SECRET-SEN Q&As

CyberArk Sentry - Secrets Manager

Pass CyberArk SECRET-SEN Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/secret-sen.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by CyberArk Official Exam Center

QUESTION 1

When an application is retrieving a credential from Conjur, the application authenticates to Follower A. Follower B receives the next request to retrieve the credential.

What happens next?

A. The Coniur Token is stateless and Follower B is able to validate the Token and satisfy the request.

B. The Coniur Token is stateful and Follower B is unable to validate the Token promptinq the application to re-authenticate.

C. The Coryur Token is stateless and Follower B redirects the request to Follower A to satisfy the request.

D. The Coniur Token is stateful and Follower B redirects the request to Follower A to satisfy the request.

Correct Answer: A

This is the correct answer because the Conjur Token is a JSON Web Token (JWT) that is signed by the Conjur master and contains the identity and permissions of the application. The Conjur Token is stateless, meaning that it does not depend on any stored session or transaction information on the server side. Therefore, any Conjur follower can validate the Token by verifying the signature and the expiration time, and satisfy the request by retrieving the credential from the local database. This allows the Conjur followers to be horizontally scalable and load balanced, and to provide high availability and performance for the applications. This answer is based on the Conjur documentation1 and the Conjur training course2.

QUESTION 2

When using the Seed Fetcher to deploy Kubernetes Followers, an error occurs in the Seed Fetcher container. You check the logs and discover that although the Seed Fetcher was able to authenticate, it shows a 500 error in the log and does not successfully retrieve a seed file. What is the cause?

A. The certificate based on the Follower DNS name is not present on the Leader.

B. The host you configured does not have access to see the certificates.

C. The synchronizer service crashed and needs to be restarted.

D. The Leader does not have the authenticator webservice enabled.

Correct Answer: A

The cause of the issue is A. The certificate based on the Follower DNS name is not present on the Leader. This means that the Leader does not have a certificate file that matches the Follower DNS name used in the seed request, and therefore cannot generate a valid seed file for the Follower. This results in a 500 error in the Seed Fetcher container log. To resolve the issue, you need to import a certificate with the Follower DNS name as the subject alt name on the Leader, and create a copy of the certificate file with a name that matches the Follower DNS name used in the seed request1.

QUESTION 3

When attempting to retrieve a credential, you receive an error 401 ?Malformed Authorization Token.

What is the cause of the issue?

A. The token is not correctly encoded.

B. The token you are trying to retrieve does not exist.

C. The host does not have access to the credential with the current token.

D. The credential has not been initialized.

Correct Answer: A

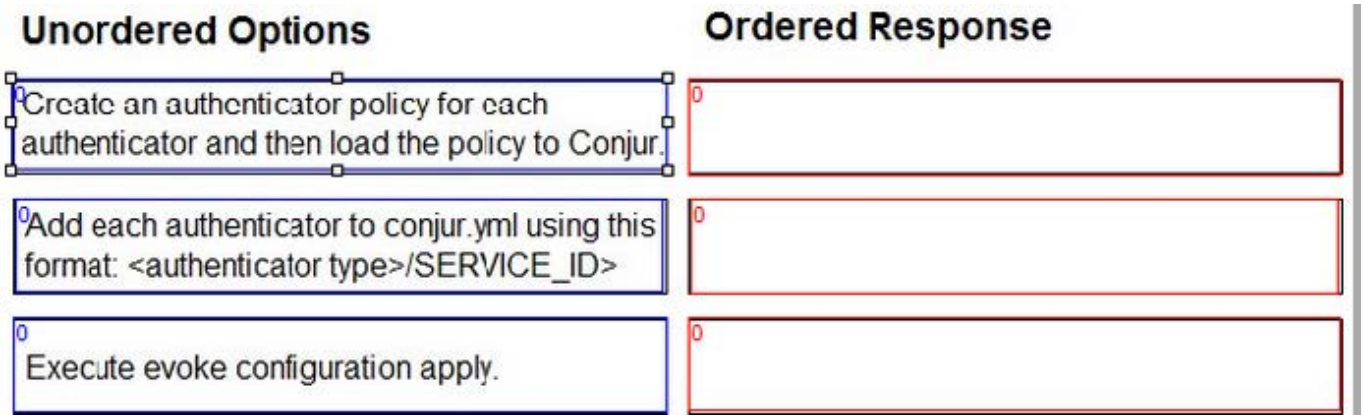
= The cause of the issue is that the token is not correctly encoded. A token is a string of characters that represents a credential or an authorization grant for accessing a resource. A token must be encoded according to a specific format and standard, such as Base64, JSON Web Token (JWT), or OAuth 2.0. If the token is malformed, meaning that it does not follow the expected format or standard, the server will reject the token and return an error 401 - Malformed Authorization Token. This error indicates that the token is invalid or expired, and the request is unauthorized. To resolve the issue, the token must be regenerated or refreshed using the correct encoding method and parameters12. References: = CyberArk Identity: Getting 401 unauthorized Error when using API calls with OAuth2 Client 2, Resolution 1 Troubleshoot CyberArk Vault Synchronizer 1, Error: Forbidden Logon Token is Empty - Cannot logon Unauthorized

