Number: Vault Associate
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: Vault Associate

Exam Name: HashiCorp Certified: Vault Associate

Exam A

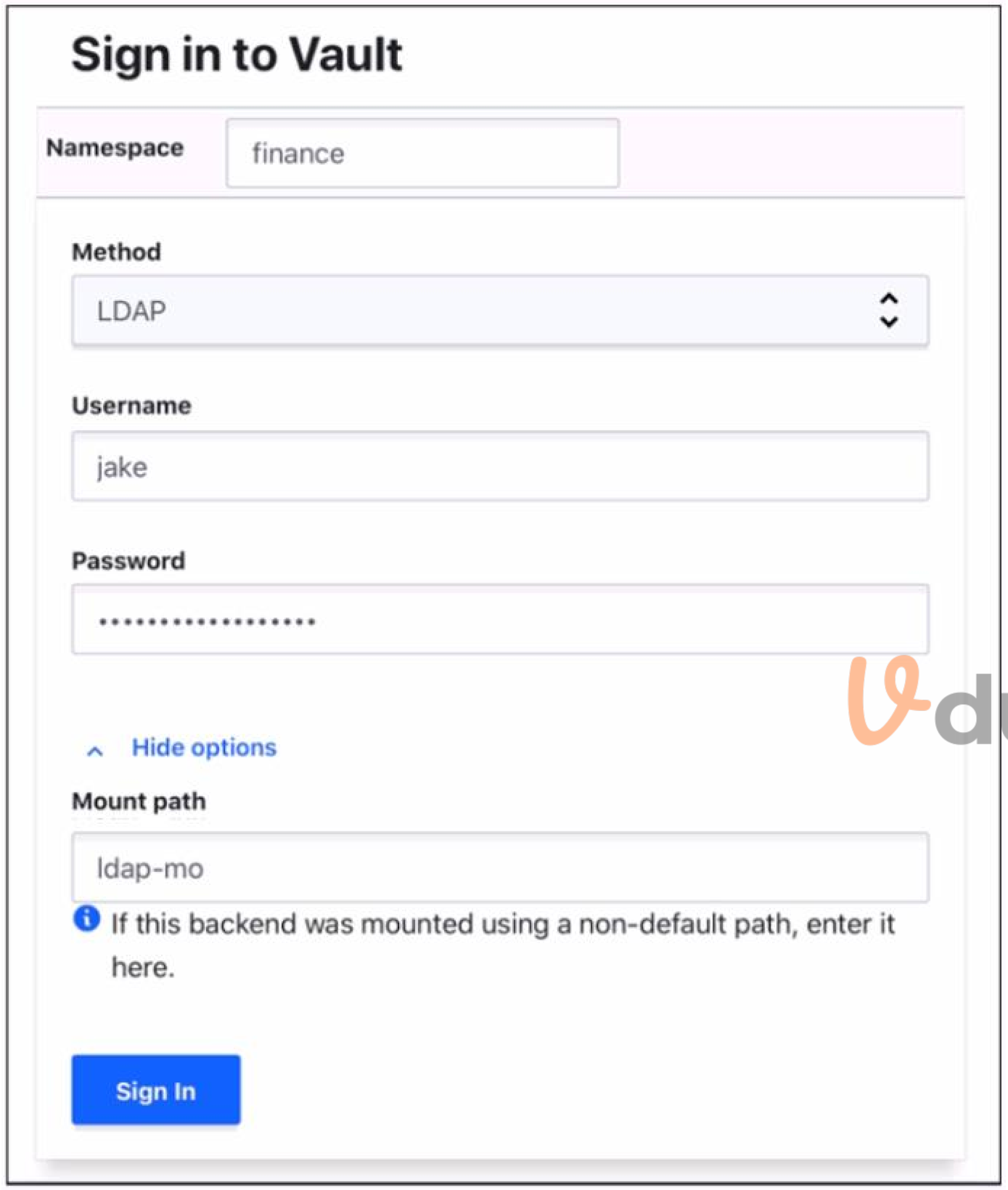
QUESTION 1
The vault lease renew command increments the lease time from:

A. The current time
B. The end of the lease

Correct Answer: A
Section:
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

QUESTION 2
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.

Hot Area:



