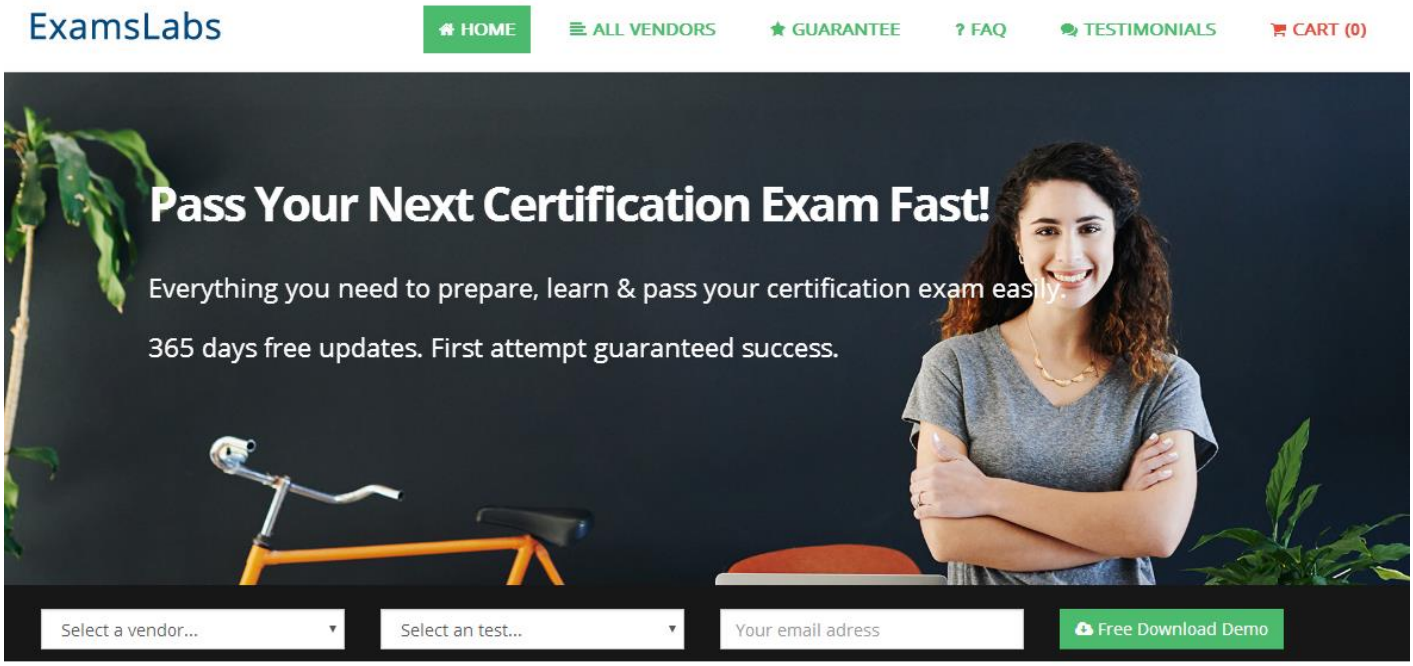
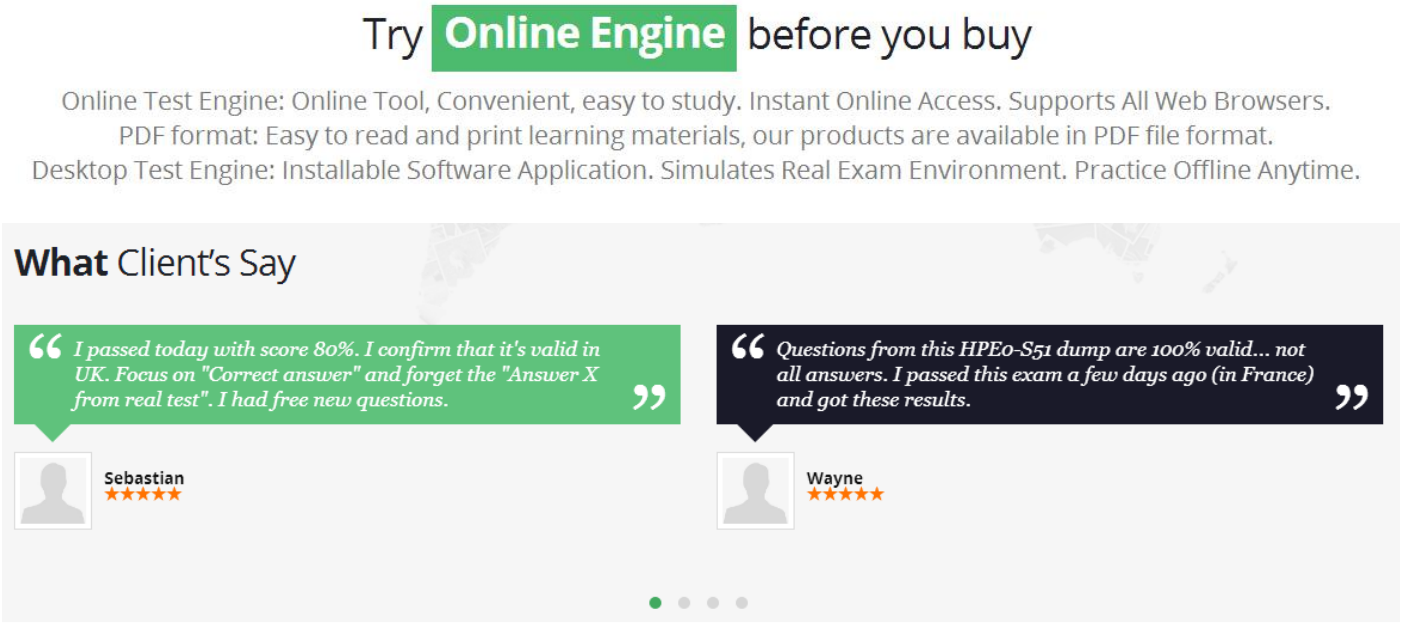
ExamsLabs





