DevOps Bootcamp

- 5 Day Course
- Lecture & Labs
- Every course includes the opportunity to earn a Python Basics certification from Alta3 Research.

Course Overview

Python is an interpreted, object-oriented, high-level language that empowers you to automate your work so it can be completed predictably and accurately. This freely available language is installed on all major platforms without a charge. Given Python's vast libraries, you'll have a head start programming most tasks.

Be it system admins, network, cloud, or storage engineers, all lessons in our courseware are highly relevant for scripting within the workplace, including; data retrieval and storage from the local system, working with RESTful APIs, and decoding JSON.

Class is a combination of live instructor demo and hands-on labs.

Managing networks can be repetitive and error-prone, but Python can make incredible changes to how you automate with all major (and most minor) network vendors. This course is driven by lessons and labs designed to utilize Python libraries designed to interact with and configure your network devices. At the conclusion of this course, you'll be empowered with the tools and skills necessary to take your network to the next level. This class is a combination of live demonstrations and hands-on labs with virtual network devices and endpoints as targets for your configuration.

Application Programming Interfaces (APIs) have become increasingly important as they provide developers with connectivity to everything from rich datasets in an array of formats (such as JSON) to exposing the configurability of software applications and network appliances. Lessons and labs focus on using Python to interact, design, and build APIs for the purposes of scripting automated solutions to complex tasks. Class is a combination of live demonstrations and hands-on labs.

This class prepares students for the Certified Kubernetes Application Developer (CKAD) exam. Kubernetes is a Cloud Orchestration Platform providing reliability, replication, and stability while maximizing resource utilization for applications and services. By the conclusion of this hands-on training you will go back to work with all necessary commands and practical skills to empower your team to succeed, as well as gain knowledge of important concepts like Kubernetes architecture and container orchestration. We prioritize covering all objectives and concepts necessary for passing the Certified Kubernetes Application Developer (CKAD) exam. You will build, command, and configure a high availability Kubernetes environment capable of demonstrating all "K8s" features discussed and demonstrated in this course. Your three days of intensive, hands-on training will conclude with a mock CKAD exam that matches the real thing.

As enterprises seek to deploy and maintain increasingly complex cloud infrastructure, there is a necessity to use "Infrastructure as Code" (IaC) tools, like Terraform. An open-source, state management tool developed by HashiCorp, Terraform allows developers to use a common coding interface to work through their various clouds safely and efficiently. Attendees will leave being able to write and understand Terraform code (HCL), have a clear understanding of Terraform's various components and supporting tools, as well as when to reach for Terraform over another IaC tool, such as Ansible.

Review this course online at https://www.alta3.com/courses/devops-bootcamp

Who Should Attend

- This course is an appropriate introduction to students of any background looking to get started with Python
- System Administrators
- Network Administrators and Engineers
- DevOps Engineers
- Management, Directors, VPs

This course was written for networking professionals looking to expand their capabilities by automating their workload with Python. This includes: Network Engineers, Network Architects, System Admins, DevOps Engineers, Cisco Certified Professionals (CCNA, CCNP, CCIE), and developers interested in network programmability with Python.

- System Administrators
- Network Engineers
- Software Developers
- Python Enthusiasts

- Anyone who plans to work with Kubernetes at any level or tier of involvement
- Any company or individual who wants to advance their knowledge of the cloud environment
- Application Developers
- Operations Developers
- IT Directors/Managers

- DevOps Engineers
- Software Developers
- Technical Managers and Leads
- System and Cloud Administrators
- Network Engineers and Developers

What You'll Learn

- Current Python3 Standard Library
- Popular 3rd party libraries
- Version control with git
- Git integration with popular SCM (GitHub)
- Parsing and building files
- Pull JSON from API queries
- Manipulate Excel and other popular formats with pandas dataframes
- Building feature rich charts and graphs
- Searching with Regular Expressions (regex)
- Best practice techniques
- AI LLM prompt engineering for generating and jumpstarting Python (and other) solutions

- Sending HTTP Requests to APIs with builtin and 3rd party libraries
- Opening Telnet Sessions with builtin libraries
- Automating SSH commands with Paramiko
- Performing ICMP Pings
- Transforming Data between JSON, YAML, and Python
- Retrieve and Push network device configuration via Netmiko and Napalm
- Use Pandas to manipulate data in a variety of formats (CSV, Excel, JSON)
- Database manipulation
- Capturing and Parsing Network Traffic

- Client side Python Scripting to RESTful (and non-RESTful) APIs
- Design RESTful API interfaces with Flask Web Framework

