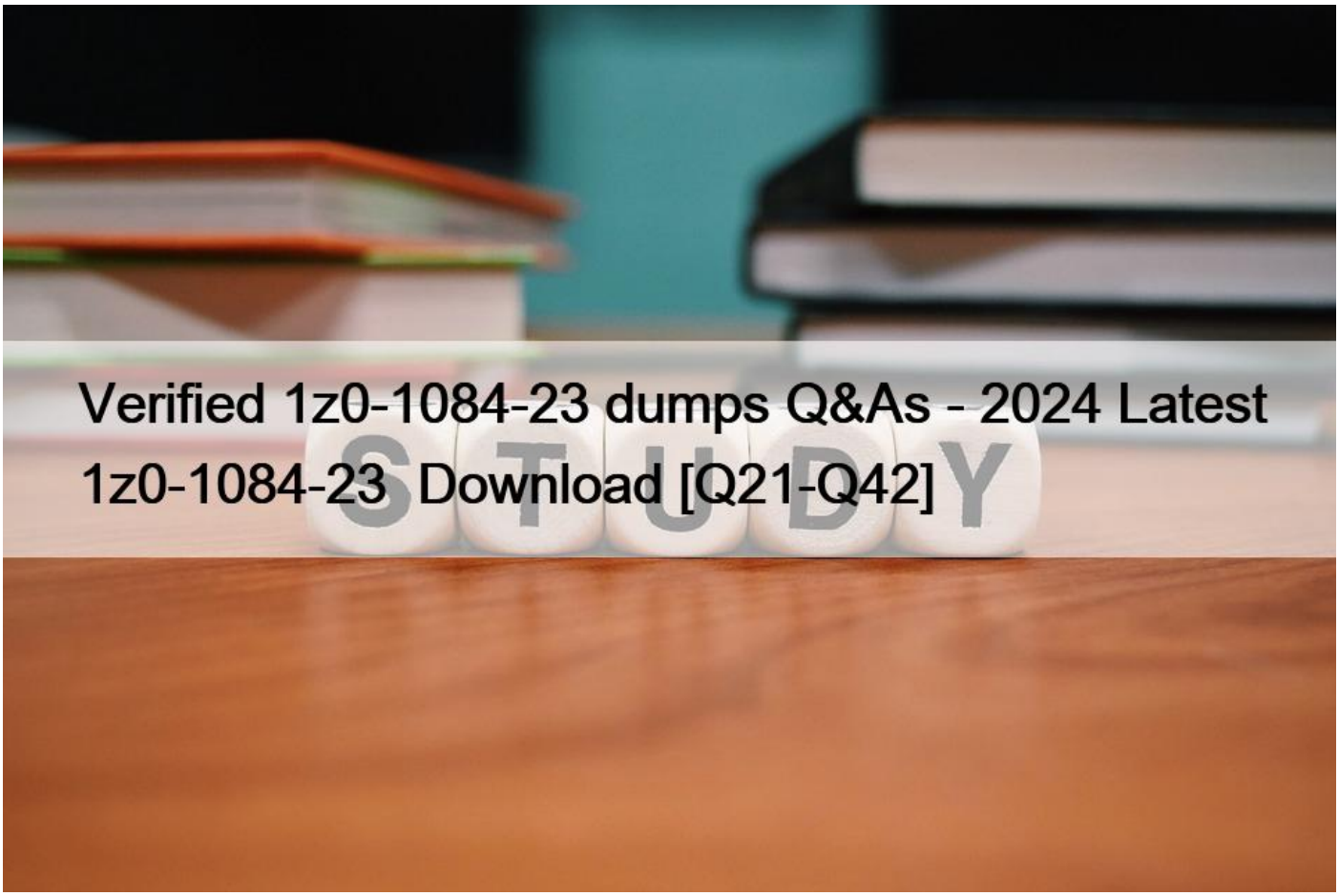
Verified 1z0-1084-23 dumps Q&As - 2024 Latest 1z0-1084-23 Download [Q21-Q42



Verified 1z0-1084-23 dumps Q&As - 2024 Latest 1z0-1084-23 Download
Updated 100% Cover Real 1z0-1084-23 Exam Questions - 100% Pass Guarantee

NO.21 You are instructed to automate manual tasks and help software teams manage complex environments at scale using the Oracle Cloud Infrastructure (OCI) services. Which THREE OCI services can be leveraged to securely store and version your application’s source code, and automate the building, testing, and deployment of applications to the OCI platform? (Choose three.)