Sign in to Vault

Namespace: finance

Method: LDAP

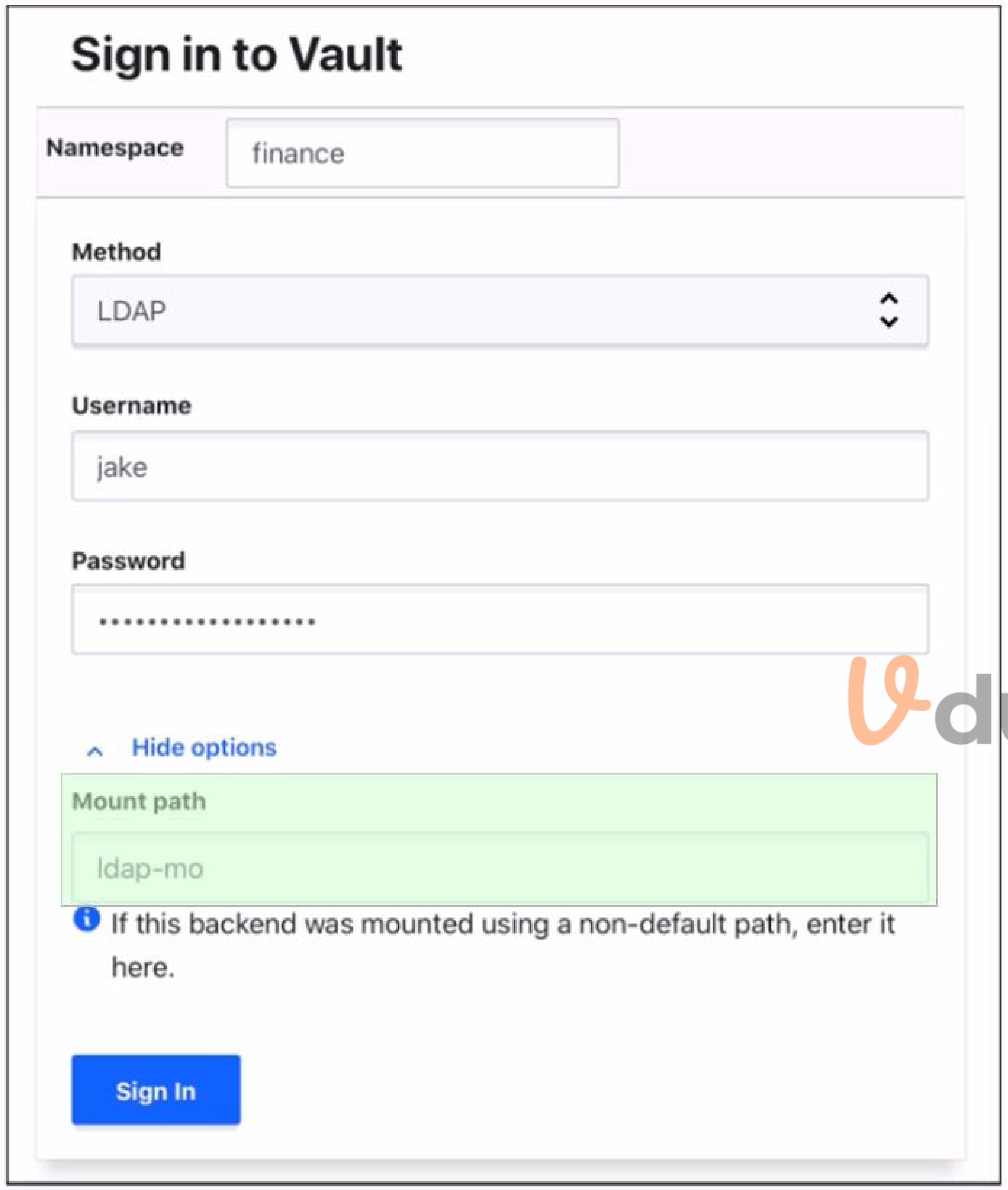
Username: jake

Password: ••••••••••••••••••

Mount path: ldap-mo

If this backend was mounted using a non-default path, enter it here.

Answer Area:


Sign in to Vault

Namespace: finance

Method: LDAP

Username: jake

Password: ••••••••••••••••••

Mount path: ldap-mo

If this backend was mounted using a non-default path, enter it here.

Section:
Explanation:

QUESTION 3
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.

Correct Answer: D
Section:
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

QUESTION 4
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

Correct Answer: D
Section:
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

QUESTION 5
What is the Vault CLI command to query information about the token the client is currently using?

A. vault lookup token
B. vault token lookup
C. vault lookup self
D. vault self-lookup

Correct Answer: B
Section:
Explanation:
The Vault CLI command to query information about the token the client is currently using is vault token lookup. This command displays information about the token or accessor provided as an argument, or the locally authenticated token if no argument is given. The information includes the token ID, accessor, policies, TTL, creation time, and metadata. This command can be useful for debugging and auditing purposes, as well as for renewing or revoking tokens.Reference:token lookup - Command | Vault | HashiCorp Developer,Tokens | Vault | HashiCorp Developer

QUESTION 6
Which of the following is a machine-oriented Vault authentication backend?

A. Okta

B. AppRole
C. Transit
D. GitHub

Correct Answer: B
Section:
Explanation:
AppRole is a machine-oriented authentication method that allows machines or applications to authenticate with Vault using a role ID and a secret ID. The role ID is a unique identifier for the application, and the secret ID is a single-use credential that can be delivered to the application securely.AppRole is designed to provide secure introduction of machines and applications to Vault, and to support the principle of least privilege by allowing fine-grained access control policies to be attached to each role1.
Okta, GitHub, and Transit are not machine-oriented authentication methods.Okta and GitHub are user-oriented authentication methods that allow users to authenticate with Vault using their Okta or GitHub credentials23.Transit is not an authentication method at all, but a secrets engine that provides encryption as a service4.
AppRole Auth Method | Vault | HashiCorp Developer
Okta Auth Method | Vault | HashiCorp Developer
GitHub Auth Method | Vault | HashiCorp Developer
Transit Secrets Engine | Vault | HashiCorp Developer

QUESTION 7
Security requirements demand that no secrets appear in the shell history. Which command does not meet this requirement?

A. generate-password | vault kv put secret/password value
B. vault kv put secret/password value-itsasecret
C. vault kv put secret/password value=@data.txt
D. vault kv put secret/password value-SSECRET_VALUE

Correct Answer: B
Section:
Explanation:
The command that does not meet the security requirement of not having secrets appear in the shell history is B. vault kv put secret/password value-itsasecret. This command would store the secret value ''itsasecret'' in the key/value secrets engine at the path secret/password, but it would also expose the secret value in the shell history, which could be accessed by other users or malicious actors. This is not a secure way of storing secrets in Vault.
The other commands are more secure ways of storing secrets in Vault without revealing them in the shell history. A. generate-password | vault kv put secret/password value would use a pipe to pass the output of the generate-password command, which could be a script or a tool that generates a random password, to the vault kv put command, which would store the password in the key/value secrets engine at the path secret/password. The password would not be visible in the shell history, only the commands. C. vault kv put secret/password value=@data.txt would use the @ syntax to read the secret value from a file named data.txt, which could be encrypted or protected by file permissions, and store it in the key/value secrets engine at the path secret/password. The file name would be visible in the shell history, but not the secret value. D. vault kv put secret/password value-SSECRET_VALUE would use the -S syntax to read the secret value from the environment variable SECRET_VALUE, which could be set and unset in the shell session, and store it in the key/value secrets engine at the path secret/password. The environment variable name would be visible in the shell history, but not the secret value.
[Write Secrets | Vault | HashiCorp Developer]

QUESTION 8
You can build a high availability Vault cluster with any storage backend.

A. True
B. False

Correct Answer: B
Section:
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

QUESTION 9
What command creates a secret with the key 'my-password' and the value '53cr3t' at path 'my-secrets' within the KV secrets engine mounted at 'secret'?

A. vault kv put secret/my-secrets/my-password 53cr3t
B. vault kv write secret/my-secrets/my-password 53cr3t
C. vault kv write 53cr3t my-secrets/my-password
D. vault kv put secret/my-secrets y-password-53cr3t

Correct Answer: A
Section:
Explanation:
The vault kv put command writes the data to the given path in the K/V secrets engine. The command requires the mount path of the K/V secrets engine, the secret path, and the key-value pair to store. The mount path can be specified with the -mount flag or as part of the secret path. The key-value pair can be given as an argument or read from a file or stdin. The correct syntax for the command is:
vault kv put -mount=secret my-secrets/my-password 53cr3t
or
vault kv put secret/my-secrets my-password=53cr3t
The other options are incorrect because they use the deprecated vault kv write command, or they have the wrong order or format of the arguments.Reference:
https://developer.hashicorp.com/vault/docs/commands/kv/put3, https://developer.hashicorp.com/vault/docs/commands/kv4

QUESTION 10
What can be used to limit the scope of a credential breach?

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

Correct Answer: C
Section:
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

QUESTION 11
What environment variable overrides the CLI's default Vault server address?

