

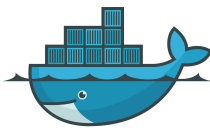


Platform scenario and sandbox topics

Interactive learning is the fastest way to explore a new technology. Because you're not just reading about it—you're also manipulating it in real time to discover how it works. In the past, companies had to set up complex environments and datasets for their teams to get their heads around cloud-based infrastructure, orchestration, NLP, and more. With O'Reilly online learning, you get access to over 500 scenarios and sandboxes ready with your browser's click.

The best way to understand how O'Reilly online learning can help drive your business is to dive in and start exploring at learning.oreilly.com/interactive.

On our radar



.NET

- Deploy ASP.NET Core as Docker Container
- .NET Sandbox
- C# Next Steps: Looping and Collections
- C#: Build and Run Console Apps
- C# Next Steps: Methods
- C#: Data Types, Variables, and Constants
- C# Next Steps: Interfaces and Inheritance
- C#: Class and Structs
- C#: Working with Strings
- C#: Looping and Arrays
- C# Next Steps: Classes and Objects
- C# Next Steps: Logic and Conditionals

Adobe design tools

- Edge Detection Using Gradients

Algorithms

- Get Started with Hashcat

Auditing and risk management

- Value at Risk and Its Application with Python

C++

- C++ Sandbox

Calculus

- Simulating Geometric Brownian Motion

Cloud computing

- Cloud Native Data: Materialized Views
- Cloud Native Data: Data Partitioning

Computer vision

- Training a Dense Neural Network
- Training a Single Layer Model
- Face Detection Using OpenCV
- Loading and Preparing the Fashion MNIST Data
- Hough Transform for Lines and Circles
- Early Stopping to Save Time and Processing Power

Container orchestrators

- Launch a Single Node Kubernetes Cluster
- Kubernetes Fundamentals: First Kubernetes Application
- Kubernetes Pipelines: Helm
- Get Started with Networking on Kubernetes
- Deploy Microk8s on Ubuntu
- Create Kubernetes Ingress Routing



Container orchestrators (cont.)

- Kubernetes Observability: Basics
- Deploy Containers to Kubernetes Using Kubectl
- Kubernetes Containers: Go Apps in Containers on Kubernetes
- Kubernetes Applications: Nginx
- Kubernetes Containers: Decomposing Container Images
- Kubernetes Fundamentals: Minikube
- Kubernetes Fundamentals: Spring Boot to Kubernetes
- Kubernetes Fundamentals: Kubernetes Logs
- Kubernetes Observability: Scaling Your Applications, Automatically
- Kubernetes Fundamentals: Jobs
- Kubernetes Fundamentals: Kubernetes API
- Deploy a Virtual Kubelet
- Deploy Guestbook Web App Example on Kubernetes
- Kubernetes Fundamentals: ConfigMaps and Secrets
- Kubernetes Applications: RabbitMQ
- Kubernetes Containers: Node.js to Kubernetes
- Liveness and Readiness Healthchecks in Kubernetes
- Kubernetes Observability: Logging
- Kubernetes: Create and Inspect a Pod
- Run Stateful Services on Kubernetes
- Kubernetes Pipelines: SonarQube
- Deploy Containers to Kubernetes Using YAML
- Kubernetes Serverless: Kubeless
- Deploying Python APIs on Kubernetes: Deploying a Development Kubernetes Cluster
- Kubernetes Fundamentals: Sidecar Containers
- Kubernetes Applications: Istio
- Kubernetes Pipelines: Tekton Pipelines
- Launch a Multi-Node Kubernetes Cluster Using Kubeadm
- Kubernetes Fundamentals: Distilled JRE Apps in Containers
- Kubernetes Extensibility: Develop an Operator with KUDO
- Kubernetes Pipelines: Python Pipeline to Kubernetes
- Use Kubernetes to Manage Secrets and Passwords
- Kubernetes Fundamentals: Pods to Services Communication
- Kubernetes Serverless: OpenFaaS
- Kubernetes Chaos: Kube Monkey
- Kubernetes Extensibility: Develop an Operator with Kubebuilder
- Kubernetes Data: Persistent Data
- Cloud Native Data: Data Stores on Kubernetes
- Kubernetes Applications: Shiny R
- Kubernetes: Fix a Misconfigured Pod
- Kubernetes: Perform Rolling Updates for a Deployment
- Kubernetes Data: Running PostgreSQL
- Kubernetes Skills: Application Run and Update with YAML Manifests
- Kubernetes: Create an Init Container
- Add Additional Nodes to an Existing Kubernetes Cluster
- Kubernetes Data: Dev Workflows for Data-Driven Applications
- Kubernetes: Define and Mount a Persistent Volume
- Deploying Python APIs on Kubernetes: Deploying a Python Application Using Kubernetes
- Kubernetes: Define and Query Labels and Annotations
- Kubernetes: Route Traffic to Pods from Inside and Outside of a Cluster
- Kubernetes Pipelines: Registries
- Kubernetes Extensibility: Custom Resource Definitions
- Kubernetes Containers: Rust to Kubernetes
- Kubernetes: Define a Pod's Readiness and Liveness Probe
- Kubernetes: Restricting Access to and from a Pod
- Kubernetes: Implement the Adapter Pattern
- Kubernetes Networking: Istio Traffic Management
- Kubernetes: Create and Use a ConfigMap



