
HashiCorp
Vault

Introduction

- Name
- Total Experience
- Background – Development / Infrastructure / Management
- Experience on K8S, Cloud
- Your expectations from this training

Problem?

Email:

Please find your access to DB:

User: Alpha
Password: Alpha@123

connection.php

```
$host = 'localhost';
$user = 'Alpha';
$password = 'Alpha@123';
```

Overview

- Centrally managing secrets is an essential part of a well-built production grade system.
- Who is accessing what secrets is a challenging task, which rapidly becomes even more complex with the introduction of new applications, features, and services.
- HashiCorp Vault is an identity-based secrets and encryption management system that aims to address those issues by providing

Overview

- Any data that is deemed sensitive by your organisation has to be protected and though this data can be in different forms, have various purposes, and be found within multiple components of a modern system, the chances are that Vault will have a solution for it.
- Vault introduces the concept of Secrets Engines, which support all the common secret types, including but not limited to:
  - Username and password
  - API Keys
  - Encryption keys
  - Certificates

HOW DOES VAULT WORK?

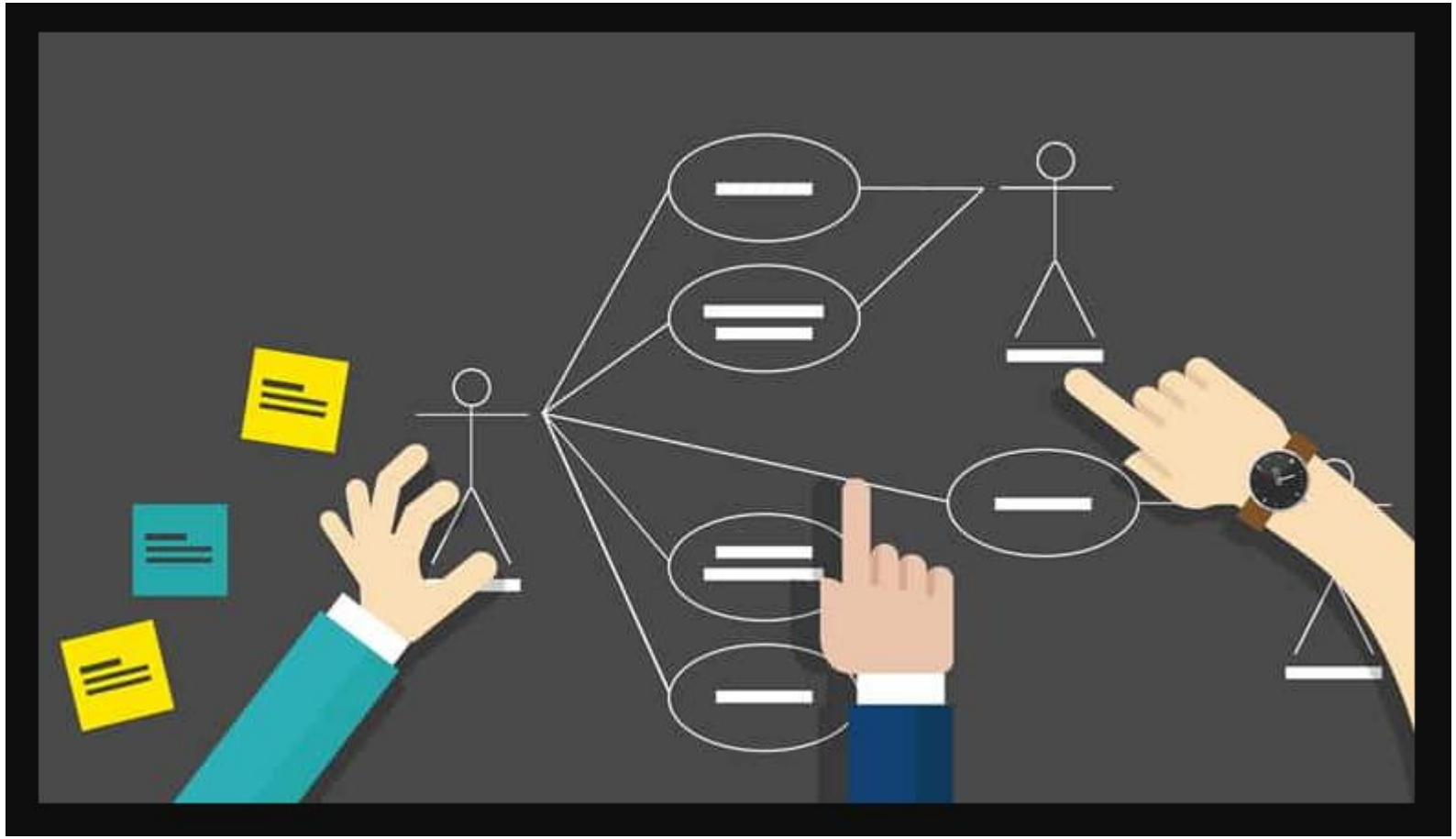
- Vault can be used to store sensitive values and simultaneously dynamically generate access for specific services/applications on a lease.
- It can also be used to authenticate users (machines or humans) to make sure they are authorised to access a particular file.


