Docker Certified Associate Study Guide

Version 1.4
August 2020

Introduction

This examination is based upon critical job activities a *Docker Certified Associate* performs. The skills and knowledge certified by this examination represent a level of expertise where a certified Docker Associate can:

- Run containerized applications from pre-existing images stored in a centralized registry
- Deploy images across the cluster
- Triage and resolve issue reports from stakeholders and resolve
- Standup up on Enterprise clusters with one UCP manager, one DTR replica, and one worker node
- Migrate traditional applications to containers
- Configure and troubleshoot Docker engine
- Perform general maintenance and configuration

Candidates for this certification should have at least six months to one year of experience with Docker, including exposure to the Docker Enterprise Edition. The knowledge, skills and experience required at this level should also include:

- container security
- experience with at least one cloud provider
- configuration management tools
- Linux and/or Windows Server

The skills and knowledge this examination measures are derived from an understanding of the jobs of current Docker users. A team of highly qualified Docker experts defined the test content and developed the test items.

Note: This examination blueprint includes weighting, test objectives, and example content. Example topics and concepts are included to clarify the test objectives; they should not be construed as a comprehensive listing of all the content of this examination.

The following table lists the domains measured by this examination and the extent to which they are represented.

Name of Domain	% of Exam
Orchestration	25%
Image Creation, Management, and Registry	20%
Installation and Configuration	15%
Networking	15%
Security	15%
Storage and Volumes	10%

Examination Format

The examination contains 55 questions consisting of 13 multiple choice items and 42 discrete option multiple choice (DOMC) items.

- **Multiple-choice:** Examinees are presented a question with at least four response options and must select one or more options to best answer the question. Distracters or wrong answers will be presented as response options. These are response options that examinees with incomplete knowledge or skill might choose, as they are generally plausible responses fitting into the content area defined by the test objective.

- **Discrete Option Multiple Choice (DOMC):** DOMC items represent a relatively simple, but useful change in the delivery of multiple-choice questions. Options are randomly presented, one at a time. For each presented option, the examinee chooses YES or NO to indicate if the option is correct. To familiarize yourself with DOMC, try out this practice exam.

Content Limits

Domain 1: Orchestration (25% of exam)

Content may include the following:

- Complete the setup of a swarm mode cluster, with managers and worker nodes
- Describe and demonstrate how to extend the instructions to run individual containers into running services under swarm.
- Describe the importance of quorum in a swarm cluster.
- Describe the difference between running a container and running a service.
- Interpret the output of “docker inspect” commands.
- Convert an application deployment into a stack file using a YAML compose file with "docker stack deploy"
- Manipulate a running stack of services.
- Describe and demonstrate orchestration activities.
- Increase the number of replicas.
- Add networks, publish ports.
- Mount volumes.
- Describe and demonstrate how to run replicated and global services.
- Apply node labels to demonstrate placement of tasks.
- Describe and demonstrate how to use templates with “docker service create”.
- Identify the steps needed to troubleshoot a service not deploying.
- Describe how a Dockerized application communicates with legacy systems.
- Describe how to deploy containerized workloads as Kubernetes pods and deployments.
- Describe how to provide configuration to Kubernetes pods using configMaps and secrets.

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

Content may include the following:

- Describe the use of Dockerfile.
- Describe options, such as add, copy, volumes, expose, entry point.
- Identify and display the main parts of a Dockerfile.
- Describe and demonstrate how to create an efficient image via a Dockerfile.
- Describe and demonstrate how to use CLI commands to manage images, such as list, delete, prune, rmi.

- Describe and demonstrate how to inspect images and report specific attributes using filter and format
- Describe and demonstrate how to tag an image.
- Describe and demonstrate how to apply a file to create a Docker image.
- Describe and demonstrate how to display layers of a Docker image
- Describe and demonstrate how to modify an image to a single layer.
- Describe and demonstrate registry functions.
- Deploy a registry.
- Log into a registry.
- Utilize search in a registry.
- Push an image to a registry.
- Sign an image in a registry.
- Pull and delete images from a registry.

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

Content may include the following:

- Describe sizing requirements for installation.
- Describe and demonstrate the setup of repo, selection of a storage driver, and installation of the Docker engine on multiple platforms.
- Describe and demonstrate configuration of logging drivers (splunk, journald, etc.).
- Describe and demonstrate how to set up swarm, configure managers, add nodes, and setup the backup schedule.
- Describe and demonstrate how to create and manage user and teams.
- Describe and demonstrate how to configure the Docker daemon to start on boot.
- 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.
- Describe the use of namespaces, cgroups, and certificate configuration.
- Describe and interpret errors to troubleshoot installation issues without assistance.
- Describe and demonstrate the steps to deploy the Docker engine, UCP, and DTR on AWS and on-premises in an HA configuration.
- Describe and demonstrate how to configure backups for UCP and DTR.

