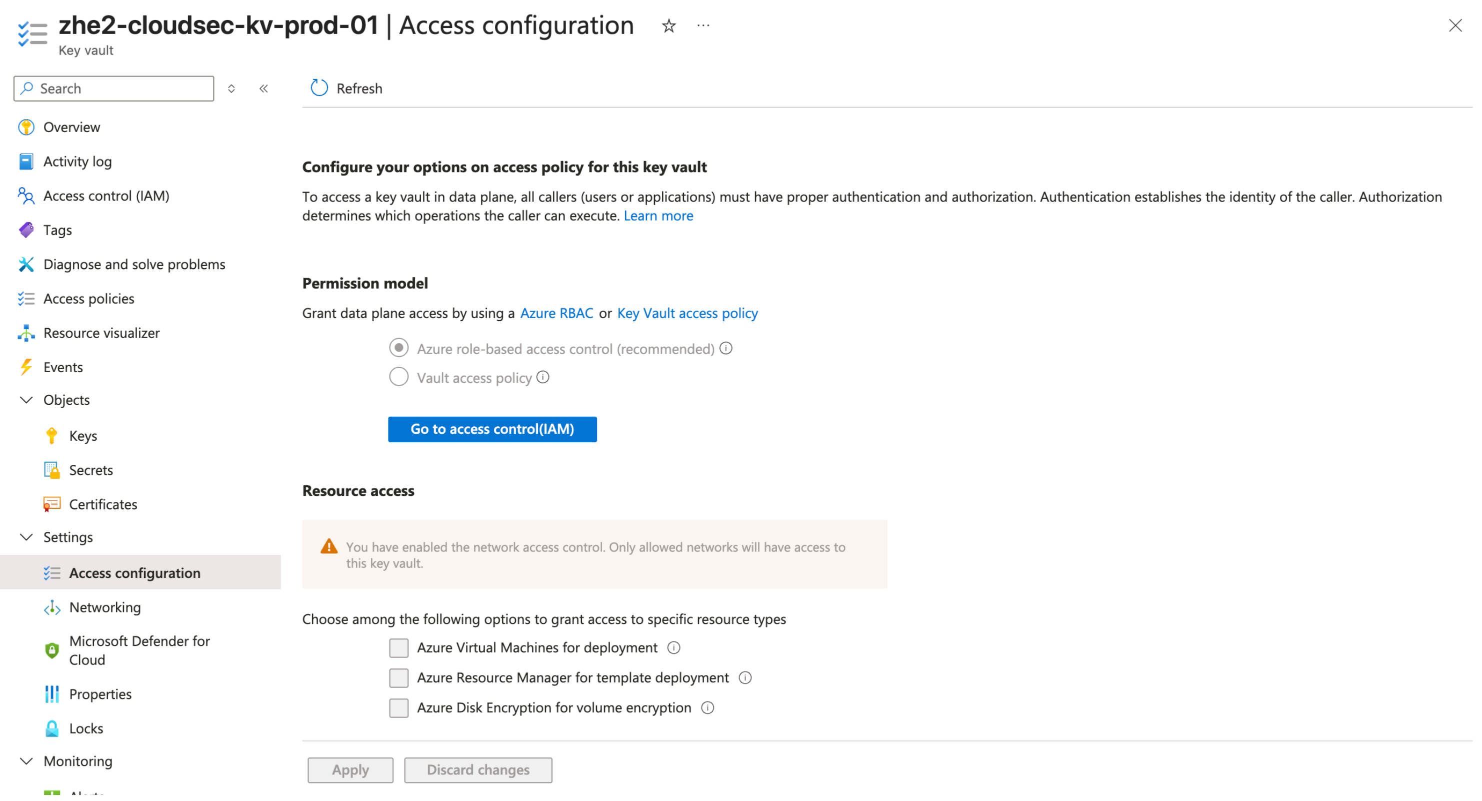
Secure storage of credentials in Azure Key Vaults

[redacted]

Purpose: Guide Cloud Security Team on [redacted] best practices to securely store API keys

Current Creation Process: Azure key vaults requests are currently created and managed by Cloud Security Team and I[redacted] via our ITSM system (ServiceNow).

