# SECRET-SEN$^{Q\&As}$

CyberArk Sentry - Secrets Manager

## Pass CyberArk SECRET-SEN Exam with 100% Guarantee

**QUESTION 1**

What is the correct command to import the root CA certificate into Conjur?

A. docker exec evoke ca import --no-restart --root;

B. docker exec evoke import --no-restart --root;

C. docker exec evoke ca import --no-restart;

D. docker exec ca import

Correct Answer: C

C. docker exec evoke ca import --no-restart

This is the correct command to import the root CA certificate into Conjur. The evoke ca import command is used to import a certificate authority (CA) certificate into the Conjur appliance. The certificate can be either a root CA or an

intermediate CA. The --no-restart option prevents the Conjur appliance from restarting after importing the certificate. The parameter specifies the path and name of the root CA certificate file to be imported. This command will

add the root CA certificate to the trusted CA store of the Conjur appliance, which is used to validate the certificates of the clients and servers that communicate with Conjur. This command is documented in the Conjur documentation and the

Conjur training course.

The other options are not correct commands to import the root CA certificate into Conjur. The evoke import command does not exist.

The --root option is not a valid option for the evoke ca import command. The ca import command is not a valid docker exec command.

**QUESTION 2**

When installing the Vault Conjur Synchronizer, you see this error:

Forbidden

Logon Token is Empty ?Cannot logon

Unauthorized

What must you ensure to remediate the issue?

A. This admin user must not be logged in to other sessions during the Vault Conjur Synchronizer installation process.

B. You specified the correct url for Conjur and it is listed as a SAN on that url\\'s certificate.

C. You correctly URI encoded the url in the installation script.

D. You ran powershell as Administrator and there is sufficient space on the server on which you are running the installation.

