Encryption as a Service with the Vault Transit Secrets Engine

Jacob Mammoliti

Consultant at Arctiq Inc.

Stop Storing Data in Plaintext!

Vault and Transit make it super easy to encrypt data.

About Me

Jacob Mammoliti
@ArctiqJacob

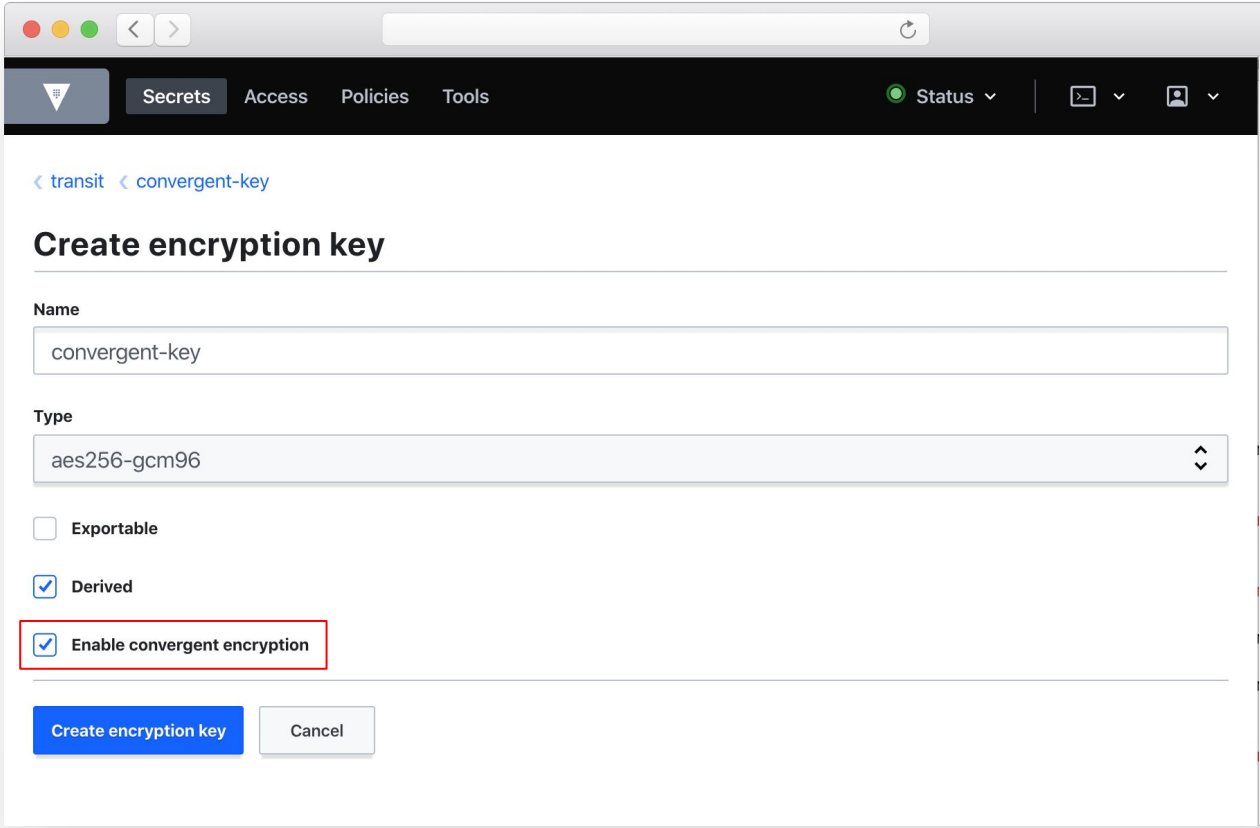
I’m a consultant at Arctiq Inc. out of Toronto, Canada. I focus on enabling customers with HashiCorp tooling, specifically Terraform, Vault and Consul. I also spend a lot of time in the microservices space working with all things Kubernetes.

Transit Secrets Engine Overview

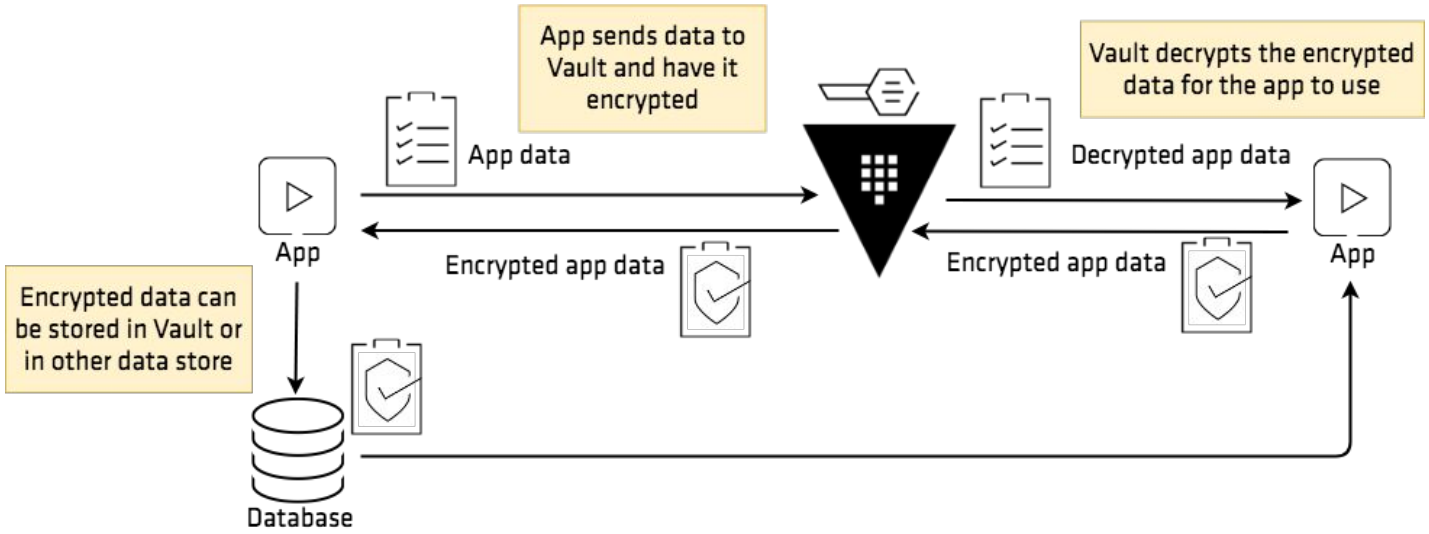
- Handles cryptographic functions on data in-transit
- Vault does not store the data it encrypts
- Primary use case for **Transit** is to encrypt data from applications while storing that data in some primary datastore
- Transit can sign and verify cryptographic signatures

