VAULT-ASSOCIATE Q&As

HashiCorp Certified: Vault Associate (002)

Pass HashiCorp VAULT-ASSOCIATE Exam with 100% Guarantee

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

https://www.geekcert.com/vault-associate.html

100% Passing Guarantee
100% Money Back Assurance

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

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

When looking at Vault token details, which key helps you find the paths the token is able to access?

A. Meta

B. Path

C. Policies

D. Accessor

Correct Answer: C

When looking at Vault token details, the policies key helps you find the paths the token is able to access. Policies are a declarative way to grant or forbid access to certain paths and operations in Vault. Policies are written in HCL or JSON and

are attached to tokens by name. Policies are deny by default, so an empty policy grants no permission in the system. A token can have one or more policies associated with it, and the effective policy is the union of all the individual policies.

You can view the token details by using the vault token lookup command or the auth/token/lookup API endpoint. The output will show the policies key with a list of policy names that are attached to the token. You can also view the contents of

a policy by using the vault policy read command or the sys/policy API endpoint. The output will show the rules key with the HCL or JSON representation of the policy. The rules will specify the paths and the capabilities (such as create, read,

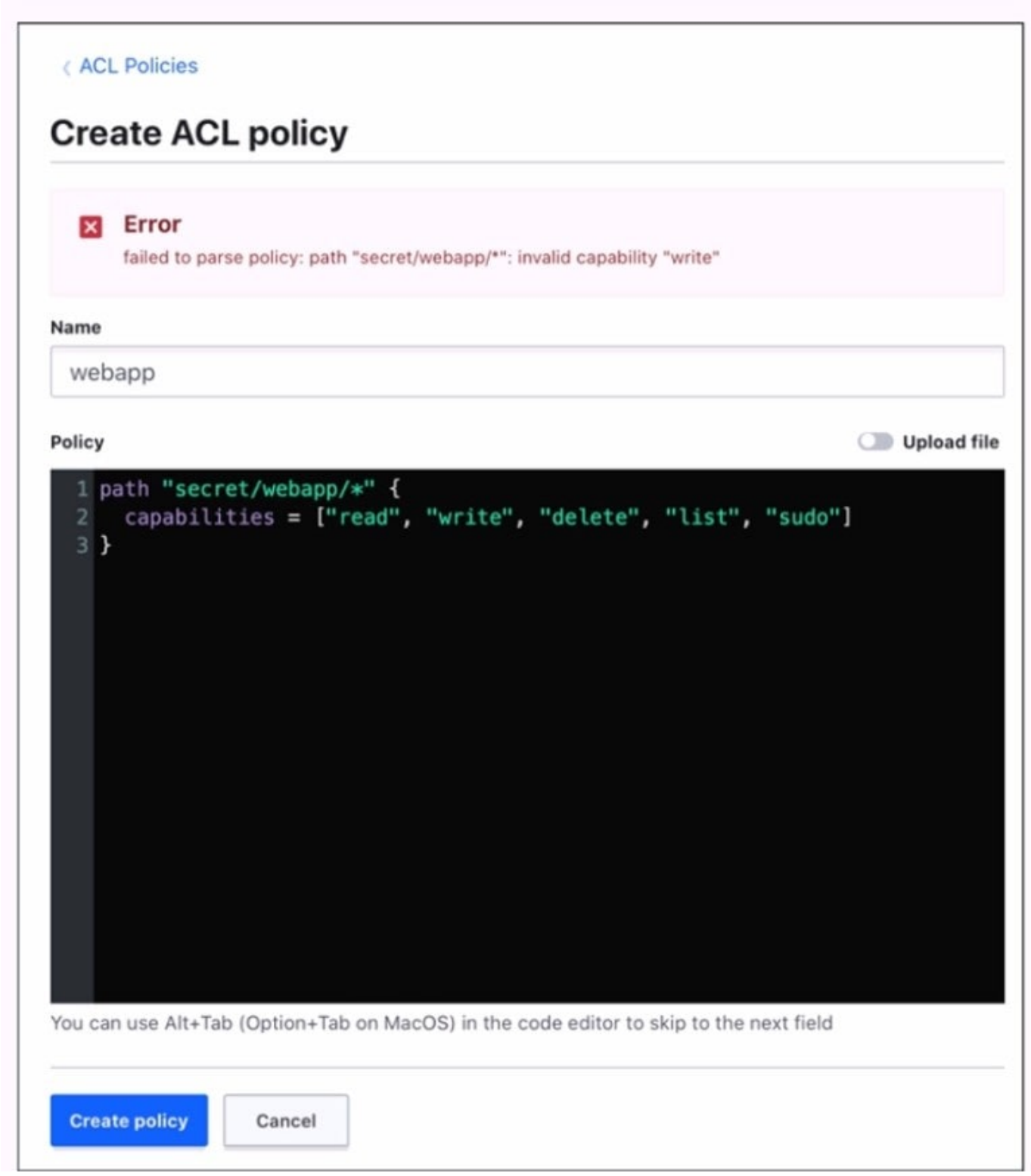
update, delete, list, etc.) that the policy allows or denies. References:

https://developer.hashicorp.com/vault/docs/concepts/policies4, https://developer.hashicorp.com/vault/docs/commands/token/lookup5, https://developer.hashicorp.com/vault/api-docs/auth/token#lookup-a- token6,https://

developer.hashicorp.com/vault/docs/commands/policy/read7, https://developer.hashicorp.com/vault/api-docs/system/policy8

QUESTION 2

When creating a policy, an error was thrown:



Which statement describes the fix for this issue?

A. Replace write with create in the capabilities list

B. You cannot have a wildcard (" ?;) in the path

C. sudo is not a capability

Correct Answer: A

The error was thrown because the policy code contains an invalid capability, "write". The valid capabilities for a policy are "create", "read", "update", "delete", "list", and "sudo". The "write" capability is not recognized by Vault and should be

replaced with "create", which allows creating new secrets or overwriting existing ones. The other statements are not correct, because the wildcard (*) and the sudo capability are both valid in a policy. The wildcard matches any number of

characters within a path segment, and the sudo capability allows performing certain operations that require root privileges.

References:

[Policy Syntax | Vault | HashiCorp Developer]

QUESTION 3

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

Correct Answer: A

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. References: Auth Methods | Vault | HashiCorp Developer, Authentication - Concepts | Vault | HashiCorp Developer

QUESTION 4

You have been tasked with writing a policy that will allow read permissions for all secrets at path secret/bar. The users that are assigned this policy should also be able to list the secrets. What should this policy look like?

A.

```
path "secret/bar/*" {
  capabilities = ["read","list"]
}
```

B.

```
path "secret/bar/*" {
  capabilities = ["list"]
}

path "secret/bar/" {
  capabilities = ["read"]
}
```

C.

```
path "secret/bar/*" {
  capabilities = ["read"]
}

path "secret/bar/" {
  capabilities = ["list"]
}
```

D.

```
path "secret/bar/+" {
  capabilities = ["read", "list"]
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

This policy would allow read permissions for all secrets at path secret/bar, as well as list permissions for the secret/bar/ path. The list permission is required to be able to see the names of the secrets under a given path1. The wildcard ()

character matches any number of characters within a single path segment, while the slash (/) character matches the end of the path2. Therefore, the policy would grant read access to any secret that starts with secret/bar/, such as secret/bar/ foo or secret/bar/baz, but not to secret/bar itself. To grant list access to secret/bar, the policy needs to specify the exact path with a slash at the end. This policy follows the principle of least privilege, which means that it only grants the minimum permissions necessary for the users to perform their tasks3. The other options are not correct because they either grant too much or too little permissions. Option A would grant both read and list permissions to all secrets under secret/bar, which is more than what is required. Option B would grant list permissions to all secrets under secret/bar, but only read permissions to secret/bar itself, which is not what is required. Option D would use an invalid character (+) in the policy, which would cause an error. References: Policy Syntax | Vault | HashiCorp Developer Policy Syntax | Vault | HashiCorp Developer Policies | Vault | HashiCorp Developer

QUESTION 5

Which of the following describes the Vault\\'s auth method component?

A. It verifies a client against an internal or external system, and generates a token with the appropriate policies attached

B. It verifies a client against an internal or external system, and generates a token with root policy

C. It is responsible for durable storage of client tokens

D. It dynamically generates a unique set of secrets with appropriate permissions attached

Correct Answer: A

The Vault\\'s auth method component is the component that performs authentication and assigns identity and policies to a client. It verifies a client against an internal or external system, and generates a token with the appropriate policies attached. The token can then be used to access the secrets and resources that are authorized by the policies. Vault supports various auth methods, such as userpass, ldap, aws, kubernetes, etc., that can integrate with different identity providers and systems. The auth method component can also handle token renewal and revocation, as well as identity grouping and aliasing. References: Auth Methods | Vault | HashiCorp Developer, Authentication - Concepts | Vault | HashiCorp Developer