HashiCorp Vault-Associate Practice Questions

Shared by Burton on 18-08-2026

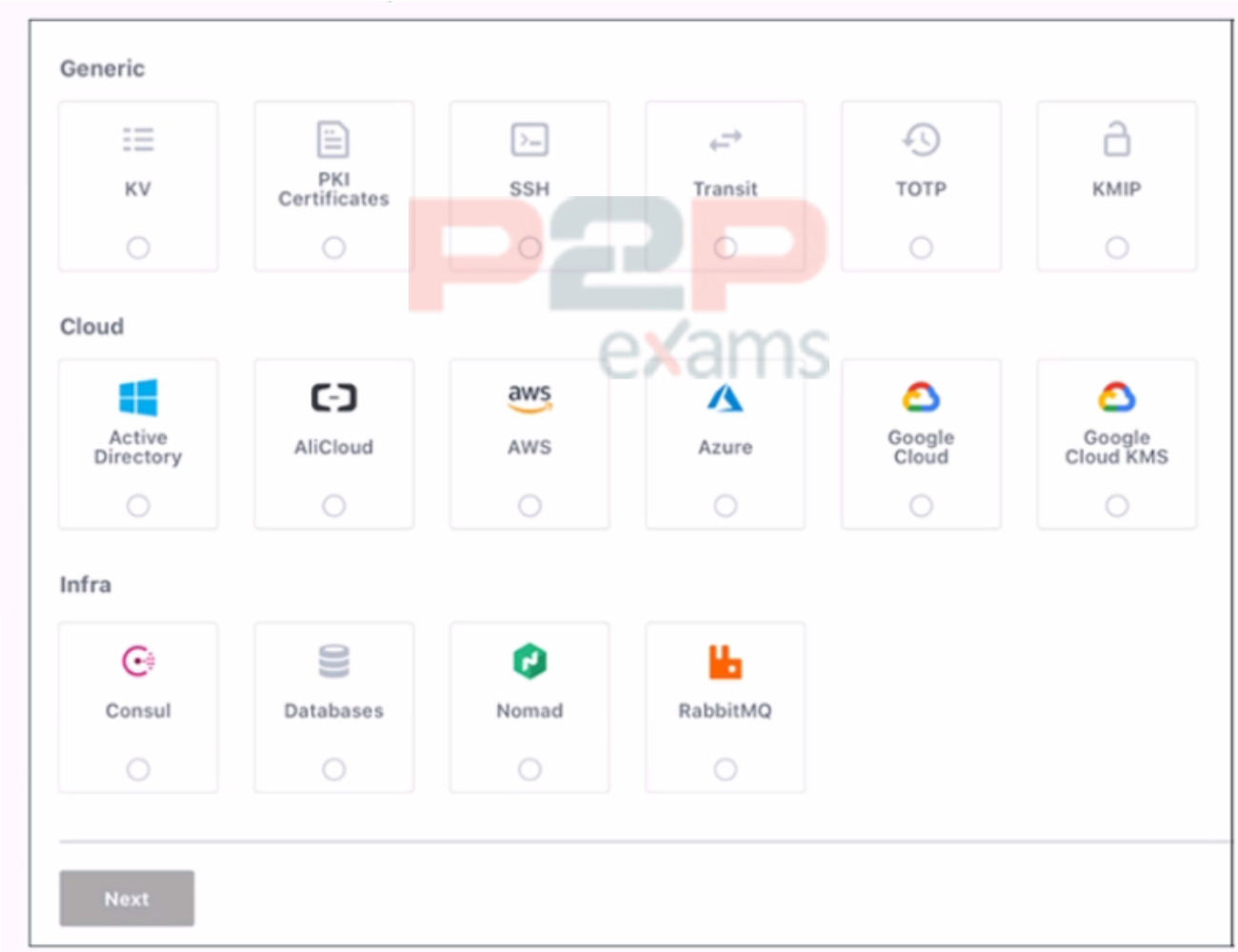
For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Use this screenshot to answer the question below:



When are you shown these options in the GUI?

