



Free Questions for 1Z0-1067-23 by [braindumpscollection](#)

Shared by [Manning](#) on 17-11-2023

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

Scenario: 4 (Write Identity and Access Management Policies to Secure a Tenancy)

Scenario Description: (Hands-On Performance Exam Certification)

Your company has signed up for an OCI tenancy to migrate an e-commerce application, a supply chain management (SCM) system, and a customer relationship management (CRM) system. You have been tasked with setting up the requisite identity and access management (IAM) policies for your team to begin developing on OCI.

You start by setting up the following compartment hierarchy:

Tenancy (root)

Common-Infra

Network

Security

Applications

E-Comm

SCM

CRM

You create the following groups:

Network-Admins

Security-Admins

E-Comm-Admins

SCM-Admins

CRM-Admins

Write the IAM policies for the following use cases:

Assumptions:

Assume that all policies will be attached to the root compartment.

Write one policy per given text box.

Keep policies as simple as possible by using verbs instead of permissions (for example, "inspect orm-stacks" instead of "ORM_STACK_INSPECT") and aggregate resource types instead of individual ones (for example, "file-family" instead of "file-systems" and "mount-targets")

Task 1

Write a policy statement to enable Network-Admins to create and destroy network-related resources, such as VCNs, subnets, gateways, and so on in the Network compartment.

Task 2

Write policy statements to enable E-Comm-Admins to provision and destroy compute instances in the E-Comm compartment by using networking resources in the Network compartment.[Write one policy per given text box]

Task 3

Write a policy statement to enable SCM-Admins to provision, destroy, and back up block volumes in the SCM compartment---but only in Phoenix and London.

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Task 1

Write a policy statement to enable Network-Admins to create and destroy network-related resources, such as VCNs, subnets, gateways, and so on in the Network compartment.

Solution- Policy Statement:

allow Network-Admins to manage virtual-network-family in compartment Common-Infra:Network

Task 2

Write policy statements to enable E-Comm-Admins to provision and destroy compute instances in the E-Comm compartment by using networking resources in the Network compartment.[Write one policy per given text box]

Solution- Policy Statement:

allow E-Comm-Admins to manage instance-family in compartment Applications:E-Commallow E-Comm-Admins to use virtual-network-family in compartment Common-Infra:Network

Task 3

Write a policy statement to enable SCM-Admins to provision, destroy, and back up block volumes in the SCM compartment---but only in Phoenix and London.

Solution- Policy Statement:

allow SCM-Admins to manage volume-family in compartment Applications:SCM where any{request.region='phx',request.region='lhr'}

Question 2

Question Type: MultipleChoice

Scenario: 3 (Use the OCI CLI to Work with Object Storage from a Compute Instance)

Scenario Description: (Hands-On Performance Exam Certification)

Your company runs a web application in OCI that generates log files. You want to upload these files to OCI Object Storage to meet data retention requirements. Some files need to be retained indefinitely, whereas others can be deleted after 30 days. Use the OCI CLI to create bucket and upload the log directory and create a lifecycle policy rule to delete temporary files after 30 days.

Pre-Configuration:

To fulfill this requirement, you are provided with the following:

Access to an OCI tenancy, an assigned compartment, and OCI credentials

A compute instance with OCI CLI installed and a set of files in `~/dir_to_upload` to use

Access to the OCI Console

Required IAM policies

Assumptions:

Perform the tasks by using the OCI CLI on the compute instance.

Use instance principal authentication for all CLI commands; the instance has been given the policies necessary.

Connect to the compute instance using Cloud Shell's private networking and the provided SSH key.

An SSH key pair has been provided to you for the compute instance.

Private Key https://objectstorage.us-ashburn-1.oraclecloud.com/n/tenancyname/b/PBT_Storage/o/PKey.key

Note: Throughout your exam, ensure to use assigned Compartment , User Name and Region.

