

DUMPSBOSS.

Oracle Cloud Infrastructure 2022 Developer Professional

Oracle 1z0-1084-22

Version Demo

Total Demo Questions: 10

Total Premium Questions: 74

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

You are developing a serverless application with Oracle Functions. You have created a function in compartment named prod. When you try to invoke your function you get the following error.

Error invoking function. status: 502 message: dhcp options ocid1.dhcpoptions.oc1.phx.aaaaaaac... does not exist or Oracle Functions is not authorized to use it

How can you resolve this error?

A. Create a policy:

Allow function-family to use virtual-network-family in compartment prod

B. Create a policy:

Allow any-user to manage function-family and virtual-network-family in compartment prod

C. Create a policy:

Allow service FaaS to use virtual-network-family in compartment prod

D. Deleting the function and redeploying it will fix the problem

ANSWER: C

Explanation:

Troubleshooting Oracle Functions:

There are common issues related to Oracle Functions and how you can address them.

Invoking a function returns a FunctionInvokeSubnetNotAvailable message and a 502 error (due to a DHCP Options issue)

When you invoke a function that you've deployed to Oracle Functions, you might see the following error message:

```
{"code": "FunctionInvokeSubnetNotAvailable", "message": "dhcp options ocid1.dhcpoptions..... does not exist or Oracle Functions is not authorized to use it"}
```

Fn: Error invoking function. status: 502 message: dhcp options ocid1.dhcpoptions..... does not exist or Oracle Functions is not authorized to use it

If you see this error:

Double-check that a policy has been created to give Oracle Functions access to network resources.

Create Policies to Control Access to Network and Function-Related Resources:

Service Access to Network Resources

When Oracle Functions users create a function or application, they have to specify a VCN and a subnet in which to create them. To enable the Oracle Functions service to create the function or application in the specified VCN and subnet, you must create an identity policy to grant the Oracle Functions service access to the compartment to which the network resources belong.

To create a policy to give the Oracle Functions service access to network resources:

Log in to the Console as a tenancy administrator.

Create a new policy in the root compartment:

Open the navigation menu. Under Governance and Administration, go to Identity and click Policies.

Follow the instructions in [To create a policy](#), and give the policy a name (for example, functions-service-network-access).

Specify a policy statement to give the Oracle Functions service access to the network resources in the compartment:

Allow service FaaS to use virtual-network-family in compartment

For example:

Allow service FaaS to use virtual-network-family in compartment acme-network

Click Create.

Double-check that the set of DHCP Options in the VCN specified for the application still exists.

References:

<https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Tasks/functionstroubleshooting.htm>

<https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Tasks/functionscreatingpolicies.htm>

QUESTION NO: 2

You want to push a new image in the Oracle Cloud Infrastructure (OCI) Registry. Which two actions do you need to perform?

- A. Assign a tag via Docker CLI to the image.
- B. Generate an auth token to complete the authentication via Docker CLI.
- C. Generate an API signing key to complete the authentication via Docker CLI.
- D. Assign an OCI defined tag via OCI CLI to the image.
- E. Generate an OCI tag namespace in your repository.

ANSWER: A B

Explanation:

Pushing Images Using the Docker CLI:

You use the Docker CLI to push images to Oracle Cloud Infrastructure Registry.

To push an image, you first use the `docker tag` command to create a copy of the local source image as a new image (the new image is actually just a reference to the existing source image). As a name for the new image, you specify the fully qualified path to the target location in Oracle Cloud Registry where you want to push the image, optionally including the name of a repository.

For example, assume you have a local image named `acme-web-app:latest`. Let's say you want to push this image to Oracle Cloud Infrastructure Registry with a name of `acme-web-app:version2.0.test` into a repository called `project01` in the Ashburn region of the `acme-dev` tenancy. When you use the `docker tag` command, you'd name the new image with the fully qualified

path to its destination, in the format `.ocir.io//:`. So in this case, you'd name the new image `iad.ocir.io/ansh81vru1zp/project01/acme-web-app:version2.0.test`. Subsequently, when you use the docker push command, the image's name ensures it is pushed to the correct destination.

To push images to Oracle Cloud Infrastructure Registry using the Docker CLI:

If you already have an auth token, go to the next step. Otherwise:

In the top-right corner of the Console, open the Profile menu () and then click User Settings to view the details.