- Overview of Django
- Deploy your Python web apps as Docker containers
- Parse and manipulate popular data structures (JSON, CSV, Excel, and YAML) as pandas dataframes
- Best practice techniques

All topics required by the CKAD exam, including:

- Deploy applications to a Kubernetes cluster
- Use Kubernetes primitives to implement common deployment strategies (e.g. blue/green or canary)
- Define, build and modify container images
- Implement probes and health checks
- Understand multi-container Pod design patterns (e.g. sidecar, init and others)
- Understand ConfigMaps
- Create & consume Secrets
- Troubleshooting and debugging tools
- Provide and troubleshoot access to applications via services
- Use Ingress rules to expose applications

- Writing Terraform HCL code
- Deploying into common clouds such as AWS, Azure, Google Cloud, Docker, Oracle, Kubernetes, and VMWare
- Where Terraform fits in the Enterprise CI/CD model
- Differences between Terraform and Ansible
- Best practices
- Prepare for HashiCorp's Terraform Associate Certification

Outline

Python Basics

- 💻 Lecture + Lab: Installing Python
- 💬 Lecture: Python Basics
- 💻 Lecture + Lab: The Shebang Line and File Permissions
- 💻 Lecture + Lab: The Standard Library, functions, and print()
- 💻 Lecture + Lab: Collecting user input()

Common Objects

- 💬 Lecture: Python Lists
- 💻 Lecture + Lab: Working with Lists
- 💻 Lecture + Lab: List Objects and Methods
- 💻 Lecture + Lab: Slicing complex lists (lists within lists)
- 💬 Lecture: Python Dictionaries
- 💻 Lecture + Lab: Python Dictionaries
- 💻 Lecture + Lab: Getting dir(obj) help() and pydoc
- 💬 Lecture: Python Strings
- 💻 Lecture + Lab: String Methods

Interacting with the OS

- 💻 Lecture + Lab: Copying Files and Folders
- 💻 Lecture + Lab: Moving and Renaming Files and Folders

Conditionals

- 💬 Lecture: Conditionals
- 💻 Lecture + Lab: Testing if conditionals

- 💻 Lecture + Lab: IPv4 Testing with if
- 🏆 Challenge: Writing your own if-logic script
- 💻 Lecture + Lab: Using while, if, elif, else (Monty Python)
- 💻 Lecture + Lab: Debugging and Troubleshooting while, if, elif, else

Loops

- 💻 Lecture + Lab: Introduction to looping
- 💻 Lecture + Lab: Looping with for
- 💻 Lecture + Lab: Using for, range(), and with

Working with Files

- 💬 Lecture: Reading and Writing to Files
- 💻 Lecture + Lab: Parsing Log Files
- 💻 Lecture + Lab: Creating Output Files from Data Sets
- 💻 Lecture + Lab: Read from Files
- 💻 Lecture + Lab: Archive with zipfile

Beyond Basics

- 💻 Lecture + Lab: Creating Functions
- 💻 Lecture + Lab: pip, import and PyPi Packages to Know
- 💻 Lecture + Lab: Exploring Network Interfaces
- 💻 Lecture + Lab: Defining Functions
- 💻 Lecture + Lab: Scripting Commands with Python
- 💻 Lecture + Lab: try and except

Working with Data Sets

- 💻 Lecture + Lab: Producing Graphs and Charts
- 💻 Lecture + Lab: os.walk() the Directory Tree
- 💻 Lecture + Lab: Excel JSON and CSV - Intro to Pandas
- 💬 Lecture: Converting JSON to Python Data Types
- 💻 Lecture + Lab: Python, APIs, and JSON
- 💻 Lecture + Lab: requests library - Open APIs

Regular Expressions

- 💻 Lecture + Lab: Searching with Regular Expressions
- 💻 Lecture + Lab: Use RegEx to Search Text

Testing and Tools

- 💻 Lecture + Lab: Best Practice and pylint
- 💻 Lecture + Lab: Testing code with pytest
- 💻 Lecture + Lab: Packaging Python Projects
- 💻 Lecture + Lab: Running Python Scripts with Crontab

Classes and Objects

- 💻 Lecture + Lab: Creating Classes
- 💻 Lecture + Lab: Inheritance
- 💻 Lecture + Lab: Using Classes

Self-Study Opportunities

- 💻 Lecture + Lab: Argument Parsing
- 💻 Lecture + Lab: Unpacking Arguments
- 💻 Lecture + Lab: Automating SMTP and Extended SMTP
- 💻 Lecture + Lab: XML Parsing with ElementTree
- 💻 Lecture + Lab: Timestamping - import time datetime