Complete the following tasks in the provisioned OCI environment:

Task 1: Create a Bucket in Object Storage

Task 2: Upload a Directory's Contents to Object Storage

Task 3: Add a Lifecycle Policy to the Bucket

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Task 1: Create a Bucket in Object Storage

Create a bucket named CloudOpsBucket_<user id> with the following properties:

Storage tier: Standard

Auto-tiering: Disabled

Object versioning: Enabled

Emit events: Disabled

Keys: Oracle-managed

Visibility: Private

Task 2: Upload a Directory's Contents to Object Storage

Upload the contents of the directory `~/dir_to_upload` and its subdirectories to the bucket `CloudOpsBucket`

Task 3: Add a Lifecycle Policy to the Bucket

Create a lifecycle policy rule that deletes all files from `~/dir_to_upload/temp` after 30 days

Task 1: Create a bucket in Object Storage

1. Open Cloud Shell in the console. Under `Network` along the top, select `Ephemeral Private Network Setup`.
2. Select the subnet of the compute instance.
3. SSH into the compute instance using the provided SSH key:

```
ssh -i /path/to/key opc@
```


4. In the compute instance, create the bucket with the following command (note that it's one long line):

```
oci os bucket create -c '<compartment_id>' --name 'CloudOpsBucket' --auth instance_principal --versioning 'Enabled'
```

Task 2: Upload a directory's contents to Object Storage

1. Upload the contents of the specified directory and subdirectories with the following command (note that it's one long line):

```
oci os object bulk-upload -bn 'CloudOpsBucket' --src-dir '~/dir_to_upload' --auth instance_principal
```

Task 3: Add a lifecycle policy to the bucket

1. Create a file named rule.json

2. Add the following content to rule.json:

```
{'items': [{'action': 'DELETE', 'is-enabled': true, 'name': 'Delete-Rule', 'object-name-filter': {'exclusion-patterns': null, 'inclusion-patterns': null, 'inclusion-prefixes': ['temp/']}, 'target': 'objects', 'time-amount': 30, 'time-unit': 'DAYS'}]}
```

3. Add the lifecycle policy rule with the following command:

```
oci os object-lifecycle-policy put -bn 'CloudOpsBucket' --from-json file://rule.json ---auth instance_principal
```

Top of Form

Question 3

Question Type: MultipleChoice

Scenario: 2 (Oracle Cloud-init and AutoScaling: Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool)

Scenario Description: (Hands-On Performance Exam Certification)

You're deploying an Apache-based web application on OCI that requires horizontal autoscaling.

To configure instances upon provisioning, write a cloud-init script for Oracle Linux 8 that installs and enables Apache (httpd), and opens the firewall for HTTP on TCP port 80. Create an instance configuration and include the cloud-init script in it. Use this instance configuration to create an instance pool and autoscaling configuration.

Pre-Configuration:

To fulfill this requirement, you are provided with the following:

Access to an OCI tenancy, an assigned compartment, and OCI credentials

A VCN Cloud-Init Challenge VCN with an Internet gateway and a public subnet. The security list for the subnet allows ingress via TCP ports 22 and 80 (SSH and HTTP). The route table forwards all egress to the Internet gateway.

Access to the OCI Console

Required IAM policies

An SSH key pair for the compute instance

Public Key https://objectstorage.us-ashburn-1.oraclecloud.com/n/tenancyname/b/PBT_Storage/o/PublicKey.pub

Private Key https://objectstorage.us-ashburn-1.oraclecloud.com/n/tenancyname/b/PBT_Storage/o/PKey.key

Note: Throughout your exam, ensure to use assigned Compartment , User Name , and Region.

Complete the following tasks in the provisioned OCI environment:

Task 1(a): Develop the cloud-init Script:

Task 1(b): Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool:

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Task 1(a): Develop the cloud-init Script:

Create a compute instance `pbt_cloud_init_vm_01` with the following properties:

Shape: `VM.Standard.A1.Flex` instance with 1 OCPU and 6 GB memory

Image: Oracle Linux 8

Placement: Use any of the availability domains

Network:

Place in the public subnet Cloud-Init Challenge SNT

Assign a public IPv4

Use the SSH public key

Add a cloud-init script and perform the following:

Use yum or dnf to install httpd.