Options:

A- Enabling policies
B- Enabling authentication engines
C- Enabling secret engines
D- Enabling authentication methods

Answer:

D

Explanation:

This screenshot is shown when you are enabling authentication methods in the GUI. Authentication methods are the ways users and applications authenticate with Vault. Vault supports many different authentication methods, including username and password, GitHub, and more. You can enable one or more authentication methods from the grid of options, which are divided into three categories: Generic, Cloud, and Infra. Each option has a name, a description, and a logo. You can also enable authentication methods using the Vault CLI or API.

Enabling policies, authentication engines, and secret engines are different tasks that are not related to this screenshot. Policies are rules that govern the access to Vault resources, such as secrets, authentication methods, and audit devices. Authentication engines are components of Vault that perform authentication and assign policies to authenticated entities. Secret engines are components of Vault that store, generate, or encrypt data. These tasks have different GUI pages and options than the screenshot.

[Authentication | Vault | HashiCorp Developer]

[Policies | Vault | HashiCorp Developer]

[Authentication | Vault | HashiCorp Developer]

[Secrets Engines | Vault | HashiCorp Developer]

Question 2

Question Type: MultipleChoice

An organization wants to authenticate an AWS EC2 virtual machine with Vault to access a dynamic database secret. The only authentication method which they can use in this case is AWS.

Options:

A- True
B- False

Answer:

B

Explanation:

The statement is false. An organization can authenticate an AWS EC2 virtual machine with Vault to access a dynamic database secret using more than one authentication method. The AWS auth method is one of the options, but not the only one. The AWS auth method supports two types of authentication: ec2 and iam. The ec2 type uses the signed EC2 instance identity document to authenticate the EC2 instance. The iam type uses the AWS Signature v4 algorithm to sign a request to the sts:GetCallerIdentity API and authenticate the IAM principal. However, the organization can also use other auth methods that are compatible with EC2 instances, such as AppRole, JWT/OIDC, or Kubernetes. These methods require the EC2 instance to have some sort of identity material, such as a role ID, a secret ID, a JWT token, or a service account token, that can be used to authenticate to Vault. The identity material can be provisioned to the EC2 instance using various mechanisms, such as user data, metadata service, or cloud-init scripts. The choice of the auth method depends on the use case, the security requirements, and the trade-offs between convenience and control.Reference:AWS - Auth Methods | Vault | HashiCorp Developer,AppRole - Auth Methods | Vault | HashiCorp Developer,JWT/OIDC - Auth Methods | Vault | HashiCorp Developer,Kubernetes - Auth Methods | Vault | HashiCorp Developer

Question 3

Question Type: MultipleChoice

To make an authenticated request via the Vault HTTP API, which header would you use?

Options:

A- The X-Vault-Token HTTP Header
B- The x-Vault-Request HTTP Header
C- The Content-Type HTTP Header
D- The X-Vault-Namespace HTTP Header

Answer:

A

Explanation:

To make an authenticated request via the Vault HTTP API, you need to use the X-Vault-Token HTTP Header or the Authorization HTTP Header using the Bearer <token> scheme. The token is a string that represents your identity and permissions in Vault. You can obtain a token by using an authentication method, such as userpass, approle, aws, etc. The token can also be a root token, which has unlimited access to Vault, or a wrapped token, which is a response-wrapping token that can be used to unwrap the actual token. The token must be sent with every request to Vault that requires authentication, except for the unauthenticated endpoints, such as sys/init, sys/seal-status, sys/unseal, etc.The token is used by Vault to verify your identity and

enforce the policies that grant or deny access to various paths and operations.Reference: https://developer.hashicorp.com/vault/api-docs3, https://developer.hashicorp.com/vault/docs/concepts/tokens4, https://developer.hashicorp.com/vault/docs/concepts/auth5

Question 4

Question Type: MultipleChoice

Which of the following cannot define the maximum time-to-live (TTL) for a token?