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

Correct Answer: B
Section:

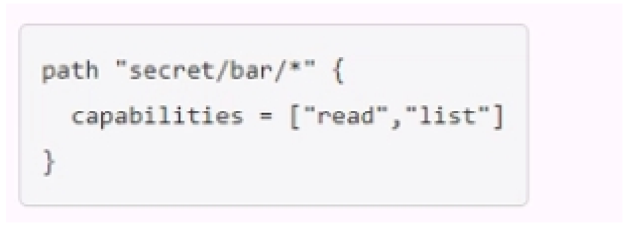
Explanation:
The environment variable VAULT_ADDR overrides the CLI's default Vault server address. The VAULT_ADDR environment variable specifies the address of the Vault server that is used to communicate with Vault from other applications or processes. By setting this variable, you can avoid hard-coding 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

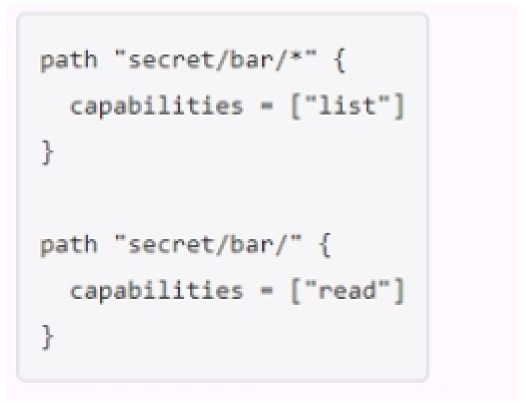
QUESTION 12
Which of the following statements describe the CLI command below?
S vault login -method-1dap username-mitche11h

A. Generates a token which is response wrapped
B. You will be prompted to enter the password
C. By default the generated token is valid for 24 hours
D. Fails because the password is not provided

Correct Answer: A
Section:
Explanation:
The CLI command vault login -method ldap username=mitchellh generates a token that is response wrapped. This means that the token contains a base64-encoded response wrapper, which is a JSON object that contains information about the token, such as its policies, metadata, and expiration time. The response wrapper is used to verify the authenticity and integrity of the token, and to prevent replay attacks. The response wrapper also allows Vault to automatically renew the token when it expires, or to revoke it if it is compromised. The -method ldap option specifies that the authentication method is LDAP, which requires a username and password to be provided. The username mitchellh is an example of an LDAP user name, and the password will be hidden when entered.Reference:Vault CLI Reference | Vault | HashiCorp Developer,Vault CLI Reference | Vault | HashiCorp Developer

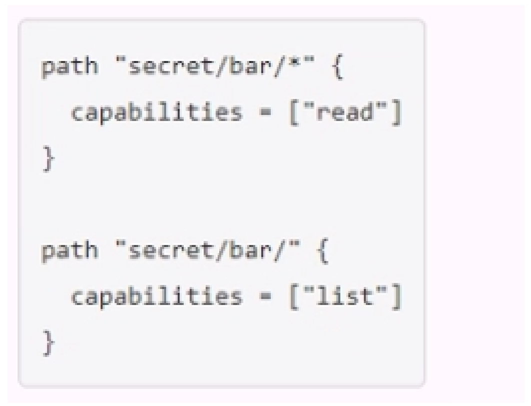
QUESTION 13
You have been tasked with writing a policy that will allow read permissions for all secrets at path secret/bar. The users that are assigned this policy should also be able to list the secrets. What should this policy look like?

A.

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

B.

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

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

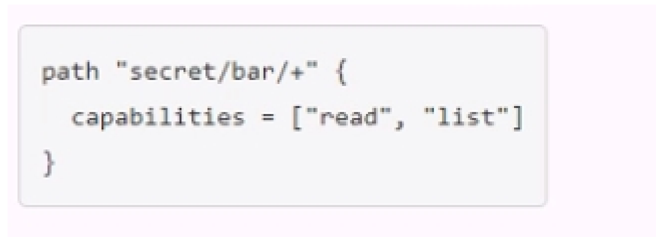
C.



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

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

D.



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

Correct Answer: C
Section:
Explanation:
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.
Policy Syntax | Vault | HashiCorp Developer
Policy Syntax | Vault | HashiCorp Developer
Policies | Vault | HashiCorp Developer

QUESTION 14
When using Integrated Storage, which of the following should you do to recover from possible data loss?

A. Failover to a standby node
B. Use snapshot
C. Use audit logs
D. Use server logs

Correct Answer: B
Section:
Explanation:
Integrated Storage is a Raft-based storage backend that allows Vault to store its data internally without relying on an external storage system. It also enables Vault to run in high availability mode with automatic leader election and failover. However, Integrated Storage is not immune to data loss or corruption due to hardware failures, network partitions, or human errors. Therefore, it is recommended to use the snapshot feature to backup and restore the Vault data periodically or on demand. A snapshot is a point-in-time capture of the entire Vault data, including the encrypted secrets, the configuration, and the metadata. Snapshots can be taken and restored using the vault operator raft snapshot command or the sys/storage/raft/snapshot API endpoint. Snapshots are encrypted and can only be restored with a quorum of unseal keys or recovery keys.Snapshots are also portable and can be used to migrate data between different Vault clusters or storage backends.Reference: https://developer.hashicorp.com/vault/docs/concepts/integrated-storage1, https://developer.hashicorp.com/vault/docs/commands/operator/raft/snapshot2, https://developer.hashicorp.com/vault/api-docs/system/storage/raft/snapshot3

QUESTION 15
How many Shamir's key shares are required to unseal a Vault instance?

A. All key shares
B. A quorum of key shares
C. One or more keys
D. The threshold number of key shares

Correct Answer: D
Section:
Explanation:
Shamir's Secret Sharing is a cryptographic algorithm that allows a secret to be split into multiple parts, called key shares, such that a certain number of key shares are required to reconstruct the secret. The number of key shares and the threshold number are configurable parameters that depend on the desired level of security and availability. Vault uses Shamir's Secret Sharing to protect its master key, which is used to encrypt and decrypt the data encryption key that secures the Vault data. When Vault is initialized, it generates a master key and splits it into a configured number of key shares, which are then distributed to trusted operators. To unseal Vault, the threshold number of key shares must be provided to reconstruct the master key and decrypt the data encryption key.This process ensures that no single operator can access the Vault data without the cooperation of other key holders.Reference: https://developer.hashicorp.com/vault/docs/concepts/seal4, https://developer.hashicorp.com/vault/docs/commands/operator/init5, https://developer.hashicorp.com/vault/docs/commands/operator/unseal6

QUESTION 16
Which of these are a benefit of using the Vault Agent?

A. Vault Agent allows for centralized configuration of application secrets engines
B. Vault Agent will auto-discover which authentication mechanism to use
C. Vault Agent will enforce minimum levels of encryption an application can use
D. Vault Agent will manage the lifecycle of cached tokens and leases automatically

Correct Answer: D
Section:
Explanation:
Vault Agent is a client daemon that provides the following features:
Auto-Auth - Automatically authenticate to Vault and manage the token renewal process for locally-retrieved dynamic secrets.
API Proxy - Allows Vault Agent to act as a proxy for Vault's API, optionally using (or forcing the use of) the Auto-Auth token.
Caching - Allows client-side caching of responses containing newly created tokens and responses containing leased secrets generated off of these newly created tokens. The agent also manages the renewals of the cached tokens and leases.
Templating - Allows rendering of user-supplied templates by Vault Agent, using the token generated by the Auto-Auth step.
Process Supervisor Mode - Runs a child process with Vault secrets injected as environment variables.
One of the benefits of using the Vault Agent is that it will manage the lifecycle of cached tokens and leases automatically. This means that the agent will handle the token renewal and revocation logic, as well as the lease renewal and revocation logic for the secrets that are cached by the agent. This reduces the burden on the application developers and operators, and ensures that the tokens and secrets are always valid and up-to-date.Reference:Vault Agent | Vault | HashiCorp Developer,Caching - Vault Agent | Vault | HashiCorp Developer

QUESTION 17
Which of the following describes usage of an identity group?