* DevOps
* Container Engine for Kubernetes
* Oracle APEX Application Development
* Resource Manager
* Oracle Cloud Infrastructure Registry
* Oracle Cloud Logging Analytics

The three OCI services that can be leveraged to securely store and version your application’s source code, and automate the building, testing, and deployment of applications to the OCI platform are: DevOps: OCI provides a comprehensive set of DevOps services, including Oracle Developer Cloud Service, which allows you to manage source code repositories, automate builds and testing, and streamline the deployment process. Container Engine for Kubernetes: OCI’s Container Engine for Kubernetes

(OKE) enables you to deploy and manage containerized applications using Kubernetes. It provides a scalable and reliable platform for automating the deployment of your applications. Oracle Cloud Infrastructure Registry: OCI Registry is a fully managed, private container registry that allows you to securely store and manage Docker images. It integrates with other OCI services, such as Container Engine for Kubernetes, to facilitate seamless deployment and orchestration of containerized applications. These services combined provide the necessary tools and infrastructure to support continuous integration and continuous deployment (CI/CD) workflows, enabling efficient and automated application development and deployment processes in the Oracle Cloud Infrastructure environment.

NO.22 Who is responsible for patching, upgrading, and maintaining the worker nodes in Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE)? (Choose the best answer.)
* Oracle Support
* It is automated
* The user
* Independent Software Vendors

The user is responsible for patching, upgrading, and maintaining the worker nodes in Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE). In OKE, the user has control over the worker nodes, which are the compute instances that run the Kubernetes worker components. As the user, you are responsible for managing and maintaining these worker nodes, including tasks such as patching the underlying operating system, upgrading Kubernetes versions, and performing any necessary maintenance activities. While Oracle provides the underlying infrastructure and support services, including managing the control plane and ensuring the availability of the OKE service, the responsibility for managing the worker nodes lies with the user. This allows you to have control and flexibility in managing your Kubernetes environment according to your specific needs and requirements.

NO.23 Which of the following step is NOT required for setting up the Container Engine for Kubernetes (OKE) cluster access using a local installation of kubectl?
* Set up the kubeconfig file.
* Install and configure the Oracle Cloud Infrastructure (OCI) CLI.
* Generate an API signing key pair (if you do not already have one) and upload the public key of the API signing key pair.
* Generate Auth token from the OCI console to access the OKE cluster using kubectl.

Explanation

The step that is NOT required for setting up the Container Engine for Kubernetes (OKE) cluster access using a local installation of kubectl is to generate an Auth token from the OCI console. The authentication for accessing the OKE cluster using kubectl can be performed using the OCI CLI configuration, specifically the API signing key pair and the kubeconfig file. Here are the correct steps for setting up the OKE cluster access using a local installation of kubectl: Set up the kubeconfig file: The kubeconfig file contains the necessary information to authenticate and access the OKE cluster using kubectl. It includes details such as the cluster endpoint, authentication method, and credentials. Generate an API signing key pair (if you do not already have one) and upload the public key of the API signing key pair: The API signing key pair is used for authentication with the OCI services. The public key of the key pair needs to be uploaded to the OCI Console to associate it with your user account. Install and configure the Oracle Cloud Infrastructure (OCI) CLI: The OCI CLI provides a command-line interface to interact with the OCI services. It needs to be installed and configured with your OCI credentials, including the user’s OCID, tenancy OCID, region, and the path to the API signing key pair. By completing these steps,you can configure kubectl to access and manage your OKE clusters from your local machine using the OCI CLI authentication configuration.

NO.24 Which kubectl command syntax is valid for implementing a rolling update deployment strategy in Kubernetes? (Choose the best answer.)
* kubectl update <deployment-name> –image=image:v2
* kubectl update -c <container> –iniage=image: v2
* kubectl rolling-update <deployment-name> –image=image:v2
* kubectl upgrade -c <container> –image=image:v2

Explanation

The correct syntax for implementing a rolling update deployment strategy in Kubernetes using the kubectl command is: kubectl rolling-update <deployment-name> –image=image:v2 This command initiates a rolling update of the specified deployment by updating the container image to image:v2. The rolling update strategy ensures that the new version of the application is gradually deployed while maintaining availability and minimizing downtime.

