Linux Fundamentals

- □ Linux Basic & Advanced Commands
- □ Linux Text Processing and Regex
- □ Linux Permission and Access Management
- □ Linux Storage Management Basics
- □ Linux Networking & Security Concepts
- □ Managing Services in Linux

Platform

Docker

- □ Docker Installation and Configuration
  - □ Introduction to Docker Community Edition
  - □ Installing Docker on Ubuntu
  - □ Selecting a Storage Driver
  - □ Running a Container
  - □ Configuring Logging Driver
  - □ Namespaces and Cgroups
- □ Dockerfile
  - □ Introduction to the Dockerfile
  - □ Dockerfile Instruction
  - □ Environment Variables
  - □ Build Arguments
  - □ Working with Non-privileged User
  - □ Order of Execution
  - □ Using the Volume Instruction
  - □ Entrypoint vs. Command
  - □ Using .dockerignore

- □ Image Creation, Management, and Registry
  - □ Introduction to Docker Images
  - □ Building Images
  - □ Using Multi-Stage Builds
  - □ Tagging
  - □ Optimizing Docker Images
  - □ Flattening a Docker Image to a Single Layer
  - □ Introduction to Docker Registries

- □ Distributing Images on Docker Hub
- □ Docker Compose
  - □ Compose Commands
  - □ Creating a Compose File
  - □ Using Volumes and Networking with Compose
  - □ Building a Microservice with Docker Compose
- □ Container Management
  - □ Inspecting Container Process
  - □ Having Container Start Automatically
  - □ Docker Events
  - □ Managing Stopped Container
- □ Docker Networking
  - □ Networking Overview
  - □ Networking Commands
  - □ Networking Containers
- □ Storage
  - □ Storage Overview
  - □ Volume Commands
  - □ Bind Mount vs. Docker Volume
- □ Docker Security
  - □ Understanding how uid and gid work in Docker containers
  - □ Run the Docker daemon as a non-root user (Rootless mode)
  - □ Using User Namespaces on Docker
  - □ Manage Volume Permission
  - □ Working with Secrets
  - □ Docker Content Trust

Kubernetes

- □ Getting Started
  - □ K8S Basics
  - □ K8s Architectural Overview
  - □ Building a Kubernetes Cluster
  - □ Using Namespaces in K8s
- □ Cluster Management
  - □ K8s Management Overview
  - □ Introduction to High Availability in K8s
  - □ Introduction to K8s Management Tools
  - □ Safely Draining a K8s Node

    - □ Upgrading K8s with Kubeadm
    - □ Backing Up and Restoring etcd Cluster Data
- □ Kubernetes Object Management
    - □ Working with kubectl
    - □ Managing K8s Role-Based Access Control (RBAC)
    - □ Creating Service Accounts
    - □ Inspecting Pod Resource Usage
- □ Pods and Containers
    - □ Managing Application Configuration
    - □ Managing Container Resources
    - □ Monitoring Container Health with Probes
    - □ Building Self-Healing Pods with Restart Policies
    - □ Creating Multi-Container Pods
    - □ Introducing Init Containers
- □ Advanced Pod Allocation
    - □ K8s Scheduling
    - □ Using DaemonSets
    - □ Using Static Pods
- □ Deployments
    - □ K8s Deployment Overview
    - □ Scaling Applications with Deployments
    - □ Managing Rolling Updates with Deployments
- □ Scaling Application Automatically
    - □ Horizontal Pod Autoscaler
    - □ Cluster Autoscaler
- □ Networking
    - □ K8s Networking Architectural Overview
    - □ Common Networking Requirements
    - □ Sample App Requirements
    - □ CNI Plugins Overview
    - □ Understanding K8s DNS
    - □ Using NetworkPolices
- □ Services
    - □ K8s Services Overview
    - □ Using K8s Services
    - □ Service Types
    - □ The Service Network
    - □ Discovering K8s Services with DNS
    - □ Managing Access from Outside with K8s Ingress
- □ Storage

    - □ K8s Storage Overview
    - □ The Container Storage Interface
    - □ Using K8s Volumes
    - □ Persistent Volumes and Persistent Volume Claims
    - □ Storage Classes
- □ RBAC and Admission Control
    - □ Authentication
    - □ Authorization
    - □ Admission Control
- □ Troubleshooting
    - □ Troubleshooting K8s Cluster
    - □ Checking Cluster and Node Logs
    - □ Troubleshooting Your Applications
    - □ Checking Container Logs
    - □ Troubleshooting K8s Networking Issues

