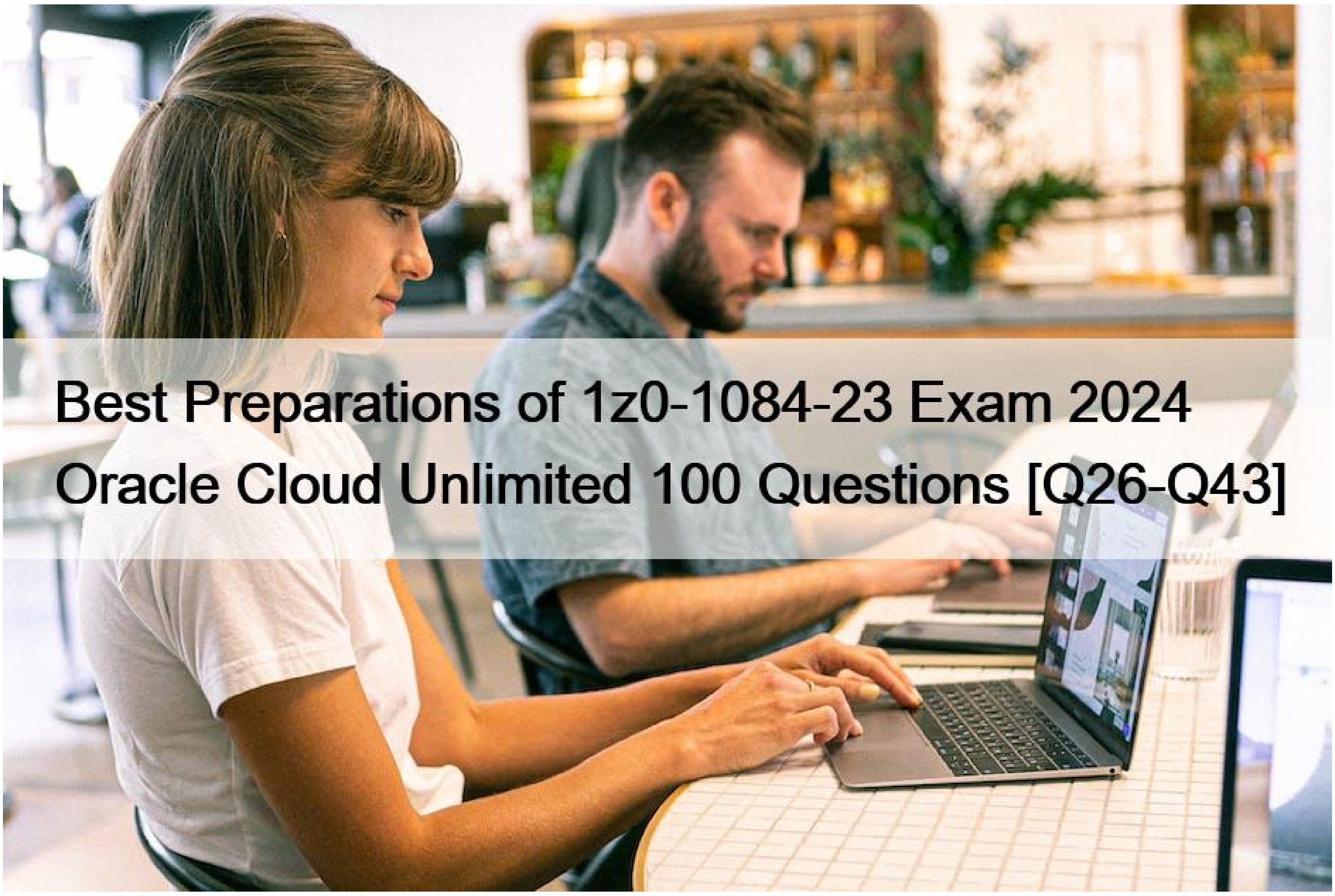
Best Preparations of 1z0-1084-23 Exam 2024 Oracle Cloud Unlimited 100 Questions [Q26-Q43



Best Preparations of 1z0-1084-23 Exam 2024 Oracle Cloud Unlimited 100 Questions
Focus on 1z0-1084-23 All-in-One Exam Guide For Quick Preparation.