Container orchestrators (cont.)

- Kubernetes: Use a Service Account
- Kubernetes: Create and Use a Secret
- Kubernetes: Create a Scheduled Container Operation
- Kubernetes: Create a Security Context for a Pod
- Kubernetes Skills: Assembling a Multi-Component Application
- Kubernetes: Define a Pod's Resource Requirements
- Kubernetes Serverless: Nuclio
- Kubernetes Chaos: Chaoskubed
- Kubernetes Pipelines: Consumer-Driven Contracts with Kubernetes
- Kubernetes Chaos: KubelInvaders
- Kubernetes Containers: Kotlin to Kubernetes
- Kubernetes Chaos: Chaos Mesh
- Kubernetes Chaos: Litmus
- Kubernetes Extensibility: Kubectl CLI Plugins
- Kubernetes Chaos: Pure Chaos
- Kubernetes Machine Learning: BentoML to Kubernetes
- Kubernetes Extensibility: Kubectl CLI Plugin Development

Container platforms

- Launch a Docker Container
- Build Container Images in Docker
- Deploy a Static HTML Website as Container
- Deploy Your First Docker Container
- Docker Sandbox
- Dockerize Node.js
- Load Balancing Docker Containers Using NGINX Proxy
- Deploying Python APIs on Kubernetes: Containerizing a Python Application
- Create Optimized Docker Images using Multi-Stage Builds
- Create Networks Between Containers using Networks in Docker
- Hack an Elasticsearch Docker Container
- Create a Docker Overlay Network
- Persist Data Using Docker Volumes
- Read Vault Secrets from Docker Containers

- Create Data Containers with Docker
- Orchestrate Containers Using Docker Compose
- First Steps with Apache Airflow: Deploying Airflow Using Docker
- Add Healthchecks to Docker Containers
- Launch a Private Docker Registry
- Run Docker Containers From Rootless Users
- Launch a Consul Cluster using Docker
- Adding Docker Metadata & Labels
- Create Networks Between Containers Using Links in Docker
- Format Docker PS Output
- Deploy Docker Compose Files with Kompose
- Test CPU Performance with a Docker Container
- Manage Docker Container Log Files
- Deploy Kotlin and a Wasabi HTTP Server as a Docker Container
- Optimize Builds with Docker OnBuild
- Optimize a ASP.NET Core Docker Container
- Access .ssh Keys During Docker Build
- See Container Metrics with Docker Stats
- Ensure Docker Container Uptime with Restart Policies

Containers

- What Is a Container Image?
- Understand Container Security with CGroups and Namespaces
- Securing Containers with User Namespaces
- Get Started with Container Security and Seccomp
- Use No New Privileges Flag to Secure Containers