Kubernetes Security

- □ Cluster Setup
    - □ Cluster Setup Intro
    - □ Restricting Default Access with NetworkPolicies
    - □ Allowing Limited Access with NetworkPolicies
    - □ Running a CIS Benchmark with kube-bench
    - □ Fixing Security Issues Detected by a CIS Benchmark
    - □ Implementing TLS with Ingress
    - □ Add TLS to a Kubernetes Service with Ingress
    - □ Securing Node Endpoints
    - □ Securing GUI Elements
    - □ Verifying Kubernetes Platform Binaries
- □ Cluster Hardening
    - □ Cluster Hardening Intro
    - □ Exploring Service Accounts
    - □ Restricting Service Account Permissions
    - □ Limit Service Account Permissions in Kubernetes
    - □ Restricting Access to the Kubernetes API
    - □ Keeping k8s Updated
- □ System Hardening
    - □ Minimizing Microservice Vulnerabilities
    - □ Managing Container Access with Security Contexts

    - □ Governing Pod Configurations with Pod Security Policies
    - □ Using Pod Security Policies
    - □ Using OPA Gatekeeper
    - □ Managing Kubernetes Secrets
    - □ Understanding Container Runtime Sandboxes
    - □ Creating a Container Runtime Sandbox
    - □ Understanding Pod-to-Pod mTLS
    - □ Signing Certificates
- □ Supply Chain Security
    - □ Supply Chain Security Intro
    - □ Minimizing Base Image Attack Surface
    - □ Whitelisting Allowed Image Registries
    - □ Validating Signed Images
    - □ Analyzing a Dockerfile
    - □ Analyzing Resource YAML Files
    - □ Scanning Images for Known Vulnerabilities
    - □ Scanning Images with an Admission Controller
    - □ Setting up an Image Scanner
    - □ Configuring the ImagePolicyWebhook Admission Controller
- □ Monitoring, Logging, and Runtime Security
    - □ Monitoring, Logging, and Runtime Security Intro
    - □ Understanding Behavioral Analytics
    - □ Analyzing Container Behavior with Falco
    - □ Ensuring Containers are Immutable
    - □ Understanding Audit Logs
    - □ Setting up Audit Logging

Infrastructure Automation

Ansible

- □ Understanding Core Components of Ansible
    - □ Understanding Core Components of Ansible
    - □ Ansible Configuration File
- □ Run Ad-Hoc Ansible Commands
    - □ Demonstration: Ansible Ad-Hoc Commands
- □ Inventory Management
    - □ Inventory Essentials and Inventory Variables
    - □ Variables and Inventories

  - □ Using YAML in Inventories
  - □ Dynamic Inventories
- □ Create Ansible Plays and Playbooks
  - □ Introduction to Playbooks and Common Modules
  - □ Create Playbooks to Configure Systems to a Specified State
  - □ Basic Playbook Syntax Demonstration
  - □ Use Variables to Retrieve the Results of Running Commands
  - □ Use Conditionals to Control Play Execution
  - □ Configure Error Handling
  - □ Selectively Run Specific Tasks In Playbooks Using Tags
- □ Create and Use Templates to Create Customized Configuration Files
  - □ Using Ansible Templates
- □ Work with Ansible Variables and Facts
  - □ Ansible Variables
  - □ Ansible Variables - Magic Variables and Jinja Filters
  - □ Ansible Variables - Variable Files
  - □ Ansible Facts
  - □ Working with Ansible Facts
- □ Create and Work with Roles
  - □ Working with Ansible Roles
  - □ Creating and Applying a Role in Ansible
  - □ Applying In-Line Roles and Role Dependencies
- □ Download roles from an Ansible Galaxy
  - □ Download Roles from Ansible Galaxy
- □ Managing Parallelism
  - □ Parallelism in Ansible
- □ Use Ansible Vault in Playbooks to Protect Sensitive Data
  - □ The Ansible-Vault Command
  - □ Using Vaults in Playbooks
- □ Install Ansible Tower and Use it to Manage Systems
  - □ Introduction to Ansible Tower
  - □ Installing Ansible Tower

Terraform

- □ Understanding Infrastructure As Code
  - □ IaC and Its Benefits
  - □ Cloud Agnostic IaC with Terraform
- □ IaC with Terraform

