DUMPSARENA

CyberArk Sentry Secrets Manager

CyberArk SECRET-SEN

Version Demo

Total Demo Questions: 9

Total Premium Questions: 90

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

Refer to the exhibit.

How can you confirm that the Follower has a current copy of the database?

A. Compare the pg*current*xlog_locationlocation from the Leader to the Follower you need to validate against.

B. Count the number of components in pg*start*replication and compare this to the total number of Followers in the deployment.

C. Validate that the Follower container ID matches the node in the info endpoint on the Leader.

D. Retrieve the credential from a test application on the Leader cluster; then retrieve against the Follower and compare if they are accurate.

ANSWER: A

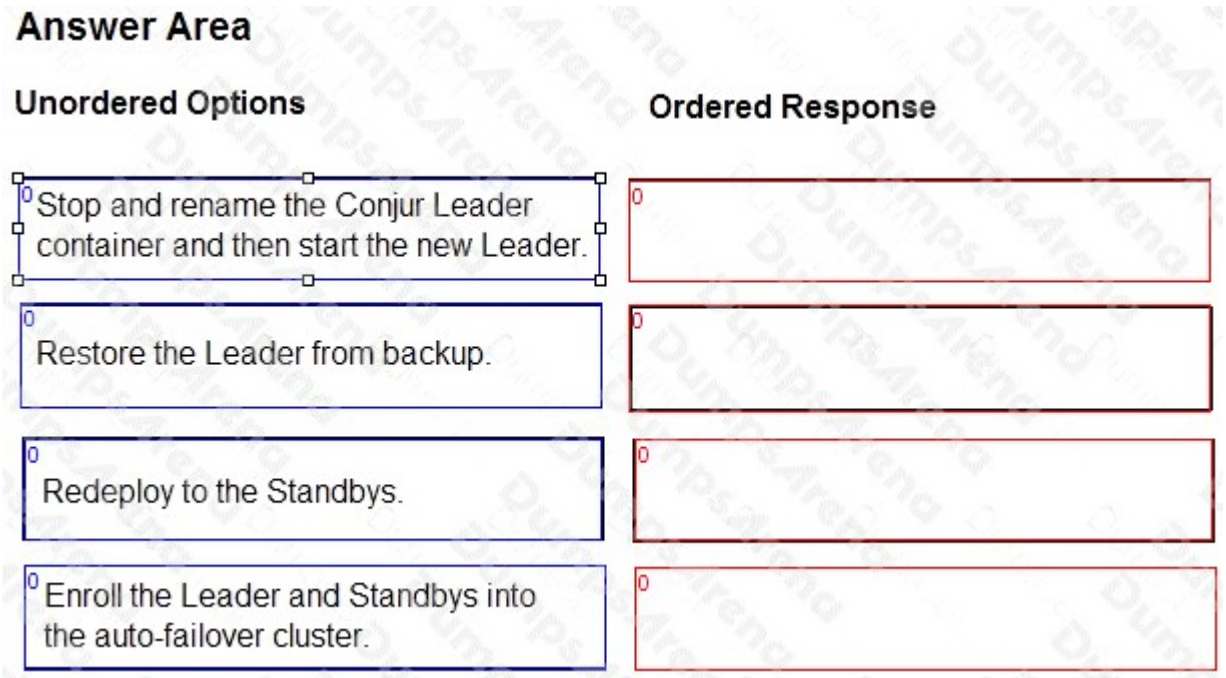
Explanation:

Compare the pg*current*xlog_locationlocation from the Leader to the Follower you need to validate against. This replication-location value represents a position in PostgreSQL's write-ahead log stream, which records database changes before they are applied to database files. In a Secrets Manager cluster, the Leader produces the changes and the Follower receives and replays them. Comparing the reported log positions therefore provides a direct indication of whether the Follower has replayed the Leader's most recent database activity. When the values are the same, or are sufficiently close for the environment's expected replication latency, the Follower has a current usable copy of the database. A material difference indicates that the Follower has not yet replayed all changes produced on the Leader and should be investigated as replication lag. This check is preferable to an application-level functional test because it validates the database replication state itself, including changes that may not be represented by a particular test credential retrieval. CyberArk's Conjur Enterprise documentation covers clustered service operation and health monitoring at CyberArk Conjur Enterprise documentation. PostgreSQL explains the purpose of write-ahead logging and its role in durability and replication at Write-Ahead Logging.

