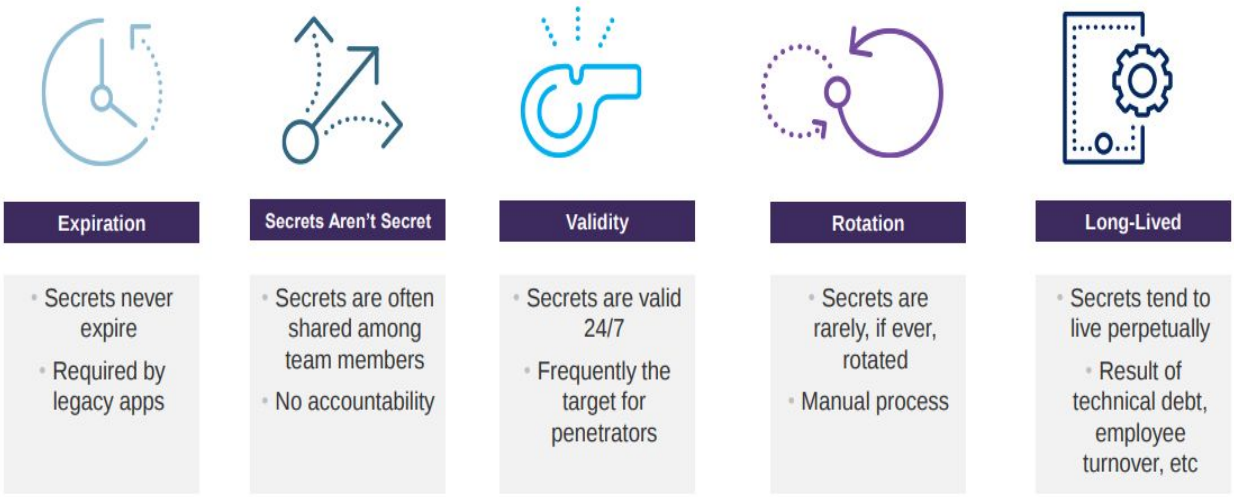
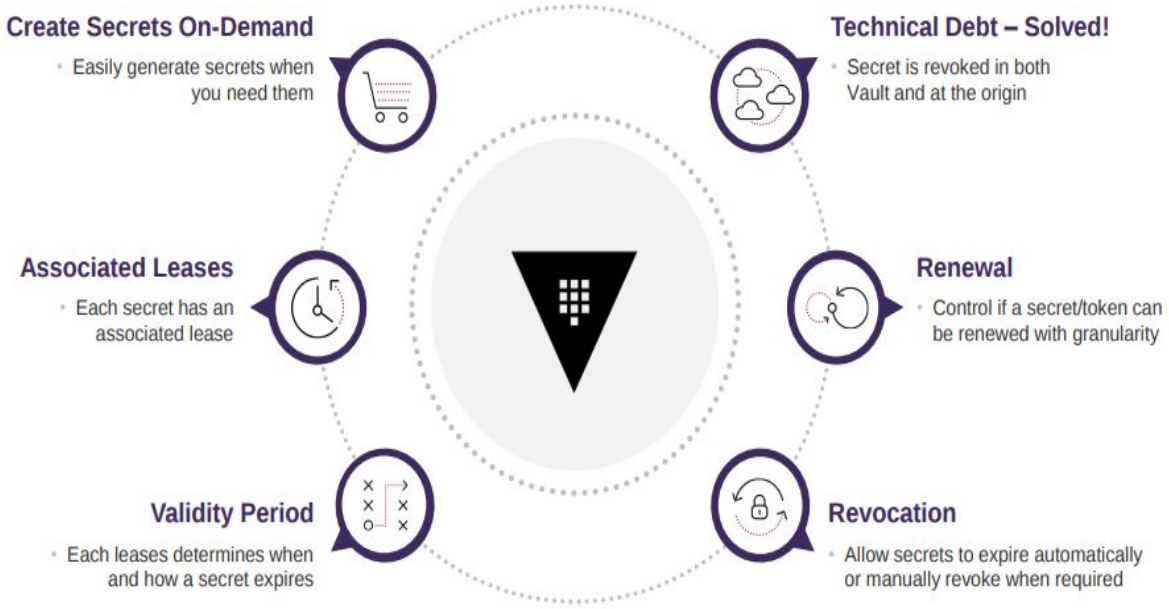
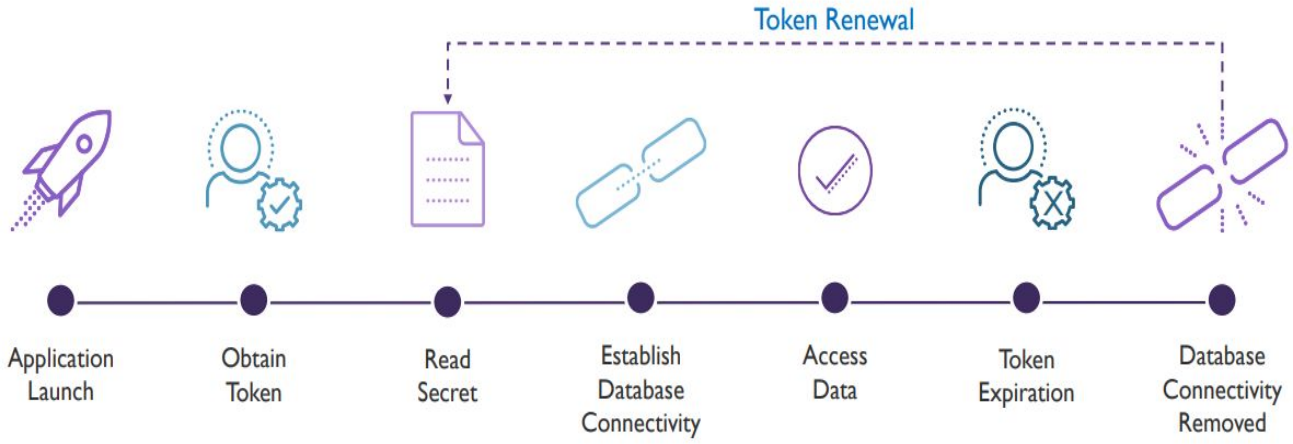
Secrets Engine