NO.25 You are using Oracle Cloud Infrastructure (OCI) Resource Manager to manage your infrastructure lifecycle and wish to receive an email each time a Terraform action begins. How should you use the OCI Events service to do this without writing any code?
* Create a rule in OCI Events service matching the “Resource Manager Stack – Update” condition. Then select “Action Type: Email” and provide the destination email address.
* Create an OCI Notifications topic and email subscription with the destination email address. Then create an OCI Events rule matching “Resource Manager Stack – Update” condition, and select the notification topic for the corresponding action.
* Create an OCI Notification topic and email subscription with the destination email address. Then create an OCI Events rule matching “Resource Manager Job – Create” condition, and select the notification topic for the corresponding action.
* Create an OCI Email Delivery configuration with the destination email address. Then create an OCI Events rule matching “Resource Manager Job – Create” condition, and select the email configuration for the corresponding action.
Explanation

The correct approach to receive an email each time a Terraform action begins in Oracle Cloud Infrastructure (OCI) Resource Manager without writing any code is as follows: Create an OCI Notification topic and email subscription with the destination email address. This will define the email delivery configuration. Create an OCI Events rule that matches the “Resource Manager Job – Create” condition. This rule will be triggered when a Resource Manager job is created. In the OCI Events rule, select the notification topic that was created in step

1 as the action for the corresponding event. This will ensure that the notification is sent to the specified email address. By following these steps, you can configure the OCI Events service to send an email notification whenever a Resource Manager job is created in OCI Resource Manager.

NO.26 What are the TWO main reasons you would choose to implement a serverless architecture? (Choose two.)
* Easier to run long-running operations
* Improved in-function state management
* Reduced operational cost
* Automatic horizontal scaling
* No need for integration testing
Explanation

The two main reasons to choose a serverless architecture are: Automatic horizontal scaling: Serverless architectures allow for automatic scaling of resources based on demand. The infrastructure automatically provisions and scales resources as needed, ensuring that applications can handle varying workloads efficiently.

This eliminates the need for manual scaling and optimizes resource utilization. Reduced operational cost:

Serverless architectures follow a pay-per-use model, where you are billed only for the actual execution time and resources consumed by your functions. This leads to cost savings as you don’t have to pay for idle resources. Additionally, serverless architectures remove the need for managing and maintaining servers, reducing operational overhead and associated costs.Explanation: No need for integration testing: Integration testing is still necessary in serverless architectures to ensure that functions integrate correctly with other components and services. Serverless functions can interact with various event sources,

databases, and APIs, and testing is required to verify the integration points. Improved in-function state management: Serverless architectures typically encourage stateless functions that operate on short-lived requests or events. While there are mechanisms to manage state within a function, serverless architectures are designed to be stateless by default, promoting scalability and fault tolerance. Easier to run long-running operations: Serverless functions are generally designed for short-lived operations rather than long-running tasks. If you have a requirement for long-running operations, a serverless architecture may not be the ideal choice, as it has execution time limits and may not provide the necessary resources for extended execution.

NO.27 (CHK_4>2) You have a scenario where a DevOps team wants to store secrets in Oracle Cloud Infrastructure (OCI) Vault so that it can inject the secrets into an app’s environment variables (for example, MYSQL_DB_PASSWD) at deployment time. Which is NOT valid about managing secrets in the OCI Vault service?
* New secret versions automatically expire in 90 days unless you configure an expiry rule.
* You can manually create new secrets as well as new secret versions using the OCI Console:
* A secret reuse rule prevents the use of secret contents across different versions of a secret.
* A unique OCID is automatically generated for each secret and remains unchanged even when creating a new secret version.
Explanation