QUESTION NO: 2 - (SIMULATION)

You are upgrading an HA Conjur cluster consisting of 1x Leader, 2x Standbys & 1x Follower. You stopped replication on the Standbys and Followers and took a backup of the Leader.

Arrange the steps to accomplish this in the correct sequence.



ANSWER: Check the explanation for the answer

Explanation:

Correct sequence:

1. Stop and rename the Conjur Leader container and then start the new Leader.
2. Restore the Leader from backup.
3. Redeploy to the Standbys.
4. Enroll the Leader and Standbys into the auto-failover cluster.

The upgraded Leader must be started before its backup can be restored. Standbys are then redeployed against the restored Leader. Finally, after the Leader and Standbys are operating at the upgraded version, enroll them in the auto-failover cluster.

QUESTION NO: 3

Kubernetes workloads authenticate through an `authn-k8s/prod-cluster` authenticator. Applications use a load balancer that distributes requests across Conjur Followers.

Which two conditions must be satisfied on every Follower that can receive authentication requests? (Choose two.)

A. The `authn-k8s/prod-cluster` authenticator is enabled.

B. The Kubernetes API can be reached and trusted for workload-token validation.

C. A Conjur host API key is stored in every application Pod.

D. The Follower is configured as the Kubernetes cluster control-plane node.

E. The authenticator is enabled only on the Conjur Leader.

ANSWER: A B

Explanation:

The Kubernetes authenticator service ID must be enabled on every receiving Follower. In addition, each Follower must have the configuration and trusted connectivity required to validate Kubernetes workload identity through the Kubernetes API.

Providing a host API key in the Pod is not part of the Kubernetes authentication flow. The workload uses its Kubernetes service-account token, and policy defines the host identity to which the authenticated Kubernetes attributes are mapped. Enabling an authenticator only on the Leader does not help when the load balancer sends requests to Followers.

QUESTION NO: 4

While retrieving a secret through REST, the secret retrieval fails to find a matching secret. You know the secret onboarding process was completed, the secret is in the expected safe with the expected object name, and the CCP is able to provide secrets to other applications.

What is the most likely cause for this issue?

A. The application ID or Application Provider does not have the correct permissions on the safe.

B. The client certificate fingerprint is not trusted.

C. The service account running the application does not have the correct permissions on the safe.

D. The OS user does not have the correct permissions on the safe

ANSWER: A

Explanation:

The application ID or Application Provider does not have the correct permissions on the safe. Central Credential Provider authenticates the requesting workload by its configured application identity and then authorizes its access to the Vault safe that contains the requested account or secret. Successful onboarding and the existence of the expected object do not by themselves grant that application identity access to the safe. The application identity must be configured as an authorized safe member with the permissions required to locate and retrieve the account, including the applicable account-listing and retrieval permissions. If those authorizations are absent or were granted to a different application identity, the provider cannot return the requested matching secret even though it can successfully serve other applications. This behavior is consistent with a request-specific authorization issue: the provider service is operational, and the secret exists, but the caller's application identity is not entitled to retrieve it. Validate the AppID used in the REST request, its allowed authentication methods, and the safe membership and permissions assigned to that AppID or provider identity. CyberArk documents Central Credential Provider REST retrieval and application authorization in its Central Credential Provider REST API documentation and application identity configuration documentation.