Python for Network Automation - 💻 Lecture + Lab: Installing Python

Python for Network Automation - 💻 Lecture + Lab: Installing Python

APIs and API Design with Python - 💬 Lecture: Object Oriented Programming for APIs - 💬 Lecture: Practical Application of Lists - 💻 Lecture + Lab: Lists - 💬 Lec-

---

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Working with JSON Data - 💬 Lecture: Python Data sets vs JSON - 💻 Lecture + Lab: Python Data to JSON file

---

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

RESTful APIs - 💬 Lecture: Introduction to HTTP - 💻 Lecture + Lab: Standard vs. Third Party Libraries and Open APIs - 💻 Lecture + Lab: requests library - Open APIs - 💻 Lecture + Lab: requests library - RESTful GET and JSON parsing - 💬 Lecture: APIs

---

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

RESTful APIs beyond HTTP GET - 💬 Lecture: HTTP GET vs HTTP POST - 💻 Lecture + Lab: requests library - GET vs POST to REST APIs

---

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Authentication - 💻 Lecture + Lab: APIs and Dev Keys - 💻 Lecture + Lab: RESTful APIs and Dev Keys - 💬 Lecture: OAuth

Python Web-Server and Client - 💻 Lecture + Lab: Construct a Simple-HTTPServer and HTTP Client

---

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Best Practice - 💬 Lecture: RESTful API Best Practices

Python for Network Automation - 💻 Lecture + Lab: Installing Python

Building APIs with Flask - 💬 Lecture: Intro to Flask - 💻 Lecture + Lab: Building APIs with Python - 💬 Lecture: Introduction to Jinja - 💻 Lecture + Lab: Flask APIs and Jinja2 - 🏆 Challenge: Jinja2 - 🏆 Challenge: Jinja2 Solution - 💻 Lecture + Lab:

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Database Integration - 💬 Lecture: Learning sqlite3 - 💻 Lecture + Lab: Tracking API Data with sqlite3 - 💻 Lecture + Lab: Tracking Inventory with sqlite3

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

APIs within Enterprise - 💻 Lecture + Lab: Flask and waitress - 💻 Lecture + Lab: Running Flask in a Docker Container

FastAPI - 💻 Lecture + Lab: Introduction to FastAPI

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Building APIs with Django

- 💬 Lecture: Introduction to Django
- 💻 Lecture + Lab: Introduction to Django

Python for Network Automation - 💻 Lecture + Lab: Installing Python

---

Django Basics - 💻 Lecture + Lab: Intro to Django Views - 💻 Lecture + Lab: Controlling HTTP Response Codes - 💻 Lecture + Lab: Returning JSON with Django - 💻 Lecture + Lab: Making requests with Django

Python for Network Automation - 💻 Lecture + Lab: Installing Python

Django App - 💻 Lecture + Lab: Django App Design - To-Do app

Designing APIs - 💻 Lecture + Lab: Swagger

DevOps

- 💬 Lecture: Introduction to DEVOPS-4
- 💬 Lecture: The 12 factor APP # very short religious lecture. Most students will hear blah, blah, blah, Cover the core DEVOPs philosophy and what happens when ignored.

Front End Basics

- 💬 Lecture: HTML Basics
- 💻 Lecture + Lab: HTML Basics
- 💬 Lecture: Styling/CSS Basics
- 💻 Lecture + Lab: HTML Styling Lab
- 💻 Lecture + Lab: The Bare Essentials of JavaScript

Javascript Frameworks

- 💬 Lecture: Basics of NPM
- 💻 Lecture + Lab: Basics of NPM
- 💬 Lecture:/💻 Intro to Browserify

- 💬 Lecture: Intro to Webpack

Front End Projects

- ✏ PROJECT: Write a static web page
- ✏ PROJECT: Write a dynamic web template

Database Essentials

- 💻 Lecture + Lab: UML Style data modeling
- 💻 Lecture + Lab: Introduction to Tables and SQL
- 💻 Lecture + Lab: Tables with SQLite
- 💬 Lecture:/💻 Relational Databases

Dynamic Backend Project

- ✏ PROJECT: Creating and Managing Relational Databases

Web Access Essentials

- 💻 Lecture + Lab: Proxy vs Web Server
- 💻 Lecture + Lab: NGINX Config File
- 💻 Lecture + Lab: NGINX Server Blocks
- 💻 Lecture + Lab: NGINX Serving Static Files
- 💬 Lecture: NGINX Config Load Balancing
- 💻 Lecture + Lab: Termshark

