Vendor: HashiCorp

Exam Code: Vault-Associate-003

Exam Name: HashiCorp Certified: Vault Associate (003)

Version: DEMO

QUESTION 1
What API endpoint is used to manage secrets engines in Vault?

A. /secret-engines/
B. /sys/mounts
C. /sys/capabilities
D. /sys/kv

Answer: B
Explanation:
This is the correct endpoint. The /sys/mounts endpoint allows operators to list all mounted secrets engines (GET), enable a new one (POST to /sys/mounts/<path>), or tune existing ones (POST to /sys/mounts/<path>/tune). For example, enabling the AWS secrets engine at aws/ uses POST /v1/sys/mounts/aws with a payload specifying the type (aws). This endpoint is the central hub for secrets engine management.

QUESTION 2
You are deploying Vault in a local data center, but want to be sure you have a secondary Vault cluster in the event the primary cluster goes offline. In the secondary data center, you have applications that are running, as they are architected to run active/active. Which type of replication would be best in this scenario?

A. Disaster Recovery replication
B. Performance replication

Answer: B
Explanation:
Performance replication creates an active secondary cluster that replicates data from the primary in near real-time. It supports read operations locally, reducing latency for applications in the secondary data center, and can handle writes (forwarded to the primary). This fits an active/active architecture, providing redundancy and performance. If the primary fails, the secondary can continue serving reads (though writes need reconfiguring).

QUESTION 3
How long does the Transit secrets engine store the resulting ciphertext by default?

A. 24 hours
B. 30 days
C. 32 days
D. Transit does not store data

Answer: D
Explanation:
Transit encrypts data and returns the ciphertext to the caller without persisting it in Vault.
Detailed Mechanics:
When you run vault write transit/encrypt/mykey plaintext=<base64-data>, Vault uses the named key (e.g., mykey) to encrypt the input and returns a response like vault:v1:<ciphertext>. This ciphertext is not stored in Vault's storage backend (e.g., Consul, Raft); it's the client's responsibility to save it (e.g., in a database). This stateless design keeps Vault lightweight and secure, avoiding data retention risks.
Real-World Example:
Encrypt a credit card: vault write transit/encrypt/creditcard plaintext=$(base64 <<< "1234-5678-

9012-3456"). Response: ciphertext=vault:v1:<data>. You store this in your app's database; Vault retains nothing.
Overall Explanation from Vault Docs:
"Vault does NOT store any data encrypted via the transit/encrypt endpoint... The ciphertext is returned to the caller for storage elsewhere."

QUESTION 4
Which of the following policies would permit a user to generate dynamic credentials on a database?

A. path "database/creds/read_only_role" { capabilities = ["generate"] }
B. path "database/creds/read_only_role" { capabilities = ["update"] }
C. path "database/creds/read_only_role" { capabilities = ["list"] }
D. path "database/creds/read_only_role" { capabilities = ["read"] }

Answer: D
Explanation:
Option D: capabilities = ["read"]
Generating dynamic credentials involves a GET request to database/creds/<role>, mapped to the read capability. This policy allows it. Correct.
Detailed Mechanics:
For a role read_only_role defined with vault write database/roles/read_only_role db_name=my-db creation_statements="CREATE USER...", a user with read on database/creds/read_only_role can run vault read database/creds/read_only_role to get temporary credentials. Vault's policy system aligns HTTP verbs to capabilities: GET = read, PUT = update. This counterintuitive mapping (GET for creation) is specific to dynamic secrets.
Overall Explanation from Vault Docs:
"Generating database credentials requires read capability on database/creds/<role>... Despite creating credentials, the HTTP request is a GET."

QUESTION 5
Which scenario most strongly indicates a need to run a self-hosted Vault cluster instead of using HCP Vault Dedicated?

