**Table of Contents**

1. [Orchestration](https://evalle.github.io/DCA/#domain-1-orchestration-25-of-exam)
2. [Image Creation, Management, and Registry](https://evalle.github.io/DCA/#domain-2-image-creation-management-and-registry-20-of-exam)
3. [Installation and Configuration](https://evalle.github.io/DCA/#domain-3-installation-and-configuration-15-of-exam)
4. [Networking](https://evalle.github.io/DCA/#domain-4-networking-15-of-exam)
5. [Security](https://evalle.github.io/DCA/#domain-5-security-15-of-exam)
6. [Storage and Volumes](https://evalle.github.io/DCA/#domain-6-storage-and-volumes-10-of-exam)
7. [Exam Details](https://evalle.github.io/DCA/#exam-details)
8. [Useful Links](https://evalle.github.io/DCA/#useful-links)
9. [Useful Preparation Links](https://evalle.github.io/DCA/#useful-preparation-links)

**Content**

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

* [Complete the setup of a swarm mode cluster, with managers and worker nodes](https://docs.docker.com/engine/swarm/swarm-tutorial/create-swarm/)
* [Describe and demonstrate how to extend the instructions to run individual containers into running services under swarm](https://docs.docker.com/engine/swarm/swarm-tutorial/deploy-service/)
* [Describe the importance of quorum in a swarm cluster.](https://docs.docker.com/engine/swarm/raft/)
* [Describe the difference between running a container and running a service.](https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/#services-tasks-and-containers)
* [Interpret the output of “docker inspect” commands](https://docs.docker.com/engine/reference/commandline/inspect/)
* [Convert an application deployment into a stack file using a YAML compose file with “docker stack deploy”](https://docs.docker.com/engine/swarm/stack-deploy/)
* [Manipulate a running stack of services](https://docs.docker.com/engine/reference/commandline/stack_services/#related-commands)
* [Describe and demonstrate orchestration activities](https://docs.docker.com/get-started/orchestration/)
* [Increase number of replicas](https://docs.docker.com/engine/reference/commandline/service_scale/)
* [Add networks, publish ports](https://docs.docker.com/network/)
* [Mount volumes](https://docs.docker.com/storage/volumes/)
* [Describe and demonstrate how to run replicated and global services](https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/#replicated-and-global-services)
* Apply node labels to demonstrate placement of tasks ([Label a Swarm Node](https://docs.docker.com/engine/reference/commandline/node_update/#add-label-metadata-to-a-node), [Use Placement Constraints](https://docs.docker.com/engine/swarm/services/#control-service-placement))
* [Describe and demonstrate how to use templates with “docker service create”](https://docs.docker.com/engine/reference/commandline/service_create/#create-services-using-templates)
* [Identify the steps needed to troubleshoot a service not deploying](https://docker-tutorial.schoolofdevops.com/docker-swarm-ucp-troubleshooting/)
* [Describe how a Dockerized application communicates with legacy systems](https://docs.docker.com/config/containers/container-networking/)
* [Describe how to deploy containerized workloads as Kubernetes pods and deployments](https://docs.docker.com/get-started/kube-deploy/)
* [Describe how to provide configuration to Kubernetes pods using configMaps and secrets](https://opensource.com/article/19/6/introduction-kubernetes-secrets-and-configmaps)

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

* [Describe the use of Dockerfile](https://docs.docker.com/engine/reference/builder/)
* [Describe options, such as add, copy, volumes, expose, entry point](https://docs.docker.com/engine/reference/builder/#from)
* [Identify and display the main parts of a Dockerfile](https://docs.docker.com/engine/reference/builder/#dockerfile-examples)
* [Describe and demonstrate how to create an efficient image via a Dockerfile](https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/)
* [Describe and demonstrate how to use CLI commands to manage images, such as list, delete, prune, rmi](https://docs.docker.com/engine/reference/commandline/image/#usage)
* [Describe and demonstrate how to inspect images and report specific attributes using filter and format](https://docs.docker.com/engine/reference/commandline/images/#filtering)
* [Describe and demonstrate how to tag an image.](https://docs.docker.com/engine/reference/commandline/tag/)
* [Describe and demonstrate how to apply a file to create a Docker image](https://docs.docker.com/engine/reference/commandline/image_load/)
* [Describe and demonstrate how to display layers of a Docker image](https://docs.docker.com/engine/reference/commandline/image_inspect/)
* Describe and demonstrate how to modify an image to a single layer ([multi-stage build](https://docs.docker.com/develop/develop-images/dockerfile_best-practices/#minimize-the-number-of-layers), [single layer](https://stackoverflow.com/questions/39695031/how-make-docker-layer-to-single-layer))
* [Describe and demonstrate registry functions](https://docs.docker.com/registry/)
* [Deploy a registry](https://docs.docker.com/registry/deploying/)
* [Log into a registry](https://docs.docker.com/engine/reference/commandline/login/)
* [Utilize search in a registry](https://docs.docker.com/engine/reference/commandline/search/)
* [Push an image to a registry](https://docs.docker.com/engine/reference/commandline/push/)
* [Sign an image in a registry](https://docs.docker.com/engine/reference/commandline/trust_sign/)
* [Pull](https://docs.docker.com/engine/reference/commandline/pull/) and [delete](https://docs.docker.com/registry/spec/api/#deleting-an-image) images from a registry