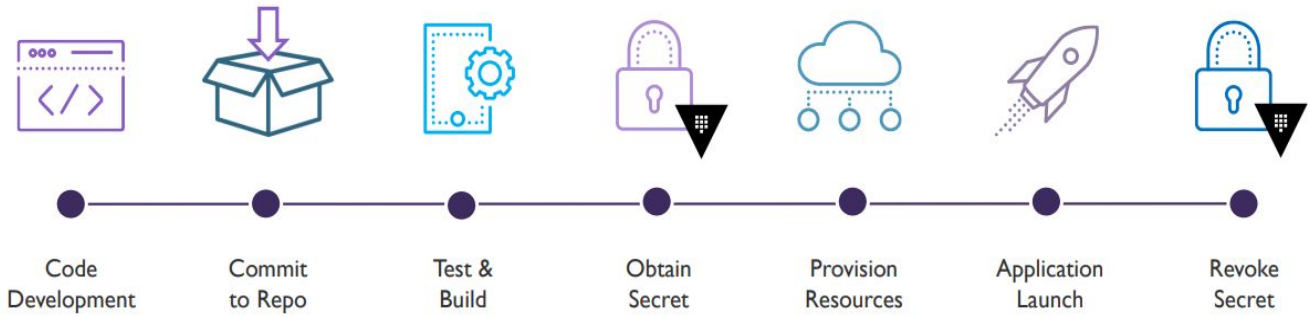
Why You Should Use Dynamic Secrets:



Application Using Vault:



Pipeline Using Vault:



Intro to Secrets Engines:

- Secrets Engines can store, generate, or encrypt data
- Many secrets engines can be enabled and used as needed
- Secret engines are enabled and isolated at a “path”
  $ vault secrets enable aws
- All interactions are done directly with the “path” itself.
  $ vault read aws/creds/aws_role

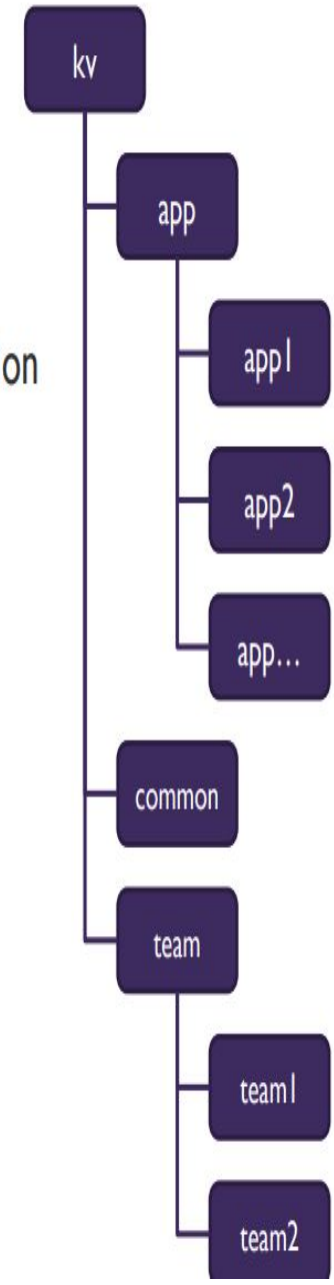
Secrets Engines in Vault:

Key/Value (KV) Secrets Engine:

- Allows you to store any information you'd like as a key & value
  - For example – secrets/webapp1/creds
    - user: skylines
    - password:skylines123!
- The most frequently used secrets engine in Vault
- Two versions available, named v1 and v2
  - KV v1 is the traditional version with standard features
  - KV v2 supports versioning

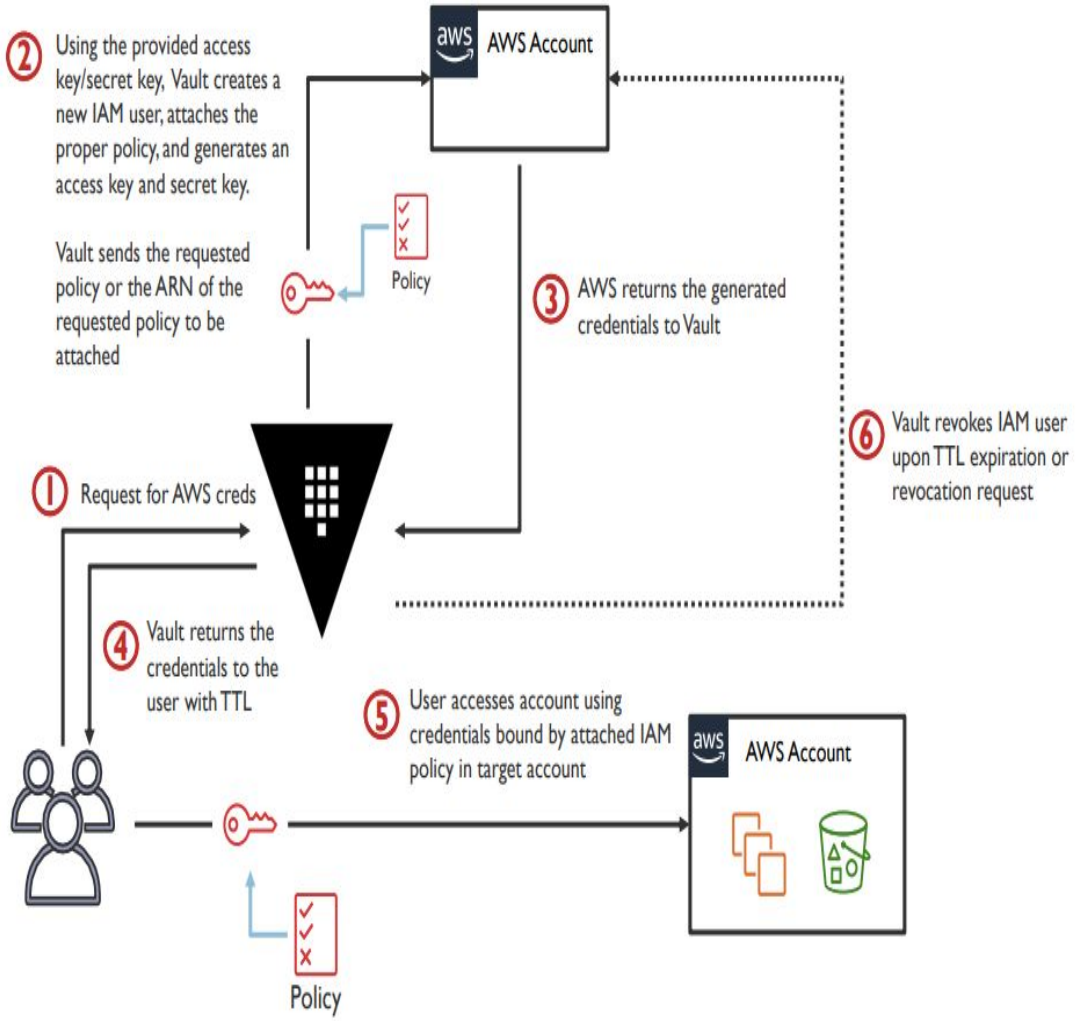
KV Structure

- Create a foundational structure
- Use parameters to simplify policies for administration
- Group by applications and teams
- Create additional mounts, if easier to manage
- Every KV structure will be different, although you should standardize between environments, where possible



AWS Secrets Engine:

- Dynamically generates AWS credentials
- Credentials still bound to a policy to permit/restrict actions



Transit Secret Engine: