

Docker Containers

Length: 2 Days

Audience: Developers, Software Architects, Technical Project Managers, System Administrators.

Summary: Docker has taken our industry by storm. Hardly anyone can ignore the usefulness of containerized applications. Containers allow encapsulating part of your system configuration making it easier to distribute and manage applications at scale. This course covers Docker functionality, use cases and deployment scenarios. It is targeted at software developers and system administrators willing to integrate containers into software delivery and maintenance processes.

COURSE CONTENT

Introduction:

- Linux containers and Docker inception

Container concepts

- Host
- Engine
- Daemon
- Client
- Image
- Container
- Layer

Container images

- Exploring Docker Hub
- Pulling images from Docker Hub and private registries
- Exploring local image cache
- Loading images into local image cache

Running containers

- Long running containers
- Throw-away containers
- Exposing ports
- Setting environment variables
- Sharing data with the host and setting limits.

Exploring container state

- Listing containers
- Status and resource usage
- Standard streams
- Stopping containers
- Killing containers
- Pausing containers
- Removing containers.

Creating images

- Images from existing containers
- Building images using Dockerfile
- Selecting base image
- Defining image parameters
- Removing images.

Distributing images

- Through a registry or as a TAR archive
- Squashing images
- Minimizing

Pipelines

- Integrating Docker containers into a software delivery pipeline.

Networking

- Creating networks and "linking" containers
- Handling communication with the host
- Port forwarding
- Managing different network types.

Orchestrating

- Creating Docker Compose configuration for defining containers
- Networks and volumes.

Clustering

- Using remote Docker engine
- Setting up Docker Swarm
- Configuring services and stacks
- Load-balancing.

Monitoring

- Monitoring Docker host
 - Visualizing image and container layer dependencies
 - Managing container logs.
- 