A. Limit the policies that would otherwise apply to an entity in the group
B. When they want to revoke the credentials for a whole set of entities simultaneously
C. Audit token usage
D. Consistently apply the same set of policies to a collection of entities

Correct Answer: D
Section:
Explanation:
An identity group is a collection of entities that share some common attributes. An identity group can have one or more policies attached to it, which are inherited by all the members of the group. An identity group can also have subgroups, which can further refine the policies and attributes for a subset of entities.
One of the use cases of an identity group is to consistently apply the same set of policies to a collection of entities. For example, an organization may have different teams or departments, such as engineering, sales, or marketing. Each team may have its own identity group, with policies that grant access to the secrets and resources that are relevant to their work. By creating an identity group for each team, the organization can ensure that the entities belonging to each team have the same level of access and permissions, regardless of which authentication method they use to log in to Vault.Reference:Identity: entities and groups | Vault | HashiCorp Developer,vault_identity_group | Resources | hashicorp/vault | Terraform | Terraform Registry

QUESTION 18
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

Correct Answer: C
Section:
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

QUESTION 19
Where does the Vault Agent store its cache?

A. In a file encrypted using the Vault transit secret engine
B. In the Vault key/value store
C. In an unencrypted file
D. In memory

Correct Answer: D
Section:
Explanation:
The Vault Agent stores its cache in memory, which means that it does not persist the cached tokens and secrets to disk or any other storage backend. This makes the cache more secure and performant, as it avoids exposing the sensitive data to potential attackers or unauthorized access. However, this also means that the cache is volatile and will be lost if the agent process is terminated or restarted. To mitigate this, the agent can optionally use a persistent cache file to restore the tokens and leases from a previous agent process. The persistent cache file is encrypted using a key derived from the agent's auto-auth token and a nonce, and it is stored in a user-specified location on disk.Reference:Caching - Vault Agent | Vault | HashiCorp Developer,Vault Agent Persistent Caching | Vault | HashiCorp Developer

QUESTION 20
Your organization has an initiative to reduce and ultimately remove the use of long lived X.509 certificates. Which secrets engine will best support this use case?

A. PKI
B. Key/Value secrets engine version 2, with TTL defined
C. Cloud KMS

D. Transit

Correct Answer: A
Section:
Explanation:
The PKI secrets engine is designed to support the use case of reducing and ultimately removing the use of long lived X.509 certificates. The PKI secrets engine can generate dynamic X.509 certificates on demand, with short time-to-live (TTL) and automatic revocation. This eliminates the need for manual processes of generating, signing, and rotating certificates, and reduces the risk of certificate compromise or misuse. The PKI secrets engine can also act as a certificate authority (CA) or an intermediate CA, and can integrate with external CAs or CRLs.The PKI secrets engine can issue certificates for various purposes, such as TLS, SSH, code signing, email encryption, etc.Reference: https://developer.hashicorp.com/vault/docs/secrets/pki1, https://developer.hashicorp.com/vault/tutorials/getting-started/getting-started-dynamic-secrets

QUESTION 21
When unsealing Vault, each Shamir unseal key should be entered:

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

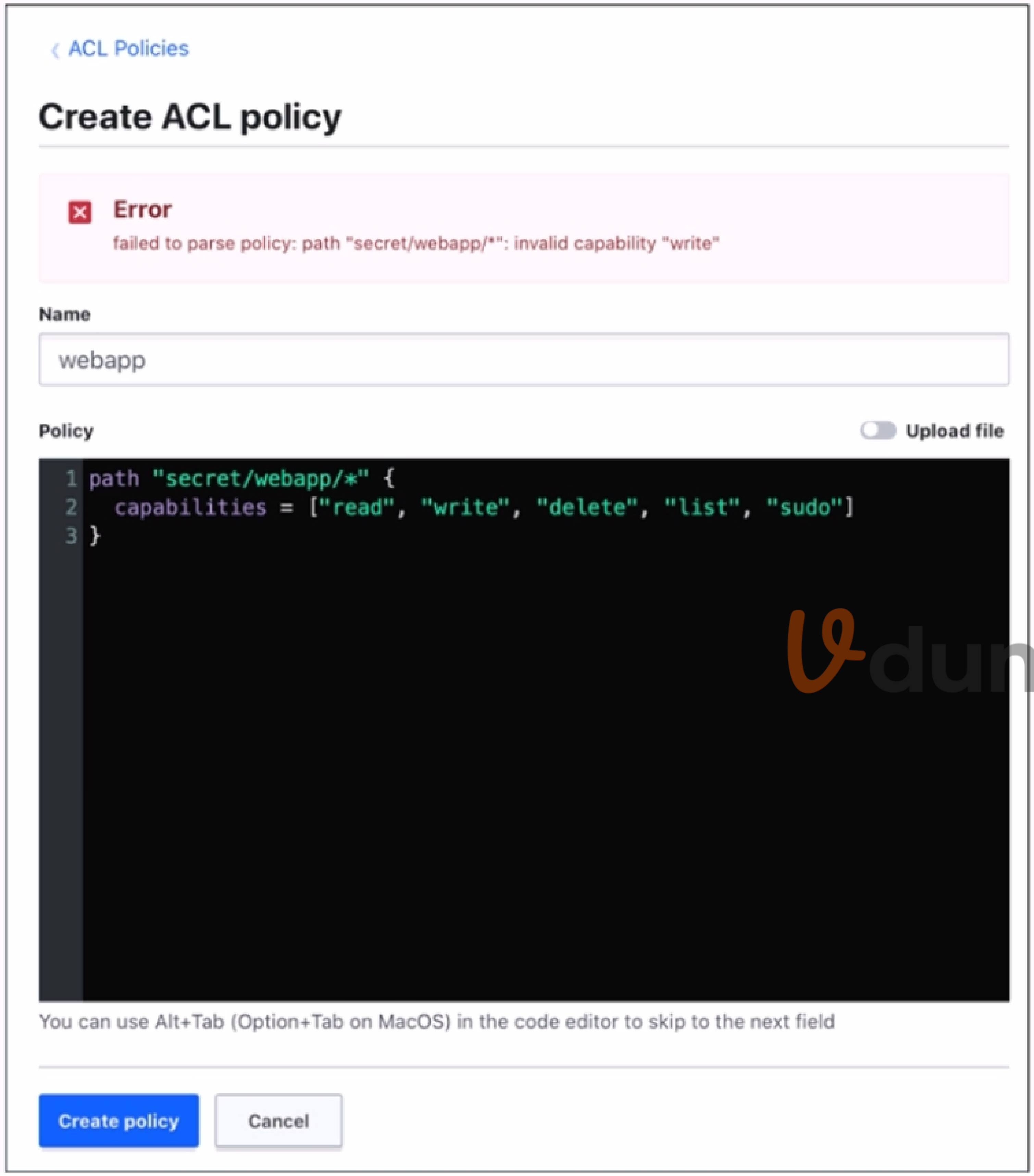
Correct Answer: B
Section:
Explanation:
When unsealing Vault, each Shamir unseal key should be entered by different administrators each connecting from different computers. This is because the Shamir unseal keys are split into shares that are distributed to trusted operators, and no single operator should 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 22
As a best practice, the root token should be stored in which of the following ways?