Data models

- Volatility Modeling and Its Application with Python



Data science

- Cleaner Code for Data Science: Writing Unit Tests in Python
- Data Preprocessing: Feature Selection
- Data Preprocessing: Feature Engineering Basics
- Data Preprocessing: Start to Finish
- Data Preprocessing: Normalizing and Scaling Data
- Data Preprocessing: Standardization and Modeling with knn
- Data Preprocessing: Feature Engineering and Extracting Features from Text
- Cleaner Code for Data Science: Logging in R
- Cleaner Code for Data Science: Writing Unit Tests in R

Data science tools

- R Cookbook: Some Basics
- Select and Filter Data with Pandas
- R Sandbox
- Getting Started with Python and Libraries
- Linear Regression from Scratch Using NumPy
- Aggregate Data with Pandas
- Merge and Join Data with Pandas
- Generate Summary Statistics with Pandas
- Concatenate Multiple Files with Pandas
- Understanding Eigenvalues and Eigenvectors Using Python
- Cleaner Code for Data Science: Using Auto Documentation Frameworks in R
- Introduction to Matrix Algebra Using Python
- Using gazpacho and pandas to Retrieve the Contents of an HTML Table
- Handling Categorical Data: Encoding Categorical Features

Data visualization

- Build a Node-RED COVID-19 Dashboard

Finance

- Tools Used in Finance and Python

Frontend web development

- Render HTML Templates with Jinja2

Functional programming

- Functional Python: Simplifying List Processing with itertools
- Java Concurrent Programming: Basics and Thread Pools
- Functional Python: Reducing Duplication with Higher-Order Functions
- Java Concurrent Programming: Creating a CompletableFuture
- Functional Python: Reducing Boilerplate with Decorators
- Java Concurrent Programming: Functional Programming a CompletableFuture
- F# Sandbox
- Using Data Types and Variables in Rust
- Understanding Ownership and Borrowing in Rust
- Using Comments, Functions, and User Input in Rust
- Java Concurrent Programming: Learning New Advancements in CompletableFuture
- Using Control Flows and Mathematical Operations in Rust
- Java Concurrent Programming: Making an Incomplete Future
- Rust Sandbox

Git

- Commit Files with Git
- Commit Changes with Git
- Work Remotely with Git
- Fix Merge Conflicts with Git
- Find Bugs with Git
- Undo Changes with Git
- Experiment Using Branches with Git
- Git Sandbox
- Re-Write History with Git



Go

- Testing Go: Implementing a Test Helper
- Go Sandbox
- Testing Go: Testing in Isolation by Mocking an Interface
- Go Modules: Initializing and Using Modules for a Project
- Testing Go: Testing a Cobra CLI Command
- Testing Go: Implementing a Table-Driven Test
- Testing Go: Avoiding Network Calls and Mocking the Response
- Testing Go: Implementation and Execution of a Test Case with the testing Package
- Testing Go: Using a Golden File for JSON Data
- Go Modules: Vendoring Dependencies
- Go Modules: Handling Transitive Dependencies
- Go Modules: Migrating from Glide to Go Modules
- Testing Go: Setting Up and Tearing Down Test Fixtures
- Go Modules: Executing Module Queries
- Go Modules: Verifying and Tidying Dependencies
- Testing Go: Implementation and Execution of a Test Case with Testify
- Testing Go: Producing Code Coverage Metrics and Reports
- Go Modules: Using a Dependency with Multiple Major Versions
- Go Modules: Using JFrog GoCenter To Proxy Dependencies

Graph analytics

- Portfolio Theory and Its Application in Python

Graphics programming

- Linear Algebra in Image Processing Using Python
- Use Monte Carlo to Price Call Options
- Using gazpacho and urllib to Retrieve and Download Images, Videos, and Audio Clippings
- Use Monte Carlo to Price Put Options