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

* [Describe sizing requirements for installation](https://docs.docker.com/docker-for-windows/install/#system-requirements)
* [Describe and demonstrate the setup of repo, selection of a storage driver, and installation of the Docker engine on multiple platforms](https://docs.docker.com/engine/install/)
* [Describe and demonstrate configuration of logging drivers (splunk, journald, etc.)](https://docs.docker.com/config/containers/logging/configure/)
* [Describe and demonstrate how to set up swarm, configure managers, add nodes, and setup the backup schedule](https://docs.docker.com/engine/swarm/admin_guide/)
* [Describe and demonstrate how to create and manage user and teams](https://docs.mirantis.com/msr/2.9/ops/manage-users.html#create-and-manage-teams)
* [Describe and demonstrate how to configure the Docker daemon to start on boot](https://docs.docker.com/install/linux/linux-postinstall/)
* [Describe and demonstrate how to use certificate-based client-server authentication to ensure a Docker daemon has the rights to access images on a registry](https://docs.docker.com/engine/security/certificates/)
* [Describe the use of namespaces, cgroups, and certificate configuration](https://docs.docker.com/get-started/overview/#the-underlying-technology)
* [Describe and interpret errors to troubleshoot installation issues without assistance](https://docs.docker.com/config/daemon/#troubleshoot-the-daemon)
* Describe and demonstrate the steps to deploy the docker engine in [Docker,](https://docs.docker.com/install/linux/docker-ce/ubuntu/)[DTR,](https://docs.mirantis.com/msr/2.9/install/install-online.html)[UCP,](https://docs.mirantis.com/mke/3.5/install/install-mke-image.html) [Docker on AWS](https://aws.amazon.com/getting-started/hands-on/deploy-docker-containers/) and possibly [on premises HA config](https://docs.docker.com/engine/swarm/admin_guide/#add-manager-nodes-for-fault-tolerance)
* Describe and demonstrate how to configure backups for UCP and DTR:
  + [UCP](https://docs.mirantis.com/mke/3.4/ops/disaster-recovery.html)
  + [DTR](https://docs.mirantis.com/msr/2.9/ops/disaster-recovery.html)

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

* [Describe the Container Network Model and how it interfaces with the Docker engine and network and IPAM drivers](https://docs.docker.com/network/)
* [Describe the different types and use cases for the built-in network drivers](https://blog.docker.com/2016/12/understanding-docker-networking-drivers-use-cases/)
* [Describe the types of traffic that flow between the Docker engine, registry and UCP controllers](https://docs.mirantis.com/msr/2.9/ref-arch/networks.html)
* [Describe and demonstrate how to create a Docker bridge network for developers to use for their containers](https://docs.docker.com/network/network-tutorial-standalone/)
* [Describe and demonstrate how to publish a port so that an application is accessible externally](https://github.com/wsargent/docker-cheat-sheet#exposing-ports)
* [Identify which IP and port a container is externally accessible on](https://docs.docker.com/engine/reference/commandline/port/#examples)
* Compare and contrast “host” and “ingress” publishing modes ([Host](https://docs.docker.com/engine/swarm/services/#publish-a-services-ports-directly-on-the-swarm-node), [Ingress](https://docs.docker.com/engine/swarm/ingress/))
* [Describe and demonstrate how to configure Docker to use external DNS](https://gist.github.com/Evalle/7b21e0357c137875a03480428a7d6bf6)
* [Describe and demonstrate how to use Docker to load balance HTTP/HTTPs traffic to an application (Configure L7 load balancing with Docker EE)](https://www.docker.com/blog/enhanced-layer-7-routing-swarm-docker-enterprise-edition-beta/)
* [Describe and demonstrate how to deploy a service on a Docker overlay network](https://docs.docker.com/network/overlay/)
* [Describe and demonstrate how to troubleshoot container and engine logs to resolve connectivity issues between containers](https://success.mirantis.com/article/troubleshooting-container-networking)
* (Since Study Guide 1.5) [Describe how to route traffic to Kubernetes pods using ClusterIP and NodePort services](https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types)
* (Since Study Guide 1.5) [Describe the Kubernetes’ container network model](https://kubernetes.io/docs/concepts/cluster-administration/networking/)