A. Should be revoked and never stored after initial setup
B. Should be stored in configuration automation tooling
C. Should be stored in another password safe
D. Should be stored in Vault

Correct Answer: A
Section:
Explanation:
The root token is the initial token created when initializing Vault. It has unlimited privileges and can perform any operation in Vault. As a best practice, the root token should be revoked and never stored after initial setup. This is because the root token is a single point of failure and a potential security risk if it is compromised or leaked. Instead of using the root token, Vault operators should create other tokens with appropriate policies and roles that allow them to perform their tasks. If a new root token is needed in an emergency, the vault operator generate-root command can be used to create one on-the-fly with the consent of a quorum of unseal key holders.Reference:Tokens | Vault | HashiCorp Developer,Generate root tokens using unseal keys | Vault | HashiCorp Developer

QUESTION 23
When creating a policy, an error was thrown:



ACL Policies

Create ACL policy

Error

failed to parse policy: path "secret/webapp/*": invalid capability "write"

Name

webapp

Policy

Upload file

```
path "secret/webapp/*" {
  capabilities = ["read", "write", "delete", "list", "sudo"]
}
```

You can use Alt+Tab (Option+Tab on MacOS) in the code editor to skip to the next field

Create policy   Cancel

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
Section:
Explanation:
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.
[Policy Syntax | Vault | HashiCorp Developer]
[Policy Syntax | Vault | HashiCorp Developer]

QUESTION 24
Where can you set the Vault seal configuration? Choose two correct answers.

A. Cloud Provider KMS
B. Vault CLI
C. Vault configuration file
D. Environment variables
E. Vault API

Correct Answer: C, D
Section:
Explanation:
The Vault seal configuration can be set in two ways: through the Vault configuration file or through environment variables. The Vault configuration file is a text file that contains the settings and options for Vault, such as the storage backend, the listener, the telemetry, and the seal. The seal stanza in the configuration file specifies the seal type and the parameters to use for additional data protection, such as using HSM or Cloud KMS solutions to encrypt and decrypt the root key. The seal configuration can also be set through environment variables, which will take precedence over the values in the configuration file. The environment variables are prefixed with VAULT_SEAL_ and followed by the seal type and the parameter name. For example, VAULT_SEAL_AWSKMS_REGION sets the region for the AWS KMS seal.Reference:Seals - Configuration | Vault | HashiCorp Developer,Environment Variables | Vault | HashiCorp Developer

QUESTION 25
Which of the following vault lease operations uses a lease _ id as an argument? Choose two correct answers.

A. renew
B. revoke -prefix
C. create
D. describe
E. revoke

Correct Answer: A, E
Section:
Explanation:
The vault lease operations that use a lease_id as an argument are renew and revoke. The renew operation allows a client to extend the validity of a lease associated with a secret or a token. The revoke operation allows a client to terminate a lease immediately and invalidate the secret or the token. Both operations require a lease_id as an argument to identify the lease to be renewed or revoked. The lease_id can be obtained from the response of reading a secret or creating a token, or from the vault lease list command. The other operations, revoke-prefix, create, and describe, do not use a lease_id as an argument. The revoke-prefix operation allows a client to revoke all secrets or tokens generated under a given prefix. The create operation allows a client to create a new lease for a secret. The describe operation allows a client to view information about a lease, such as its TTL, policies, and metadata.Reference:Lease, Renew, and Revoke | Vault | HashiCorp Developer,vault lease - Command | Vault | HashiCorp Developer

QUESTION 26
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.

A. True
B. False

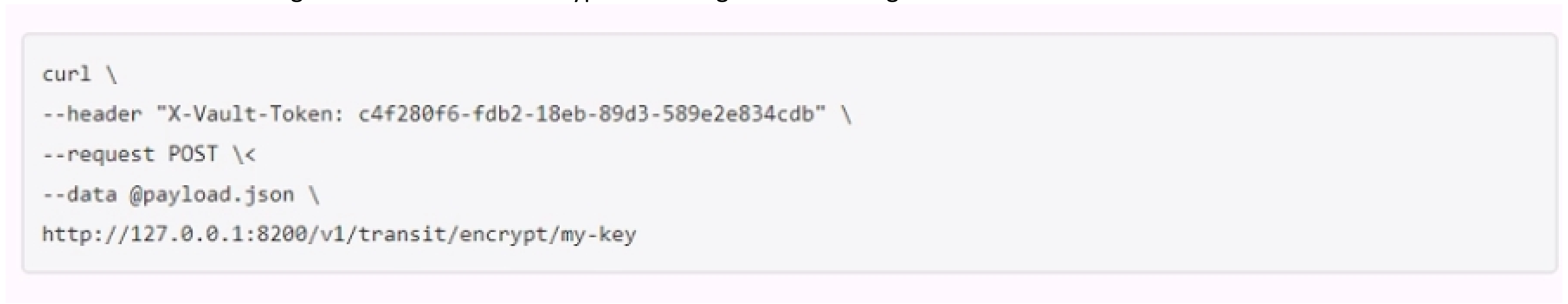
Correct Answer: B
Section:
Explanation:
The statement is false. An organization can authenticate an AWS EC2 virtual machine with Vault to access a dynamic database secret using more than one authentication method. The AWS auth method is one of the options, but not the only one. The AWS auth method supports 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 27
You are using Vault's Transit secrets engine to encrypt your dat
a. You want to reduce the amount of content encrypted with a single key in case the key gets compromised. How would you do this?

A. Use 4096-bit RSA key to encrypt the data
B. Upgrade to Vault Enterprise and integrate with HSM
C. Periodically re-key the Vault's unseal keys
D. Periodically rotate the encryption key

Correct Answer: D
Section:
Explanation:
The Transit secrets engine supports the rotation of encryption keys, which allows you to change the key that is used to encrypt new data without affecting the ability to decrypt data that was already encrypted. This reduces the amount of content encrypted with a single key in case the key gets compromised, and also helps you comply with the NIST guidelines for key rotation. You can rotate the encryption key manually by invoking the /transit/keys/<name>/rotate endpoint, or you can configure the key to automatically rotate based on a time interval or a number of encryption operations. When you rotate a key, Vault generates a new key version and increments the key's latest_version metadata. The new key version becomes the encryption key used for encrypting any new data. The previous key versions are still available for decrypting the existing data, unless you specify a minimum decryption version to archive the old key versions.You can also delete or disable old key versions if you want to revoke access to the data encrypted with those versions.Reference: https://developer.hashicorp.com/vault/docs/secrets/transit1, https://developer.hashicorp.com/vault/api-docs/secret/transit2

