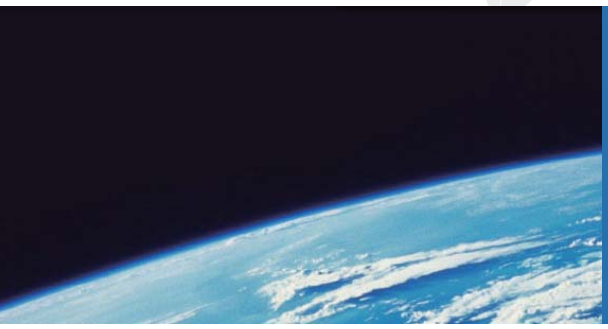
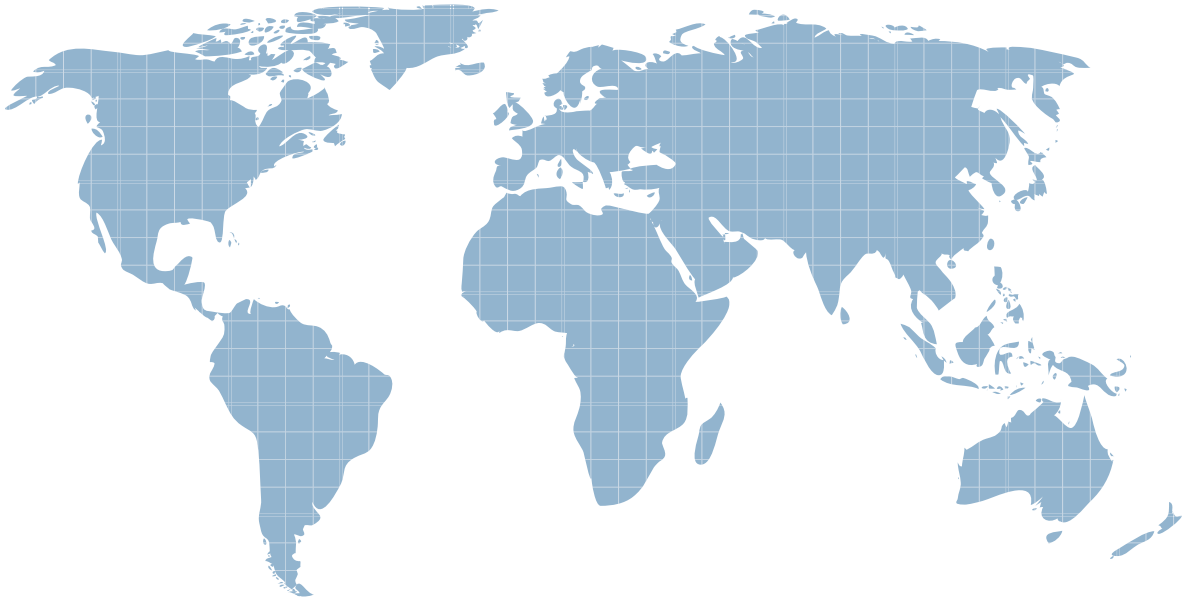


ITTEST

QUESTION & ANSWER

Guías de estudio precisos, Alta tasa de paso!



Ittest ofrece información actualizada de forma gratuita en un año!

http://www.ittest.es/

Exam : **Vault Associate 003**

Title : HashiCorp Certified: Vault Associate (003)

Version : DEMO

1.You are using the Vault userpass auth method mounted at auth/userpass.
How do you create a new user named "sally" with password "h0wN0wB4r0wnC0w"? This new user will need the power-users policy.
A)

```
vault put auth/userpass/users/sally \
password=h0wN0wB4r0wnC0w \
policies=power-users
```

B)

```
vault write userpass/sally \
password=h0wN0wB4r0wnC0w \
policies=power-users
```

C)

```
vault kv write userpass/sally \
password=h0wN0wB4r0wnC0w \
policies=power-users
```

