
Organization メンバーとして考えられるロールをリストアップする
Organization メンバー、オーナー、支払いマネージャーの権限を比較する
Organization メンバーと外部コラボレーターであることの違いを説明する
インスタンスや Organization におけるユーザーのメンバーシップの重要性をリストアップする
リポジトリ、Organization、またはチームへのアクセスに対して、最低限必要な権限をユーザーに付与する方法を説明する
新しい Organization を作成するメリットとデメリットをリストアップする

チーム権限について説明する
GitHub Organization での Teams を定義する
チーム メンバーとして考えられるロールを定義する
異なる権限体系について説明する

リポジトリ権限
リポジトリ ロール、チーム メンバーシップ、Organization メンバーシップなどの権限のリストに基づいたユーザーのアクションを説明する (https://github.com/organizations/<ORG_NAME>/settings/member_privileges)
リポジトリ メンバーシップのオプションをリストアップする
リポジトリへの監査アクセスについて説明する

ドメイン 5: 安全なソフトウェア 開発を可能にしてコンプライアンスを確保する

安全なソフトウェア 開発を可能にしてコンプライアンスを確保する
GitHub がどのように Enterprise のセキュリティ態勢をサポートしているかを説明する
Git リポジトリ から極秘データを削除する方法 (filter-branch/BFG) を説明する
GitHub から極秘データを削除する方法 (サポートに問い合わせ) を説明する
どの程度の制御が必要かに基づいてポリシーを選択する方法を説明する
特定のポリシーセットを選択した場合の影響を説明する
Organization ポリシーを定義する
Enterprise ポリシーを定義する

監査ログ API (REST と GraphQL) を使用して不足しているアセットについて説明する
監査ログのユース ケースを定義する
GitHub のセキュリティとコンプライアンスの概念を説明する
監査用レポートの提供方法を説明する

GitHub リポジトリのセキュリティ機能に関する重要性を定義して説明する
セキュリティポリシーの重要性を説明する
脆弱性を定義する
脆弱な依存関係について説明する
シークレット スキャンの重要性を説明する
コード スキャンの重要性を説明する
自動コード スキャン (CodeQL) について説明する
依存関係グラフについて説明する
セキュリティ アドバイザリの重要性を説明する
Dependabot について説明する
セキュリティの脆弱性のある古い依存関係を検出して修正する
セキュリティの脆弱性アラートについて説明する
GitHub リポジトリ上の極秘データに対処するセキュリティ対応計画を作成して実施する
SSH キーとデプロイキーを使用してリポジトリデータにアクセスする方法を説明する

API アクセスとインテグレーション
サポートされているアクセス トークン (例: PAT、インストール トークン、OAuth トークンとGitHub アプリケーションの OAuth トークン、デバイス トークン、リフレッシュ トークン)
トークンのレート制限を確認する方法を説明する
GitHub アプリケーション、アプリのリポジトリ権限、ユーザー権限、イベント サブスクリプションを説明する
OAuth アプリケーション、アプリの権限、イベント サブスクリプションを説明する
個人用アクセス トークン (PAT) とマシン アカウントを認証するための GitHub アプリケーションの使用方法を比較する
マシン アカウントと GitHub アプリケーションの使用方法を説明する
セキュリティポリシーに基づいて、ユーザーが作成した GitHub アプリケーションと OAuth アプリケーションを承認または却下する方法を説明する
Enterprise Managed User (EMU) を定義する

ドメイン 6: GitHub Actions を管理する

アクションとワークフローを企業に配布する
アクションとワークフローの再利用テンプレートを特定する
再利用可能なコンポーネント (例: ストレージのリポジトリ、ファイル/フォルダーの命名規則、継続的なメンテナンス計画) を管理および活用するためのアプローチを定義する
企業向けにアクションを配布する方法を定義する
Enterprise 内のアクションへのアクセスを制御する方法を説明する
GitHub Actions の組織内利用のポリシーを構成する

企業向けのランナーを管理する
GitHub ホストおよびセルフホスト ランナーで IP 許可リストを構成した場合の影響について説明する
内部のアプリケーションとシステムの IP 許可リストを設定し、GitHub ホステッド ランナーとのやり取りを許可する
パブリック リポジトリでセルフホステッド ランナーを有効にすることによる影響と潜在的な攻撃ベクトルをリストアップする
ワークロードをサポートするのに適切なランナーを選択する (例: サポートされているオペレーティングシステムを選択してセルフホステッドまたは GitHub ホステッド ランナーを使用)
GitHub ホステッド ランナーとセルフホステッド ランナーを比較する
企業利用向けにセルフホスト ランナーを構成する (プロキシ、ラベル、ネットワーキングを含む)
グループを使用してセルフホステッド ランナーを管理する (例: アクセスを管理する、ランナーをグループ内およびグループ間で移動させる)
セルフホステッド ランナーをモニタリング、トラブルシューティング、更新する

企業内の暗号化されたシークレットを管理する
暗号化されたシークレットの範囲を特定する
アクションとワークフロー内の暗号化されたシークレットにアクセスする方法を説明する
Organization アカウントレベルで暗号化された機密情報を管理する方法を説明する
リポジトリ レベルで暗号化されたシークレットの管理方法を説明する
サードパーティーの Vault の使用方法を説明する

ドメイン 7: GitHub Packages を管理する

企業内の暗号化されたシークレットを管理する
サポートされている GitHub Packages について説明する
GitHub Packages にアクセス、記述、共有する方法を説明する
ワークフローでの GitHub Packages の使用方法を説明する (例: GitHub Actions または他の CI/CD ツールを使用)
GitHub Packages と GitHub Releases の違いとユース ケースを説明する