http://www.examslabs.com/

Latest Study Materials, Valid Dumps - ExamsLabs

Exam : **Vault-Associate**

Title : HashiCorp Certified: Vault Associate (002)

Vendor : HashiCorp

Version : DEMO

NO.1 Vault supports which type of configuration for source limited token?

A. Cloud-bound tokens

B. Domain-bound tokens

C. CIDR-bound tokens

D. Certificate-bound tokens

Answer: C

Explanation:

Vault supports CIDR-bound tokens, which are tokens that can only be used from a specific set of IP addresses or network ranges. This is a way to limit the scope and exposure of a token in case it is compromised or leaked. CIDR-bound tokens can be created by specifying the bound_cidr_list parameter when creating or updating a token role, or by using the -bound-cidr option when creating a token using the vault token create command. CIDR-bound tokens can also be created by some auth methods, such as AWS or Kubernetes, that can automatically bind the tokens to the source IP or network of the client. Reference: Token - Auth Methods | Vault | HashiCorp Developer, vault token create - Command | Vault | HashiCorp Developer

NO.2 An authentication method should be selected for a use case based on:

A. The auth method that best establishes the identity of the client

B. The cloud provider for which the client is located on

C. The strongest available cryptographic hash for the use case

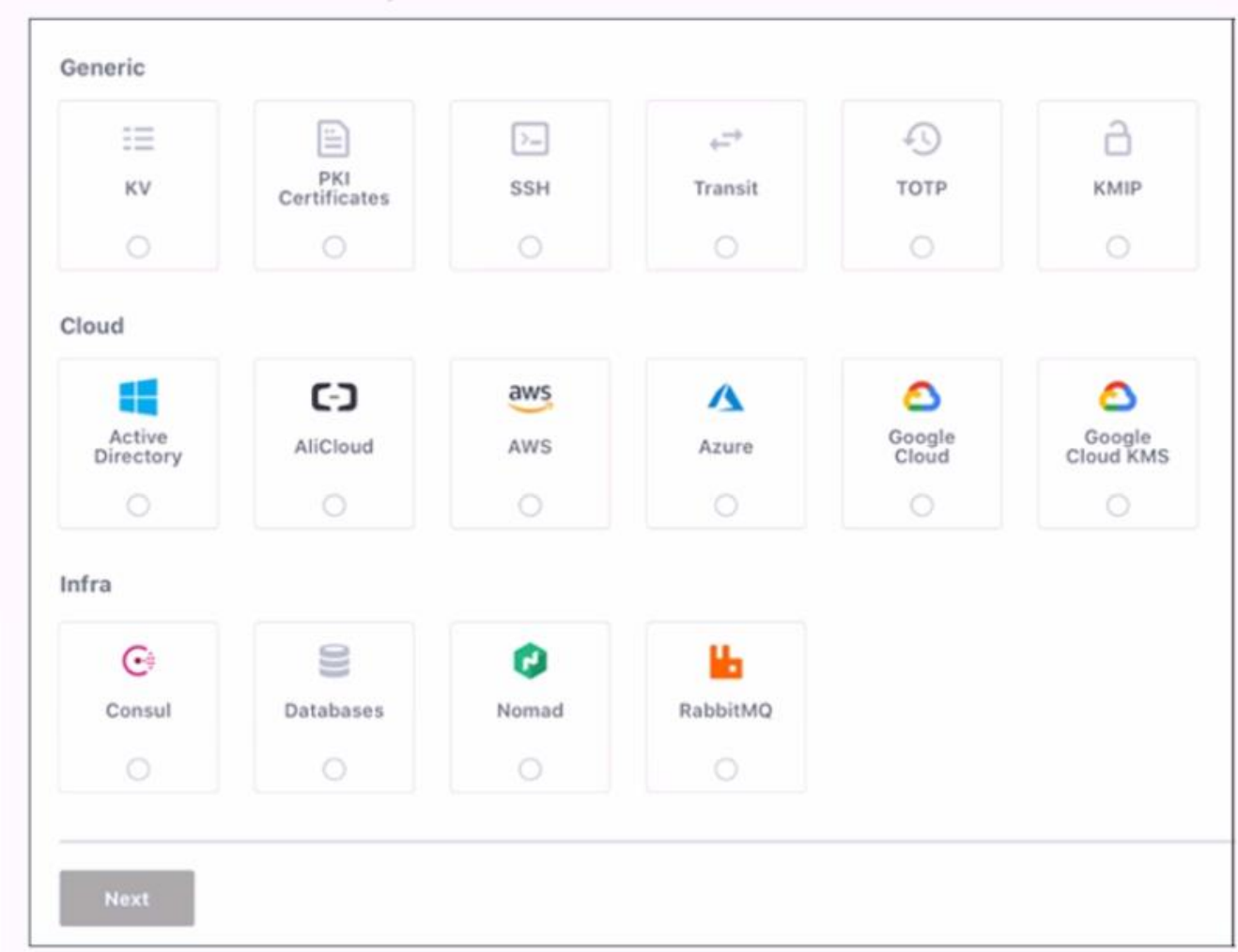
D. Compatibility with the secret engine which is to be used

