INDEX

Contents

ABOUT DOCKER

This open-source containerization platform is based on Linux that is useful for programmers to design, develop and run the package applications for deployment using containers. Basically, it divides the application's performance into various components which able to process independently when needed. Docker containers offer efficient development and test, quick execution, interoperability, and more. With the docker environment, the applications/services can run with individual host access.

ABOUT KUBERNETES

Kubernetes is the open-source orchestrator which is used to create, manage and deploy containerized applications (microservices). It supports container APIs with quick shipping and scaling with decoupled architecture. It acts as a self-service platform that creates a hardware abstraction layer for the development team. Thereby, they can quickly access the resources as well as can handle the additional load.

ABOUT MICROSERVICES

Microservice architecture divides the whole application into a different pool of loosely coupled services that are similar to Service-Oriented Architecture (SOA). So that it should perform continuous delivery even for a large and complex application. With microservices, you can develop flexible, scalable, and reliable independent applications. It's clear and separate boundaries make its features as highlighted and able to provide a continuous supply model.

WHY ONE SHOULD LEARN MICROSERVICE, DOCKER, AND KUBERNETES?

We have noticed that technologies today have evolved a lot to suit the requirements of today's needs of customers and projects. So that monolithic architectures are replaced with microservices and docker/Kubernetes open-source platforms.

If you are a hardcore developer in Microsoft stack and not using the latest features of the technology especially .NET Core very soon you will find yourself outdated. Building a Microservices application using docker/Kubernetes will give you an opportunity to touch base with all the advanced features of the language and expand your scope.

MICROSERVICE, DOCKER, AND KUBERNETES VIRTUAL TRAINING BY DECCANSOFT

The goal of this LIVE Training is to deliver high-quality training that covers solid fundamental knowledge on core concepts with a practical approach. Attend this LIVE training and explore your cloud-native career on microservices with Docker/Kubernetes.

A fresh LIVE batch starts from November 14, 2021, from 5.00 PM-8.00 PM (IST). Attend this 8 Weekend training.

Enroll now for the live batch and get an interactive and immersive learning experience.

OUR TRAINING METHODOLOGY

In this course, we will start with the basics of microservice applications using Docker and Kubernetes containerization. The same is reflected in our syllabus in the form of an eStore Application at the end of each module.

Then we will start progressively to develop docker applications using .NET core. This will give the experience of working with a live project in real-time development scenarios. Additionally, you will be trained under Docker volume, compose methodology, Port forwarding, Docker registry, and more. The same is reflected in our syllabus for your detailed reference.

Our **star trainer Mr. Sandeep Soni** having 25+ years of experience in the IT industry has compiled this course and he will be himself delivering it.

The teaching methodology of Mr. Sandeep Soni is very **simple but comprehensive**, every topic in the docker/Kubernetes begins with in-depth concepts. You will get a comfortable state about what is being talked regards the course, followed by **practical demos.** All the topics and their features should be incorporated in real-time situations.

WHO CAN DO THIS COURSE?

1. Any MS.NET developer with experience in web programming /basic Linux system administration
2. Architects looking forward to building Docker/Kubernetes container-based applications using microservices.

WHAT WILL YOU GET IN THIS LIVE TRAINING? (SPECIAL OFFER for this batch only!)

- Exhaustive training by Microsoft Certified Trainer, **Mr. Sandeep Soni** having **25+ years** of experience.
- You will get video access to the recorded sessions of the live training.
- You will get in-detailed and **Complete Courseware** prepared by Mr. Sandeep Soni himself and the same can be used for practice and reference.
- You can attend **Microservice/Docker/Kubernetes Full-day Boot camps** in the future.

DOCKER SYLLABUS

Introduction to Containerization and Docker

- Understanding VM's and Containers.
- Benefits of Containerization
- What is Docker?
- Docker Benefits.
- Docker Architecture
- Docker Taxonomy.

Working with Images and Container

- Installing Docker for Windows / Mac Desktop
- Installing Docker on Linux VM
- Pulling Images
- Starting and Stopping Containers
- Running a Container in interactive mode
- Creating Image from current state of Container
- Important Docker Commands

Developing Custom Images

- Understanding Base Images.
- Dockerfile and Building Docker Images
- Breaking down and understanding Dockerfile
- Executing custom applications as Containers.
- Inspecting the Image Architecture
- Multiple stages in Dockerfile
- Benefits of Multi-Stage builds
- Creating efficient, small images
    - Alpine Linux, Ubuntu 18.04, Debian slim, and other distributions.