Current RBAC model: Our current RBAC model for key vaults follows Microsoft recommended best practices, by leveraging Azure RBAC to grant access to key vaults. (See Below).



zhe2-cloudsec-kv-prod-01 | Access configuration

Key vault

Configure your options on access policy for this key vault

To access a key vault in data plane, all callers (users or applications) must have proper authentication and authorization. Authentication establishes the identity of the caller. Authorization determines which operations the caller can execute. Learn more

Permission model

Grant data plane access by using a Azure RBAC or Key Vault access policy

● Azure role-based access control (recommended)

○ Vault access policy

Resource access

You have enabled the network access control. Only allowed networks will have access to this key vault.

Choose among the following options to grant access to specific resource types

□ Azure Virtual Machines for deployment

□ Azure Resource Manager for template deployment

□ Azure Disk Encryption for volume encryption

Azure RBAC roles assigned to access key vaults:

1. `Key Vault Contributor:` RBAC role for control plane operations only to manage key vaults. It does not allow access to keys, secrets and certificates.
2. `Key Vault Administrator:` RBAC Role that has full access to data plane operations including read, write, and delete on all objects.
3. `Key Vault Secrets Officer:` RBAC Role to perform all actions on the ***secrets*** only within a Key Vault but cannot control access to the vault itself
4. `Key Vault Reader` : RBAC role that provides read-only access to key vaults.

Current key and secret creation process (ownership and access controls):

NOTE: No user will be able to manage objects within the key vault or access its objects unless they are given a RBAC role. The requestor's manager must approve all requests for permissions.

RBAC requests:

1. All requests to create a Azure Key Vault or request RBAC role to access the secrets/keys/certificates must come from a ServiceNow ticket and must obtain approval from the requestor's manager.
2. Important details that are needed in ServiceNow ticket

- Purpose of Key Vault creation
- Purpose of object created
- Approval from requestor's manager

Why do we need to securely store secrets, certificates, and API keys?

1. Prevents unauthorized access to sensitive data
2. System integrity
3. Reputational damage
4. Compliance

Common bad practices of storing secrets and API keys:

1. Storing API keys in OneNote or Notes application
2. Hardcoding API keys in source code and not using environmental variables

Process: Store API keys in Azure Key Vaults

Step 1: Create a key vault following 's naming conventions.

- <region>-<workload>-kv-<environment>-<instance>

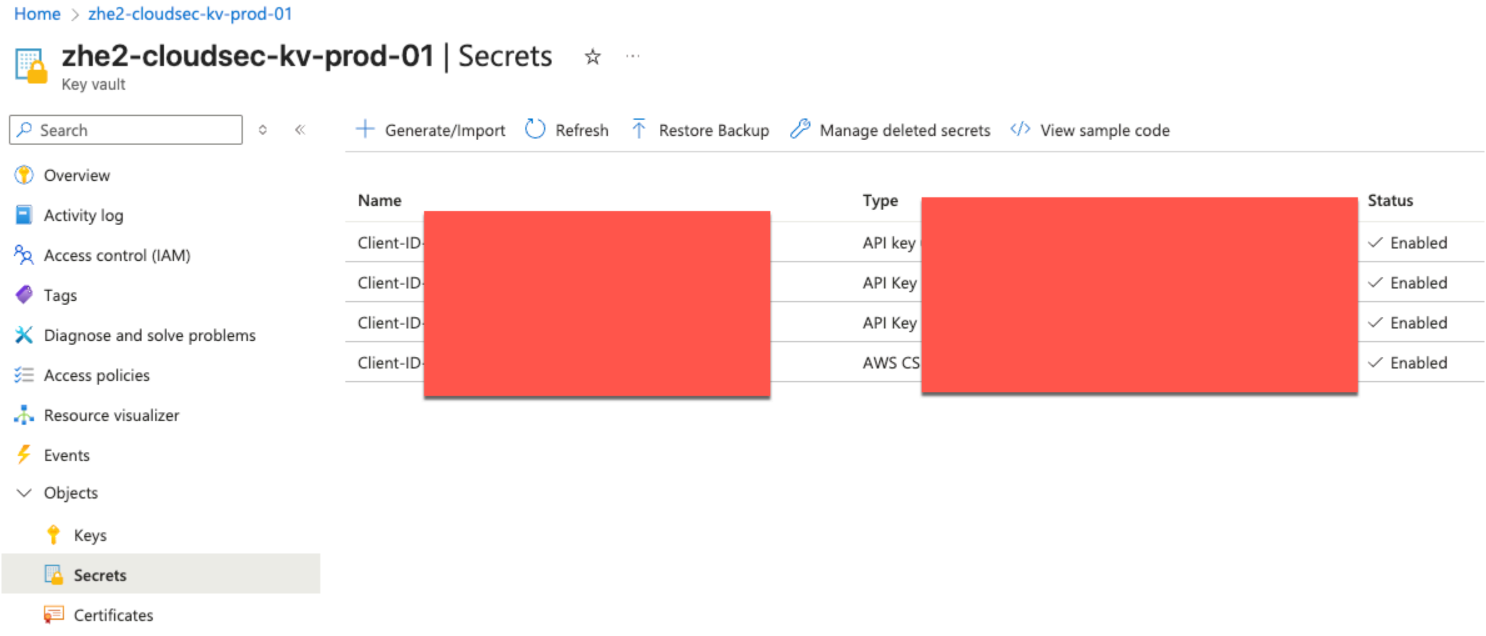
Example:

- zhe2-epic-kv-dev-01
- zhe2-clinical-kv-dev-01

Step 2: For API keys or passwords:

- Select key vault
- Store under Objects > Secrets > Select Generate/import
- Provide Client ID
- Enter Description and API key details
- Click "Save"

Note: In below screenshot, I have saved all our Falcon CrowdStrike API keys in our Key Vault with detailed description of the purpose of each key.

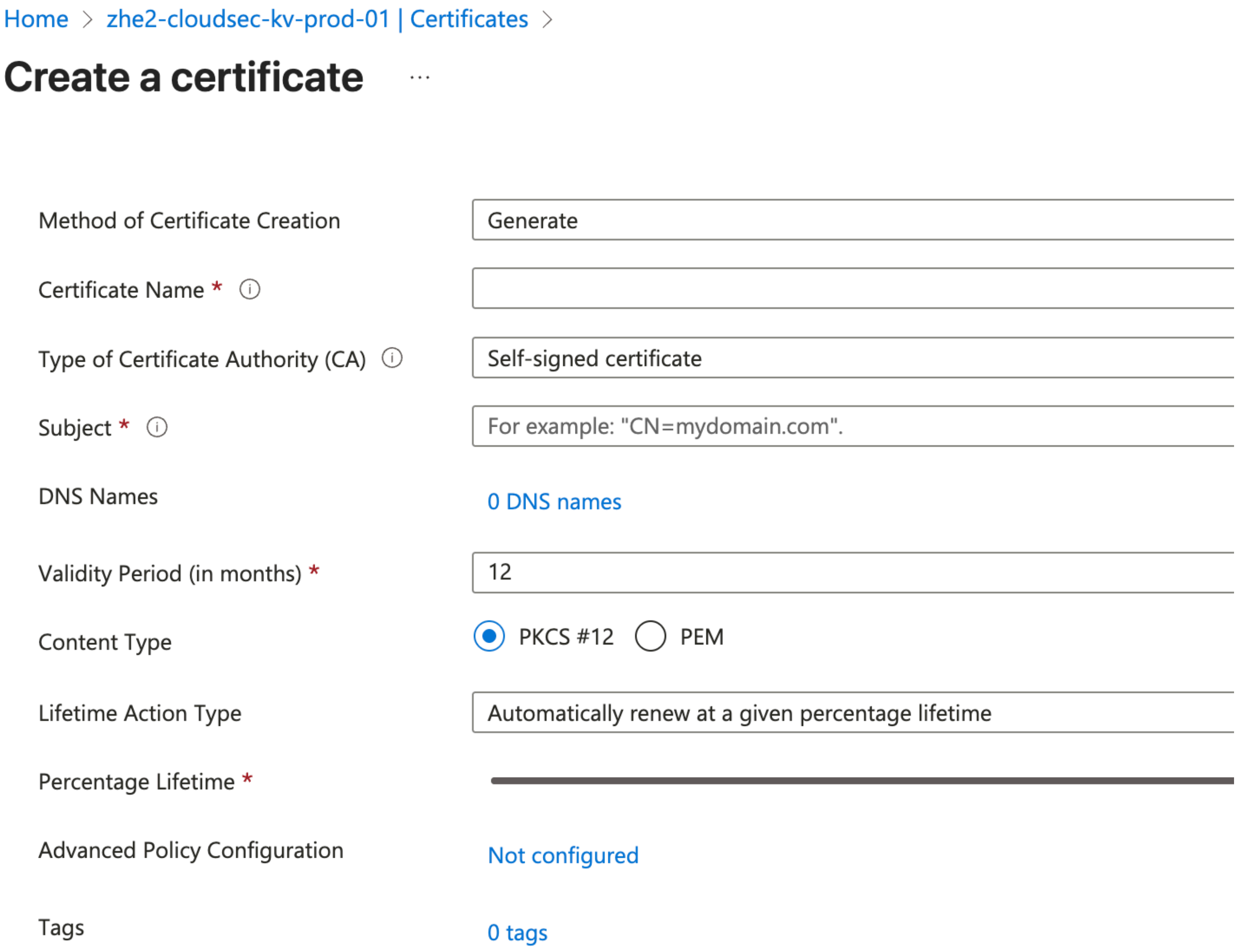


Name	Type	Status
Client-ID-[redacted]	API key [redacted]	✓ Enabled
