# GitHub Action

## Wipro Training

### GitHub Action:

GitHub Actions is a core CI/CD and automation feature that operates natively within an integrated DevOps platform.

### Actions & SDLC:

Use cases across SDLC using GitHub Actions:

1. **Plan:** Tirage Issues and Project Boards using Actions
2. **Code:** Automates code builds, linting upon pull request creation
3. **Build:** Compile and build, Store artifacts
4. **Test:** Automation of the unit tests, integration tests
5. **Release:** Publish release, Release notes, Versioning and Upload artifacts
6. **Deploy:** Deploy applications to chosen environment

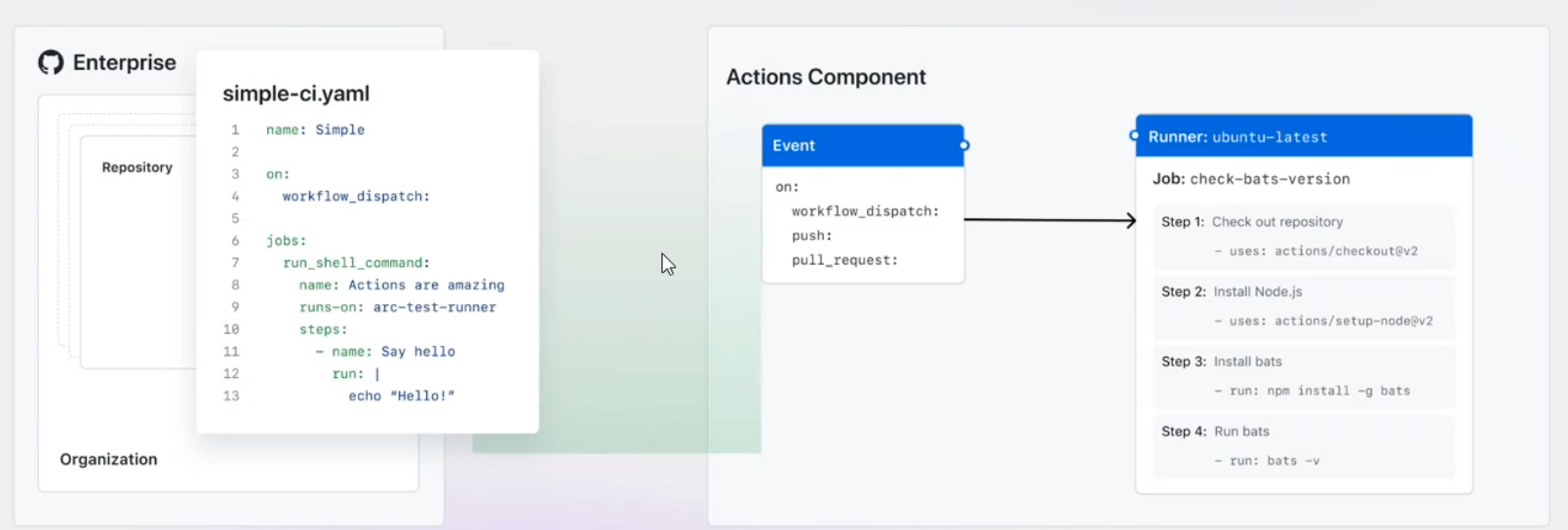


### Automation accessible to every developer:

**Actions:** Enterprise-grade CI/CD that supports Windows, Linux, Mac

1. Marketplace of 20,000+ Actions by our community
2. Any Operating System, Cloud and On-Prem
3. Natively integrated into GitHub workflows
4. Friction-free service or self-hosted runners
5. Integrated Azure Pipelines and any other CI/CD
6. DRY with Reusable Workflows
7. API for viewing cache usage

### GitHub Actions Key Components:



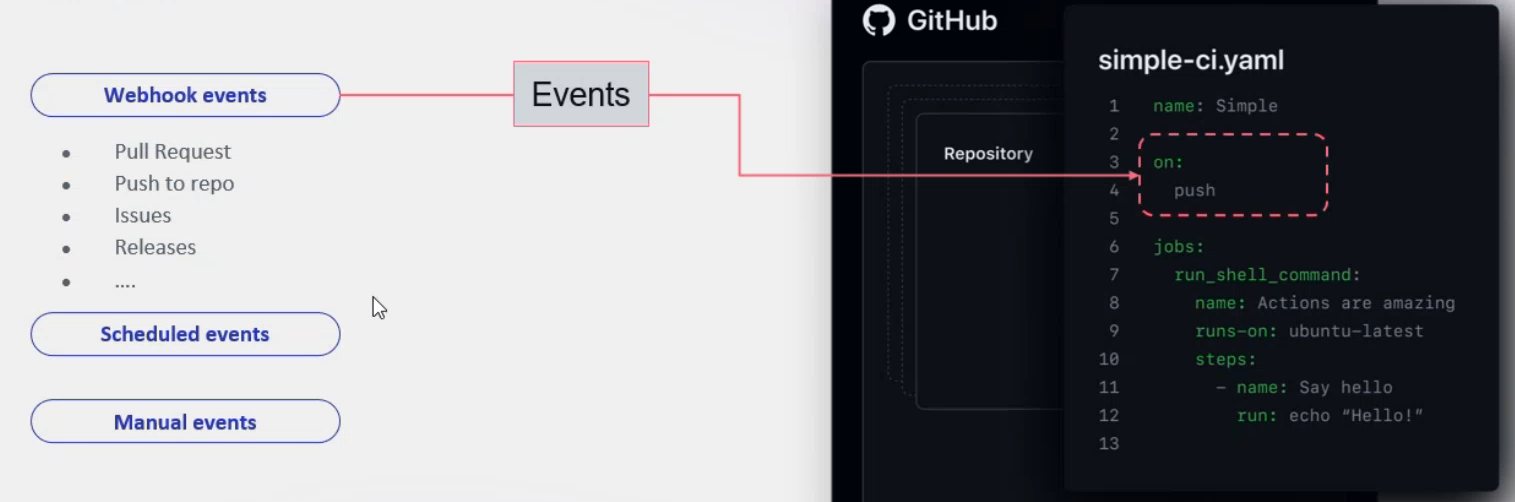
1. **Workflows:** A workflow is an automated process that you define in your repository. It's made up of one or more jobs and is defined by a YAML file in the .github/workflows directory. Workflows can be triggered by events, scheduled, or run manually.
2. **Events:** Events are specific activities that trigger a workflow run. These can be GitHub events like push, pull request, or issue creation, or they can be scheduled or triggered via the API.
3. **Jobs:** A job is a set of steps that execute on the same runner. Jobs can run in parallel by default, or you can configure dependencies between jobs to control the order of execution.
4. **Steps:** Steps are individual tasks within a job. They can run commands or use actions. Steps are executed sequentially within a job and share the same environment, allowing them to share data.
5. **Actions:** Actions are reusable units of code that can be used to simplify workflows. They can be used to set up the environment, run commands, or perform other tasks. You can use actions from the GitHub Marketplace or create your own.
6. **Runners:** Runners are servers that run your workflows. GitHub provides hosted runners with Linux, Windows, and macOS environments, or you can use self-hosted runners for custom environments.

### Actions:

1. Reusable units of code that can be referenced in a workflow
2. GitHub runs them in Node.js runtime or in containers
3. Reference in Action or run scripts directly



### Events:



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Action Workflow Syntax:

1. The **name of the workflow** as it will appear in the “Actions” tab of the GitHub repository
2. The **name for workflow runs** generated from the workflow which will appear in the list of workflow runs on your repository’s “Actions” tab
3. Specifies **the trigger for this workflow**, this example uses the push event, so a workflow run is triggered every time someone pushes a change to the repository or merges a pull request.
4. Groups together all the **jobs** that run in the learn-github-action workflow
5. Defines a job named **check-bats-version.** The child keys will define properties of the job
6. Configures the **job runner** on the latest version of an Ubuntu Linux runner
7. Groups together all the **steps** that run in the **check-bats-version** job and the uses keyword specifies that this step will run **v4** of the **actions/checkout action**



### Conditional Statements:

## YouTube

### Basic:

GitHub Actions is a **CI/CD pipeline directly integrated with your GitHub repository**.

GitHub Actions allows to **automate:**

1. Running test suites
2. Building images
3. Compiling static sites
4. Deploying code to servers

GitHub Actions has **templates** you can use to get started

Within a GitHub repo you will have a **tab for Actions**.

GitHub Actions files are defined as YAML files located in the **.github/workflow** folder in your repo.

When you run GitHub Actions you will get **a history of workflow runs** where it will indicate if it was success a failure, and how long it took to run.

GitHub has a repo example workflow you can use to get your started <https://github.com/actions/starter-workflows>

### Types of GitHub Actions:

Event Triggers causes a GitHub Action to run.

The **on** attribute specifies the **event trigger** to be used. GitHub Actions has 35+ event triggers.

Examples of common GitHub Actions that could be triggered:

1. **Pushes:** Triggers an action on any push to the repository
2. **Pull Requests:** Run actions when pull requests are opened, updated or merged
3. **Issues:** Execute actions based on issue activities, like creation or labelling
4. **Releases:** Automate workflows when a new release is published.
5. **Scheduled Events:** Schedule actions to run at specific times.
6. **Manual Triggers:** Allow manual triggering of actions through the GitHub UI.

### Workflows:

A **workflow** is a **configurable automated process** that will run one or more jobs.

Workflows are defined by YAML file checked in to your repository.

Workflows in your repo are defined in the following directory: **.github/workflows**

A repo can contain multiple workflows.

Workflow triggers are events that cause a workflow to run:

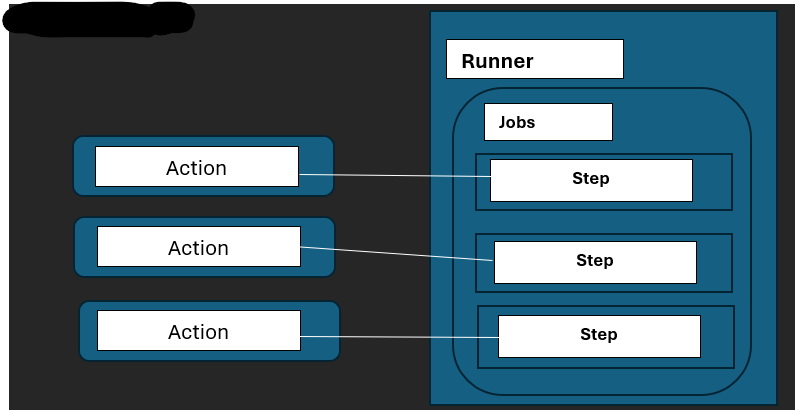
1. Events that occur in your workflow’s repository
2. Events that occur outside of GitHub and trigger a **repository\_dispatch** event on GitHub
3. Scheduled times
4. Manual

The **name property** lets you easily identify workflows.

### Workflow Components:

The following GitHub Actions workflow components:

1. **Actions:** Reusable tasks that perform specific jobs within a workflow.
2. **Workflows:** Automated processes defined in your repository that coordinate one or more jobs, triggered by events or on a schedule.
3. **Jobs:** Groups of steps that execute on the same runner, typically running in parallel unless configured otherwise.
4. **Steps:** Individual tasks within a job that run commands or actions sequentially.
5. **Runs:** Instances of workflow execution triggered by events, representing the complete run-through of a workflow.
6. **Runners:** Servers that host the environment where the jobs are executed, available as GitHub-hosted or self-hosted options.
7. **Marketplace:** A platform to find and share reusable actions, enhancing workflow capabilities with community-developed tools.



### Triggering Schedule Event:

**Schedule** can use a **cron expression** to trigger a workflow at a specific time or day.

You can use <https://crontab.guru/> to translate time into a cron expression

### Single vs Multiple Events

A screenshot of a computer program

Description automatically generated

If you specify multiple events, only one of those events needs to occur to trigger your workflow. If multiple triggering events for your workflow occur at the same time, multiple workflow runs will be triggered.

## Udemy

## Quiz

### Set 1

#### If job B requires job A to be finished you have to:

1. use the needs keyword in job A to create this dependency
2. **use the needs keyword in job B to create this dependency**
3. use the requires keyword in job A to create this dependency
4. use the requires keyword in job B to create this dependency

#### How can you use the GitHub API to download workflow run logs?

1. **GET /repos/{owner}/{repo}/actions/runs/{run\_id}/logs**
2. HEAD /repos/{owner}/{repo}/actions/runs/{run\_id}/logs
3. PUT /repos/{owner}/{repo}/actions/runs/{run\_id}/logs
4. POST /repos/{owner}/{repo}/actions/runs/{run\_id}/logs

#### Which is true about environments?

1. Each job in a workflow can reference a maximum of two environments.
2. Each workflow can reference a single environment.
3. Each workflow can reference a maximum of two environments.
4. **Each job in a workflow can reference a single environment.**

#### Which is a correct way to print a debug message?

1. **echo "::debug::Watch out here!"**
2. echo "::debug::message=Watch out here!"
3. echo "Watch out here!" >> $GITHUB\_DEBUG
4. echo ":debug:Watch out here!"

#### When can you delete workflow runs?

1. When workflow run is in progress
2. **When workflow run has been completed**
3. When workflow run is 10 days old
4. When workflow run is two weeks old

#### Dave wants to be notified when a comment is created on an issue within a GitHub repository. Which event trigger should be used within the workflow configuration?

1. **issue\_comment**
2. issues
3. issues.comment
4. comment

#### Which context holds information about the event that triggered a workflow run?

1. jobs.<job\_id>.result
2. github.repository
3. **github.event**
4. github.job

#### How does the actions/cache action in GitHub Actions handle a cache miss?

1. by searching for a cache in other repositories
2. by requiring manual intervention to create a new cache
3. by terminating the workflow if a cache miss occurs
4. **by automatically creating a new cache if the job is completed successfully**

#### What is the recommended approach for storing secrets larger than 48 KB?

1. secrets larger than 48 KB cannot be stored
2. store large secrets directly as repository secrets to avoid limitations
3. avoid storing large secrets entirely to ensure security
4. **encrypt and store secrets in the repository but keep the decryption passphrase as a secret**

#### What does the matrix keyword do in a GitHub Actions workflow?

1. Sets environment variables for the job
2. Triggers workflows based on a schedule
3. Defines secrets for the workflow
4. **Allows defining multiple job configurations to run in parallel**

#### Which of the following GitHub Actions syntax is used to run multiple commands in a single step?

1. Using && to chain commands
2. Defining commands in an array
3. Separating commands with a semicolon ;
4. **Using a multiline string with |**

#### A job called job2 is using artifacts created in job1. Therefore it's important to make sure job1 finishes before job2 starts looking for the artifacts. How should you create that dependency?

1. create this dependency by defining job2 after job1 in the workflow's .yaml definition
2. this dependency is created implicitly when using actions/download-artifact to download artifact from job1
3. **create this dependency using the needs keyword in job2**
4. create this dependency using the concurrency keyword in job2

#### What jobs.job\_id.if conditional will make sure that job production-deploy is triggered only on my-org/my-repo repository? (Select two.)

jobs:

production-deploy:

if: <CONDITION>

runs-on: ubuntu-latest

steps:

...

1. **if: ${{ github.repository == 'my-org/my-repo' }}**
2. if: ${{ github.org == 'my-org' && github.repository == 'my-repo' }}
3. if: ${{ github.organization == 'my-org' && github.repository == 'my-repo' }}
4. **if: github.repository == 'my-org/my-repo'**

#### How can you reQuire manual approvals by a maintainer if the workflow run is targeting the production environment?

1. **Using deployment protection rules**
2. Setting the required reviewers in the production workflow
3. Manual approvals are not supported by GitHub Actions
4. Using branch protection rules

#### What is true about the following workflow configuration if triggered against the octo/my-dev-repo repository?

name: deploy-workflow

on: [push]

jobs:

production-deploy:

if: github.repository == 'octo/my-prod-repo'

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with:

node-version: '14'

- run: npm install -g bats

1. the production-deploy job will error
2. the production-deploy job will execute three steps
3. the production-deploy job will run if the node-version is 14
4. **the production-deploy job will be marked as skipped**

#### What is the correct syntax for specifying a cleanup script in a container action?

runs:

using: 'docker'

image: 'Dockerfile'

entrypoint: 'entrypoint.sh'

cleanup: 'cleanup.sh'

runs:

using: 'docker'

image: 'Dockerfile'

entrypoint: 'entrypoint.sh'

after-entrypoint: 'cleanup.sh'

runs:

using: 'docker'

image: 'Dockerfile'

entrypoint: 'entrypoint.sh'

after: 'cleanup.sh'

**runs:**

**using: 'docker'**

**image: 'Dockerfile'**

**entrypoint: 'entrypoint.sh'**

**post-entrypoint: 'cleanup.sh'**

runs:

using: 'docker'

image: 'Dockerfile'

entrypoint: 'entrypoint.sh'

post: 'cleanup.sh'

#### To run a step only if the secret MY\_SECRET has been set, you can:

1. By creating the following conditional on step level

my-job:

runs-on: ubuntu-latest

steps:

- if: ${{ secrets.MY\_SECRET == '' }}

1. By creating the following conditional on job level

my-job:

runs-on: ubuntu-latest

if: ${{ secrets.MY\_SECRET == '' }}

1. **Set the secret MY\_SECRET as a job level environment variable, then reference that environment variable to conditionally run that step**

**my-job:**

**runs-on: ubuntu-latest**

**env:**

**my\_secret: ${{ secrets.MY\_SECRET }}**

**steps:**

**- if: ${{ env.my\_secret != '' }}**

1. By creating the following conditional on step level

my-job:

runs-on: ubuntu-latest

steps:

- if: ${{ secrets.MY\_SECRET }}

#### Which event is triggered by a webhook action from outside of the repository?

1. workflow\_dispatch
2. remote\_dispatch
3. **repository\_dispatch**
4. api\_dispatch
5. webhook\_dispatch

#### You want to create a reusable workflow CI that runs some Quality checks, linting and tests on code changes. What event trigger should the CI workflow define to allow reusing it in other workflows?

1. workflow\_trigger
2. workflow\_run
3. workflow\_dispatch
4. **workflow\_call**

#### Where can you find network connectivity logs for a GitHub self-hosted-runner?

1. **In the \_diag folder directly on the runner machine**
2. In the job run logs of a job that ran on that Runner
3. In the job run logs of a job that ran on that Runner with debug logging enabled
4. On GitHub.com on that specific Runner's page

#### In a workflow with multiple jobs, if job A fails then:

1. **the jobs that are dependent on job A are skipped**
2. the jobs that are dependent on job A fail
3. the workflow immediately cancels all other jobs

#### Dave is creating a templated workflow for his organization. Where must Dave store the workflow files and associated metadata files for the templated workflow?

1. inside a directory named workflow-templates within the current repository
2. inside a directory named .github/workflow-templates
3. inside a directory named .github/org-templates
4. **inside a directory named workflow-templates within a repository named .github**

#### A workflow that has only workflow\_dispatch event trigger can be triggered using GitHub's REST API

1. **True**
2. False

#### How can organizations which are using GitHub Enterprise Server enable automatic syncing of third party GitHub Actions hosted on GitHub.com to their GitHub Enterprise Server instance?

1. GitHub Enterprise Server cannot use GitHub.com Actions because of it's on-premise nature and no internet access
2. Using actions-sync tool
3. GitHub Enterprise Server has access to all GitHub.com Actions by default
4. **Using GitHub Connect**

#### To stop a workflow from running temporarily without modifying the source code you should

1. Prevent any new commits to main branch
2. **Use the Disable workflow option in GitHub Actions**
3. Delete environment that is required for this workflow
4. Remove secrets that are required for this workflow

#### Which event allows you to manually trigger a workflow from the GitHub UI?

1. workflow\_trigger
2. manual\_dispatch
3. **workflow\_dispatch**
4. manual\_trigger

#### Select status check functions in GitHub Actions

1. status(), always(), cancelled() and failure()
2. state(), always(), cancelled() and failure()
3. **success(), always(), cancelled() and failure()**
4. status(), always(), cancelled() and failure()

#### What are the valid use cases for using defaults? (Select two.)

1. Using defaults.env on workflow level to set default environment variables for an entire workflow
2. **Using defaults.run on job level to set default working-directory for all steps in a single job**
3. Using defaults.run on step level to set default shell (e.g bash) for that single step
4. **Using defaults.run on workflow level to set default shell (e.g bash) for an entire workflow**
5. Using defaults.env on job level to set default environment variables for all steps in a single job

#### Your open-source publicly available repository contains a workflow with a pull\_request event trigger. How can you reQuire approvals for workflow runs triggered from forks of your repository?

1. **Setup required approvals for fork runs in the repository**
2. The workflow will not trigger for forks if using pull\_request event. If you want to do that you should use fork\_pull\_request event trigger with require-approval flag.
3. Setup branch protection rules for the repository
4. Setup deployment protection rules for the repository

#### How do you reference a secret stored in GitHub Secrets in a workflow?

1. **${{ secrets.SECRET\_NAME }}**
2. ${{ env.SECRET\_NAME }}
3. ${{ secret.SECRET\_NAME }}
4. ${{ config.SECRET\_NAME }}

#### For any action published in GitHub Marketplace, you can often use it in multiple versions, which approach is the most stable and secure?

1. Reference a version tag
2. **Reference the commit SHA**
3. Reference the main branch

#### How do you ensure that Upload Failure test report step is executed only if Run Tests step fails?

**- name: Run Tests**

**id: run-tests**

**run: npm run test**

**- name: Upload Failure test report**

**if: failure() && steps.run-tests.outcome == 'failure'**

**run: actions/upload-artifact@v3**

**with:**

**name: test-report**

**path: test-reports.html**

- name: Run Tests

id: run-tests

run: npm run test

- name: Upload Failure test report

if: steps.run-tests.outcome == 'failure'

run: actions/upload-artifact@v3

with:

name: test-report

path: test-reports.html

- name: Run Tests

id: run-tests

run: npm run test

- name: Upload Failure test report

if: always() && steps.run-tests.outcome == 'failure'

run: actions/upload-artifact@v3

with:

name: test-report

path: test-reports.html

- name: Run Tests

id: run-tests

run: npm run test

- name: Upload Failure test report

run: actions/upload-artifact@v3

with:

name: test-report

path: test-reports.html

#### What's the maximum amount of reusable workflows that can be called from a single workflow file?

1. 5
2. 10
3. **20**
4. 1

#### Which of these is not a valid event that could trigger a workflow?

1. Committing a file to master branch
2. A branch is created
3. Adding a label to a pull request
4. **Cloning the repository**

#### How can you ensure that a workflow called Deploy Prod is always running at most one at a time?

1. Use Queue on workflow level

Queue: ${{ github.workflow }}

1. Use order on workflow level

order: ${{ github.workflow }}

1. **Use concurrency on workflow level**

**concurrency: ${{ github.workflow }}**

1. Use parallel on workflow level

parallel: ${{ github.workflow }}

#### When creating custom GitHub Actions - in what file does all the action metadata have to be defined?

1. Metadata examples: name, description, outputs or required inputs
2. In the repository README file
3. In the action.yml or action.yaml file in the action repository, but it is not required if the action is not meant to be shared and used by the public
4. **In the action.yml or action.yaml file in the action repository**
5. It's edited in GitHub Marketplace UI when published for sharing

#### You defined a matrix job example\_matrix. How can limit the matrix to run a maximum of 2 jobs at a time?

jobs:

example\_matrix:

strategy:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

1. Use GitHub's REST API to check if the job count is lesser than 2
2. It's not possible, a matrix will always run all of the jobs in parallel if there are runners available
3. Set jobs.example\_matrix.strategy.concurrency to 2
4. **Set jobs.example\_matrix.strategy.max-parallel to 2**

#### Which is the correct way of triggering a job only if configuration variable MY\_VAR has the value of MY\_VALUE?

1. It's not possible because configuration variables cannot be used in job level if conditionals
2. By creating the following conditional on job level

my-job:

if: ${{ vars.MY\_VAR }} == 'MY\_VALUE'

1. It's not possible because configuration variables cannot be used in if conditionals
2. **By** creating **the following conditional on job level**

**my-job:**

**if: ${{ vars.MY\_VAR == 'MY\_VALUE' }}**

#### Which keyword allows you to define environment variables in a GitHub Actions workflow?

1. secrets
2. **env**
3. vars
4. config

#### How can you specify the schedule of a GitHub actions workflow to run on weekdays only?

1. **use the on: schedule: cron event trigger**
2. use the on: schedule: weekdays event trigger
3. add a condition in the workflow YAML for weekdays
4. it is not possible in GitHub actions

#### Your organization defines a secret SomeSecret, however when you reference that secret in a workflow using ${{ secrets.SomeSecret }} it provides a different value than expected. What may be the reason for that?

1. ${{ secrets.SomeSecret }} expression is only used for repository scoped secrets
2. **The secret SomeSecret is also declared in repository scope**
3. The secret SomeSecret is also declared in enterprise scope
4. You need to use the GitHub API to access organization scoped secrets

#### In job deploy, if you want to access binaries (containing your application) that were created in job build you should

1. cache the binaries in build and read the files from cache in deploy
2. **upload the binaries as artifacts in build and download them in deploy**
3. cache the binaries in deploy and read the files from cache in build
4. upload the binaries as artifacts in deploy and download them in build

#### What is the default shell used by GitHub Actions on Windows runners?

1. sh
2. bash
3. **powershell**
4. cmd

#### How can you use the GitHub API to create or update a repository secret?

1. GET /repos/{owner}/{repo}/actions/secrets/{secret\_name}
2. HEAD /repos/{owner}/{repo}/actions/secrets/{secret\_name}
3. POST /repos/{owner}/{repo}/actions/secrets/{secret\_name}
4. **PUT /repos/{owner}/{repo}/actions/secrets/{secret\_name}**

#### How can you skip the following workflow run when you commit or create a PR?

name: Build

on: [push, pull\_request]

jobs:

build:

runs-on: ubuntu-latest

name: Extract artifact version

...

1. Provide SKIP\_WORKFLOW in the commit message
2. The above workflow will run in every event of push or pull request in every case
3. **By including any one of the following keywords in the commit message or in the title of the pull-request**

**[skip ci]**

**[ci skip]**

**[no ci]**

**[skip actions]**

**[actions skip]**

#### When using GitHub Actions to access resources in one of the cloud providers (such as AWS, Azure or GCP) the safest and recommended way to authenticate is

1. Using Vault
2. Storing access keys in secrets
3. **Using OIDC**
4. Storing access keys in variables

#### How can you cache dependencies to speed up workflow execution?

1. Using the cache keyword
2. **Using the actions/cache action**
3. By storing them in the repository
4. By using the store keyword

#### On which commit and branch do scheduled workflows run in GitHub Actions?

1. Scheduled workflows run on the latest commit on the main branch.
2. Scheduled workflows run on the specific commit on the main branch.
3. **Scheduled workflows run on the latest commit on the repository default branch.**
4. Scheduled workflows run on the specific commit on last modified branch.

#### When creating a custom GitHub Action you have to store the source code in .github/workflows directory

1. True
2. **False**

#### Fill in the blank: When using push event trigger filters you can use <\_\_\_\_> patterns to target multiple branches

1. scheme
2. **glob**
3. action
4. regex

#### How can you specify the operating system for a job in GitHub Actions?

1. **Using the runs-on keyword**
2. Using the env keyword
3. Using the platform keyword
4. Using the os keyword

#### What are activity types of an event used for ?

1. Reacting to new activity on a repository (e.g new contributor)
2. **Limiting workflow runs to specific activity types using the types filter**
3. Checking if the activity comes from an user or a bot

#### Is this statement true? Docker container actions are usually slower than JavaScript actions

1. **True**
2. False

#### Which of the following statements accurately describes the behavior of workflow jobs referencing an environment's protection rules?

1. **workflow jobs won't start until all the environment's protection rules pass**
2. workflow jobs will only start if some of the environment's protection rules pass
3. workflow jobs will never start if the environment has protection rules
4. workflow jobs will start immediately, regardless of the environment's protection rules

#### Your Pull Request analysis workflow uses multiple code analysis tools and takes about 20minutes to fully complete. It is triggered on pull\_request event with branches filter set to master. Therefore if a developer pushes multiple commits within few minutes multiple workflows are running in parallel. How can you stop all previous workflow runs and only run the one with latest changes?

1. Use activity types filter

on:

pull\_request:

branches:

- master

types: [latest]

1. Use concurrency

concurrency:

group: ${{ github.ref }}

1. Use cancel-in-progress flag for pull\_request event

on:

pull\_request:

branches:

- master

cancel-in-progress: true

1. **Use concurrency with cancel-in-progress**

**concurrency:**

**group: ${{ github.workflow }}-${{ github.ref }}**

**cancel-in-progress: true**

#### When should you use artifacts? (Select two.)

1. **Use artifacts to save files produced by a job to view after a workflow run has ended, such as test results or build logs.**
2. Use artifacts to create new versions of your application together with release notes, mentions and/or contributors
3. **Use artifacts to save binaries produced by a build job to use in a subsequent deploy job to deploy a new version of an application**
4. Use artifacts to reuse files that don't change often between jobs or workflow runs, such as build dependencies from a package management system.

#### Secrets and configuration variables can be scoped to: (Select three.)

1. A specific workflow in a repository
2. A specific job in a workflow
3. Multiple repositories that do not share an organization/enterprise
4. An environment shared across multiple repositories
5. **A single repository**
6. **An environment in a repository**
7. **The entire organization, or selected repositories in an organization**

#### How many jobs will be executed in the following workflow?

jobs:

matrix-job:

runs-on: ubuntu-latest

strategy:

matrix:

pet: [cat, dog]

color: [pink, brown]

include:

- color: white

pet: dog

steps:

- run: echo "Hello ${{ matrix.color }} ${{ matrix.pet }}"

1. 7
2. 6
3. 4
4. **5**

#### What is the purpose of the restore-keys parameter in actions/cache in GitHub Actions?

1. enable cross-OS cache functionality
2. indicate whether a cache hit occurred
3. specify the location of the cached files
4. **provide alternative keys to use in case of a cache miss**

#### What are the different permission levels you can assign to GITHUB\_TOKEN in the permissions block?

1. **none, write, read**
2. read, write
3. read, write, delete

#### When will job3 run?

jobs:

job1:

job2:

needs: job1

job3:

if: ${{ always() }}

needs: [job1, job2]

1. job3 will run after job1 and job2 have been successfully completed
2. job3 will run after both job1 and job2 have failed
3. You cannot use if: ${{ always() }} and needs together. The workflow will fail on startup.
4. **job3 will run after job1 and job2 have completed, regardless of whether they were successful**

#### Which of the following statements are true about adding a self-hosted runner in GitHub Actions? (Choose three.)

1. **You can add a self-hosted runner to an organization**
2. You can add a self-hosted runner to a step
3. You can add a self-hosted runner to a workflow
4. **You can add a self-hosted runner to a repository**
5. **You can add a self-hosted runner to an enterprise**

#### Which of the following are default environment variables in GitHub Actions? (Select three.)

1. **GITHUB\_REPOSITORY**
2. GITHUB\_ORGANIZATION
3. **GITHUB\_WORKFLOW**
4. GITHUB\_TOKEN
5. GITHUB\_USER
6. **GITHUB\_ACTOR**

#### Where should you store sensitive data such as passwords or certificates that will be used in workflows

1. vault
2. **secrets**
3. config variables
4. environment variables

#### Is this statement true? Workflows can be reused, but a reusable workflow cannot call another reusable workflow.

1. **False**
2. True

#### To prevent a job from failure when one of the steps fails you can include:

1. failure() conditional in the failing step

steps:

- uses: my-org/failing-action@v1

if: failure()

1. ignore-error flag in the failing step

steps:

- uses: my-org/failing-action@v1

ignore-error: true

1. **continue-on-error flag in the failing step**

**steps:**

**- uses: my-org/failing-action@v1**

**continue-on-error: true**

1. always() conditional in the failing step

steps:

- uses: my-org/failing-action@v1

if: always()

#### What level of access is required on a GitHub repository in order to delete log files from workflow runs?

1. owner
2. admin
3. **write**
4. read

#### Which of the following default environment variables contains the name of the person or app that initiated the workflow run?

1. GITHUB\_WORKFLOW
2. GITHUB\_USER
3. **GITHUB\_ACTOR**
4. GITHUB\_REPOSITORY

#### A workflow was initially run on commit A and failed. You fixed the workflow with the subseQuent commit B. When you re-run that workflow it will run with code from which commit?

1. It will run with code from commit B
2. **It will run with code from commit A**
3. It will trigger two workflows, one with code from commit A and one with code from commit B
4. You cannot re-run workflows in GitHub Actions. You have to trigger a new workflow which will run with latest changes

#### How can you validate that your GitHub self-hosted-runner can access all required GitHub services?

1. By trying to access the runner machine by ssh to validate the network connectivity
2. By using the predefined GitHub Actions workflow network-connectivity.yml
3. **Using a GitHub provided script on the runner machine**
4. GitHub will validate the network connectivity automatically when the runner application is installed on the runner machine

#### What is the purpose of the timeout-minutes keyword in a step?

1. it sets the timeout for waiting on external events before proceeding to the next step
2. it specifies the maximum duration a job is allowed to run
3. it defines the time interval for individual commands within a step
4. **it limits the execution time for individual step**

#### This workflow will run on all pull requests where:

on:

pull\_request:

branches:

- 'release/\*\*'

- '!release/\*\*-alpha'

1. the target branch name starts with release
2. the source branch name starts with release
3. **the target branch name starts with release but does not end with -alpha**
4. the source branch name starts with release but does not end with -alpha

#### Q73: You can use permissions to modify the GITHUB\_TOKEN permissions on: (Select two.)

1. Step level
2. **Workflow level**
3. **Job level**

#### How can you access the current values of variables in a matrix within a job in the example below:

jobs:

example\_matrix:

strategy:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

1. **reference variables through the matrix context with syntax likematrix.version and matrix.os**
2. by using the matrix.property syntax
3. by using the context keyword within the job configuration
4. by accessing the variables directly with the syntax version and os

#### In a workflow with multiple jobs the default behavior is:

1. Jobs run in seQuence
2. **All jobs run in parallel**

#### What is a self-hosted runner?

1. A self-hosted runner is a system to be able to create workloads automatically
2. A self-hosted runner is a system to upload code to a private server
3. **A self-hosted runner is a system that you deploy and manage to execute jobs from GitHub Actions on GitHub.com**
4. A self-hosted runner is a system to manage pull requests from users of the organization

#### What components can be reused within a GitHub Organization? (Select four.)

1. Environment Variables
2. **Workflow Templates**
3. **Secrets**
4. Artifacts
5. **Self Hosted Runners**
6. Cache
7. **Configuration Variables**

#### What are the three types of Actions?

1. **Docker container actions, JavaScript Actions, Composite Actions**
2. Docker container Actions, JavaScript Actions, Custom Actions
3. Python Actions, JavaScript Actions, Custom Actions
4. Docker container actions, Java Actions, Composite Actions

#### In a GitHub Actions workflow, how do you specify a specific version of Node.js to use in a job?

**uses: actions/setup-node@v4**

**with:**

**node-version: 20**

uses: setup-node@v4

with:

version: 20

uses: setup-node@v4

with:

node: 20

uses: actions/node-setup@v4

with:

node-version: 20

#### Select the default environment variable that contains the operating system of the runner executing the job

1. **RUNNER\_OS**
2. GITHUB\_RUNNER\_OS
3. RUNNER\_NAME
4. RUNNER\_ARCH

#### Which of the following default environment variables contains the full name (e.g octocat/hello-world) of the repository where the workflow is running?

1. GITHUB\_REPOSITORY\_ID
2. GITHUB\_REPOSITORY\_OWNER\_ID
3. **GITHUB\_REPOSITORY**
4. GITHUB\_REPOSITORY\_OWNER

#### Workflows are defined in which format

1. json
2. xml
3. **yaml**
4. toml

#### Which is true about workflows? (Select three.)

1. **Workflows can run one or multiple jobs at a time**
2. **Workflows can be triggered manually, by an event or run on a schedule**
3. Workflows are written in any of .yaml, .json or .toml formats
4. Workflow can run only one job at a time
5. Workflows can only be run on a schedule
6. **Workflows have to be defined in the .github/workflows directory**
7. Workflows can be shared in GitHub Marketplace

#### What GitHub-hosted runner types are available to use? (Select three.)

1. **Windows**
2. **Ubuntu Linux**
3. **macOS**
4. Android

#### What level of permission is required to re-run the workflows

1. owner
2. read
3. admin
4. **write**

#### What are the scopes defined for custom variables in a workflow? (choose three)

1. A specific environment in the repository, by using environment.<environment\_id>.env at the top level of the workflow file
2. **The entire workflow, by using env at the top level of the workflow file**
3. All the jobs within a workflow, by using jobs.env
4. **The contents of a job within a workflow, by using jobs.<job\_id>.env**
5. **A specific step within a job, by using jobs.<job\_id>.steps[\*].env**
6. The entire workflow, by using custom.env at the top level of the workflow file

#### Which matrix job definition is syntactically correct?

**jobs:**

**example\_matrix:**

**strategy:**

**matrix:**

**version: [10, 12, 14]**

**os: [ubuntu-latest, windows-latest]**

jobs:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

jobs:

example\_matrix:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

jobs:

example\_matrix:

matrix:

strategy:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

#### In the following example, workflow A passes all of its secrets to workflow B, by using the inherit keyword. Then workflow B calls workflow C. Which statement regarding secrets is true for that example?

jobs:

workflowA-calls-workflowB:

uses: octo-org/example-repo/.github/workflows/B.yml@main

secrets: inherit

jobs:

workflowB-calls-workflowC:

uses: different-org/example-repo/.github/workflows/C.yml@main

1. All secrets from octo-org organization and octo-org/example-repo repository will be available to workflow B, but not to workflow C
2. Only repository and environment secrets available to workflow A will be available to workflow B, but not to workflow C. Organization scoped secrets cannot be inherited
3. **All secrets available to workflow A will be also available to workflow B, but not to workflow C**
4. All secrets available to workflow A will be also available to workflow B and workflow C

#### In a workflow that has multiple jobs, all running on GitHub-hosted runners, is it true that all jobs are guaranteed to run on the same runner machine?

1. Yes
2. **No**

#### Who can bypass configured deployment protection rules to force deployment (by default)

1. Anyone with repository read permission
2. **Repository administrators**
3. Anyone with repository write permission

#### When should you use caching?

1. When you want to reuse files that do change often between jobs or workflow runs, such as build dependencies from a package management system.
2. **When you want to reuse files that don't change often between jobs or workflow runs, such as build dependencies from a package management system.**
3. When you want to save binaries produced by a build job to use in a subseQuent deploy job to deploy a new version of an application
4. When you want to save files produced by a job to view after a workflow run has ended, such as built binaries or build logs.

#### You can only upload a single file at a time when using actions/upload-artifact action

1. True
2. **False**

#### Which of the following can be used to limit the number of concurrent jobs running in a GitHub Actions workflow?

1. parallelism
2. limit
3. max-jobs
4. **concurrency**

#### Which of these is a proper way of setting an output parameter PET with a value of DOG in a step.

1. echo "DOG=PET" >> "$GITHUB\_OUTPUT"
2. **echo "PET=DOG" >> "$GITHUB\_OUTPUT"**
3. gh set-output "DOG=PET"
4. gh set-output "PET=DOG"

#### How can you reuse a defined workflow in multiple repositories? (Choose two.)

1. By copying the workflow file to each repository
2. **By using workflow templates**
3. By creating a reusable action
4. **By defining the workflow in a central repository**

#### How can you determine if an action is a container action by looking at its action.yml file?

1. runs.main has container as value
2. runs.using has Dockerfile as value
3. runs.using has container as value
4. **runs.using has docker as value**

#### What’s true about default variables? (choose three)

1. **Default environment variables are set by GitHub and not defined in a workflow**
2. You can add a new default environment variable adding the prefix “GITHUB\_” to it
3. Currently, the value of the default CI environment variable can be overwritten, but it's not guaranteed this will always be possible
4. **Default environment variables can be accessed using the env context**
5. Most of the default environment variables have a corresponding context property
6. **Default environment variables always have the prefix “GITHUB\_”**

#### To access an artifact that was created in another, previously triggered workflow run you can:

1. Use the actions/upload-artifact action.
2. **You cannot access artifacts that were created in a different workflow run**
3. Use the actions/download-artifact action.
4. Use the actions/download-artifact action and make sure the artifact is not expired

#### What are the possible types of an input variable for a manually triggered workflow? (Select five.)

1. environment
2. **boolean**
3. **choice**
4. dropdown
5. select
6. **number**
7. **string**

#### How can you override an organization-level GitHub Secret API\_KEY with a different value when working within a repository? (Select two.)

1. By creating a enterprise secret with the same name API\_KEY
2. By creating a environment secret with the name OVERRIDE\_API\_KEY
3. **By creating a environment secret with the same name API\_KEY**
4. By creating a repository secret with the name OVERRIDE\_API\_KEY
5. **By creating a repository secret with the same name API\_KEY**
6. By creating a repository secret with the name REPOSITORY\_API\_KEY
7. By creating a enterprise secret with the name OVERRIDE\_API\_KEY
8. By creating a environment secret with the name ENVIRONMENT\_API\_KEY

#### Which of these is a way of using action\_state in step\_two?

steps:

- name: Set the value

id: step\_one

run: |

echo "action\_state=yellow" >> "$GITHUB\_ENV"

- name: Use the value

id: step\_two

run: ?

1. **run: echo "$action\_state"**
2. run: echo "${{ action\_state }}"
3. run: echo "${{ steps.step\_one.outputs.action\_state }}"
4. run: echo "$steps.step\_one.outputs.action\_state"

#### Which configuration is appropriate for triggering a workflow to run on webhook events related to check\_run actions?

**on:**

**check\_run:**

**types: [rerequested, completed]**

on:

check\_run:

types: [started]

on:

check\_run:

type: [closed]

on:

check\_run:

filter: [requested]

#### Which variable would you set to true in order to enable step debug logging?

1. ACTIONS\_JOB\_DEBUG
2. **ACTIONS\_STEP\_DEBUG**
3. ACTIONS\_WORKFLOW\_DEBUG
4. ACTIONS\_RUNNER\_DEBUG

#### What is the correct syntax for setting the directory for all run commands in a workflow?

1. **set working-directory under defaults.run**

**defaults:**

**run:**

**shell: bash**

**working-directory: ./scripts**

1. set working-directory under job

defaults:

run:

shell: bash

job:

working-directory: ./scripts

1. set directory under job

defaults:

run:

shell: bash

job:

directory: ./scripts

1. set directory under defaults.run

defaults:

run:

shell: bash

directory: ./scripts

#### How can you ensure a job runs only on a specific branch?

1. By using the jobs filter
2. **By using the branches filter**
3. By using the branch keyword
4. By using the runs-on filter

#### If a workflow runs on a feature-a branch, can it restore caches created in the default main branch?

1. Yes but only if no files were changed on feature-a branch
2. **Yes, all branches can restore caches created on the default branch**
3. No, caches can only be restored from the same branch
4. Yes, all caches can be accessed by workflows on any branch within the same repository

#### What is the purpose of the with keyword in a GitHub Actions workflow?

1. To set up dependencies
2. To trigger another workflow
3. To define environment variables
4. **To specify input parameters for an action**

#### What is the default timeout for a GitHub Actions job?

1. **360 minutes**
2. 30 minutes
3. 60 minutes
4. 120 minutes

#### What is the recommended practice for treating environment variables in GitHub Actions, regardless of the operating system and shell used?

1. **treat environment variables as case-sensitive**
2. depend on the behavior of the operating system in use
3. use only uppercase letters for environment variable names
4. ignore case sensitivity as GitHub Actions handles it automatically

#### Which components are required for a workflow? (Select two.)

1. **One or more events that will trigger the workflow**
2. Workflow name
3. Defined branches on which the workflow will run
4. **One or more jobs**

#### Is this statement true? Not all steps run actions, but all actions run as a step

1. False
2. **True**

#### A reusable workflow named build creates zip file artifacts. How do you pass the zip file location to the caller workflow that is calling the build workflow? (Select three.)

1. All outputs are automatically passed to the caller workflows
2. **You define an output on job level in the build workflow**
3. **In the build workflow you write the output into $GITHUB\_OUTPUT in one of the steps**
4. **You define an output on workflow level in the build workflow**

#### Are GitHub Actions free for public repositories?

1. No
2. **Yes**

#### Which of the following is a correct statement about GitHub Workflows and Actions?

1. Each workflow is composed of one or more action which is composed of one or more jobs, and each job is composed of one or more step
2. Each action is composed of one or more job which is composed of one or more step, and each step is a workflow
3. Each action is composed of one or more workflows which is composed of one or more jobs, and each job is composed of one or more step
4. **Each workflow is composed of one or more job which is composed of one or more step, and each step is an action or a script**

#### This code will launch 6 different jobs in parallel using the matrix strategy. Can you use the matrix strategy to parallelize entire workflows?

jobs:

example\_matrix:

strategy:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

1. Yes
2. **No**

#### What should you use to store coverage reports or screenshots generated during a workflow that runs automated testing for a repository?

1. **Artifacts**
2. Packages
3. Caches
4. Releases

#### Which statement is correct regarding passing permissions to reusable workflows?

1. **The GITHUB\_TOKEN permissions passed from the caller workflow can be only downgraded by the called workflow.**
2. The GITHUB\_TOKEN permissions passed from the caller workflow can be only elevated by the called workflow.
3. The GITHUB\_TOKEN permissions passed from the caller workflow can be both downgraded and elevated by the called workflow.
4. The GITHUB\_TOKEN permissions passed from the caller workflow can be neither downgraded or elevated by the called workflow.

#### What does the needs keyword do in a GitHub Actions workflow?

1. Defines environment variables
2. **Specifies the dependencies of a job**
3. Triggers a job based on an event
4. Sets up the environment

#### How do you access matrix variables in a matrix strategy job?

1. Using the jobs context
2. Using the vars context
3. Using the job context
4. **Using the matrix context**

#### Which is true about Starter Workflows ? (Select three.)

1. **Your organization can create custom starter workflows for users in your organization**
2. Starter workflows are provided ready-to-use and cannot be modified or enhanced
3. **GitHub provides and maintains starter workflows for different categories, languages and tooling**
4. Starter workflows cannot call reusable workflows
5. Starter workflows are a paid GitHub feature
6. **They allow users to leverage ready-to-use (or reQuiring minimal changes) workflow templates**

#### When using the pull\_request and pull\_request\_target events, how do you configure the workflow to run only when targeting the prod branch?

1. Using branch filter
2. **Using branches filter**
3. Using glob patterns
4. You create the workflow only on prod branch

#### In GitHub Actions, if you define both branches and paths filter, what is the effect on the workflow execution?

1. the workflow will run when either branches or paths are satisfied
2. the workflow will run when either branches or paths are satisfied, but will only apply the matching filter
3. **the workflow will only run when both branches and paths are satisfied**
4. the workflow will not run when both branches and paths are satisfied

### Set 2

#### How many PowerShell commands are executed on the Windows runner with the following workflow configuration?

name: Run commands Windows

on:

push:

branches: [ main ]

 jobs:

Run-PSScriptAnalyzer-on-Windows:

name: Run PSScriptAnalyzer on Windows

runs-on: windows-latest

steps:

- uses: actions/checkout@v4

- name: Install PSScriptAnalyzer module

shell: pwsh

run: |

Set-PSRepository PSGallery -InstallationPolicy Trusted

Install-Module PSScriptAnalyzer -ErrorAction Stop

- name: Get list of rules

shell: pwsh

run: |

Get-ScriptAnalyzerRule

1. **3**
2. 2
3. 4
4. 1

#### Your organization uses various custom actions and scripts within GitHub Actions workflows across projects. To enhance collaboration and manage components, which file and folder naming convention approach would be most beneficial?

1. use random names or abbreviations for shorter filenames
2. stick to existing platform/language conventions without requiring organization-specific guidelines
3. allow individual teams to set their own naming conventions for their reusable components
4. **implement and enforce an organization-wide naming convention that clearly identifies the component type, purpose, and version**

#### How should you demonstrate creating a release strategy for a GitHub Action?

1. by periodically updating the action without any versioning
2. by making all updates immediately available without any documentation
3. **by defining a release-management strategy and documenting major version updates, critical fixes, and security patches**
4. by maintaining a clear changelog but not versioning the updates

#### Which of the following are true about Javascript actions? (select three)

1. **Javascript actions can run on Linux, Windows, or macOS runners**
2. **You can speed up development by using the GitHub Actions toolkit**
3. **Javascript actions run directly on the runner and will use existing binaries**
4. You should include binaries with the Javascript actions code to simplify the workflow

#### Your organization uses GitHub Actions in Enterprise Cloud and wants to ensure automation is reused and maintained when creating new workflows in the organization's repositories. What feature should be used?

1. contribution guidelines
2. **workflow templates**
3. GitHub wiki
4. naming conventions

#### What information is essential when drafting a new release and publishing an action to GitHub Marketplace?

1. the action’s metadata file must be in the root directory of the repository
2. the release title should be left blank for automatic generation
3. two-factor authentication is not required for publishing releases
4. **the action’s metadata file’s category must match an existing GitHub Marketplace category**

#### As the lead developer of the GitHub Action library, you are concerned about the reliability of versioning using tags. What is a recommended alternative to ensure consistency and security in versioning?

1. use abbreviated commit SHAs instead of full SHAs for versioning to simplify tracking
2. **implement commit SHAs for versioning to ensure reliability and security**
3. utilize branch names instead of tags for versioning to prevent potential inconsistencies
4. continue to use tags but avoid deletions or movements to maintain consistency

#### What happens if a job is not approved within 30 days while awaiting review in a workflow?

1. the job with be put on hold indefinitely until approved
2. the job will remain in the “Waiting” status until approved
3. **the job will automatically fail**
4. the job will automatically start without approval

#### Dave is building a workflow configuration that includes a reference to a filesystem path. What is the best practice for referencing a filesystem within a workflow configuration?

1. use relative paths to reference the filesystem
2. filesystem references should never be included in a workflow configuration
3. **use environment variables to reference the filesystem**
4. hard-coded file paths to ensure the correct path

#### In a private repository, why are workflow badges not accessible externally?

1. badges are only visible to repository collaborators
2. private repositories have disabled the badge visibility feature by default
3. it is a limitation in GitHub Actions
4. **to prevent external embedding or linking from unauthorized sources**

#### How can the retention period for artifacts be customized?

1. retention periods can only be configured at the organization level
2. custom retention periods are automatically applied to all repositories
3. the retention period for artifacts cannot be customized
4. **custom retention periods can be defined for individual artifacts using the actions/upload-artifact action**

#### How can encrypted secrets be accessed within actions and workflows for GitHub Actions?

1. using the actions/secrets GitHub Actions package to decrypt and access encrypted secrets programmatically within workflows
2. directly referencing the secret's name within the workflow file, which automatically decrypts the secret at runtime
3. **using the secrets context within GitHub Actions, which allows encrypted secrets to be accessed as environment variables**
4. embedding the encrypted secret directly within the workflow file, which GitHub Actions automatically decrypts during execution

#### GitHub Packages is compatible with the following package managers: (select three)

1. RPM, a package manager for linux distributions
2. **npm, a NodeJS package manager**
3. **Maven and Gradle, two package managers for Java**
4. **NuGet, the .NET package manager**

#### What is the primary purpose of caching dependencies in a GitHub Actions workflow?

1. automate the creation of workflows
2. eliminate the need for GitHub-hosted runners
3. **decrease network utilization, runtime, and cost**
4. reduce the size of the workflow YAML

#### What status should you filter on to see only failed workflow runs on the GitHub Actions tab?

1. failed
2. completed
3. **failure**
4. errored

#### Your team manages its own infrastructure costs using a chargeback model and wants to ensure that development workflows do not utilize the runners paid for by your team. Which GitHub Actions feature can help achieve this goal?

1. runner environments
2. runner labels
3. runner sets
4. **runner groups**

#### You are trying to run a new Docker container action but getting a permission denied error when running the entrypoint.sh script. How can you resolve this?

1. **modify the entrypoint.sh script to explicitly set executable permissions before running**
2. grant executable permissions to the entrypoint.sh script using the chmod command before running the Docker container action
3. update the Dockerfile to include the appropriate permissions for the entrypoint.sh script
4. adjust the Docker container action's configuration to specify a different entrypoint.sh script

#### Why is it important to avoid passing secrets between processes from the command line?

1. passing secrets through the command line is actually recommended practice
2. command-line processes cannot capture security audit events
3. command-line processes automatically redact any secrets they handle
4. **passing secrets through the command line may expose them to other users and security audits**

#### You are building a new custom action and must pass data from one step to subsequent steps in a GitHub Actions workflow. Which key should you use in the action's metadata syntax?

1. description
2. runs
3. environment
4. **outputs**

#### How can you ensure a script file in your repository is executable in a workflow job?

1. by specifying the script file path in the workflow configuration file
2. by embedding the script directly within the workflow configuration file and marking it as executable using a special flag
3. by using a special keyword in the workflow YAML to mark the script as executable
4. **by manually granting execute permissions to the script file on the runner**

#### Which of the following statements accurately describes the syntax rules for indentation in YAML used for defining workflow jobs in GitHub Actions?

1. YAML strictly follows the JSON syntax, allowing only spaces for indentation and prohibiting the use of newlines
2. YAML permits the use of literal tab characters for indentation, similar to Python
3. YAML syntax rules are identical to Python, including the allowance of literal tab characters for indentation
4. **YAML allows significant newlines and indentation, similar to Python, but unlike Python, it prohibits the use of literal tab characters for indentation**

#### You encounter an issue while executing a GitHub Actions workflow and need to print a debug message to the log for troubleshooting. How can you accomplish this?

1. logs ::debug "executing the setup script"
2. log ::warning "executing the setup script"
3. **echo "::debug::executing the setup script"**
4. you cannot print a debug message to the log in GitHub Actions

#### Why does GitHub recommend using variables to access the filesystem instead of hardcoded file paths?

1. GitHub does not support the use of hardcoded file paths in workflow files
2. **variables provide a dynamic way to adapt to different runner environments**
3. hardcoded file paths may lead to conflicts with default environment variables
4. GitHub Actions imposes restrictions on hardcoded file paths

#### Which default environment variables cannot be overwritten using the GITHUB\_ENV file in a workflow?

1. CI
2. all default environment variables can be overwritten
3. **GITHUB\_\* and RUNNER\_\***
4. NODE\_OPTIONS

#### What are valid ways to specify the version of the checkout action within a GitHub workflow configuration? (select three)

1. **- uses: actions/checkout@main**
2. **- uses: actions/checkout@8f4b7f84864484a7bf31766abe9204da3cbe65b3**
3. - uses: https://github.com/actions/checkout
4. **- uses: actions/checkout@v4**

#### What is the minimum time granularity available for scheduling GitHub Actions?

1. one hour
2. **five minutes**
3. one day
4. thirty seconds

#### How can you access an environment variable corresponding to an input in a Docker container action?

1. use the process.env.INPUT\_<VARIABLE\_NAME> syntax within the action's code
2. define the input as a command-line argument when running the Docker container
3. **use the args keyword in the action metadata file to pass the input to the Docker container**
4. the environment variable is automatically accessible within the Docker container without any additional configuration

#### What is the GITHUB\_TOKEN secret used for in a workflow?

1. **to authenticate on behalf of GitHub Actions**
2. to trigger workflow runs manually
3. to configure repository settings
4. to store sensitive information like API keys

#### In the context of actions and workflows, what roles do steps play in the overall process?

1. they refer to the overall execution of actions
2. **they represent individual tasks within a job**
3. they are synonymous with workflows
4. they are part of the marketplace integration process

#### Which keyword is used to conditionally execute a step based on a specific expression or condition?

1. only
2. condition
3. when
4. **if**

#### John has configured his workflow to save artifacts created from the build job. Where can John access the artifacts from the GitHub user interface that were saved within the build job?

1. from the pull request associated with the workflow run
2. **from the Artifacts section within the Actions workflow run**
3. from the job details of the build job
4. from the Releases section in the repository

#### What additional steps does GitHub add to each job in a workflow run?

1. GitHub does not add any steps to a job.  All steps must be configured in the workflow file.
2. "Set up job" and "Tear down job"
3. **"Set up job" and "Complete job"**
4. "Checkout" and "Post-job cleanup"

#### Which configuration is appropriate for triggering a workflow on a pull request?

**on:**

**pull\_request:**

**branches:**

**- main**

on:

pull\_request:

types:

- main

1. on: push
2. on: fork

#### Your operations team plans to use GitHub-hosted runners for continuous integration tasks, but the security team insists on implementing an IP address allowlist to enhance security measures. Why might this approach be considered cumbersome for your operations team?

1. managing the IP address allowlist would introduce additional administrative overhead for the operations team
2. **having to update the allowlist for GitHub-hosted runner IP addresses on a weekly basis could be time-consuming and error-prone**
3. ensuring compatibility with services lacking static IP addresses could be challenging, impacting workflow flexibility.
4. coordination with the security team might cause delays in setting up and scaling CI workflows

#### You have created a secret named api\_key to use in a workflow that deploys a new application. Which of the following is the correct syntax to reference the secret as an environment variable?

steps:

- shell: bash

env:

ENV\_API\_KEY: ${{ secrets.environment.api\_key }}

run: |

./app\_install.sh

**steps:**

**- shell: bash**

**env:**

**ENV\_API\_KEY: ${{ secrets.api\_key }}**

**run: |**

**./app\_install.sh**

steps:

- shell: bash

with:

ENV\_API\_KEY: ${{ api\_key }}

run: |

./app\_install.sh

steps:

- shell: bash

with:

ENV\_API\_KEY = api\_key

run: |

./app\_install.sh

#### You have created a GitHub Action that requires a specific Linux operating system and custom tools to run. What type of action is best suited for this environment?

1. Composite Action
2. **Docker container action**
3. JavaScript action
4. Actions cannot be built using customized operating systems and tools

#### Which of the following events can trigger workflows? (select three)

1. **when a commit is pushed to the repository**
2. when somebody is invited to a GitHub repository
3. **when a GitHub issue is created**
4. **when a discussion is created**

#### What is the primary purpose of dependent jobs in a workflow?

1. to control the workflow’s overall concurrency settings
2. to run concurrently and independently of each other
3. **to define sequential execution order within a workflow**
4. to enforce conditional logic based on job outcomes

#### You have a workflow configured to run on branch protection rule events in your GitHub repository. However, you want to limit the workflow execution to exclude the deletion of branch protection rules. Which configuration should you use in your workflow file?

on:

branch\_protection\_rule:

types: [created, edited, deleted]

on:

branch\_protection\_rule:

types: [deleted]

**on:**

**branch\_protection\_rule:**

**types: [created, edited]**

on:

branch\_protection\_rule:

notTypes: [deleted]

#### What is required to manually run a private repository’s workflow using the GitHub REST API?

1. **Personal Access Token**
2. SSH Key
3. API Key
4. Username and Password

#### What directory is used to store workflow files within a code repository?

1. actions/yaml
2. actions/workflows
3. **.github/workflows**
4. .github/actions/workflows

#### What is the filename of the metadata file that defines the inputs, outputs, and runs configuration for your action?

1. config.yaml
2. **action.yaml**
3. requirements.yaml
4. workflow.yaml

#### What are the requirements to publish an action to the GitHub Marketplace?

1. each repository must contain multiple actions
2. the action's metadata file must be in a subdirectory of the repository
3. **the action must be in a public repository**
4. the name in the action's metadata file must match an existing GitHub Marketplace category

#### Ryan is looking for the GitHub Actions workflow files for his repository. Where should he look?

1. the repository's Settings >> Actions tab page on GitHub
2. the .github directory of the repository
3. **the .github/workflows directory of the repository**
4. the README.md file of the repository

#### When creating a custom action for GitHub Actions, which of the following files are required? (select three)

1. **main.js or index.js for JavaScript actions**
2. script.py for Python actions
3. **action.yml or action.yaml for action metadata**
4. **Dockerfile for containerized actions**

#### You're using ephemeral runners in containers for your GitHub Actions workflows. However, you've noticed that these runners repeatedly update whenever a new runner version is released, causing disruptions. What action can you take to address this issue?

1. schedule software updates using GitHub's integrated update scheduler
2. configure workflows to handle software updates for the runners as needed
3. enable automatic updates to ensure continuous runner software updates
4. **disable automatic updates to have manual control over updating the runner version on the container image**

#### Which API does GitHub Actions use to output statuses, results, and logs for a workflow?

1. Logs API
2. Health API
3. **Checks API**
4. Actions API

#### When the workflow below is triggered, what will the Print name step evaluate to?

name: Java CI with Maven

env:

NAME: 'My Action'

on:

push:

branches: [ "main" ]

jobs:

build:

env:

JAVA\_VERSION: '11'

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Set up JDK ${{env.JAVA\_VERSION}}

uses: actions/setup-java@v3

with:

java-version: ${{env.JAVA\_VERSION}}

distribution: 'temurin'

cache: maven

- name: Build with Maven

run: mvn -B package --file pom.xml

- name: Print name

run: echo "Hello $NAME. $BUILD. Using Java Version $JAVA\_VERSION"

env:

BUILD: 'We are currently running the Build job'

1. Hello $NAME. $BUILD. Using Java Version $JAVA\_VERSION
2. Hello My Action.  $BUILD  Using Java Version 11
3. **Hello My Action.  We are currently running the Build job.  Using Java Version 11**
4. this run will error because of incorrect syntax

#### What functionality does a composite action offer in GitHub Actions?

1. creating custom Docker containers for isolated workflow execution
2. integrating third-party APIs and services directly into workflows
3. **combining multiple workflow steps into one action for streamlined execution**
4. automating the deployment of applications to cloud platforms like AWS and Azure

#### For an action that was triggered on: pull request, where can you see the workflow run status? (select three)

1. on the Issues tab of the repository
2. **on the Checks tab of the pull request**
3. **in a pull request before a merge**
4. **from the Actions tab of the repository**

#### What is the primary purpose of custom labels in GitHub Actions for self-hosted runners?

1. **routing jobs to specific types of self-hosted runners based on their labels**
2. improving code readability in workflows
3. assigning descriptive names to workflow jobs
4. enhancing security measures for runners

#### After creating a new workflow, GitHub Actions will suggest starter workflows for your repository. What option should you click on if there is a starter workflow that you want to use?

1. Deploy
2. Install
3. Use
4. **Configure**

#### You have developed a new GitHub Action and want to share it with the greater community. Where should you publish it?

1. personal blog or website
2. a private repository
3. a public repository
4. **GitHub Marketplace**

#### You're developing a custom GitHub Action for your organization's CI/CD pipeline. You're considering how to manage versioning for your action. Which practice aligns best with industry standards and simplifies version control?

1. **implement semantic versioning for your tag version to clearly communicate changes and maintain compatibility**
2. assign arbitrary version numbers to each release, ensuring uniqueness across all actions
3. avoid versioning altogether and rely on Git commit hashes to identify the state of the action's codebase
4. utilize descriptive version names based on the release date and time to denote the progression of the action's development

#### Which YAML keyword is used to specify the events that should trigger a workflow?

1. event
2. trigger
3. workflow
4. **on**

#### How many jobs will be executed in the following workflow?

jobs:

matrix-job:

runs-on: ubuntu-latest

strategy:

matrix:

animal: [cat, dog, bear]

color: [black, brown]

steps:

- run: echo "Hello ${{ matrix.color }} ${{ matrix.animal }}"

1. **6**
2. 0
3. 2
4. 3

#### What capability does GitHub provide to enable runners to download actions from internal or private repositories, ensuring access control and security?

1. GitHub enables direct access to the repository for the duration of the workflow run
2. **GitHub creates a scoped installation token with read access to the repository, automatically expiring after one hour**
3. GitHub generates a personal access token with read access to the repository, valid for one hour
4. GitHub prompts users to authenticate with their GitHub credentials each time an action is downloaded

#### When might it be appropriate to use a combination of GitHub-hosted and self-hosted runners in a workflow?

1. **when dealing with resource-intensive tasks**
2. when working on a personal project with minimal dependencies
3. when executing short-lived, stateless jobs in isolation
4. when aiming for the highest level of security

#### Your organization uses a self-hosted runner deployed within a network that requires a proxy server for internet access. Which environment variable should you configure on the runner to ensure it can successfully communicate with GitHub?

1. network\_proxy
2. outbound
3. **https\_proxy**
4. proxy\_server

#### You want to limit the use of public actions and reusable workflows so that people can only use reusable workflows in your enterprise. Where would this be configured?

1. In the shared repository settings, using the setting Disable GitHub Actions
2. By setting the token policies of a personal access token
3. **In the Policies section for the targeted enterprise for your organization**
4. Under the Policies section of the GitHub Codespaces page

### Set 3

#### Which keyword within a GitHub workflow configuration is used to match a triggering event?

1. **on:**
2. event:
3. run:
4. when:

#### What level of access is required to download workflow artifacts?

1. admin
2. write
3. **read**
4. owner

#### How can you specify dependencies between jobs in a workflow?

1. by using the depends-on keyword in each job definition
2. **by defining dependencies in the workflow YAML file**
3. by setting environment variables between jobs
4. by organizing jobs into separate workflow files

#### Heather is troubleshooting a failure on one of her GitHub Actions workflow runs for recent pull request.  Where can she view the logs to determine why the run may have failed? (select two)

1. in the "Issues" tab of the repository
2. **in the "Checks" tab of a pull request**
3. in the "Insights" tab of the repository
4. **in the "Actions" tab of the repository**

#### Rather than using code to create an error annotation, what can you use to send commands to the runner to create the same error annotation?

1. python scripts that create the specific error messages required
2. the set-env command
3. environment variables prepended with the RUNNER\_\* annotation
4. **workflow commands provided by the actions/toolkit**

#### What action should be taken if you want to find the expiration date of a specific artifact?

1. the expiration date of an artifact cannot be determined in GitHub Actions
2. use the “Artifacts” tab on the GitHub repository page
3. **execute a specific API call to retrieve the expiration date**
4. navigate to the “Actions” tab, click on the workflow, and check the summary for expiration information

#### Where can you set custom environment variables in a workflow?

1. in your profile settings
2. only in command-line arguments
3. in the main code file
4. **in the workflow file**

#### What is the purpose of a job and its associated steps when using job steps for actions and shell commands?

1. **define a specific task or unit of work with a sequence of steps**
2. specify the repository owner and collaborators
3. configure the webhook events triggering the workflow
4. define the overall workflow structure

#### Which context property can be used to access information about the event that triggered a workflow run?

1. jobs.<job\_id>.result
2. github.job
3. github.repository
4. **github.event**

#### As you prepare to distribute your custom GitHub Action, what best practice should you follow to enhance its visibility and utility for potential users?

1. **offer a clear description of what the action accomplishes and select the most relevant category to accurately represent its utility**
2. provide a brief description of the action's functionality and select multiple categories to maximize its exposure
3. include an extensive list of features in the description to attract a wider audience.
4. keep the description minimal to encourage exploration and experimentation by users

#### You are planning to manage reusable workflows for your organization within GitHub Actions. Which approach is recommended for optimal organization and maintainability?

1. **create a dedicated repository to store and manage all reusable workflows**
2. utilize the workflow\_call trigger within workflows to call reusable workflows from other repositories
3. reuse workflows directly from individual project repositories.
4. implement version control for reusable workflows using branches and tags

#### How does the cache action in a GitHub Actions handle a cache miss?

1. **by automatically creating a new cache if the job is completed successfully**
2. by requiring manual intervention to create a new cache
3. by searching for a cache in other repositories
4. by terminating the workflow if a cache miss occurs

#### How can you add the action you've created to the GitHub Marketplace?

1. **tagging it as a new release and then publishing it**
2. merging a pull request from another user
3. creating a new branch in your repository
4. renaming the repository to match an existing GitHub feature

#### You need to create a new action metadata file. What syntax should you use?

1. Python
2. Javascript
3. **YAML**
4. JSON

#### What is the primary purpose of using workflow commands as a run step in a GitHub Actions workflow?

1. to trigger the workflow to move to the next step
2. **to communicate instructions and information to the runner environment**
3. to define environment variables for the entire workflow
4. to execute custom scripts on the runner

#### What is one of the main reasons for hosting a GitHub Action in a separate repository when making it public?

1. **it makes it easier for developers to extend and fix issues with the action**
2. it reduces the need for version control
3. it increases the complexity of managing the action
4. it limits the visibility of the action to the repository where it's hosted

#### What essential step is involved in deploying a release to a cloud provider using a GitHub Actions workflow?

1. avoiding the use of environment variables during the deployment process
2. **defining deployment steps in the GitHub Actions workflow YAML file**
3. configuring multiple workflows in separate repositories
4. excluding the workflow YAML file from the repository

#### Your organization primarily runs its workloads on the Windows operating system and wants to start using Docker for building and testing jobs as it rearchitects its primary applications. What requirements must be met before using Docker container actions?

1. install and configure Docker on the developer's local machines
2. upgrade all GitHub repositories to use the Windows operating system as the primary environment
3. **the organization must use runners with a Linux operating system and have Docker installed**
4. the Windows-based runners must have Docker installed

#### John is troubleshooting a failed workflow run and would like to view the workflow file associated with the failed run.  What option can he select within the failed run's menu to easily view the workflow file?

1. **View workflow file**
2. Download log archive
3. View raw logs
4. Create status badge

#### How can you customize the schedule of a GitHub actions workflow to run on weekdays only?

1. add a condition in the workflow YAML for weekdays
2. **use the on: schedule: cron syntax**
3. specify the schedule in the repository settings
4. utilize the on: schedule: weekdays configuration

#### You are working on a project within your organization that requires a custom GitHub Action. However, due to the sensitive nature of the project, you cannot make the action public. What approach should you take to ensure you can still utilize this action within your project?

1. utilize a separate public repository to host the custom action and grant access to specific users or organizations
2. **allow GitHub Actions workflows in your private repository to access another private repository containing the custom action**
3. copy the action code into your project's repository and include it directly in your workflow files
4. convert the custom action into a Docker container and store it in a private container registry.

#### On GitHub-hosted runners, what is recorded in the "Set up job" step of a given job? (select three)

1. **operating system**
2. **GITHUB\_TOKEN permissions**
3. **runner image**
4. code vulnerability scan

#### While executing a GitHub Actions workflow, you encounter an issue where one of the actions fails unexpectedly. How does GitHub interpret the exit code of an action?

1. GitHub treats all exit codes, regardless of their value, as failures, leading to the termination of the workflow run
2. GitHub interprets a nonzero exit code as success, allowing the workflow to continue without interruption
3. **GitHub interprets a zero exit code as success, indicating that the action was completed successfully and other tasks can proceed**
4. GitHub ignores the exit code of actions and relies solely on manual intervention to determine the success or failure of a workflow run

#### Which of the following statements accurately describes the behavior of workflow jobs referencing an environment's protection rules?

1. workflow jobs will start immediately, regardless of the environment's protection rules
2. workflow jobs will only start if some of the environment's protection rules pass
3. **workflow jobs won't start until all of the environment's protection rules pass**
4. workflow jobs will never start if the environment has protection rules

#### April is in charge of auditing the operations team. While conducting a review, she noticed that many workflows are accessing secrets to carry out deployment and testing functions and is concerned that these secrets may appear in logs. What information can you provide to alleviate April's concerns about workflow logs?

1. **GitHub automatically redacts secrets printed to workflow logs, replacing them with placeholders**
2. GitHub prohibits the printing of workflow logs entirely to ensure security
3. GitHub prompts users to manually confirm before printing secrets to workflow logs
4. GitHub automatically encrypts secrets before printing them to workflow logs

#### Your company uses GitHub Actions for internal projects. You want to share custom actions and scripts across teams but keep them private. How should you distribute these components?

1. use publicly available actions even if they need modifications
2. store them in a public repository with access control
3. have developers store them directly in project repositories
4. **store them in a private repository with access control**

#### Which of the following default environment variables contains the operating system of the runner executing the job?

1. RUNNER\_ARCH
2. RUNNER\_DEBUG
3. GITHUB\_RUNNER\_OS
4. **RUNNER\_OS**

#### A new self-hosted runner was recently registered with your organization, but you don't see it in the runner group assigned to your team. Why can't you use the new runner?

1. **new runners are automatically assigned to a default group, therefore it needs to be moved to the group used by your team**
2. there might be network connectivity issues preventing the runner from appearing in the assigned group
3. the new runner is still undergoing initialization and configuration processes
4. the permissions for accessing the runner group might not be granted to your team

#### How long does the GITHUB\_TOKEN last, and when does it expire?

1. **after a job finishes or after a maximum of 24 hours**
2. 12 hours, regardless of job completion
3. it does not expire, and its validity is unlimited
4. 48 hours, regardless of workflow events

#### In a workflow requiring review, what action is taken if a job is rejected?

1. the job is put on hold until further notice
2. the job is automatically re-submitted for review
3. **the workflow fails**
4. the workflow proceeds as normal

#### You have created a YAML workflow file for a common task for your team's testing. However, you are unsure where to store it in your organization. Where do the workflow file and associated metadata file need to be placed?

1. in a separate dedicated repository specifically for workflow files
2. **in the .github/workflows directory within the repository where the task will be executed**
3. in the organization's settings, under the Workflows section
4. in the root directory of the repository where the task will be executed

#### What are valid custom actions types within GitHub? (select three)

1. **JavaScript actions**
2. **docker container actions**
3. composable actions
4. **composite actions**

#### What is the recommended approach for storing secrets larger than 48 KB?

1. avoid storing large secrets entirely to ensure security
2. store large secrets directly as repository secrets to avoid limitations
3. secrets larger than 48 KB cannot be stored
4. **use a workaround involving encryption with GPG and storing the decryption passphrase as a secret**

#### You are a developer working on a project hosted on GitHub, and you've created a custom action to automate the process of deploying your application to a staging environment. The action is designed to run in a Docker container and requires several input variables to function correctly. This action could benefit other developers in the GitHub community and want to share it. What steps should you take to share your custom action with the GitHub community?

1. **ensure your repository is public, define the action's inputs, outputs, and environment variables, and publish the action as a Docker container**
2. verify that your repository is public, define the action's inputs, outputs, and environment variables, and publish the action to the GitHub Marketplace
3. confirm that your repository is public, define the action's inputs, outputs, and environment variables, and manually distribute the action's Docker image to interested developers
4. make sure your repository is public, define the action's inputs, outputs, and environment variables, and share the action's code directly in the repository README

#### How can you define a matrix for a job in a GitHub Actions workflow?

1. use the matrix keyword within the runs-on parameter
2. use the variables section within the job definition
3. **use the matrix keyword within the strategy configuration of the job**
4. use the matrix keyword within the workflow configuration

#### You need to create a custom Javascript action for your organization, but there are problems in the actions.yml file in the code snippet shown below. What is the problem with the code?

runs:

using: 'node12'

steps: 'main.js'

1. **The Javascript action does not use the steps argument. Change it and use the main argument instead**
2. The Javascript action requires node20, and the code above references node12
3. The Javascript action does not require Node.js and therefore this action will fail
4. The Javascript action does not use the action.yml file, it requires a index.js file instead

#### When selecting a runner for GitHub Actions that is cost-effective and requires minimal management, which option would be most suitable?

1. **GitHub-hosted runners provided by GitHub**
2. Self-hosted runners on virtual machines managed by the organization
3. Self-hosted runners on physical servers maintained by the organization
4. Cloud-hosted runners from a third-party provider

#### Steve would like to use an action in his GitHub project but wants to validate that it is trustworthy before using it.  What steps can Steve take to verify the integrity of a GitHub action before deciding to use it? (select three)

1. Check the action's GitHub stars
2. **Review the action's action.yml file to make sure the code does what it says it does**
3. **Check if the action is verified in the GitHub Marketplace**
4. **Check if the action is in the GitHub Marketplace**

#### What is a valid workflow configuration to automatically publish a Node.js package to GitHub Packages when a release is published?

name: Node.js Package

on:

release:

types: [published]

jobs:

publish:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with:

node-version: '20.x'

- run: npm ci

- run: npm publish

env:

NODE\_AUTH\_TOKEN: ${{secrets.GITHUB\_TOKEN}}

**name: Node.js Package**

**on:**

**release:**

**types: [published]**

**jobs:**

**publish:**

**runs-on: ubuntu-latest**

**steps:**

**- uses: actions/checkout@v4**

**- uses: actions/setup-node@v4**

**with:**

**node-version: '20.x'**

**registry-url: 'https://npm.pkg.github.com/'**

**- run: npm ci**

**- run: npm publish**

**env:**

**NODE\_AUTH\_TOKEN: ${{secrets.GITHUB\_TOKEN}}**

name: Node.js Package

on:

release:

types: [published]

jobs:

publish:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- run: npm ci

- run: npm publish

env:

NODE\_AUTH\_TOKEN: ${{secrets.GITHUB\_TOKEN}}

name: Node.js Package

on:

release:

types: [published]

jobs:

publish:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with:

node-version: '20.x'

registry-url: 'https://npm.pkg.github.com/'

- run: npm ci

- run: npm publish

#### What is the purpose of the continue-on-error keyword in a GitHub Actions step?

1. it indicates that the step should always be executed, regardless of errors
2. **it allows the workflow to continue even if the step fails**
3. it defines conditions under which a step should be retried
4. it specifies the steps to run only on error conditions

#### In GitHub Actions, if you define both branches and paths filter, what is the effect on the workflow execution?

1. **the workflow will only run when both branches and paths are satisfied**
2. the workflow will run when either branches or paths are satisfied, but will only apply the matching filter
3. the workflow will not run when both branches and paths are satisfied
4. the workflow will run when either branches or paths are satisfied

#### How do actions, workflows, jobs, steps, and runs collaborate in a typical scenario?

1. jobs initiate runs, which consist of actions with individual steps in a collaborative workflow
2. steps initiate actions, which are executed within workflows and result in runs within jobs
3. **actions initiate runs, which consist of jobs that execute workflows with individual steps**
4. runs initiate actions, which consist of steps executed within jobs and workflows

#### How do custom labels determine the eligibility of a self-hosted runner to process a job?

1. labels operate independently, and any matching label makes the runner eligible
2. labels are mutually exclusive, and only one matching label is required
3. **labels are cumulative, and the runner must have all assigned labels to be eligible**
4. labels are automatically assigned based on runner characteristics

#### What is the recommended practice for treating environment variables in GitHub Actions, regardless of the operating system and shell used?

1. **treat environment variables as case-sensitive**
2. ignore case sensitivity as GitHub Actions handles it automatically
3. use only uppercase letters for environment variable names
4. depend on the behavior of the operating system in use

#### What will the value of the NAME variable be when this workflow is triggered?

name: Java CI with Maven

env:

NAME: 'My Action'

on:

push:

branches: [ "main" ]

jobs:

build:

env:

NAME: 'My Action 2'

runs-on: ubuntu-latest

- name: Print name

run: echo "$NAME"

env:

NAME: 'My Action 3'

1. **My Action 3**
2. My Action 2
3. My Action
4. this run will error because of incorrect syntax

#### You need to store a sensitive database password for your organization's GitHub Actions workflows. As an organization owner, how can you create a secure secret accessible only to specific repositories within your organization?

1. allow public access to the main repository and utilize private workflow permissions for individual runs
2. hardcode the password directly within your workflows for ease of access
3. create a repository secret in the main project repository and manually share it with other needed repositories
4. **create an organization-level secret and configure a policy to limit access to only the specific repositories that can use it**

#### If an organization's templated workflow contains secret information such as ${{ secrets.token }}, what needs to be configured before using the workflow?

1. replace the ${{ secrets.token }} with the token's value
2. skip configuring secrets and rely on default values
3. **create a secret named token in your repository**
4. configure the workflow file to remove the secret information

#### Which configuration is appropriate for triggering a workflow on a commit to a feature branch?

**on:**

**push:**

**branches:**

**- 'feature/\*'**

on: push

on:

push:

types:

- 'feature'

on:

commit:

branches:

- 'feature/\*'

#### Jeff is troubleshooting an error within his workflow configuration causing it not to run.  What is wrong with this workflow configuration?

name: learn-github-actions

run-name: ${{ github.actor }} is testing out GitHub Actions 🚀

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by a ${{ github.event\_name }} event."

- run: echo "🐧 This job is now running on a ${{ runner.os }} server hosted by GitHub!"

- run: echo "🔎 The name of your branch is ${{ github.ref }} and your repository is ${{ github.repository }}."

- name: List files in the repository

run: |

ls ${{ github.workspace }}

1. the name: keyword is not allowed under the steps section of the configuration
2. the indentation of the configuration is invalid
3. each run: keyword must have an unqiue step: keyword before it
4. **missing the on: keyword to specify what events will trigger the workflow**

#### What distinguishes JavaScript actions from traditional Node.js projects regarding their development and distribution?

1. JavaScript actions do not require end-to-end testing due to their simplicity
2. JavaScript actions are limited to using only GitHub's APIs and do not integrate with third-party APIs
3. **JavaScript actions involve committing dependent packages alongside the code and can be published as tagged releases to the GitHub Marketplace**
4. JavaScript actions do not support dependent packages and tagged releases

#### What is the primary purpose of service containers in a GitHub Actions workflow?

1. define workflow triggers
2. create Docker images
3. host the workflow steps
4. **provide a simple and portable way to host services needed for testing or operating applications**

#### Jess is looking to programmatically access a set of workflow logs for a public repository.  What pieces of information are required for accessing a set of workflow logs?

1. owner, repo, and authentication token
2. owner, repo and job\_id
3. repo, authentication token, and run\_id
4. **owner, repo and run\_id**

#### What is the purpose of the jobs section in a GitHub Actions workflow?

1. **to organize and define the steps to be executed in parallel or sequentially**
2. to define environment variables
3. to declare the workflow’s triggering events
4. to specify the workflow name

#### Which of the following practices are recommended when managing releases for a GitHub Action using tags? (select three)

1. create and validate a release directly on the main branch before creating the release tag
2. **release major versions with a beta tag to indicate their status**
3. **move the major version tag to point to the Git ref of the current release**
4. **introduce a new major version tag for changes that will break existing workflows**
5. use non-semantic versioning to name the release tags for better clarity

#### You have committed the required files to your GitHub repo for the Docker container action, but the action is failing to run. What could be the issue?

prd-app-repo/

|--dockerfile

|--action.yml

|--script.sh

|--README

1. the README file should be outside of the directory supporting the Docker container action
2. a Docker container action doesn't use the action.yml file, therefore causing it to fail
3. the script.sh file is not referenced in the README file
4. **the file names are case-sensitive, therefore change dockerfile to Dockerfile**

#### Your organization requires IP allowlists to protect internal resources accessed by GitHub Actions workflows. Most of your workflows run on GitHub-hosted runners, with both Windows and macOS needs. How can you achieve this desired security while ensuring workflow reliability?

1. implement self-hosted runners on-premises with specific IP addresses added to the allowlist
2. configure IP allowlists on internal resources to include the entire Azure IP address range, as GitHub-hosted Windows and Ubuntu runners operate within Azure.
3. **utilize large runners with static IP address ranges and add these ranges to the allowlist**
4. create separate workflows for accessing internal resources, using only self-hosted runners and bypassing IP allowlists entirely

#### When setting up a GitHub Actions workflow, which of the following components are required? (select three)

1. directory for storing artifacts
2. **workflow file in YAML format**
3. **trigger events to initiate the workflow**
4. **action metadata file**
5. environment variables for configuring the workflow environment

#### Laura would like to add a step to her workflow configuration that adds the /tmp directory to the PATH on an Ubuntu runner.  Which of the following configurations is valid?

name: my-workflow

on:

push:

branches: [ main ]

jobs:

Set-path-Ubuntu:

name: Add user's local bin to Ubuntu PATH

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- run: echo "/tmp" >> $GITHUB\_ENV

name: my-workflow

on:

push:

branches: [ main ]

jobs:

Set-path-Ubuntu:

name: Add user's local bin to Ubuntu PATH

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- run: echo "/tmp" >> $GITHUB\_STEP\_SUMMARY

name: my-workflow

on:

push:

branches: [ main ]

jobs:

Set-path-Ubuntu:

name: Add user's local bin to Ubuntu PATH

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- run: echo "/tmp" >> $GITHUB\_OUTPUT

**name: my-workflow**

**on:**

**push:**

**branches: [ main ]**

**jobs:**

**Set-path-Ubuntu:**

**name: Add user's local bin to Ubuntu PATH**

**runs-on: ubuntu-latest**

**steps:**

**- uses: actions/checkout@v4**

**- run: echo "/tmp" >> $GITHUB\_PATH**

#### The workflow logs do not provide enough detail to diagnose the problem with the recent Javascript action you completed. What steps should you take to continue troubleshooting the issue?

1. convert the JavaScript action to a Docker container action for more detailed logging and isolation
2. review the JavaScript action's code for potential errors or misconfigurations
3. use the GitHub Actions API to retrieve additional diagnostic information about the failed job
4. **enable debug logging to increase the verbosity of the job's logs**

#### Your organization uses self-hosted runners for GitHub Actions and wants to implement security best practices. How can you control access to specific runners for different repositories across teams?

1. configure individual access policies for each runner within each repository
2. limit public access to repositories and rely on individual user permissions
3. use workflow labels to specify which runners each workflow can utilize
4. **assign runners to groups and grant repository access permissions at the group level**

### Set 4

#### Your development team frequently executes steps involving setting up environments, running tests, and deploying applications across various workflows in GitHub Actions. However, configuring these steps individually in each workflow has become cumbersome and repetitive. Which feature of GitHub Actions can help streamline this process?

1. use environment-specific workflows for automating tasks based on different deployment environments
2. develop some custom JavaScript actions for executing complex logic within workflows
3. create some Docker container actions for isolating workflow steps within separate containers
4. **create composite actions to combine multiple workflow steps into a single reusable action**

#### What is a requirement for publishing an action to GitHub Marketplace?

1. each repository can contain multiple actions
2. the action must be in a private repository
3. the actions metadata file must be in a subdirectory
4. **the action’s metadata file must be in the root directory of the repository**

#### What condition prevents a user from approving a deployment from a workflow run they initiated?

1. the job is not properly configured
2. the user’s account is not a GitHub Pro or Enterprise plan
3. the user does not have sufficient permissions
4. **the targeted environment variable has a self-approval prevention configuration**

#### When establishing corporate standards for managing GitHub Actions workflows within a large organization, which of the following elements are essential to document for optimal clarity and organization? (select three)

1. **plans for ongoing maintenance and version control of workflows**
2. **naming conventions for files and folders within workflows**
3. instructions for setting up and integrating individual workflows
4. **repositories used for storing different workflow components**

#### You've developed a custom GitHub Action to automate a specific task in your organization's CI/CD pipeline. What is the recommended first step in creating comprehensive documentation for the action?

1. include a comprehensive troubleshooting guide to address potential issues users may encounter
2. create a detailed list of all the features and capabilities of the action
3. **develop a thorough README.md file containing a detailed description, input and output arguments, secrets, environment variables, and usage examples for the action**
4. design an interactive tutorial to guide users through the setup and usage of the action

#### Your organization requires a runner to execute multiple GitHub Actions workflows that include CPU-intensive tasks and high-memory processes that access sensitive internal resources. Which runner type best aligns with these requirements?

1. self-hosted runner on a virtual machine within your cloud infrastructure
2. **self-hosted runner with dedicated hardware**
3. standard GitHub-hosted runner
4. GitHub-hosted large runner

#### What is the purpose of the jobs.<job\_id>.runs-on configuration in a GitHub Actions workflow?

1. **specifies the operating system and virtual environment for job execution**
2. sets up conditional statements for job execution
3. declares the GitHub Actions runner version to be used
4. defines environment variables for a specific job

#### What is the standard syntax for referencing secrets so you can use the GITHUB\_TOKEN?

1. ${{GITHUB\_TOKEN.secrets }}
2. ${{ GITHUB\_TOKEN }}
3. ${{ mySecrets.GITHUB\_TOKEN }}
4. **${{ secrets.GITHUB\_TOKEN }}**

#### When using job steps for actions, which YAML key is used to specify the GitHub Action to be executed within a job step?

1. action
2. perform
3. execute
4. **uses**

#### You are collaborating with a colleague using their repository in GitHub, but you are having trouble adding a secret for a workflow. What action is required to resolve the issue?

1. fork the repository to your personal account and add the secret to the forked repository
2. **request your colleague, who is the repository owner, to add the secret to the workflow**
3. contact GitHub support to grant you the required permissions to add secrets to the repository
4. submit a pull request to the repository, including the secret addition in the changes, for your colleague to review and merge

#### Which configuration is appropriate for triggering a workflow to run on webhook events related to check\_run actions?

on:

check\_run:

filter: [requested]

on:

check\_run:

types: [started]

on:

check\_run:

type: [closed]

**on:**

**check\_run:**

**types: [requested, completed]**

#### Which statement accurately describes the accessibility of default environment variables?

1. default environment variables are only accessible to steps explicitly requesting them
2. **default environment variables are set by GitHub are are available at every step in a workflow**
3. default environment variables are defined within the workflow and can be overwritten
4. default environment variables are accessible through the env context in the workflow file

#### In which scenario is using GitHub-hosted runners more suitable?

1. when you want to run workflows exclusively on your local machine
2. when you need complete control over the runner environment
3. **when you leverage the resources provided by GitHub for continuous integration**
4. when you require specific security configurations for you CI/CD processes

#### How can you execute a command stored in your GitHub repository using a workflow?

1. by embedding the script directly within the workflow configuration file
2. by using a third-party tool to trigger the execution of the script
3. by manually copying the script to the runner and executing it
4. **by running a script directly using the run keyword and specifying its location on the runner**

#### When a GitHub repository contains multiple custom actions, where is the recommended location to store the action files?

1. **.github/actions with separate subdirectories for each action**
2. actions with a single directory for all action files
3. workflows with separate workflow files defining each action
4. src with language-specific subdirectories for each action

#### Dan would like Drew's input on a particular line in the logs of a recently run workflow.  What is an efficient way to provide Drew access to the line in the logs?

1. copy the relevant line from the logs and paste it into a new issue on the repository's issue tracker.
2. download the log archive and email to Drew
3. **click on the step's line number to get a link to the specific line and share the link with Drew**
4. take a screenshot of the relevant portion of the logs and share it with Drew

#### What happens if a job marked as a dependency fails in a workflow with a dependent job?

1. only the dependent jobs is rerun
2. the dependent jobs is skipped, and subsequent jobs continue to run
3. all subsequent jobs continue to run
4. **the entire workflow is marked as failed**

#### How does a GitHub Actions workflow contribute to the deployment process to a cloud provider?

1. by directly deploying releases without any configuration
2. **by defining a series of actions in the workflow YAML file for deployment**
3. by triggering deployment based on the repository’s creation file
4. by automatically generating API keys for secure deployment

#### Which version of the actions/checkout action will be used for the following workflow configuration?

# This is a basic workflow to help you get started with Actions

 name: CI

# Controls when the workflow will run

on:

# Triggers the workflow on push or pull request events but only for the "main" branch

push:

branches: [ "main" ]

pull\_request:

branches: [ "main" ]

# Allows you to run this workflow manually from the Actions tab

workflow\_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel

jobs:

# This workflow contains a single job called "build"

build:

# The type of runner that the job will run on

runs-on: ubuntu-latest

# Steps represent a sequence of tasks that will be executed as part of the job

steps:

# Checks-out your repository under $GITHUB\_WORKSPACE, so your job can access it

- uses: actions/checkout

  # Runs a single command using the runners shell

- name: Run a one-line script

run: echo Hello, world!

1. default version of the action
2. **no version, - uses: actions/checkout is an invalid syntax**
3. version 3
4. latest version of the action

#### Which configuration provides a valid check to proceed with the steps only if the main branch triggered the workflow? (select two)

name: CI

on: push

jobs:

prod-check:

if: github.ref == main

runs-on: ubuntu-latest

steps:

- run: echo "the main branch triggered this workflow"

name: CI

on: push

jobs:

prod-check:

if: github.ref == 'main'

runs-on: ubuntu-latest

steps:

- run: echo "the main branch triggered this workflow"

**name: CI**

**on: push**

**jobs:**

**prod-check:**

**if: github.ref == 'refs/heads/main'**

**runs-on: ubuntu-latest**

**steps:**

**- run: echo "the main branch triggered this workflow"**

**name: CI**

**on: push**

**jobs:**

**prod-check:**

**if: ${{ github.ref == 'refs/heads/main' }}**

**runs-on: ubuntu-latest**

**steps:**

**- run: echo "the main branch triggered this workflow"**

#### What considerations should be made when developing JavaScript actions for GitHub Actions to ensure compatibility with all GitHub-hosted runners?

1. **write the JavaScript code to be pure JavaScript and not dependent on other binaries to ensure compatibility with all GitHub-hosted runners**
2. ensure that the JavaScript code relies on other binaries present in the runner image to speed up execution
3. package the JavaScript code using Node.js packages provided by the GitHub Actions Toolkit for faster development
4. develop the JavaScript code to run within a Docker container to ensure consistent execution across different operating systems

#### When creating a new GitHub action, what are the valid names of the metadata file that can be used? (select two)

1. job.yml
2. workflow.yaml
3. **action.yml**
4. **action.yaml**

#### You are trying to print a debug message using workflow commands but cannot locate it in the debug logs. What could be the issue?