Docker Volume

- Purpose of using Volumes
- Access Data in Docker Containers
- Use Cases for Volumes
- Docker volume commands
- Creating Container with Volumes
- Sharing Volumes
- Managing state inside containers

Docker Compose

- Overview
- Docker compose features
- Building docker-compose.yml file
- Docker-compose command
- Working with multiple images in a single application
- Environment Variables and Configuration File

Networking & Port Forwarding

- Introduction to Container Networking
- Exposing Containers with Port Redirect

Docker Registry

- Creating a Docker Hub Account
- Pushing an Image to Docker Hub
- Pulling the Image from Docker Hub
- Create Local / Private Registry
- Pushing and Pulling Image from Registry
- Docker Content Trust

KUBERNETES SYLLABUS

Introduction to Kubernetes and its Architecture

- What is Kubernetes
- Why Kubernetes
- Kubernetes features
- Kubernetes Architecture
- Kubernetes Cluster
- Kubernetes Master
  - API Server
  - Etcd
  - Scheduler
  - Controller Manager
- Worker nodes

- Container Runtime
  - Kubelet
  - Kube-proxy
  - cAdvisor
- Kubernetes Objects Overview
  - Kubernetes Pods
  - Replication Controllers and Replication sets
  - Deployments
  - Services
  - Volumes and Persistent Volumes
  - Stateful Sets
  - Daemon Sets
  - Jobs and Cron Jobs

Installing Kubernetes Installation

- Docker for Desktop
- Minikube
- AKS/EKS

Working with Pods and Kubectl Commands

- Create out first pod with kubectl
- Basic Kubectl Commands
- Inspecting Kubernetes Objects using kubectl
- About Kubernetes Generators
- Imperative vs Declarative Commands
- Exploring YAML Syntax
  - Name and Metadata
  - Labels and Label Selectors
- Kubernetes Namespace
- Kubernetes Generators
- Working with Dashboard

Working with Kubernetes Objects

- Pods
- Replication Controller

- Replica Sets
- Creating Deployment
- Self Healing Applications.
- Handling Rolling Updates and Rollbacks
- Best practices in rolling upgrades, canary deploys, blue-green deploys etc

Services and Ingress

- Service Types
    - Creating a ClusterIP Service
    - Creating a NodePort
    - LoadBalancer Service
- Working with Ingress
- Ingress Controllers
- Ingress Annotations and rewrite target
- Kubernetes Services DNS
- Network Policies
- Working with Probes

Advanced Kubernetes Objects

- configMap and Environment Variables
- Working with Secrets and sensitive information
- Kubernetes Volumes
- Persistent Volumes and Persitent Volume Claims
- StatefulSet
- DaemonSets
- Jobs
- Scheduling using Cron Jobs
- Monitoring and Probes

Azure Kubernetes Service

- About AKS
- Creating AKS Cluster using Portal

- Creating AKS Cluster using CLI
- Connecting to AKS Cluster using Kubectl
- Deploying Kubernetes Objects to AKS
- Pulling Images from ACR and DockerHub
- System Nodes vs User Nodes
- AKS Networking: Azure CNI vs Kubenet
- AKS Logging and Monitoring

Cluster Administration

- Kubernetes Dashboard
- Manage Memory, CPU and API resources.
- Authentication and Authorization
- Using RBAC Authorization
- Using ABAC Authorization

MICROSERVICE SYLLABUS

Containerization of Applications using Docker

Introduction to Containers and Docker

- Understanding VM's and Containers.
- What is Docker?
- Docker Benefits.
- Docker Architecture and Docker Taxonomy.

Developing Docker Custom Images using .NET Core

- Docker Images for .NET Core.
- Executing .NET Core applications in Containers.
- Inspecting the Image Architecture
- Developing and Publishing .NET Core Applications
- Dockerfile and Building Docker Images
- Breaking down and understanding dockerfile in-depth.

Multi-Stage Builds

- Multiple stages in Docker file
- Hosting ASP.NET Applications in Docker

State and Data in Docker Applications

- Purpose of using Data Volumes
- Access Data in Docker Containers
- Creating Container with Volumes
- Data volume containers

Docker Compose

- Overview
- Docker compose features
- Building docker-compose.yml file
- Docker-compose command
- Working with multiple images in a single application
- Environment Variables and Configuration File

Microservices Architecture – Get perfect in concepts and practical

Understanding Microservices

- Understanding Monolithic Architecture
  - What are Monolithic Applications
  - Deploying
  - Containerizing using Docker
  - Scaling Applications
  - Managing State and Data
  - Benefits and Drawbacks of Monolithic Architecture
- Microservices Architecture
  - What are Microservices
  - Monolithic vs Microservices Architecture
  - Characteristics of Microservices Architecture
  - Benefits of using Microservices Architecture
  - Microservices Design Principles.