Use systemctl to enable and start httpd

Open the firewall to http:

```
sudo firewall-offline-cmd --add-service=http
```

```
systemctl restart firewalld
```

Mark Complete

Task 1(b): Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool:

You're deploying an Apache-based web application on OCI that requires horizontal autoscaling.

To configure instances upon provisioning, write a cloud-init script for Oracle Linux 8 that installs and enables Apache (httpd), and opens the firewall for HTTP on TCP port 80. Create an instance configuration and include the cloud-init script in it. Use this instance configuration to create an instance pool and autoscaling configuration.

Task 2: Create an Autoscaling Instance Pool Including the cloud-init Script:

Create an instance configuration named `pbt_cloud_init_config_01` with the following properties:

Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory

Image: Oracle Linux 8

Placement: Use any of the availability domains

Network:

Place in the public subnet Cloud-Init Challenge SNT

Assign a public IPv4

Use the SSH public key

Attach the cloud-init script created in Task 1

Create an instance pool named `pbt_cloud_init_pool_01` with one instance by using the instance configuration `pbt_cloud_init_config_01`

Create and attach an autoscaling configuration named `pbt_cloud_autoscaling_config_01` with the following settings:

Metric-based autoscaling

Cooldown: 300 second

Performance metric: CPU utilization

Scale-out rule:

Operator: Greater than (>)

Threshold: 75%

Number of instances to add: 1

Scale-in rule:

Operator: Less than (<)

Threshold: 25%

Number of instances to remove: 1

Scaling limits:

Minimum number of instances: 1

Maximum number of instances: 2

Initial number of instances: 1

Task 1: Develop the cloud-init script

In the main menu, go to Compute > Instances and click Create an Instance

In the instance creation menu, enter the following details

- a. Name: Provide name given in the instructions
- b. Compartment: Use the assigned compartment
- c. Placement: Use any of the availability domains
- d. Image: Oracle Linux 8
- e. Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory
- f. Network:
 - i. Place in the public subnet
 - ii. Assign a public IPv4
- g. SSH keys: Upload or paste the provided SSH public key
- h. Boot volume: Leave as default
- i. Under advanced options, add the following cloud-init script:

```
#!/bin/sh  
sudo dnf install httpd --assumeyes --quiet  
sudo systemctl enable httpd  
sudo systemctl start httpd  
sudo firewall-offline-cmd --add-service=http  
systemctl restart firewalld
```

j. Create the instance.

Task 2: Create an autoscaling instance pool including the cloud-init script

1. In the main menu, go to Compute>Instance Configurations. Click Create instance configuration.

a. In the instance configuration creation menu, enter the same details as before:

b. Name: Provide name given in the instruction/if not specified provide any name

c. Compartment: Assigned compartment

d. Placement: Use any of the availability domains

e. Image: Oracle Linux 8

f. Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory

g. Network:

i. Place in the public subnet

ii. Assign a public IPv4

h. SSH keys: Upload or paste the provided SSH public key

i. Boot volume: Leave as default

j. Under advanced options, add the following cloud-init script:


```
#!/bin/shsudo dnf install httpd --assumeyes --quietsudo systemctl enable httpdsudo systemctl start httpdsudo firewall-offline-cmd --add-service=httpsudo systemctl restart firewalld
```

k. Create the instance configuration.

Task 2: In the main menu, go to Compute > Instance Pools. Click Create instance pool.

Enter the following details:

a. Name: Provide name given in the instruction/if not specified provide any name

b. Compartment: Assigned compartment

c. Instance configuration: Created in last step

d. Number of instances: 1

e. Select any availability domain

f. Leave fault domain unselected

g. Primary VNIC: Provided VCN in the instructions

h. Subnet: Public subnet

i. Do not attach a load balancer

j. Create the instance pool

Task 3: In the main menu, go to Compute > Autoscaling Configurations. Click Create autoscaling configuration and enter the following details:

a. Name: Provide name given in the instruction/if not specified provide any name

b. Compartment: Assigned compartment

c. Instance Pool: Created in last step

d. Select Metric-based autoscaling

e. Autoscaling policy name: Does not matter

f. Cooldown: 300 seconds

g. Performance metric: CPU utilization

h. Scale-out rule:

i. Operator: Greater than (>)

ii. Threshold: 75%

iii. Number of instances to add: 1

i. Scale-in rule:

i. Operator: Less than (<)

- ii. Threshold: 25%
- iii. Number of instances to remove: 1
- j. Scaling limits:
 - i. Minimum number of instances: 1
 - ii. Maximum number of instances: 2
 - iii. Initial number of instances: 1
- k. Create the autoscaling configuration.

