#_ Important Collection of GitHub Actions

1. 🛠️ **Workflow Syntax:**

   - **Component:** Workflow File
   - **Importance:** Defines the structure, triggers, and steps of a GitHub Actions workflow.
   - **Resources:**
     - GitHub Actions Syntax

2. 🧰 **Actions:**

   - **Component:** Actions
   - **Importance:** Reusable units of work defined as Docker containers or JavaScript code, used in workflow steps.
   - **Resources:**
     - GitHub Marketplace - Actions

3. 🏗️ **Workflow Triggers:**

   - **Component:** Event Triggers
   - **Importance:** Define events that trigger the execution of workflows, such as push, pull request, schedule, etc.
   - **Resources:**
     - GitHub Actions Events

4. 🔑 **Secrets Management:**

   - **Component:** Secrets
   - **Importance:** Store and manage sensitive information (API tokens, credentials) securely, accessible during workflow execution.
   - **Resources:**
     - GitHub Actions Secrets

By: Waleed Mousa

5. 🌐 **Environment Setup:**

- **Component:** Environment
- **Importance:** Define runtime environments (e.g., OS, Node.js version) for workflow steps.
- **Resources:**
  - GitHub Actions Environments

6. 🔄 **Continuous Integration Workflows:**

- **Workflow:** Continuous Integration
- **Importance:** Ensures that code changes are continuously integrated, built, and tested.
- **Uses:** Prevents integration conflicts and ensures a consistent codebase.
- **Example:**
  - Continuous Integration Workflow Example

7. 🚀 **Deployment:**

- **Component:** Deployment Steps
- **Importance:** Automate the deployment process to servers, platforms, and cloud services.
- **Resources:**
  - Deploying to GitHub Pages
  - Deploying to Azure

8. 📊 **Reporting and Notifications:**

- **Component:** Reporting and Notifications Steps
- **Importance:** Generate reports, send notifications, and create artifacts for further analysis.
- **Resources:**
  - GitHub Actions Artifacts

9. 🧹 Cleanup:

- **Component:** Cleanup Steps
- **Importance:** Perform cleanup tasks, such as removing temporary files or resources.
- **Resources:**
  - GitHub Actions Workflow Syntax

10. ⏰ Scheduled Workflows:

- **Component:** Scheduled Events
- **Importance:** Trigger workflows at specified times or intervals using cron syntax.
- **Resources:**
  - Scheduled Events Syntax

11. 👥 Collaboration:

- **Component:** Workflow Dispatch
- **Importance:** Manually trigger workflows using the GitHub API, facilitating collaboration and automation.
- **Resources:**
  - Workflow Dispatch Event

12. 🤖 Self-hosted Runners:

- **Component:** Self-hosted Runners
- **Importance:** Set up and use self-hosted runners to execute workflows on your infrastructure.
- **Resources:**
  - Self-hosted Runner Documentation

13. 🎯 Matrix Builds:

- **Component:** Matrix Strategies
- **Importance:** Define a matrix of configurations for testing on various environments.
- **Resources:**
  - Workflow Matrix Strategies

By: Waleed Mousa

14. 🔐 Workflow Permissions:

- **Component:** Permissions and Access Control
- **Importance:** Configure which branches and events can trigger workflows and restrict access to workflow outputs.
- **Resources:**
  - GitHub Actions Permissions

15. 📜 Conditional Execution:

- **Component:** Conditional Steps
- **Importance:** Conditionally execute steps based on certain criteria, such as the outcome of previous steps.
- **Resources:**
  - GitHub Actions Conditional Steps

16. 🚦 Workflow Status Checks:

- **Component:** Workflow Status Checks
- **Importance:** Define checks on pull requests before merging to ensure the workflow passes.
- **Resources:**
  - GitHub Actions Status Checks

17. 💬 Workflow Comments:

- **Component:** Workflow Comments
- **Importance:** Add comments to pull requests or issues based on workflow results.
- **Resources:**
  - GitHub Actions Workflow Comments

18. 📂 Code Analysis Workflows:

- **Workflow:** Code Analysis
- **Importance:** Runs static code analysis tools to identify code quality issues, security vulnerabilities, and more.
- **Uses:** Ensures adherence to coding standards and identifies potential issues early.

By: Waleed Mousa

- **Example:**
  - Code Analysis Workflow Example

19. 🔒 Security Scanning Workflows:

- **Workflow:** Security Scanning
- **Importance:** Runs security scanning tools to identify vulnerabilities and security risks.
- **Uses:** Helps prevent security breaches and data leaks.
- **Example:**
  - Security Scanning Workflow Example

20. 🚀 Continuous Delivery Workflows:

- **Workflow:** Continuous Delivery
- **Importance:** Automates the process of delivering code to production environments.
- **Uses:** Ensures fast and reliable software delivery.
- **Example:**
  - Continuous Delivery Workflow Example

21. 🌐 Containerization Workflows:

- **Workflow:** Containerization
- **Importance:** Builds and pushes container images to a container registry for deployment.
- **Uses:** Enables scalable and consistent deployment of applications.
- **Example:**
  - Containerization Workflow Example

22. 📋 Documentation Workflows:

- **Workflow:** Documentation
- **Importance:** Automates the process of generating and publishing documentation.
- **Uses:** Keeps documentation up-to-date and accessible.
- **Example:**
  - Documentation Workflow Example

23. 🚨 Incident Response Workflows:

- **Workflow:** Incident Response
- **Importance:** Coordinates actions during incidents, such as triggering alerts, communicating with teams, and taking remedial actions.
- **Uses:** Helps manage incidents and minimize downtime.
- **Example:**
    - Incident Response Workflow Example