CLIENT
SECURITY TEAM
DEFINES POLICIES
Enforced at Runtime
TOKEN
AUTHENTICATION
Identity-Based Access/Backends
aws
Azure
GitHub
okta
LDAP
User + Password
TLS Certificate
Custom +
SECRETS
Secrets Management/Backends
aws
Custom +
TOTP
Key/Values
Databases
Transit

HOW DOES VAULT WORK?

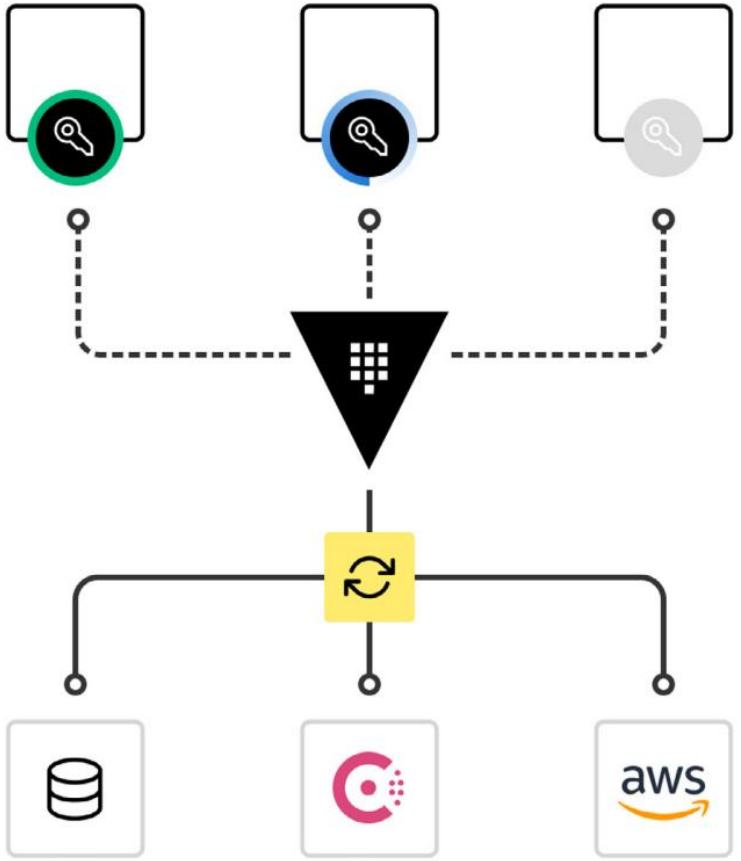
- Whenever a user or a machine needs to access a particular secret, they need to authenticate using a configured auth method.
- Depending on the auth method type, the authentication can happen via static or dynamic credentials.
- Once authenticated, policies attached to the user/machine are evaluated to determine whether they can access the secret or not.
- Finally, the secrets engine storing the secret issues a token, granting access to the requester.

USE CASES FOR VAULT



CENTRALISING STORAGE OF SECRETS

- Modern systems require a centralised, dynamically scalable solution for storing secrets.
- Vault centrally manages and enforces access to secrets and systems based on trusted sources of application and user identity.