Question 4

Question Type: MultipleChoice

Scenario: 1 (Create a reusable VCN Configuration with Terraform)

Scenario Description: (Hands-On Performance Exam Certification)

You'll launch and destroy a VCN and subnet by creating Terraform automation scripts and issuing commands in Code Editor. Next, you'll download those Terraform scripts and create a stack by uploading them into Oracle Cloud Infrastructure Resource Manager.

You'll then use that service to launch and destroy the same VCN and subnet.

In this scenario, you will:

- a. Create a Terraform folder and file in Code Editor.
- b. Create and destroy a VCN using Terraform.
- c. Create and destroy a VCN using Resource Manager.

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Create a Terraform Folder and File in Code Editor:

You'll create a folder and file to hold your Terraform scripts.

1. Log in to your tenancy in the Cloud Console and open the Code Editor, whose icon is at the top-right corner, to the right of the CLI Cloud Shell icon.
2. Expand the Explorer panel with the top icon on the left panel. It looks like two overlapping documents.
3. Expand the drop-down for your home directory if it isn't already expanded. It's okay if it is empty.
4. Create a new folder by clicking `File`, then `New Folder`, and name it `terraform-vcn`.
5. Create a file in that folder by clicking `File`, then `New File`, and name it `vcn.tf`. To make Code Editor, create the file in the correct folder, click the folder name in your home directory to highlight it.
6. First, you'll set up Terraform and the OCI Provider in this directory. Add these lines to the file:

```
terraform {required_providers {oci = {source = 'oracle/oci' version = '>=4.67.3'}}required_version = '>= 1.0.0'}
```
7. Save the changes by clicking `File`, then `Save`.
8. Now, run this code. Open a terminal panel in Cloud Editor by clicking `Terminal`, then `New Terminal`.
9. Use `pwd` to check that you are in your home directory.
10. Enter `ls` and you should see your `terraform_vcn` directory.
11. Enter `cd terraform_vcn/` to change to that directory with.
12. Use `terraform init` to initialize this directory for Terraform.
13. Use `ls -a` and you should see that Terraform created a hidden directory and file.

Create and Destroy a VCN Using Terraform

You'll create a Terraform script that will launch a VCN and subnet.

You'll then alter your script and create two additional files that will apply a compartment OCID variable to your Terraform script.

Write the Terraform

1. Add the following code block to your Terraform script to declare a VCN, replacing `<your_compartment_ocid>` with the proper OCID. The only strictly required parameter is the compartment OCID, but you'll add more later.

If you need to retrieve your compartment OCID, navigate to **Identity & Security**, then **Compartments**. Find your compartment, hover the cursor over the OCID, and click **Copy**.

```
resource 'oci_core_vcn' 'example_vcn' {compartment_id = '<your_compartment_ocid>'}
```

This snippet declares a resource block of type `oci_core_vcn`. The label that Terraform will use for this resource is `example_vcn`.

2. In the terminal, run `terraform plan`, and you should see that Terraform would create a VCN. Because most of the parameters were unspecified, terraform will list their values as `"(known after apply)."` You can ignore the `"-out option to save this plan"` warning.

Note that `terraform plan` parses your Terraform configuration and creates an execution plan for the associated stack, while `terraform apply` applies the execution plan to create (or modify) your resources.