The correct answer is: “A unique OCID is automatically generated for each secret and remains unchanged even when creating a new secret version.” The statement that is NOT valid about managing secrets in the OCI Vault service is: “A unique OCID is automatically generated for each secret and remains unchanged even when creating a new secret version.” In OCI Vault, a secret is identified by its OCID (Oracle Cloud Identifier), which is a unique identifier for each resource inOracle Cloud Infrastructure. However, when a new secret version is created for an existing secret, the OCID remains the same for the secret itself, but a new OCID is generated for the secret version. This allows you to track and manage different versions of a secret while maintaining a consistent OCID for the secret itself. The other statements mentioned are valid: You can manually create new secrets as well as new secret versions using the OCI Console. This means you have control over creating and managing secrets within the Vault service. A secret reuse rule prevents the use of secret contents across different versions of a secret. This ensures that each secret version maintains its own unique set of contents and avoids accidental reuse or sharing of secrets across versions. By default, new secret versions automatically expire in 90 days unless you configure an expiry rule. This helps enforce good security practices by automatically rotating secrets periodically, reducing the risk of unauthorized access in case of compromise. Therefore, the statement that is NOT valid is the one regarding the uniqueness and consistency of the OCID when creating new secret versions.

NO.28 From a DevOps process standpoint, it is a good practice to keep changes to an application under version control. Which of the following allows changes to a Docker image to be stored in a version control system?
* Updating docker-compose.yml
* Executing docker commit
* Executing docker save
* Updating Dockerfile
The option that allows changes to a Docker image to be stored in a version control system is: docker commit The docker commit command is used to create a new image from a container’s changes. It takes a running container as input, captures the changes made to it, and creates a new image with those changes. This new image can then be tagged and pushed to a registry, or saved locally. By using docker commit, you can effectively capture the changes made to a container as a new image and store it in a version control system along with the Dockerfile and other project files. This allows for reproducibility and traceability of changes to the Docker image over time.

NO.29 You are developing a polyglot serverless application using Oracle Functions. Which language cannot be used to write your function code?
* PL/SQL
* Python
* Node.js
* Go

* Java

Oracle Functions does not currently support PL/SQL as a language for writing function code. PL/SQL is a procedural language used in Oracle Database for developing stored procedures, triggers, and other database-related code. However, Oracle Functions supports several other popular programming languages such as Go, Node.js, Python, and Java, allowing developers to choose the language that best suits their application requirements and their familiarity with the language. While PL/SQL is powerful for working with the Oracle Database, it is not an option for writing function code in the Oracle Functions serverless architecture.

NO.30 What are the TWO main reasons you would choose to implement a serverless architecture? (Choose two.)

* No need for integration testing
* Automatic horizontal scaling
* Easier to run long-running operations
* Reduced operational cost
* Improved in-function state management

The two main reasons to choose a serverless architecture are: Automatic horizontal scaling: Serverless architectures allow for automatic scaling of resources based on demand. The infrastructure automatically provisions and scales resources as needed, ensuring that applications can handle varying workloads efficiently. This eliminates the need for manual scaling and optimizes resource utilization. Reduced operational cost: Serverless architectures follow a pay-per-use model, where you are billed only for the actual execution time and resources consumed by your functions. This leads to cost savings as you don’t have to pay for idle resources. Additionally, serverless architectures remove the need for managing and maintaining servers, reducing operational overhead and associated costs. No need for integration testing: Integration testing is still necessary in serverless architectures to ensure that functions integrate correctly with other components and services. Serverless functions can interact with various event sources, databases, and APIs, and testing is required to verify the integration points. Improved in-function state management: Serverless architectures typically encourage stateless functions that operate on short-lived requests or events. While there are mechanisms to manage state within a function, serverless architectures are designed to be stateless by default, promoting scalability and fault tolerance. Easier to run long-running operations: Serverless functions are generally designed for short-lived operations rather than long-running tasks. If you have a requirement for long-running operations, a serverless architecture may not be the ideal choice, as it has execution time limits and may not provide the necessary resources for extended execution.

NO.31 A Docker image consists of one or more layers, each of which represents a Dockerfile instruction. The layers are stacked and each one is a delta of the changes from the previous layer. What permission is associated with these layers?

* read mostly
* write only
* movable
* read only
* write once

The correct answer is: “read only.” The layers of a Docker image are read-only. Once a layer is created, it cannot be modified. Each layer represents a Dockerfile instruction, and it is stacked on top of the previous layer, forming a stack of immutable layers. These layers are designed to be read-only to ensure consistency and integrity of the image. When a Docker image is built, each instruction in the Dockerfile creates a new layer. Each layer represents the changes made by that instruction relative to the previous layer. The layers are stacked on top of each other to form the complete image. This layer-based approach allows for efficient storage and distribution of Docker images. Because the layers are read-only, any changes or modifications to the image result in the creation of new layers rather than modifying the existing ones. This immutability ensures that each layer remains intact and preserves the integrity of the image. It also enables Docker’s caching mechanism, where previously built layers can be reused if the corresponding instructions haven’t changed, speeding up the image build process. The other options mentioned, such as “write only,” “write once,” “movable,” and “read mostly,” do not accurately describe the permission associated with Docker image layers. Docker image layers are specifically designed to be read-only.

