MSEBC
Microservices Engineering Boot Camp

Duration: 3 Days

Course Overview:

Learn a Practical Approach to Microservices and Get Hands-on Practice With the Real-world Toolchain.

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. This course includes 16 hands-on exercises that give you real-world practice on the engineering tools and skills a team needs in order 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.

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.

Class exercises use Java, but for private corporate training deliveries, exercises can be done using your preferred language.

Course Objectives:

Upon completing this course, the learner will be able to meet these overall objectives:

- Adopt, plan, or improve your transition to microservices
- Map technical practices to the business strategy behind microservices
- Navigate different tools for enabling microservices and how to use them
- Communicate with stakeholders, management, and teams regarding needs and expectations around microservices
- Build microservices with Docker, Kubernetes, Jenkins, and JFrog
- Build more mature DevOps practices through microservice adoption
- Refactor monolithic systems into more modular, component-based systems
- Apply microservice use cases to continuous integration, delivery, and testing
- Enable more automated testing and self-service QA capability

Who Should Attend:

The primary audience for this course is as follows:

- 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

Course Outline:

Part 1: Intro to Microservices

Optimize for speed, not efficiency

Case Study: General Electric

- Throughput
- Waste

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

Intro to Containers (Encapsulation)

- What is Docker
- 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 and stateless workloads
- Docker Architecture
- Docker File System
- Docker Images
- Exercise: Stateless Web App
- Local Registry
- Data Volumes
- Continuous integration patterns
- Docker Security

Course Outline (Cont.):

- Continuous Integration
  - Canary Release
  - Blue Green Deployment
  - A/B Testing
  - Rolling Update
  - Jenkins Plugin

Microservice Challenge: Continuous Integration Service
- On-Premise
  - Jenkins
- SaaS Service
  - Shippable
  - Jenkins
  - TravisCI

Part 2: Microservices in Development

Uber Case Study
- 2000 services, 1000 engineers
- Tradeoffs
  - Plus – overall development speed
  - Cons – technical challenges

Box Case Study
- Traditional service deployment with bare metal
- 10x faster workflow with DevOps practices

Microservice Challenge: Image Repository
- Docker repository development instance
- On-Premise Service
  - Quay by CoreOS
- SaaS solution
  - Docker Hub
  - JFrog

Intro to Kubernetes (Containers at Google)
- Prerequisites
- Containers
- Linux Kernel Features
- Container User Experience
- New Container Capabilities
- Gaps using Containers in Production

Core Concepts
- Cluster Orchestration
- Originated at Google
- Open Source
- Benefits
- Design Principles

Architecture
- Master/Node
- Kubectl
- Replication Controller
- Kubelet
- Kube-Proxy
- Persistent Volumes
- Etcd
- High Availability

Kubernetes Features
- Pods
- Labels
- Services
- Namespaces
- Resource Quota

Part 3: Microservices in Production

Spotify Case Study
- 810 Services, 477 Engineers

Microservice Challenge: Service Discovery
- Skydns
- Consul

Security
- Goals
- Roles
- Attribute Based Access Control
- Policies
- Service Accounts
- Secrets

Forth Microservice Challenge: Secrets
- Vault
- Kubernetes Secrets API

Course Outline (Cont.):

Cluster Add-ons
- Cluster DNS
- Logging with Elasticsearch and Fluentd
- Container Level Monitoring
- cAdvisor
- InfluxDB
- Prometheus

Managing state with disposable architectures
- Tradeoffs, Standalone vs Containerized Databases
- CAP Theorem
- SQL Databases
- NOSQL Databases

Practicing Failure
- Optimize MTTR

Netflix Case Study
- Simian Army
- Graceful Handling of Failure

Part 4: Putting it All Together

Why Microservices?
- Scale an Organization
- Tradeoffs
- Fault Tolerance
- Throughput
- Waste

Kubernetes Alpha Features
- Multi-Datacenter Control Plane
- RBAC/Multi-tenancy

Openshift/Mesos/Other PaaS platforms

Review of Microservice Challenges
- Secure Images
- Highly Available Application
- Secrets
- Continuous Integration
- DNS Name Resolution

Exercise Outline:

Exercises are designed to assure learners a whole practical experience, through the following practical activities:
- Exercise: Install Docker
- Exercise: Docker Hello World
- Exercise: Docker 101 – Web App
- Exercise: Docker 201 – Compose Multi-tier App
- Exercise: Trigger Build/Tests from Change
- Exercise: Submit Image to Service
- Exercise: Pull Image from Service
- Exercise: Kubernetes 100: Hello World
- Exercise: Kubernetes 101: Stateless Web App
- Exercise: Kubernetes 201: Guestbook App
- Exercise: Resolve Service with DNS
- Exercise: Kubernetes – Store Database Credentials in Cluster
- Exercise: WordPress on Kubernetes
- Exercise: Cassandra on Kubernetes
- Exercise: Customize Microservice App
- Exercise: Scale App for Simulated Demand