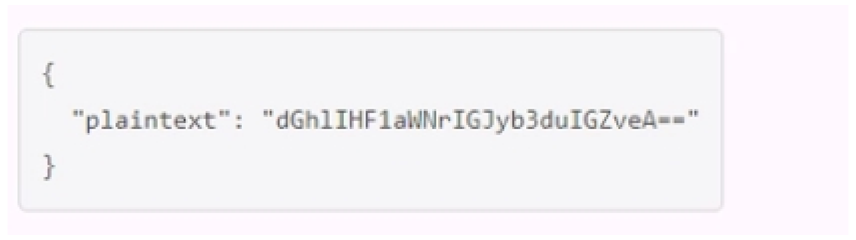
QUESTION 28
A user issues the following cURL command to encrypt data using the transit engine and the Vault AP:



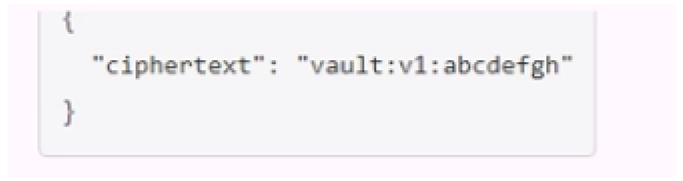
```
curl \
--header "X-Vault-Token: c4f280f6-fdb2-18eb-89d3-589e2e834cdb" \
--request POST \<
--data @payload.json \
http://127.0.0.1:8200/v1/transit/encrypt/my-key
```

Which payload.json file has the correct contents?

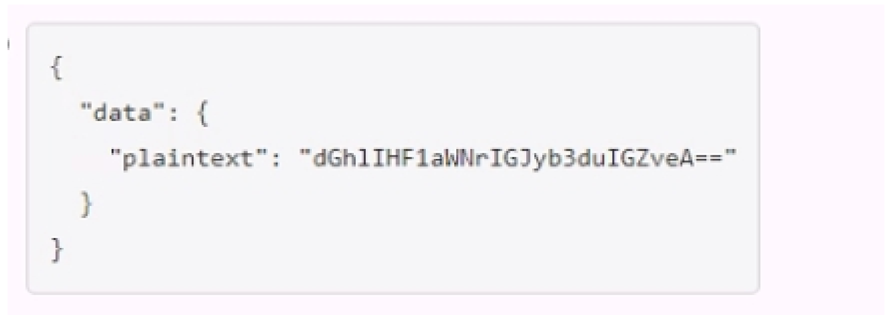
A.


```
{
  "plaintext": "dGhlIHF1aWNrIGJyb3duIGZveA=="
}
```

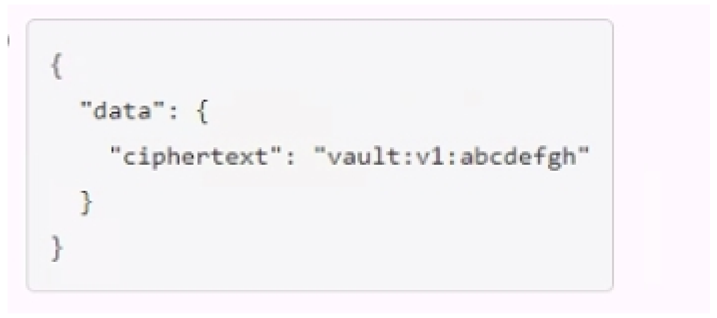
B.


```
{
  "ciphertext": "vault:v1:abcdefgh"
}
```

C.


```
{
  "data": {
    "plaintext": "dGhlIHF1aWNrIGJyb3duIGZveA=="
  }
}
```

D.


```
{
  "data": {
    "ciphertext": "vault:v1:abcdefgh"
  }
}
```

Correct Answer: C
Section:
Explanation:
The payload.json file that has the correct contents is C. This file contains a JSON object with a single key, ''plaintext'', and a value that is the base64-encoded string of the data to be encrypted.This is the format that the Vault API expects for the transit encrypt endpoint1. The other files are not correct because they either have the wrong key name, the wrong value format, or the wrong JSON syntax.
Encrypt Data - Transit Secrets Engine | Vault | HashiCorp Developer

QUESTION 29
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
Section:
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

QUESTION 30
A web application uses Vault's transit secrets engine to encrypt data in-transit. If an attacker intercepts the data in transit which of the following statements are true? Choose two correct answers.

A. You can rotate the encryption key so that the attacker won't be able to decrypt the data
B. The keys can be rotated and min_decryption_version moved forward to ensure this data cannot be decrypted
C. The Vault administrator would need to seal the Vault server immediately
D. Even if the attacker was able to access the raw data, they would only have encrypted bits (TLS in transit)

Correct Answer: B, D
Section:
Explanation:
A web application that uses Vault's transit secrets engine to encrypt data in-transit can benefit from the following security features:
Even if the attacker was able to access the raw data, they would only have encrypted bits (TLS in transit). This means that the attacker would need to obtain the encryption key from Vault in order to decrypt the data, which is protected by Vault's authentication and authorization mechanisms. The transit secrets engine does not store the data sent to it, so the attacker cannot access the data from Vault either.
The keys can be rotated and min_decryption_version moved forward to ensure this data cannot be decrypted. This means that the web application can periodically change the encryption key used to encrypt the data, and set a minimum decryption version for the key, which prevents older versions of the key from being used to decrypt the data. This way, even if the attacker somehow obtained an old version of the key, they would not be able to decrypt the data that was encrypted with a newer version of the key.
The other statements are not true, because:
You cannot rotate the encryption key so that the attacker won't be able to decrypt the data. Rotating the key alone does not prevent the attacker from decrypting the data, as they may still have access to the old version of the key that was used to encrypt the data. You need to also move the min_decryption_version forward to invalidate the old version of the key.
The Vault administrator would not need to seal the Vault server immediately. Sealing the Vault server would make it inaccessible to both the attacker and the legitimate users, and would require unsealing it with the unseal keys or the recovery keys. Sealing the Vault server is a last resort option in case of a severe compromise or emergency, and is not necessary in this scenario, as the attacker does not have access to the encryption key or the data in Vault.Reference:Transit - Secrets Engines | Vault | HashiCorp Developer,Encryption as a service: transit secrets engine | Vault | HashiCorp Developer

QUESTION 31
The Vault encryption key is stored in Vault's backend storage.

A. True
B. False

Correct Answer: B
Section:
Explanation:
The statement is false. The Vault encryption key is not stored in Vault's backend storage, but rather in Vault's memory. The Vault encryption key is the key that is used to encrypt and decrypt the data that is stored in Vault's backend storage, such as secrets, tokens, policies, etc. The Vault encryption key is derived from the master key, which is generated when Vault is initialized. The master key is split into unseal keys using Shamir's secret sharing algorithm, and the unseal keys are distributed to trusted operators. To start Vault, a quorum of unseal keys is required to reconstruct the master key and derive the encryption key. The encryption key is then kept in memory and used to protect the data in Vault's backend storage. The encryption key is never written to disk or exposed via the API.Reference:Seal/Unseal | Vault | HashiCorp Developer,Key Rotation | Vault | HashiCorp Developer

QUESTION 32
Which of the following statements describe the secrets engine in Vault? Choose three correct answers.

