https://www.geekcert.com/1z0-1084-20.html
GeekCert.com

1Z0-1084-20^{Q&As}

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.geekcert.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

Which is NOT a supported SDk Oracle Cloud Infrastructure (OCI)?

A. Go SDK

B. Java SDK

C. NET SDK

D. Ruby SDK

E. Python SDK

Correct Answer: C

https://docs.cloud.oracle.com/en-us/iaas/Content/API/Concepts/sdks.htm

Software Development Kits (SDKs) Build and deploy apps that integrate with Oracle Cloud Infrastructure services. Each SDK provides the tools you need to develop an app, including code samples and documentation to create, test, and troubleshoot. In addition, if you want to contribute to the development of the SDKs, they are all open source and available on GitHub. SDK for Java Python SDK Ruby SDK Go SDK

QUESTION 2

Who is responsible for patching, upgrading and maintaining the worker nodes in Oracle Cloud Infrastructure Container Engine for Kubernetes (OKE)?

A. It Is automated

B. Independent Software Vendors

C. Oracle Support

D. The user

Correct Answer: D

After a new version of Kubernetes has been released and when Container Engine for Kubernetes supports the new version, you can use Container Engine for Kubernetes to upgrade master nodes running older versions of Kubernetes. Because Container Engine for Kubernetes distributes the Kubernetes Control Plane on multiple Oracle-managed master nodes (distributed across different availability domains in a region where supported) to ensure high availability, you\\'re able to upgrade the Kubernetes version running on master nodes with zero downtime. Having upgraded master nodes to a new version of Kubernetes, you can subsequently create new node pools running the newer version. Alternatively, you can continue to create new node pools that will run older versions of Kubernetes (providing those older versions are compatible with the Kubernetes version running on the master nodes). Note that you upgrade master nodes by performing an `in-place\\' upgrade, but you upgrade worker nodes by performing an `out-of-place\\' upgrade. To upgrade the version of Kubernetes running on worker nodes in a node pool, you replace the original node pool with a new node pool that has new worker nodes running the appropriate Kubernetes version. Having \\'drained\\' existing worker nodes in the original node pool to prevent new pods starting and to delete existing pods, you can then delete the original node pool.

QUESTION 3

Your Oracle Cloud Infrastructure Container Engine for Kubernetes (OKE) administrator has created an

OKE cluster with one node pool in a public subnet. You have been asked to provide a log file from one of

the nodes for troubleshooting purpose.

Which step should you take to obtain the log file?

