

Prepare Smart for Success Free Oracle 1Z0-1084-25 Exam Questions and Answers

Ready to pass faster? Grab free and updated Oracle Cloud Infrastructure 2025 Developer Professional exam PDF questions now. Get authentic 1Z0-1084-25 dumps packed with verified answers and secure your certification success with [PrepBolt](https://prepbolt.com/1Z0-1084-25.html) 1Z0-1084-25 exam pdf questions and answers.

Thank you for Downloading 1Z0-1084-25 exam PDF Demo

<https://prepbolt.com/1Z0-1084-25.html>

Question 1

Question Type: MultipleChoice

You are developing a real-time monitoring application for a fleet of vehicles, which will be deployed on Oracle Cloud Infrastructure (OCI). You need to choose between using OCI Queue or OCI Streaming to handle the real-time data feeds from the vehicles. Based on the scenario described, which is the most appropriate choice for handling real-time data feeds?

Options:

- A- OCI Streaming, because it is designed for high-volume, continuous ingestion and processing of data, making it the best choice for a fleet of vehicles
- B- OCI Streaming, because it offers exactly-once message delivery, which is necessary for real-time applications
- C- OCI Queue, because it is optimized for low-latency messaging and ideal for real-time applications
- D- OCI Queue, because it provides at-least-once message delivery, which is critical for real-time monitoring applications

Answer:

A

Explanation:

OCI Streaming is a fully managed, scalable, and durable messaging solution for ingesting continuous, high-volume streams of data that you can consume and process in real-time. Streaming is suitable for any use case in which data is produced and processed continually and sequentially in a publish-subscribe messaging model. Streaming can handle millions of messages per second with low latency. Therefore, OCI Streaming is the most appropriate choice for handling real-time data feeds from a fleet of vehicles. Verified Reference: [Overview of Streaming, Container Engine for Kubernetes](#)

Question 2

Question Type: MultipleChoice

Which technique is used for testing the entire user flow as well as the moving parts of a cloud native app, ensuring that there are no high-level discrepancies?

Options:

- A- Contract Testing
- B- Integration Testing
- C- Unit Testing
- D- Component Testing
- E- End-to-end Testing

Answer:

E

Explanation:

End-to-end testing is a technique that involves checking the entire user flow as well as the moving parts of a cloud native app, ensuring that there are no high-level discrepancies³. End-to-end testing simulates real user scenarios and validates the functionality, performance, reliability, and security of the app from start to finish³. End-to-end testing has several benefits, such as³:

Comprehensive testing: You can test your app as a whole and verify that all the components work together as expected.

User-centric testing: You can test your app from the user's perspective and ensure that it meets the user's needs and expectations.

Quality assurance: You can test your app in a realistic environment and identify any issues or defects before releasing it to the users.

Question 3

Question Type: MultipleChoice

Kubernetes includes various elements such as compute, network, and storage. Compute is essentially CPU (units) and memory (bytes). Within an OKE cluster, what is considered to be the smallest unit of deployment with respect to compute?

Options:

- A- Container
- B- Service
- C- Pod
- D- Namespace
- E- Deployment resource

Answer:

C

Explanation:

A pod is the smallest and simplest unit in the Kubernetes object model that you create or deploy. A pod represents a single instance of a running process in your cluster. Pods contain one or more containers, such as Docker containers. When you create a pod, you define how much CPU and memory (RAM) each container needs. A pod can also include storage volumes, IP addresses, options that govern how the container(s) should run, and more. Pods are the basic building blocks of larger Kubernetes constructs such as deployments, replica sets, and services.

Question 4

Question Type: MultipleChoice

In the DevOps lifecycle, what is the difference between continuous delivery and continuous deployment? (Choose two.)

Options:

- A- Continuous delivery involves automation of developer tasks, while continuous deployment involves manual operational tasks.
- B- Continuous delivery utilizes automatic deployment to a development environment, while continuous deployment involves automatic deployment to a production environment.
- C- Continuous delivery requires more automatic linting, while continuous deployment testing must be run manually.
- D- Continuous delivery is a process that initiates deployment manually, while continuous deployment is based on automating the deployment process.

Answer:

B, D

Explanation:

The two correct differences between continuous delivery and continuous deployment in the DevOps lifecycle are: Continuous delivery is a process that initiates deployment manually, while continuous deployment is based on automating the deployment process. In continuous delivery, the software is

ready for deployment, but the decision to deploy is made manually by a human. On the other hand, continuous deployment automates the deployment process, and once the software passes all the necessary tests and quality checks, it is automatically deployed without human intervention. Continuous delivery involves automatic deployment to a development environment, while continuous deployment involves automatic deployment to a production environment. In continuous delivery, the software is automatically deployed to a development or staging environment for further testing and validation. However, the actual deployment to the production environment is performed manually. In continuous deployment, the software is automatically deployed to the production environment, eliminating the need for manual intervention in the deployment process. These differences highlight the level of automation and human involvement in the deployment process between continuous delivery and continuous deployment approaches in the DevOps lifecycle.

Question 5

Question Type: MultipleChoice

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 includes service B. Which approach should you take to test service A?

Options:

- A- Test using a previous test version of service B.
- B- Test using an API mock of service B.
- C- Test using the current production version of service B.
- D- This is not possible because service B is a dependency.

Answer:

B

Explanation:

The correct answer is: Test using an API mock of service B. To test service A without deploying all of its dependencies, including service B, you can use an API mock of service B. An API mock is a simulated version of the API that mimics the behavior of the actual service. By using an API mock, you can isolate the testing of service A and simulate the responses and behavior of service B's APIs. With an API mock, you can define the expected responses and behavior of service B's APIs, allowing you to test the integration between service A and the mocked version of service B. This approach enables you to verify the functionality of service A without relying on the availability or changes in the actual service B. By decoupling the dependencies and using an API mock, you can perform independent testing of service A, ensuring its functionality in isolation.

Question 6

Question Type: MultipleChoice

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 the error?

Options:

- A- `fn --verbose invoke myapp myfunction`
- B- `fn --debug invoke myapp myfunction`
- C- Contact Oracle support with your error message
- D- `DEBUG=1 fn invoke myapp myfunction`

Answer:

D

Explanation:

The option that will get you more information on the error when invoking Oracle Functions from your Cloud Shell session is: `'DEBUG=1 fn invoke myapp myfunction'`. 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 `'DEBUG=1 fn invoke myapp myfunction'`, 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 (`--verbose`) or debug option (`--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.

Thank You for trying 1Z0-1084-25 PDF Demo

To try our 1Z0-1084-25 practice exam software
visit link below

<https://prepbolt.com/1Z0-1084-25.html>

Start Your 1Z0-1084-25 Preparation

Use Coupon "SAVE50" for extra 50% discount on the purchase of
Practice Test Software. Test your 1Z0-1084-25 preparation with actual
exam questions.