

**CI/CD (using GitHub) and Microservices Development in Python**

**Course Number:** DVOP-170WA
**Duration:** 5 days

**Overview**

This CI/CD (using GitHub) and Microservices Development in Python training course teaches attendees how to use Python and Git to build and deploy applications. Students learn how to use scripts to build applications, manage source code, implement DevOps practices with Git, build microservices with Python and Flask, implement automated testing with Python unit-testing frameworks, and deploy applications using different strategies.

**Prerequisites**

Attendees must have a basic understanding of software development and some experience with programming.

**Materials**

All DevOps CI/CD training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* A modern web browser and an Internet connection free of restrictive firewalls, so that the student can connect by SSH and Remote Desktop (RDP) into remote virtual machines.
* Any Windows, Linux, or macOS operating system
* Python 3.x installed (Anaconda bundle recommended)
* An IDE with Python support (PyCharm Community Edition is an excellent free option, but there are several other good ones)

**Objectives**

* Understand the basics of Python and Git
* Use Python to build applications
* Use Git to manage source code
* Implement DevOps practices with CI/CD using Python and GitHub Actions
* Understand the fundamentals of building microservices with Python and Flask
* Implement automated testing with Python
* Learn how to use Python vulnerability scanning & code coverage tools
* Deploy applications using different strategies