3. Add a display name and CIDR block (the bolded portion) to the code. Note that we want to set the `cidr_block` parameter, rather than `cidr_block` (which is deprecated).

```
resource 'oci_core_vcn' 'example_vcn' {compartment_id = '<your_compartment_ocid>'display_name = 'VCN-01'cidr_blocks = ['10.0.0.0/16']}
```

4. Save the changes and run terraform plan again. You should see the display name and CIDR block reflected in Terraform's plan.

5. Now add a subnet to this VCN. At the bottom of the file, add the following block:

```
resource 'oci_core_subnet' 'example_subnet' {compartment_id = '<your_compartment_ocid>'display_name = 'SNT-01'vcn_id = oci_core_vcn.example_vcn.idcidr_block = '10.0.0.0/24'}
```

Note the line where we set the VCN ID. Here we reference the OCID of the previously declared VCN, using the name we gave it to Terraform: example_vcn. This dependency makes Terraform provision the VCN first, wait for OCI to return the OCID, then provision the subnet.

6. Run terraform plan to see that it will now create a VCN and subnet.

Add Variables

7. Before moving on there are a few ways to improve the existing code. Notice that the subnet and VCN both need the compartment OCID. We can factor this out into a variable. Create a file named variables.tf

8. In variables.tf, declare a variable named compartment_id:

```
variable 'compartment_id' {type = string}
```

9. In vcn.tf, replace all instances of the compartment OCID with var.compartment_id as follows:

```
terraform {required_providers {oci = {source = 'oracle/oci'version = '>=4.67.3'}}required_version = '>= 1.0.0'}resource 'oci_core_vcn'  
'example_vcn' {compartment_id = var.compartment_iddisplay_name = 'VCN-01'cidr_blocks = ['10.0.0.0/16']}resource 'oci_core_subnet'  
'example_subnet' {compartment_id = var.compartment_iddisplay_name = 'SNT-01'vcn_id = oci_core_vcn.example_vcn.idcidr_block =  
'10.0.0.0/24'}
```

Save your changes in both vcn.tf and variables.tf

10. If you were to run terraform plan or apply now, Terraform would see a variable and provide you a prompt to input the compartment OCID. Instead, you'll provide the variable value in a dedicated file. Create a file named exactly terraform.tfvars

11. Terraform will automatically load values provided in a file with this name. If you were to use a different name, you would have to provide the file name to the Terraform CLI. Add the value for the compartment ID in this file:

```
compartment_id = '<your_compartment_ocid>'
```

Be sure to save the file.

12. Run terraform plan and you should see the same output as before.

Provision the VCN

13. Run terraform apply and confirm that you want to make the changes by entering yes at the prompt.

14. Navigate to VCNs in the console. Ensure that you have the right compartment selected. You should see your VCN. Click its name to see the details. You should see its subnet listed.

Terminate the VCN

15. Run terraform destroy. Enter y to confirm. You should see the VCN terminate. Refresh your browser if needed.

Create and Destroy a VCN Using Resource Manager (You will most probably be tested on this in the actual certification)

We will reuse the Terraform code but replace the CLI with Resource Manager.

1. Create a folder named terraform_vcn on your host machine. Download the vcn.tf, terraform.tfvars, and variables.tf files from Code Editor and move them to the terraform_vcn folder to your local machine. To download from Code Editor, right-click the file name in the Explorer panel and select Download. You could download the whole folder at once, but then you would have to delete Terraform's hidden files.

Create a Stack

2. Navigate to Resource Manager in the Console's navigation menu under Developer Services. Go to the Stack page.

3. Click Create stack.

a. The first page of the form will be for stack information.

1) For the origin of the Terraform configuration, keep My configurations selected.

2) Under Stack configuration, upload your terraform_vcn folder.

3) Under Custom providers, keep Use custom Terraform providers deselected.

4) Name the stack and give it a description.

5) Ensure that your compartment is selected.

6) ClickNext.

b. The second page will be for variables.

1) Because you uploaded a terraform.tfvars file, Resource Manager will auto-populate the variable for compartment OCID.

2) ClickNext.

c. The third page will be for review.

1) KeepRun applydeselected.

2) ClickCreate. This will take you to the stack's details page.