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

* (Since Study Guide 1.5) Describe [security administration](https://docs.docker.com/engine/security/) and [tasks](https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/#tasks-and-scheduling)
* [Describe the process of signing an image](https://docs.docker.com/engine/security/trust/content_trust/#push-trusted-content)
* [Describe default engine security](https://docs.docker.com/engine/security/security/)
* [Describe swarm default security](https://docs.docker.com/engine/swarm/how-swarm-mode-works/pki/)
* [Describe MTLS](https://diogomonica.com/2017/01/11/hitless-tls-certificate-rotation-in-go/)
* [Identity roles](https://docs.mirantis.com/mke/3.6/ref-arch/rbac.html#rbac)
* Describe the difference between UCP workers and managers:
  + [workers](https://docs.mirantis.com/mke/3.4/ref-arch/worker-nodes.html)
  + [managers](https://docs.mirantis.com/mke/3.4/ref-arch/manager-nodes.html)
* Describe process to use external certificates with:
  + UCP
    - [from cli](https://docs.mirantis.com/mke/3.6/ops/administer-cluster/use-your-own-tls-certificates.html)
    - [from GUI](https://docs.mirantis.com/docker-enterprise/v3.0/dockeree-products/ucp/admin/configure/use-your-own-tls-certificates.html)
    - [print the public certificates](https://docs.mirantis.com/mke/3.3/cli-ref/mke-cli-dump-certs.html)
  + [**DTR** is now Mirantis Secure Registry or **MSR**](https://docs.mirantis.com/containers/v3.1/dockeree-products/msr/msr-configure/use-your-own-tls-certificates.html)
* [Describe and demonstrate that an image passes a security scan](https://docs.mirantis.com/msr/2.9/ops/manage-images/scan-images-for-vulnerabilities.html)
* [Describe and demonstrate how to enable Docker Content Trust.](https://docs.docker.com/engine/security/trust/content_trust/)
* [Describe and demonstrate how to configure RBAC with UCP](https://docs.mirantis.com/containers/v3.0/dockeree-products/mke/admin/configure/configure-kube-rbac.html)
* [Describe and demonstrate how to integrate UCP with LDAP/AD](https://docs.mirantis.com/containers/v3.0/dockeree-products/mke/admin/configure/integrate-with-LDAP-directory.html)
* [Describe and demonstrate how to create UCP client bundles](https://blog.docker.com/2017/09/get-familiar-docker-enterprise-edition-client-bundles/)

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

* [Identify the correct graph drivers to uses with various operating systems](https://docs.docker.com/storage/storagedriver/select-storage-driver/)
* [Describe and demonstrate how to configure devicemapper](https://docs.docker.com/storage/storagedriver/device-mapper-driver/#configure-docker-with-the-devicemapper-storage-driver)
* [Compare and contrast object and block storage and when they should be used](https://rancher.com/block-object-file-storage-containers/)
* [Describe how an application is composed of layers and where these layers reside on the filesystem](https://docs.docker.com/storage/storagedriver/#images-and-layers)
* [Describe how volumes are used with Docker for persistent storage](https://docs.docker.com/storage/volumes/)
* Identify the steps you would take to clean up unused images on a filesystem, also on DTR. ([image prune](https://docs.docker.com/engine/reference/commandline/image_prune/), [system prune](https://docs.docker.com/engine/reference/commandline/system_prune/) and [from DTR](https://docs.mirantis.com/msr/2.9/ops/manage-images/delete-images.html))
* [Demonstrate how storage can be used across cluster nodes](https://docs.docker.com/engine/extend/legacy_plugins/#volume-plugins), [ex.](https://www.digitalocean.com/community/questions/how-to-attach-digitalocean-block-storage-to-docker-container)
* (Since Study Guide 1.5) [Describe how to provision persistent storage to a Kubernetes pod using persistentVolumes](https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/)
* (Since Study Guide 1.5) Describe the relationship between [container storage interface drivers](https://kubernetes.io/blog/2019/01/15/container-storage-interface-ga/), [storageClass](https://kubernetes.io/docs/concepts/storage/storage-classes/), [persistentVolumeClaim](https://kubernetes.io/docs/concepts/storage/persistent-volumes/) and [volume objects](https://kubernetes.io/docs/concepts/storage/persistent-volumes/#storage-object-in-use-protection) in Kubernetes