

# HPE Digital Learner Containers: Docker and Kubernetes Content Pack

<b>HPE Content Pack number</b>	CP028
<b>Content Pack length</b>	25 Hours
<b>Content Pack category</b>	Category 2
<b>Learn more</b>	<a href="#">View now</a>

In this course, you will learn to create, deploy, run and scale containerized applications using Docker and Kubernetes.

## Why HPE Education Services?

- IDC MarketScape leader 5 years running for IT education and training\*
- Recognized by IDC for leading with global coverage, unmatched technical expertise, and targeted education consulting services\*
- Key partnerships with industry leaders OpenStack®, VMware®, Linux®, Microsoft®, ITIL, PMI, CSA, and SUSE
- Complete continuum of training delivery options—self-paced eLearning, custom education consulting, traditional classroom, video on-demand instruction, live virtual instructor-led with hands-on lab, dedicated onsite training
- Simplified purchase option with HPE Training Credits

## Audience

System administrators, DevOps personnel, operations staff, managers, software developers, testers, architects and deployment engineers

## Content Pack objectives

- Install and configure Docker, work with Docker containers and images; link containers and manage data in containers
- Work with Docker containers, Docker images, link containers and manage data in containers
- Learn advanced Docker concepts such as managing images, working with the Dockerfile, volumes, and networking
- Work with the Docker Trusted Registry, use the Docker machine and set up Docker security
- Learn how to work with Docker Hub and deploy Docker in the cloud
- Learn how to configure and use Kubernetes on a local machine
- Describe Docker and how it is used with Kubernetes
- Learn the key concepts and controllers involved in Kubernetes
- Learn and use the Kubernetes API, controller manager, scheduler and network proxy

## Detailed Content Pack outline

---

### Using Docker

#### Benefits and Installation

Docker is an open source platform that enables developers, administrators and others to access and deploy the latest updated images into their respective environments. It uses automated scripts and manual commands to perform tasks. It allows software application to get into production as quickly as possible by eliminating friction and dependencies that normally occur during shipment of code. In this course, you will learn the basic concepts of Docker and the Docker architecture, and how to install Docker on Ubuntu and CentOS.

- Describe Docker and the problem it solves
- Describe the uses of Docker
- Compare Docker with virtual machine deployments
- Describe Docker architecture
- Identify internal components on which Docker is based
- Differentiate between public and private registries in Docker
- Describe the process of accessing Docker hub
- Identify the use of namespaces in Docker
- Identify the use of control groups in Docker
- Identify the use of union systems in Docker
- Verify minimum requirements for Ubuntu
- Install Docker on Ubuntu
- Create a Docker group
- Configure memory for Docker
- Configure UFW forwarding in Docker
- Configure DNS for Docker
- Describe CentOS version and kernel support
- Install a Docker package on CentOS 6.5
- Start and verify Docker daemon in CentOS
- Describe the process of templating an image
- Use the run command to spawn a Docker container
- Run and interact with a Docker container
- List the steps to access, interact and see results from a command executed on an image from a local or remote registry

#### Dockerizing Applications

Applications can be packaged and deployed in multiple ways using Docker. Docker containers, images and client daemons help administrators, developers and testers quickly access updated applications and make necessary changes before production. In this course, you will learn the concept of Dockerization, how to work with Docker containers, Docker images, how to link containers and how to manage data in containers.

- Identify the features of a Dockerized application
- List the steps to spawn an interactive Docker container
- List the steps to daemonize a Docker container
- List the steps and commands to access a Docker daemon
- Identify the syntax and version of a Docker command
- Identify Docker client commands and usage
- Describe the command to execute an application in Docker
- Describe container and port details
- Use commands to name and inspect config details in JSON
- Identify the step to list available Docker images
- List the steps to pull a Docker image
- List the steps to search for Docker images
- List the steps to update, create and commit a Docker image
- List the steps to push and remove an image from the host or registry
- Identify the use of the -P and -D flags to control port access
- Identify the steps to link containers by referencing through names
- Use environment variables to link containers
- List available environment variables and its impact on /etc/hosts file
- Understand data volumes
- Identify the steps to configure data volumes
- List the steps to create a data volume container to share data between containers
- Identify the steps to back up and restore data
- Define steps to edit an Ubuntu image for adding a command to it

#### Customizing and Clustering

Docker containers can be customized and clustered to effectively manage Dockerization of applications. Custom metadata can be used for labeling and creating data structures. Docker clusters can be used to group a pool of Docker hosts into a single virtual host. In this course, you learn about custom meta data, Docker Compose, Docker Clustering and Swarm, Docker Swarm filters and Docker Swarm strategies.