NO.32 As a developer, you have been tasked with implementing a microservices-based application. Which THREE technologies are best suited to accomplish the task? (Choose three.)

* Terraform
* Big Data
* Anomaly Detection
* Service Mesh
* Docker
* Kubemetes

The three technologies best suited for implementing a microservices-based application are: Service Mesh: A service mesh is a dedicated infrastructure layer that provides features like service discovery, load balancing, encryption, authentication, and observability for microservices. It helps in managing the communication and interactions between microservices in a scalable and secure manner. Kubernetes: Kubernetes is an open-source container orchestration platform that enables the deployment, scaling, and management of containerized applications. It provides features like automated scaling, service discovery, load balancing, and self-healing capabilities, which are essential for managing microservices in a distributed environment. Docker: Docker is a popular containerization platform that allows packaging applications and their dependencies into lightweight containers. It provides a consistent and portable environment for running microservices, enabling easy deployment and scalability. Docker also facilitates isolation and resource efficiency, making it an ideal choice for deploying microservices. While Big Data, Anomaly Detection, and Terraform are valuable technologies, they are not specifically focused on enabling the implementation of microservices-based applications.

NO.33 Oracle Functions monitors all deployed functions and collects and reports various metrics. Which is NOT available when viewing the Application metrics in the Oracle Cloud Infrastructure (OCI) Console?

* The length of time a function runs for.
* The number of retries made by the function before failing due to an error.
* The number of requests to invoke a function that failed due to throttling.
* The number of requests to invoke a function that failed with an error response.

The option that is NOT available when viewing the Application metrics in the Oracle Cloud Infrastructure (OCI) Console is: “The number of retries made by the function before failing due to an error.” When viewing the Application metrics in the OCI Console for Oracle Functions, you can typically see metrics related to the performance and usage of your functions. These metrics provide insights into how your functions are performing and being utilized. The following metrics are usually available: The number of requests to invoke a function that failed due to throttling: This metric indicates the number of requests that were not processed by the function due to reaching the configured concurrency limit or throttling settings. The length of time a function runs for: This metric represents the duration of each function invocation, measuring the time it takes for the function to complete its execution. The number of requests to invoke a function that failed with an error response: This metric counts the number of requests that encountered an error during the function invocation, resulting in a failed response. However, the number of retries made by the function before failing due to an error is not typically available as part of the Application metrics in the OCI Console. The retries made by the function are usually handled at the invoker level, and the specific details of retries may not be captured as part of the application-level metrics. It’s important to note that the availability of metrics and their specific details may vary depending on the version and configuration of Oracle Functions and the monitoring setup. It is recommended to refer to the Oracle Functions documentation and consult the official documentation for accurate and up-to-date information on available metrics.

NO.34 Which TWO are required to access the Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE) cluster from the kubectl CLI? (Choose two.)

* Install and configure the OCI CLI.
* OCI Identity and Access Management (IAM) Auth Token.
* Tiller enabled on the OKE cluster.
* An SSH key pair with the public key added to the cluster worker nodes.
* A configured OCI API signing key pair.

Explanation

The correct options are: A configured OCI API signing key pair: The API signing key pair is used for authentication and

authorization to access OCI resources, including the OKE cluster. The private key should be configured on your local machine to authenticate API requests. An SSH key pair with the public key added to the cluster worker nodes: This is required for secure SSH access to the worker nodes in the OKE cluster.

You need to generate an SSH key pair and add the public key to the cluster’s worker node pool during cluster creation or update. Therefore, the correct options are having a configured OCI API signing key pair and an SSH key pair with the public key added to the cluster worker nodes.