A. Some secrets engines simply store and read data
B. Once enabled, you cannot disable the secrets engine

C. You can build your own custom secrets engine
D. Each secrets engine is isolated to its path
E. A secrets engine cannot be enabled at multiple paths

Correct Answer: A, C, D
Section:
Explanation:
Secrets engines are components that store, generate, or encrypt data in Vault. They are enabled at a specific path in Vault and have their own API and configuration. Some of the statements that describe the secrets engines in Vault are:
Some secrets engines simply store and read data, such as the key/value secrets engine, which acts like an encrypted Redis or Memcached.Other secrets engines perform more complex operations, such as generating dynamic credentials, encrypting data, issuing certificates, etc1.
You can build your own custom secrets engine by using the plugin system, which allows you to write and run your own secrets engine as a separate process that communicates with Vault over gRPC.You can also use the SDK to create your own secrets engine in Go and compile it into Vault2.
Each secrets engine is isolated to its path, which means that the secrets engine cannot access or interact with other secrets engines or data outside its path. The path where the secrets engine is enabled can be customized and can have multiple segments.For example, you can enable the AWS secrets engine at aws/ or aws/prod/ or aws/dev/3.
The statements that are not true about the secrets engines in Vault are:
You can disable an existing secrets engine by using the vault secrets disable command or the sys/mounts API endpoint.When a secrets engine is disabled, all of its secrets are revoked and all of its data is deleted from the storage backend4.
A secrets engine can be enabled at multiple paths, with a few exceptions, such as the system and identity secrets engines. Each secrets engine enabled at a different path is independent and isolated from others.For example, you can enable the KV secrets engine at kv/ and secret/ and they will not share any data3.

QUESTION 33
What is a benefit of response wrapping?

A. Log every use of a secret
B. Load balanc secret generation across a Vault cluster
C. Provide error recovery to a secret so it is not corrupted in transit
D. Ensure that only a single party can ever unwrap the token and see what's inside

Correct Answer: D
Section:
Explanation:
Response wrapping is a feature that allows Vault to take the response it would have sent to a client and instead insert it into the cubbyhole of a single-use token, returning that token instead. The client can then unwrap the token and retrieve the original response. Response wrapping has several benefits, such as providing cover, malfeasance detection, and lifetime limitation for the secret data. One of the benefits is to ensure that only a single party can ever unwrap the token and see what's inside, as the token can be used only once and cannot be unwrapped by anyone else, even the root user or the creator of the token.This provides a way to securely distribute secrets to the intended recipients and detect any tampering or interception along the way5.
The other options are not benefits of response wrapping:
Log every use of a secret: Response wrapping does not log every use of a secret, as the secret is not directly exposed to the client or the network.However, Vault does log the creation and deletion of the response-wrapping token, and the client can use the audit device to log the unwrapping operation6.
Load balance secret generation across a Vault cluster: Response wrapping does not load balance secret generation across a Vault cluster, as the secret is generated by the Vault server that receives the request and the response-wrapping token is bound to that server.However, Vault does support high availability and replication modes that can distribute the load and improve the performance of the cluster7.
Provide error recovery to a secret so it is not corrupted in transit: Response wrapping does not provide error recovery to a secret so it is not corrupted in transit, as the secret is encrypted and stored in the cubbyhole of the token and cannot be modified or corrupted by anyone. However, if the token is lost or expired, the secret cannot be recovered either, so the client should have a backup or retry mechanism to handle such cases.

QUESTION 34
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
Section:
Explanation:
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.Reference:Auth Methods | Vault | HashiCorp Developer,Authentication - Concepts | Vault | HashiCorp Developer

QUESTION 35
Which Vault secret engine may be used to build your own internal certificate authority?

A. Transit
B. PKI
C. PostgreSQL
D. Generic

Correct Answer: B
Section:
Explanation:
The Vault secret engine that can be used to build your own internal certificate authority is the PKI secret engine. The PKI secret engine generates dynamic X.509 certificates on-demand, without requiring manual processes of generating private keys and CSRs, submitting to a CA, and waiting for verification and signing. The PKI secret engine can act as a root CA or an intermediate CA, and can issue certificates for various purposes, such as TLS, code signing, email encryption, etc. The PKI secret engine can also manage the certificate lifecycle, such as rotation, revocation, renewal, and CRL generation. The PKI secret engine can also integrate with external CAs, such as Venafi or Entrust, to delegate the certificate issuance and management.Reference:PKI - Secrets Engines | Vault | HashiCorp Developer,Build Your Own Certificate Authority (CA) | Vault - HashiCorp Learn

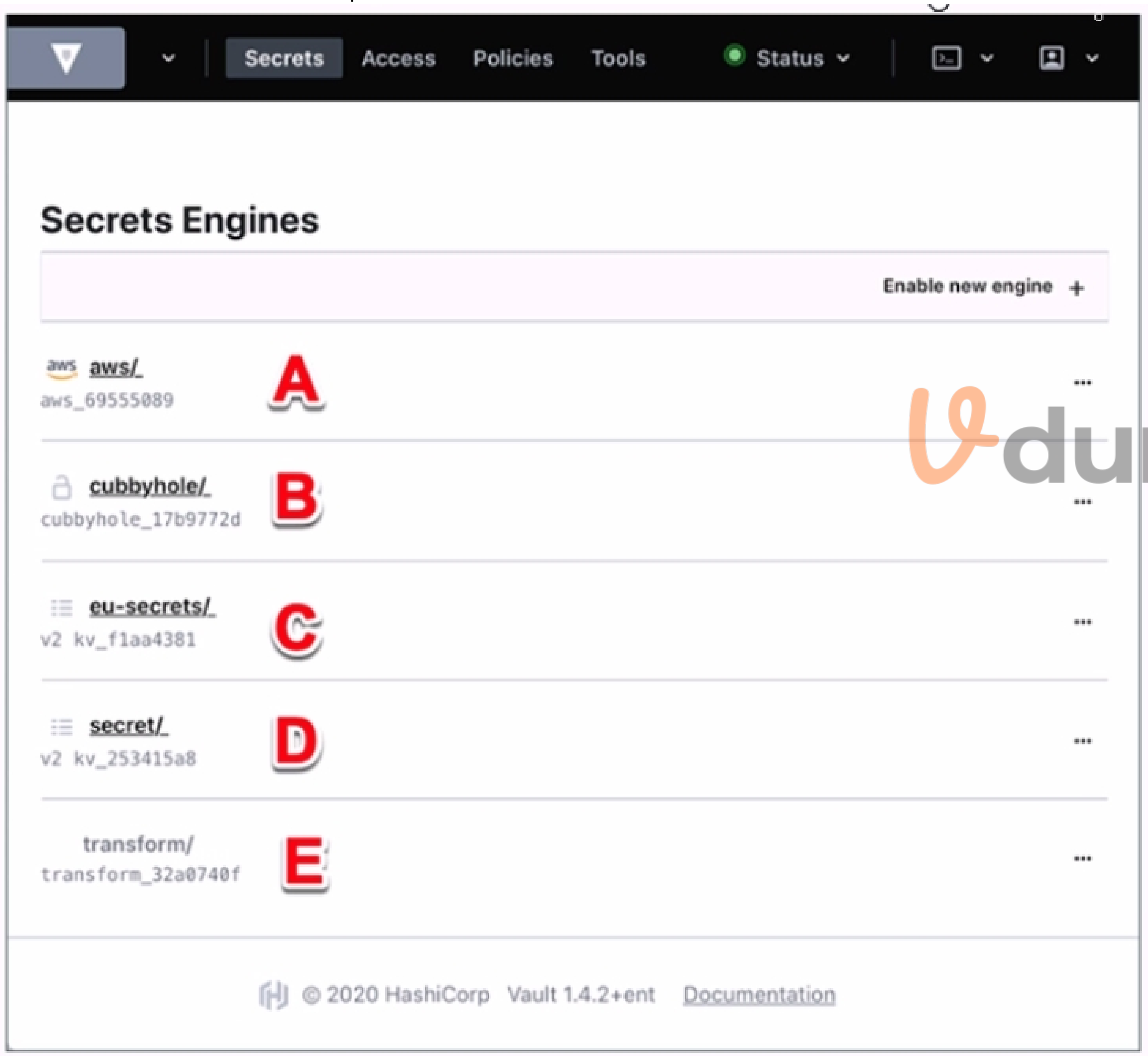
QUESTION 36
Which of the following statements are true about Vault policies? Choose two correct answers.