NEW QUESTION 26

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 &#8220;GetMessages&#8221;. When consuming messages from a stream in OCI Streaming, you use the

&#8220;GetMessages&#8221; 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 &#8220;GetMessages&#8221; 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 &#8220;GetMessages&#8221; 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.

NEW QUESTION 27

You are developing a distributed application and you need a call to a path to always return a specific JSON content deploy an OCI API Gateway with the below API deployment specification. What is the correct value for type? { &#8220;routes&#8221; : [{ &#8220;path&#8221; : &#8220;/hello&#8221;, &#8220;methods&#8221; : [&#8220;Get&#8221;), &#8220;backend&#8221; : { &#8220;type&#8221; : &#8221; &#8212;&#8212;&#8212;&#8212;&#8212;- &#8220;, &#8220;status&#8221;: 200, &#8220;headers&#8221; : [{ &#8220;name&#8221; : &#8220;Content-Type&#8221;, &#8220;value&#8221; : &#8220;application/json&#8221; }] &#8220;body&#8221; : &#8220;{&#8220;myjson&#8221;:

&#8220;consistent response&#8221;}&#8221; }}]}
* JSON_BACKEND
* STOCK_RESPONSE_BACKEND
* CONSTANT_BACKEND
* HTTP_BACKEND
Explanation

The correct value for the &#8220;type&#8221; field in the API deployment specification is

&#8220;STOCK_RESPONSE_BACKEND&#8221;. By setting the &#8220;type&#8221; to &#8220;STOCK_RESPONSE_BACKEND&#8221;, you are indicating that the backend for the specified route should return a pre-defined response. This type of backend is commonly used when you want a specific response to be returned consistently, regardless of the actual backend service implementation. In this case, the API deployment specification is configured to have a single route with the path &#8220;/hello&#8221; and the method &#8220;GET&#8221;. The backend section specifies the type as

&#8220;STOCK_RESPONSE_BACKEND&#8221;. Additionally, it defines the response status code as 200, setsthe

&#8220;Content-Type&#8221; header to &#8220;application/json&#8221;, and provides the JSON content in the &#8220;body&#8221; field. Using this configuration, any request to the &#8220;/hello&#8221; path with the &#8220;GET&#8221; method will always receive a consistent JSON response with the content &#8220;{&#8220;myjson&#8221;: &#8220;consistent response&#8221;}&#8221;.

NEW QUESTION 28

Which feature is typically NOT associated with Cloud Native?
* Declarative APIs
* Containers
* Application Servers
* Immutable Infrastructure
* Service Meshes
Explanation

The feature that is typically NOT associated with Cloud Native is &#8220;Application Servers.&#8221; Cloud Native architecture emphasizes lightweight, scalable, and containerized deployments, which often replace traditional monolithic application servers.

Instead of relying on application servers, Cloud Native applications are typically deployed as containerized microservices that can be orchestrated and managed using container orchestration platforms like Kubernetes. This approach enables greater flexibility, scalability, and agility in deploying and managing applications. While application servers have been widely used in traditional application architectures, they are not a characteristic feature of Cloud Native architectures. Cloud Native architectures focus on containerization, declarative APIs, immutable infrastructure, and service meshes to enable efficient and scalable deployment and management of applications.

NEW QUESTION 29

To effectively test your cloud native applications for &#8220;unknown unknowns&#8221;, you need to employ various testing and deployment strategies. Which strategy involves exposing new functionality or features to only a small set of users?

* Canary Deployment
* Component Testing
* A/B Testing
* Blue/Green Deployment

Explanation

The strategy that involves exposing new functionality or features to only a small set of users is called Canary Deployment. Canary deployment is a technique used in software development and deployment where a new version of an application or feature is released to a small subset of users or a specific group of servers. This allows for testing and gathering feedback on the new functionality in a controlled and limited environment before making it available to a wider audience. In a canary deployment, a small portion of the traffic is routed to the new version while the majority of the traffic still goes to the stable version. This allows for monitoring and evaluation of the new functionality in real-world conditions while minimizing the impact of any potential issues or bugs. If the new version performs well and meets the desired criteria, it can then be gradually rolled out to a larger user base or all servers. By exposing the new functionality or features to a small set of users initially, canary deployment helps in identifying any unforeseen issues, gathering feedback, and ensuring the stability and reliability of the application before a full deployment.

