Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

GitHub Copilot as a Development Platform

Class Duration

14 hours of live training delivered over 2-4 days to accommodate varied scheduling needs

Student Prerequisites

- Experience with the Python and JavaScript programming languages
- Experience with GitHub Copilot or similar AI tool

Target Audience

This course is designed for experienced software engineers, DevOps professionals, and developer tool creators who are looking to build production-ready AI-powered tools using emerging standards like MCP and GitHub Copilot Extensions. It is particularly relevant for developers working on internal tooling, developer experience (DevEx) platforms, or AI system integration. The course is ideal for organizations seeking to streamline workflows, automate complex tasks, and deliver intelligent features across engineering teams.

Description

This course equips software developers and technical teams with the skills to build scalable, secure, and standards-based AI integrations using the Model Context Protocol (MCP) and GitHub Copilot Extensions. Starting with the fundamentals of MCP and its Python SDK, participants will learn how to define tools and resources, build robust servers, and extend capabilities to connect with external APIs, databases, and file systems. The second half of the course focuses on architecting and deploying Copilot Extensions—defining use cases, integrating with GitHub Apps, and building intelligent, multi-platform backends that respond to user input with contextual relevance. Emphasizing real-world examples, best practices, and production deployment strategies, this course provides everything needed to build developer-first AI automation solutions that scale.

Learning Outcomes

1. Explain the role of MCP in AI system integration and describe its architecture, including JSON-RPC messaging and transport layers.

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

2. Use the MCP Python SDK to define tools and resources with decorators, manage development environments, and create robust MCP servers.
3. Apply best practices in tool validation, error handling, and documentation for maintainable and secure tool development.
4. Integrate external APIs, databases, and file systems to expand server functionality using MCP.
5. Design and implement real-world AI tools such as task managers, weather bots, and search utilities.
6. Build and deploy GitHub Copilot Extensions using the SDK, including skillset development and backend service integration.
7. Configure and manage GitHub Apps, handle authentication, and integrate extensions into GitHub chat clients.
8. Plan, test, publish, and maintain Copilot Extensions with secure hosting, logging, versioning, and marketplace strategies.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students will need a web browser to access the cloud-based virtual machine for the class.

Training Topics

Introduction to MCP

- Why MCP matters
- “USB-C for AI” analogy
- Standardizing how AI connects to external systems
- Who uses MCP today

MCP Architecture

- Core building blocks
- JSON-RPC messaging
- Transport options

Python SDK Basics

- Installing the SDK

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Server vs. client roles
- Quickstart CLI tools

Development Setup

- Creating a new project
- Virtual environments
- Dependency management
- Editor integration
- Version control tips

Defining Tools

- The @tool decorator
- Simple arithmetic example
- Handling arguments
- Error handling and validation

Working with Resources

- The @resource decorator
- Dynamic URIs
- Returning structured data

Running & Testing Servers

- Running with mcp install
- Using mcp dev for hot-reload
- Debug logging
- MCP Inspector
- Troubleshooting failed calls

Expanding Capabilities

- Adding multiple tools
- External API integration
- Database access
- File system utilities

Real-world Examples

- Weather forecast server
- Task manager
- Knowledge base search

Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Best Practices

- Naming conventions
- Clear error messaging
- Consistent documentation

Security & Scaling

- Limiting tool exposure
- Handling sensitive data
- Rate limiting
- Authentication options
- Async and performance tuning
- Deployment strategies

Ecosystem & Future Directions

- Interoperability with Copilot
- Benefits of open MCP servers
- Evolution of the MCP spec
- Community contributions

Overview of Copilot Extensions

- Extension definition and purpose
- Skillsets vs. agents
- Supported platforms and clients

Planning Your Extension

- Identifying use cases
- Defining project scope
- Deciding on visibility (private, org, public)
- Matching extension type to goals
- Balancing automation with complexity

Setting Up Your Environment

- Installing Copilot Extensions SDK
- Preparing Node.js or Python environment
- Exploring sample agents
- Running setup scripts and tests

Building the Extension Backend

- Creating API endpoints or skillset files

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Handling requests and context inputs
- Generating structured responses
- Designing user prompts and actions
- Supporting multiple workflows

Hosting and Deployment

- Using tunneling (ngrok, Codespaces)
- Hosting on cloud providers
- Configuring HTTPS and callback URLs

GitHub App Integration

- Registering a new GitHub App
- Configuring permissions and events
- Setting callback URLs
- Adding inference metadata

Testing and Debugging

- Running local tests with CLI tools
- Capturing logs and request flow
- Handling invalid inputs and errors
- Simulating user interactions

Managing Extension Availability

- Restricting to private or org use
- Enabling invite-only testing
- Publishing to GitHub Marketplace
- Updating versions and changelogs
- Reviewing user feedback

Using the Extension

- Installing and authorizing the app
- Invoking with @extension-name in chat
- Testing across supported clients