Dynamic Database Credentials in Kubernetes

About me

Principal DevOps at Digitalis.io & AxonOps

- Born in the beautiful region of Galicia
- I have lived in Leeds for just over 20 years
- Where I worked as a contractor for more 15 years
- I've been with Digitalis more than 4 years and I'm still enjoying it
- I've worked in multiple positions, from *Networks Engineer* to *C programmer*
- But what I enjoy the most is **DevOps** and specially K8s
- I love to travel and I adore my Harley Davidson motorcycle

Digitalis.io and AxonOps

Digitalis.io

- Managed services and consultancy
- Focus on data platforms such as Apache Cassandra, Kafka, Elastic, etc
- Most customers are in the financial sector and social media
- AxonOps is the sister company

AxonOps

- Digitalis developed AxonOps as a tool for managing and monitoring Apache Cassandra internally
- The value created led us to market AxonOps as an standalone product
- We added last month support for provisioning clusters via the SaaS UI
- We're adding support for Kafka this year with Postgres and OpenSearch to follow soon

kubernetes

cassandra

Introduction

What is the problem?

Credentials improperly stored

- Passwords written down
- Shared across Email / Slack / Teams

Passwords reused

- It's very common to use the same passwords in different environments

Passwords never rotated

- It's too hard
- No one knows where the passwords are used
- It's considered too risky by many organizations

Compliance

- It is a **must** on all regulated industries
- It requires adherence to password policies and rotation

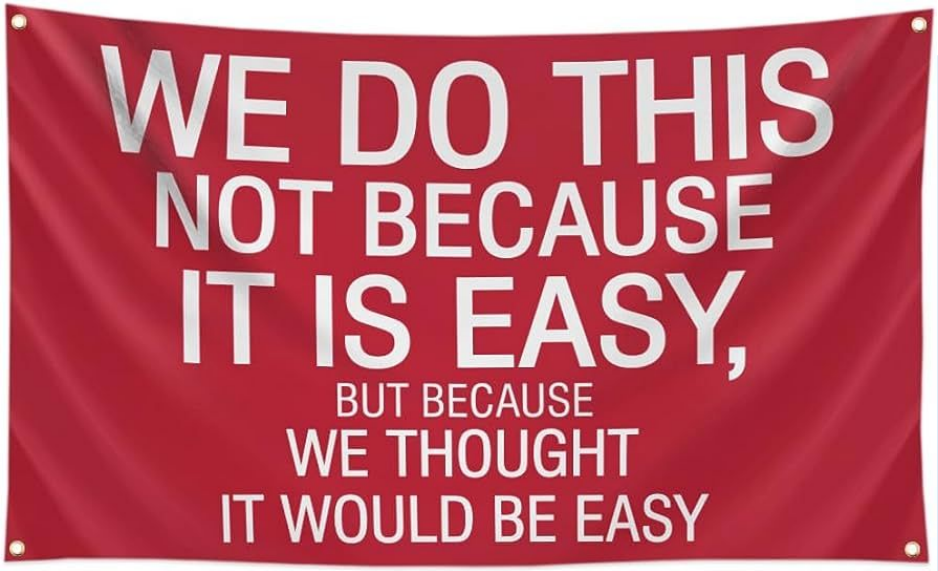
What should you be doing?