On the Auth Tokens page, click Generate Token.

Enter a friendly description for the auth token. Avoid entering confidential information.

Click Generate Token. The new auth token is displayed.

Copy the auth token immediately to a secure location from where you can retrieve it later, because you won't see the auth token again in the Console.

Close the Generate Token dialog.

References:

<https://docs.cloud.oracle.com/en-us/iaas/Content/Registry/Tasks/registrypushingimagesusingthedockercli.htm>

QUESTION NO: 3

What is the minimum amount of storage that a persistent volume claim can obtain In Oracle Cloud Infrastructure Container Engine for Kubernetes (OKE)?

- A. 1 TB
- B. 10 GB
- C. 1 GB
- D. 50 GB

ANSWER: D

Explanation:

Provisioning Persistent Volume Claims on the Block Volume Service:

Block volume quota: If you intend to create Kubernetes persistent volumes, sufficient block volume quota must be available in each availability domain to meet the persistent volume claim. Persistent volume claims must request a minimum of 50 gigabytes.

References:

<https://docs.cloud.oracle.com/en-us/iaas/Content/ContEng/Tasks/contengcreatingpersistentvolumeclaim.htm>

<https://docs.cloud.oracle.com/en-us/iaas/Content/ContEng/Concepts/contengprerequisites.htm>

QUESTION NO: 4

You have created a repository in Oracle Cloud Infrastructure Registry in the us-ashburn-1 (iad) region in your tenancy with a namespace called "heyoci".

Which three are valid tags for an image named "myapp"?

- A. iad.ocir.io/heyoci/myproject/myapp:0.0.1
- B. us-ashburn-l.ocirJo/heyoci/myapp:0.0.2-beta
- C. us-ashburn-l.ocir.io/heyoci/myproject/myapp:0.0.2-beta
- D. us-ashburn-l.ocir.io/myproject/heyoci/myapp:latest
- E. iad.ocir.io/myproject/heyoci/myapprlatest
- F. iad.ocir.io/heyoci/myapp:0.0.2-beta
- G. iad.ocir.io/heyoci/myapp:latest

ANSWER: A F G

Explanation:

Give a tag to the image that you're going to push to Oracle Cloud Infrastructure Registry by entering:

docker tag

where:

uniquely identifies the image, either using the image's id (for example, 8e0506e14874), or the image's name and tag separated by a colon (for example, acme-web-app:latest).

is in the format .ocir.io///: where:

is the key for the Oracle Cloud Infrastructure Registry region you're using. For example, iad. See [Availability by Region](#).

ocir.io is the Oracle Cloud Infrastructure Registry name.

is the auto-generated Object Storage namespace string of the tenancy that owns the repository to which you want to push the image (as shown on the Tenancy Information page). For example, the namespace of the acme-dev tenancy might be ansh81vru1zp. Note that for some older tenancies, the namespace string might be the same as the tenancy name in all lower-case letters (for example, acme-dev). Note also that your user must have access to the tenancy.

(if specified) is the name of a repository to which you want to push the image (for example, project01). Note that specifying a repository is optional (see [About Repositories](#)).

is the name you want to give the image in Oracle Cloud Infrastructure Registry (for example, acme-web-app).

is an image tag you want to give the image in Oracle Cloud Infrastructure Registry (for example, version2.0.test).

For example, for convenience you might want to group together multiple versions of the acme-web-app image in the acme-dev tenancy in the Ashburn region into a repository called project01. You do this by including the name of the repository in the image name when you push the image, in the format .ocir.io///:. For example, iad.ocir.io/ansh81vru1zp/project01/acme-web-app:4.6.3. Subsequently, when you use the docker push command, the presence of the repository in the image's name ensures the image is pushed to the intended repository.

If you push an image and include the name of a repository that doesn't already exist, a new private repository is created automatically. For example, if you enter a command like `docker push iad.ocir.io/ansh81vru1zp/project02/acme-web-app:7.5.2` and the `project02` repository doesn't exist, a private repository called `project02` is created automatically.

If you push an image and don't include a repository name, the image's name is used as the name of the repository. For example, if you enter a command like `docker push iad.ocir.io/ansh81vru1zp/acme-web-app:7.5.2` that doesn't contain a repository name, the image's name (`acme-web-app`) is used as the name of a private repository.