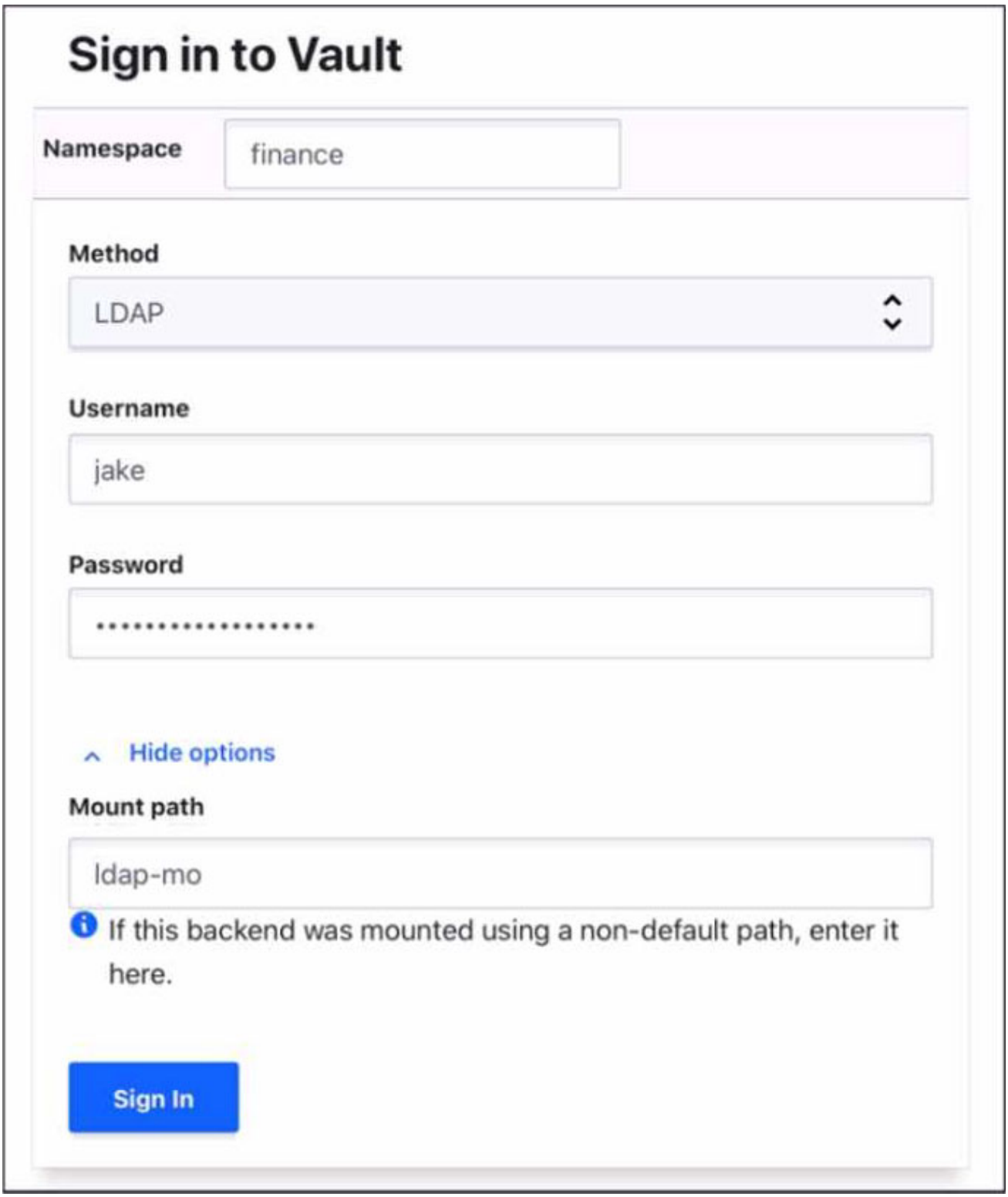
D)

```
vault write auth/userpass/users/sally \
password=h0wN0wB4r0wnC0w \
policies=power-users
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: A
Explanation:
When using Vault's userpass authentication method, you need to use the correct path and command to create a new user. In this case, the path is.auth/userpass/users/<username>
Option A uses this path correctly, and also correctly sets the username to "sally", the password to "h0wN0wB4r0wnC0w", and gives the "power-users" policy.
Option B uses the wrong path, which is not the correct command format to create the user.userpass/sally
Option C tries to use the key-value store command, which is incorrect because we're creating a user, not a key-value pair.kv write
The path for option D is also wrong, and it is missing the start command.vault put
So, to create a new user named "sally" with a password of "h0wN0wB4r0wnC0w" and a "power-users" policy, you should use the command of option A.

2.The vault lease renew command increments the lease time from:
A. The current time
B. The end of the lease
Answer: A
Explanation:
The vault lease renew command increments the lease time from the current time, not the end of the lease. This means that the user can request a specific amount of time they want remaining on the lease, termed the increment. This is not an increment at the end of the current TTL; it is an increment from the current time. For example, vault lease renew -increment=3600 my-lease-id would request that the TTL of the lease be adjusted to 1 hour (3600 seconds) from now. Having the increment be rooted at the current time instead of the end of the lease makes it easy for users to reduce the length of leases if they don't actually need credentials for the full possible lease period, allowing those credentials to expire sooner and resources to be cleaned up earlier. The requested increment is completely advisory. The backend in charge of the secret can choose to completely ignore it1.
Reference: Lease, Renew, and Revoke | Vault | HashiCorp Developer

3.HOTSPOT
Where do you define the Namespace to log into using the Vault UI?
To answer this question
Use your mouse to click on the screenshot in the location described above. An arrow indicator will mark where you have clicked. Click the "Answer" button once you have positioned the arrow to answer the question. You may need to scroll down to see the entire screenshot.



Answer:

The namespace can be defined in the “Mount path” field in the “Advanced options” section of the login screen. The mount path is the path where the auth method is enabled, and it can include a namespace prefix. For example, if the LDAP auth method is enabled at the path ns1/auth/ldap, where ns1 is the namespace, then the mount path field should be set to ns1/auth/ldap. This way, the
Vault UI will log in to the correct namespace and auth method. Alternatively, the namespace can also