Run a Plan Job

4. The stack itself is only a bookkeeping resource---no infrastructure was provisioned yet. You should be on the stack's page. ClickPlan. A form will pop up.

a. Name the job RM-Plan-01.

b. ClickPlanagain at the bottom to submit a job for Resource Manager to run terraform plan. This will take you to the job's details page.

5. Wait for the job to complete, and then view the logs. They should match what you saw when you ran Terraform in Code Editor.

Run an Apply Job

6. Go back to the stack's details page (use the breadcrumbs). ClickApply. A form will pop up.

- a. Name the job RM-Apply-01.
 - b. Under Apply job plan resolution, select the plan job we just ran (instead of "Automatically approve"). This makes it execute based on the previous plan, instead of running a new one.
 - c. Click Apply to submit a job for Resource Manager to run terraform apply. This will take you to the job's details page.
7. Wait for the job to finish. View the logs and confirm that it was successful.

View the VCN

8. Navigate to VCNs in the Console through the navigation menu under Networking and Virtual Cloud Networks.
9. You should see the VCN listed in the table. Click its name to go to its Details page.
10. You should see the subnet listed.

Run a Destroy Job

11. Go back to the stack's details page in Resource Manager.
12. Click Destroy. Click Destroy again on the menu that pops up.
13. Wait for the job to finish. View the logs to see that it completed successfully.
14. Navigate back to VCNs in the Console. You should see that it has been terminated.
15. Go back to the stack in Resource Manager. Click the drop-down for More actions. Select Delete stack. Confirm by selecting Delete.

Question 5

Question Type: MultipleChoice

Which statement about Oracle Cloud Infrastructure paravirtualized block volume attachments is TRUE? (Choose the best answer.)

Options:

- A-** Paravirtualized volumes may reduce the maximum IOPS performance for larger block volumes.
- B-** Paravirtualized volumes become immediately available on bare metal compute instances.
- C-** Paravirtualization utilizes the internal storage stack of compute instance OS and net-work hardware virtualization to access block volumes.
- D-** Paravirtualized is required to manage iSCSI configuration for virtual machine instances.

Answer:

A

Explanation:

Topic 2, Hands-on Performance Based

Question 6

Question Type: MultipleChoice

Scenario: 2 (Oracle Cloud-init and AutoScaling: Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool)

Scenario Description: (Hands-On Performance Exam Certification)

You're deploying an Apache-based web application on OCI that requires horizontal autoscaling.

To configure instances upon provisioning, write a cloud-init script for Oracle Linux 8 that installs and enables Apache (httpd), and opens the firewall for HTTP on TCP port 80. Create an instance configuration and include the cloud-init script in it. Use this instance configuration to create an instance pool and autoscaling configuration.

Pre-Configuration:

To fulfill this requirement, you are provided with the following:

Access to an OCI tenancy, an assigned compartment, and OCI credentials

A VCN Cloud-Init Challenge VCN with an Internet gateway and a public subnet. The security list for the subnet allows ingress via TCP ports 22 and 80 (SSH and HTTP). The route table forwards all egress to the Internet gateway.

Access to the OCI Console

Required IAM policies

An SSH key pair for the compute instance

Public Key https://objectstorage.us-ashburn-1.oraclecloud.com/n/tenancyname/b/PBT_Storage/o/PublicKey.pub

Private Key https://objectstorage.us-ashburn-1.oraclecloud.com/n/tenancyname/b/PBT_Storage/o/PKey.key

Note: Throughout your exam, ensure to use assigned Compartment , User Name , and Region.

Complete the following tasks in the provisioned OCI environment:

Task 1(a): Develop the cloud-init Script:

Task 1(b): Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool:

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Task 1(a): Develop the cloud-init Script:

Create a compute instance pbt_cloud_init_vm_01 with the following properties:

Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory

Image: Oracle Linux 8

Placement: Use any of the availability domains

Network:

Place in the public subnet Cloud-Init Challenge SNT

Assign a public IPv4

Use the SSH public key

Add a cloud-init script and perform the following:

Use yum or dnf to install httpd.