Integration solutions

- Handling Categorical Data: Handling Imbalanced Classes

Java

- Java Sandbox
- Spring Boot: Create a REST Controller
- Java Challengers: Adding Elements to an ArrayList
- Spring Boot: Create a New Project
- Java 11 File API: Read and Write a String to a File
- Java Challengers: Every Class in Java Is an Object
- Maven: Install Maven and Create Your First Application
- Spring Boot: Dependency Injection
- Java Challengers: Creating a Class
- Java Lambdas: Packaging Reusable Logic
- Java Challengers: Sort String Elements from an ArrayList
- Java 11 Network API: Use HttpClient for Asynchronous HTTP Communication
- Java Challengers: Creating Customized Sorting for Customized Objects
- Spring Boot: Handling Errors
- Java IO: TextIO
- Spring Boot: Accessing Data with JdbcTemplate
- Java Lambdas: Sorting Data
- Spring Boot: Handling JSON
- Java Lambdas: Filtering Large Data Sets
- Java Challengers: Using Polymorphism with the List Interface
- Maven: Using Maven Plugins
- Spring Boot: Connect to a Database
- Maven: Compiling, Packaging, and Running Your Application
- Java 11 Network API: Use HttpClient for Synchronous HTTP Communication
- Java Challengers: Using Basic Looping
- Java Challengers: Manipulating an Instance
- Spring Security: Understanding the Default Configuration
- Java Challengers: Creating a Constructor
- Maven: Using Maven Archetypes
- Maven: Dependency Management
- Spring Boot: Populate a Database



Java (cont.)

- Java Challengers: Implementing a Method
- Java 11 String API: Check for Empty or Whitespaces in a String
- Java Challengers: Manipulating Primitive Variables
- Java 11 Collection API: Convert a Collection into an Array
- Java 11 String API: Process Multiline Strings
- Java Lambdas: Mapping Complex Data
- Java 11 Functional API: Check Whether a Value is Empty in Optional
- Java 11 String API: Repeat a String
- Spring Security: Creating a Custom User
- Spring Boot: Manipulating Data with JdbcTemplate
- Spring Security: Defining Authorization Based on User Roles
- Java Challengers: Multiple Values with Arrays
- Java Challengers: Handling Flow Control
- Java Challengers: Local Variable Scope
- Spring Boot: Using the Command Line Runner Interface
- Java 11 Functional API: Define Filtering Negation for Predicate
- Java Core Beam: ParDo OneToMany
- Java Core Beam: GroupByKey
- Java Challengers: Inheritance with the Animal Class
- Spring Security: Adding URL-Based Access Control
- Java Core Beam: ParDo
- Java Challengers: Returning Values from a Method
- Java Challengers: Abstract Classes and Polymorphism with the Animal Class
- Java Challengers: Using Polymorphism with the Animal Methods
- Java 11 String API: Remove Leading and Trailing Whitespaces from a String
- Java 11 CLI: Launch Single-File Program without Compilation
- Java Challengers: Initializing an Object with Values in the Constructor
- Java Challengers: Exposing an Attribute with a Public Method

- Java Challengers: Creating Private Fields and Methods
- Java Challengers: Printing Values with Access Methods

JavaScript

- Angular Fundamentals: Creating and Using Reusable Angular Services
- Angular Fundamentals: Template Variables and Styling Components
- Angular Fundamentals: Template Syntax and the Safe Navigation Operator
- Angular Fundamentals: Routing and Navigation Basics
- Angular Fundamentals: Creating a Template-Based Form
- Angular Fundamentals: Data-Bound Components and Component Communication
- Angular Fundamentals: Content Trimming and Conditional Content Styling
- Angular Fundamentals: Using Data Input in a Template-Based Form
- Angular Fundamentals: Creating Models for Type Safety
- Angular Fundamentals: Lazy Load a Feature Module
- Angular Fundamentals: Preloading Data and Styling Active/Inactive Links
- Angular Fundamentals: Guarding Against Route Activation and Deactivation

JUnit

- JUnit 5 Testing Basics: Descriptive Test Names
- JUnit 5 Testing Basics: Defining Assertions
- JUnit 5: Nested Tests
- JUnit 5: Fixture Setup and Teardown
- JUnit 5 Testing Basics: Disabling Tests
- JUnit 5: Expecting Thrown Exceptions



JVM languages

- Scala Sandbox
- Kotlin Sandbox
- Scala Implicits: Identifying Implicits in Scala
- Scala Implicits: Using an Implicit Value
- Scala Implicits: Creating an Implicit Converter
- Scala Implicits: Creating an Implicit Wrapper
- Scala Implicits: Looking up Implicit Bindings with Implicitly

Kali linux

- Kali Linux: Disassemble Malware Sample
- Kali Linux: Analyze Malware Looking at Text Strings
- Kali Linux: Identify Packed Executables Using Pescan
- Kali Linux: Perform an Anti-Virus Scan on a Malware Sample
- Kali Linux: Get File Details about Malware Sample

Linear algebra

- Introduction to Vector Algebra Using Python
- Exploring Vector Spaces Using Python
- System of Linear Equations Using Python

Linux

- Ubuntu Sandbox
- Linux Sandbox

Log data

- Fluentd: Logging Model Usage
- First Steps with Apache Airflow: Viewing the Logs of Your DAG

Machine learning

- Linear Regression Using Scikit-Learn
- Tensorflow Sandbox
- Python ML Cookbook: Vectors, Matrices, & Arrays
- Python ML Cookbook: Wrangling Data (Part 2)

- Python ML Cookbook: Handling Numerical Data
- Deploy Kubeflow
- Data Preprocessing: First Steps with pandas and scikit-learn
- Build and Save Robust Machine Learning Models
- Keras: Build and Debug a Simple Neural Network
- Deploy MLflow
- Machine Learning: Create a Basic ML Model Using Python
- Unix: Set Up a Basic ML Development Environment
- Serve Machine Learning Models with Flask
- Multiple Linear Regression Using Python
- NLP for TensorFlow: Introduction to TensorFlow
- Flask: Create an Endpoint for Your ML Model
- Deploy Machine Learning Apps with Dokku
- DVC: Creating a Machine Learning Pipeline
- Scikit-Learn: Build a Nearest Neighbors Recommendation Engine
- Finding Outliers: Anomaly Detection in scikit-learn
- LightFM: Build an Explicit Feedback Recommendation Engine
- Reducing Dimension with PCA in scikit-learn
- Polynomial Regression Using Python
- NLP for TensorFlow: Text Classification
- NLP for TensorFlow: Text Sentiment Analysis
- NLP for TensorFlow: Text
- MLflow: Tracking Experiment Runs
- Getting Started with Clustering in scikit-learn
- Spotlight: Build an Implicit Feedback Recommendation Engine
- Regularization in Regression Using Python
- NLP for TensorFlow: Text Translation with RNN
- NLP for TensorFlow: Word Embeddings



Machine learning (cont.)

- Optimization for Deep Learning: Using Gradient Descent to Train Your First Model in TensorFlow
- NLP for TensorFlow: Text Generation
- Presto First Steps: Expanding Your Installation to be a Cluster
- Optimization for Deep Learning: Putting It All into Practice with a Cats Versus Dogs Image Classifier in TensorFlow
- Optimization for Deep Learning: Optimizing the Hyperparameters for Faster Training with TensorFlow
- Putting It All into Practice with Intent Classification in TensorFlow
- TensorFlow Lite: Use Transfer Learning to Classify Pneumonia with TensorFlow
- Reproducible Deep Learning: Semantic Segmentation on Oxford Pets Dataset
- TensorFlow Lite: Use TensorFlow Lite Interpreter & Android Studio to Deploy a Model to Mobile
- Optimization for Deep Learning: Getting Started with Stochastic Methods in TensorFlow
- Spectral Clustering from Scratch Using Python
- Reproducible Deep Learning: Achieving 100% Reproducibility
- TensorFlow Lite: Convert a TensorFlow Pneumonia Classifier to TensorFlow Lite with Quantization
- Variance Reduction Techniquesââ Control Variates
- Reproducible Deep Learning: Random Seeds and Process-Parallelism
- Variance Reduction Techniquesââ Antithetic Variables
- Reproducible Deep Learning: Identifying the Reproducibility Challenge

