**Hasicorp vault**

What is vault and its usage ?

HashiCorp Vault is an identity-based secrets and encryption management system. A secret is anything that you want to tightly control access to, such as API encryption keys, passwords, and certificates. Vault provides encryption services that are gated by authentication and authorization methods. Using Vault’s UI, CLI, or HTTP API, access to secrets and other sensitive data can be securely stored and managed, tightly controlled (restricted), and auditable.

**How to implement vault in springboot**

A computer screen with white text

Description automatically generated

Install hashicorp vault

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Default path for secret in vault is secret -> application which can be configured to different location in vault.

A screenshot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

How it is implemented in our application ?

Module app ->Vault Client ->Vault Server

Vault server info(host , post, token, scheme ) are made available to application through VM arguments

That will be passed to vault client.

We use application.yml or application.properties **for configuring the application context**.

When a Spring Boot application starts, it creates an application context that doesn't need to be explicitly configured – it's already autoconfigured. However, **Spring Boot offers different ways to override these properties**.

A close-up of a computer screen

Description automatically generated

We use bootstrap.yml or bootstrap.properties **for configuring the bootstrap context**. This way we keep the external configuration for bootstrap and main context nicely separated.

When the Spring Cloud application starts, it creates a *bootstrap* context. The first thing to remember is that the *bootstrap* context is the parent context for the main application.

In contrast to a Spring Boot application, a Spring Cloud application features a bootstrap context that is the parent of the application context. Although both of them share the same Environment, they have different conventions for locating the external configuration files.

The bootstrap context is searching for a bootstrap.properties or a bootstrap.yaml file, whereas the application context is searching for an application.properties or an application.yaml file.

Oauth2: a framework

A diagram of a client

Description automatically generated

JWT: a protocol to create a toke it’s a just part of oauth2

A diagram of a server

Description automatically generated