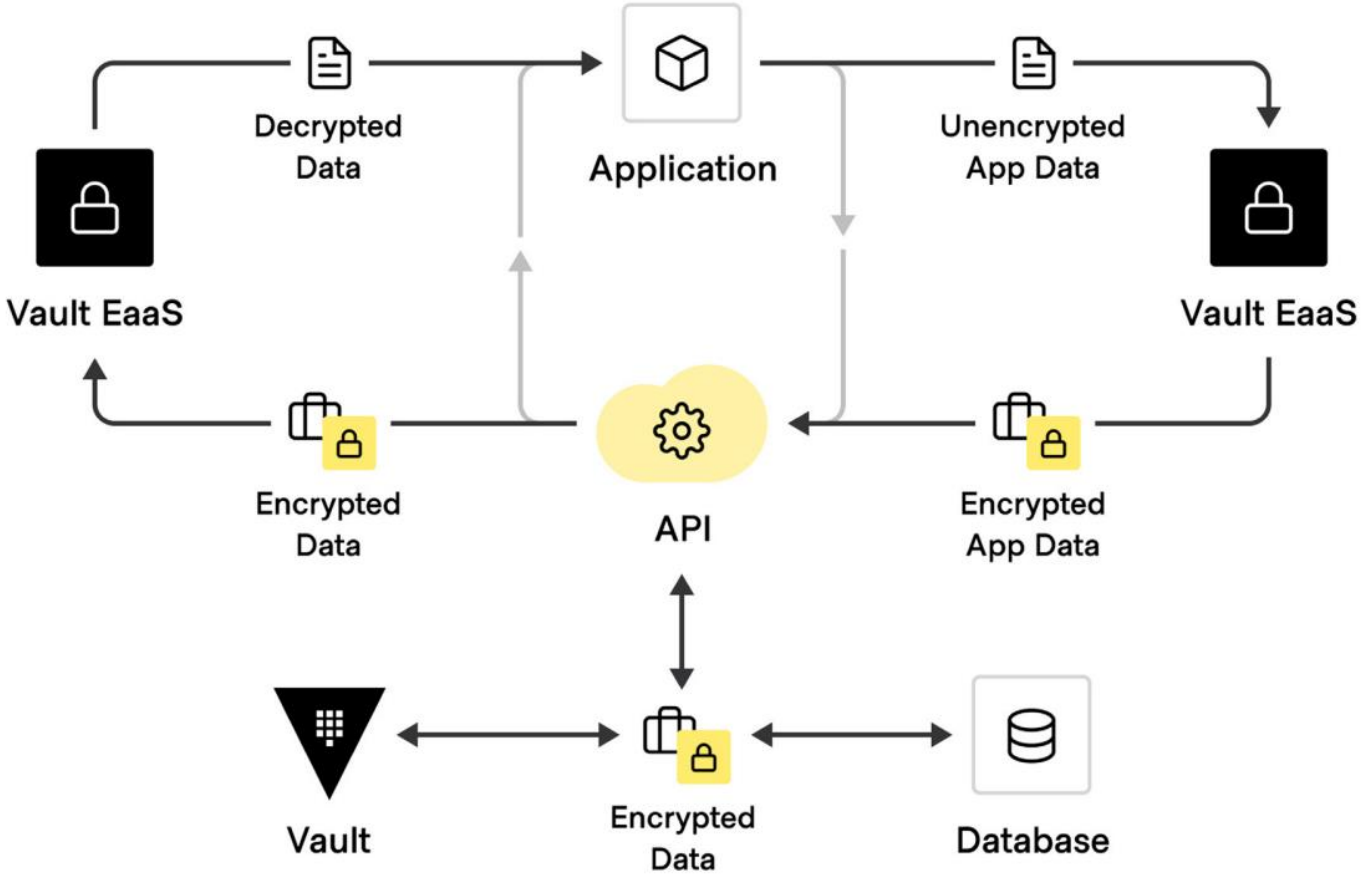
MIGRATING FROM STATIC TO DYNAMICALLY GENERATED SECRETS

- The ability to create and enforce dynamic credentials allows us to adhere to least privilege principles without human interaction.

Static Credential	Dynamic Credential
▪ Validate 24/7/365	▪ Short-lived
▪ Long-lived	▪ Follows principle of least privilege
▪ Manual password rotation	▪ Automatically revoked (based on lease)
▪ Frequently shared across teams	▪ Each system can have unique credentials
▪ Reused across systems	▪ Programmatically retrieved
▪ Susceptible to being added to VCS	▪ No human interaction needed
▪ Often highly privileged	
▪ Manually created	

DATA ENCRYPTION

According to best practices, it is recommended that all application data should be encrypted in transit and at rest. Vault provides encryption as a service with centralised key management to simplify encrypting data.