- ☐ What is the Terraform Workflow?
    - ☐ Terraform Init (Initializing the Working Directory)
    - ☐ Terraform Key Concepts: Plan, Apply, and Destroy
    - ☐ Resource Addressing in Terraform: Understanding Terraform Code
- ☐ Terraform Fundamentals
    - ☐ Installing Terraform and Terraform Providers
    - ☐ Terraform State: The Concept
    - ☐ Terraform Variables and Outputs
    - ☐ Terraform Provisioners: When to Use Them
- ☐ Terraform State
    - ☐ Terraform State Command
    - ☐ Local and Remote State Storage
    - ☐ Persisting Terraform State in AWS S3
    - ☐ Exploring Terraform State Functionality
- ☐ Terraform Modules
    - ☐ Accessing and Using Terraform Modules
    - ☐ Interacting with Terraform Module Inputs and Outputs
    - ☐ Building and Testing a Basic Terraform Module
- ☐ Built-in Functions and Dynamic Blocks
    - ☐ Terraform Built-in Functions
    - ☐ Terraform Type Constraints (Collections & Structural)
    - ☐ Terraform Dynamic Blocks
- ☐ Terraform CLI
    - ☐ Terraform fmt, taint, and import Commands
    - ☐ Terraform Workspaces
    - ☐ Debugging Terraform
    - ☐ Using Terraform CLI Commands (workspace and state) to Manipulate a Terraform Deployment

Vagrant

- ☐ Getting Started
- ☐ Concepts
    - ☐ What Is Virtualization?
    - ☐ What Is a Hypervisor?
    - ☐ What Is Vagrant?
- ☐ Installation
    - ☐ Linux

    - □ Mac
    - □ Windows
- □ Plugins
    - □ Plugin Overview
    - □ Provider Plugins
    - □ Additional Plugins
- □ The Basics
    - □ Boxes
    - □ The Vagrantfile
    - □ Vagrant Up!
    - □ Accessing the VM
    - □ VM Management
    - □ Snapshots
- □ Syncing Files
    - □ Basic Syncing
    - □ NFS
    - □ Rsync
    - □ SMB
- □ Networking
    - □ Port Management
    - □ Working with Multiple Machines
    - □ Private Networking
    - □ Public Networking
- □ Provisioning
    - □ Provisioning Overview
    - □ Shell
    - □ File
    - □ Ansible
    - □ Docker
- □ Boxes
    - □ Box Overview
    - □ Packaging an Existing Machine
    - □ Creating a Base Box
- □ Share
    - □ Vagrant Share Overview
    - □ HTTP Sharing
    - □ SSH Sharing
    - □ Vagrant Connect
- □ Advanced Vagrant

    - □ Windows Considerations
    - □ Tools and Scripting

Packer

- □ Packer
    - □ What Is Packer?
    - □ Why Use Packer?
    - □ Packer Breakdown
- □ Template Languages
    - □ Formatting a Packer Template in JSON
    - □ Formatting a Packer Template in HCL2
- □ Installation
    - □ Setting Up Packer
- □ Environment
    - □ Environment Configurations
    - □ Packer Plugins
- □ Building a Base Template
    - □ Builders
    - □ Communicators
    - □ Building the Image
    - □ Using Packer to Create an AWS AMI
- □ Provisioning
    - □ Bash
    - □ File
    - □ Ansible
- □ Post-Processors
    - □ Post-Processors
- □ Extending the Template
    - □ Parallel Builds
    - □ Creating Dev and Production Images with Packer in Parallel
    - □ HCL "Building Blocks"
- □ Debugging
    - □ The `-debug` Flag
    - □ The Breakpoint Provisioner
- □ Cloud
    - □ Building an Amazon EBS-Based Packer Image
    - □ Building an Azure ARM-Based Packer Image
    - □ Building a Google Compute-Backed Image with Packer

- □ Containers
  - □ Building a Docker Image with Packer
  - □ Building a LXD Image with Packer
- □ Virtualization
  - □ Building a VirtualBox Image with Packer
  - □ Building a Vagrant Box with Packer