NO.35 Having created a Container Engine for Kubernetes (OKE) cluster, you can use Oracle Cloud Infrastructure (OCI) Logging to view and search the logs of applications running on the worker node compute instances in the cluster. Which task is NOT required to collect and parse applicationlogs? (Choose the best answer.)
* Set the OCI Logging option to Enabled for the cluster.
* Configure a custom log in OCI Logging with the appropriate agent configuration.
* Create a dynamic group with a rule that includes all worker nodes In the cluster.
* Enable monitoring for all worker nodes in the cluster.

Explanation

The correct answer is: Enable monitoring for all worker nodes in the cluster. Enabling monitoring for all worker nodes in the cluster is not required to collect and parse application logs using Oracle Cloud Infrastructure (OCI) Logging. Monitoring is a separate feature that allows you to collect metrics and monitor the health and performance of the worker nodes. To collect and parse application logs, you need to perform the following tasks: Set the OCI Logging option to Enabled for the cluster: This enables the OCI Logging service for the cluster. Create a dynamic group with a rule that includes all worker nodes in the cluster: This helps in targeting the logs generated by the worker nodes. Configure a custom log in OCI Logging with the appropriate agent configuration: This involves specifying the log source, log path, and log format to parse and collect the application logs. By completing these tasks, you can collect and parse the application logs generated by the applications running on the worker node compute instances in the OKE cluster.

NO.36 Your team has chosen to use master encryption key (MEK) within an Oracle Cloud Infrastructure (OCI) Vault for encrypting Kubernetes secrets associated with your microservice deployments in OCI Container Engine for Kubernetes (OKE) clusters so that you can easily manage key rotation. Which of the following is NOT valid about rotating keys in the OCI Vault service?
* Once rotated, older key versions can be used for encryption until they are deleted.
* Both software and HSM-protected MEKS can be rotated.
* When you rotate an MEK, a new key version is automatically generated.
* Each key version is tracked internally with separate unique OCIDS.

The correct answer is: “Once rotated, older key versions can be used for encryption until they are deleted.” The statement that is NOT valid about rotating keys in the OCI Vault service is: “Once rotated, older key versions can be used for encryption until they are deleted.” In the OCI Vault service, when you rotate a master encryption key (MEK), a new key version is automatically generated. However, once a key is rotated and a new version is created, the older key versions are no longer usable for encryption. The purpose of key rotation is to ensure that the encryption keys are regularly updated and that older keys are no longer used to protect sensitive data. This enhances security by minimizing the impact of potential key compromises. The other statements mentioned are valid: Both software and hardware security module (HSM)-protected MEKs can be rotated. This provides flexibility in choosing the type of MEK and ensures that key rotation can be performed regardless of the encryption method used. Each key version is tracked internally with separate unique OCIDs (Oracle Cloud Identifiers). This allows for easy management and tracking of different key versions within the OCI Vault service. In summary, the statement that is NOT valid is the one suggesting that older key versions can still be used for encryption until they are deleted. Key rotation is designed to ensure the use of the latest key version and to retire older key versions to enhance security.

NO.37 As a cloud-native developer, you are designing an application that depends on Oracle Cloud Infrastructure (OCI) Object Storage wherever the application is running. Therefore, provisioning of storage buckets should be part of your Kubernetes deployment process for the application. Which of the following should you leverage to meet this requirement? (Choose the best

answer.)
* Oracle Functions
* OCI Service Broker for Kubernetes
* Open Service Broker API
* OCI Container Engine for Kubernetes

To provision storage buckets as part of your Kubernetes deployment process for an application that depends on Oracle Cloud Infrastructure (OCI) Object Storage, you should leverage the OCI Service Broker for Kubernetes. OCI Service Broker for Kubernetes enables you to provision and manage OCI resources, including Object Storage buckets, directly from Kubernetes. It provides a Kubernetes-native experience for managing OCI services, allowing you to define and manage OCI resources as part of your application deployment process. By using the OCI Service Broker for Kubernetes, you can define the required Object Storage buckets in your Kubernetes manifests, and the service broker will handle the provisioning and management of those buckets in OCI, ensuring that they are available for your application wherever it is running.