Market segmentation and pricing

- Pricing European Options
- Asset Pricing Models and Its Application in Python

Microservices

- Handle Timeouts Between Microservices with Istio

Microsoft Office

- Pricing Knock-Out Options
- Understanding Word Embeddings Using Python

Monitoring

- Get Started with Prometheus
- Graph Docker Metrics with Prometheus

Natural language processing

- NLP with Python: Introduction to NLTK
- Regular Expressions: Repeating Regex Patterns with Quantifiers
- NLP with Python: Text Classification
- Regular Expressions: Literals and Ranges
- NLP with Python: Introduction to Word Embeddings
- Regular Expressions: Wildcards
- NLP with Python: Text Cleaning
- NLP with Python: Introduction to Bag of Words
- Regular Expressions: Anchoring
- Regular Expressions: Groups and Alternators
- NLP with Python: Introduction to n-Grams

Node.js

- Node.js Sandbox

NoSQL Databases

- Cassandra: Run Cassandra in Docker
- Cassandra: Build a Microservice with Apache Cassandra and Apache Kafka
- Cassandra: Get to Know CQL
- MongoDB: Getting Started with CRUD Operations
- Cassandra: Design Tables
- Cassandra: Analyzing Data with the DataStax Spark-Cassandra Connector
- Cassandra: Push Data to Apache Cassandra with the DataStax Apache Kafka Connector
- Cassandra: Install Cassandra
- MongoDB: Creating Your First Full-Text Search Query
- MongoDB: Getting Started with Grouping
- Cassandra: Use Data Models



NoSQL Databases (cont.)

- MongoDB: First Steps with Transactions
- Cassandra: Types in CQL
- MongoDB: Advanced ETL with Projections
- MongoDB: Advanced Operations with Full-Text Search
- Cassandra: Query Options in CQL
- MongoDB: Finding Nearest Restaurants with Geospatial Queries
- MongoDB: Handling Arrays with \$unwind
- Cassandra: Extend Data Models
- Cassandra: Getting Started with the DataStax Java Driver
- Cassandra: Simple Statements with the DataStax Java Driver
- Cassandra: Prepared Statements and the DataStax Java Driver
- Cassandra: Lightweight Transactions with the DataStax Java Driver
- Cassandra: Object Mapping with the DataStax Java Driver
- Cassandra: Batch Statements with the DataStax Java Driver

Python

- Python Sandbox
- Python Cookbook: Classes & objects
- Python Cookbook: Strings & Text
- Python Cookbook: Data Structures & Algorithms
- Python Cookbook: Functions
- Python ML Cookbook: Wrangling Data (Part 1)
- Python Cookbook: Numbers, Dates & Times
- Python IO: TextIO
- Python Cookbook: Metaprogramming
- Python Cookbook: Data encoding & processing
- Python Cookbook: Iterators & generators
- Cleaner Code for Data Science: Logging in Python
- Threading in Python: Create a Single-Threaded Program
- Threading in Python: Create a Multi-Threaded Program Using a Thread Pool

- Threading in Python: Create a Multi-Threaded Program Using Thread Objects
- Cleaner Code for Data Science: Using Auto Documentation Frameworks in Python
- Python Core Beam: ParDo
- Time Value of Money and Python Implementation
- Functional Python: Implementing Immutable Data Structures
- Simulation Analysis and Its Application in Python
- Python Core Beam: GroupByKey
- PCA and SVD from Scratch Using Python
- Python Core Beam: ParDo OneToMany

Relational databases

- MySQL Sandbox
- PostgreSQL Replication: Getting Started
- PostgreSQL Replication: WAL
- PostgreSQL Replication: Physical Streaming Replication
- PostgreSQL Replication: Logical Replication
- PostgreSQL Replication: Logical Decoding

Ruby

- Ruby Sandbox

Selenium

- Using gazpacho and Selenium to Retrieve the Contents of a Password-Protected Web Page