Client-ID-[redacted]	API Key [redacted]	✓ Enabled
Client-ID-[redacted]	API Key [redacted]	✓ Enabled
Client-ID-[redacted]	AWS CS[redacted]	✓ Enabled

Process: Creating Certificates and secure storage in Key Vault

Step 1: Go to portal.azure.com
Step 2: Select designated key vault where you want to generate Certificate
Step 3: Under objects > certificate

Step 4: Enter certificate details and save



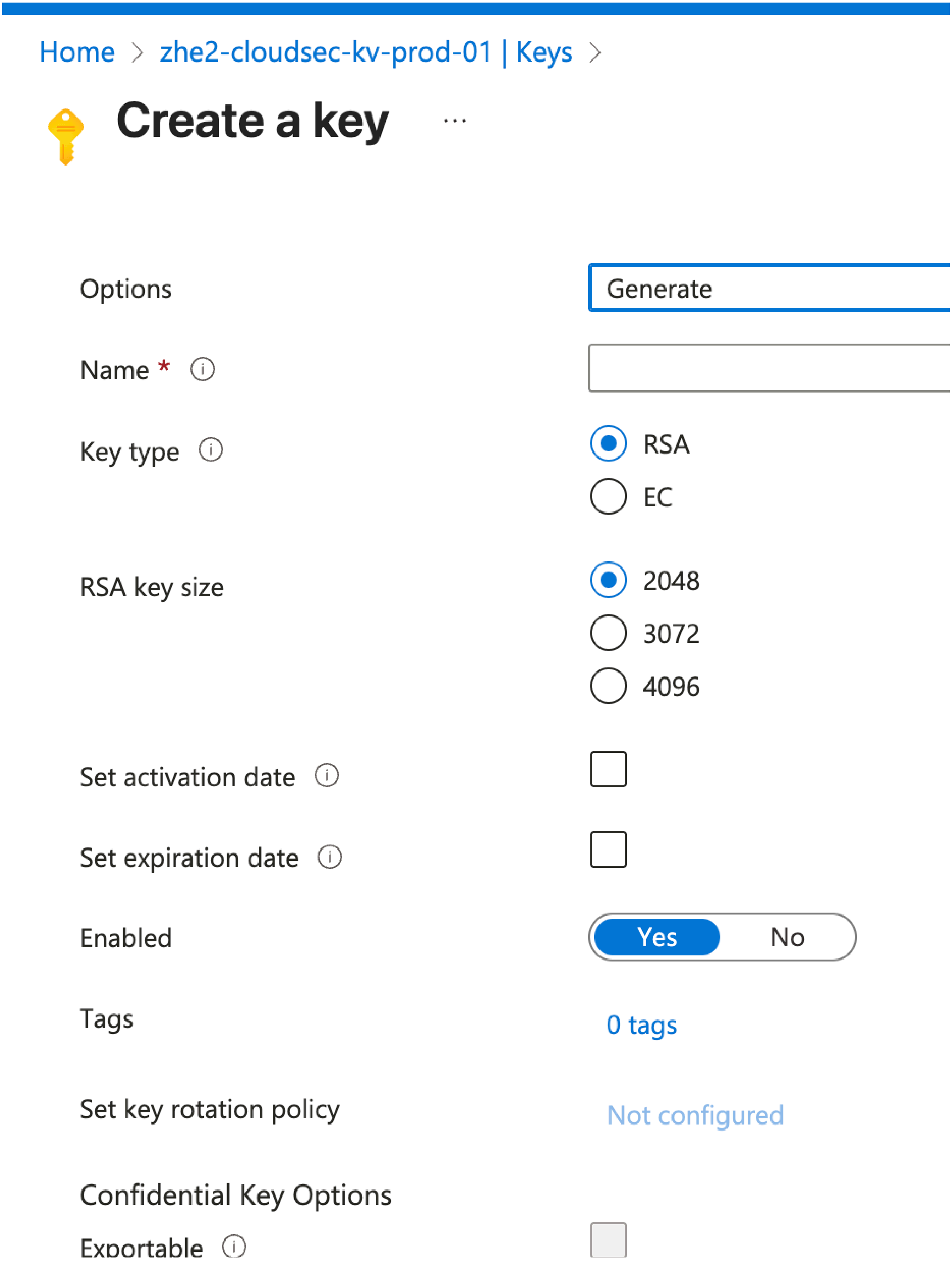
Home > zhe2-cloudsec-kv-prod-01 | Certificates >