- Identify the need for custom metadata in Docker
- Describe how different types of data can be stored in labels
- Describe steps to add labels to an image
- Describe steps to filter images by label
- Identify the need for the Docker Compose tool to define and run applications
- Describe the steps to install Compose in Docker
- Describe how to build a DockerFile for defining images
- Build a .yaml file for defining Docker services
- Build and run an application in Docker Compose
- Identify the need for clustering Docker hosts and prerequisites for Swarm
- List the steps to set up Swarm nodes for clustering
- Use commands to access and list nodes
- Work with Discovery services and list the different approaches
- Describe the need for filters and its various types
- List the steps to configure a constraint filter
- List the steps to configure an affinity filter
- List the steps to configure a port filter
- List the steps to configure a dependency and health filter
- Identify the need for Swarm strategies and list different types of strategies
- Use BinPack strategy to choose a Docker node in a cluster
- Use Spread strategy to choose a Docker node in a cluster
- Implement steps to create a data label

## Docker – Beyond the Basics

### Docker Configuration and Management

Since the demand for IT personnel having an in depth knowledge of Docker is exploding, a detailed understanding of the configuration and management of Docker has become a fundamental requirement for a skilled DevOps developer. This course will teach you the configuration and management techniques of Docker while covering some advanced topics, such as managing images, working with Dockerfiles, volumes and networking.

- Describe the various steps to create an account on Docker Hub
- Define the process of creating a Docker image from an existing container
- Apply the Docker push command to push the image to the registry
- Use the Docker history command to find all the intermediate image layers
- Distinguish between the Docker commands when performing image maintenance
- Recall the components and the steps in building Docker images
- Summarize the different approaches to building Docker images
- Use Docker commands to run an interactive shell
- Create a Dockerfile that layers on the server and the site content in a single build
- Create Docker images using a Dockerfile
- Install the fedora-dockerfiles package to get Dockerfiles
- Create a Docker container that just runs Firefox
- Run an application that requires running multiple services simultaneously
- Run Docker inspect to examine volume configuration
- Describe the methods and the attributes to gain access to volumes
- Analyze how a deployed web server is configured and built
- Describe where to get files to create and write the map
- List the steps to run the container server
- Use the worker to bridge multiple servers together with volumes
- List the Docker network-related command line options
- Describe how Docker configures DNS with containers and virtual files
- Describe how the communication between two containers is governed at the operating system level
- Recognize whether a container can communicate to the outside world
- Describe how to configure containers to accept incoming connections
- Describe how to use Pipework for Docker container network configuration
- Use Docker to create and layer an image

### Working with Docker Machine, Security and APIs

Security is an important aspect of an open source platform. Therefore, in depth knowledge of Docker security is a must-have for an application developer. In this course, you will learn to work with the Docker Trusted Registry, use the Docker machine and set up Docker security. You will also learn about the various Docker applications and services.

- Describe how Docker handles IPv4 and IPv6 addressing
- Use NDP proxying to connect Docker containers to the Internet via IPv6
- Describe how Docker uses switched and routed network environments
- Set up SELinux on CentOS/RHEL/Fedora systems
- Describe how to allow a nonprivileged volume access to host files from a container
- Describe how Linux divides the privileges of the root user
- Share a namespace between two or more containers
- Create a Docker image with a preinstalled Riak
- Create an SSHd service in a Docker container
- Create a Docker container for MongoDB
- Use Shippable to perform CI/CD and then deploy it on Red Hat's OpenShift
- Use Drone to configure projects to automatically build, test and deploy
- Use Docker with OpenShift Origin to configure Platform-as-a-Service
- Describe how OpenShift triggers the Docker build and how to supply to the Docker context to build images
- Use the Docker Trusted Registry Accounts API
- Use the Docker Trusted Registry Teams API
- Use the Docker Trusted Registry Accounts Repositories API
- Use the Docker User Owned Repository Access API
- Use the Docker Organization Owned Repository Access API
- Use the Docker Organization Owned Repository Namespace API
- Describe how the Docker machine creates Docker hosts on your computer or data center
- Use the Docker machine to create, use and manage a Docker host inside of VirtualBox
- Use the Docker machine to use the same interface to create hosts in local, virtual or cloud platforms
- Distinguish between the different drivers on the Docker machine and describe the tasks they perform
- Describe the Docker Machine commands and how they are used
- Use Docker to create a complex deployment using IP addressing, security and DTR APIs

## Docker for Windows Server

### Docker Introduction

This course provides an introduction to containerization and Docker terminology. You will learn how Dockerfiles are used to build Docker images and how a container is a running instance of an image.