Don’t store password

Don’t reuse password

Expire and rotate passwords frequently

Hide password from the users if you can



kubernetes

cassandra

Hashicorp Vault

Why HashiCorp Vault

HashiCorp Vault supports multiple backends

- Cassandra
- Postgres, etc

Easy to set up

- Well documented and easy to configure

One time and long time passwords

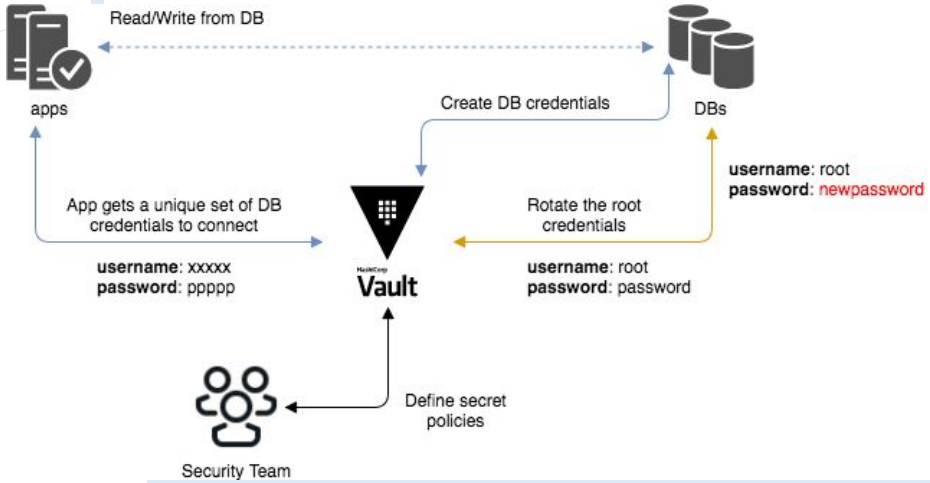
- Control access with TTL

Password policies

- Define password policies to align with the required security compliance

Access policies

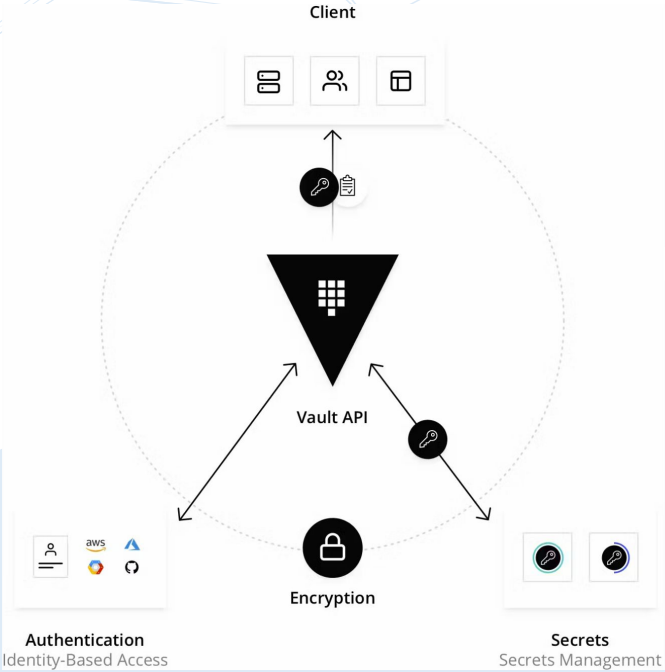
- Define different access levels per team
- Multiple authentication methods supported



Why HashiCorp Vault

Authentication

- You authentication against Vault, not the DB
  - LDAP
  - AD
  - OIDC
  - etc
- Request DB credentials and access level you're eligible
- Short TTL ensure password leakage is mitigated



Vault Configuration 1/2

```
resource "vault_database_secret_backend_connection" "default" {
  for_each          = toset(local.cluster_names)
  backend           = vault_mount.default[each.value].path
  name              = each.value
  allowed_roles     = ["admin", "cassandra-writer", "cassandra-query", "readonly", "vero"]
  verify_connection = false # This is required if the cassandra cluster is down

  cassandra {
    hosts            = var.cassandra_hosts[each.value]
    username         = data.vault_generic_secret.default.data["superuser"]
    password         = data.vault_generic_secret.default.data["superuser_passwd"]
    tls              = true
    insecure_tls     = true
    protocol_version = 4
  }
}
```

Vault Configuration 1/2

```
/* List of roles and permisisons */
resource "vault_database_secret_backend_role" "admin" {
  for_each    = toset(local.cluster_names)
  backend     = vault_mount.default[each.value].path
  name        = "admin"
  db_name     = vault_database_secret_backend_connection.default[each.value].name
  default_ttl = local.max_ttl
  max_ttl     = local.admin_ttl
  creation_statements = [
    "CREATE ROLE '{{username}}' WITH PASSWORD = '{{password}}' AND SUPERUSER=true AND LOGIN=true AND ACCESS TO ALL DATACENTERS;",
    "GRANT ALL ON ALL KEYSPACES TO '{{username}}';"
  ]
  revocation_statements = [
    "DROP ROLE IF EXISTS '{{username}}'",
  ]
}
```

