PassQueen
a.k.a dumpsqueen.com

Oracle Cloud Infrastructure for Sunbird Ed Specialty - Rel 1

Oracle 1z0-1119-1

Version Demo

Total Demo Questions: 5

Total Premium Questions: 52

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

An infrastructure team manages the OCI resources for Sunbird ED by using Terraform configurations. Before applying a change to the production VCN, the team wants OCI Resource Manager to identify the proposed changes and retain the configuration as a managed deployment definition.

Which two actions should the team perform? Select two.

A. Create or update a Resource Manager stack that contains the Terraform configuration.

B. Run a Resource Manager plan job and review the proposed changes.

C. Push the Terraform configuration to OCI Container Registry.

D. Run an apply job before generating a plan.

E. Use an OCI DevOps Code Repository as the Terraform execution engine.

ANSWER: A B

Explanation:

Create or update an OCI Resource Manager stack containing the Terraform configuration, and run a plan job before running an apply job. A stack is the managed Resource Manager definition for Terraform configuration and associated state. A plan job evaluates the configuration and shows the proposed resource changes without provisioning them.

An apply job performs the actual create, update, or delete operations and should follow review of the plan. OCI DevOps Code Repositories can store source files but do not execute Terraform plans by themselves. OCI Container Registry stores container images rather than infrastructure definitions.

QUESTION NO: 2

In Angular, what would you inject using Declarations in MGModule?

@NgModule({

declarations: [],

})

A. Services

B. Components

C. Modules

D. Styles

ANSWER: B

Explanation:

Components is correct because an Angular NgModule's `declarations` metadata defines the declarables that belong to that module. A component is a declarable: it supplies a view, a selector, and behavior that Angular compiles and uses in templates. Declaring a component associates it with the NgModule so that it is available within that module's compilation context. In NgModule-based Angular applications, each non-standalone component must be declared by exactly one NgModule. The same `declarations` array can also contain directives and pipes, but among the supplied choices, components are the applicable Angular artifact. Although current Angular development commonly favors standalone components and does not require NgModules for many new applications, NgModules remain supported and are important when working with existing NgModule-based code. Angular's NgModule documentation describes declarations as the place

for components, directives, and pipes that belong to the module. See the official Angular NgModules overview and the NgModule API reference for the definition of the `declarations` field.

QUESTION NO: 3

When it comes to the Subbird platform, which open-source platform is specifically employed for the purpose of hosting a discussion forum?

A. FluxBB

B. Discourse

C. NodeBB

ANSWER: B

Explanation:

Discourse is the correct answer because Sunbird's discussion capability is implemented using Discourse, an open-source platform built specifically for modern online conversations and community forums. It provides the core forum functions needed for a learning ecosystem, including topic-based discussions, threaded replies, user notifications, moderation workflows, trust levels, and search. These capabilities enable learners, teachers, and other community members to participate in structured, persistent discussions around educational content and learning activities. Discourse also exposes APIs and supports extensibility, which makes it suitable for integration into a broader platform such as Sunbird while retaining a dedicated forum experience. Sunbird's discussion service source repository identifies its Discourse-based implementation and deployment components, while the Discourse project documents the platform as open-source forum software designed for communities. See the Sunbird Discussion repository and the Discourse project website.

QUESTION NO: 4

A Sunbird Knowlg deployment must store high-volume operational service data and also retain large learning-content packages such as uploaded files and content archives. The design must use the primary storage technologies appropriate to these two different data types.

Which two technologies should be used? Select two.

A. Cassandra

B. Cloud Storage

C. Redis Cluster

D. Apache Druid

E. Neo4j

ANSWER: A B

Explanation:

Cassandra is appropriate for distributed, high-volume operational data used by microservices. **Cloud Storage** is appropriate for large binary and unstructured content objects, avoiding the need to store packages directly in an operational database.

Neo4j is used where graph relationships among concepts and resources must be modeled. Redis is an in-memory data store commonly used for caching, not durable primary storage for content packages. Apache Druid is intended for analytical queries over event-oriented data.

QUESTION NO: 5

Which service key's public key must be configured for validating authentication tokens in

microservices?

A. sunbird-utils

B. Keycloak

C. sunbird-auth

D. sunbird-apimanager-util

ANSWER: B

Explanation:

Keycloak is correct because it acts as the identity and access-management service that authenticates users or clients and issues signed authentication tokens. A microservice receiving a token must verify that the token was issued by the trusted Keycloak realm and has not been altered. It does this by using Keycloak's public signing key to validate the token's cryptographic signature. This signature validation allows services to trust token claims, such as the subject, expiration, audience, and assigned roles, without requiring every service to call the identity service for each request.

Keycloak publishes the public keys used for this purpose through its OpenID Connect discovery and certificate/JWKS endpoints. In a microservices deployment, configuring the appropriate Keycloak realm public key, or retrieving and caching its JWKS keys, establishes the trust relationship required for secure token validation. Key rotation should also be supported so that services can recognize newly published signing keys without interrupting authenticated requests. See Keycloak's OpenID Connect endpoints documentation and its server key management documentation.