- Describe the concept and benefits of virtualization
- Describe containerization
- Compare virtual machines and containers
- Describe when containers should be used
- Define the purpose of Docker
- Describe the purpose of Dockerfiles
- Describe the purpose of Docker images
- Describe when to use Docker Compose
- Describe how continuous integration benefits software development and testing
- Determine when and how to use Docker Swarm cluster and orchestration features
- Describe how UCP is used as a single Docker management interface
- Describe how DTR manages Docker images
- Recall Docker container-related terminology

### Docker and Windows

This course describes how Docker integrates with Windows 10 and Windows Server 2016. The distinction between Windows Server and Hyper-V containers are also covered.

- Describe how Docker is integrated with Server 2016
- Identify Docker Community Edition characteristics
- Identify Docker Enterprise Edition characteristics
- Describe how Docker is integrated with Windows 10
- Compare and contrast Windows container types
- Describe how Hyper-V containers are used
- List command PowerShell cmdlets used to manage Windows containers
- Describe how Visual Studio is used with Windows containers
- Describe the concept of containers and how they compare to virtual machines

### Docker Installation and Management

In this course, you will learn how to install Docker on Windows 10 and Windows Server 2016. You will then learn how to manage images and containers.

- Determine what is needed to support a Docker infrastructure
- Walk through the process of creating a Docker account
- Install Docker support on the Server 2016 platform
- Install Docker support on the Windows 10 platform
- Install Docker updates on the Windows platform
- Use the Docker CLI to pull images down to a Windows 10 Docker host
- Use the Docker CLI to pull images down to a Windows Server 2016 Docker host
- Run a Linux container on the Windows platform
- Describe Docker network and port configurations
- Ensure connectivity to a Docker web app is functional
- Make the Docker PowerShell module available in PowerShell
- Use New-Container in PowerShell
- Describe how to share and tag a Docker container
- Initialize Docker Swarm nodes
- Determine when a Docker cloud deployment is appropriate
- Create a Docker configuration in the Microsoft Azure cloud
- Recall how to deploy Docker containers on-premises and in the cloud

### Docker Programmatic Access

In this course, Docker command line tools and APIs are discussed. Then, the Docker CLI is used to manage various aspects of the Docker ecosystem.

- Describe Docker APIs and how they are used
- Describe the Engine CLI and related commands
- Manage Docker using Engine CLI commands
- Describe the Compose CLI and related commands
- Manage Docker using Compose CLI commands
- Describe the Machine CLI and related commands
- Use the CLI to provision a Docker virtual machine
- Describe the DTR CLI and related commands
- Manage Docker containers using command line tools

## Mastering Docker for DevOps

### Docker Containers, Images and Swarms

This course demonstrates how to work with Docker containers and swarms. You will also learn how Docker fits into the DevOps landscape.

- Describe the role Docker can play in the DevOps landscape
- Describe the key concepts of containers and how they are managed
- Develop with Dockerfile while ensuring that best practices are followed
- Use Dockerfile to add an application to a container and run the application
- Create a web server using Dockerfile
- Recall key concepts of Docker swarm mode and how it relates to DevOps
- Use Docker to create a new swarm and add nodes to the swarm
- Deploy and inspect a service in Docker swarm
- Use the Docker CLI to scale containers in the service
- Upgrade a service in Docker using rolling updates
- Use Docker secrets to work with sensitive data
- Work with configuration files using swarm service configs
- Create and configure a Docker swarm

### Managing Stacks and Application Deployment

This course demonstrates and outlines management of Docker stacks and Distributed Application Bundles (DABs). You will also learn how to work with Docker Hub and deploy Docker in the cloud.

- Define stacks and use Docker to create one
- Add a new service to a stack and redeploy
- Demonstrate how to incorporate data persistence using redis
- Use Docker Compose to create a DAB
- Describe the Docker Hub registry service, including Docker ID
- Define how you can use Docker Hub repositories to search for images
- Describe how to create both public and private repositories
- Tag and publish a Docker image
- Recall how to perform automated builds
- Describe the various Docker cloud deployment models
- Deploy an AWS virtual machine running Docker
- Deploy a Microsoft Azure virtual machine running Docker
- Deploy a Docker app on the cloud and verify the deployment

## Kubernetes and Application Containers

### Introduction to Kubernetes

In this course, you will learn how to configure and use Kubernetes on a local machine.

