

試験の準備方法-権威のあるGH-200学習範囲試験-ユニークなGH-200無料ダウンロード



BONUS!!! Jpshiken GH-200ダンプの一部を無料でダウンロード: https://drive.google.com/open?id=1EJhfHRUozrbBOwLY0Ko3_upzKeLVG_1

GH-200試験問題の継続的な刷新により、当社は大きな市場シェアを占めています。強力な研究センターを構築し、GH-200トレーニングガイドでより良い仕事をするために強力なチームを所有しています。これまで、GH-200学習教材に関する多くの特許を取得しています。一方で、当社Microsoftは改修の恩恵を受けています。お客様は当社の製品を選択する可能性が高くなります。一方、私たちが投資したお金は有意義なものであり、GH-200試験の新しい学習スタイルを刷新するのに役立ちます。

Microsoft GH-200 認定試験の出題範囲:

トピック	出題範囲
トピック 1	企業におけるGitHub Actionsの管理: このセクションでは、企業管理者とプラットフォームエンジニアが組織レベルでGitHub Actionsとワークフローを配布および管理する専門知識を評価します。これには、テンプレートの再利用と共有、リポジトリと命名規則を介した再利用可能なコンポーネントの管理戦略、アクションへのアクセス制御、組織全体の使用ポリシーの設定、そしてGitHub Actionsを企業全体で効率的に展開するためのメンテナンス計画が含まれます。
トピック 2	ワークフローの作成と管理: このセクションでは、DevOpsエンジニアと自動化スペシャリストのスキルを評価し、プッシュ、スケジュールされた時間、手動トリガー、Webhookなどのイベントによってトリガーされるワークフローの構築と管理を網羅します。ジョブ、ステップ、アクション、ランナーなどのワークフローコンポーネントの理解、構文の正確性、環境変数、シークレット管理、ジョブ間の依存関係などが含まれます。また、パッケージの公開、サービスコンテナの使用、ジョブのルーティング、クラウドプロバイダーへのリリースのデプロイなど、様々な目的のワークフローを作成するための実践的な能力も証明されます。
トピック 3	ワークフローの利用: このドメインはソフトウェア開発者と品質保証エンジニアを対象としており、ワークフローの実行とその結果の解釈に焦点を当てています。トリガーイベントの特定、ワークフロー構成の読み取り、ログ分析による障害のトラブルシューティング、デバッグログの有効化、環境変数の管理、依存関係のキャッシュ、ジョブ間のデータの受け渡しなどを網羅しています。また、ワークフローの実行、成果物、承認、ステータスパッジの管理、リポジトリ内でのワークフローの特定、組織でテンプレート化されたワークフローの活用についても学習します。

- アクションの作成と保守: このドメインでは、アクション開発者と自動化エンジニアが、JavaScript、Dockerコンテナ、実行ステップなど、適切な種類のGitHub Actionsを選択・作成する能力を評価します。アクションコードのトラブルシューティング、アクションのコンポーネントとファイル構造の理解、そしてアクション内でワークフローコマンドを使用してランナーと通信する能力（終了コードの管理を含む）に重点を置きます。

>> GH-200学習範囲 <<

Microsoft GH-200無料ダウンロード、GH-200日本語対策問題集

GH-200試験は優秀なあなたにとって難しいかもしれませんが、試験の合格を保証するために、参照できる資料を購入することができます。我々のGH-200問題集は通過率が高いので、あなたの要求を満たすことができます。資料を購入するなら、弊社のGH-200問題集を選んでください。

Microsoft GitHub Actions 認定 GH-200 試験問題 (Q34-Q39):

質問 # 34

As a developer, you need to integrate a GitHub Actions workflow with a third-party code quality provider that uses the Checks API. How should you trigger a follow-up workflow?