Correct Answer: A

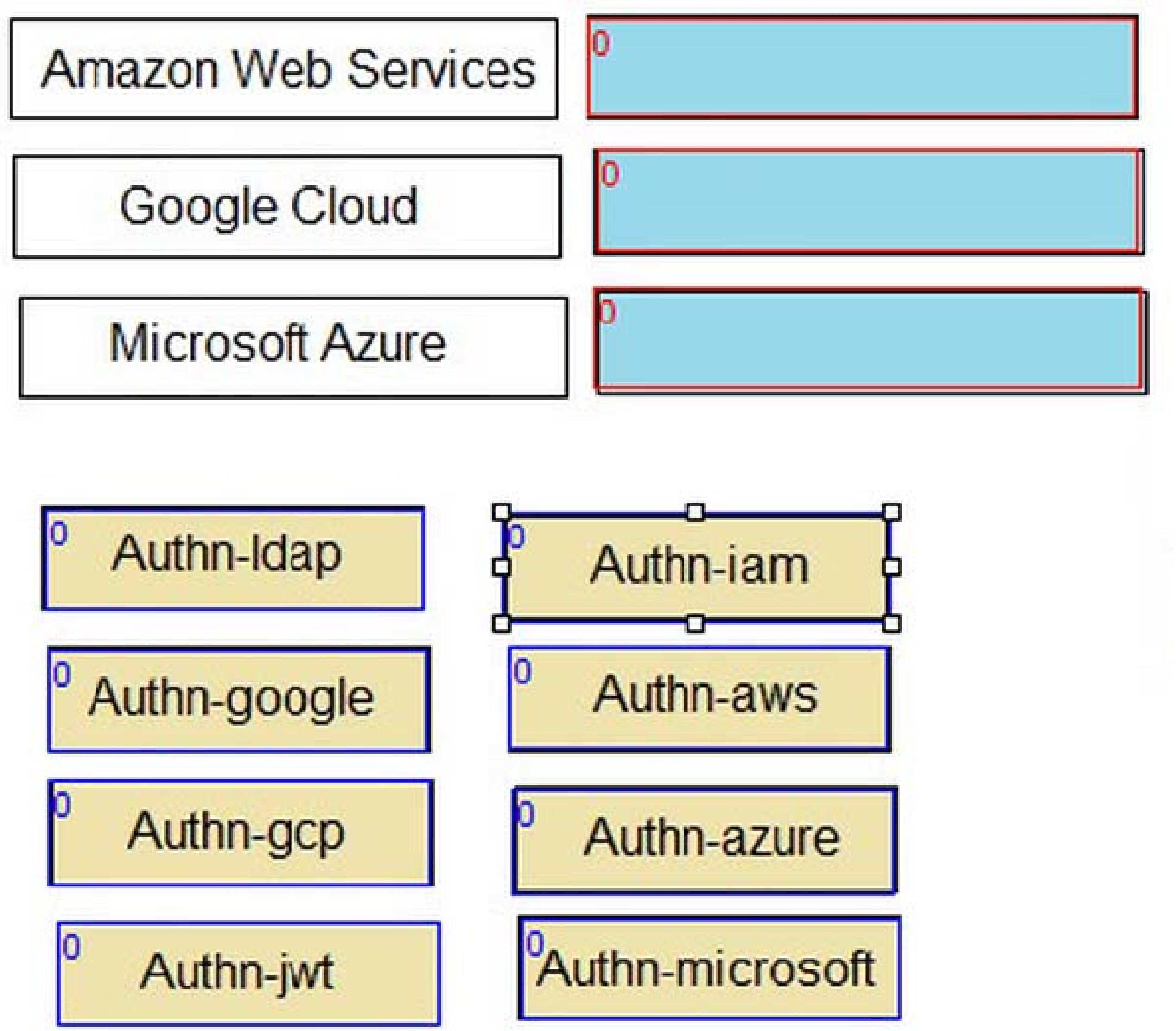
This error occurs when the Vault Conjur Synchronizer installation script tries to log in to the Vault using the admin user credentials, but the admin user is already logged in to other sessions. The Vault has a limit on the number of concurrent sessions per user, and the default value is one. Therefore, the installation script fails to authenticate the admin user and returns the error message: Forbidden Logon Token is Empty - Cannot logon Unauthorized. To remediate the issue, the admin user must log out of any other sessions before running the installation script, or increase the limit on the number of concurrent sessions per user in the Vault configuration file12. References: = Troubleshoot CyberArk Vault Synchronizer 1, Error: Forbidden Logon Token is Empty - Cannot logon Unauthorized Vault.ini File Parameters 2, ConcurrentSessionsPerUser

