MicroServices Engineering Boot Camp

Course ID #: 7000-652-ZZ-Z
Hours: 24

Course Content

Course Description:
For IT professionals, developers, software engineers, and DevOps practitioners – microservices training provides the technical practices and tooling fundamentals necessary to begin realizing the benefits of microservices as a foundation for IT architecture, software engineering, and service/release delivery. The workshop includes 16 hands-on exercises that give you real-world practice on the engineering tools and skills a team needs to realistically implement your own flavor of Microservices architecture patterns so you can address the team needs of your own organization.

Whether you want to create new services, decouple a few services from your overall architecture, or refactor an entire monolithic architecture into a microservice design pattern, this course quickly teaches you the practical toolset and skills to get up and running with microservices in your own systems. Loosely coupled components and services allow teams to deploy more freely and independently, with less risk to the architecture.

At Course Completion:
Students will be able to:
- Professionals who may benefit include:
- System and software architects
- Developers
- Testers and QA teams
- Release engineers
- IT operations staff
- Site reliability engineers
- DevOps practitioners
- DBAs and data engineering teams
- Information Security Pros

Prerequisites:
This is a lab-intensive microservices training course. Professionals who take this course should have some familiarity with Docker, Kubernetes, and AWS before attending.

MicroServices Engineering Boot Camp

Course ID #: 7000-652-ZZ-Z

Hours: 24

Target Student:

Professionals who may benefit include:

- System and software architects
- Developers
- Testers and QA teams
- Release engineers
- IT operations staff
- Site reliability engineers
- DevOps practitioners
- DBAs and data engineering teams
- Information Security Pros

Topics:

Part 1: Intro to Microservices

1. Optimize for speed, not efficiency
2. Case Study: General Electric
   - Throughput
   - Waste
3. **Amazon Web Services Case Study (SOA/Microservices)**
   - Problem: Scaling the Organization and the 'Big ball of mud'
   - Conway's Law
   - Service Oriented Architecture
   - Forced Self Service Mandate
   - Result: Amazon dominance of cloud
   - Result: High velocity at scale
4. **Intro to Containers (encapsulation)**
   - What is Docker
   - Exercise: Install Docker
   - Exercise: Docker Hello World
   - Docker ecosystem
   - Docker concepts
   - Container encapsulation/ideal use cases
     - Encapsulation
     - Speed
     - Increased utilization of computing resources
   - Benefits
     - Configure once, run everywhere
   - VM's vs Container use cases
     - Databases & stateless workloads
   - Docker Architecture
   - Exercise: Docker 101 – Web App
   - Docker File System
   - Docker Images
   - Exercise: Stateless Web App
   - Local Registry
   - Data Volumes
   - Exercise: Docker 201 – Compose Multi-tier app
   - Continuous integration patterns
   - Docker Security
   - Continuous Integration
     - Canary Release
     - Blue Green Deployment
     - A/B Testing
     - Rolling Update
     - Jenkins Plugin

5. **Microservice challenge: Continuous Integration Service**
    - On-Premise
        - Jenkins
    - SaaS Service
        - Shippable
        - Jenkins
        - TravisCI
    - Exercise: Trigger build/tests from change

Part 2: Microservices in Development

1. **Uber Case Study**
    - 2000 services, 1000 engineers
    - Tradeoffs
        - Plus – overall development speed
        - Cons – technical challenges
2. **Box Case Study**
    - Traditional service deployment with bare metal
    - 10x faster workflow with DevOps practices
3. **Microservice challenge: Image repository**
    - Docker repository development instance
    - On-Premise Service
        - Quay by CoreOS
    - SaaS solution
        - Docker Hub
        - JFrog
    - Exercise: Submit image to service
    - Exercise: Pull image from service
4. **Intro to Kubernetes (Containers at Google)**
    - Prerequisites
    - Containers
    - Linux Kernel Features
    - Container User Experience
    - New Container Capabilities
    - Gaps using Containers in Production
5. **Exercise: Kubernetes 100: Hello World**
6. **Core Concepts**
    - Cluster Orchestration
    - Originated at Google
    - Open Source
    - Benefits
    - Design Principles
7. **Architecture**
    - Master/Node
    - Kubectl
    - Replication Controller
    - Kubelet
    - Kube-Proxy
    - Persistent Volumes
    - Etcd
    - High Availability
    - Exercise: Kubernetes 101: Stateless web app
8. **Kubernetes Features**
    - Pods
    - Labels
    - Services
    - Namespaces
    - Resource Quota
9. **Exercise: Kubernetes 201: Guestbook app**

Part 3: Microservices in Production

1. **Spotify Case Study**
    - 810 Services, 477 engineers
2. **Microservice challenge: Service discovery**
    - Skydns
    - Consul
3. **Exercise: Resolve service with DNS**
4. **Security**
    - Goals
    - Roles
    - Attribute Based Access Control
    - Policies

MicroServices Engineering Boot Camp

Course ID #: 7000-652-ZZ-Z
Hours: 24

- Service Accounts
- Secrets

5. **Forth Microservice challenge: Secrets**
   - Vault
   - Kubernetes Secrets API
6. **Exercise: Kubernetes – Store database credentials in cluster**
7. **Cluster Add-ons**
   - Cluster DNS
   - Logging with Elasticsearch and Fluentd
   - Container Level Monitoring
   - cAdvisor
   - InfluxDB
   - Prometheus
8. **Exercise: WordPress on Kubernetes**
9. **Managing state with disposable architectures**
   - Tradeoffs, standalone vs containerized databases
   - CAP Theorem
   - SQL Databases
   - NOSQL Databases
10. **Exercise: Cassandra on Kubernetes**
11. **Practicing Failure**
    - Optimize MTTR
12. **Netflix Case Study**
    - Simian Army
    - Graceful handling of failure

Part 4: Putting it all together

1. **Why Microservices?**
   - Scale an organization
   - Tradeoffs
   - Fault Tolerance
   - Throughput
   - Waste
2. **Kubernetes Alpha Features**
   - Multi-Datacenter Control Plane
   - RBAC/Multi-tenancy
3. **Openshift/Mesos/Other PaaS platforms**
4. **Exercise: Customize Microservice App**
5. **Exercise: Scale app for simulated demand**
6. **Review of Microservice Challenges**
   - Secure Images
   - Highly available application
   - Secrets
   - Continuous Integration
   - DNS Name resolution
7. **Summary**