



Docker and Kubernetes Fundamentals Training

Course ISI-1563 Three days Instructor-led

Introduction

Are you a developer, architect, or technical team lead without a foundational understanding of Kubernetes and Docker? To meet the demands of your workplace, you will need to have a strong grasp of the Kubernetes and Docker technologies in the development, production, and staging environments.

Kubernetes is one of the leading emerging technologies and has become famous for the flexibility it brings to the management of containerized applications in the cloud.

This 3-day intensive Docker and Kubernetes Training course offers a combination of theoretical learning and actional labs, so you have the opportunity to apply your knowledge in a hands-on setting. The course covers essential Kubernetes and Docker skills, as well as a basic introduction to Continuous Integration.

Audience Profile

Developers, system architects, technical team leads, systems administrators, and DevOps professionals who want to understand and use Kubernetes in cloud environments.

Course Objectives

- Learn the fundamental uses of container technology and the architectural principles of how to apply containers to technical challenges.
- Create containers with both packaged solutions and custom software.
- Use the Docker and Kubernetes CLI environments to deploy solutions in local (Docker) and cluster (Kubernetes) environments.
- Learn how stateful services – such as databases, streaming solutions , and other specialty applications – can be deployed in highly-available configurations with redundancy and service-discovery/load-balancing.
- Show how other components – can be used to integrate services within the Kubernetes platform.
- Explore how Docker and technologies such as Jenkins can be used to continuously test and build containers based on changes to source code

Prerequisites

- Proficiency with the Linux Command Line Interface (CLI)

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>



- Broad understanding of Linux system administration: Linux file system, networking, and bash scripting.
- Computer programming concepts and methodologies. Full code examples for the course are provided in the Python and Java programming languages.

Student Materials

Each student receives a printed copy of the courseware

Course Outline

Module 1: Docker and Linux Container Technology: Introduction and Use-Cases

- Modern Infrastructure Terminology
- Virtualization
- Hypervisors
- Hypervisor Types
- Type 1 Hypervisors
- Type 2 Hypervisors
- Type 1 vs Type 2 Processing
- Paravirtualization
- Virtualization Qualities (1/2)
- Virtualization Qualities (2/2)
- Disadvantages of Virtualization
- Containerization
- Virtualization vs Containerization
- Where to Use Virtualization and Containerization
- Containerization: High-Level
- Popular Containerization Systems
- What are Linux Containers
- Docker
- OpenVZ
- Solaris Zones (Containers)
- Container Orchestration Tools
- Docker Swarm
- Kubernetes
- Mesos and Marathon
- Mesos and Marathon (contd.)
- Docker Use-Cases
- Microservices
- Microservices and Containers / Clusters
- Summary

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

Module 2: Docker in Action

- Docker Basics
- Where Can I Run Docker?
- Installing Docker Container Engine
- Docker Toolbox
- What is Docker?
- Docker Architecture
- Docker Architecture Diagram
- Docker Images
- Docker Containers
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Docker Run Command
- Starting, Inspecting, and Stopping Docker Containers
- Docker Volume
- Dockerfile
- Docker Compose
- Using Docker Compose
- Dissecting docker-compose.yml
- Specifying services
- Dependencies between containers
- Injecting Environment Variables
- Summary

Module 3: Managing Docker State

- State and Data in Docker
- Volumes
- More About Volumes
- Uses for Volumes
- Working With Volumes
- Create Volume
- Use Volumes with Containers
- Bind Mounts
- Using Bind Mounts
- tmpfs Mounts
- Storing Data in the Container
- Storage Drivers
- Remote Data Storage
- Networking
- The Default Bridge Network
- User-Defined Bridge Networks

- Docker Network Commands
- Creating a User-Defined Bridge Network
- Summary

Module 4: Open Container Initiative and Container Runtime Interface

- Open Container Initiative (OCI)
- Docker
- Docker Engine Architecture
- runC
- containerd
- containerd Benefits
- CRI-O
- CRI-O Components
- Kubernetes and CRI-O
- Using Container Runtimes with Minikube
- Docker Runtime and Kubernetes
- Putting Things Together
- Summary

Module 5: Kubernetes Architecture

- Kubernetes Basics
- What is Kubernetes?
- Container Orchestration
- Architecture Diagram
- Components
- Kubernetes Cluster
- Master Node
- Kube-Control-Manager
- Nodes
- Pod
- Using Pods to Group Containers
- Label
- Label Syntax
- Label Selector
- Annotation
- Persistent Storage
- Resource Quota
- Interacting with Kubernetes
- Summary

Module 6: Working with Kubernetes

- Installation
- Startup
- Kubernetes Tools
- kubectl Command Line Interface
- API Proxy
- Dashboard
- Kubernetes Component Hierarchy
- Deployments
- Deployment Commands
- Updating Deployments
- Network Considerations
- Services
- Namespaces
- Labels
- Annotations
- Other Useful Commands
- Summary