NGINX

- □ Introducing NGINX
  - □ What is NGINX?
  - □ What is HTTP and How Does It Work?
  - □ NGINX vs. Apache
  - □ Differences Between NGINX and NGINX+
- □ Installing and Running NGINX
  - □ Installing NGINX on Ubuntu
- □ Basic Web Server Configuration
  - □ Understanding the Default NGINX Configuration
  - □ Simple Virtual Host and Serving Static Content
  - □ Error Pages
  - □ Access Control with HTTP Basic Auth
- □ Basic NGINX Security
  - □ Generating Self-Signed Certificates
  - □ Configuring the Host for SSL/TLS/HTTPS
- □ NGINX Rewrites
  - □ Cleaning Up URLs
  - □ Redirecting All Traffic to HTTPS
- □ NGINX Modules
  - □ Overview of NGINX Modules
  - □ Adding Functionality to NGINX with Dynamic Modules
  - □ HANDS-ON LAB Installing and Configuring NGINX as an HTTP Server
  - □ HANDS-ON LAB Forcing HTTPS Traffic with NGINX
- □ Reverse Proxy
  - □ What is a Reverse Proxy?
  - □ Preparing a Node.js Sample Application
  - □ Reverse Proxy with proxy_pass
  - □ Setting Up The LEMP Stack
  - □ FastCGI Proxy for PHP/WordPress with fastcgi_pass
  - □ Deploying a uWSGI Application

- □ Proxying to uWSGI Python Web Application with uwsgi_pass
    - □ Simple Caching for Static Content
    - □ Microcaching for Non-Personalized Dynamic Content
    - □ HANDS-ON LAB Using NGINX as a Reverse Proxy for an Application and Blog
- □ Load Balancing
    - □ Load Balancing to Multiple Servers
    - □ Examining Load Balancing Methods
- □ Logging
    - □ Configuring Logging
- □ Security
    - □ Improving SSL Configuration
    - □ Use Case: Generating SSL Certificates using Let's Encrypt
- □ Performance
    - □ Content Compressions and Decompression
    - □ Workers & Connections
    - □ HTTP/2
    - □ Use Case: PageSpeed by Google
    - □ HANDS-ON LAB Configuring NGINX for Performance

CICD & Software Collaboration Tools

Source Code Management - Git

- □ The Basics of Using Git
- □ Tags, Branching, Merging and Reverting
- □ Git’s Logs and Auditing
- □ Git Advanced Concepts
    - □ Merging vs. Rebasing
    - □ Resetting, checking out, and reverting
    - □ Advanced Git log
    - □ Git hooks
    - □ Refs and the reflog
    - □ Git submodules
    - □ Large Repositories in Git

Jenkins

- □ Jenkins and Continuous Integration (CI) / Continuous Delivery (CD)

    - □ CI/CD
    - □ What Is a Job?
    - □ What Is a Build?
    - □ Source Code Management (SCM)
    - □ Testing and Notifications
    - □ Distributed Builds
    - □ Plugins
    - □ Jenkins Rest API
    - □ Security
    - □ Artifacts and Fingerprints
- □ Installing Jenkins
    - □ Installation Walkthrough
- □ Features and Functionality
    - □ Jobs
    - □ Builds
    - □ Source Code Management (SCM)
    - □ Testing
    - □ Notifications and Alerts
    - □ Distributed Builds
    - □ Plugins
    - □ Rest API
    - □ Security
    - □ Artifacts
- □ Building CD Pipelines
    - □ Pipeline Concepts
    - □ Upstream, Downstream, and Triggers
    - □ Parameters
    - □ Promotions
    - □ Pipeline
    - □ Pipeline Multibranch and Repository Scanning
    - □ Pipeline Global Libraries
- □ Pipeline as Code
    - □ Jenkinsfile
    - □ Distributed Builds Architecture
    - □ Replaceable Agents
    - □ Master Agent Connectors and Protocol
    - □ Tool Installation on Agents
    - □ Cloud Agents
    - □ High Availability

GitHub CICD

- □ What Is GitHub Actions?
    - □ Understanding Workflows, Jobs, and Actions
    - □ Introducing Community Actions
    - □ Getting Running with Runners
    - □ HANDS-ON LAB Setting Up a Custom GitHub Actions Runner
    - □ Securing GitHub Actions
    - □ Designing Workflows and Pipelines
- □ Building Your Workflow
    - □ Introducing Your Microservice
    - □ Building Your Code
    - □ Storing Your Artifact
    - □ HANDS-ON LAB Creating a Release with GitHub Actions
    - □ Uploading to AWS
    - □ Deploying Your Function
    - □ GitHub Actions Workflow Design
    - □ Enhancing Your Workflow
    - □ Catching Errors Sooner: Code Quality Checks
    - □ Setting Up Non-Production Environments
    - □ HANDS-ON LAB Deploying a Static Site with GitHub Actions
    - □ Testing Before Production
    - □ Adding Documentation
    - □ HANDS-ON LAB Deploying Documentation to GitHub Pages
    - □ Reviewing the Workflow
    - □ GitHub Actions Workflow Enhancement

GitOps

- □ What is GitOps
    - □ Understanding GitOps Principles
    - □ Declarative Infrastructure and Infrastructure as Code (IaC)
    - □ Using a Version Control System as the Foundation of DevOps
    - □ Continuous Integration and Git Flow
    - □ Continuous Delivery vs. Continuous Deployment
    - □ Synchronizing Target Platforms with Upstream VCS Repos
    - □ Application Build and Deployment Automation
- □ Hands-On GitOps Labs

  - □ HANDS-ON LAB Installing and Configuring Flux with GitHub
  - □ HANDS-ON LAB Installing and Configuring Flux with GitLab
  - □ HANDS-ON LAB Operating and Troubleshooting Flux in a Kubernetes Cluster
  - □ HANDS-ON LAB Using Pull Requests For Release Gating
  - □ HANDS-ON LAB Deploying Applications with GitHub Actions Workflow and Flux
- □ GitOps with ArgoCD
  - □ Overview of Argo CD and its architecture
  - □ Install and configure ArgoCD
  - □ Use Argo CD CLI and Web UI to manage applications
  - □ Deploy applications using GitOps methodology
  - □ Synchronize the state of the application
  - □ Configure application rollbacks and rollouts
  - □ Use Argo CD to deploy Helm charts

Helm

- □ Introduction to Helm
  - □ What is Helm?
  - □ Deploying to Kubernetes without Helm
  - □ Deploying to Kubernetes Using Helm
- □ Getting Familiar with Helm
  - □ Installing Helm
  - □ Working with Chart Repositories
  - □ Updating Releases in Helm
  - □ HANDS-ON LAB Installing Helm
- □ Helm Charts
  - □ Getting into Helm Charts
  - □ Modifying Charts
  - □ Understanding the Language of Charts
  - □ Speaking the Language of Charts
  - □ Working with Subcharts
  - □ HANDS-ON LAB Modifying Helm Charts
  - □ HANDS-ON LAB Creating Helm Charts
- □ More Advanced Charts
  - □ Implementing Pre- and Post- Actions with Hooks
  - □ Testing Charts
  - □ Creating and Using Libraries
  - □ Validating Charts
  - □ HANDS-ON LAB Advanced Helm
- □ Administration and Plugins

  - □ Adding Role Based Access Control
  - □ Troubleshooting Helm
  - □ Working with Plugins
  - □ Exploring the Storage Backend
  - □ HANDS-ON LAB Troubleshooting Helm

ArgoCD

- □ Introduction to GitOps and Argo CD
  - □ Understanding GitOps
  - □ Introduction to Argo CD
  - □ Why GitOps with Argo CD?
- □ Setting Up Your Environment
  - □ Kubernetes Cluster Setup
  - □ Installing Argo CD on Your Cluster
- □ GitOps Workflows and Best Practices
  - □ Git Repository Structure for GitOps
  - □ Manifests, Helm Charts, and Kustomize
  - □ GitOps Best Practices
- □ Argo CD Deep Dive
  - □ Argo CD Architecture
  - □ Deploying a sample application to Argo CD
  - □ Deploying Helm charts to Argo CD
  - □ Deploying applications to Argo CD using Kustomize
  - □ Managing Secrets in GitOps
  - □ Synchronization and Rollbacks
- □ Advanced Argo CD Features and Integrations
  - □ Multi-Cluster Deployment with Argo CD
  - □ Introducing Argo CD ApplictionSets
  - □ Implementing Blue-Green Deployments
  - □ Implementing Canary Deployments
- □ Using Argo CD in CI/CD pipelines
  - □ Preparing our sample application (the currency converter)
  - □ Connecting Argo CD to GitHub
  - □ Setting up the CI pipeline using GitHub Actions
  - □ Using Argo CD for the Continuous Deployment (CD) part
- □ Using Argo CD Plugins for Custom Configuration Management
  - □ What are Argo CD plugins and why we may need them?
  - □ Creating an Argo CD plugin for envsubst