1. **debug messages are not displayed in the logs by default**
2. the debug message was printed with incorrect syntax
3. a relevant event did not trigger the workflow run
4. the debug message was printed in the wrong step

#### What is the purpose of the restore-keys parameter in the GitHub Actions cache action?

1. enable cross-OS cache functionality
2. specify the location of the cached files
3. indicate whether a cache hit occurred
4. **provide alternative keys to use in case of a cache miss**

#### What is the primary purpose of an automated release management strategy?

1. streamline the release process by committing dependencies to every branch
2. **prioritize security by only committing dependencies to tagged release commits and performing builds during a release**
3. discourage users from referencing named tags or shas
4. encourage users to directly commit dependencies to the main branch

#### How can you view detailed logs to troubleshoot issues with a Docker container action in GitHub Actions?

1. use the git log command to view commit history related to the Docker container action
2. use the docker logs command to view detailed logs for the Docker container action
3. **check the GitHub Actions logs for detailed information about the execution of the Docker container action**
4. review the Docker container action's README file for troubleshooting tips

#### Which variable would you set to true in order to enable step debug logging?

1. ACTIONS\_WORKFLOW\_DEBUG
2. ACTIONS\_JOB\_DEBUG
3. ACTIONS\_RUNNER\_DEBUG
4. **ACTIONS\_STEP\_DEBUG**

#### When selecting the appropriate runners for support workloads in GitHub Actions, what factor is essential to consider specifically concerning supported operating systems?

1. cost of licensing for the operating system
2. **compatibility of the operating system with the tools and dependencies required for support tasks**
3. popularity of the operating system among team members
4. frequency of updates provided by the operating system vendor

#### How long does GitHub store logs and artifacts by default?

1. **90 days**
2. 1 year
3. indefinitely
4. 24 hours

#### What is the purpose of the workflow\_dispatch event?

1. **enables manual triggering of the workflow**
2. triggers the workflow when any branch is pushed
3. initiates the workflow for pull requests
4. runs the workflow on a predefined schedule

#### What are the benefits of using organization-templated workflows?  (select three)

1. **promotes consistency**
2. **promotes best practices**
3. **saves time**
4. leverages AI

#### You've developed a GitHub Action that automates the deployment process for your organization's web applications. You're thinking about distributing this action publicly or keeping it within your organization's private repositories. What factors might lead you to publish this action in a public repository?

1. you aim to increase the visibility of your organization's brand and expertise in automated deployment practices
2. you need to integrate the action seamlessly with other public repositories and workflows across various organizations
3. you want to restrict access to the action only to authorized users within your organization, maintaining control over its usage and distribution
4. **you want to foster collaboration and contribution from the broader GitHub community to enhance the features and functionality over time**

#### What additional parameter can be added to the workflow status badge URL to display the status of a specific branch?

1. **?branch=BRANCH-NAME**
2. ?commit=COMMIT-HASH
3. ?workflow=WORKFLOW-NAME
4. ?branch=WORKFLOW-NAME

#### Tom has built a workflow that sends requests to a service that is currently unavailable.  What is an appropriate action for Tom to take so that his workflow doesn't log errors until the service is restored?

1. delete the workflow file
2. modify the workflow file to skip the failing step
3. pause the workflow
4. **disable the workflow**

#### How many required inputs are declared in the metadata of this actions example?

inputs:

num-servers:

description: 'Number of Servers'

required: false

default: '3'

server-cpu-count:

description: 'CPU count of the Servers'

required: true

1. 0
2. 2
3. 3
4. **1**

#### You need to reference the value created by the step, as shown in the code below. What value should you use for the output?

runs:

using: "composite"

steps:

- id: random-number-generator

run: echo "random-id=$(echo $RANDOM)" >> $GITHUB\_OUTPUT

shell: bash

outputs:

random-number:

description: "Random number"

value: ${{ <what-should-go-here> }}

1. random-number-generator.outputs.random-id
2. **steps.random-number-generator.outputs.random-id**
3. outputs.random-id
4. random-id

#### Phil would like to filter all workflow runs triggered by a pull request.  Which filter can Phil use to achieve this in the GitHub Actions tab of his repository?

1. Branch
2. **Event**
3. Actor
4. Status

#### You want to publish your new action on the GitHub Marketplace to allow others to use it. What are some of the requirements that must be met before publishing? (select four)

1. **the name in the action's metadata file must be unique**
2. **you must accept the terms of service to publish actions in GitHub Marketplace**
3. **the action must be in a public repository**
4. the name cannot match an existing GitHub Marketplace category
5. **the action's metadata file must be in the root directory of the repository**
6. each repository must contain only a single action

#### What is the URL for the GitHub Container Registry?

1. **ghcr.io**
2. github.com/container
3. docker.io
4. github.com

#### Which configuration will trigger a workflow when an issue is opened in the repository?

on:

issues:

types: [opened]

**all of the above**

on:

issues

on:

issues:

types:

- opened

#### Which key in a workflow file is used to set a custom environment variable for a single workflow?

1. variables
2. steps
3. **env**
4. jobs

#### You need to monitor the status of the self-hosted runners that have been deployed in your organization. After logging into the GitHub UI, what valid status types can you expect to see? (select three)

1. **overloaded**
2. **idle**
3. **active**
4. offline

#### Your development team is responsible for deploying your organization's flagship application, which requires custom software tools for deployment. What type of runner would be ideal for this environment?

1. **self-hosted runners on virtual machines managed by the organization**
2. cloud-hosted runners from a third-party provider
3. GitHub-hosted runners provided by GitHub
4. self-hosted runners on physical servers maintained by the organization

#### What is the purpose of the timeout-minutes keyword in a step?

1. it specifies the maximum duration a job is allowed to run
2. it defines the time interval for individual commands within a step
3. it sets the timeout for waiting on external events before proceeding to the next step
4. **it limits the execution time for individual commands within a step**

#### What will occur if the .github/workflows directory contains an invalid workflow file?

1. GitHub Actions will place the invalid workflow files into a separate branch inside the repository
2. GitHub Actions will be disabled for the repository to prevent any failed workflow runs
3. GitHub Actions will automatically fix the syntax errors in the invalid workflow file
4. **GitHub Actions generates a failed workflow run for every new commit**

#### What will be the 4 colors echoed back for the jobs in the workflow configuration below?

name: my\_color\_workflow

on:

workflow\_dispatch:

env:

favorite\_color: orange

jobs:

my\_color\_1:

runs-on: ubuntu-latest

env:

favorite\_color: blue

steps:

- name: Set the color

id: step\_one

env:

favorite\_color: green

run: |

echo "my\_color=$favorite\_color" >> "$GITHUB\_ENV"

- name: echo the color

id: step\_two

run: |

echo "🎉 My color is ${{ env.my\_color }}"

echo "🎉 My favorite color is ${{ env.favorite\_color }}"

my\_color\_2:

runs-on: ubuntu-latest

steps:

- name: echo the color

id: step\_one

run: |

echo "🎉 My color is ${{ env.my\_color }}"

echo "🎉 My favorite color is ${{ env.favorite\_color }}"

🎉 My color is green

🎉 My favorite color is orange

🎉 My color is green

🎉 My favorite color is orange

🎉 My color is green

🎉 My favorite color is blue

🎉 My color is green

🎉 My favorite color is green

**🎉 My color is green**

**🎉 My favorite color is blue**

**🎉 My color is**

**🎉 My favorite color is orange**

🎉 My color is green

🎉 My favorite color is blue

🎉 My color is green

🎉 My favorite color is orange

#### You were provided a secret used by your production application, so you create a secret in the repository where the application code is managed. After creating the secret, users are complaining they can't access the secret when deploying from their own repository. What can you do to fix this?

1. add the other repositories to the policy to allow them to use the secret
2. **create the secret at the organization level and configure a policy that permits access to use the secret in the required repositories**
3. store the secret using Base64 encoding so the secret is unreadable
4. change the policy to permit anyone to access the secret

#### When are service containers created and destroyed for each configured service?

1. created at the start of the workflow and destroyed at the end
2. created at the start of the repository and destroyed when the repository is archived
3. created on-demand whenever a specific step requires the service container
4. **created for each job and destroyed when the job is completed**

#### What are actions in the context of GitHub development workflows?

1. predefined workflows provided by GitHub for common development tasks
2. automated scripts that execute code without user intervention
3. **individual tasks that can be customized to perform specific actions within development workflows**
4. integrated plugins for popular IDEs like Visual Studio Code

#### Your development team is troubleshooting connectivity issues with a self-hosted runner. What parameter can be used to validate that a self-hosted runner can access all required network services on GitHub?

1. --diag
2. **--check**
3. --verify-connection
4. --validate-network

#### What is the maximum number of jobs that a matrix can generate per workflow run?

1. 512 jobs
2. **256 jobs**
3. 128 jobs
4. unlimited amount of jobs

#### Sam would like to trigger a workflow when a push is made to any branch in the repository, or somebody creates a tag. How can Sam specify these events within the GitHub workflow configuration?

1. on: [push, tag]
2. **on: [push, create]**
3. on: {push, tag}
4. on: {push, create}

#### Why is it important to ensure that the repository only includes the metadata file, code, and files necessary for the action?

1. **to package the action in a single unit for tagging and releasing**
2. to comply with GitHub's review process
3. to increase the repository's visibility on GitHub
4. to reduce the size of the repository

#### What is a common purpose of using custom environment variables?

1. define global constants for the entire project
2. specify the file paths for input and output
3. control the flow of execution in the code
4. **store sensitive information like API keys**

#### GitHub Actions users in your organization are complaining that they can no longer use popular actions, such as checkout and labeler. What is the reason for this?

1. users are trying to use the actions with self-hosted runners
2. **the owner has restricted actions and reusable workflows to only those from your organization**
3. the organization has not been added to the public GitHub repository
4. the default GITHUB\_TOKEN does not have read access to reuse these actions

#### When using required: true for inputs in a GitHub Actions workflow file, what behavior should you expect when manually running the workflow with workflow\_dispatch?

1. inputs are automatically populated based on the previous run's inputs
2. the workflow will automatically run without requiring input specification
3. the workflow will return an error if inputs are not specified
4. **GitHub will automatically prompt the user to specify inputs before running the workflow**

#### Which workflow command is commonly used to set an output variable that can be consumed by subsequent steps in a workflow?

1. echo
2. export
3. set
4. **setOutput**

#### Tony would like to include a job that echoes the current repository name.  Which of the following code snippets provides an example that Tony could use?

**name: Code Repository**

**on:**

**push:**

**branches:**

**- main**

**jobs:**

**example-job:**

**runs-on: ubuntu-latest**

**steps:**

**- name: Print Repository Information**

**run: |**

**echo "Current Repository: $GITHUB\_REPOSITORY"**

name: Code Repository

on:

  push:

    branches:

      - main

jobs:

  example-job:

    runs-on: ubuntu-latest

    steps:

      - name: Print Repository Information

        run: |

          echo "Current Repository: $GITHUB\_REPOSITORY"

        env:

          GITHUB\_REPOSITORY: actions/code-examples

name: Code Repository

on:

  push:

    branches:

      - main

jobs:

  example-job:

    runs-on: ubuntu-latest

    steps:

      - name: Print Repository Information

        run: |

          echo "Current Repository: GITHUB\_REPOSITORY"

name: Code Repository

on:

  push:

    branches:

      - main

 jobs:

  example-job:

    runs-on: ubuntu-latest

    steps:

      - name: Print Repository Information

        run: |

          echo "Current Repository: $my\_repo"

        env:

          my\_repo: pwd

#### What is a drawback of using tags for versioning GitHub Actions, and what is the recommended alternative?

1. Tags can be easily manipulated or moved, leading to potential versioning inconsistencies. The recommended alternative is to use branch names for versioning.
2. Tags can only be applied to released versions, limiting flexibility in managing interim changes. The recommended alternative is to use commit SHAs for versioning
3. **Tags may not accurately represent the commit history and can be prone to deletion or movement. The recommended alternative is to use commit SHAs for versioning**
4. Tags are vulnerable to being truncated or abbreviated, leading to inaccuracies in version tracking. The recommended alternative is to use full commit SHAs for versioning.

#### What distinguishes YAML syntax from JSON when configuring workflow jobs in GitHub Actions?

1. **YAML allows significant newlines and indentation**
2. YAML syntax is more concise compared to JSON
3. YAML does not permit the use of comments within the configuration file
4. YAML allows the use of literal tab characters for indentation

### Set 5

#### How can you customize a workflow status badge to show the status of workflow runs triggered by the push event?

1. replace the event name with “push” in the workflow file
2. use the branch name as the event parameter
3. embed the badge in an HTML file with event-specific styling
4. **add ?event=push to the end of the status badge URL**

#### When you specify an input in a workflow file or use a default input value, what naming convention does GitHub use to create the corresponding environment variable?

1. it converts input names to lowercase letters and replaces spaces with \_ characters
2. **it converts input names to uppercase letters and replaces spaces with \_ characters**
3. it concatenates the input names with the keyword INPUT\_ and removes spaces
4. it keeps the input names unchanged but replaces spaces with - characters

#### What is the primary purpose of organization-level secrets?

1. organization-level secrets are only used for personal repositories
2. organization-level secrets are only accessible to repository owners
3. **organization-level secrets enable sharing secrets between multiple repositories, reducing the need for duplicate secrets**
4. organization-level secrets are primarily for storing non-sensitive information

#### You are reviewing a Docker container action and see this code as part of the entrypoint.sh file. What does this code indicate?

if <condition> ; then

echo "Writing Message"

exit 1

fi

1. the workflow has failed to write logs, but is exiting with a successful exit code
2. the workflow was successful, and the code is exiting as successful
3. the workflow was successful, but an optional step failed
4. **a task has failed, and the code is setting a failure exit code**

#### What are the downsides to consider when choosing to use self-hosted runners in GitHub Actions? (select three)

1. limited scalability compared to GitHub-hosted runners
2. **managing the potential security risks associated with self-hosted infrastructure**
3. **the requirement of stable network connectivity for runner operation**
4. inability to access GitHub's integrated features and updates
5. **maintenance overhead for managing and updating the runner environment**

#### What is the purpose of the GITHUB\_ACTIONS variable?

1. **it always evaluates to true when GitHub Actions is running in the workflow**
2. it indicates the status of the GitHub Actions workflow file
3. it is a placeholder for the GitHub Actions workflow file
4. it represents the API URL for GitHub

#### Evaluating the configuration below, which container registry does this workflow publish to?

name: Create and publish a Docker image

on:

push:

branches: ['release']

env:

REGISTRY: ghcr.io

IMAGE\_NAME: ${{ github.repository }}

jobs:

build-and-push-image:

runs-on: ubuntu-latest

permissions:

contents: read

packages: write

steps:

- name: Checkout repository

uses: actions/checkout@v4

- name: Log in to the Container registry

uses: Build and push Docker image@v3

with:

registry: ${{ env.REGISTRY }}

username: ${{ github.actor }}

password: ${{ secrets.GITHUB\_TOKEN }}

- name: Extract metadata (tags, labels) for Docker

id: meta

uses: docker/metadata-action@v5

with:

images: ${{ env.REGISTRY }}/${{ env.IMAGE\_NAME }}

- name: Build and push Docker image

uses: docker/build-push-action@v5

with:

context: .

push: true

tags: ${{ steps.meta.outputs.tags }}

labels: ${{ steps.meta.outputs.labels }}

1. **GitHub Container Registry**
2. Amazon ECR
3. Artifactory
4. Docker Hub

#### Dani wants to be notified when a comment is created on an issue within a GitHub repository. Which event should be used within the configuration?

