Vendor: Docker

Exam Code: DCA

Exam Name: Docker Certified Associate

Version: DEMO

QUESTION 1
You configure a local Docker engine to enforce content trust by setting the environment variable

DOCKER_CONTENT_TRUST=1.

If myorg/myimage: 1.0 is unsigned, does Docker block this command?

Solution: docker service create myorg/myimage:1.0

A. Yes
B. No

Answer: A
Explanation:
Docker will block this command, because docker service create requires content trust to be enabled by default and will not pull or run unsigned images. According to the official documentation, docker service create is one of the commands that verify content trust by default and will fail if the image is not signed or if the signing key is not trusted.
Reference:
https://docs.docker.com/engine/security/trust/#using-docker-content-trust-in-docker-engine
https://docs.docker.com/engine/reference/commandline/service_create/#extended-description

QUESTION 2
Can this set of commands identify the published port(s) for a container?

Solution: docker container inspect', 'docker port'

A. Yes
B. No

Answer: A
Explanation:
This set of commands can identify the published port(s) for a container, because docker container inspect can show the port mappings for a container and docker port can list the public port that is NAT-ed to the private port. According to the official documentation, these commands can be used to find the exposed ports for a container.
Reference:
https://docs.docker.com/engine/reference/commandline/container_inspect/
https://docs.docker.com/engine/reference/commandline/port/

QUESTION 3
You add a new user to the engineering organization in DTR.

Will this action grant them read/write access to the engineering/api repository?

Solution: Add the user directly to the list of users with read/write access under the repository's Permissions tab.

A. Yes
B. No

Answer: A
Explanation:

This action will grant them read/write access to the engineering/api repository, because adding a user directly to the list of users with read/write access under the repository's Permissions tab is one way to grant permissions to a user in DTR. According to the official documentation, this is an example of using fine-grained permissions for a repository.
Reference:
https://docs.docker.com/ee/dtr/user/manage-repos/permission-levels/

QUESTION 4
You add a new user to the engineering organization in DTR.

Will this action grant them read/write access to the engineering/api repository?

Solution: Add them to a team in the engineering organization that has read/write access to the engineering/api repository.

A. Yes
B. No

Answer: A
Explanation:
This action will grant them read/write access to the engineering/api repository, because adding them to a team in the engineering organization that has read/write access to the engineering/api repository is another way to grant permissions to a user in DTR. According to the official documentation, this is an example of using teams and organizations for managing permissions.
Reference:
https://docs.docker.com/ee/dtr/user/manage-repos/permission-levels/

QUESTION 5
Two development teams in your organization use Kubernetes and want to deploy their applications while ensuring that Kubernetes-specific resources, such as secrets, are grouped together for each application.

Is this a way to accomplish this?

Solution: Create one pod and add all the resources needed for each application

A. Yes
B. No

Answer: B
Explanation:
his is not a way to accomplish this, because creating one pod and adding all the resources needed for each application is not a good practice for deploying applications in Kubernetes. According to the official documentation, pods are not intended to run multiple instances of an application or different applications that are tightly coupled. Pods are also not meant to hold resources that are shared across applications, such as secrets or configMaps.
Reference:
https://kubernetes.io/docs/concepts/workloads/pods/#what-is-a-pod

QUESTION 6
Two development teams in your organization use Kubernetes and want to deploy their applications while ensuring that Kubernetes-specific resources, such as secrets, are grouped

together for each application.

Is this a way to accomplish this?

Solution: Add all the resources to the default namespace.

A. Yes
B. No

Answer: B
Explanation:
This is not a way to accomplish this, because adding all the resources to the default namespace is not a good practice for isolating applications in Kubernetes. According to the official documentation, namespaces are used to group resources into logical units that correspond to different projects, teams, or environments. Using namespaces can help avoid naming collisions and enforce resource quotas and access policies.
Reference:
https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

QUESTION 7
Two development teams in your organization use Kubernetes and want to deploy their applications while ensuring that Kubernetes-specific resources, such as secrets, are grouped together for each application.

Is this a way to accomplish this?

Solution: Create one namespace for each application and add all the resources to it.

A. Yes
B. No

Answer: A
Explanation:
This is a way to accomplish this, because creating one namespace for each application and adding all the resources to it is a good practice for deploying applications in Kubernetes. According to the official documentation, namespaces are used to group resources into logical units that correspond to different projects, teams, or environments. Using namespaces can help avoid naming collisions and enforce resource quotas and access policies.
Reference:
https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

QUESTION 8
Seven managers are in a swarm cluster.

Is this how should they be distributed across three datacenters or availability zones?

Solution: 3-3-1

A. Yes
B. No

Answer: B
Explanation:

This is not how they should be distributed across three datacenters or availability zones, because having one manager in one datacenter or availability zone creates a single point of failure and reduces the fault tolerance of the swarm. According to the official documentation, managers should be distributed evenly across datacenters or availability zones to ensure that the swarm can survive the loss of any one datacenter or availability zone.
Reference:
https://docs.docker.com/engine/swarm/admin_guide/#add-manager-nodes-for-fault-tolerance

QUESTION 9
Seven managers are in a swarm cluster.