**QUESTION 3**

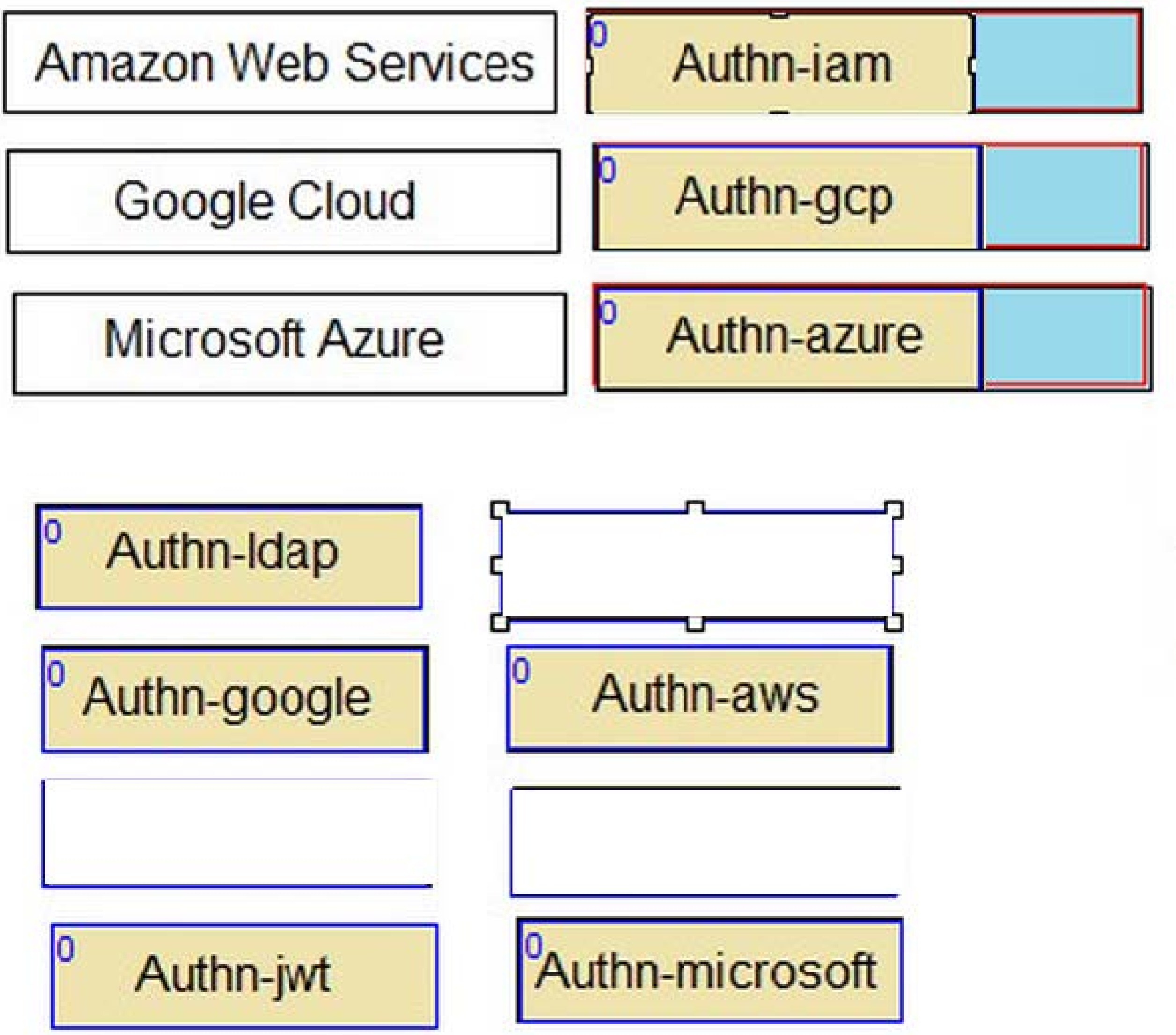
DRAG DROP

Match each cloud platform to the correct Conjur authenticator.

Select and Place:



Correct Answer:



AWS -> authn-iam Azure -> authn-azure GCP -> authn-gcp JWT Provider -> authn-jwt Conjur supports different authenticators for different cloud platforms. Each authenticator allows a resource or service running on the cloud platform to authenticate to Conjur using a unique identity token signed by the cloud provider. The following are the descriptions of each authenticator: authn-iam: Enables an AWS resource to use its AWS IAM role to authenticate with Conjur. The resource sends a request to the AWS Security Token Service (STS) to get a signed AWS access token, and then sends the token to Conjur for verification. authn-azure: Enables an Azure resource to authenticate with Conjur. The resource sends a request to the Azure Instance Metadata Service (IMDS) to get a signed Azure access token, and then sends the token to Conjur for verification. authn-gcp: Enables a Google Cloud Platform resource to authenticate with Conjur. The resource sends a request to the Google Cloud Identity and Access Management (IAM) service to get a signed Google identity token, and then sends the token to Conjur for verification. authn-jwt: Enables an application to authenticate to Conjur using a JWT from a JWT Provider. The application obtains a JWT from the JWT Provider, and then sends the JWT to Conjur for verification. References: You can find more information about the Conjur authenticators in the following resources: Supported Conjur Cloud authenticators Configure Conjur Cloud authenticators GCP Authenticator

**QUESTION 4**

What does "Line of business (LOB)" represent?