1. **issue\_comment**
2. issues
3. comment
4. issues.comment

#### What advantage do JavaScript actions offer over Docker container actions in GitHub Actions?

1. Docker container actions ensure consistent execution across different operating systems, enhancing workflow reliability
2. **JavaScript actions run directly on runner machines, simplifying action code and improving execution speed**
3. Docker container actions provide better isolation and security compared to JavaScript actions
4. JavaScript actions allow for easier integration with third-party APIs and services

#### What step should you take to verify the environment variables passed to your Docker container action?

1. **use the env command within the Docker container action to display the environment variables**
2. inspect the environment variables within the Docker container action using the docker inspect command
3. review the environment variables section of the Docker container action's YAML file
4. check the GitHub Actions logs for a detailed breakdown of environment variable values

#### GitHub Actions will allow for deleting a workflow run under the following conditions (select two)

1. a workflow run that is more then one week old
2. **a workflow run that is more than two weeks old**
3. a workflow run that is queued
4. **a workflow run that has been completed**

#### The IT governance team needs to create policies for using GitHub Actions to ensure compliance and security. What is the most effective approach to achieving this objective?

1. conduct regular training sessions to educate developers on best practices for creating secure and compliant GitHub Actions workflows
2. enforce mandatory code reviews for all GitHub Actions workflow changes to ensure adherence to organizational policies
3. use GitHub Actions environment variables to enforce custom policies and restrictions across all repositories
4. **implement organization-wide templates defining approved GitHub Actions workflows with pre-configured security measures**

#### Which statement best explains why a workflow triggered by the check\_suite event is not executing when run from a feature branch?

1. GitHub Actions does not support workflow runs from feature branches triggered by the check\_suite event
2. **the check\_suite event is designed to exclusively trigger workflow runs on the default branch**
3. workflows triggered by the check\_suite event are restricted to running only on the default branch
4. the check\_suite event is independent of branch location and should trigger workflow runs from any branch

#### Which configuration is appropriate for triggering a workflow when a release is published?

on: published

on:

release:

event: [published]

**on:**

**release:**

**types: [published]**

on:

release:

when: [published]

#### While executing a complex GitHub Actions workflow, you realize that the workflow logs are becoming cluttered, and you need to improve readability and organization. Which workflow command should you use to accomplish this task?

1. **group**
2. add-mask
3. echo
4. stop-commands

#### As an enterprise owner, you want to restrict the use of GitHub Actions within your organization but still allow access to essential workflows. Which of the following configurations would achieve this goal?

1. set a policy to require two-factor authentication for all users running GitHub Actions workflows
2. allow all Marketplace actions but restrict workflows to specific organizations and repositories
3. disable GitHub Actions for all organizations in the enterprise
4. **enforce a policy to allow only local actions and reusable workflows**

#### Which action can be used to download artifacts from a GitHub Actions workflow?

1. the actions/download action
2. **the actions/download-artifact action**
3. the actions/upload action
4. the actions/upload-artifacts action

#### Which of the following statements are true regarding GitHub default environment variables? (select three)

1. **default environment variables are not accessible through the env context**
2. **default environment variables are available to every step in a workflow**
3. default environment variables are usually denoted using the dollar sign and curly braces, as ${{ variable.property }}
4. **default environment variables are all uppercase**

#### When troubleshooting issues related to custom environment variables in a workflow, what is a recommended step?

1. rely solely on external forums for solutions and ignore internal resources
2. ignore the environment variables and focus on other code aspects
3. hardcode the variables temporarily for testing purposes
4. **review workflow logs and documentation for configuration details**

#### Daniel is managing secrets for GitHub Actions workflows and is considering whether to store a sensitive API key at the organization or environment level. Which factors should be considered when making this decision? (select three)

1. the need for individual approval before workflows can access the key
2. **the number of repositories requiring access to the API key**
3. **the frequency with which the API key needs to be updated**
4. **the level of access control required for different teams using the key**

#### You are the maintainer of a GitHub Action used by various teams in your organization. Recently, you made significant changes to the action that will break compatibility with existing workflows. How should you manage the release of this new version using tags?

1. release the new version v2.0.0 without indicating its status, as it is intended for internal use only
2. create and validate the release directly on the main branch before creating the release tag
3. create a new release branch named release/v2 and directly create the release tag v2.0.0 on this branch without validation
4. follow semantic versioning and create a new major version tag v2.0.0 along with a beta tag v2-beta to indicate its experimental status
5. **introduce a new major version tag v2.0.0 on the main branch and update the action's metadata accordingly**

#### What metadata keywords within an actions.yml file is used to indicate the type of action being executed?

runs:

type:

**runs:**

**using:**

inputs:

name:

#### Which of the following are deployment protections that can be configured on an environment? (select three)

1. **require reviewers before deployment**
2. **prevent self-reviews for deployment**
3. enforce a minimum code coverage
4. **restrict which branches can deploy to the environment**

#### Steve wants to create a configuration variable for use across multiple workflows.  He has learned that he can define this variable at the organization, repository, or environment level.  Which value takes precedence if Steve configures a variable with the same name at each level?

1. there is no precedence; the values are merged
2. **environment-level**
3. organization
4. repository

#### Why might it be beneficial to configure workflows to authenticate directly to a cloud provider that supports OpenID Connect?

1. streamline the workflow execution time
2. **enhance security by avoiding the storage of long-lived credentials as secrets**
3. enable automatic deployment without any authentication
4. avoid the need for a GitHub repository

#### How can you share actions and reusable workflows from your private repository without making them public?

1. copy the action code into your project's repository and include it in your workflow files.
2. **grant access to specific users or organizations for your private repository**
3. use a separate public repository to host the actions and workflows and grant access to specific users or organizations
4. convert the actions into Docker containers and store them in a private registry

#### What is a recommended practice for users when referencing a GitHub Action in their workflows?

1. **specify a major version when using the action, and direct to a more specific version only if issues arise**
2. always reference the action's default branch to ensure access to the latest code
3. configure the workflow to automatically update to the latest version of the action
4. directly target the latest commit's SHA to ensure compatibility with the action

#### Which of the following should be included in a comprehensive README.md file for a custom GitHub Action? (select five)

1. **environment variables the action uses**
2. troubleshooting guide for common issues
3. **required input and output arguments**
4. **secrets the action uses**
5. an interactive tutorial for users
6. **detailed description of what the action does**
7. **optional input and output arguments**

#### Dave is creating a templated workflow for his organization.  Where must Dave store the workflow files and associated metadata files for the templated workflow?

1. inside a directory named .github/workflow-templates
2. inside a directory named workflow-templates within the current repository
3. inside a directory named .github/org-templates
4. **inside a directory named workflow-templates within a repository named .github**

#### You've recently encountered an issue with a custom JavaScript action in your GitHub Actions workflow, where the action fails to execute as expected. What is the recommended approach for troubleshooting this issue?

1. review the action's code for syntax errors and debugging statements
2. run the action locally using GitHub's Actions CLI tool for debugging purposes
3. revert to using Docker container actions for improved stability and reliability
4. **examine the workflow logs to see which step caused the failure and review the failed step's build logs to troubleshoot**

#### What are the benefits of reusing workflows in an organization? (select four)

1. eliminates security threats to the application
2. **promotes best practice throughout the organization**
3. **avoids duplication**
4. **makes workflows easier to maintain**
5. enables you to create a workflow that includes jobs for test, staging, and qa
6. **can build on the work of others**

#### You're assisting a colleague who wants to understand the differences between GitHub-hosted runners and self-hosted runners. They must choose the best option for running their team's GitHub Actions workflows. What key points would you include in your explanation to differentiate these two runner types effectively? (select three)

1. **self-hosted runners enable access to resources within your private network, unlike GitHub-hosted runners**
2. **GitHub-hosted runners are generally more cost-effective for low-volume workflows**
3. both types of runners provide the same level of control over workflow execution
4. self-hosted runners require less technical expertise to set up and manage
5. **GitHub-hosted runners offer a wider variety of operating systems and hardware configurations**

#### What does this  badge indicate about a GitHub Action within the Marketplace?

1. **verified creator badge**
2. security check badge
3. approved usage badge
4. status badge

#### You have been tasked with creating a GitHub Action to automate the deployment process of your organization's web application. What would be the appropriate directory to store the workflow files?

1. **the appropriate directory is .github/workflows for workflow files**
2. the appropriate directory is src/main/java for workflow files
3. the appropriate directory is actions/deploy for workflow files
4. the appropriate directory is scripts/development for workflow files

#### What level of access is required on a GitHub repository in order to delete log files from workflow runs?

1. admin
2. owner
3. **write**
4. read

#### Which of the following is a benefit of manually triggering workflows?

1. restricts workflow runs to specific branches only
2. ensures that workflows run automatically without any user intervention
3. **allows for the testing of workflows in a controlled environment**
4. speeds up the overall execution time of a workflow

#### Jeff is troubleshooting a failed GitHub actions workflow run and is searching the build logs for a particular step. His search results are empty. What are the possible reasons for empty search results? (select two)

1. Jeff has exceeded the search query limit for the workflow run
2. **Jeff's searched text is misspelled or not present in the build logs**
3. **Jeff has not expanded each of the steps for the job and, therefore, cannot see the results**
4. The build logs have been archived and are not accessible for searching

#### What network requirement is necessary for self-hosted runners in GitHub Actions for connectivity to GitHub?

1. access to all of GitHub's public APIs and services
2. high-speed internet connection to minimize latency
3. **permitting outbound connectivity to GitHub hosts using long polling**
4. the use of a proxy for outbound connectivity to GitHub

#### How can you access the current values of variables in a matrix within a job in the example below:

jobs:

example\_matrix:

strategy:

matrix:

version: [10, 12, 14]

os: [ubuntu-latest, windows-latest]

1. by using the context keyword within the job configuration
2. by accessing the variables directly with the syntax version and os
3. by using the matrix.property syntax
4. **reference variables through the matrix context with syntax like matrix.version and matrix.os**

#### You want to configure a GitHub Actions workflow to run only for specific activity types triggered by a webhook event payload. Which keyword should you use in your workflow configuration, and how can you restrict it to specific activity types?

1. use the on keyword with a conditional statement based on the event payload
2. use the events keyword with a regex pattern to match specific event names
3. **use the types keyword with a list of specific activity types**
4. use the workflow keyword with a list of event names to include

#### How can you configure a workflow to run on a scheduled basis?

1. **use the on: schedule event**
2. add a custom webhook for scheduled events
3. trigger the workflow manually at scheduled intervals
4. specify the schedule in the jobs section

#### What is the primary purpose of the marketplace in the context of actions, workflows, jobs, steps, and runs?

1. **providing a platform for publishing and sharing actions**
2. managing the overall workflow structure
3. executing individual steps
4. monitoring the progress of individual jobs

#### What action must be taken if the “Publish” checkbox is disabled when attempting to release an action to GitHub Marketplace?

1. **accept the GitHub Marketplace Developer Agreement**
2. modify the repository’s workflow files
3. update the action’s metadata file
4. create a separate repository for the action

#### What can be viewed directly on the Actions tab in GitHub?  (select three)

1. **the branch for each workflow run**
2. the configuration for each workflow run
3. **the status of each workflow run**
4. **the length of time for each workflow run**

#### What is the primary purpose of creating a comprehensive README.md file for a custom GitHub Action?

1. showcase the action's features and capabilities with screenshots and videos
2. **facilitate the action's adoption by other users by offering detailed documentation**
3. troubleshoot issues that users may encounter while using the action
4. provide an interactive tutorial for users to learn how to use the action

#### What scope levels can you create an encrypted secret in a GitHub organization? (select three)

1. **organization**
2. **environment**
3. **repository**
4. workflow

#### What is a critical consideration when deleting an artifact?

1. deleting artifacts does not impact GitHub Actions storage
2. write access to the repository is not required
3. **once deleted, an artifact cannot be restored**
4. deleted artifacts can be restored upon request

#### What is true about the following workflow configuration if triggered against the myorg/my-dev-repo repository?

name: example-workflow

on: [push]

jobs:

production-deploy:

if: github.repository == 'myorg/my-prod-repo'

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with:

node-version: '14'

- run: npm install -g bats

1. the production-deploy job will run if the node-version is 14
2. **the production-deploy job will be marked as skipped**
3. the production-deploy job will error
4. the production-deploy job will execute three steps

#### Which statement describes how GitHub utilizes an action's metadata on the GitHub Marketplace page?

1. the action's metadata is displayed as a separate webpage for each action
2. the metadata is used to generate documentation for the action's usage
3. **the actions' metadata is used to provide key information and details about the action**
4. the metadata is employed to categorize actions based on their functionality

#### Which are valid names for a workflow within GitHub actions?  (select two)

1. name: ${{ learn-github-actions }}
2. name: ${{ "learn-github-actions" }}
3. **name: learn-github-actions**
4. **name: ${{ 'learn-github-actions' }}**

#### What are the steps needed to publish your action to the GitHub Marketplace?

1. merge the action's metadata file into the repository's main branch, draft a release, select "Publish this Action to the GitHub Marketplace," and choose categories
2. tag a new release, push it to the repository, draft a release, select "Publish this Action to the GitHub Marketplace," and set a version tag
3. submit the action's metadata file for review by GitHub, select categories, set a version tag, draft a release, and publish it
4. **add the action's metadata file to the repository root directory, draft a release, select "Publish this Action to the GitHub Marketplace," choose categories, set a version tag, and publish the release**

#### How can you make an environment variable available to subsequent steps in a workflow job using Bash?

1. set the variable in a separate configuration file
2. use the export command
3. manually pass the variable as an argument to each step
4. **define or update the variable use the echo command with >>$GITHUB\_ENV**

#### What advantage do composite run steps actions offer when compared to other methods of task automation in GitHub Actions?

1. they automatically generate documentation based on the shell scripts included
2. they enable the seamless integration of third-party APIs without writing additional code
3. they provide a graphical user interface for easier configuration and management
4. **they allow for the direct execution of shell scripts without the need for additional setup**

#### Which of the following is an advantage of using shell commands in job steps within a GitHub Actions workflow?

1. **shell commands provide flexibility in executing custom scripts and commands**
2. shell commands can only be used for debugging purposes
3. none of the above - shell commands cannot be used in workflows
4. shell commands allow for manual triggering of the workflows

#### What is a key benefit of using service containers in a workflow for testing databases and services?

1. **easy access to actual or simulated external dependencies**
2. improved code compilation speed
3. automatic detection of code vulnerabilities
4. seamless integration with GitHub Pages

#### How can you use workflow commands to customize the runner environment for a specific step?

1. **using the :: syntax to send commands to the runner during the run step**
2. using the environment keyword in the workflow YAML
3. adding a dedicated workflow step with the desired configurations
4. updating the GitHub Actions configuration file

#### What is the purpose of setting a default working default directory for run commands in a workflow?

1. **specify the location of the script files within the repository**
2. define where the script output will be saved after execution
3. enhance the security of the workflow by isolating script execution environments
4. restrict the access of the script to specific directories on the runner

#### The ACTIONS\_STEP\_DEBUG can be set to true to enable step debug logging.  How can this setting be configured?

1. **as a secret or variable with the value of the secret taking precedence**
2. as a variable
3. as a secret or variable with the value of the variable taking precedence
4. as a secret

#### You have multiple types of self-hosted runners in your organization provisioned for different workloads. Which GitHub Actions feature allows you to categorize these different runners and easily incorporate them into workflows?

1. runner environments
2. runner groups
3. runner sets
4. **runner labels**

#### What is a key advantage of using self-hosted runners in a CI/CD pipeline?

1. **minimal setup and maintenance effort**
2. integration with GitHub Actions marketplace
3. enhanced security provided by GitHub’s infrastructure
4. access to unlimited computing resources