Use systemctl to enable and start httpd

Open the firewall to http:

```
sudo firewall-offline-cmd --add-service=http
```

```
systemctl restart firewalld
```

Mark Complete

Task 1(b): Use cloud-init to Configure Apache on Instances in an Autoscaling Instance Pool:

You're deploying an Apache-based web application on OCI that requires horizontal autoscaling.

To configure instances upon provisioning, write a cloud-init script for Oracle Linux 8 that installs and enables Apache (httpd), and opens the firewall for HTTP on TCP port 80. Create an instance configuration and include the cloud-init script in it. Use this instance configuration to create an instance pool and autoscaling configuration.

Task 2: Create an Autoscaling Instance Pool Including the cloud-init Script:

Create an instance configuration named `pbt_cloud_init_config_01` with the following properties:

Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory

Image: Oracle Linux 8

Placement: Use any of the availability domains

Network:

Place in the public subnet Cloud-Init Challenge SNT

Assign a public IPv4

Use the SSH public key

Attach the cloud-init script created in Task 1

Create an instance pool named `pbt_cloud_init_pool_01` with one instance by using the instance configuration `pbt_cloud_init_config_01`

Create and attach an autoscaling configuration named pbt_cloud_autoscaling_config_01 with the following settings:

Metric-based autoscaling

Cooldown: 300 second

Performance metric: CPU utilization

Scale-out rule:

Operator: Greater than (>)

Threshold: 75%

Number of instances to add: 1

Scale-in rule:

Operator: Less than (<)

Threshold: 25%

Number of instances to remove: 1

Scaling limits:

Minimum number of instances: 1

Maximum number of instances: 2

Initial number of instances: 1

Task 1: Develop the cloud-init script

In the main menu, go to Compute > Instances and click Create an Instance

In the instance creation menu, enter the following details

- a. Name: Provide name given in the instructions
- b. Compartment: Use the assigned compartment
- c. Placement: Use any of the availability domains
- d. Image: Oracle Linux 8
- e. Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory
- f. Network:
 - i. Place in the public subnet
 - ii. Assign a public IPv4
- g. SSH keys: Upload or paste the provided SSH public key
- h. Boot volume: Leave as default
- i. Under advanced options, add the following cloud-init script:

```
#!/bin/shsudo dnf install httpd --assumeyes --quietsudo systemctl enable httpdsudo systemctl start httpdsudo firewall-offline-cmd --add-service=httpsudo systemctl restart firewalld
```

j. Create the instance.

Task 2: Create an autoscaling instance pool including the cloud-init script

1. In the main menu, go to Compute>Instance Configurations. Click Create instance configuration.

a. In the instance configuration creation menu, enter the same details as before:

b. Name: Provide name given in the instruction/if not specified provide any name

c. Compartment: Assigned compartment

d. Placement: Use any of the availability domains

e. Image: Oracle Linux 8

f. Shape: VM.Standard.A1.Flex instance with 1 OCPU and 6 GB memory

g. Network:

i. Place in the public subnet

ii. Assign a public IPv4

h. SSH keys: Upload or paste the provided SSH public key

i. Boot volume: Leave as default

j. Under advanced options, add the following cloud-init script:

```
#!/bin/shsudo dnf install httpd --assumeyes --quietsudo systemctl enable httpdsudo systemctl start httpdsudo firewall-offline-cmd --add-service=httpsudo systemctl restart firewalld
```

k. Create the instance configuration.

Task 2: In the main menu, go to Compute > Instance Pools. Click Create instance pool.

Enter the following details:

a. Name: Provide name given in the instruction/if not specified provide any name

b. Compartment: Assigned compartment

c. Instance configuration: Created in last step

d. Number of instances: 1

e. Select any availability domain

f. Leave fault domain unselected

g. Primary VNIC: Provided VCN in the instructions

h. Subnet: Public subnet

i. Do not attach a load balancer

j. Create the instance pool

Task 3: In the main menu, go to Compute > Autoscaling Configurations. Click Create autoscaling configuration and enter the following details:

a. Name: Provide name given in the instruction/if not specified provide any name

b. Compartment: Assigned compartment

c. Instance Pool: Created in last step

d. Select Metric-based autoscaling

e. Autoscaling policy name: Does not matter

f. Cooldown: 300 seconds

g. Performance metric: CPU utilization

h. Scale-out rule:

i. Operator: Greater than (>)

ii. Threshold: 75%

iii. Number of instances to add: 1

- i. Scale-in rule:
 - i. Operator: Less than (<)
 - ii. Threshold: 25%
 - iii. Number of instances to remove: 1
- j. Scaling limits:
 - i. Minimum number of instances: 1
 - ii. Maximum number of instances: 2
 - iii. Initial number of instances: 1
- k. Create the autoscaling configuration.

Question 7

Question Type: MultipleChoice

Scenario: 4 (Write Identity and Access Management Policies to Secure a Tenancy)

Scenario Description: (Hands-On Performance Exam Certification)

Your company has signed up for an OCI tenancy to migrate an e-commerce application, a supply chain management (SCM) system, and a customer relationship management (CRM) system. You have been tasked with setting up the requisite identity and access management (IAM) policies for your team to begin developing on OCI.

You start by setting up the following compartment hierarchy:

Tenancy (root)

Common-Infra

Network

Security

Applications

E-Comm

SCM

CRM

You create the following groups:

Network-Admins

Security-Admins

E-Comm-Admins

SCM-Admins

CRM-Admins

Write the IAM policies for the following use cases:

Assumptions:

Assume that all policies will be attached to the root compartment.

Write one policy per given text box.

Keep policies as simple as possible by using verbs instead of permissions (for example, "inspect orm-stacks" instead of "ORM_STACK_INSPECT") and aggregate resource types instead of individual ones (for example, "file-family" instead of "file-systems" and "mount-targets")

Task 1

Write a policy statement to enable Network-Admins to create and destroy network-related resources, such as VCNs, subnets, gateways, and so on in the Network compartment.

Task 2

Write policy statements to enable E-Comm-Admins to provision and destroy compute instances in the E-Comm compartment by using networking resources in the Network compartment.[Write one policy per given text box]

Task 3

Write a policy statement to enable SCM-Admins to provision, destroy, and back up block volumes in the SCM compartment---but only in Phoenix and London.

Options:

A- See the solution below with Step by Step Explanation

Answer:

A

Explanation:

Task 1

Write a policy statement to enable Network-Admins to create and destroy network-related resources, such as VCNs, subnets, gateways, and so on in the Network compartment.

Solution- Policy Statement:

allow Network-Admins to manage virtual-network-family in compartment Common-Infra:Network

Task 2

Write policy statements to enable E-Comm-Admins to provision and destroy compute instances in the E-Comm compartment by using networking resources in the Network compartment.[Write one policy per given text box]

Solution- Policy Statement:

allow E-Comm-Admins to manage instance-family in compartment Applications:E-Commallow E-Comm-Admins to use virtual-network-family in compartment Common-Infra:Network

Task 3

Write a policy statement to enable SCM-Admins to provision, destroy, and back up block volumes in the SCM compartment---but only in Phoenix and London.

Solution- Policy Statement:

allow SCM-Admins to manage volume-family in compartment Applications:SCM where any{request.region='phx',request.region='lhr'}

Question 8

Question Type: MultipleChoice

Which statement about Oracle Cloud Infrastructure paravirtualized block volume attachments is TRUE? (Choose the best answer.)

Options:

- A- Paravirtualized volumes may reduce the maximum IOPS performance for larger block volumes.
- B- Paravirtualized volumes become immediately available on bare metal compute instances.
- C- Paravirtualization utilizes the internal storage stack of compute instance OS and net-work hardware virtualization to access block volumes.
- D- Paravirtualized is required to manage iSCSI configuration for virtual machine instances.

Answer:

A

Explanation:

Topic 2, Hands-on Performance Based

To Get Premium Files for 1Z0-1067-23 Visit

<https://www.p2pexams.com/products/1z0-1067-23>

For More Free Questions Visit

<https://www.p2pexams.com/oracle/pdf/1z0-1067-23>