Answer: A

Explanation:

An authentication method should be selected for a use case based on the auth method that best establishes the identity of the client. The identity of the client is the basis for assigning a set of policies and permissions to the client in Vault. Different auth methods have different ways of verifying the identity of the client, such as using passwords, tokens, certificates, cloud credentials, etc. Depending on the use case, some auth methods may be more suitable or convenient than others. For example, for human users, the userpass or ldap auth methods may be easy to use, while for machines or applications, the approle or aws auth methods may be more secure and scalable. The choice of the auth method should also consider the trade-offs between security, performance, and usability. Reference: Auth Methods | Vault | HashiCorp Developer, Authentication - Concepts | Vault | HashiCorp Developer

NO.3 Use this screenshot to answer the question below:



When are you shown these options in the GUI?

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.

Reference:

[Authentication | Vault | HashiCorp Developer]

[Policies | Vault | HashiCorp Developer]
[Authentication | Vault | HashiCorp Developer]
[Secrets Engines | Vault | HashiCorp Developer]

NO.4 Examine the command below. Output has been trimmed.

```
  $ vault write auth/approle/login \
    role_id="debb8f13-79ea-3e3d-8100-10711d85c1fb" \
    secret_id="31d52faa-5b0b-711d-2ea2-c197cff6081b"Key Value
---                    -----
token                  b.AAAAAQIlWH-DExezQvz-ZGWMhzsy8uWXEoQYHH6O...trimmed...
token_accessor         n/a
token_duration         1m
token_renewable        false
token_policies         ["shipping"]
identity_policies      []
policies               ["shipping"]
token_meta_role_name   shipping
```

Which of the following statements describe the command and its output?

A. Missing a default token policy

B. Generated token's TTL is 60 hours

C. Generated token is an orphan token which can be renewed indefinitely

D. Configures the AppRole auth method with user specified role ID and secret ID

Answer: B,C

Explanation:

The command shown in the image is:

vault token create -policy=approle -orphan -period=60h

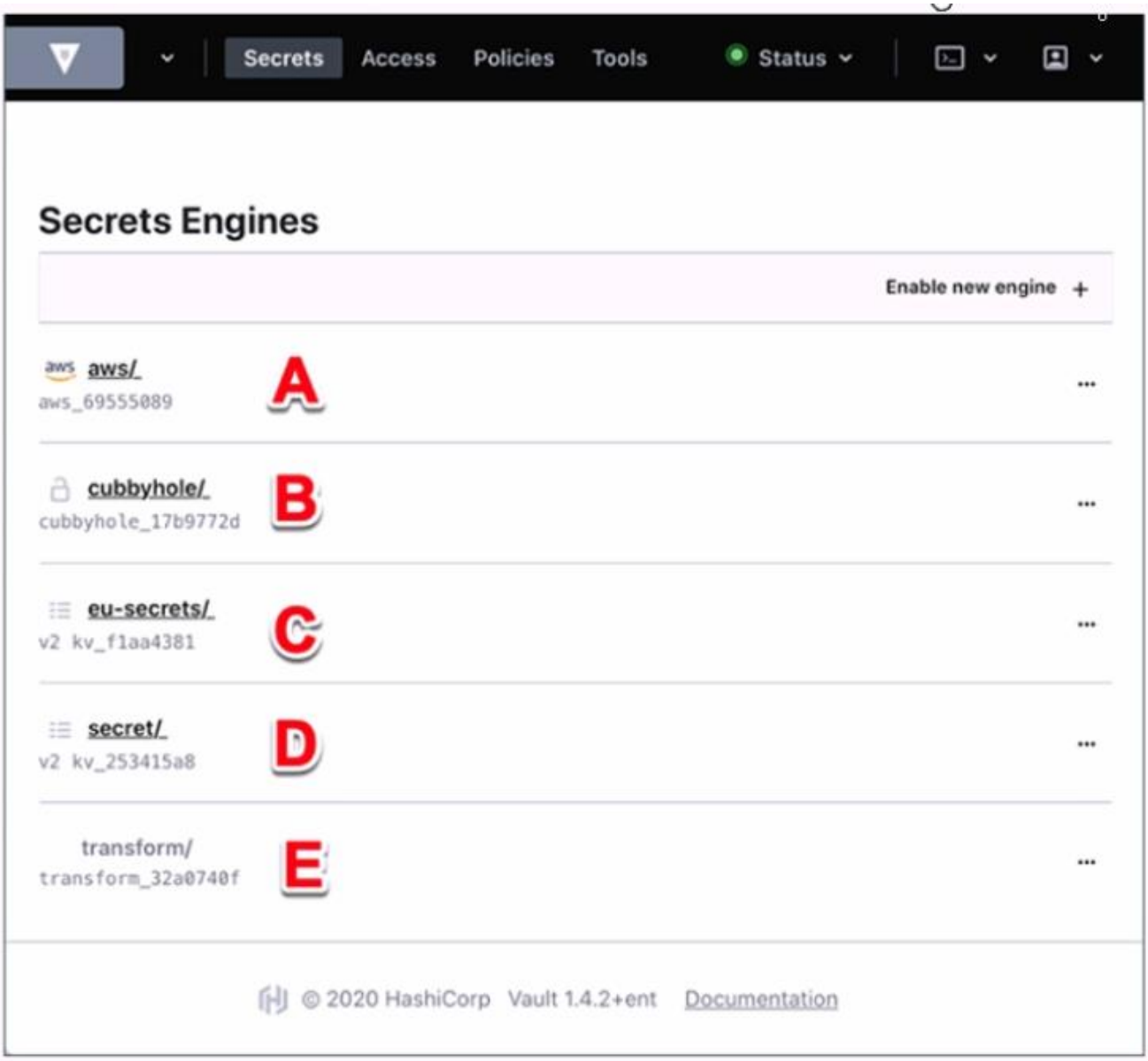
This command creates a new token with the following characteristics:

It has the policy "approle" attached to it, which grants or denies access to certain paths and operations in Vault according to the policy rules. The policy can be defined by using the vault policy write command or the sys/policy API endpoint12.

It is an orphan token, which means it has no parent token and it will not be revoked when its parent token is revoked. Orphan tokens can be useful for creating long-lived tokens that are not affected by the token hierarchy3.

It has a period of 60 hours, which means it has a renewable TTL of 60 hours. This means that the token can be renewed indefinitely as long as it does not go past the 60-hour mark from the last renewal time. The token's TTL will be reset to 60 hours upon each renewal. Periodic tokens are useful for creating tokens that have a fixed lifetime and can be easily revoked4.

NO.5 Use this screenshot to answer the question below:



Where on this page would you click to view a secret located at secret/my-secret?

A. A

B. B

C. C

D. D

E. E

Answer: C

Explanation:

In the HashiCorp Vault UI, secrets are organized in a tree-like structure. To view a secret located at secret/my-secret, you would click on the "secret/" folder in the tree, then click on the "my-secret" file. In this screenshot, the "secret/" folder is located at option C. This folder contains the secrets that are stored in the key/value secrets engine, which is the default secrets engine in Vault. The key/value secrets engine allows you to store arbitrary secrets as key/value pairs. The key is the path of the secret, and the value is the data of the secret. For example, the secret located at secret/my-secret has a key of "my-secret" and a value of whatever data you stored there.

Reference:

[KV - Secrets Engines | Vault | HashiCorp Developer]

NO.6 You can build a high availability Vault cluster with any storage backend.

A. True

B. False

Answer: B

Explanation:

Not all storage backends support high availability mode for Vault. Only the storage backends that support locking can enable Vault to run in a multi-server mode where one server is active and the others are standby. Some examples of storage backends that support high availability mode are Consul, Integrated Storage, and ZooKeeper. Some examples of storage backends that do not support high availability mode are Filesystem, MySQL, and PostgreSQL. Reference:
https://developer.hashicorp.com/vault/docs/concepts/ha1,
https://developer.hashicorp.com/vault/docs/configuration/storage2