NO.38 Which is NOT a valid backend-type option available when configuring an Oracle Cloud Infrastructure (OCI) API Gateway Deployment?
* HTTP_BACKEND
* ORACLE STREAMS_BACKEND
* ORACLE_FUNCTIONS_BACKEND

Explanation

When configuring an OCI API Gateway deployment, you need to specify the backend type for each route in your API deployment specification3. The backend type determines how the API gateway handles requests to that route and forwards them to the appropriate backend service3. The following backend types are valid options for an OCI API Gateway deployment3:

* HTTP_BACKEND: The API gateway forwards requests to an HTTP or HTTPS URL as the backend service.

* ORACLE_FUNCTIONS_BACKEND: The API gateway invokes an Oracle Functions function as the backend service.

* STOCK_RESPONSE_BACKEND: The API gateway returns a stock response without invoking any backend service. ORACLE STREAMS_BACKEND is not a valid backend type for an OCI API Gateway deployment. Oracle Streams is a fully managed, scalable, and durable messaging service that you can use to ingest and consume large amounts of data in real-time4. However, Oracle Streams is not supported as a backend service for an OCI API Gateway deployment.

NO.39 Assuming that your function does NOT have the –provisioned-concurrency option enabled, which parameter is used to configure the time period during which an idle function will remain in memory before Oracle Functions removes its container image from memory?
* idle-timeout
* access-timeout
* timeout
* None, as this time is not configurable.

Explanation

Idle-timeout is the parameter that is used to configure the time period during which an idle function will remain in memory before Oracle Functions removes its container image from memory2. The idle-timeout parameter is specified in seconds and can be set when creating or updating a function2. The default value for idle-timeout is 30 seconds and the maximum value is 900 seconds (15 minutes)2. If a function has the

–provisioned-concurrency option enabled, the idle-timeout parameter is ignored and the function instances are always kept in memory3.Verified References: Creating Functions, Provisioned Concurrency

NO.40 You are a developing a microservices application that will be a consumer of the Oracle CloudInfrastructure (OCI) Streaming service. Which API method should you use to read and process a stream?
* GetMessages
* GetStream
* ReadMessages
* ReadStream
* ProcessStream

Explanation

The correct API method to read and process a stream in the Oracle Cloud Infrastructure (OCI) Streaming service is “GetMessages”. When consuming messages from a stream in OCI Streaming, you use the

“GetMessages” API method. This method allows you to retrieve a batch of messages from the stream for processing. You can specify parameters such as the number of messages to retrieve, the maximum size of the messages, and the timeout for the request. By using the “GetMessages” API method, you can retrieve messages from the stream and then process them in your microservices application. This allows you to consume and handle the data in real-time as it becomes available in the stream. The “GetMessages” method provides flexibility in how you consume and process the messages, enabling you to implement custom logic and workflows based on your specific application requirements.

NO.41 Which testing strategy achieves high velocity of deployments and releases of cloud native applications? (Choose the best answer.)
* Penetration testing
* Automated testing
* Integration testing
* A/B testing

The testing strategy that achieves high velocity of deployments and releases of cloud native applications is “Automated testing.” Automated testing involves the use of automated tools and frameworks to execute tests, validate functionality, and detect issues or bugs in an application. By automating the testing process, developers and DevOps teams can rapidly test and validate code changes, ensuring that new features and updates are functioning correctly before being deployed to production. This approach helps increase the speed and efficiency of the testing process, allowing for faster and more frequent deployments of cloud native applications.

NO.42 You are tasked with developing an application that requires the use of Oracle Cloud Infrastructure (OCI) APIs to POST messages to a stream in the OCI Streaming service. Which statement is incorrect? (Choose the best answer.)
* The Content-Type header must be set to application/json
* The request must include an authorization signing string including (but not limited to) x-content-sha256, content-type, and content-length headers.
* The request does not require an Authorization header.
* An HTTP 401 will be returned if the client’s clock is skewed more than 5 minutes from the server’s.

The statement that is incorrect is: “The request does not require an Authorization header.” In order to POST messages to a stream in the OCI Streaming service using OCI APIs, the request does require an Authorization header. The Authorization header is used to provide authentication and ensure the request is authorized to access the stream. The correct approach is to include the Authorization header in the request, along with other required headers such as x-content-sha256, content-type, and content-length. Therefore, the incorrect statement is that the request does not require an Authorization header.

Use Real Dumps - 100% Free 1z0-1084-23 Exam Dumps:
https://www.examcollectionpass.com/Oracle/1z0-1084-23-practice-exam-dumps.html]