Container Essentials

- 💬 Lecture: The Base DockerFile
- 💬 Lecture: Building on Base Image
- 💬 Lecture: Why Microservices?
- 💬 Lecture: Containerization
- 💬 Lecture: Docker Commands
- 💬 Lecture: Dockerfile
- 💻 Lecture + Lab: Running Flask Servers in a Docker Container
- 💬 Lecture: Tagging
- 💻 Lecture + Lab: Signing Docker Images
- 💻 Lecture + Lab: Pushing to GitLab Container Registry
- 💬 Lecture: Docker Multi-Stage
- 💻 Lecture + Lab: Docker SCRATCH container

Persisting Container Data

- 💻 Lecture + Lab: Using Environment Variables with Python
- 💬 Lecture:/💻 Mounts: Bind mounts & Volume Mounts

Container Deployment Tools

- 💬 Lecture: Docker Compose
- 💻 Lecture + Lab: Deploying a Microservice Using Docker Compose

Testing Python Applications

- 💻 Lecture + Lab: Writing a Python Unit Test Using Pytest
- 💻 Lecture + Lab: Creating a Unit Test Method
- 💻 Lecture + Lab: Determining What to Test/Verify
- 💬 Lecture: Developing a Test Suite for Python Applications
- 💻 Lecture + Lab: Python API Input Testing - Fuzzing (DAST)
- 💬 Lecture: Testing Python Applications in a Containerized Environment
- 💻 Lecture + Lab: Automate the Setup Between Unit Tests Using Parameterized Data

Building the DevOps/DevSecOps Pipeline

- 💻 Lecture + Lab: Automating Your Build Pipeline with Gitlab (gitlab-ci.yml)
- 💻 Lecture + Lab: Checking the Security of Your Application with GitLeaks (SAST) and Trivy (Vulnerability Scanning)

DevOps Project

- ✏ PROJECT: Launching your Python App in Docker Containers using Docker-Compose

---

Kubernetes Bootcamp - 💬 Lecture: Kubernetes Architecture - 💻 Lecture + Lab: Define, build and modify container images - 💬 Lecture: Pods and the Control Plane - 💻 Lecture + Lab: Deploy Kubernetes using Ansible

---

Pod Basics - 💬 Lecture: Namespaces and Fundamental Kubectl Commands - 💬 Lecture: Understanding YAML - 💬 Lecture: Pod Manifests - 💻 Lecture + Lab: Create and Configure Basic Pods - 💬 Lecture: Understanding API Versioning and Deprecations

Container Health, Security, and Observability

- 💬 Lecture: Kubectl port-forward
- 💻 Lecture + Lab: Debugging via kubectl port-forward
- 💬 Lecture: Kubectl exec and cp
- 💻 Lecture + Lab: Performing Commands inside a Pod
- 💬 Lecture: Readiness and Liveness Probes
- 💻 Lecture + Lab: Implement Probes and Health

Resource Management - 💬 Lecture: Limits, Requests, and Namespace ResourceQuotas - 💻 Lecture + Lab: Understanding and Defining Resource Requirements, Limits and Quotas - 💻 Lecture + Lab: Kubectl Top and Application Monitoring - 💬 Lecture: Admission Controller - 💻 Lecture + Lab: Create a LimitRange Admission

---

RBAC - 💬 Lecture: Role Based Access Control - 💻 Lecture + Lab: Service Accounts - 💬 Lecture: Contexts - 💻 Lecture + Lab: Cluster Access with Kubernetes Context

---

Logging - 💬 Lecture: Utilize Container Logs - 💻 Lecture + Lab: Kubectl Log Command - 💬 Lecture: Advanced Logging Techniques

---

Ephemeral Storage - 💬 Lecture: ConfigMaps and Volume Mounting - 💻 Lecture + Lab: Consistent Configuration with ConfigMaps - 💬 Lecture: Secrets - 💻 Lecture + Lab: Create and Consume Secrets

---

Persistent Storage - 💬 Lecture: Persistent Volumes, Claims, and StorageClasses - 💻 Lecture + Lab: Using PersistentVolumeClaims for Storage

---

Multi-Container Pod Design

- 💬 Lecture: Why Use Multi-Container Pods?
- 💻 Lecture + Lab: Creating Ephemeral Storage For Fluentd Logging Sidecar
- 💬 Lecture: Init Containers
- 💻 Lecture + Lab: Understand the Init Container Multi-Container Pod Design Pattern

