1Z0-1084-21$^{Q\&As}$

Oracle Cloud Infrastructure Developer 2021 Associate

Pass Oracle 1Z0-1084-21 Exam with 100% Guarantee

QUESTION 1

Which statement accurately describes Oracle Cloud Infrastructure (OCI) Load Balancer integration with OCI Container Engine for Kubernetes (OKE)?

A. OKE service provisions an OCI Load Balancer instance for each Kubernetes service with LoadBalancer type in the YAML configuration.

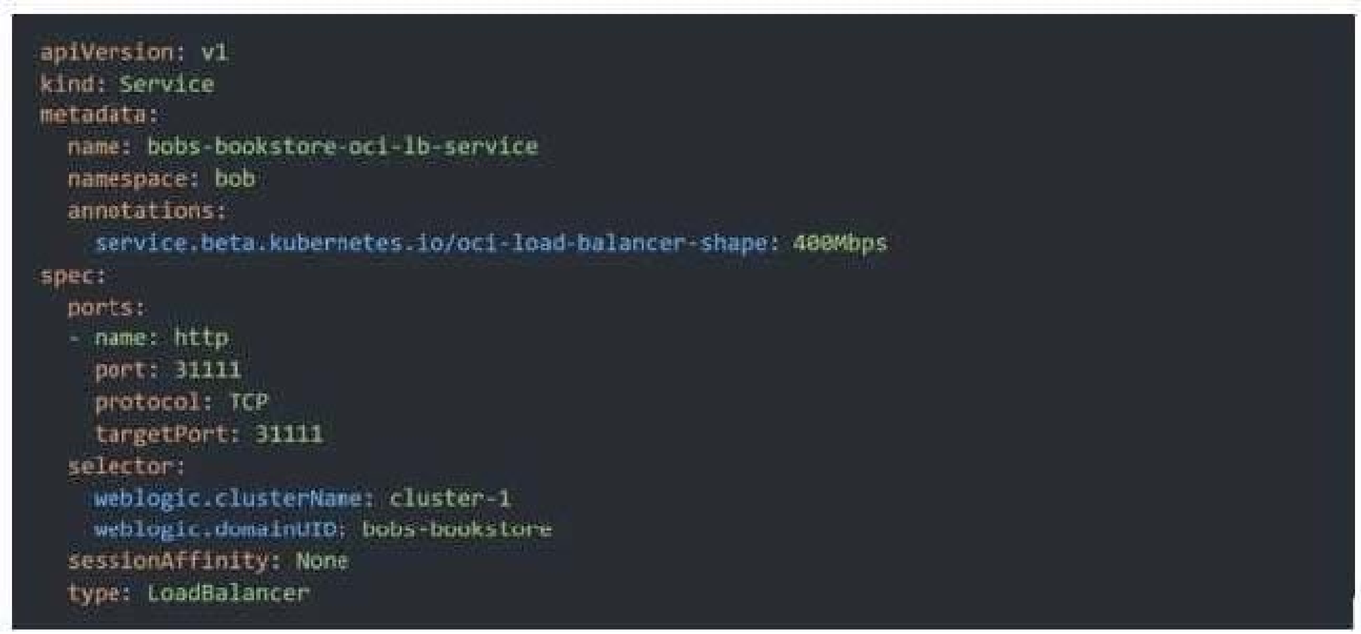
B. OCI Load Balancer instance provisioning is triggered by OCI Events service for each Kubernetes service with LoadBalancer type in the YAML configuration.

C. OCI Load Balancer instance must be manually provisioned for each Kubernetes service that requires traffic balancing.

D. OKE service provisions a single OCI Load Balancer instance shared with all the Kubernetes services with LoadBalancer type in the YAML configuration.