    - □ Introducing Terraform and our sample project
    - □ Using an Argo CD dynamic plugin with Terraform for a zero-touch workflow
- □ Argo CD Resource Hooks
    - □ Using PreSync Hook for performing database migrations
    - □ Using Sync Hook for clearing cache
    - □ Using PostSync and SyncFail for sending Slack notifications
    - □ Using the PostDelete hook for database backup and deletion
    - □ Enabling hostPath volumes in KinD clusters

DevOps Monitoring

- □ Prometheus Basics
    - □ What Is Prometheus?
    - □ Prometheus Architecture — Bird's-Eye View
    - □ Prometheus Use Cases — Strengths and Limitations
- □ Installation and Configuration
    - □ Building a Prometheus Server
    - □ Configuring Prometheus
    - □ Configuring an Exporter
    - □ HANDS-ON LAB Building a Prometheus Server
    - □ HANDS-ON LAB Collecting Linux Server Metrics with Prometheus
- □ Prometheus Data Model
    - □ What Is Time-Series Data?
    - □ Metrics and Labels
    - □ Metric Types
    - □ HANDS-ON LAB Locating Time-Series Data in Prometheus
    - □ HANDS-ON LAB Working with Prometheus Metric Types
- □ Querying
    - □ Introduction to Prometheus Querying
    - □ Query Basics
    - □ Query Operators
    - □ Query Functions
    - □ Prometheus HTTP API
    - □ HANDS-ON LAB Working with Prometheus Queries
    - □ HANDS-ON LAB Advanced Prometheus Queries
    - □ HANDS-ON LAB Using the Prometheus HTTP API
- □ Introduction to Visualization
    - □ Prometheus Visualization Methods
- □ Native Visualization Methods

- □ Expression Browser
- □ Console Templates
- □ Console Template Graph Library
- □ HANDS-ON LAB Building a Prometheus Console Template
- □ HANDS-ON LAB Using the Graph Library in a Prometheus Console Template

□ Grafana

- □ What Is Grafana?
- □ Installing and Configuring Grafana
- □ Building Prometheus Dashboards in Grafana
- □ HANDS-ON LAB Building a Grafana Instance to Work with Prometheus Data
- □ HANDS-ON LAB Building a Prometheus Dashboard in Grafana

□ Exporters

- □ Introduction to Exporters
- □ Application Monitoring
- □ Jobs and Instances
- □ HANDS-ON LAB Collecting Application Metrics with Prometheus
- □ HANDS-ON LAB Docker Daemon Monitoring with Prometheus
- □ HANDS-ON LAB Docker Container Monitoring with Prometheus
- □ HANDS-ON LAB Kubernetes Monitoring with Prometheus

□ Prometheus Pushgateway

- □ Introduction to Pushgateway
- □ Installing Pushgateway
- □ Pushing Data to Pushgateway
- □ HANDS-ON LAB Installing Prometheus Pushgateway
- □ HANDS-ON LAB Monitoring a Batch Job with Prometheus Pushgateway

□ Recording Rules

- □ Introduction to Recording Rules
- □ Configuring Recording Rules
- □ HANDS-ON LAB Using Prometheus Recording Rules

□ Alertmanager Setup and Configuration

- □ What Is Alertmanager?
- □ Installing Alertmanager
- □ Alertmanager Configuration
- □ High Availability and Alertmanager
- □ HANDS-ON LAB Installing Prometheus Alertmanager
- □ HANDS-ON LAB Configuring Prometheus Alertmanager
- □ HANDS-ON LAB Configuring Prometheus Alertmanager for High Availability

□ Prometheus Alerts

- □ Alerting Rules
- □ Managing Alerts

    - □ HANDS-ON LAB Configuring Prometheus Alerts
    - □ HANDS-ON LAB Advanced Configuration for Prometheus Alerts
- □ Using Multiple Prometheus Servers
    - □ High Availability
    - □ Federation
    - □ HANDS-ON LAB Building a Highly Available Prometheus Setup
    - □ HANDS-ON LAB Implementing Hierarchical Federation With Prometheus
- □ Security
    - □ Prometheus Security Assumptions
- □ Client Libraries
    - □ Introduction to Prometheus Client Libraries
    - □ Using the Prometheus Java Client Library
    - □ HANDS-ON LAB Using the Java Client Library for Prometheus