Domain 4: Networking (15% of exam)

Content may include the following:

- Describe the Container Network Model and how it interfaces with the Docker engine and network and IPAM drivers.
- Describe the different types and use cases for the built-in network drivers.
- Describe the types of traffic that flow between the Docker engine, registry and UCP controllers.
- Describe and demonstrate how to create a Docker bridge network for developers to use for their containers.
- Describe and demonstrate how to publish a port so that an application is accessible externally.
- Identify which IP and port a container is externally accessible on.
- Compare and contrast “host” and “ingress” publishing modes.
- Describe and demonstrate how to configure Docker to use external DNS.
- Describe and demonstrate how to use Docker to load balance HTTP/HTTPs traffic to an application (Configure L7 load balancing with Docker EE).
- Understand and describe the types of traffic that flow between the Docker engine, registry, and UCP controllers
- Describe and demonstrate how to deploy a service on a Docker overlay network.
- Describe and demonstrate how to troubleshoot container and engine logs to resolve connectivity issues between containers.
- Describe how to route traffic to Kubernetes pods using ClusterIP and NodePort services.
- Describe the Kubertnetes’ container network model.

Domain 5: Security (15% of exam)

Content may include the following:

- Describe security administration and tasks.
- Describe the process of signing an image.
- Describe default engine security.
- Describe swarm default security.
- Describe MTLS.
- Describe identity roles.
- Compare and contrast UCP workers and managers.
- Describe the process to use external certificates with UCP and DTR.
- Describe and demonstrate that an image passes a security scan.

- Describe and demonstrate how to enable Docker Content Trust.
- Describe and demonstrate how to configure RBAC with UCP.
- Describe and demonstrate how to integrate UCP with LDAP/AD.
- Describe and demonstrate how to create UCP client bundles.

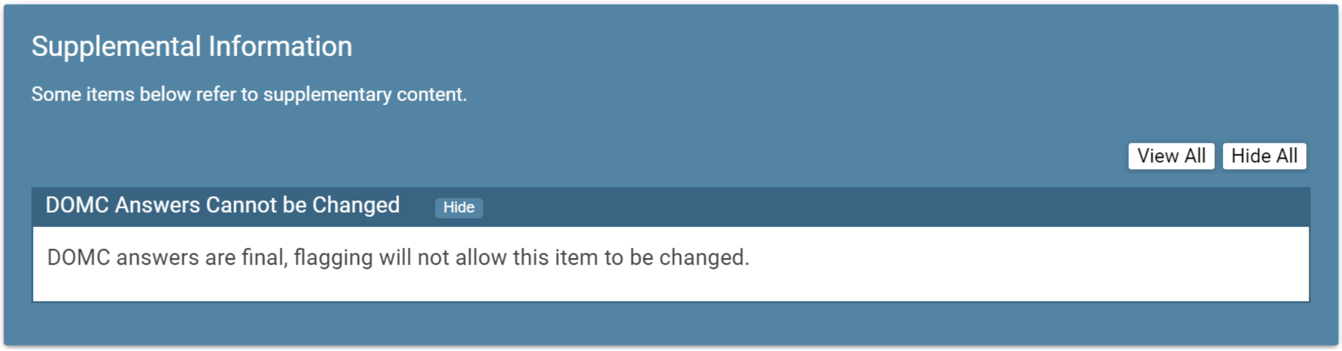
Domain 6: Storage and Volumes (10% of exam)

Content may include the following:

- Identify the correct graph drivers to uses with various operating systems.
- Describe and demonstrate how to configure devicemapper.
- Compare and contrast object and block storage and when they should be used.
- Describe how an application is composed of layers and where these layers reside on the filesystem.
- Describe the use of volumes are used with Docker for persistent storage.
- Identify the steps to take to clean up unused images on a filesystem and DTR.
- Describe and demonstrate how storage can be used across cluster nodes.
- Describe how to provision persistent storage to a Kubernetes pod using persistentVolumes.
- Describe the relationship between container storage interface drivers, storageClass, persistentVolumeClaim and volume objects in Kubernetes.