Create a certificate ...

Method of Certificate Creation	Generate
Certificate Name * ⓘ	
Type of Certificate Authority (CA) ⓘ	Self-signed certificate
Subject * ⓘ	For example: "CN=mydomain.com".
DNS Names	0 DNS names
Validity Period (in months) *	12
Content Type	● PKCS #12 ○ PEM
Lifetime Action Type	Automatically renew at a given percentage lifetime
Percentage Lifetime *	
Advanced Policy Configuration	Not configured
Tags	0 tags

Process: Creating keys and securely storing the secret in Key Vault
Note: Keys created will need to be rotated on annual basis

Step 1: Go to portal.azure.com
Step 2: Select designated key vault where you want to generate Certificate
Step 3: Under objects > keys
Step 4: Enter key details like expiration date and activation date and save



Home > zhe2-cloudsec-kv-prod-01 | Keys >

Create a key

...

Options: Generate

Name * ⓘ

Key type ⓘ: ● RSA ○ EC

RSA key size: ● 2048 ○ 3072 ○ 4096

Set activation date ⓘ □

Set expiration date ⓘ □

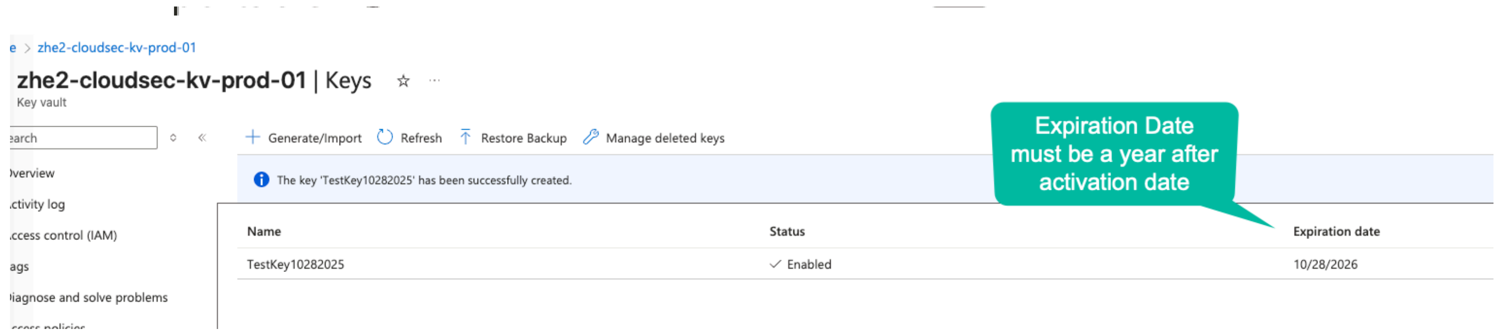
Enabled: [✓ Yes] No

Tags: 0 tags

Set key rotation policy: Not configured

Confidential Key Options

Exportable ⓘ □



e > zhe2-cloudsec-kv-prod-01

zhe2-cloudsec-kv-prod-01 | Keys ☆ ···

Key vault

The key 'TestKey10282025' has been successfully created.

Name	Status	Expiration date
TestKey10282025	✓ Enabled	10/28/2026

Expiration Date must be a year after activation date