QUESTION NO: 5

A Kubernetes application is being integrated with Secrets Provider for Kubernetes in Push-to-File mode. The application already supports reading its database settings from files and the development team does not want to add Conjur API calls to the application.

Which design choices support this requirement? (Choose two.)

A. Mount a volume shared by the Secrets Provider and application containers for the retrieved secret files.

B. Grant the workload's Conjur identity permission to retrieve the required variables.

C. Embed a long-lived Conjur host API key in the application container image.

D. Copy the Conjur secret values into a Kubernetes Secret before starting the application.

E. Add Conjur REST calls to the application for each secret that it reads.

ANSWER: A B

Explanation:

Secrets Provider for Kubernetes can retrieve secrets from Conjur and write them as files to a volume shared with the application container. The application can then use its existing file-based configuration mechanism without implementing Conjur authentication or secret-retrieval calls.

The workload identity used by the provider must be authorized in Conjur to retrieve the required variables. A Kubernetes Secret is not required as an intermediate store for values that Secrets Provider retrieves from Conjur, and embedding a host API key in the application image would undermine the workload-identity design.

QUESTION NO: 6

You are deploying Secrets Provider for Kubernetes in Push-to-File mode. Passwords are rotated in Conjur while the application Pod remains running.

Which two design choices are required for the application to use rotated values without recreating the Pod? (Choose two.)

A. Deploy Secrets Provider as a Sidecar container.

B. Deploy Secrets Provider only as an init container.

C. Configure the application to reload its credential when the shared file changes.

D. Inject the secret into the application as an environment variable.

E. Write the retrieved secret into a Kubernetes Secret for the application to consume.

ANSWER: A C

Explanation:

A Sidecar deployment keeps Secrets Provider running for the life of the Pod, allowing it to retrieve updated values and rewrite the files in the shared volume. The application must also be capable of detecting and reloading the updated file contents; Conjur cannot force an application to reload a credential that it has already read into memory.

An init container retrieves secrets only during Pod initialization, so it cannot refresh files after startup. Environment variables are fixed for a running process, and writing the secret to a Kubernetes Secret would unnecessarily create another stored copy of the secret.

QUESTION NO: 7

A Kubernetes application attempting to authenticate to the Follower load balancer receives this error:

ERROR: 2024/10/30 06:07:08 authenticator.go:139: CAKC029E Received invalid response to certificate signing request. Reason: status code 401

When checking the logs, you see this message:

authn-k8s/prd-cluster-01 is not enabled

How do you remediate the issue?

A. Check the info endpoint on each Follower behind the load balancer and enable the authenticator on the Follower.

B. Modify `/opt/conjur/etc/conjur.conf` to add the authenticator webservice ID to `CONJUR_AUTHENTICATORS`, then restart the Conjur service.

C. A network issue is preventing the application from reaching the Follower; correct the issue and verify that it is resolved.

D. Enable the appropriate authenticator webservice in the UI under Webservices > Authenticators.

ANSWER: B

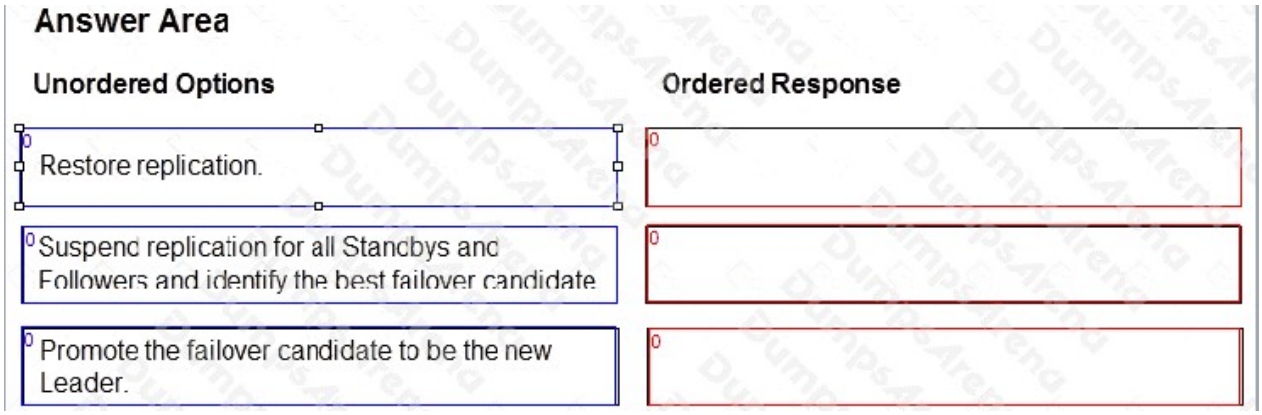
Explanation:

Modify `/opt/conjur/etc/conjur.conf` to add the Kubernetes authenticator service ID, `authn-k8s/prd-cluster-01`, to the `CONJUR_AUTHENTICATORS` setting, then restart the Conjur service so the configuration is loaded. The log entry explicitly identifies the cause of the authentication failure: the requested authenticator endpoint exists as a named service but is not enabled by the Conjur server. A 401 response during the certificate-signing request is therefore expected because Conjur will not process requests for an authenticator that is absent from its enabled-authenticator configuration.

For a Kubernetes authenticator, the service ID must match the authenticator ID configured in Conjur policy and in the Kubernetes-side authenticator configuration. In an HA deployment where applications use a Follower load balancer, the enabled-authenticator configuration must be consistently applied to the Conjur nodes that can receive those requests. After the configuration change and service restart, confirm that the authenticator is enabled through the relevant Conjur status or info endpoint before retesting the workload authentication flow. CyberArk documents the Kubernetes authenticator setup and server-side authenticator enablement requirements in its Kubernetes Authenticator documentation and Conjur authentication documentation.

QUESTION NO: 8 - (SIMULATION)

Arrange the manual failover configuration steps in the correct sequence.



ANSWER: Check the explanation for the answer

Explanation:

The correct manual failover sequence is:

Replication is suspended first to prevent divergence or data corruption while selecting the most current Standby. That Standby is promoted to Leader, then replication is restored and the remaining nodes are rebased on the new Leader.

QUESTION NO: 9

When working with Credential Providers in a Privileged Cloud setting, what is a special consideration?

A. If there are installation issues, troubleshooting may need to involve the Privileged Cloud support team.

B. Credential Providers are not supported in a Privileged Cloud setting.

C. The AWS Cloud account number must be defined in the file main appprovider.conf. . found in the AppProviderConf Safe.

D. Debug logging for Credential Providers deployed in a Privileged Cloud setting can inadvertently exhaust available disk space.

ANSWER: A

Explanation:

“If there are installation issues, troubleshooting may need to involve the Privileged Cloud support team.” is correct because Credential Providers used with CyberArk Privileged Cloud operate within a managed SaaS environment. Although the Credential Provider is installed and administered on the customer-side application host or designated provider host, the Privileged Cloud service includes tenant-side configuration and managed components that customers do not directly administer. Successful deployment depends on the appropriate connectivity, tenant configuration, certificates, and service-side enablement being in place.

Consequently, an installation or connection issue can span both the customer-managed host and the Privileged Cloud environment. The local administrator can collect Credential Provider logs, validate network reachability, and verify application and authentication configuration, but CyberArk support may need to review or adjust elements within the managed Privileged Cloud service. Escalating with the relevant installation details, error messages, timestamps, and logs enables the support team to correlate the client-side symptoms with tenant-side service activity. This shared-responsibility troubleshooting model is the key special consideration for Credential Providers in a Privileged Cloud deployment. CyberArk’s Credential Providers documentation describes the provider’s role in securely retrieving application credentials, while Privileged Cloud documentation describes the managed-service context: CyberArk Credential Providers documentation and CyberArk Privilege Cloud documentation.