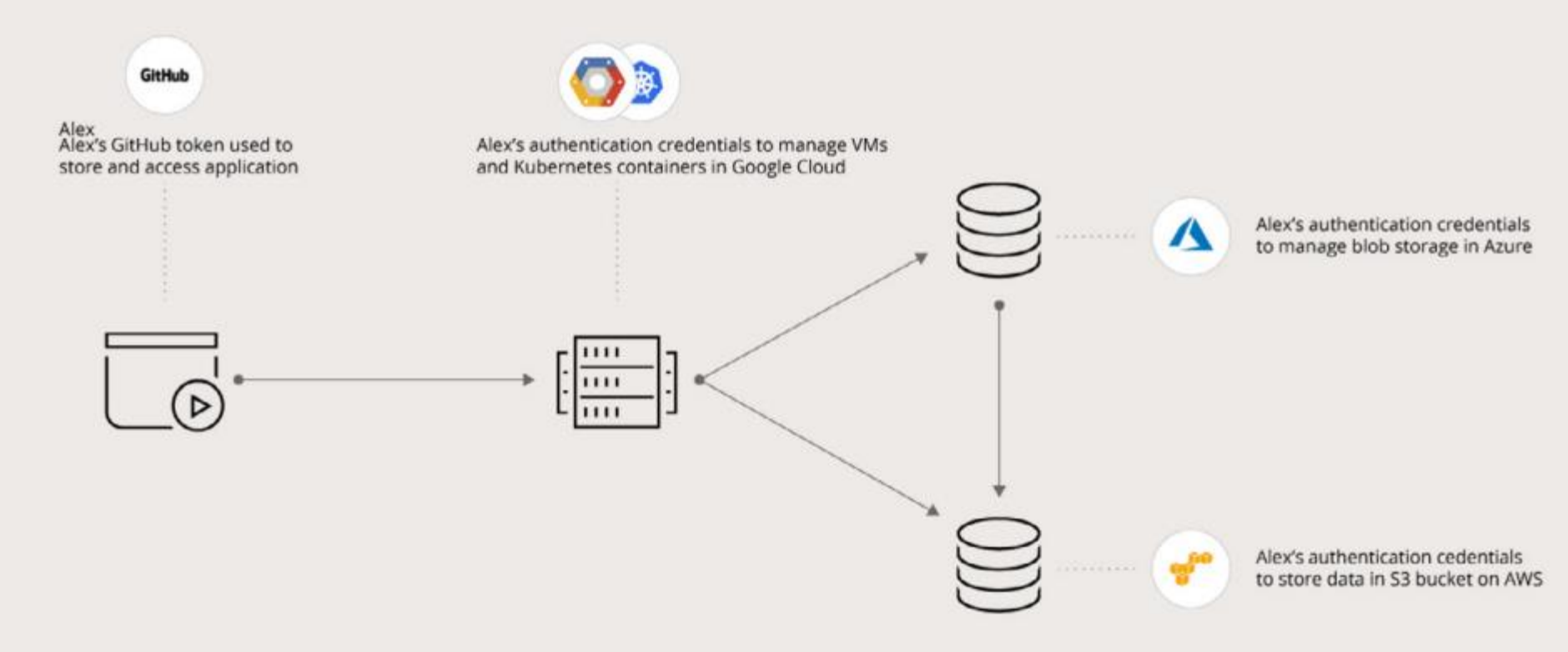
AUTOMATED X.509 CERTIFICATES MANAGEMENT

- Generating, signing and retrieving certificates is usually a lengthy process that may have many manual steps associated with it. Your organisation might be using a different solution to automate it but that automation might still be overcomplicated and hard to manage. Vault solves this problem with its PKI secrets engine.
- From a PKI engine perspective, Vault acts as a non-opinionated tool that focuses on automating PKI Operations and the process around fetching and renewing certificates in an automated fashion, Vault can act as a Root CA or as an intermediate (requires CSR signing from the root CA).