A. Your organization doesn't require any custom security policies or intricate network topologies
B. You want to offload all operational tasks and rely on HashiCorp to manage patching, upgrades, and infrastructure
C. You prefer a fully managed environment that is readily scalable with minimal configuration overhead
D. You must maintain specific compliance or custom integration requirements that demand full control over the Vault environment, including infrastructure provisioning and plugin development

Answer: D
Explanation:
Compliance, custom integrations, and plugin development need full control, only possible with self-hosted Vault.
Detailed Mechanics:
Self-hosted Vault allows custom plugins, FIPS 140-2 compliance, and specific network configs (e.g., air-gapped setups), unavailable in HCP Vault Dedicated due to its standardized, managed nature.
Overall Explanation from Vault Docs:
"Self-managed Vault supports custom requirements... HCP Vault Dedicated offloads operations

but limits control."

QUESTION 6
From the options below, select the benefits of using a batch token over a service token (select four).

A. Often used for ephemeral, high-performance workloads
B. Can be a root token
C. Can be used on performance replication clusters (if orphan)
D. Has accessors
E. Lightweight and scalable
F. No storage cost for token creation

Answer: ACEF
Explanation:
A: Designed for short-lived, high-performance tasks.
C: Orphan batch tokens work in replication.
E: Minimal overhead makes them scalable.
F: No disk storage reduces cost.
Overall Explanation from Vault Docs:
"Batch tokens are encrypted blobs... lightweight, scalable, no storage cost, ideal for ephemeral workloads."

QUESTION 7
Which of the following are accurate statements regarding the use of a KV v2 secrets engine (select three)?

A. Issuing a vault kv destroy command permanently deletes the current version of the secret
B. Issuing a vault kv destroy command deletes all versions of a secret
C. Issuing a vault kv delete command performs a soft delete of the current version
D. Issuing a vault kv metadata delete command permanently deletes the secret

Answer: ACD
Explanation:
A: destroy removes a specific version permanently.
C: delete soft-deletes the current version.
D: metadata delete removes all versions and metadata.
Overall Explanation from Vault Docs:
"kv delete soft-deletes... kv destroy permanently removes versions... kv metadata delete wipes everything."

QUESTION 8
Which of the following is NOT a valid way in which a lease can be revoked in Vault?

A. Using the user interface (UI)
B. Automatically when the TTL or Max-TTL expires
C. Using the API to call the /v1/sys/leases endpoint
D. Via the CLI using the vault token command

Answer: D

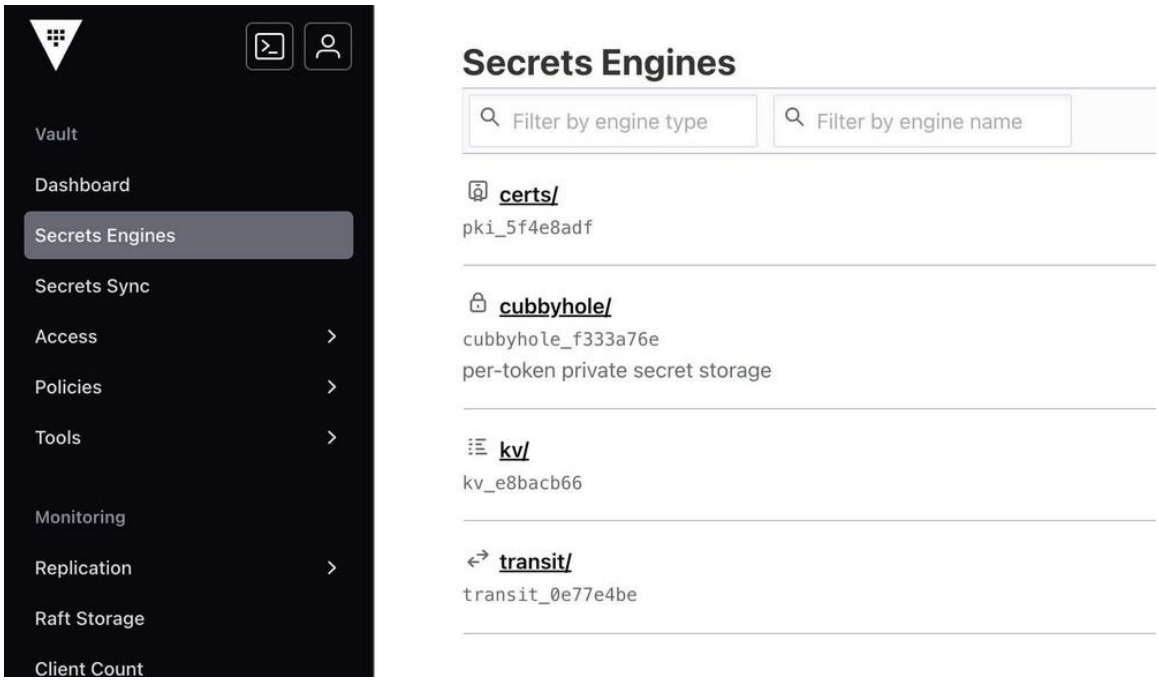
Explanation:
vault token manages tokens, not leases directly.
Overall Explanation from Vault Docs:
"Leases can be revoked via API, UI, CLI (vault lease revoke), or TTL expiry... vault token is for tokens."