**Outline**

* Introduction
* DevOps Fundamentals
	+ What is DevOps?
	+ What is culture?
	+ DevOps, People, and the Agile Manifesto
	+ Agile Design Principles
	+ DevOps Tools
	+ Tools that Drive DevOps Success
	+ Cloud Computing and AWS
	+ Cloud, IaaS, and DevOps
	+ DevOps Processes and Methods
	+ Highly Effective DevOps Processes
	+ The Phoenix Project: The 3 Ways
	+ The Unicorn Project – The 5 Ideals
	+ Other DevOps Definitions
	+ What Isn’t DevOps?
	+ DevOps Anti-patterns to Avoid
* Introduction to Git and Git Flow
	+ What is Git?
	+ Benefits of Git
	+ Key Git Commands for Local Development
	+ Git Configuration
	+ Hard Git Resets
	+ Distributed Development
	+ Branch Based Development
	+ Branch Creation and Merges
	+ What is Git Flow?
	+ Benefits of Git Flow
	+ Git Flow Branches
	+ Git Flow Workflow Diagram
	+ How does Git Flow work?
	+ Git Flow Hotfixes
	+ Reflection on GitFlow
	+ Git Flow Alternatives
	+ Git vs. GitHub
	+ GithHub Flow
	+ GitHub Flow Workflow
	+ GitHub – Benefits
	+ GitLab Flow
	+ GitLab – Benefits
	+ Trunk-based Development – When it Works?
* Introduction to Python and Build Tools
	+ What is Python?
	+ Benefits of Python?
	+ Features of Python
	+ Python Syntax Overview?
	+ Example Python Syntax
	+ Cross Platform Python
	+ Python Artifacts
	+ Python Build Processes
	+ Dependency Management with Pip
	+ Virtual Environments
	+ Pyenv for Python Version Management
	+ Using Pyenv
	+ Combining Pyenv with Virtual Environments
	+ Python IDEs
	+ Conda as an alternate to Pip
	+ Automation with Python
	+ Python and Cloud Services
	+ Python and Containerization
	+ Python in CI/CD Pipelines
	+ Python's Role in DevOps
* Introduction to Continuous Integration, Continuous Delivery and Continuous Deployment
	+ Distributed Version Control Systems and Merged Code
	+ The Need for Continuous Integration Tools
	+ Commit, Integrate, Compile, Test and Repeat
	+ When is Integration Successful?
	+ Benefits of Continuous Integration
	+ What should a CI tool do?
	+ How far should a CI tool go?
	+ Continuous Delivery
	+ A Definition of Continuous Delivery
	+ Martin Fowler on Continuous Delivery
	+ The Continuous Deployment Awakening
	+ What constitutes Deployment?
	+ DevOps and Continuous Deployment
	+ Continuous Deployment Challenges
	+ CI vs CD(elivery) vs CD(eployment)
* Understanding the GitHub Interface
	+ Creating a GitHub Account
	+ Key GitHub Interface Components
	+ The GitHub Interface
	+ Configuring your GitHub Profile
	+ Creating Repositories in GitHub
	+ GitHub Forking and Cloning
	+ Setting up Git locally
	+ Managing Access via SSH and HTTP/s
	+ Setting up SSH Keys (example)
	+ GitHub Notifications
	+ GitHub Markdown
	+ GitHub Community
* Understanding GitHub Actions and Workflows
	+ What are GitHub Actions?
	+ Key Benefits of GitHub Actions
	+ Defining Workflows
	+ Workflow File Syntax and Structure
	+ Workflow Event Types
	+ Workflow Events in Practice
	+ Utilizing GitHub Actions Marketplace
	+ Continuous Integration and Delivery (CI/CD)
* Understanding Pipelines in GitHub Actions
	+ GitHub Actions Workflows vs Pipelines
	+ Simple Workflow with Jobs
	+ Pipelines with Stages
	+ Leveraging GitHub Actions for Pipelines
	+ Benefits of Pipeline-based Workflows
	+ Developing Pipelines in Github Actions
	+ Managing and Modifying Pipelines
	+ Leveraging GitHub Actions Marketplace in Pipelines
	+ Pipeline Versioning
	+ Monitoring and Reporting in Pipelines
	+ Utilizing Workflow Templates in GitHub Actions
* Understanding the GitHub Interface
	+ Importance of Best Practices and Security
	+ Version Control Best Practices
	+ Code Protection and Branch Policies
	+ Secrets Management and Actions
	+ Security Testing and Continuous Monitoring
	+ Two-Factor Authentication (2FA) and Security Alerts
	+ User Access control
	+ GitHub Issue Tracker and Code Review Practices
	+ Code Scanning and Vulnerability Management
	+ Consistent Coding Standards and Security
	+ Ongoing Security Assessments
	+ Compliance and Regulatory Considerations
	+ Incident Response and Disaster Recovery
	+ Third-Party Dependency Management
	+ Documentation and Community Engagement
* The GitHub Marketplace
	+ What is the GitHub Marketplace?
	+ Benefits of GitHub Marketplace
	+ Exploring the Marketplace
	+ Installing and Using Marketplace Apps
	+ Building Your Own Marketplace App
* GitHub Actions – Advanced Scripting
	+ Personalizing Workflow Behavior
	+ Workflow Customization
	+ Managing Conditional Execution
	+ Managing Environment Variables
	+ Managing Secrets in GitHub Actions
	+ Utilizing Expressions and Functions
	+ Implementing Error Handling and Failure Conditions
	+ Understanding Artifacts and Outputs
	+ Boosting Optimization and Efficiency
	+ Effective Debugging Strategies in GitHub Actions
	+ Problem Matchers
	+ Integrating with External Services
* Microservice Development
	+ Foundational Enterprise Development Principles
	+ What is a Microservice?
	+ Understanding Monoliths
	+ What isn't a microservice?
	+ Monoliths Aren't Microservices
	+ Traditional Monolith (Java Example)
	+ SOA Architectures vs. Microservices
	+ Microservice (General Example)
	+ Microservices vs. Monoliths
	+ Microservice Development and Design
	+ Domain-Driven Design: Benefits
	+ Benefits of Microservices
	+ Microservices Drawbacks
	+ Docker and Microservices
	+ DevOps, Docker, Microservices and CI/CD
	+ Kubernetes
	+ Beyond Kubernetes with OpenShift
	+ OpenShift Architecture
	+ Client-Side Impact of Microservices
	+ Single Page Applications (SPA) and Interfaces (SPI)
	+ Data Tier Impact of Microservices
	+ AWS Lambdas
* Introduction to Flask and REST
	+ Why Python?
	+ Why Flask?
	+ The Flask Framework
	+ Why Flask for Microservices?
	+ Features of Flask
	+ Flask Microservice and Docker Support
	+ Microservices Development and Flask
	+ Flask Application Structure
	+ Flask Based Development
	+ Flask Leverages Python's Power
	+ REST and Microservices
	+ The Six RESTful Constraints
	+ Key RESTful Constraints Takeaways
	+ RESTful HTTP Methods
	+ HTTP Request Method Mapping
	+ RESTful Resources
	+ Path Variables in Flask
	+ Query Parameters in Flask
	+ Automatic JSON to Python Conversions
* Twelve-factor Applications [OPTIONAL]
	+ Twelve-factor Applications
	+ Twelve Factors, Microservices, and App Modernization
* Introduction to Automated Testing [OPTIONAL]
	+ Why Automated Testing?
	+ Automated Testing in the DevOps Lifecycle
	+ Types of Automated Tests
	+ Benefits of Unit Testing
	+ Benefits of Integration Testing
	+ Benefits of System Testing
	+ Benefits of Acceptance Testing
	+ Test Automation Tools
	+ Best Practices for Automated Testing
* Deployment Strategies (with a focus on Blue-Green) [OPTIONAL]
	+ Deployment Strategies Overview?
	+ Incremental Deployment Strategy
	+ Rolling Deployment Strategy
	+ Canary Deployment Strategy
	+ A/B Testing Strategy
	+ Blue-Green
* Conclusion