NEW QUESTION 30

You are building a container image and pushing it to Oracle Cloud Infrastructure Registry (OCIR). You need to ensure that these images never get deleted from the repository. Which action should you take?

* Write a policy to limit access to the specific repository in your compartment.
* Create a group and assign a policy to perform lifecycle operations on images.
* Set global policy of image retention to &#8220;Retain All Images&#8221;.
* Edit the tenancy global retention policy.

The correct answer is: &#8220;Edit the tenancy global retention policy.&#8221; To ensure that container images never get deleted from the Oracle Cloud Infrastructure Registry (OCIR), you should edit the tenancy global retention policy. The tenancy global retention policy is a setting that determines the retention behavior for all the images in the OCIR across the entire tenancy. By editing this policy, you can define the retention behavior that suits your requirements. To edit the tenancy global retention policy, you would typically perform the following steps: Access the Oracle Cloud Infrastructure Console and navigate to the OCIR service. Go to the &#8220;Policies&#8221; section or &#8220;Settings&#8221; section in the OCIR service. Locate the tenancy global retention policy settings. Modify the retention policy to specify the desired retention behavior. In this case, you would set the policy to retain all images, ensuring they are never deleted from the repository. By setting the global policy of image retention to &#8220;Retain All Images,&#8221; you can ensure that the container images in your OCIR repository are permanently retained and not subject to deletion based on any default or automatic retention rules. The other options mentioned are not directly related to ensuring that container images are never deleted from the repository: Creating a group and assigning a policy to perform lifecycle operations on images or writing a policy to limit access to the specific repository in your compartment are access control measures and do not address the retention of images. Setting the global policy of image retention to &#8220;Retain All Images&#8221; is the correct action to achieve the desired outcome of preventing image deletion from the repository.

NEW QUESTION 31

You want to push a new image in the Oracle Cloud Infrastructure (OCI) Registry. Which TWO actions would you need to perform? (Choose two.)
* Generate an API signing key to complete the authentication via Docker CLI.
* Generate an auth token to complete the authentication via Docker CLI.
* Assign an OCI defined tag via OCI CLI to the image.
* Assign a tag via Docker CLI to the image.
* Generate an OCI tag namespace in your repository.
To push a new image to the Oracle Cloud Infrastructure (OCI) Registry, you would need to perform the following two actions: Assign a tag via Docker CLI to the image: Before pushing the image, you need to assign a tag to it using the Docker CLI. The tag helps identify the image and associate it with a specific version or label. Generate an auth token to complete the authentication via Docker CLI: To authenticate and authorize the push operation, you need to generate an auth token. This token is used to authenticate your Docker CLI with the OCI Registry, allowing you to push the image securely. Note: Generating an API signing key, assigning an OCI defined tag via OCI CLI, and generating an OCI tag namespace are not required steps for pushing a new image to the OCI Registry.

NEW QUESTION 32

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.

NEW QUESTION 33

Which kubectl command syntax is valid for implementing a rolling update deployment strategy in Kubernetes? (Choose the best answer.)
* kubectl update <deployment-name> &#8211;image=image:v2
* kubectl update -c <container> &#8211;iniage=image: v2
* kubectl rolling-update <deployment-name> &#8211;image=image:v2
* kubectl upgrade -c <container> &#8211;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> &#8211;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.

NEW QUESTION 34

Which is the smalled unit of Kubernetes architecture?
* Node
* Pod
* Cluster
* Container
Explanation

The smallest unit of Kubernetes architecture is a Pod. A Pod is a logical grouping of one or more containers that are deployed together on the same host and share the same network namespace, storage, and other resources. It represents the smallest deployable unit in Kubernetes and is used to encapsulate and manage one or more closely related containers. Containers within a Pod are scheduled and deployed together, allowing them to communicate and share resources efficiently.