- SOA vs. Microservice
- Handling Data in Microservices
- Communication between Microservices
    - Synchronous Communication across Microservices.
    - Asynchronous communication across Microservices.
- API Gateway Pattern
- Microservices Patterns
    - Domain Driven Design
    - Command and Query Responsibility Separation (CQRS)
    - Event Sourcing
- Creating Composite UI with Microservices
- Drawbacks of Microservices

Setup Microservice Based Application and Perform CRUD Operations

- Creating a Solution and Project Layout
- Implementing a CRUD microservice
- Writing Domain Classes and Controllers
- Data Context Class and Data Seeding
- Using Repository Classes
- Swagger and SwashBuckle Integration
- Practical Demonstration using eStoreApplication
    - Product CatalogService with SQL Server
    - Invoking both services using Swagger UI
    - Implementing Layered Architecture
    - Generic Repository Pattern

Build UI Service

- Adding ASP.NET MVC Project
- Writing Model Classes
- Writing Service Classes
- Building Web Controller and Views

- Practical Demonstration using eStoreApplication
  a. Writing Backend for Frontend (BFF)
  b. UI Microservice to display Product Catalog

Hosting Microservices using Docker Containers

- Adding Docker Support to the Microservice Application
- Creating a Dockerfile file
- Designing and Developing Multi-Container Microservices
- Database Connection string and environment variables in Docker containers
- Handling Configuration Data
- Use a database server running as a container
- Practical Demonstration using eStoreApplication
  - Creating an SQL Server database container
  - Create a docker images for Product Catalog Microservices
  - Create a docker image for UI Microservice
  - Writing a YAML for deploying and executing the application.

Using Redis Cache in Microservices

- Understanding Redis Cache importance
- Programming Microservice to use Redis Cache
- Consuming Microservice in Web Client
- Practical Demonstration using eStoreApplication
  - Building Shopping Cart Microservice
  - Persisting Cart data in Redis Cache
  - Create a docker image for Shopping Cart Microservice

Understanding OAuth2 and OpenIdConnect

- Authentication and Authorization
- Introduction to Basic Authentication Workflow
- Understanding OAuth
- OAuth Grant Types
- Understanding OpenIDConnect

- Securing Services and Middleware in ASP.NET Core
- Using JWT Token to Authenticate and Authorize

IdentityServer4 Server

- What is IdentityServer4
- Authorization Grant Types
  - Authorization Code
  - Implicit
  - Client Credentials
  - Resource Owner Password Credentials (ROPC)
- Including UserInfo in ID Token and Access Token

Implementing Security for Microservices

- The Big Picture
- Implementing Identity Microservice
- Using Client Credential Token to Access Microservice
- Using Access Token to call Microservices
- Authentication between Microservices
- Implementing Role based and Policy based Authorization
- Practical Demonstration using eStoreApplication
  - Adding Authentication Microservice to Solution
  - Securing ProductCatalog Microservice
  - Accessing secure ProductCatalog in Swagger UI
  - Accessing secure ProductCatalog in Client Application

API Gateway Integration

- Introduction to API Gateway
- Understanding Ocelot Middleware
- Integrating API Gateway for Routing
- Handling Secure Microservices in API Gateway
- Practical Demonstration using eStoreApplication
  - Adding API Gateway Service using Ocelot Middleware

  - ○ Updating ocelot configuration for routing to ProductCatalog and ShoppingCart Microservices
  - ○ Deploying Gateway Service in Docker Container.

Microservices Communication

- Synchronous Communication using REST API
- Asynchronous Communication using Service Bus
- Integration Events and Event Handlers
- Handing Atomicity and Resiliency when Publishing to EventBus
- Practical Demonstration using eStoreApplication
  - Adding Shopping Cart Service with Redis Cache
  - Adding Product to Shopping Cart Service for logged-in client
  - Create a docker images for Product Catalog Microservices
  - Asynchronous Communicating with Notification Service and Service Bus Queue

Implementing CQRS Pattern

- Overview of CRQS Pattern
- Understanding Command Pattern
- Domain Model and Read Model
- Comparing CQRS with traditional CRUD approach
- Apply CQRS and CQS approach in DDD microservice
- Practical Demonstration using eStoreApplication
  a. Build a New ProductCatalog Service Demo
  b. Update ShoppingCart if Product Price changes.
  c. Update ProductCatalog inventory if order is placed.

Domain Driven Pattern

- Overview of Domain and Domain Driven Design
- Layered Architecture in DDD Microservices
- Implementing the Command and Command Handler Pattern
- About Domain Events

- Command and Command Handler Classes
- Practical Implementation of DDD Pattern
- Practical Demonstration using eStoreApplication
    a. Implementing DDD using Order Microservices

Handling Failures

- Handle Partial failure
- Implement retries and exponential backoff
- Using Polly policies
- Circuit Breaker Pattern
- Practical Demonstration using eStoreApplication
    a. Handle Order Service failure
    b. Handling temporary downtime of SQL Database.

Microservices Deployment in App Services

Microservices and Kubernetes

- What is Monolithic Application
- What are Microservices
- Kubernetes Architecture
- Azure Kubernetes Service (AKS)
- Practical Demonstration using eStoreApplication
    a. Deploying the above service in Azure Kubernetes Service.

WE PROMISE 100% MONEY BACK IF THE STUDENT ATTENDS COMPLETE COURSE AND IS NOT SATISFIED WITH THE QUALITY OF TRAINING

https://www.bestdotnettraining.com/microservices-online-training
https://www.bestdotnettraining.com/docker-online-training
https://www.bestdotnettraining.com/kubernetes-online-training