Options:

A- By the authentication method t natively provide a method of expiring credentials
B- By the client system f credentials leaking
C- By the mount endpoint configuration very password used
D- A parent token TTL e password rotation tools and practices
E- System max TTL

Answer:

B

Explanation:

The maximum time-to-live (TTL) for a token is defined by the lowest value among the following factors:

The authentication method that issued the token. Each auth method can have a default and a maximum TTL for the tokens it generates. These values can be configured by the auth method's mount options or by the auth method's specific endpoints.

The mount endpoint configuration that the token is accessing. Each secrets engine can have a default and a maximum TTL for the leases it grants. These values can be configured by the secrets engine's mount options or by the secrets engine's specific endpoints.

A parent token TTL. If a token is created by another token, it inherits the remaining TTL of its parent token, unless the parent token has an infinite TTL (such as the root token). A child token cannot outlive its parent token.

System max TTL. This is a global limit for all tokens and leases in Vault. It can be configured by the system backend's max_lease_ttl option.

The client system that uses the token cannot define the maximum TTL for the token, as this is determined by Vault's configuration and policies. The client system can only request a specific TTL for the token, but this request is subject to the limits imposed by the factors above.

Question 5

Question Type: MultipleChoice

What environment variable overrides the CLI's default Vault server address?

Options:

A- VAULT_ADDR
B- VAULT_HTTP_ADORESS
C- VAULT_ADDRESS
D- VAULT _HTTPS_ ADDRESS

Answer:

B

Explanation:

The environment variable VAULT_ADDR overrides the CLI's default Vault server address. The VAULT_ADDR environment variable specifies the address of the Vault server that is used to communicate with Vault from other applications or processes. By setting this variable, you can avoid hard-coding the Vault server address in your code or configuration files, and you can also use different addresses for different environments or scenarios. For example, you can use a local development server for testing purposes, and a production server for deploying your application.Reference:Commands (CLI) | Vault | HashiCorp Developer,Vault Agent - secrets as environment variables | Vault | HashiCorp Developer

Question 6

Question Type: MultipleChoice

When unsealing Vault, each Shamir unseal key should be entered:

Options:

A- Sequentially from one system that all of the administrators are in front of
B- By different administrators each connecting from different computers
B- While encrypted with each administrators PGP key
D- At the command line in one single command

Answer:

B, B

Explanation:

When unsealing Vault, each Shamir unseal key should be entered by different administrators each connecting from different computers. This is because the Shamir unseal keys are split into shares that are distributed to trusted operators, and no single operator should have access to more than one share. This way, the unseal process requires the cooperation of a quorum of key holders, and enhances the security and availability of Vault. The unseal keys can be entered via multiple mechanisms from multiple client machines, and the process is stateful. The order of the keys does not matter, as long as the threshold number of keys is reached. The unseal keys should not be entered at the command line in one single command, as this would expose them to the history and compromise the security.The unseal keys should not be encrypted with each administrator's PGP key, as this would prevent Vault from decrypting them and reconstructing the master key.Reference: https://developer.hashicorp.com/vault/docs/concepts/seal3, https://developer.hashicorp.com/vault/docs/commands/operator/unseal

Question 7

Question Type: MultipleChoice

What can be used to limit the scope of a credential breach?

Options:

A- Storage of secrets in a distributed ledger
B- Enable audit logging
C- Use of a short-lived dynamic secrets
D- Sharing credentials between applications

Answer:

C

Explanation:

Using a short-lived dynamic secrets can help limit the scope of a credential breach by reducing the exposure time of the secrets. Dynamic secrets are generated on-demand by Vault and automatically revoked when they are no longer needed. This way, the credentials are not stored in plain text or in a static database, and they can be rotated frequently to prevent unauthorized access. Dynamic secrets also provide encryption as a service, which means that they perform cryptographic operations on data in-transit without storing any data. This adds an extra layer of security and reduces the risk of data leakage or tampering.Reference:Dynamic secrets | Vault | HashiCorp Developer,What are dynamic secrets and why do I need them? - HashiCorp