Docker Fundamentals

Docker Training Program

The Docker training program leverages the pedagogical approach of learning by doing with extensive hands-on labs, enterprise-focused scenarios, and practical examples. Docker training courses are updated regularly to ensure that learners are exposed to the latest product releases and current best practices informed by Docker’s extensive field experience.

Each course features a variety of assessment instruments from practice quiz questions, lab exercises, to project-based signature assignment for learners to practice and meet the learning objectives of each course.

Course Description

The Docker Fundamentals training course features the foundational concepts and practices of containerization on a single Docker node. The course offers learners the opportunity to assimilate basic container orchestration and how to scale Docker across multiple nodes in a simple swarm cluster. This course provides essential foundational knowledge for subsequent Docker courses.

Learning Objectives

By the end of the course successful learners will be able to:

* Understand the foundations of containerization on a single Docker node
* Create an image using Dockerfile best practices
* Use volumes in the application development process
* Apply concepts of the Docker networking model
* Understand the goal of services as a method of scaling containers
* Utilize two different orchestrators (Swarm and Kubernetes) to deploy a single application across multiple machines
* Create a secret and understand its accessibility capabilities

Course Outline

Day 1

* Introducing Docker
* Containerization Fundamentals
* Creating Images
* Docker Volumes

Day 2

* Docker Networking Basics
* Introduction to Docker Compose
* Introduction to Swarm Mode
* Introduction to Kubernetes
* Secrets
* Fundamentals Signature Assignment

Docker for Enterprise Developers

Docker Training Program

The Docker training program leverages the pedagogical approach of learning by doing with extensive hands-on labs, enterprise-focused scenarios, and practical examples. Docker training courses are updated regularly to ensure that learners are exposed to the latest product releases and current best practices informed by Docker’s extensive field experience.

Each course features a variety of assessment instruments from practice quiz questions, lab exercises, to project-based signature assignment for learners to practice and meet the learning objectives of each course.

Course Description

As the follow-on to the Docker Fundamentals course, Docker for Enterprise Developers is a role-based course designed for an organization’s Development and DevOps teams to accelerate their Docker journey in the enterprise. The course covers best practices to containerize and modernize legacy applications or build containerized applications from scratch that are secure, robust, highly available, resilient and self-healing. It is highly recommended to complete the Docker Fundamentals course as a pre-requisite.

Learning Objectives

By the end of the course successful learners will be able to:

* Describe the essential patterns used in a highly distributed EE application
* Understand how to configure EE applications for different environments without code changes
* Produce and containerize EE applications that are scalable, accessible, and fault-tolerant
* Apply different debugging and testing techniques to containerized EE applications
* Build and run the sample application on a local system using Kubernetes

Course Outline

Day 1

* Distributed Application Architecture
* Sample Application
* Edit and Continue
* Debugging
* Docker Compose
* Testing
* Service Discovery
* Health Checks
* Defensive Programming
* Logging and Error Handling
* Builder
* Docker Swarm and Kubernetes
* Secrets

Day 2

* Configuration Management
* Development Pipeline Overview
* Universal Control Plane
* Context Based Routing
* Docker Trusted Registry
* Content Trust
* Image Security Scanning
* Repository Automation
* Tagging and Versioning Strategies
* Build Server

Docker Security

Docker Training Program

The Docker training program leverages the pedagogical approach of learning by doing with extensive hands-on labs, enterprise-focused scenarios, and practical examples. Docker training courses are updated regularly to ensure that learners are exposed to the latest product releases and current best practices informed by Docker’s extensive field experience.

Each course features a variety of assessment instruments from practice quiz questions, lab exercises, to project-based signature assignment for learners to practice and meet the learning objectives of each course.

Course Description

The Docker Security course is an advanced workshop style course designed to be inclusive of multiple roles: Developer, Operations, DevOps, or Architects. The course offers learners a hands-on overview of important security features and best practices to protect containerized services. Completion of the Docker Fundamentals and Enterprise Operations course is strongly recommended as a prerequisite.

Learning Objectives

By the end of the course successful learners will be able to

* List all the Linux and network features imposed on containers by the Docker platform, and configure them where configurable
* Design and implement discretionary access control for users on the Docker EE platform
* Fully audit the provenance, contents, and actions taken by containerized software from creation as an image, through testing and QA, and into deployment as a container in production

Course Outline

Day 1

* Container Security
* Secure Networking
* Authentication and Authorization
* Secure Software Supply Chain
* Introspection
* Signature Assignment

## Content

### Domain 1: Orchestration (25% of exam)

Content may include the following:

* [Complete the setup of a swarm mode cluster with managers and worker nodes](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Complete_the_setup_of_a_swarm_mode_cluster_with_managers_and_worker_nodes.md)
* [State the differences between running a container vs running a service](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/State_the_differences_between_running_a_container_vs_running_a_service.md)
* [Demonstrate steps to lock a swarm cluster](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Demonstrate_steps_to_lock_a_swarm_cluster.md)
* [Extend the instructions to run individual containers into running services under swarm](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Extend_the_instructions_to_run_individual_containers_into_running_services_under_swarm.md)
* [Interpret the output of docker inspect commands](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Interpret_the_output_of_docker_inspect_commands.md)
* [Convert an application deployment into a stack file using a YAML compose file with docker stack deploy](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Convert_an_application_deployment_into_a_stack_file_using_a_YAML_compose_file_with_docker_stack_deploy.md)
* [Manipulate a running stack of services](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Manipulate_a_running_stack_of_services.md)
* [Increase number of replicas](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Increase_number_of_replicas.md)
* [Add networks publish ports](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Add_networks_publish_ports.md)
* [Mount volumes](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Mount_volumes.md)
* [Illustrate running a replicated vs global service](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Illustrate_running_a_replicated_vs_global_service.md)
* [Identify the steps needed to troubleshoot a service not deploying](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Identify_the_steps_needed_to_troubleshoot_a_service_not_deploying.md)
* [Apply node labels to demonstrate placement of tasks](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Apply_node_labels_to_demonstrate_placement_of_tasks.md)
* [Sketch how a Dockerized application communicates with legacy systems](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Sketch_how_a_Dockerized_application_communicates_with_legacy_systems.md)
* [Paraphrase the importance of quorum in a swarm cluster](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Paraphrase_the_importance_of_quorum_in_a_swarm_cluster.md)
* [Demonstrate the usage of templates with docker service create](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_1_Orchestration/Demonstrate_the_usage_of_templates_with_docker_service_create.md)

### Domain 2: Image Creation, Management, and Registry (20% of exam)

Content may include the following:

* [Describe Dockerfile options](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Describe_Dockerfile_options.md)
* [Show the main parts of a Dockerfile](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Show_the_main_parts_of_a_Dockerfile.md)
* [Give examples on how to create an efficient image via a Dockerfile](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Give_examples_on_how_to_create_an_efficient_image_via_a_Dockerfile.md)
* [Use CLI commands such as list delete prune rmi etc to manage images](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Use_CLI_commands_such_as_list_delete_prune_rmi_etc_to_manage_images.md)
* [Inspect images and report specific attributes using filter and format](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Inspect_images_and_report_specific_attributes_using_filter_and_format.md)
* [Demonstrate tagging an image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Demonstrate_tagging_an_image.md)
* [Utilize a registry to store an image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Utilize_a_registry_to_store_an_image.md)
* [Display layers of a Docker image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Display_layers_of_a_Docker_image.md)
* [Apply a file to create a Docker image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Apply_a_file_to_create_a_Docker_image.md)
* [Modify an image to a single layer](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Modify_an_image_to_a_single_layer.md)
* [Describe how image layers work](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Describe_how_image_layers_work.md)
* [Deploy a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Deploy_a_registry.md)
* [Configure a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Configure_a_registry.md)
* [Log into a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Log_into_a_registry.md)
* [Utilize search in a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Utilize_search_in_a_registry.md)
* [Tag an image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Tag_an_image.md)
* [Push an image to a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Push_an_image_to_a_registry.md)
* [Sign an image in a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Sign_an_image_in_a_registry.md)
* [Pull an image from a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Pull_an_image_from_a_registry.md)
* [Describe how image deletion works](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Describe_how_image_deletion_works.md)
* [Delete an image from a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_2_Image_Creation_Management_and_Registry/Delete_an_image_from_a_registry.md)

### Domain 3: Installation and Configuration (15% of exam)

Content may include the following:

* [Demonstrate the ability to upgrade the Docker engine](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Demonstrate_the_ability_to_upgrade_the_Docker_engine.md)
* [Complete setup of repo select a storage driver and complete installation of Docker engine on multiple platforms](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Complete_setup_of_repo_select_a_storage_driver_and_complete_installation_of_Docker_engine_on_multiple_platforms.md)
* [Configure logging drivers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Configure_logging_drivers.md)
* [Setup swarm, configure managers, add nodes, and setup backup schedule](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Setup_swarm_configure_managers_add_nodes_and_setup_backup_schedule.md)
* [Create and manager user and teams](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Create_and_manager_user_and_teams.md)
* [Interpret errors to troubleshoot installation issues without assistance](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Interpret_errors_to_troubleshoot_installation_issues_without_assistance.md)
* [Outline the sizing requirements prior to installation](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Outline_the_sizing_requirements_prior_to_installation.md)
* [Understand namespaces cgroups and configuration of certificates](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Understand_namespaces_cgroups_and_configuration_of_certificates.md)
* [Use certificate-based client-server authentication to ensure a Docker daemon has the rights to access images on a registry](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Use_certificate-based_client-server_authentication_to_ensure_a_Docker_daemon_has_the_rights_to_access_images_on_a_registry.md)
* [Consistently repeat steps to deploy Docker engine UCP and DTR on AWS and on premises in an HA config](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Consistently_repeat_steps_to_deploy_Docker_engine_UCP_and_DTR_on_AWS_and_on_premises_in_an_HA_config.md)
* [Complete configuration of backups for UCP and DTR](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Complete_configuration_of_backups_for_UCP_and_DTR.md)
* [Configure the Docker daemon to start on boot](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_3_Installation_and_Configuration/Configure_the_Docker_daemon_to_start_on_boot.md)