A. The default policy can not be modified
B. You must use YAML to define policies
C. Policies provide a declarative way to grant or forbid access to certain paths and operations in Vault
D. Vault must be restarted in order for a policy change to take an effect
E. Policies deny by default (empty policy grants no permission)

Correct Answer: C, E
Section:
Explanation:
Vault policies are written in HCL or JSON format and are attached to tokens or roles by name. Policies define the permissions and restrictions for accessing and performing operations on certain paths and secrets in Vault.Policies are deny by default, which means that an empty policy grants no permission in the system, and any request that is not explicitly allowed by a policy is implicitly denied1. Some of the features and benefits of Vault policies are:
Policies are path-based, which means that they match the request path to a set of rules that specify the allowed or denied capabilities, such as create, read, update, delete, list, sudo, etc2.
Policies are additive, which means that if a token or a role has multiple policies attached, the effective policy is the union of all the individual policies.The most permissive capability is granted if there is a conflict3.
Policies can use glob patterns, such as * and +, to match multiple paths or segments with a single rule.For example, path ''secret/*'' matches any path starting with secret/, and path ''secret/+/config'' matches any path with two segments after secret/ and ending with config4.
Policies can use templating to interpolate certain values into the rules, such as identity information, time, randomness, etc.For example, path ''secret/{{identity.entity.id}}/*'' matches any path starting with secret/ followed by

the entity ID of the requester5.
Policies can be managed by using the vault policy commands or the sys/policy API endpoints.You can write, read, list, and delete policies by using these interfaces6.
The default policy is a built-in policy that is attached to all tokens by default and cannot be deleted. However, the default policy can be modified by using the vault policy write command or the sys/policy API endpoint.The default policy provides common permissions for tokens, such as renewing themselves, looking up their own information, creating and managing response-wrapping tokens, etc7.
You do not have to use YAML to define policies, as Vault supports both HCL and JSON formats.HCL is a human-friendly configuration language that is also JSON compatible, which means that JSON can be used as a valid input for policies as well8.
Vault does not need to be restarted in order for a policy change to take effect, as policies are stored and evaluated in memory. Any change to a policy is immediately reflected in the system, and any token or role that has that policy attached will be affected by the change.

QUESTION 37
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

Correct Answer: C
Section:
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.
[KV - Secrets Engines | Vault | HashiCorp Developer]

QUESTION 38
An organization would like to use a scheduler to track & revoke access granted to a job (by Vault) at completion. What auth-associated Vault object should be tracked to enable this behavior?

A. Token accessor
B. Token ID
C. Lease ID
D. Authentication method

Correct Answer: C
Section:
Explanation:
A lease ID is a unique identifier that is assigned by Vault to every dynamic secret and service type authentication token. A lease ID contains information such as the secret path, the secret version, the secret type, etc. A lease ID can be used to track and revoke access granted to a job by Vault at completion, as it allows the scheduler to perform the following operations:
Lookup the lease information by using the vault lease lookup command or the sys/leases/lookup API endpoint. This will return the metadata of the lease, such as the expire time, the issue time, the renewable status, and the TTL.
Renew the lease if needed by using the vault lease renew command or the sys/leases/renew API endpoint. This will extend the validity of the secret or the token for a specified increment, or reset the TTL to the original value if no increment is given.
Revoke the lease when the job is completed by using the vault lease revoke command or the sys/leases/revoke API endpoint. This will invalidate the secret or the token immediately and prevent any further renewals. For example, with the AWS secrets engine, the access keys will be deleted from AWS the moment a lease is revoked.
A lease ID is different from a token ID or a token accessor. A token ID is the actual value of the token that is used to authenticate to Vault and perform requests. A token ID should be treated as a secret and protected from unauthorized access. A token accessor is a secondary identifier of the token that is used for token management without revealing the token ID. A token accessor can be used to lookup, renew, or revoke a token, but not to authenticate to Vault or access secrets. A token ID or a token accessor can be used to revoke the token itself, but not the leases associated with the token. To revoke the leases, a lease ID is required.
An authentication method is a way to verify the identity of a user or a machine and issue a token with appropriate policies and metadata. An authentication method is not an object that can be tracked or revoked, but a configuration that can be enabled, disabled, tuned, or customized by using the vault auth commands or the sys/auth API endpoints.

QUESTION 39
Which statement describes the results of this command: $ vault secrets enable transit

A. Enables the transit secrets engine at transit path
B. Requires a root token to execute the command successfully
C. Enables the transit secrets engine at secret path
D. Fails due to missing -path parameter
E. Fails because the transit secrets engine is enabled by default

Correct Answer: A
Section:

Explanation:
The command vault secrets enable transit enables the transit secrets engine at the transit path. The transit secrets engine is a secrets engine that handles cryptographic functions on data in-transit, such as encryption, decryption, signing, verification, hashing, and random bytes generation. The transit secrets engine does not store the data sent to it, but only performs the requested operations and returns the results. The transit secrets engine can also be viewed as ''cryptography as a service'' or ''encryption as a service''. The command vault secrets enable transit uses the default path of transit for the secrets engine, but this can be changed by using the -path option. For example, vault secrets enable -path=my-transit transit would enable the transit secrets engine at the my-transit path.Reference:Transit - Secrets Engines | Vault | HashiCorp Developer,vault secrets enable - Command | Vault | HashiCorp Developer