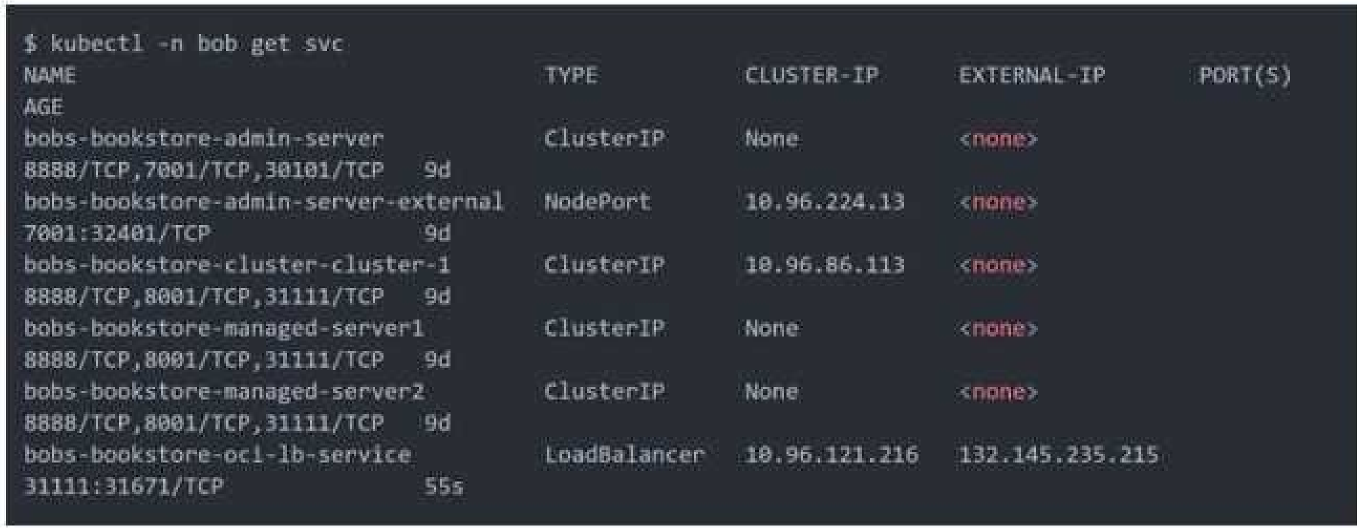
Correct Answer: D

If you are running your Kubernetes cluster on Oracle Container Engine for Kubernetes (commonly known as OKE), you can have OCI automatically provision load balancers for you by creating a Service of type LoadBalancer instead of (or in addition to) installing an ingress controller like Traefik or Voyage YAML file



```
apiVersion: v1
kind: Service
metadata:
  name: bobs-bookstore-oci-lb-service
  namespace: bob
  annotations:
    service.beta.kubernetes.io/oci-load-balancer-shape: 400Mbps
spec:
  ports:
  - name: http
    port: 31111
    protocol: TCP
    targetPort: 31111
  selector:
    weblogic.clusterName: cluster-1
    weblogic.domainUID: bobs-bookstore
  sessionAffinity: None
  type: LoadBalancer
```

When you apply this YAML file to your cluster, you will see the new service is created. After a short time (typically less than a minute) the OCI Load Balancer will be provisioned.



```
$ kubectl -n bob get svc
NAME                                    TYPE           CLUSTER-IP      EXTERNAL-IP       PORT(S)
AGE
bobs-bookstore-admin-server             ClusterIP      None            <none>
8888/TCP,7001/TCP,30101/TCP    9d
bobs-bookstore-admin-server-external    NodePort       10.96.224.13    <none>
7001:32401/TCP                 9d
bobs-bookstore-cluster-cluster-1        ClusterIP      10.96.86.113    <none>
8888/TCP,8001/TCP,31111/TCP    9d
bobs-bookstore-managed-server1          ClusterIP      None            <none>
8888/TCP,8001/TCP,31111/TCP    9d
bobs-bookstore-managed-server2          ClusterIP      None            <none>
8888/TCP,8001/TCP,31111/TCP    9d
bobs-bookstore-oci-lb-service           LoadBalancer   10.96.121.216   132.145.235.215
31111:31671/TCP                55s
```

https://oracle.github.io/weblogic-kubernetes-operator/faq/oci-lb/

QUESTION 2

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.

