

GitHub and GitHub Actions

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: WA3621

Duration: 2 days

Overview

Course Description

This GitHub and GitHub Actions course teaches developers fundamental concepts, including the distinction between Git and GitHub, effective repository management, and collaborative workflows. Furthermore, the course delves into using GitHub Actions to streamline Continuous Integration (CI) and Continuous Deployment (CD) processes.

Skills Gained

- Differentiate between Git and GitHub and explain their roles in version control and collaboration.
- Navigate the GitHub interface.
- Create and clone repositories, and synchronize local repositories with GitHub using Push, Fetch, and Pull.
- Manage branches, merge changes, and resolve conflicts.
- Understand GitHub Codespaces for enhanced development workflows.
- Create and manage workflows to automate CI/CD processes.

Prerequisites

All attendees must have taken Ascendient Learning's course [Git for Developers](#) or have the equivalent knowledge.

Training Materials

All Git training attendees receive comprehensive course materials.

Software Requirements

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Ascendient Learning will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

Audience

Course Details

Git vs. GitHub

- Key differences between Git and GitHub

GitHub interface overview

- Navigation: Repositories, Pull Requests, Issues, and Actions
- Settings: Managing repository permissions and configurations

Creating and cloning repositories

- Steps to create a new repository on GitHub (public vs. private)
- Cloning a repository locally using HTTPS, SSH, or GitHub CLI

Synchronize GitHub and local repositories (Push, Fetch, Pull)

Branching and merging

- Creating, switching, and deleting branches using Git commands
- Performing merges

Resolving conflicts

- Understanding why merge conflicts occur and common scenarios
- Conflict resolution using tools like VS Code or the Git CLI

Forking

- Keeping a forked repository in sync with the original repository

Pull requests

- Creating pull requests for code review and contribution
- Managing pull request workflows: Assigning reviewers and merging changes

Managing issues and discussions

- Creating and organizing GitHub Issues with labels and milestones

GitHub Codespaces overview

GitHub Actions overview

- Overview of GitHub Actions workflow syntax and events
- Key use cases: Automating tests, builds, and deployments

Creating and managing workflows

- Writing workflow YAML files for specific triggers (push, pull requests, etc.)
- Managing workflow runs and viewing logs for debugging

Managing GitHub secrets

- Adding and using secrets in workflows for secure credentials
- Rotating and removing secrets when no longer needed

Continuous Integration (CI)

- Setting up CI pipelines to automate testing

Continuous Deployment (CD)

- Automating deployment pipelines to staging or production environments
- Using GitHub Actions to manage multi-environment deployments

GitHub Packages registry

- Publishing packages to GitHub Packages (npm, Docker, etc.)
- Accessing and consuming packages in workflows or applications

Caching and artifact management

- Configuring caching in workflows to speed up builds and tests

Integration security and code quality tools

Monitoring and troubleshooting workflows