NEW QUESTION 35

Which concept in OCI Queue is responsible for hiding a message from other consumers for a predefined amount of time after it has been delivered to a consumer?
* Maximum retention period
* Visibility timeout
* Delivery count
* Polling timeout
Visibility timeout is the concept in OCI Queue that is responsible for hiding a message from other consumers for a predefined amount of time after it has been delivered to a consumer1. The visibility timeout can be set at the queue level when creating a queue, or it can be specified when consuming or updating messages1. If a consumer is having difficulty successfully processing a message, it can update the message to extend its invisibility1. If a message&#8217;s visibility timeout is not extended, and the consumer does not delete the message, it returns to the queue1. Verified Reference: Overview of Queue

NEW QUESTION 36

Which TWO statements are true for serverless computing and serverless architectures? (Choose two.)
* Serverless function execution is fully managed by third party.
* Serverless function state should never be stored externally.
* Application DevOps team is responsible for scaling.
* Applications running on a FaaS (Functions as a Service) platform.
* Long running tasks are perfectly suited for serverless.
Explanation

The two true statements for serverless computing and serverless architectures are: Applications running on a FaaS (Functions as a Service) platform: Serverless architectures typically involve running code in the form of functions on a serverless platform. These functions are event-driven and executed in response to specific triggers or events. Serverless function execution is fully managed by a third party: In serverless computing, the cloud provider takes care of the infrastructure management and resource provisioning. The execution of serverless functions is handled automatically by the platform, relieving developers from the responsibility of managing servers orinfrastructure. It&#8217;s important to note that long running tasks are not typically suited for serverless architectures due to the event-driven nature of serverless functions. Also, while serverless functions may have state, it is recommended to avoid external storage dependencies and instead leverage stateless functions whenever possible. Additionally, scaling in serverless architectures is typically handled automatically by the platform, rather than being the responsibility of the application DevOps team.

NEW QUESTION 37

You are building a cloud native serverless travel application with multiple Oracle Functions in Java, Python, and Node.js. You need to build and deploy these functions to a single application named travel-app. Which command will help you complete this task successfully?
* fn function deploy app travel-app&#8211;all
* fn app deploy &#8211;app travel-app &#8211;all
* fn app &#8211;app travel-app deploy &#8211;ext java pyljs
* fn deploy&#8211;app travel-app &#8211;all
The correct answer is: fn deploy &#8211;app travel-app &#8211;all To build and deploy multiple Oracle Functions as part of a single application named &#8220;travel-app,&#8221; you can use the fn deploy command with the appropriate options. The command fn deploy &#8211;app travel-app &#8211;all is the correct syntax. Here&#8217;s what each part of the command does: fn deploy: This command is used to deploy functions and applications in Oracle Functions. &#8211;app travel-app: This option specifies the application name as &#8220;travel-app,&#8221; indicating that you want to deploy functions to this application. &#8211;all: This option indicates that you want to deploy all the functions within the application. By using fn deploy &#8211;app travel-app &#8211;all, you can build and deploy all the functions in your travel application across different programming languages (Java, Python, and Node.js) to the &#8220;travel-app&#8221; application in Oracle Functions.

NEW QUESTION 38

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

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.

NEW QUESTION 39

You developed a microservices-based application that runs in an Oracle Cloud Infrastructure (OCI) Container Engine for Kubernetes (OKE) cluster. It has multiple endpoints that need to be exposed to the public internet.

What is the most cost-effective way to expose multiple application endpoints without adding unnecessary complexity to the application?
* Use a NodePort service type in Kubernetes for each of your service endpoints using the node&#8217;s public IP address to access the applications.
* Use a ClusterIP service type in Kubernetes for each of your service endpoints using a load balancer to expose the endpoints.
* Deploy an Ingress Controller and use it to expose each endpoint with its own routing endpoint.

* Create a separate load balancer instance for each service using the lowest 100 Mbps option.
Explanation

An Ingress Controller is a Kubernetes resource that provides advanced routing and load balancing for your applications running on a Kubernetes cluster1. An Ingress Controller allows you to define rules that specify how to route traffic to different services in your cluster based on the host name or path of the incoming request1. By deploying an Ingress Controller and using it to expose multiple application endpoints, you can achieve the following benefits1:

* Cost-effectiveness: You only need to create one load balancer instance per cluster, instead of one per service, which reduces the cost of exposing your applications.

* Simplicity: You only need to manage one set of routing rules for all your services, instead of configuring each service separately, which simplifies the application deployment and maintenance.

* Flexibility: You can use different types of Ingress Controllers, such as NGINX or Traefik, that offer various features and customization options for your routing needs.