QUESTION 9
Once you create a KV v1 secrets engine and place data in it, there is no way to modify the mount to include the features of a KV v2 secrets engine.

A. True
B. False

Answer: B
Explanation:
vault kv enable-versioning upgrades it.
Overall Explanation from Vault Docs:
"kv enable-versioning turns on versioning for an existing KV v1 engine at its path."

QUESTION 10
Given the following screenshot, how many secrets engines have been enabled by a Vault user?



A. 2
B. 3
C. 4
D. 5

Answer: B
Explanation:

cubbyhole is default; kv and transit are user-enabled. Total: 3.
Overall Explanation from Vault Docs:
"cubbyhole is enabled by default... User-enabled engines add to this."

QUESTION 11
When configuring Vault replication and monitoring its status, you keep seeing something called 'WALs'. What are WALs?

A. Warning of allocated logs
B. Write along logging
C. Write-ahead logs
D. Wake after LAN

Answer: C
Explanation:
WALs (Write-Ahead Logs) ensure data consistency in replication.
Overall Explanation from Vault Docs:
"Replication uses Write-Ahead Logs (WALs) for log shipping between clusters..."

QUESTION 12
A Jenkins server is using the following token to access Vault. Based on the lookup shown below, what type of token is this?

```
$ vault token lookup hvs.FGP1A77Hxa1Sp6Pkp1yURcZB

Key                 Value
---                 -----
accessor            RnH8jtgrxBrYanizlyJ7Y8R
creation_time       1604604512
creation_ttl        24h
display_name        token
entity_id           n/a
expire_time         2025-11-06T14:28:32.8891566-05:00
explicit_max_ttl    0s
id                  hvs.FGP1A77Hxa1Sp6KRau5eNB
issue_time          2025-11-06T14:28:32.8891566-05:00
meta                <nil>
num_uses            0
orphan              false
path                auth/token/create
period              24h
policies            [admin default]
renewable           true
ttl                 23h59m50s
type                service
```

A. Periodic token
B. Batch token
C. Orphaned token
D. Secondary token

Answer: A

Explanation:
period indicates a renewable periodic token.
Overall Explanation from Vault Docs:
"A periodic token has a period... renewable without a max TTL."