Module 7: Kubernetes Workload

- Kubernetes Workload
- Kubernetes Workload (contd.)
- Managing Workloads
- Imperative commands
- Imperative Object Configuration
- Declarative Object Configuration
- Configuration File Schema
- Understanding API Version
- Obtaining API Versions
- Obtaining API Versions (contd.)
- Stateless Applications
- Sample Deployment Manifest File
- Working with Deployments
- Stateful Applications
- Sample Stateful Manifest File
- Sample Stateful Manifest File (Contd.)
- Working with StatefulSet
- Jobs
- Sample Job Manifest File
- Sample Job Manifest File (Contd.)
- Working with Batch Job
- DaemonSets



- DaemonSets (contd.)
- Sample Daemon Manifest File
- Rolling Updates
- Rolling Updates (Contd.)
- Summary

Module 8: Scheduling and Node Management

- Kubernetes Scheduler
- Kubernetes Scheduler Overview (contd.)
- Skip Kubernetes Scheduler
- Scheduling Process
- Scheduling Process - Predicates
- Scheduling Process - Priorities
- Scheduling Algorithm
- Kubernetes Scheduling Algorithm
- Scheduling Conflicts
- Controlling Scheduling
- Label Selectors
- Label Selectors (contd.)
- Label Selectors (Contd.)
- Node Affinity and Anti-affinity
- Node Affinity Example
- Node Antiaffinity Example
- Taints and Tolerations
- Taints and Tolerations (Contd.)
- Taints and Tolerations - Example
- Summary

Module 9: Managing Networking

- Kubernetes Networking Components
- The Kubernetes Network Model
- Networking Scenarios
- Container-Container Communication
- Pod-Pod Communication
- 1.3 Pod-Service Communication
- External-Service Communication
- Accessing Applications
- Useful Commands
- Container Network Interface (CNI)
- What is CNI's Role?
- CNI Configuration Format
- Sample CNI Configuration
- Running the CNI Plugins

Contact ISINC for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Summary

Module 10: Managing Persistent Storage

- Storage Methods
- Container OS file system storage
- Docker Volumes
- Kubernetes Volumes
- K8S Volume Types
- Cloud Resource Types
- configMaps
- Creating configMaps from Literals
- Creating configMaps from files
- Using configMaps
- emptyDir
- Using an emptyDir Volume
- Other Volume Types
- Persistent Volumes
- Creating a Volume
- Persistent Volume Claim
- Persistent Volume
- Pod that uses Persistent Volume
- Secrets
- Creating Secrets from Files
- Creating Secrets from Literals
- Using Secrets
- Security Context
- Security Context Usage
- Summary

Module 11: Working with Helm

- What is Helm?
- Installing Helm
- Helm and KUBECONFIG
- Helm Features
- Helm Terminology
- Searching for Charts with helm CLI
- Adding Repositories
- Helm Hub - Search
- Helm Hub - Chart Page
- Installing a Chart
- Upgrading a Release
- Rolling Back a Release
- Creating Custom Charts

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>



- Common Chart Files
- Helm Templates
- Installing A Custom Chart
- Packaging Custom Charts
- Summary

Module 12: Logging, Monitoring, and Troubleshooting

- Differences Between Logging and Monitoring
- Logging in Kubernetes
- Basic Logging
- Logging Agents
- Fluentd and Elastic Stack
- Monitoring with Prometheus
- Kubernetes and Prometheus - Metrics
- Alerting
- Debugging Pods
- Debugging Pods (Contd.)
- Debugging Nodes
- Debugging Replication Controllers and Services
- Upgrading Kubernetes
- Upgrade Process
- Determine Which Version to Upgrade To
- Upgrade kubeadm
- Upgrade Control Plane Node
- Upgrade kubelet and kubectl
- Upgrade Worker Nodes
- Recovering From a Failure State
- Summary

Module 13: Continuous Integration Fundamentals

- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins
- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- Installing Jenkins as a Windows Service
- Different types of Jenkins job
- Configuring Source Code Management(SCM)
- Working with Subversion
- Working with Subversion (cont'd)
- Working with Git
- Build Triggers

Contact ISINC for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>



- Schedule Build Jobs
- Polling the SCM
- Maven Build Steps
- Configuring Jenkins to Access Kubernetes
- Jenkins Pipeline
- Jenkins Pipeline Output
- Installing Jenkins Plugins
- Summary

Module 14: Lab Exercises

Lab 1. Managing Containers

Lab 2. Building Images

Lab 3. Dockerfiles

Lab 4. Deploying Stateful Services in Docker

Lab 5. Custom Network Management

Lab 6. Docker Volumes

Lab 7. Accessing the Kubernetes API

Lab 8. Working with Kubernetes Workloads

Lab 9. Scheduling and Node Management

Lab 10. Accessing Applications

Lab 11. Using Persistent Storage

Lab 12. Getting Started with Helm

Lab 13. Build CI Pipeline with Jenkins