



Free Questions for 1Z0-1084-23 by certsinside

Shared by Lee on 07-08-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

In the shared responsibility model, who should perform patching, upgrading, and maintaining of the worker nodes in provisioned Oracle Container Engine for Kubernetes (OKE) clusters?

Options:

- A- Oracle Support does it.
- B- It is the responsibility of the customer.
- C- It is an automated process.

Answer:

B

Explanation:

In the shared responsibility model, Oracle is responsible for securing the underlying cloud infrastructure and platform services, while customers are responsible for securing their data and applications within the cloud⁴. For provisioned OKE clusters, Oracle manages the control plane (master nodes) of the Kubernetes cluster, while customers manage the data plane (worker nodes) of the

cluster5. Therefore, it is the responsibility of the customer to perform patching, upgrading, and maintaining of the worker nodes in provisioned OKE clusters5. Customers can use tools such as Terraform or kubectl to automate these tasks5.

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 discrepancies. End-to-end testing simulates real user scenarios and validates the functionality, performance, reliability, and security of the app from start to finish. End-to-end testing has several benefits, such as:

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

Which of these is a valid use case for OCI Queue?

Options:

- A- Managing network traffic between services
- B- Storing and retrieving large files
- C- Sending real-time streaming data
- D- Building decoupled and scalable systems

Answer:

D

Explanation:

OCI Queue is a fully managed serverless service that helps decouple systems and enable asynchronous operations. Queue handles high-volume transactional data that requires independently processed messages without loss or duplication. A valid use case for OCI Queue is building decoupled and scalable systems, such as event-driven architectures or microservices-based applications. For example, you can use Queue to decouple your application and build an event-driven architecture. Decoupling ensures that individual application components can scale independently and that you can future-proof your design so that as new application components are built, they can publish or subscribe to the queue.

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

Options:

- A- Test using API mocks.
- B- Test the APIs in private environments.

C- Test against production APIs.

D- There is no need to explicitly test APIs.

Answer:

A

Explanation:

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.

Question 6

Question Type: MultipleChoice

What can you use to dynamically make Kubernetes resources discoverable to public DNS servers? (Choose the best answer.)

Options:

- A- kubeDNS
- B- DynDNS
- C- CoreDNS
- D- ExternalDNS

Answer:

D

Explanation:

To dynamically make Kubernetes resources discoverable to public DNS servers, you can use ExternalDNS. ExternalDNS is a Kubernetes add-on that automates the management of DNS records for your Kubernetes services and ingresses. It can be configured to monitor the changes in your Kubernetes resources and automatically update DNS records in a supported DNS provider. By integrating ExternalDNS with your Kubernetes cluster, you can ensure that the DNS records for your services and ingresses are automatically created, updated, or deleted based on changes in your Kubernetes resources. This allows your Kubernetes resources to be discoverable by external systems through public DNS servers.

Question 7

Question Type: MultipleChoice

A DevOps engineer is troubleshooting the Meshifyd application, which is running in an Oracle Cloud Infrastructure (OCI) environment. The engineer has set up the OCI Logging service to store access logs for the application but notices that the logs from the Meshifyd application are not showing up in the logging service. The engineer suspects that there might be an issue with the logging configuration. Which two statements are potential reasons for logs from the Meshifyd application not showing up in the OCI Logging service?

Options:

- A-** The logconfig.json file has incorrect or missing OCID for the custom log in the logobjectId field.
- B-** The OCI Logging service is set up to pre access logs by creating a log group and custom log within the same compartment.
- C-** The logconfig.json file has incorrect or missing information in the application namespace in the paths field.
- D-** The logconfig.json file has incorrect or missing information in the application namespace in the src field.
- E-** The logconfig.json file has incorrect or missing OCID for the custom log group in the logGroupObjectId field.

Answer:

A, E

Explanation:

The logconfig.json file is a configuration file that specifies how the Unified Monitoring Agent collects and uploads custom logs to the OCI Logging service². The logconfig.json file contains an array of objects, each representing a custom log configuration². Each custom log configuration object has the following fields²:

logGroupObjectId: The OCID of the log group where the custom log is stored.

logObjectId: The OCID of the custom log.

paths: An array of paths to files or directories containing the custom logs.

src: A regular expression that matches the files containing the custom logs.

parser: A parser definition that specifies how to parse the custom logs. If the logconfig.json file has incorrect or missing OCID for the custom log in the logObjectId field, or incorrect or missing OCID for the custom log group in the logGroupObjectId field, then the Unified Monitoring Agent will not be able to upload the custom logs to the OCI Logging service². Therefore, these are potential reasons for logs from the Meshifyd application not showing up in the OCI Logging service. Verified Reference: [Unified Monitoring Agent Configuration File](#)

To Get Premium Files for 1Z0-1084-23 Visit

<https://www.p2pexams.com/products/1z0-1084-23>

For More Free Questions Visit

<https://www.p2pexams.com/oracle/pdf/1z0-1084-23>