Sample Test Questions

The following supplemental information is displayed above each DOMC item on the live exam:

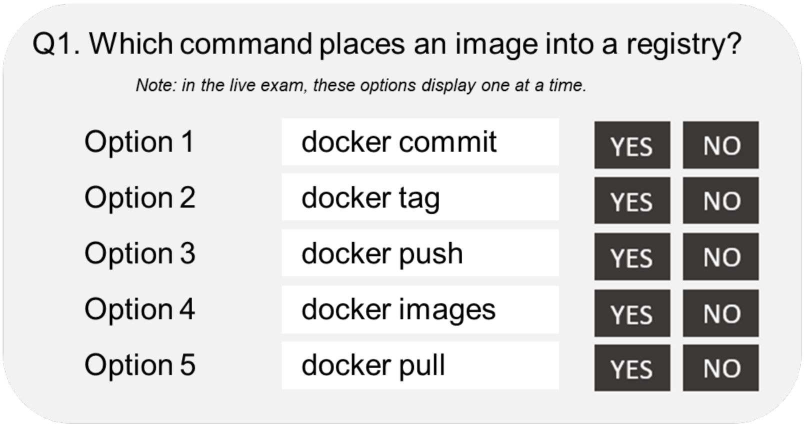


Supplemental Information

Some items below refer to supplementary content.

DOMC Answers Cannot be Changed

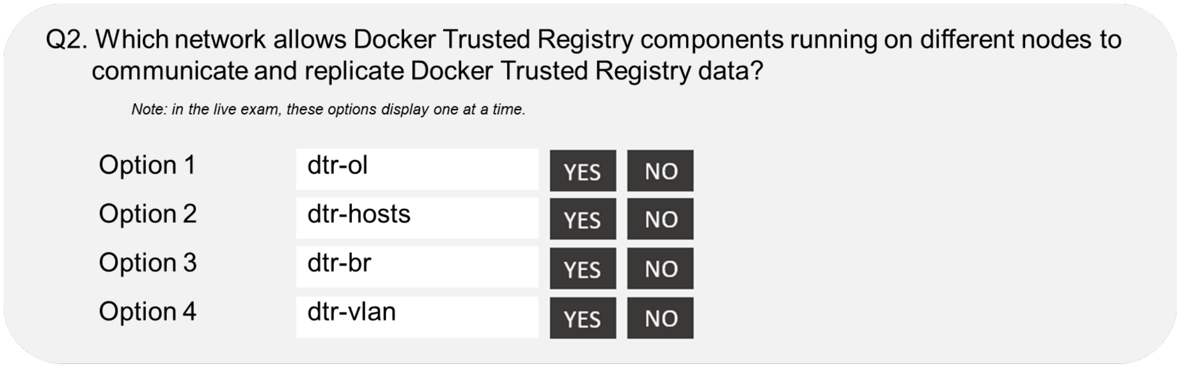
DOMC answers are final, flagging will not allow this item to be changed.



Q1. Which command places an image into a registry?

Note: in the live exam, these options display one at a time.

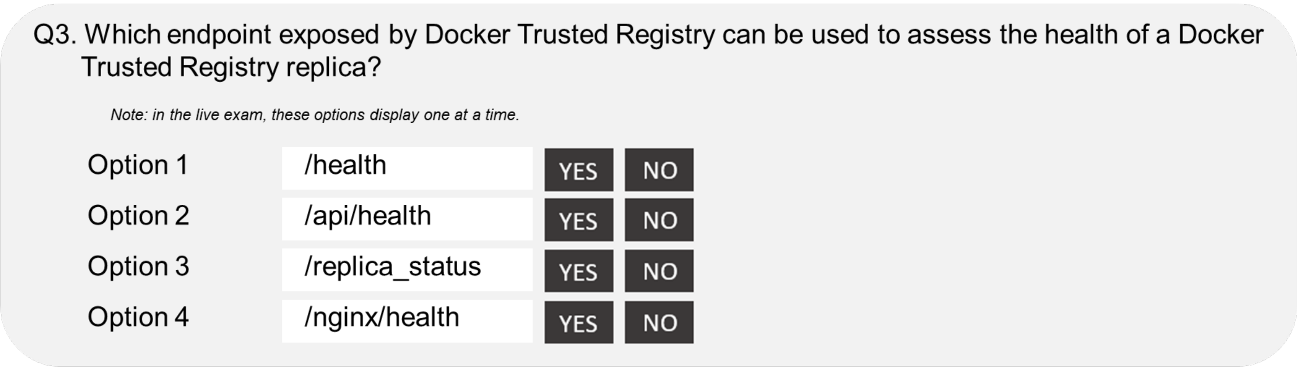
Option 1	docker commit	YES	NO
Option 2	docker tag	YES	NO
Option 3	docker push	YES	NO
Option 4	docker images	YES	NO
Option 5	docker pull	YES	NO



Q2. Which network allows Docker Trusted Registry components running on different nodes to communicate and replicate Docker Trusted Registry data?

