

Docker

Course Summary

Description

Linux containers are changing the way companies think about service development and deployment. Containers play a vital role in the modern data-center, and Docker is leading the way. This course covers all the core features of Docker including: container creation and management, interacting with Docker hub, using Dockerfile to create and manage custom images, advanced Docker networking (how to safely expose container services to the world, and link containers), the use of Docker volumes to manage persistent data, and Docker Compose to build multi-container applications. Emphasis is placed on best practices and how to secure Docker installations and containers. The course culminates with comprehensive labs where students use Docker, Git, and a continuous integration server to automate the testing of containerized applications.

Many large companies are moving an increasing number of applications to run inside containers. Containers can provide a high level of security while also making maximal use of hardware allowing higher density than traditional VMs. Docker is a dominant force in Linux containers and the core building block for nearly all higher level container management systems.

Supported Distributions: Red Hat Enterprise Linux 8

Objectives

After taking this course, sysadmins will be proficient with Docker so that they can immediately be productive as organizations continue to move applications onto containers.

Topics

- Container Technology Overview
- Installing Docker
- Managing Containers
- Managing Images
- Creating Images with Dockerfile
- Docker Volumes
- Docker Compose/Swarm
- Docker Networking
- Docker Logging

Prerequisites

Students should have proficiency with the Linux CLI. A broad understanding of Linux system administration.

Duration

Three days

Docker

Course Outline

I. *Container Technology Overview*

- A. Application Management Landscape
- B. Application Isolation
- C. Resource Measurement and Control
- D. Container Security
- E. OverlayFS Overview
- F. Container Security
- G. Open Container Initiative
- H. Docker Alternatives
- I. Docker Ecosystem

II. *Lab Tasks*

- A. Container Concepts runC
- B. Container Concepts Systemd

III. *Installing Docker*

- A. Installing Docker
- B. Docker Architecture
- C. Starting the Docker Daemon
- D. Docker Daemon Configuration
- E. Docker Control Socket
- F. Enabling TLS for Docker
- G. Validating Docker Install

IV. *Lab Tasks*

- A. Installing Docker
- B. Protecting Docker with TLS

V. *Managing Containers*

- A. Creating a New Container
- B. Listing Containers
- C. Managing Container Resources
- D. Running Commands in an Existing Container
- E. Interacting with a Running Container
- F. Stopping, Starting, and Removing Containers
- G. Copying files in/out of Containers
- H. Inspecting and Updating Containers
- I. Docker Output Filtering & Formatting

VI. *Lab Tasks*

- A. Managing Containers
- B. Configure a docker container to start at boot.

VII. *Managing Images*

- A. Docker Images
- B. Listing and Removing Images

- C. Searching for Images
- D. Downloading Images
- E. Uploading Images
- F. Export/Import Images
- G. Save/Load Images
- H. Committing Changes

VIII. *Lab Tasks*

- A. Docker Images
- B. Docker Platform Images

IX. *Creating Images With Dockerfile*

- A. Dockerfile
- B. Caching
- C. docker image build
- D. Dockerfile Instructions
- E. ENV and WORKDIR
- F. Running Commands
- G. Getting Files into the Image
- H. Defining Container Executable
- I. HEALTHCHECK
- J. Best Practices
- K. Multi-Stage builds with Dockerfile

X. *Lab Tasks*

- A. Dockerfile Fundamentals
- B. Optimizing Image Build Size
- C. Image Builds and Caching

XI. *Docker Volumes*

- A. Volume Concepts
- B. The docker volume Command
- C. Creating and Using Internal Volumes
- D. Internal Volume Drivers
- E. Removing Volumes
- F. Creating and Using External Volumes
- G. SELinux Considerations
- H. Mapping Devices

XII. *Lab Tasks*

- A. Docker Internal Volumes
- B. Docker External Volumes

Docker

Course Outline (continued)

XIII. Docker Compose/Swarm

- A. Writing YAML Files
- B. Concepts
- C. Compose CLI
- D. Defining a Service Set
- E. Compose Versions
- F. Docker Engine Swarm Mode
- G. Docker Swarm Terms
- H. Docker Swarm Command Overview
- I. Creating a Swarm
- J. Creating Services
- K. Creating Secrets
- L. Stack Files
- M. Stack Command
- N. Swarm Placements
- O. Swarm Resource Limits & Reservations
- P. Swarm Networking
- Q. Swarm Networking Troubleshooting

XIV. Lab Tasks

- A. Docker Compose
- B. Docker Engine Swarm Mode

XV. Docker Networking

- A. Overview
- B. Data-Link Layer Details
- C. Network Layer Details
- D. Hostnames and DNS
- E. Service Reachability
- F. Container to Container Communication
- G. Container to Container: Links (deprecated)
- H. Container to Container: Private Network
- I. Managing Private Networks
- J. Remote Host to Container

XVI. Lab Tasks

- A. Docker Networking
- B. Exposing Ports
- C. Docker Networking

Docker Logging

- D. Docker Logging
- E. Docker Logging with json-file and journald
- F. Docker Logging with syslog
- G. Docker Logging with Graylog or Logstash
- H. Docker Logging with Fluentd
- I. Docker Logging with Amazon or Google
- J. Docker Logging with Splunk

XVII. Lab Tasks

- A. Logging to syslog

XVIII. Docker Registry Lab Tasks

- A. Docker Registry
- B. Docker Registry (secured)
- C. Docker Content Trust