NEW QUESTION 40

To effectively test your cloud native applications for &#8220;unknown unknowns&#8221;, you need to employ various testing and deployment strategies. Which strategy involves exposing new functionality or features to only a small set of users?
* A/B Testing
* Component Testing
* Blue/Green Deployment
* Canary Deployment
The strategy that involves exposing new functionality or features to only a small set of users is called Canary Deployment. Canary deployment is a technique used in software development and deployment where a new version of an application or feature is released to a small subset of users or a specific group of servers. This allows for testing and gathering feedback on the new functionality in a controlled and limited environment before making it available to a wider audience. In a canary deployment, a small portion of the traffic is routed to the new version while the majority of the traffic still goes to the stable version. This allows for monitoring and evaluation of the new functionality in real-world conditions while minimizing the impact of any potential issues or bugs. If the new version performs well and meets the desired criteria, it can then be gradually rolled out to a larger user base or all servers. By exposing the new functionality or features to a small set of users initially, canary deployment helps in identifying any unforeseen issues, gathering feedback, and ensuring the stability and reliability of the application before a full deployment.

NEW QUESTION 41

You encounter an unexpected error when invoking Oracle Functions from your Cloud Shell session named myfunction in the myapp application. Which option will get you more information on theerror?
* Contact Oracle support with your error message
* fn &#8211;verbose invoke myapp myfunction
* fn &#8211;debug invoke myapp myfunction
* DEBUG=1 fn invoke myapp myfunction
Explanation

The option that will get you more information on the error when invoking Oracle Functions from your Cloud Shell session is: &#8220;DEBUG=1 fn invoke myapp myfunction&#8221;. Setting the environment variable DEBUG=1 before invoking the function using the fn command allows you to enable debug mode, which provides more detailed information about the execution of the function. This can be useful for troubleshooting and understanding the root cause of the error. By using the command &#8220;DEBUG=1 fn invoke myapp myfunction&#8221;, the function invocation will be executed with debug mode enabled, and additional debug information will be displayed in the console output. This information can include stack traces, detailed error messages, and other relevant details that can help identify and resolve the issue. Using the verbose option (&#8211;verbose) or debug option (&#8211;debug) with the fn command may also provide additional information, but the specific behavior may depend on the version and configuration of the fn CLI tool. While contacting Oracle support with the error message is always an option, enabling debug mode using the DEBUG=1 environment variable provides immediate access to more detailed information and can help in diagnosing and resolving the error more efficiently.

NEW QUESTION 42

Which &#8220;Action Type&#8221; option is NOT available in an Oracle Cloud Infrastructure (OCI) Events rule definition?
* Streaming
* Email
* Notifications
* Functions
An action is a response that you define for the rule to perform when the filter finds a matching event1. The action type specifies the service that you want to invoke by delivering the event message1. The following action types are available in OCI Events rule definition1:

Streaming: Send to a stream from Oracle Streaming Service.

Notifications: Send to an Oracle Notification Service topic.

Functions: Send to an Oracle Functions Service endpoint. Email is not a valid action type for OCI Events rule definition. To send an email as an action, you need to use the Notifications service and subscribe to a topic with an email protocol2.

NEW QUESTION 43

You have two microservices, A and B, running in production. Service A relies on APIs from service B. You want to test changes to service A without deploying all of its dependencies, which include service B. Which approach should you take to test service A?
* Test using API mocks.
* Test the APIs in private environments.
* Test against production APIs.
* There is no need to explicitly test APIs.
API mocking is a technique that simulates the behavior of real APIs without requiring the actual implementation or deployment of the dependent services1. API mocking allows you to test changes to service A without deploying all of its dependencies, such as service B, by creating mock responses for the APIs that service A relies on1. API mocking has several benefits, such as1:

Faster testing: You can test your service A without waiting for service B to be ready or available, which reduces the testing time and feedback loop.

Isolated testing: You can test your service A in isolation from service B, which eliminates the possibility of external factors affecting the test results or causing errors.

Controlled testing: You can test your service A with different scenarios and edge cases by creating mock responses that mimic various situations, such as success, failure, timeout, etc.

Guaranteed Success with 1z0-1084-23 Dumps: https://www.validbraindumps.com/1z0-1084-23-exam-prep.html]