Note: in the live exam, these options display one at a time.

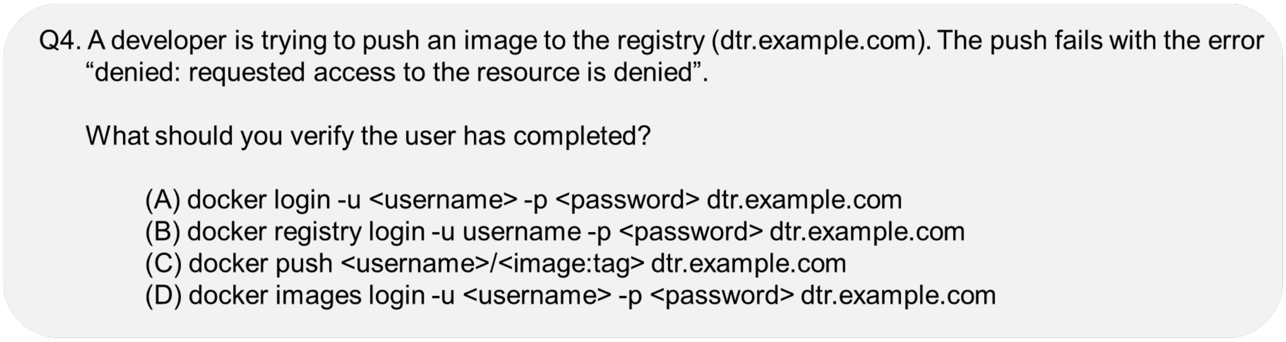
Option 1	dtr-ol	YES	NO
Option 2	dtr-hosts	YES	NO
Option 3	dtr-br	YES	NO
Option 4	dtr-vlan	YES	NO



Q3. Which endpoint exposed by Docker Trusted Registry can be used to assess the health of a Docker Trusted Registry replica?

Note: in the live exam, these options display one at a time.

Option 1	/health	YES	NO
Option 2	/api/health	YES	NO
Option 3	/replica_status	YES	NO
Option 4	/nginx/health	YES	NO



Q4. A developer is trying to push an image to the registry (dtr.example.com). The push fails with the error “denied: requested access to the resource is denied”.

What should you verify the user has completed?

(A) docker login -u <username> -p <password> dtr.example.com
(B) docker registry login -u username -p <password> dtr.example.com
(C) docker push <username>/<image:tag> dtr.example.com
(D) docker images login -u <username> -p <password> dtr.example.com

Q5. You have been asked to back up the swarm state on a Linux installation.

By default, where do Docker manager nodes store the swarm state and manager logs?

(A) /var/run/docker/swarm
(B) /var/lib/docker/swarm
(C) /etc/docker/swarm
(D) /run/docker/swarm

Q6. How do you deploy 4 new instances of nginx with a single command?

(A) docker service create --replicas 4 --name myservice nginx
(B) docker service create --instances 4 --name myservice nginx
(C) docker service scale myservice=4 nginx
(D) docker service scale --replicas 4 --name myservice nginx

Q7. You are using self-signed UCP certs and have a second DNS name that points to your internal controllers.

When installing UCP, which flag should you use to add this additional name?

Note: in the live exam, these options display one at a time.

Option 1	--internal-server-cert	YES	NO
Option 2	--dns	YES	NO
Option 3	--san	YES	NO
Option 4	--external-server-cert	YES	NO

Q8. The Kubernetes yaml shown below describes a clusterIP service:

```
yaml
  apiVersion: v1
  kind: Service
  metadata:
    name: dca
  spec:
    type: clusterIP
    selector:
        app: nginx
    ports:
    - port: 8080
      targetPort: 80
    - port: 4443
      targetPort: 443
```

What port on pods matching this service's selector will receive traffic sent to the service on port 8080?

(A) 8080/tcp
(B) 8080/udp
(C) 80/tcp
(D) 80/udp

Q9. A node has 10 cores and 4 GB of memory available for scheduling Kubernetes workloads. Under the following specification, what is the maximum number of pods that can be scheduled on this host?

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox3
spec:
  containers:
  - name: dca
    image: busybox
    resources:
      limits:
        memory: "500M"
        cpu: "2"
      requests:
        memory: "250M"
        cpu: "1"
```

(A) 4
(B) 16
(C) 10
(D) 8

Answer Key

Q1 = Option 3

Q2 = Option 1

Q3 = Option 1

Q4 = A

Q5 = B

Q6 = A

Q7 = Option 3

Q8 = C

Q9 = C

Notes

success.docker.com/certification

Questions? Email certification@docker.com