Convergent Encryption

Use Transit to generate the same ciphertext based on plaintext and a context.



Transit Encryption Workflow



Enable Transit

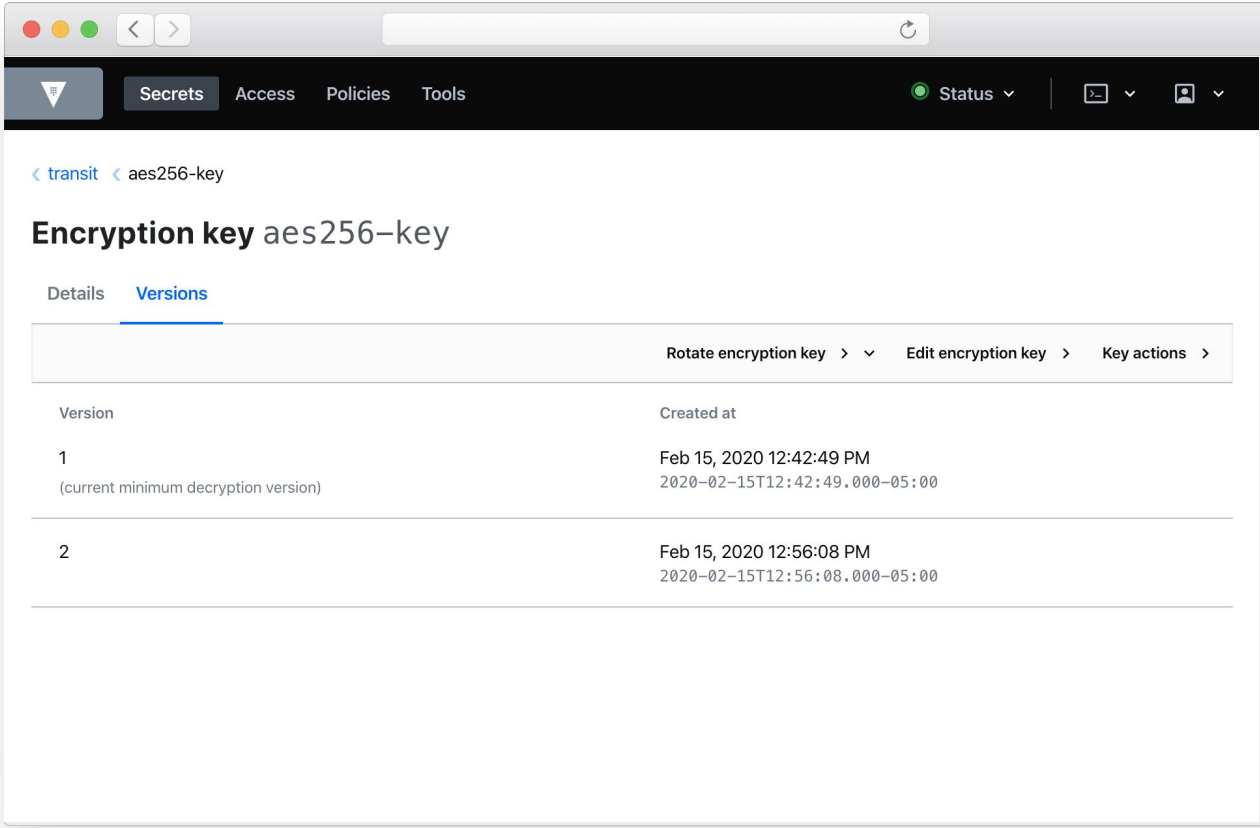
Enabling the Secrets Engine and setting up an encryption key is simple.

TERMINAL

```
$ vault secrets enable transit
Success! Enabled the transit secrets engine at:
transit/

$ vault write transit/keys/demo-key type=aes256-gcm96
Success! Data written to: transit/keys/demo-key


$ vault write transit/encrypt/demo-key \
  plaintext=$(base64 <<< "my secret data")

Key           Value
---           -----
ciphertext
vault:v1:zltup9jDYN6vf2Pqfdff3dfdf7dg2...
```

Transit in the UI

A look at the encryption key in the Vault UI.



Demo!