Introducing Docker Training

Duration: **1 Day**

BILGINC
IT Academy
info@bilginc.com
bilginc.com

Overview

Docker is an open-source containerisation platform used to package and run distributed applications in a consistent and portable way. Its core principle is that an application should be built once with its required dependencies and configuration, then run reliably across different environments.

Containerisation is similar to virtualisation but generally uses a lighter-weight approach. Applications are packaged into containers that include the information and configuration they need to run. Because containers can share underlying system resources, they can reduce the overhead associated with running multiple full virtual machines.

This course introduces the fundamental concepts behind Docker and containerisation before progressing into images, running containers, Dockerfile-based image creation, networking, registries, and persistent data management.

Participants learn how to run and inspect containers, manage images, build their own images, connect containers to networks, and persist or share data beyond the lifecycle of an individual container.

The course is strongly hands-on. Instructor explanations and demonstrations are combined with practical exercises, with the majority of the learning focused on direct experience using Docker.

Prerequisites

Participants should be comfortable with basic Linux terminal operations.

They should be able to:

- Change directories,
- Install packages using `apt-get` or `yum`,
- Edit files from the command line using tools such as`vim`, `vi`, `nano`, or a similar editor.

A good understanding of source control concepts and previous experience with Git are useful but not essential.

Who Should Attend

This course is particularly suitable for:

- Developers who are new to Docker and container technologies,
- Technical professionals beginning to work with DevOps practices,
- Teams that want to deploy applications more consistently across environments,
- Software and infrastructure professionals who want to understand container-based development and deployment.

Introducing Docker Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of this course, participants will be able to:

- Explain Docker images and containers,
- Understand how images provide consistent container configuration,
- Describe image layers and tagging conventions,
- Push and pull images from registries,
- Run and inspect containers,
- Build custom images using Dockerfiles,
- Understand multi-stage build concepts,
- Create and inspect Docker networks,
- Enable communication between containers,
- Explain the difference between Bind Mounts and Volume Mounts,
- Persist container data and make it available to other containers.

Outline

Images

- The role of images in providing consistent container configuration
- Understanding Docker images
- Image layer structure
- Benefits of layered image construction
- Standard image tagging conventions
- Image registries
- How registries store and distribute images
- Pulling images
- Pushing images
- Retagging existing images

Running Containers

- Introduction to common container deployment parameters
- Frequently used container runtime options
- Interacting with running containers
- Starting and stopping containers
- Extracting information from containers
- Inspecting container state
- Reviewing logs and runtime information

Building Images

- Introduction to Dockerfiles
- Core Dockerfile instructions
- Understanding the build process
- Building images from Dockerfiles
- Modifying Dockerfiles
- Creating custom images
- Multi-stage Builds

Introducing Docker Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Use cases for multi-stage builds
- Reading and understanding a multi-stage Dockerfile

Networking

- Why container-to-container networking is required
- Docker networking concepts
- Overview of built-in network drivers
- The four main built-in network driver types
- Benefits and use cases of each network driver
- Creating Docker networks
- Inspecting networks
- Using network information for troubleshooting
- Working with default networks
- Working with non-default networks
- Connecting containers to custom networks

External Mounts

- Why containers sometimes need to share files with the host
- Sharing data between containers
- Understanding Bind Mounts
- Understanding Volume Mounts
- Comparing Bind and Volume use cases
- Modifying application configuration at runtime with a bind-mounted configuration file
- Persisting data in a managed volume
- Making persistent data available to another container
- Managing data independently from the container lifecycle

Hands-On Practice

Throughout the course, participants complete practical activities covering:

- Docker installation and basic configuration,
- Running a first container,
- Working with existing images,
- Creating Dockerfiles,
- Building custom images,
- Pushing and pulling images,
- Container networking,
- Persistent data management,
- Bind and Volume mounts.