Serverless architecture

- Kubernetes Serverless: Knative

Single board computers

- Estimate Pi Using Monte Carlo

Software estimation

- Understanding Estimates and Error Bands



SQL

- SQL: Filter Text, Dates, and Times with WHERE
- SQL: Retrieve Data with SELECT
- SQL: Joining Multiple Tables
- SQL: Joining Tables with INNER JOIN
- SQL: Joining Tables with LEFT JOIN
- Presto First Steps: Writing and Running SQL Queries
- First Steps with Apache Airflow: Querying the Airflow Database
- SQL: Aggregating and Filtering Data
- Presto First Steps: Installation to First Queries
- SQL: Conditionally Aggregating Data Using the Zero/NULL Case Trick
- SQL: Counting, Grouping, and Ordering Data
- SQL: Filter Booleans, Nulls, and Precedence with WHERE
- SQL: Appending and Conditionally Transforming Data
- SQL: Filter Numbers and Logic with WHERE
- SQL Ranges: Locate a Range
- SQL Ranges: Generate a Series of Values
- SQL Ranges: Fill In Missing Values
- SQL Ranges: Find the Start and End of a Range

Statistics

- Probability from Scratch: The Monty Hall Problem: Monte Carlo Simulation
- Probability from Scratch: Probability Logic
- Probability from Scratch: Conditional Probability and Bayes Theorem
- Probability from Scratch: Discovering Binomial and Beta Distribution
- Probability from Scratch: Discovering the Normal Distribution
- Necessary Statistics in Finance

Storage repositories

- Cloud Native Data: Data Store per Service

Streaming and messaging

- Java Streams: Using the Java Streams API
- Kafka Streams: Installing Kafka
- Getting Started with Queuing Using Apache Pulsar
- Java Streams: Mapping, Reducing, and Collecting Rich Data
- Kafka Streams: Installing Kafka Streams
- Getting Started with Pub-Sub Messaging Using Apache Pulsar
- First Steps with Apache Airflow: Running Your DAG
- Kafka: Schema Evolution
- Pulsar: Produce and Consume Events to Pulsar Topics Using Java Client
- Pulsar: Submit a Connector to Consume Events from Pulsar
- Kafka Streams: Create a Custom News Feed
- Kafka Streams: Publish News Feeds to Kafka
- Java Streams: Creating Streams and Building Generators
- Kafka: Schema Registry
- Pulsar: Produce and Consume Events to Pulsar Topics Using AVRO and JSON Schema
- Pulsar: Submit a Connector to Write Events to Pulsar
- Pulsar: Develop a Pulsar Connector
- Pulsar: Write a Pulsar Function to Process Events
- Kafka: Configuration for Beginners

Swift

- Swift Sandbox

Technical writing

- First Steps with Apache Airflow: Writing a DAG

Time series

- Simulating Entire Time Series of GBM Paths



Trading stocks

- Quantitative Trading with Python: Download Data and Set Up a Trading Strategy
- Quantitative Trading with Python: Loading Data Using Python
- Quantitative Trading with Python: Initialize Backtest Using the Backtrader Framework
- Quantitative Trading with Python: Developing a Trading Strategy
- Quantitative Trading with Python: Moving Averages and Volatility
- Quantitative Trading with Python: Methods of Calculating Returns
- Quantitative Trading with Python: Run the Backtest and Plotting Results
- Quantitative Trading with Python: Calculating Bollinger Bands

Unix

- bash Cookbook: Beginning bash
- bash Cookbook: Standard Output
- Bash Sandbox

Web API's

- Financial Data Sources and APIs in Python

Web performance

- Test Network Performance with iPerf

Web scraping

- Using gazpacho to Download and Parse the Contents of a Website

Web servers

- Deploy an NGINX Container Using Terraform

WebAssembly

- WebAssembly with C/C++: Calculating the Digits of Pi with WebAssembly
- WebAssembly with C/C++: Compiling a Fractals Generator to WebAssembly
- WebAssembly with C/C++: Compressing Files in the Browser with WebAssembly