<https://docs.cloud.oracle.com/en-us/iaas/Content/Registry/Concepts/registrywhatisarepository.htm>

QUESTION NO: 5

Which two statements accurately describe an Oracle Functions application?

- A. A small block of code invoked in response to an Oracle Cloud Infrastructure (OCI) Events service
- B. A Docker image containing all the functions that share the same configuration
- C. An application based on Oracle Functions, Oracle Cloud Infrastructure (OCI) Events and OCI API Gateway services
- D. A common context to store configuration variables that are available to all functions in the application
- E. A logical group of functions

ANSWER: D E

Explanation:

Oracle Functions Concepts:

This topic describes key concepts you need to understand when using Oracle Functions.

Applications:

In Oracle Functions, an application is:

1. a logical grouping of functions
2. a common context to store configuration variables that are available to all functions in the application
3. a way to ensure function runtime isolation

When you define an application in Oracle Functions, you specify the subnets in which to run the functions in the application. When functions from different applications are invoked simultaneously, Oracle Functions ensures these function executions are isolated from each other.

Oracle Functions shows applications and their functions in the Console.

References:

<https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Concepts/functionsconcepts.htm>

QUESTION NO: 6

Which two statements are true for service choreography?

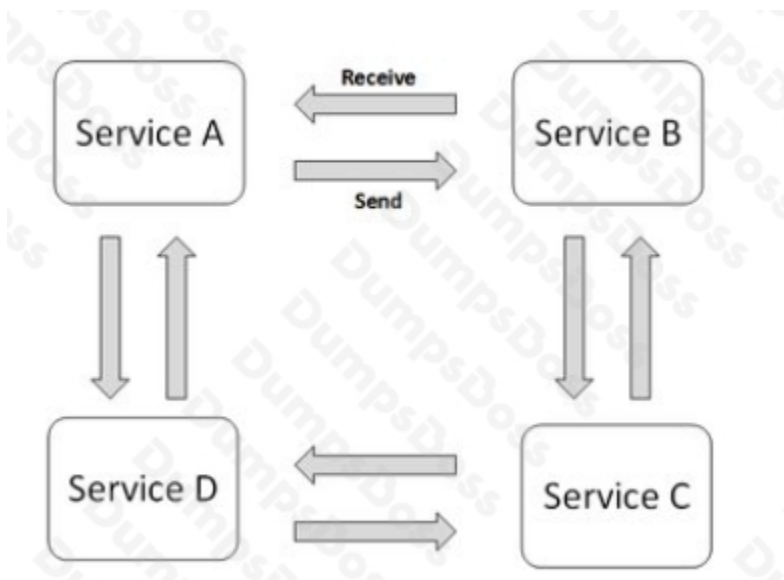
- A. Service choreographer is responsible for invoking other services.
- B. Services involved in choreography communicate through messages/messaging systems.
- C. Service choreography relies on a central coordinator.
- D. Service choreography should not use events for communication.
- E. Decision logic in service choreography is distributed.

ANSWER: B E

Explanation:

Service Choreography

Service choreography is a global description of the participating services, which is defined by exchange of messages, rules of interaction and agreements between two or more endpoints. Choreography employs a decentralized approach for service composition. the decision logic is distributed, with no centralized point.



Choreography, in contrast, does not rely on a central coordinator. and all participants in the choreography need to be aware of the business process, operations to execute, messages to exchange, and the timing of message exchanges.

References:

<https://stackoverflow.com/questions/4127241/orchestration-vs-choreography/33316988>

QUESTION NO: 7

What can you use to dynamically make Kubernetes resources discoverable to public DNS servers?

- A. ExternalDNS

- B. CoreDNS
- C. DynDNS
- D. kubeDNS

ANSWER: A

Explanation:

Setting up ExternalDNS for Oracle Cloud Infrastructure (OCI):

Inspired by [Kubernetes DNS](#), Kubernetes' cluster-internal DNS server, ExternalDNS makes Kubernetes resources discoverable via public DNS servers. Like KubeDNS, it retrieves a list of resources (Services, Ingresses, etc.) from the [Kubernetes API](#) to determine a desired list of DNS records.

In a broader sense, ExternalDNS allows you to control DNS records dynamically via Kubernetes resources in a DNS provider-agnostic way

Deploy ExternalDNS

Connect your kubectl client to the cluster you want to test ExternalDNS with. We first need to create a config file containing the information needed to connect with the OCI API.