QUESTION 4

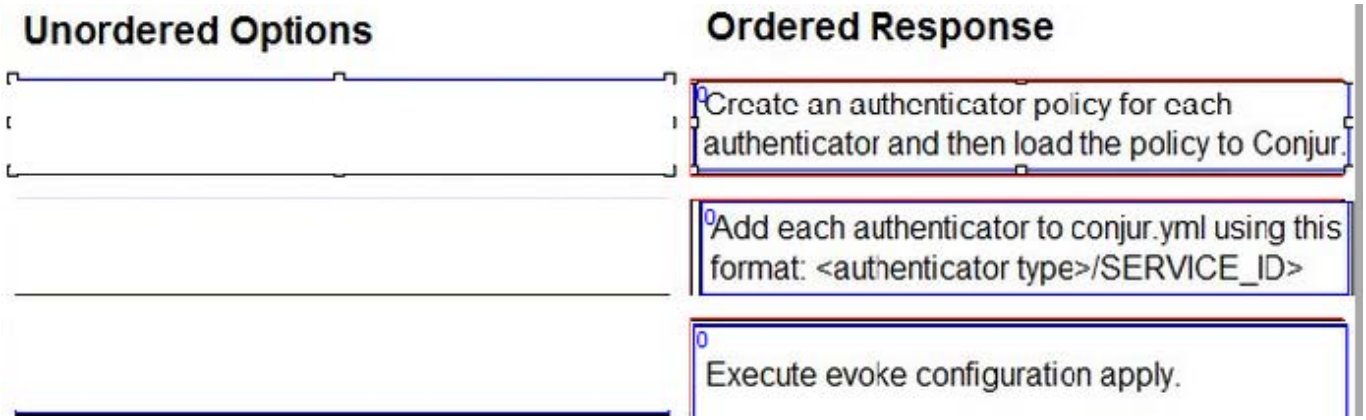
DRAG DROP

Arrange the steps to configure authenticators in the correct the sequence.

Select and Place:



Correct Answer:



Create an authenticator policy for each authenticator and then load the policy to Conjur.

Add each authenticator to conjur.yml using this format: .

Execute evoke configuration apply.

Comprehensive Authenticators are plugins that enable Conjur to authenticate requests from different types of clients, such as Kubernetes, Azure, or LDAP. To configure authenticators, you need to follow these steps:

Create an authenticator policy for each authenticator and then load the policy to Conjur. This step defines the authenticator as a resource in Conjur and grants permissions to the users or hosts that can use it. You can use the policy templates

provided by Conjur for each authenticator type, or create your own custom policy. For more information, see Define Authenticator Policy. Add each authenticator to conjur.yml using this format: . This step

enables the authenticator service on the Conjur server and specifies the service ID that identifies the authenticator instance. The service ID must match the one used in the policy. For more information, see Enable Authenticators.

Execute evoke configuration apply. This step applies the changes made to the conjur.yml file and restarts the Conjur service. This is necessary for the authenticator configuration to take effect. For more information, see Apply Configuration

Changes.

References: The steps to configure authenticators are explained in detail in the Configure Authenticators section of the CyberArk Conjur Enterprise documentation. The image in the question is taken from the same source.

QUESTION 5

During the configuration of Conjur, what is a possible deployment scenario?

A. The Leader and Followers are deployed outside of a Kubernetes environment; Slandbys can run inside a Kubernetes environment.

B. The Conjur Leader cluster is deployed outside of a Kubernetes environment; Followers can run inside or outside the environment.

C. The Leader cluster is deployed outside a Kubernetes environment; Followers and Standbys can run inside or outside the environment.

D. The Conjur Leader cluster and Followers are deployed inside a Kubernetes environment.

Correct Answer: C

Conjur is a secrets management solution that securely stores and manages secrets and credentials used by applications, DevOps tools, and other systems. Conjur can be deployed in different scenarios, depending on the needs and preferences of the organization. One of the possible deployment scenarios is to deploy the Leader cluster outside a Kubernetes environment, and the Followers and Standbys inside or outside the environment. The Leader cluster is the primary node that handles all write operations and coordinates the replication of data to the Follower and Standby nodes. The Leader cluster consists of one active Leader node and one or more Standby nodes that can be promoted to Leader in case of a failure. The Leader cluster can be deployed outside a Kubernetes environment, such as on a virtual machine or a physical server, using Docker or other installation methods. This can provide more control and flexibility over the configuration and management of the Leader cluster, as well as better performance and security. The Follower and Standby nodes are read-only replicas of the Leader node that can serve requests from clients and applications that need to retrieve secrets or perform other read- only operations. The Follower and Standby nodes can be deployed inside or outside a Kubernetes environment, depending on the use case and the availability requirements. For example, if the clients and applications are running inside a Kubernetes cluster, it may be convenient and efficient to deploy the Follower and Standby nodes inside the same cluster, using Helm charts or other methods. This can reduce the network latency and complexity, and leverage the Kubernetes features such as service discovery, load balancing, and health checks. Alternatively, if the clients and applications are running outside a Kubernetes cluster, or if there is a need to distribute the Follower and Standby nodes across different regions or availability zones, it may be preferable to deploy the Follower and Standby nodes outside the Kubernetes cluster, using Docker or other methods. This can provide more scalability and resiliency, and avoid the dependency on the Kubernetes cluster. References: Conjur Deployment Scenarios; Conjur Cluster Installation; Conjur Kubernetes Integration