A. ssh into the node using public key.

B. ssh into the nodes using private key.

C. It is impossible since OKE is a managed Kubernetes service.

D. Use the username open and password to login.

Correct Answer: B

Kubernetes cluster is a group of nodes. The nodes are the machines running applications. Each node can be a physical machine or a virtual machine. The node\\'s capacity (its number of CPUs and amount of memory) is defined when the node is created. A cluster comprises: - one or more master nodes (for high availability, typically there will be a number of master nodes) - one or more worker nodes (sometimes known as minions) Connecting to Worker Nodes Using SSH If you provided a public SSH key when creating the node pool in a cluster, the public key is installed on all worker nodes in the cluster. On UNIX and UNIX-like platforms (including Solaris and Linux), you can then connect through SSH to the worker nodes using the ssh utility (an SSH client) to perform administrative tasks. Note the following instructions assume the UNIX machine you use to connect to the worker node: Has the ssh utility installed. Has access to the SSH private key file paired with the SSH public key that was specified when the cluster was created. How to connect to worker nodes using SSH depends on whether you specified public or private subnets for the worker nodes when defining the node pools in the cluster. Connecting to Worker Nodes in Public Subnets Using SSH Before you can connect to a worker node in a public subnet using SSH, you must define an ingress rule in the subnet\\'s security list to allow SSH access. The ingress rule must allow access to port 22 on worker nodes from source 0.0.0.0/0 and any source port To connect to a worker node in a public subnet through SSH from a UNIX machine using the ssh utility: 1- Find out the IP address of the worker node to which you want to connect. You can do this in a number of ways: Using kubectl. If you haven\\'t already done so, follow the steps to set up the cluster\\'s kubeconfig configuration file and (if necessary) set the KUBECONFIG environment variable to point to the file. Note that you must set up your own kubeconfig file. You cannot access a cluster using a kubeconfig file that a different user set up. See Setting Up Cluster Access. Then in a terminal window, enter kubectl get nodes to see the public IP addresses of worker nodes in node pools in the cluster. Using the Console. In the Console, display the Cluster List page and then select the cluster to which the worker node belongs. On the Node Pools tab, click the name of the node pool to which the worker node belongs. On the Nodes tab, you see the public IP address of every worker node in the node pool. Using the REST API. Use the ListNodePools operation to see the public IP addresses of worker nodes in a node pool. 2- In the terminal window, enter ssh opc@ to connect to the worker node, where is the IP address of the worker node that you made a note of earlier. For example, you might enter ssh opc@192.0.2.254. Note that if the SSH private key is not stored in the file or in the path that the ssh utility expects (for example, the ssh utility might expect the private key to be stored in ~/.ssh/id_rsa), you must explicitly specify the private key filename and location in one of two ways: Use the -i option to specify the filename and location of the private key. For example, ssh -i ~/.ssh/ my_keys/my_host_key_filename opc@192.0.2.254 Add the private key filename and location to an SSH

configuration file, either the client configuration file (~/.ssh/config) if it exists, or the system-wide client

configuration file (/etc/ssh/ssh_config). For example, you might add the following:

Host 192.0.2.254 IdentityFile ~/.ssh/my_keys/my_host_key_filename

For more about the ssh utility\\\'s configuration file, enter man ssh_config Note also that permissions on the

private key file must allow you read/write/execute access, but prevent other users from accessing the file.

For example, to set appropriate permissions, you might enter chmod 600 ~/.ssh/my_keys/

my_host_key_filename. If permissions are not set correctly and the private key file is accessible to other

users, the ssh utility will simply ignore the private key file.

QUESTION 4

Which pattern can help you minimize the probability of cascading failures in your system during partial loss of connectivity or a complete service failure?

A. Retry pattern

B. Anti-corruption layer pattern

C. Circuit breaker pattern

D. Compensating transaction pattern

Correct Answer: C

A cascading failure is a failure that grows over time as a result of positive feedback. It can occur when a portion of an overall system fails, increasing the probability that other portions of the system fail. the circuit breaker pattern prevents the service from performing an operation that is likely to fail. For example, a client service can use a circuit breaker to prevent further remote calls over the network when a downstream service is not functioning properly. This can also prevent the network from becoming congested by a sudden spike in failed retries by one service to another, and it can also prevent cascading failures. Self-healing circuit breakers check the downstream service at regular intervals and reset the circuit breaker when the downstream service starts functioning properly.
https://blogs.oracle.com/developers/gettingstarted-with-microservices-part-three

QUESTION 5

You are developing a distributed application and you need a call to a path to always return a specific JSON content deploy an Oracle Cloud Infrastructure API Gateway with the below API deployment specification.

```
{
  "routes": [{
    "path": "/hello",
    "methods": ["GET"],
    "backend": {
      "type": "_____",
      "status": 200,
      "headers": [{
        "name": "Content-Type",
        "value": "application/json"
      }],
      "body" : "{\"myjson\": \"consistent response\"}"
    }
  }]
}
```

What is the correct value for type?

A. STOCK_RESPONSE_BACKEND

B. CONSTANT_BACKEND

C. JSON_BACKEND

D. HTTP_BACKEND

Correct Answer: A

"type": "STOCK_RESPONSE_BACKEND" indicates that the API gateway itself will act as the back end and return the stock response you define (the status code, the header fields and the body content). https://docs.cloud.oracle.com/en-us/iaas/Content/APIGateway/Tasks/ apigatewayaddingstockresponses.htm