Create a new file (oci.yaml) and modify the contents to match the example below. Be sure to adjust the values to match your own credentials:

auth:

region: us-phoenix-1

tenancy: ocid1.tenancy.oc1...

user: ocid1.user.oc1...

key: |

-----BEGIN RSA PRIVATE KEY-----

-----END RSA PRIVATE KEY-----

fingerprint: af:81:71:8e...

compartment: ocid1.compartment.oc1...

References:

<https://github.com/kubernetes-sigs/external-dns/blob/master/README.md>

<https://github.com/kubernetes-sigs/external-dns/blob/master/docs/tutorials/oracle.md>

QUESTION NO: 8

You are deploying an API via Oracle Cloud Infrastructure (OCI) API Gateway and you want to implement request policies to control access Which is NOT available in OCI API Gateway?

- A. Limiting the number of requests sent to backend services
- B. Enabling CORS (Cross-Origin Resource Sharing) support
- C. Providing authentication and authorization
- D. Controlling access to OCI resources

ANSWER: D

Explanation:

Adding Request Policies and Response Policies to API Deployment Specifications:

You can control the behavior of an API deployment you create on an API gateway by adding request and response policies to the API deployment specification:

a request policy describes actions to be performed on an incoming request from a caller before it is sent to a back end

a response policy describes actions to be performed on a response returned from a back end before it is sent to a caller

You can use request policies to:

limit the number of requests sent to back-end services

enable CORS (Cross-Origin Resource Sharing) support

provide authentication and authorization

You can add request and response policies that apply globally to all routes in an API deployment specification, and also (in some cases) request and response policies that apply only to particular routes.

Note the following:

No response policies are currently available.

API Gateway request policies and response policies are different to IAM policies, which control access to Oracle Cloud Infrastructure resources.

You can add request and response policies to an API deployment specification by:

using the Console

editing a JSON file

References:

<https://docs.cloud.oracle.com/en-us/iaas/Content/APIGateway/Tasks/apigatewayaddingrequestpolicies.htm>

QUESTION NO: 9

Which two statements are true for serverless computing and serverless architectures?

- A. Long running tasks are perfectly suited for serverless
- B. Serverless function state should never be stored externally

- C. Application DevOps team is responsible for scaling
- D. Serverless function execution is fully managed by a third party
- E. Applications running on a FaaS (Functions as a Service) platform

ANSWER: B E

Explanation:

Oracle Functions is a fully managed, multi-tenant, highly scalable, on-demand, Functions-as-a-Service platform. It is built on enterprise-grade Oracle Cloud Infrastructure and powered by the Fn Project open source engine. Use Oracle Functions (sometimes abbreviated to just Functions) when you want to focus on writing code to meet business needs.

The serverless and elastic architecture of Oracle Functions means there's no infrastructure administration or software administration for you to perform. You don't provision or maintain compute instances, and operating system software patches and upgrades are applied automatically. Oracle Functions simply ensures your app is highly-available, scalable, secure, and monitored

Applications built with a serverless infrastructure will scale automatically as the user base grows or usage increases. If a function needs to be run in multiple instances, the vendor's servers will start up, run, and end them as they are needed.

Oracle Functions is based on Fn Project. Fn Project is an open source, container native, serverless platform that can be run anywhere - any cloud or on-premises.

Serverless architectures are not built for long-running processes. This limits the kinds of applications that can cost-effectively run in a serverless architecture. Because serverless providers charge for the amount of time code is running, it may cost more to run an application with long-running processes in a serverless infrastructure compared to a traditional one.

<https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Concepts/functionsconcepts.htm>

<https://www.cloudflare.com/learning/serverless/why-use-serverless/>

QUESTION NO: 10

In a Linux environment, what is the default locations of the configuration file that Oracle Cloud Infrastructure CLI uses for profile information?

- A. /etc/.oci/config
- B. /usr/local/bin/config
- C. \$HOME/.oci/config
- D. /usr/bin/oci/config

ANSWER: C

Explanation:

By default, the Oracle Cloud Infrastructure CLI configuration file is located at `~/.oci/config`.

You might already have a configuration file as a result of installing the Oracle Cloud Infrastructure CLI.