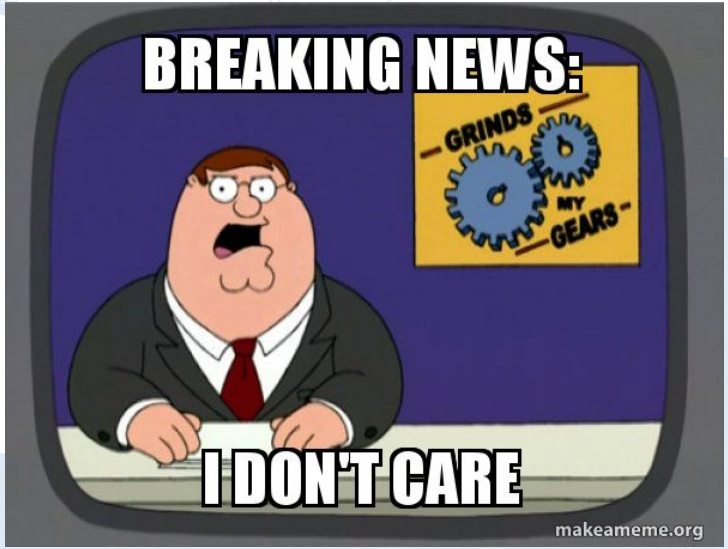
Usage

Yes! There is a password in the screenshot

```
ubuntu@mgmt-tkn-bastion-000:~$ vault read dev-exg/creds/admin
WARNING! The following warnings were returned from Vault:

  * TTL of "1080h" exceeded the effective max_ttl of "1h"; TTL value is capped
  accordingly

Key                Value
---                -----
lease_id           dev-exg/creds/admin/XmFuK4BpSEDR8mHfSVcB78P9
lease_duration     1h
lease_renewable    true
password           Ri-RPTYEknM3zYN6KfCY
username           v-root-admin-4J0n5YyiF3wig1Iq4mF4-1696422248
ubuntu@mgmt-tkn-bastion-000:~$
```



kubernetes

Kubernetes

Kubernetes integration using Vals-Operator

Automatic credentials creation

- The application requests the credentials
- The DevOps / Developer do not need to see it

Automatic password rotation

- Credentials have a defined **short** TTL
- Credentials are automatically renewed
- Pods can be restarted when credentials change

Vals-Operator

Digitalis' own secrets manager

- It can pull credentials from many different backends
- It also supports HashiCorp databases engines

```yaml
apiVersion: digitalis.io/v1
kind: ValsSecret
metadata:
  name: vals-secret-sample
  labels:
    owner: digitalis.io
spec:
  name: my-secret # Optional, default is the resource name
  ttl: 3600       # Optional, default is 0. The secret will be checked at every "reconcile period".
  type: Opaque    # Default type, others supported
  data:
    username:
      ref: ref+vault://secret/database/username
      encoding: text
    password:
      ref: ref+vault://secret/database/password
      encoding: text
```

Supported Backends

- Vault
- AWS SSM Parameter Store
- AWS Secrets Manager
- AWS S3
- GCP Secrets Manager
- Google Sheets
- Google GCS
- SOPS powered by sops
- Terraform (tfstate) powered by tfstate-lookup
- Echo
- File
- Azure Key Vault
- EnvSubst
- GitLab

Vals-Operator Databases

Database Secret

- It supports templating
- It can optionally perform a rolling restart

```yaml
apiVersion: digitalis.io/v1beta1
kind: DbSecret
metadata:
  name: cassandra
spec:
  renew: true # this is the default, otherwise a new credential will be generated every time
  vault:
    role: readonly
    mount: cass000
  template: # optional: change the secret format
    CASSANDRA_USERNAME: "{{ .username }}"
    CASSANDRA_PASSWORD: "{{ .password }}"
  rollout: # optional: run a `rollout` to make the pods use new credentials
    - kind: Deployment
      name: cassandra-client
    - kind: StatefulSet
      name: cassandra-client-other
```

kubernetes

Demo

Demo

Revoking a secret
https://youtu.be/lRANYtxPKqc

kubernetes

Thank you!

Thank you!

Sergio Rua
sergio.rua@digitalis.io

https://github.com/digiserg/YorkshireDevOpsDemo

https://www.linkedin.com/in/sergiorua/