### Domain 4: Networking (15% of exam)

Content may include the following:

* [Create a Docker bridge network for a developer to use for their containers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Create_a_Docker_bridge_network_for_a_developer_to_use_for_their_containers.md)
* [Troubleshoot container and engine logs to understand a connectivity issue between containers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Troubleshoot_container_and_engine_logs_to_understand_a_connectivity_issue_between_containers.md)
* [Publish a port so that an application is accessible externally](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Publish_a_port_so_that_an_application_is_accessible_externally.md)
* [Identify which IP and port a container is externally accessible on](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Identify_which_IP_and_port_a_container_is_externally_accessible_on.md)
* [Describe the different types and use cases for the built-in network drivers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Describe_the_different_types_and_use_cases_for_the_built-in_network_drivers.md)
* [Understand the Container Network Model and how it interfaces with the Docker engine and network and IPAM drivers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Understand_the_Container_Network_Model_and_how_it_interfaces_with_the_Docker_engine_and_network_and_IPAM_drivers.md)
* [Configure Docker to use external DNS](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Configure_Docker_to_use_external_DNS.md)
* [Use Docker to load balance HTTP HTTPs traffic to an application](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Use_Docker_to_load_balance_HTTP_HTTPs_traffic_to_an_application.md)
* [Understand and describe the types of traffic that flow between the Docker engine registry and UCP controllers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Understand_and_describe_the_types_of_traffic_that_flow_between_the_Docker_engine_registry_and_UCP_controllers.md)
* [Deploy a service on a Docker overlay network](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Deploy_a_service_on_a_Docker_overlay_network.md)
* [Describe the difference between host and ingress port publishing mode](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_4_Networking/Describe_the_difference_between_host_and_ingress_port_publishing_mode.md)

### Domain 5: Security (15% of exam)

Content may include the following:

* [Describe the process of signing an image](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_the_process_of_signing_an_image.md)
* [Demonstrate that an image passes a security scan](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Demonstrate_that_an_image_passes_a_security_scan.md)
* [Enable Docker Content Trust](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Enable_Docker_Content_Trust.md)
* [Configure RBAC in UCP](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Configure_RBAC_in_UCP.md)
* [Integrate UCP with LDAP AD](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Integrate_UCP_with_LDAP_AD.md)
* [Demonstrate creation of UCP client bundles](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Demonstrate_creation_of_UCP_client_bundles.md)
* [Describe default engine security](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_default_engine_security.md)
* [Describe swarm default security](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_swarm_default_security.md)
* [Describe MTLS](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_MTLS.md)
* [Identity roles](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Identity_roles.md)
* [Describe the difference between UCP workers and managers](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_the_difference_between_UCP_workers_and_managers.md)
* [Describe process to use external certificates with UCP and DTR](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_5_Security/Describe_process_to_use_external_certificates_with_UCP_and_DTR.md)

### Domain 6: Storage and Volumes (10% of exam)

Content may include the following:

* [State which graph driver should be used on which OS](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/State_which_graph_driver_should_be_used_on_which_OS.md)
* [Demonstrate how to configure devicemapper](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Demonstrate_how_to_configure_devicemapper.md)
* [Compare object storage to block storage and explain which one is preferable when available](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Compare_object_storage_to_block_storage_and_explain_which_one_is_preferable_when_available.md)
* [Summarize how an application is composed of layers and where those layers reside on the filesystem](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Summarize_how_an_application_is_composed_of_layers_and_where_those_layers_reside_on_the_filesystem.md)
* [Describe how volumes are used with Docker for persistent storage](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Describe_how_volumes_are_used_with_Docker_for_persistent_storage.md)
* [Identify the steps you would take to clean up unused images on a filesystem also on DTR](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Identify_the_steps_you_would_take_to_clean_up_unused_images_on_a_filesystem_also_on_DTR.md)
* [Demonstrate how storage can be used across cluster nodes](https://github.com/DevOps-Academy-Org/dca-prep-guide/blob/master/Domain_6_Storage_and_Volumes/Demonstrate_how_storage_can_be_used_across_cluster_nodes.md)