Deployments - 💬 Lecture: Labels - 💻 Lecture + Lab: Understanding Labels and Selectors - 💬 Lecture: ReplicaSets - 💬 Lecture: Purpose and Advantages of Deployments - 💻 Lecture + Lab: Writing a Deployment Manifest - 💬 Lecture: Deployment Version Control - 💻 Lecture + Lab: Performing Rolling Updates and

---

Jobs and CronJobs - 💬 Lecture: Jobs and CronJobs - 💻 Lecture + Lab: Running and Executing a Job #### NetworkPolicy - 💬 Lecture: Controlling Connectivity with NetworkPolicy - 💻 Lecture + Lab: Namespace Network Policy

---

Services and Ingress

- 💬 Lecture: Networking with Services
- 💻 Lecture + Lab: Provide and troubleshoot access to applications via services
- 💬 Lecture: Ingress Controllers
- 💻 Lecture + Lab: Use Ingress Rules to Expose Applications

The Helm Package Manager - 💬 Lecture: Helm - 💻 Lecture + Lab: Using the Helm Package Manager to Deploy Existing Packages #### Extending Kubernetes - 💬 Lecture: Custom Resource Definitions - 💻 Lecture + Lab: Custom Resource Definitions (CRDs)

Terraform

- 💬 Lecture: Introduction to Terraform

Software Control Management

Overview of Terraform

- 💻 Lecture + Lab: Terraform Install

Terraform Modules

- 💬 Lecture: Terraform HCL Syntax
- 💻 Lecture + Lab: Up and Running with Terraform
- 💻 Lecture + Lab: Terraform Variables
- 💻 Lecture + Lab: Output Values
- 💬 Lecture: Avoid the :latest Tag
- 🏆 Challenge: Terraform and Docker

Beyond Basics

- 💻 Lecture + Lab: Terraform CLI Workspaces
- 💻 Lecture + Lab: Terraform Expressions and Errors
- 💻 Lecture + Lab: Resources - replace vs taint
- 💻 Lecture + Lab: Dynamic Operations with Functions
- 💻 Lecture + Lab: Creating a Terraform Module
- 💻 Lecture + Lab: Moving State - terraform state mv
- 💻 Lecture + Lab: Dynamic Provisioning with tfvars Files
- 💻 Lecture + Lab: Data Sources and HTTP Provider

Loops

- 💬 Lecture: for_each
- 💻 Lecture + Lab: Looping Constructs - for_each

Provisioning

- 💻 Lecture + Lab: local-exec Provisioner
- 💻 Lecture + Lab: Creating Delays

Terraform Cloud

- 💻 Lecture + Lab: Terraform Cloud and Terraform Enterprise
- 💻 Lecture + Lab: Triggering Cloud Builds via Git Commits

AWS

- 💻 Lecture + Lab: Terraform and AWS
- 💻 Lecture + Lab: Output Values and AWS
- 💻 Lecture + Lab: AWS and looping with count vs for_each
- 💻 Lecture + Lab: Correcting Resource Drift and AWS
- 🏆 Challenge: Terraform and AWS

Azure

- 💻 Lecture + Lab: Terraform and Azure

Google Cloud Platform

- 💻 Lecture + Lab: Terraform and Google Cloud Platform

Oracle

- 💻 Lecture + Lab: Terraform and Oracle Cloud Infrastructure

Terraform and Enterprise

- 💻 Lecture + Lab: Deploy a Go RESTful API microservice with Terraform
- 💬 Lecture: Terraform vs. Ansible
- 💻 Lecture + Lab: Terraform and Ansible

VMWare

- 💬 Lecture: Terraform and VMWare
- 💻 Lecture + Lab: Terraform and VMWare

Prerequisites

- Keyboard proficiency
- Recommended Prerequisite: Python Basics (5 days)
- Coding experience in another language serves as an adequate prerequisite

Although not required, students with some experience programming, or pre-existing knowledge of cloud architecture, will most appreciate the technical nature of this hands-on course.

Next Courses

- CKA
- Developing Microservices
- 3 Days
- Lecture and Hands-on Labs
- Includes all objectives found on HashiCorp's Terraform Associate Certification
- Jenkins Automation Server Essentials (2 days)
- Ansible Essentials (5 days)
- Go Essentials (5 days)
- Git and GitHub (2 days)
- Git and GitLab (2 days)

Certification

- Python Basics - Certification Project
- Alta3 Research Python 201 - API and RESTful API - Certification Project
- Terraform Essentials - Certification Project