A. a business group requiring access to secrets from the Vault/Privilege Claud to facilitate syncing accounts to Conjur

B. the services that Conjur offers and typically refers to a group of application identities in Conjur

C. a business group that meets a certain set of Conjur policies for entitlements and policy management

D. the services that Conjur offers and typically refers to the list of configured and enabled authenticators in Conjur

Correct Answer: B

Line of business (LOB) is a term used by CyberArk Secrets Manager to describe the services that Conjur offers and typically refers to a group of application identities in Conjur. A LOB can be defined by a Conjur policy that grants permissions and access to secrets for a specific set of applications. For example, a LOB can represent a business unit, a project, a product, or a team within an organization. A LOB can also have sub-LOBs that inherit the permissions and secrets from the parent LOB, but can also have their own specific policies and secrets. A LOB can help organize and manage secrets for different applications in a hierarchical and scalable way. References: CyberArk Secrets Manager - Line of Business; CyberArk Secrets Manager - Policy Management; CyberArk Secrets Manager - Application Identity Management

**QUESTION 5**

You are deploying Kubernetes resources/objects as Conjur identities.

In addition to Namespace and Deployment, from which options can you choose? (Choose two.)

A. ServiceAccount

B. Replica sets

C. Secrets

D. Tokenreviews

E. StatefulSet

Correct Answer: AE

ServiceAccount and StatefulSet are two of the Kubernetes resources/objects that can be used as Conjur identities, in addition to Namespace and Deployment. Conjur identities are the entities that can authenticate with Conjur and retrieve secrets from it. Conjur supports authenticating Kubernetes resources/objects using the Conjur Kubernetes Authenticator, which is a sidecar or init container that runs alongside the application container and injects the Conjur access token into a shared volume. The application container can then use the access token to fetch secrets from Conjur. A ServiceAccount is a Kubernetes resource that represents an identity for processes that run in a pod. ServiceAccounts can be used to grant specific privileges and permissions to the pod, and to enable communication with the Kubernetes API server. A ServiceAccount can be used as a Conjur identity by annotating it with the Conjur authentication policy branch ID, and by creating a Conjur host entity that matches the ServiceAccount name and namespace. The Conjur Kubernetes Authenticator will then use the ServiceAccount token to authenticate the pod with Conjur and obtain the Conjur access token. A StatefulSet is a Kubernetes resource that manages the deployment and scaling of a set of pods, and provides guarantees about the ordering and uniqueness of these pods. StatefulSets are useful for applications that require stable and persistent identities, such as databases, message brokers, or distributed systems. A StatefulSet can be used as a Conjur identity by annotating it with the Conjur authentication policy branch ID, and by creating a Conjur host entity that matches the StatefulSet name and namespace. The Conjur Kubernetes Authenticator will then use the pod name and namespace to authenticate the pod with Conjur and obtain the Conjur access token. The other options are not valid Kubernetes resources/objects that can be used as Conjur identities. Replica sets are a lower-level resource that are usually managed by higher-level resources such as Deployments or StatefulSets, and do not have their own identity or annotations. Secrets are a Kubernetes resource that store sensitive information such as passwords, tokens, or keys, and are not meant to be used as identities. Tokenreviews are a Kubernetes resource that are used to verify the validity of a ServiceAccount token, and are not meant to be used as identities either. References: Securing Secrets in Kubernetes - CyberArk Developer, Section "Conjur Kubernetes Authentication: A Hands-On Demonstration" GitHub - cyberark/secrets-provider-for-k8s: Cyberark secrets provider ..., Section "Consuming Secrets from CyberArk Secrets Provider" Secure your Kubernetes-deployed applications with CyberArk Conjur, Section "How it works" Simplify and Improve Container Security Using New CyberArk Conjur ..., Section "CyberArk Conjur Enterprise" Keeping Secrets Secure on Kubernetes - CyberArk Developer, Section "The Solution"