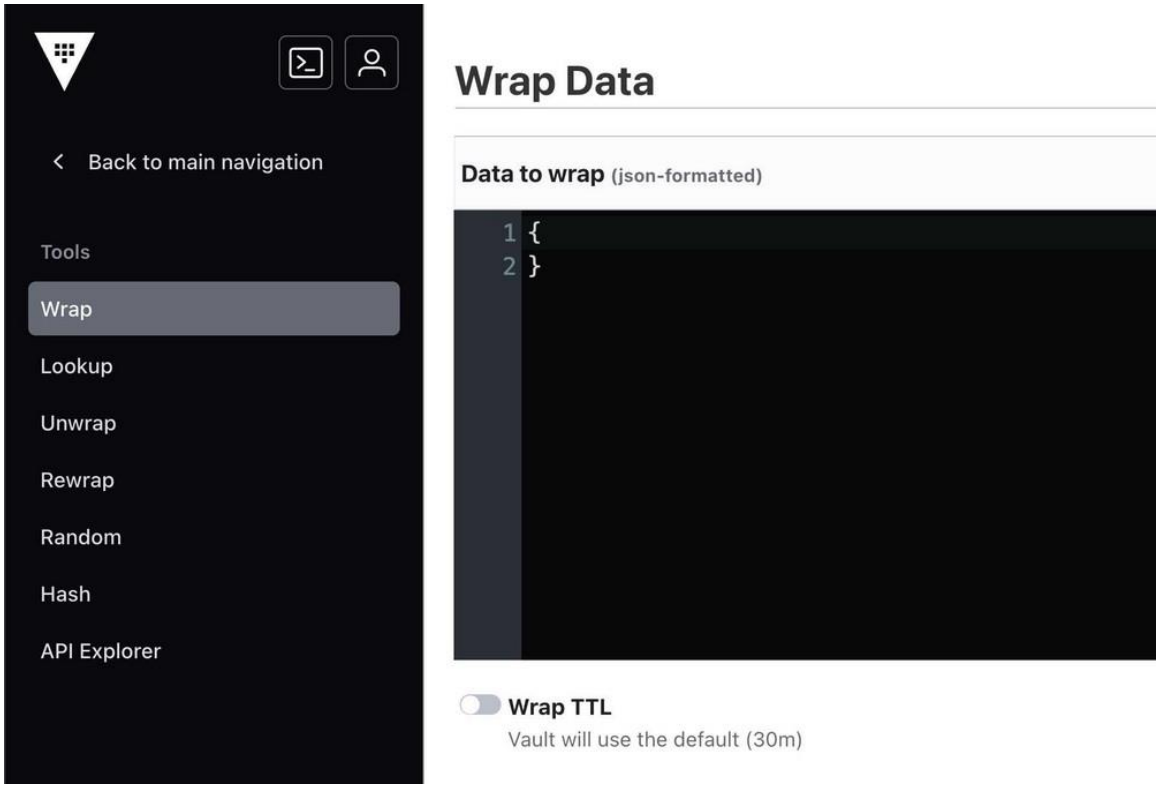
QUESTION 13
After encrypting data using the Transit secrets engine, you've received the following output.
Which of the following is true based on the output displayed below?

```
Key: ciphertext Value:
vault:v2:45f9zW6cglbrzCjI0yCyC6DBYtSBSxnMgUn9B5aHcGEit71xefPEmmjMbrk3
```

A. The original encryption key has been rotated at least once
B. The data is stored in Vault using a KV v2 secrets engine
C. This is the second version of the encrypted data
D. Similar to the KV secrets engine, the Transit secrets engine was enabled using the transit v2 option

Answer: A
Explanation:
v2 shows the key was rotated once.
Overall Explanation from Vault Docs:
"Ciphertext is prepended with the key version (e.g., v2)... Indicates rotation."

QUESTION 14
What could you do with the feature found in the screenshot below (select two)?



A. Using a short TTL, you could encrypt data in order to place only the encrypted data in Vault
B. Encrypt the Vault master key that is stored in memory
C. Encrypt sensitive data to send to a colleague over email
D. Use response-wrapping to protect data

Answer: CD
Explanation:
Option C: Encrypt sensitive data to send to a colleague over email This aligns perfectly with response wrapping. You can retrieve a secret (e.g., vault read secret/data/my-secret), wrap it with a short TTL (e.g., 5 minutes), and receive a token (e.g., hvs.<token>). You email this token to a colleague, who unwraps it with vault unwrap <token> to access the secret. The data is encrypted within the token, secure during transit, and expires after the TTL. This is a textbook use case for wrapping. Correct. Vault Docs Insight: "Response wrapping... can be used to securely send sensitive data to another party, such as over email, with a limited lifetime." (Directly supported use case.)
Option D: Use response-wrapping to protect data
This is the essence of the feature. Wrapping protects data by encapsulating it in a single-use token, accessible only via an unwrap operation. For example, vault write -wrap-ttl=60s secret/data/my- secret returns a wrapped token, protecting the secret until unwrapped. This ensures confidentiality and controlled access, making it a core benefit of the feature. Correct. Vault Docs Insight: "Vault can wrap a response in a single-use token... protecting the data until unwrapped by the recipient." (Core definition.)