- Describe Kubernetes and how it is used for container management
- Describe Docker and how it is used with Kubernetes
- Install Docker
- Work with Docker
- Navigate and use Docker Hub
- Work with Docker Swarms
- Work with Docker commands to clean up images and containers
- Describe using Kubernetes on a local machine
- Install Minikube on a Windows 10 local machine
- Install Minikube locally on a Mac operating system
- Install Kubernetes on an Ubuntu Linux local machine
- Use Kubectl command line interface tool
- Use Kubernetes on a local machine

### Kubernetes on Google Cloud Platform

In this course, you will learn how to run Kubernetes on the Google Cloud Platform.

- Create a Kubernetes cluster using the Google Cloud Platform Console
- Use Google Cloud Shell to create a Kubernetes cluster
- Install the Google Cloud SDK
- Install the kubectl command line tool in the Google Cloud SDK shell
- Use gcloud and kubectl
- Delete clusters and services using the Google Cloud SDK Shell
- Deploy containers from Google Cloud Launcher to Kubernetes
- Use the Kubernetes Engine dashboards on the Google Cloud Platform
- Use the compute engine dashboard and API
- Work with Kubernetes on the Google Cloud Platform

**Kubernetes Concepts**

In this course, you will learn the key concepts and controllers involved in Kubernetes, which will help you get an in depth understanding of how it works.

- Work with containers in Kubernetes
- Use namespaces in Kubernetes
- Use nodes in Kubernetes
- Work with PetSets in Kubernetes
- Work with pods in Kubernetes
- Work with Ingress objects in Kubernetes
- Work with a ReplicationController in Kubernetes
- Implement autoscaling in Kubernetes
- Use Services in Kubernetes
- Use ReplicaSets in Kubernetes
- Work with volumes in Kubernetes
- Use Deployments in Kubernetes
- Use Secrets and ConfigMaps in Kubernetes
- Work with StatefulSets in Kubernetes
- Describe Names in Kubernetes
- Work with DaemonSets in Kubernetes
- Work with annotations in Kubernetes
- Implement Jobs in Kubernetes
- Work with labels and selectors in Kubernetes
- Work with Kubernetes concepts

**Kubernetes Clusters**

In this course, you will learn about Kubernetes clusters and how to use the kubectl command line tool to create, start, stop, delete and much more.

- Create a Kubernetes cluster on Google Cloud Platform and using Minikube on Windows 10
- Work with Kubernetes Cluster Autoscaler
- Resize a Kubernetes cluster
- Prepare an environment to work with multicluster Ingress with Kubernetes clusters
- Stop a Kubernetes cluster
- Deploy an application to multiple Kubernetes clusters
- Delete a cluster in Kubernetes
- Work with the kubemci command line tool to deploy global multicluster Ingress with Kubernetes clusters
- Upgrade a Kubernetes cluster
- Test global multicluster Ingress with Kubernetes clusters
- Describe Kubeconfig files
- Perform a tear down of a multicluster Ingress deployment
- Share a Kubernetes cluster
- Work with Kubernetes clusters
- Authenticate a cluster in Kubernetes
- Create Kubernetes alpha clusters

**Kubernetes API and Components**

In this course, you will learn about the Kubernetes API and other key components of Kubernetes.

- Describe the Kubernetes Master
- Describe the Kubernetes API server
- Describe Kubernetes nodes and add-ons
- Work with the Kubernetes controller manager
- Use the Kubernetes API
- Describe the use of the Kubernetes scheduler function
- Work with the autoscaling API in Kubernetes
- Use the Kubernetes network proxy
- Use the batch API in Kubernetes
- Work with kubelet
- Work with the apps API in Kubernetes
- Work with the Kubernetes API
- Recall how to work with the extensions API in Kubernetes

Learn more at

[www.hpe.com/ww/digitallearner](http://www.hpe.com/ww/digitallearner)

[www.hpe.com/ww/digitallearner-contentpack](http://www.hpe.com/ww/digitallearner-contentpack)

Follow us:



---

© Copyright 2019 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. The OpenStack Word Mark is either a registered trademark/service mark or trademark/service mark of the OpenStack Foundation, in the United States and other countries and is used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community. Pivotal and Cloud Foundry are trademarks and/or registered trademarks of Pivotal Software, Inc. in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

CP028 A.00, February 2019