



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

Distributed Task Automation with Python

Duration

2 days

Overview

The Distributed Task Automation with Python training course is for students with Python experience who desire to distribute automated tasks using Python. The course focuses on core Python skills needed to write almost any automation script and run it in a distribution fashion. Skills covered include containerization with Docker, accessing database data with Python, sending messages with Python and RabbitMQ, orchestrating distributed tasks with Celery, and executing remote SSH commands.

Objectives

- Deploy tasks at scale with Celery.
- Employ Python and Celery in a Containerized Environment
- Utilize the PostgreSQL database with Python
- Send Messages with Python and RabbitMQ
- Learn and apply basic and more advanced Python Celery concepts

Prerequisites

All students should have taken the Python Task Automation course or have significant experience with the topics covered in that course.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentations and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Python, Visual Studio Code, and Docker Desktop on their computers. Also, students will need permission and sufficient network bandwidth to install Python Packages, Visual Studio Extensions and download Docker images. If students are unable to configure a local environment, a cloud-based environment can be provided.



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

Outline - Development Environment (Very Quick Overview) - Configure VS Code for Python script development - Code Reformatting with Black - Debugging Python Scripts with VS Code - Simple Task Distribution - Use SSH to run Remote Commands - Use SSH to Download/Upload Files - Write Python Scripts to run code on a remote Linux Computers - Collect Data from remote Linux Computers - Containerization - What is a Container? - What is Docker? - What is Docker Hub? - Images and Containers - Create an Image with Dockerfile - Run Containers - Configure Containers with Environment Variables - Docker Compose - Docker Compose Networking - Docker Compose Volume - Remote Data Storage - What is Remote Data Storage? - Running a Data Storage in a Container - Running PostgreSQL in a Container - Running PostgreSQL Client Tool in a Container - Configure with Docker Compose - Read/Write Data to PostgreSQL with Python SQLAlchemy - Remote Message Broker - What is a Message Broker? - Running a Message Broker in a Container - Running RabbitMQ in a Container - Running RabbitMQ Client Tool in a Container - Configure with Docker Compose - Read/Write Data to RabbitMQ with Python and Pika - Task Automation at Scale with Celery - Overview of Celery and its features - Installing Celery and its dependencies - Setting up a simple Celery project - Celery Basics - Defining and running tasks - Task decorators and options - Passing arguments and results between tasks - Task retries and error handling - Monitoring and managing Celery workers - Advanced Celery Concepts - Task serialization and message brokers - Task routing and prioritization - Task result backends - Grouping and chaining tasks - Scheduling periodic tasks with Celery beat - Scaling and Deployment - Load balancing tasks with multiple workers - Deploying Celery in a production environment - Configuring Celery for high availability - Best practices for handling long-running tasks - Monitoring and performance tuning