

Technical Overview on MicroServices platforms and Environments

Course Objectives

This course provides technical overview on MicroService applications, environments and tools used in deployment of MicroServices to Auditing Team.

Training Methodology

The theoretical topics are discussed interactively and technical overviews are discussed with practical demonstrations to the participants.

The training material including documents, presentations shared with the participants in soft format.

Software Installations

No specific software installations needed.

Training duration: 1.5 days

Course plan

Day1 (8 hrs)

Overview on MicroService Applications

- Desktop, Mobile and web applications
- Limitations of web applications
- Service Oriented Architecture for application to application data sharing
- Overview on Soap(web services) and REST services
- Data exchange with services
- Monolithic and n-tiered applications and challenges
- Need of decomposition at application level
- MicroService Architecture
 - Independent services
 - Loosely coupled service applications
 - performance
- Breaking the monolith into MicroServices
- MicroService development platforms overview
- Application Security for MicroService applications
 - Login with userid and password
 - Secured (SSL) encrypted communication
 - Token authentication/Single sign in/Third party authentication
- MicroService pros and Cons

MicroServices in Deployment

- Web server for running the MicroService applications.

- Load balancer
- Deployment nodes
- Server farm and cloud environment
- Fault tolerant services with redundant nodes/services
- Circuit Breaker
- Expose Service API Documentation
- Service Discovery
- Gateway and External Configuration Service
- Distributed Logging
- Service aggregation and split services
- Message Broker and Asynchronous process
- Transaction management
- Netflix tools overview

Moving MicroServices to Docker Containers

- Virtual Machine vs. Containers
- Docker Container Engine architecture
- Docker installation requirements
- Docker daemon server and Docker client
- Application container and image
- Docker-hub the image registry for sharing the images
- Docker usage overview
 - Create and run the Docker containers
 - Inspect docker process and containers
 - Start, Stop and remove containers
 - Interactive and detached containers
 - Manage the docker images
 - Docker container Logging and monitoring
 - Commit the local modified container as image
 - Define your own image with Dockerfile
 - Build and share the customized image
 - Copy files/directories from container to host and vice versa
 - Clean the docker environment
- Define customized docker container image for MicroService application
- Build and share the MicroService application image
- MicroService application deployment in container
- Configure and create the database container
- Deploy the MicroService application with link to database container
- Docker container state management
- Sharing the data across containers with volume
- Make the Volumes data persistent across container restarts

- Mount volumes to containers from host system
- Docker-compose for container configuration with YML files
- More docker usage
 - Usage of local Docker Registry for private sharing
 - Monitoring services
 - Host network mapping
 - Host Port Mapping
 - Dynamic port mapping
 - Default bridge network

Docker Container Orchestration requirements

- Making the MicroService application Highly available
 1. Configure the cluster
 2. Load balancing and instance management
 3. Run the application across multiple host nodes
 4. Application updates (rolling updates) and roll back
 5. Scaling the containers
 6. Monitor the containers
 7. Inspect the services

Overview on Kubernetes for Container orchestration

- Kubernetes architecture
- Kubernetes setup requirements
- Kubernetes on top of Docker container engine
- Kubernetes Core Components
- Kubectl and kubeadm tools
- Overview on the basic cluster with kubeadm
- Basic objects of deployment
 - Pod
 - Deployment
 - Service
 - Volume
 - Namespace
 - Job
- The configuration structure in YML files
- Kubernetes cluster management with kubectl
 1. Create and manage pods
 2. Run and monitor the pods with logs
 3. Inspect the pods
 4. Interact with pods
 5. Manage the data sharing with Persistent Volume
 6. Create deployments for MicroServices in cluster
 7. Pods with multiple containers: inspect, interact and logs

8. Create and manage deployments and services
9. Expose the services to external world
10. Manage replications
11. Load balancing internal and external
12. Inspect service logging
13. Monitor and manage the cluster with Dashboard application
14. Collect and analyse the logs with ELK

Kubernetes in Cloud

- Cloud overview
- Infrastructure-as-a-Service (IaaS): Use the available hardware with limited software platforms to compute, network, and storage on-demand, over the internet and on a pay-as-you-go basis.
- Platform-as-a-Service (PaaS) : Use the complete platform—hardware, software, and infrastructure: Kubernetes as Service
- Expose the services to external world

Service Mesh Usage overview

- Single shop for deployment and sharing of services
- Reduce the downtime
- Optimise the performance
- MicroService applications in Istio mesh.

Day2 (4hrs)

Continuous Integration and Continuous Delivery with Jenkins

- Philosophy of Continuous Integration
- Jenkins CI server overview
- Components of CI Process
 - Code repository: Git, SVN etc.
 - Build automation tools Apache Ant, Maven etc.
 - Build job in Jenkins
 - Application Deployment
 - Testing tools
 - Code quality analyzers: Sonar, FindBugs etc.
 - Update to team members via email/sms etc.
 - Monitoring of build activities
- Jenkins integration with docker and Kubernetes

Stackdriver on GCP: Overview

- Application logging
- Sharing performance and diagnostics data for usage
- Collect metrics, logs, and traces across Google Cloud
- dashboards and views to monitor the platform and applications
- Query and analyze
- Set up alerts and notification rules

Overview on DevOps and Agile Practices

- Impact of Agile practices
 - Test Driven Development
 - Continuous Testing
 - Continuous Integration
 - Continuous and iterative delivery
- What is DevOps?
- Phases of DevOps
 - Development
 - Configuration management
 - Continuous Integration and delivery
 - Integration Testing
 - Operation and Monitoring management
 - Feedback loop
- Impact of DevOps process

DevSecOps vs. DevOps

- Similarities and focus on Security
- Apply security standards from the beginning
- Continuous feedback loop
- Automated Security

Question-Answer session