Scripting & Automation

Introduction to JavaScript

- □ The History of JavaScript
    - □ Implementing Client-Side Scripting
    - □ Creating a Modern and Dynamic Internet
- □ Client-Side Scripting Basics
    - □ Exploring the Document Object Model (DOM)
    - □ Understanding JavaScript Syntax (Hello World)
    - □ Working with Image Tags (Cats and Dogs)
    - □ HANDS-ON LAB Getting Stylish with JavaScript
- □ More Advanced Client-Side Scripting
    - □ Using Variables and Arrays
    - □ Working with Conditionals and Looping
    - □ Understanding the Basics of Functions
    - □ Introducing JavaScript Frameworks
- □ Server-Side Basics
    - □ Exploring Server-Side JavaScript with NodeJS
    - □ Introducing REST and APIs
    - □ HANDS-ON LAB Creating an HTTP Server in NodeJS
- □ Build Toolchain and Unit Testing

Introduction to Python Development

- □ Environment Setup
    - □ Installing Python 3.7 on CentOS 7
    - □ (Optional) Installing Python 3.7 on Debian/Ubuntu
    - □ Picking a Text Editor or IDE
    - □ (Optional) Setting Up a Vim Development Environment
- □ Running Python
    - □ Using the REPL (Read, Evaluate, Print, Loop)
    - □ Creating and Running Python Files
    - □ Using Comments
- □ Built-in Data Types
    - □ Strings
    - □ Numbers
    - □ Booleans and None
    - □ Working with Variables
    - □ Lists
    - □ Tuples & Ranges
    - □ Dictionaries (dicts)
- □ Control Flow
    - □ Conditionals and Comparisons
    - □ Logic Operations
    - □ The while loop
    - □ The for Loop
- □ Encapsulating Code
    - □ Writing Functions
    - □ HANDS-ON LAB Solving Problems with Built-in Python Types
    - □ HANDS-ON LAB Utilizing Control Flow Structures in Python
- □ Object-Oriented Programming Basics
    - □ Creating Classes
    - □ Composition
    - □ Inheritance
    - □ Polymorphism
    - □ HANDS-ON LAB Modeling Data with Classes in Python
- □ Using Packages
    - □ Using Standard Library Packages
    - □ Working with Third-Party Packages
- □ Common Uses and Debugging
    - □ Interacting with Files

- □ Environment Variables
  - □ Error Handling
  - □ Decorators
  - □ Breakpoint Debugging with PDB
- □ Building CLIs with Python
  - □ Project Setup
  - □ Setting Up External Dependencies
  - □ Building the CLI: Handling Arguments and Flags
  - □ Interacting with External Processes: Utilizing pg_dump
  - □ Storing Data Locally
  - □ Wiring the Pieces Together
  - □ Distributing the Package
  - □ HANDS-ON LAB Building a Command Line Tool with Python
- □ Intro to Web Development
  - □ Visualizing Web Development
  - □ Project Setup
  - □ Creating the Flask Application and Database
  - □ Modeling Data with an Object-Relational Mapper
  - □ Building User Registration
  - □ Building User Authentication
  - □ Implementing Notes CRUD - Creating and Reading
  - □ Implementing Notes CRUD - Updating and Deleting
  - □ HANDS-ON LAB Building a Web Application with Python and Flask

System Tooling with Go

- □ Introduction to Go
  - □ The History and Benefits of Go
- □ Environment Setup
  - □ Installing Go on Unix Systems
  - □ HANDS-ON LA Installing and Setting Up Go on Linux
- □ Running Go
  - □ Creating and Running Simple Go Programs
  - □ Using Comments
- □ Common Data Types
  - □ Strings and Characters
  - □ Numbers
  - □ Booleans and nil
  - □ Working with Variables

    - □ Arrays and Slices
    - □ Maps
- □ Control Flow
    - □ Conditionals
    - □ The for Loop
- □ Basic Interactions
    - □ Reading the Documentation
    - □ Function Basics
    - □ Reading User Input
    - □ Interacting with Files
- □ Building a CLI
    - □ Creating a CLI
    - □ HANDS-ON LAB Building a Go Command Line Tool
- □ Working with Third-Party Packages
    - □ Downloading and Installing Packages
    - □ Revamping a CLI Using the Cobra Package
- □ Distributing Go
    - □ Compiling a Cross-Platform Go Application
    - □ HANDS-ON LAB Building a Cross-Platform Go Binary

Bash Scripting