QUESTION 15
From the options below, select the benefits of using the PKI (x.509 certificates) secrets engine (select three):

A. TTLs on Vault certs are longer to ensure certificates are valid for a longer period of time
B. Reducing, or eliminating certificate revocations
C. Reduces time to get a certificate by eliminating the need to generate a private key and CSR
D. Vault can act as an intermediate CA

Answer: BCD
Explanation:
Option B: Reducing, or eliminating certificate revocations A key advantage of the PKI engine is issuing short-lived certificates. With short TTLs (e.g., 24h), certificates expire naturally before revocation is needed, minimizing CRL maintenance. For example, an app can fetch a new cert daily, reducing revocation events compared to traditional multi-year certs. This aligns with Vault's ephemeral certificate model. Correct. Vault Docs Insight: "By keeping TTLs relatively short, revocations are less likely to be needed, keeping CRLs short..." (Direct benefit.)
Option C: Reduces time to get a certificate by eliminating the need to generate a private key and CSR Traditionally, obtaining a certificate involves generating a private key, creating a Certificate Signing Request (CSR), and submitting it to a CA--a manual, time-consuming process. The PKI engine automates this: vault write pki/issue/my-role common_name=app.example.com instantly generates a private key and signed certificate. This eliminates manual steps, speeding up issuance significantly. Correct. Vault Docs Insight: "Services can get certificates without... generating a private key and CSR, submitting to a CA, and waiting..." (Automation reduces time.)
Option D: Vault can act as an intermediate CA
The PKI engine can be configured as an intermediate CA, signed by a root CA (internal or external). For example, vault write pki/intermediate/generate/internal common_name="Intermediate CA" creates an intermediate, which can issue certificates under a trust chain. This supports hierarchical PKI setups, a major feature. Correct.

Vault Docs Insight: "The PKI secrets engine can act as an intermediate CA... issuing certificates on behalf of a root CA." (Explicit capability.)
Detailed Mechanics:
The PKI engine operates at paths like pki/ (root) or pki_int/ (intermediate). Roles (e.g., my-role) define parameters like TTL and allowed domains. Issuing a cert (vault write pki/issue/my-role...) returns a JSON payload with certificate, private_key, and issuing_ca. Short TTLs leverage Vault's lease system, auto-revoking certs on expiry. As an intermediate CA, it signs certificates with its key, validated against a root, enhancing trust management.
Real-World Example:
An app needs a cert: vault write pki/issue/web common_name=web.example.com ttl=24h. Vault returns a cert and key instantly, valid for 24 hours. No CSR, no revocation needed--expires tomorrow. Another PKI mount at pki_int/ issues certs under a corporate root CA.
Overall Explanation from Vault Docs:
"The PKI secrets engine generates dynamic X.509 certificates... Services can get certificates without the usual manual process... By keeping TTLs short, revocations are less likely... Vault can act as an intermediate CA, issuing certificates efficiently." These benefits--automation, reduced revocation, and CA flexibility--define its value.

QUESTION 16
According to the screenshot below, what auth method did this client use to log in to Vault? (Screenshot shows a lease path: auth/userpass/login/student01)

A. Userpass
B. Auth
C. Root token
D. Child token

Answer: A
Explanation:
The path auth/userpass/login/student01 explicitly includes userpass, matching the userpass auth method. This method authenticates users with a username (e.g., student01) and password, typically via vault login -method=userpass username=student01. The /login endpoint confirms a login operation, and the lease ties to the resulting token. This is the clear, correct answer based on the path. Correct.
Vault Docs Insight: "The userpass auth method allows users to authenticate with a username and password... mounted at auth/userpass by default." (Matches the path.)

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

★ More than **99,900** Satisfied Customers Worldwide.

★ Average **99.9%** Success Rate.

★ **Free Update** to match latest and real exam scenarios.

★ **Instant Download** Access! No Setup required.

★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.

★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.

★ **100%** Guaranteed Success or **100%** Money Back Guarantee.

★ **Fast**, helpful support **24x7**.

View list of all certification exams: http://www.lead2pass.com/all-products.html