**GitHub Actions 3-Day Course Content**

**Course Overview**

This course is designed to provide participants with an understanding of GitHub Actions, enabling them to automate workflows within their GitHub repositories. By the end of this course, participants will be able to create, manage, and optimize CI/CD pipelines using GitHub Actions.

**Prerequisites**

* Familiarity with Git and GitHub.
* Basic knowledge of programming (preferably in JavaScript, Python, or any language commonly used in CI/CD pipelines).
* Understanding of continuous integration and continuous deployment (CI/CD) concepts.
* Access to a GitHub account.

**Day 1: Introduction to GitHub Actions**

**Objectives:**

* Understand the fundamentals of GitHub Actions.
* Explore the GitHub Actions interface and YAML syntax.
* Learn how to create your first simple workflow.

**Content:**

1. **Introduction to GitHub Actions**
   * What is GitHub Actions?
   * Benefits of using GitHub Actions.
2. **Getting Started with GitHub Actions**
   * Navigating the GitHub repository interface.
   * Workflow files and YAML syntax.
   * Creating your first workflow.
3. **Triggers and Events**
   * Understanding triggers (push, pull request, schedule).
   * Exploring GitHub events.
4. **Jobs and Steps**
   * Defining jobs and steps within a workflow.
   * Using actions from the GitHub Marketplace.

**Lab Requirements:**

* A GitHub account.
* A simple project repository for hands-on practice.

**Lab Activity:**

* Create a workflow that runs on a push event, echoing a message to confirm it’s working.

**Day 2: Advanced GitHub Actions**

**Objectives:**

* Learn to use GitHub Actions for testing and deployment.
* Explore action sharing and community contributions.
* Implement secrets management in workflows.

**Content:**

1. **Testing with GitHub Actions**
   * Setting up test workflows.
   * Using different environments for tests.
2. **Deployment with GitHub Actions**
   * Deploying to popular platforms (e.g., AWS, Heroku).
   * Create a continuous deployment pipeline.
3. **Using Community Actions**
   * Finding and using actions from the GitHub Marketplace.
   * Writing reusable actions.
4. **Secrets Management**
   * Managing sensitive information with GitHub Secrets.
   * Best practices for handling secrets.

**Lab Requirements:**

* A sample application that can be tested and deployed.
* Access to a deployment platform (e.g., AWS, Heroku) for lab demonstrations.

**Lab Activity:**

* Set up a testing and deployment action for a sample application to a cloud service.

**Day 3: Optimization and Best Practices**

**Objectives:**

* Learn how to optimize workflows for speed and reliability.
* Discover debugging tips and strategies.
* Explore advanced features and case studies.

**Content:**

1. **Optimizing Workflows**
   * Caching dependencies to speed up builds.
   * Parallelizing jobs for efficiency.
2. **Debugging Workflows**
   * Common troubleshooting techniques.
   * GitHub Action logs and debugging strategies.
3. **Advanced GitHub Action Features**
   * Matrix builds for different configurations and environments.
   * Scheduled workflows for maintenance tasks.
4. **Case Study: Implementing a CI/CD Pipeline**
   * Real-world case studies of GitHub Actions in production.
   * Discussion on challenges and solutions.

**Lab Requirements:**

* A more complex project or multiple repositories for workflow implementation.
* Access to necessary tools for debugging actions.

**Lab Activity:**

* Create a complete CI/CD pipeline for a sample application, incorporating tests, builds, and deployments, while applying optimization techniques learned.

**Conclusion**

At the end of the course, participants will have a comprehensive understanding of GitHub Actions, enabling them to set up and manage workflows effectively in their own projects. They will also have practical experience with both basic and advanced features of GitHub Actions.