Is this how should they be distributed across three datacenters or availability zones?

Solution: 5-1-1

A. Yes
B. No

Answer: B
Explanation:
This is not how they should be distributed across three datacenters or availability zones, because having one manager in two datacenters or availability zones creates a risk of losing quorum if those datacenters or availability zones become unavailable. According to the official documentation, managers should be distributed evenly across datacenters or availability zones to ensure that the swarm can survive the loss of any one datacenter or availability zone.
Reference:
https://docs.docker.com/engine/swarm/admin_guide/#add-manager-nodes-for-fault-tolerance

QUESTION 10
Seven managers are in a swarm cluster.

Is this how should they be distributed across three datacenters or availability zones?

Solution: 3-2-2

A. Yes
B. No

Answer: A
Explanation:
This is how they should be distributed across three datacenters or availability zones, because having an even distribution of managers across datacenters or availability zones ensures that the swarm can survive the loss of any one datacenter or availability zone and maintain quorum. According to the official documentation, managers should be distributed evenly across datacenters or availability zones to ensure that the swarm can survive the loss of any one datacenter or availability zone.
Reference:
https://docs.docker.com/engine/swarm/admin_guide/#add-manager-nodes-for-fault-tolerance

QUESTION 11
Does this command create a swarm service that only listens on port 53 using the UDP protocol?

Solution: 'docker service create --name dns-cache -p 53:53/udp dns-cache'

A. Yes
B. No

Answer: A
Explanation:
This command creates a swarm service that only listens on port 53 using the UDP protocol, because the -p flag specifies the published port and target port separated by a colon (:) and followed by the protocol name (/udp). According to the official documentation, this is an example of using the short version of the publish option to expose a port for a service.

QUESTION 12
Does this command create a swarm service that only listens on port 53 using the UDP protocol?

Solution: 'docker service create -name dns-cache -p 53:53 -service udp dns-cache'

A. Yes
B. No

Answer: B
Explanation:
This command does not create a swarm service that only listens on port 53 using the UDP protocol, because it has several syntax errors and missing options. According to the official documentation, the correct command should be:

docker service create --name dns-cache -p 53:53/udp --mode global dns-cache

The errors and missing options are:

The --name flag is missing a dash (-).
The -p flag is missing the /udp suffix to specify the protocol.
The --service flag does not exist and should be replaced by --mode to specify the service mode.

The global option should follow the --mode flag to indicate that one task should run on each node.

QUESTION 13
You want to provide a configuration file to a container at runtime. Does this set of Kubernetes tools and steps accomplish this?

Solution: Mount the configuration file directly into the appropriate pod and container using the .spec.containers.configMounts key.

A. Yes
B. No

Answer: B
Explanation:
This set of Kubernetes tools and steps does not accomplish this, because there is no such key as .spec.containers.configMounts in the pod specification. According to the official documentation, the correct key to use for mounting a configuration file directly into a container is .spec.containers.volumeMounts, which requires a corresponding volume definition in .spec.volumes.

Reference:
https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#add-configmap-data-to-a-volume
https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.22/#container-v1-core

QUESTION 14
You want to provide a configuration file to a container at runtime. Does this set of Kubernetes tools and steps accomplish this?

Solution: Turn the configuration file into a configMap object, use it to populate a volume associated with the pod, and mount that file from the volume to the appropriate container and path.

A. Yes
B. No

Answer: A
Explanation:
This set of Kubernetes tools and steps accomplishes this, because it follows the recommended way of providing a configuration file to a container at runtime using a configMap object. According to the official documentation, this method allows you to decouple configuration artifacts from image content and update them without restarting containers.
Reference:
https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#add-configmap-data-to-a-volume

QUESTION 15
In Docker Trusted Registry, is this how a user can prevent an image, such as 'nginx:latest', from being overwritten by another user with push access to the repository?

Solution: Use the DTR web UI to make all tags in the repository immutable.

A. Yes
B. No

Answer: B
Explanation:
This is not how a user can prevent an image from being overwritten by another user with push access to the repository, because making all tags in the repository immutable will also prevent the user from updating their own image tags. According to the official documentation, a better way to prevent an image from being overwritten is to use promotion policies that only allow certain users or teams to push images with specific tags.
Reference:
https://docs.docker.com/ee/dtr/user/promotion-policies/overview/

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

★ More than **99,900** Satisfied Customers Worldwide.

★ Average **99.9%** Success Rate.

★ **Free Update** to match latest and real exam scenarios.

★ **Instant Download** Access! No Setup required.

★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.

★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.

★ **100%** Guaranteed Success or **100%** Money Back Guarantee.

★ **Fast**, helpful support **24x7**.

View list of all certification exams: http://www.lead2pass.com/all-products.html

10% Discount Coupon Code: **ASTR14**