



# 1Z0-1084-20<sup>Q&As</sup>

Oracle Cloud Infrastructure Developer 2020 Associate

**Pass Oracle 1Z0-1084-20 Exam with 100% Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/1z0-1084-20.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle  
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





## QUESTION 1

You are developing a serverless application with Oracle Functions. Your function needs to store state in a database. Your corporate security Standards mandate encryption of secret information like database

passwords.

As a function developer, which approach should you follow to satisfy this security requirement?

- A. Use the Oracle Cloud Infrastructure Console and enter the password in the function configuration section in the provided input field.
- B. Use Oracle Cloud Infrastructure Key Management to auto-encrypt the password. It will inject the auto-decrypted password inside your function container.
- C. Encrypt the password using Oracle Cloud Infrastructure Key Management. Decrypt this password in your function code with the generated key.
- D. All function configuration variables are automatically encrypted by Oracle Functions.

Correct Answer: A

Passing Custom Configuration Parameters to Functions The code in functions you deploy to Oracle Functions will typically require values for different parameters. Some pre-defined parameters are available to your functions as environment variables. But you'll often want your functions to use parameters that you've defined yourself. For example, you might create a function that reads from and writes to a database. The function will require a database connect string, comprising a username, password, and hostname. You'll probably want to define username, password, and hostname as parameters that are passed to the function when it's invoked. Using the Console To specify custom configuration parameters to pass to functions using the Console: Log in to the Console as a functions developer. In the Console, open the navigation menu. Under Solutions and Platform, go to Developer Services and click Functions. Select the region you are using with Oracle Functions. Oracle recommends that you use the same region as the Docker registry that's specified in the Fn Project CLI context (see 6. Create an Fn Project CLI Context to Connect to Oracle Cloud Infrastructure). Select the compartment specified in the Fn Project CLI context (see 6. Create an Fn Project CLI Context to Connect to Oracle Cloud Infrastructure). The Applications page shows the applications defined in the compartment. Click the name of the application containing functions to which you want to pass custom configuration parameters: To pass one or more custom configuration parameters to every function in the application, click Configuration to see the Configuration section for the application. To pass one or more custom configuration parameters to a particular function, click the function's name to see the Configuration section for the function. In the Configuration section, specify details for the first custom configuration parameter: Key: The name of the custom configuration parameter. The name must only contain alphanumeric characters and underscores, and must not start with a number. For example, username Value: A value for the custom configuration parameter. The value must only contain printable unicode characters. For example, jdoe Click the plus button to save the new custom configuration parameter. Oracle Functions combines the key-value pairs for all the custom configuration parameters (both application-wide and function-specific) in the application into a single, serially-encoded configuration object with a maximum allowable size of 4Kb. You cannot save the new custom configuration parameter if the size of the serially-encoded configuration object would be greater than 4Kb. (Optional) Enter additional custom configuration parameters as required.

## QUESTION 2

You created a pod called "nginx" and its state is set to Pending. Which command can you run to see the reason why the "nginx" pod is in the pending state?

- A. `kubectl logs pod nginx`



- B. kubectl describe pod nginx
- C. kubectl get pod nginx
- D. Through the Oracle Cloud Infrastructure Console

Correct Answer: B

Debugging Pods

The first step in debugging a pod is taking a look at it. Check the current state of the pod and recent events with the following command:

```
kubectl describe pods ${POD_NAME}
```

Look at the state of the containers in the pod. Are they all Running? Have there been recent restarts?

Continue debugging depending on the state of the pods.

My pod stays pending

If a pod is stuck in Pending it means that it can not be scheduled onto a node. Generally this is because

there are insufficient resources of one type or another that prevent scheduling. Look at the output of the kubectl describe ... command above. There should be messages from the scheduler about why it can not schedule your pod. <https://kubernetes.io/docs/tasks/debug-application-cluster/debug-pod-replication-controller/>

---

### QUESTION 3

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-1.ocir.io/heyoci/myapp:0.0.2-beta
- C. us-ashburn-1.ocir.io/heyoci/myproject/myapp:0.0.2-beta
- D. us-ashburn-1.ocir.io/myproject/heyoci/myapp:latest
- E. iad.ocir.io/myproject/heyoci/myapp:latest
- F. iad.ocir.io/heyoci/myapp:0.0.2-beta
- G. iad.ocir.io/heyoci/myapp:latest

Correct Answer: AFG

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](https://docs.cloud.oracle.com/en-us/iaas/Content/Registry/Concepts/registrywhatisarepository.htm)

#### QUESTION 4

You are developing a serverless application with Oracle Functions and Oracle Cloud Infrastructure Object Storage- Your function needs to read a JSON file object from an Object Storage bucket named "inputbucket" in compartment "qa-compartment". Your corporate security standards mandate the use of Resource Principals for this use case. Which two statements are needed to implement this use case?

- A. Set up a policy with the following statement to grant read access to the bucket:allow dynamic-group read-file-dg to read objects in compartment qa-compartment where target .bucket .name=\\ input-bucket \*
- B. Set up the following dynamic group for your function's OCID: Name: read-file-dg Rule: resource.id = `ocid1.fnfunc.oc1.phx.aaaaaaakeaobctakezjz5i4ujj7g25q7sx5mvr55pms6f4da`
- C. Set up a policy to grant all functions read access to the bucket:allow all functions in compartment qacompartment to read objects in target.bucket.name=\\input-bucket`
- D. Set up a policy to grant your user account read access to the bucket:allow user XYZ to read objects in compartment qa-compartment where target .bucket, name-\\input-bucket`
- E. No policies are needed. By default, every function has read access to Object Storage buckets in the tenancy

Correct Answer: AB

When a function you've deployed to Oracle Functions is running, it can access other Oracle Cloud Infrastructure resources. For example:

-

You might want a function to get a list of VCNs from the Networking service.



-  
You might want a function to read data from an Object Storage bucket, perform some operation on the

data, and then write the modified data back to the Object Storage bucket. To enable a function to access another Oracle Cloud Infrastructure resource, you have to include the function in a dynamic group, and then create a policy to grant the dynamic group access to that resource. <https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm>

---

#### QUESTION 5

In order to effectively test your cloud-native applications, you might utilize separate environments (development, testing, staging, production, etc.). Which Oracle Cloud Infrastructure (OCI) service can you use to create and manage your infrastructure?

- A. OCI Compute
- B. OCI Container Engine for Kubernetes
- C. OCI Resource Manager
- D. OCI API Gateway

Correct Answer: C

Resource Manager is an Oracle Cloud Infrastructure service that allows you to automate the process of provisioning your Oracle Cloud Infrastructure resources. Using Terraform, Resource Manager helps you install, configure, and manage resources through the "infrastructure-as-code" model.

[Latest 1Z0-1084-20 Dumps](#)

[1Z0-1084-20 PDF Dumps](#)

[1Z0-1084-20 Study Guide](#)