be specified in the URL of the Vault UI, such as
https://vault.example.com/ui/vault/auth/ns1/auth/ldap/login.

4.You have a 2GB Base64 binary large object (blob) that needs to be encrypted.
Which of the following best describes the transit secrets engine?
A. A data key encrypts the blob locally, and the same key decrypts the blob locally.
B. To process such a large blob. Vault will temporarily store it in the storage backend.
C. Vault will store the blob permanently. Be sure to run Vault on a compute optimized machine
D. The transit engine is not a good solution for binaries of this size.
Answer: D
Explanation:
The transit secrets engine is not a good solution for binaries of this size, because it is designed to handle cryptographic functions on data in-transit, not data at-rest. The transit secrets engine does not store any data sent to it, so it would require sending the entire 2GB blob to Vault for encryption or decryption, which would be inefficient and impractical. A better solution would be to use the transit secrets engine to generate a data key, which is a high-entropy key that can be used to encrypt or decrypt data locally. The data key can be returned in plaintext or wrapped by another key, depending on the use case. This way, the transit secrets engine only handles the encryption or decryption of the data key, not the data itself, and the data can be stored in any primary data store.
Reference: Transit - Secrets Engines | Vault | HashiCorp Developer, Encryption as a service: transit secrets engine | Vault | HashiCorp Developer

5.How would you describe the value of using the Vault transit secrets engine?
A. Vault has an API that can be programmatically consumed by applications
B. The transit secrets engine ensures encryption in-transit and at-rest is enforced enterprise wide
C. Encryption for application data is best handled by a storage system or database engine, while storing encryption keys in Vault
D. The transit secrets engine relieves the burden of proper encryption/decryption from application developers and pushes the burden onto the operators of Vault
Answer: D
Explanation:
The transit secrets engine relieves the burden of proper encryption/decryption from application developers and pushes the burden onto the operators of Vault. The transit secrets engine provides encryption as a service, which means that it performs cryptographic operations on data in-transit without storing any data. This allows developers to delegate the responsibility of managing encryption keys and algorithms to Vault operators, who can define and enforce policies on the transit secrets engine. This way, developers can focus on their application logic and data, while Vault
handles the encryption and decryption of data in a secure and scalable manner.
Reference: Transit - Secrets Engines | Vault | HashiCorp Developer, Encryption as a service: transit secrets engine | Vault | HashiCorp Developer