+++ title = "Hashicorp Vault" description = "Using Hashicorp Vault to encrypt database secrets" keywords = ["grafana", "Hashicorp Vault integration"] weight = 3 +++

Using Hashicorp Vault to encrypt database secrets

You can use an encryption key from Hashicorp Vault to encrypt secrets in the Grafana database.

Prerequisites:

- Permissions to manage Hashicorp Vault to enable secrets engines and issue tokens.
- Access to the Grafana [configuration]({{< relref "../../administration/configuration/#config-file-locations" >}}) file

1. Enable the transit secrets engine in Hashicorp Vault.
2. Create a named encryption key.
3. Create a periodic service token.
4. From within Grafana, turn on [envelope encryption]({{< relref "../../administration/database-encryption.md" >}}).
5. Add your Hashicorp Vault details to the Grafana configuration file; depending on your operating system, is usually named `grafana.ini`:

   a. Add a new section to the configuration file, with a name in the format of `[security.encryption.hashicorpvault.<KEY-NAME>]`, where `<KEY-NAME>` is any name that uniquely identifies this key among other provider keys.

   b. Fill in the section with the following values:

   - `token`: a periodic service token used to authenticate within Hashicorp Vault.
   - `url`: URL of the Hashicorp Vault server.
   - `transit_engine_path`: mount point of the transit engine.
   - `key_ring`: name of the encryption key.
   - `token_renewal_interval`: specifies how often to renew token; should be less than the `period` value of a periodic service token.

   An example of a Hashicorp Vault provider section in the `grafana.ini` file is as follows:

```
# Example of Hashicorp Vault provider setup
;[security.encryption.hashicorpvault.example-encryption-key]
# Token used to authenticate within Vault. We suggest to use periodic tokens:
more on token types https://www.vaultproject.io/docs/concepts/tokens#service-
tokens
;token =
# Location of the Hashicorp Vault server
;url = http://localhost:8200
# Mount point of the transit secret engine
;transit_engine_path = transit
# Key ring name
;key_ring = grafana-encryption-key
```

```
# Specifies how often to check if a token needs to be renewed, should be less
than a token's period value
token_renewal_interval = 5m
```

6. Update the `[security]` section of the `grafana.ini` configuration file with the new Encryption Provider key that you created:

```
[security]
# previous encryption key, used for legacy alerts, decrypting existing secrets
or used as default provider when external providers are not configured
secret_key = AaaaAaaa
# encryption provider key in the format <PROVIDER>.<KEY-NAME>
encryption_provider = hashicorpvault.example-encryption-key
# list of configured key providers, space separated
available_encryption_providers = hashicorpvault.example-encryption-key
```

> **Note:** The encryption key stored in the `secret_key` field is still used by Grafana's legacy alerting system to encrypt secrets. Do not change or remove that value.

7. Restart Grafana.

8. (Optional) From the command line and the root directory of Grafana Enterprise, re-encrypt all of the secrets within the Grafana database with the new key using the following command:

```
grafana-cli admin secrets-migration re-encrypt
```

If you do not re-encrypt existing secrets, then they will remain encrypted by the previous encryption key. Users will still be able to access them.

> **Note:** This process could take a few minutes to complete, depending on the number of secrets (such as data sources or alert notification channels) in your database. Users might experience errors while this process is running, and alert notifications might not be sent.

> **Note:** If you are updating this encryption key during the initial setup of Grafana before any data sources, alert notification channels, or dashboards have been created, then this step is not necessary because there are no secrets in Grafana to migrate.