Ref. No. : EEA/TTD-DC/IT/2024 **Date : 10/11/2024**

IT Professional Corporate Training Curriculum for IT/Non-IT Participants

Name of the Corporate Training :

Docker: Containerization and Deployment

Course Duration: 40 Hours
Course Code: TTD-DC
Target Audience: Developers, IT professionals, and DevOps engineers interested in containerization for development and deployment.
Prerequisites: Familiarity with command-line tools, Linux basics, and basic knowledge of application development.

Course Outline

Module 1: Introduction to Docker and Containerization

- **Topics Covered:**
  - Introduction to containerization: what it is and why it matters
  - Understanding Docker vs. traditional virtualization
  - Key components: Docker Engine, images, containers, registries
  - Installing Docker and setting up a development environment
- **Milestone 1 Project:**
  - Install Docker, set up a basic environment, and run the “hello-world” container to verify the setup.

Module 2: Working with Docker Images and Containers

- **Topics Covered:**
  - Understanding Docker images and containers
  - Basic Docker commands: *docker run*, *docker ps*, *docker stop*, *docker rm*
  - Pulling and running images from Docker Hub
  - Managing containers and understanding container lifecycles
- **Milestone 2 Project:**
  - Pull an existing Docker image (e.g., Nginx or MySQL), run it as a container, and connect to it.

Module 3: Building Custom Docker Images

- **Topics Covered:**
  - Introduction to Dockerfiles and creating custom images
  - Writing a Dockerfile: FROM, RUN, CMD, COPY, EXPOSE, etc.
  - Building and tagging Docker images
  - Best practices for writing efficient Dockerfiles
- **Milestone 3 Project:**
  - Create a Dockerfile to containerize a simple Node.js or Python application and build the custom image.

Module 4: Docker Volumes and Networking

- **Topics Covered:**
  - Introduction to volumes and data persistence
  - Mounting volumes and managing persistent data in containers
  - Understanding Docker networks and inter-container communication
  - Creating custom networks for containers
- **Milestone 4 Project:**
  - Create a multi-container setup with a web application and a database, using volumes for persistent storage.

Module 5: Docker Compose for Multi-Container Applications

- **Topics Covered:**
  - Introduction to Docker Compose and YAML syntax
  - Defining multi-container applications with *docker-compose.yml*
  - Managing dependencies and network configurations with Docker Compose
  - Common Docker Compose commands (*up*, *down*, *ps*)
- **Milestone 5 Project:**
  - Write a Docker Compose file to set up a simple web application with a frontend, backend, and database services.

Module 6: Managing Environment Variables and Secrets

- **Topics Covered:**
  - Using environment variables in Dockerfiles and Docker Compose

- Best practices for managing sensitive information with secrets
- Using *.env* files to handle configuration data
- Injecting environment variables at runtime

- **Milestone 6 Project:**
  - Refactor a Dockerized application to use environment variables for configuration and securely store secrets.

Module 7: Docker Registry and Image Management

- **Topics Covered:**
  - Introduction to Docker Hub and private registries
  - Pushing, pulling, and managing images in Docker Hub
  - Setting up a private Docker registry
  - Image versioning and tagging best practices
- **Milestone 7 Project:**
  - Push a custom Docker image to Docker Hub and pull it on a different environment, or set up a local Docker registry.

Module 8: Docker Swarm and Container Orchestration

- **Topics Covered:**
  - Introduction to Docker Swarm for orchestration
  - Creating and managing Docker services
  - Scaling applications with Docker Swarm
  - Rolling updates and service management
- **Milestone 8 Project:**
  - Set up a simple Docker Swarm with multiple services and deploy a scalable web application.

Module 9: Monitoring, Logging, and Debugging Containers

- **Topics Covered:**
  - Techniques for monitoring container performance (Docker stats, third-party tools)
  - Managing container logs and troubleshooting with Docker
  - Debugging Docker containers and resolving common issues
  - Overview of logging solutions: ELK stack, Prometheus, and Grafana
- **Milestone 9 Project:**
  - Set up monitoring and logging for a Dockerized application using a third-party tool (e.g., Prometheus, ELK).

Module 10: Advanced Docker Concepts and Best Practices

- **Topics Covered:**
  - Optimizing Docker images for performance and reduced size
  - Advanced networking: overlay networks, bridge networks
  - Using Docker in production: deployment strategies, security considerations
  - Overview of Kubernetes as an alternative orchestration tool

- **Final Project:**
    - Build and deploy a multi-container application with Docker Compose, integrate it with a private registry, and implement logging and monitoring solutions.

Assessment and Evaluation

- Milestone Projects: 60% of final grade
- Final Project: 30% of final grade
- Participation and Attendance: 10% of final grade

Resources

- Recommended Books:
    - "Docker Deep Dive" by Nigel Poulton
    - "The Docker Book" by James Turnbull
- Online Platforms:
    - Docker Documentation
    - Play with Docker

Course Delivery

- Method: Blended learning (theory and practical)
- Format: Lectures, hands-on containerization sessions, and project work