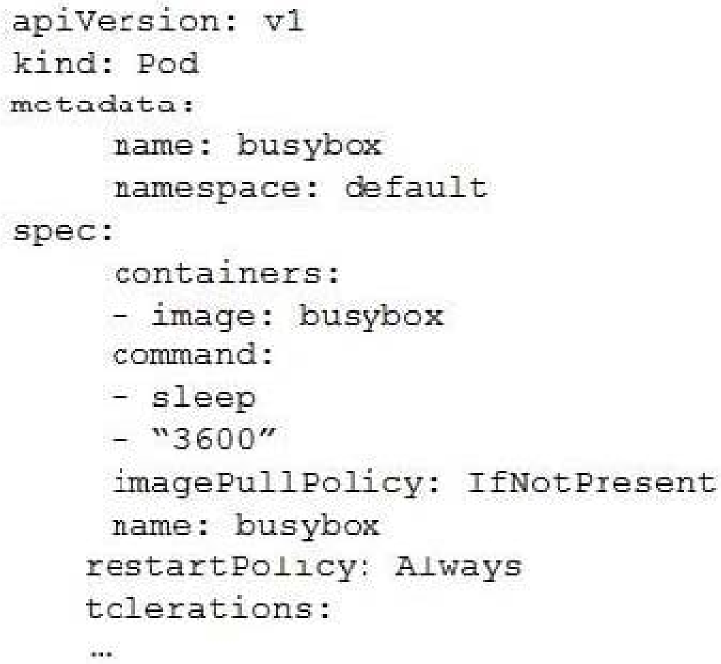
For example, to set appropriate permissions, you might enter chmod 600 ~/.ssh/my_keys/

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 3

How can you find details of the tolerations field for the sample YAML file below?



```
apiVersion: v1
kind: Pod
metadata:
      name: busybox
      namespace: default
spec:
      containers:
      - image: busybox
      command:
      - sleep
      - "3600"
      imagePullPolicy: IfNotPresent
      name: busybox
     restartPolicy: Always
     tolerations:
     …
```

A. kubectl list pod.spec.tolerations

B. kubectl explain pod.spec.tolerations

C. kubectl describe pod.spec tolerations

D. kubectl get pod.spec.tolerations

Correct Answer: B

kubectl explain to List the fields for supported resources

https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#explain

QUESTION 4

Which is NOT a valid option to execute a function deployed on Oracle Functions?

A. Send a signed HTTP requests to the function\\'s invoke endpoint

B. Invoke from Oracle Cloud Infrastructure CLI

C. Invoke from Docker CLI

D. Trigger by an event in Oracle Cloud Infrastructure Events service

E. Invoke from Fn Project CLI

Correct Answer: C

You can invoke a function that you\\'ve deployed to Oracle Functions in different ways:

Using the Fn Project CLI.

Using the Oracle Cloud Infrastructure CLI.

Using the Oracle Cloud Infrastructure SDKs.

Making a signed HTTP request to the function\\'s invoke endpoint. Every function has an invoke endpoint.

Each of the above invokes the function via requests to the API. Any request to the API must be authenticated by including a signature and the OCID of the compartment to which the function belongs in the request header. Such a request is referred to as a \\'signed\\' request. The signature includes Oracle Cloud Infrastructure credentials in an encrypted form.

QUESTION 5

Your organization uses a federated identity provider to login to your Oracle Cloud Infrastructure (OCI) environment. As a developer, you are writing a script to automate some operation and want to use OCI CLI to do that. Your security team doesn\\'t allow storing private keys on local machines.

How can you authenticate with OCI CLI?

A. Run oci setup keys and provide your credentials

B. Run oci session refresh --profile

C. Run oci session authenticate and provide your credentials

D. Run oci setup oci-cli-rc --file path/to/target/file

Correct Answer: C

Token-based authentication for the CLI allows customers to authenticate their session interactively, then use the CLI for a single session without an API signing key. This enables customers using an identity provider that is not SCIM- supported to use a federated user account with the CLI and SDKs.

Starting a Token-based CLI Session

To use token-based authentication for the CLI on a computer with a web browser:

In the CLI, run the following command. This will launch a web browser.

oci session authenticate

In the browser, enter your user credentials. This authentication information is saved to the .config file.