- A. Add the pull_request webhook event as a trigger for the workflow when the code quality integration is synchronized
- **B. Add the check_run webhook event as a trigger for the workflow when the code quality integration is completed**
- C. Add the deployment webhook event as a trigger for the workflow when the code quality integration is completed
- D. Add the workflow_run webhook event as a trigger for the workflow for the code quality integration name

正解: B

解説:

The check_run event is triggered when a check (such as a code quality check) completes, including when the status of a check changes. By adding this event as a trigger, you can initiate a follow-up workflow when the code quality integration finishes its checks.

Reference:

<https://docs.github.com/en/actions/reference/workflows-and-actions/events-that-trigger-workflows>

質問 # 35

As a developer, you need to create a custom action written in Python. Which action type should you choose?

As a developer, you need to create a custom action written in Python. Which action type should you choose?

- A. composite run step
- **B. Docker container action**
- C. JavaScript action
- D. Python action

正解: B

解説:

A Docker container action is ideal for custom actions that require specific environments or dependencies, such as Python. By creating a Docker container, you can define the environment with the necessary Python version and dependencies, and your Python code can run inside that container.

質問 # 36

You need to make a script to retrieve workflow run logs via the API. Which is the correct API to download a workflow run log?

- A. POST /repos/owner/repo/actions/runs/run_id/logs
- **B. GET /repos/owner/repo/actions/runs/run_id/logs**
- C. GET /repos/owner/repo/actions/artifacts/logs

- D. POST /repos/:owner/repo/actions/runs/run_id

正解: B

解説:

REST API endpoints for workflow jobs

You can use the REST API to view logs and workflow jobs in GitHub Actions.

Download job logs for a workflow run

Gets a redirect URL to download a plain text file of logs for a workflow job.

Code samples for "Download job logs for a workflow run"

Request example

Get /repos/{owner}/{repo}/actions/jobs/{job_id}/logs

Reference:

<https://docs.github.com/en/rest/actions/workflow-jobs>

質問 # 37

Which of the following scenarios would require the use of self-hosted runners instead of GitHub-hosted runners?

- A. running more than the three concurrent workflows supported by GitHub-hosted runners
- B. exceeding 50,000 monthly minutes of build time
- C. using Docker containers as part of the workflow
- D. performing builds on macOS
- E. using specialized hardware configurations required for workflows

正解: A、E

解説:

GitHub-hosted runners have a limit on the number of concurrent workflows (typically 20 for free-tier accounts and 5 for enterprise). If your organization needs to run more workflows simultaneously, you would need to use self-hosted runners to increase the available concurrency.

Self-hosted runners allow you to configure specialized hardware or software setups that are necessary for certain workflows.

GitHub-hosted runners may not have access to custom hardware configurations like GPUs or other specialized resources, so self-hosted runners are required in such cases.

質問 # 38

You installed specific software on a Linux self-hosted runner. You have users with workflows that need to be able to select the runner based on the identified custom software. Which steps should you perform to prepare the runner and your users to run these workflows? (Each correct answer presents part of the solution. Choose two.)

- A. Inform users to identify the runner based on the group.
- B. Create the group custom-software-on-linux and move the runner into the group.
- C. Add the label custom-software to the runner.
- D. Configure the webhook and network to enable GitHub to trigger workflow.
- E. Inform users to identify the runner with the labels custom-software and linux.
- F. Add the label linux to the runner.

正解: C、E

解説:

[F] Add a label for the custom software as required.

Note: Using labels with self-hosted runners

You can use labels to organize your self-hosted runners based on their characteristics.

Creating a custom label

You can create custom labels for runners at the repository and organization levels.

[C, not D] Linux is a default label and does not have to be created to be used.

Reference:

<https://docs.github.com/en/actions/how-to/manage-runners/self-hosted-runners/apply-labels>

• • • • •

GH-200無料ダウンロード：https://www.jpshiken.com/GH-200_shiken.html

- P.S. JpshikenがGoogle Driveで共有している無料かつ新しいGH-200ダンプ: https://drive.google.com/open?id=1EJhfHRUozrbBOwLY0Ko3_upzKeLVG 1