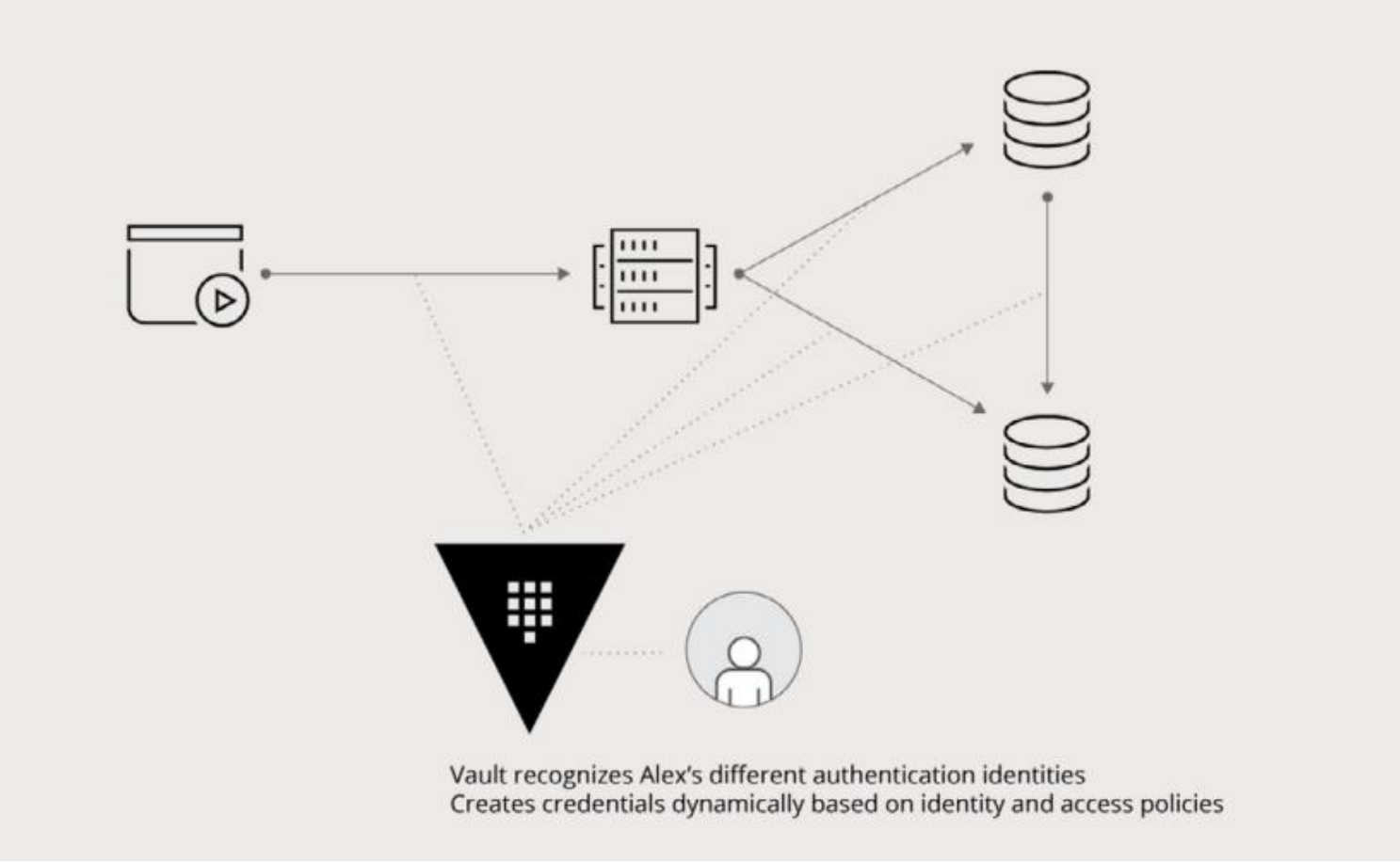
MIGRATING TO IDENTITY-BASED ACCESS

- With Identity and Access Management (IAM) in Vault, users and organisations can map global access and policies to Vault Identity Entities and Groups.
- For example, let’s say Alex is running an application in a Kubernetes container on Google Cloud that gets deployed from GitHub, it connects to a database in Azure, and replicates to AWS.
- Alex can integrate his different cloud identities into an “Alex” entity and centrally create policies granting and restricting access to secrets for the different components to connect securely to one another, all within one workflow and API.

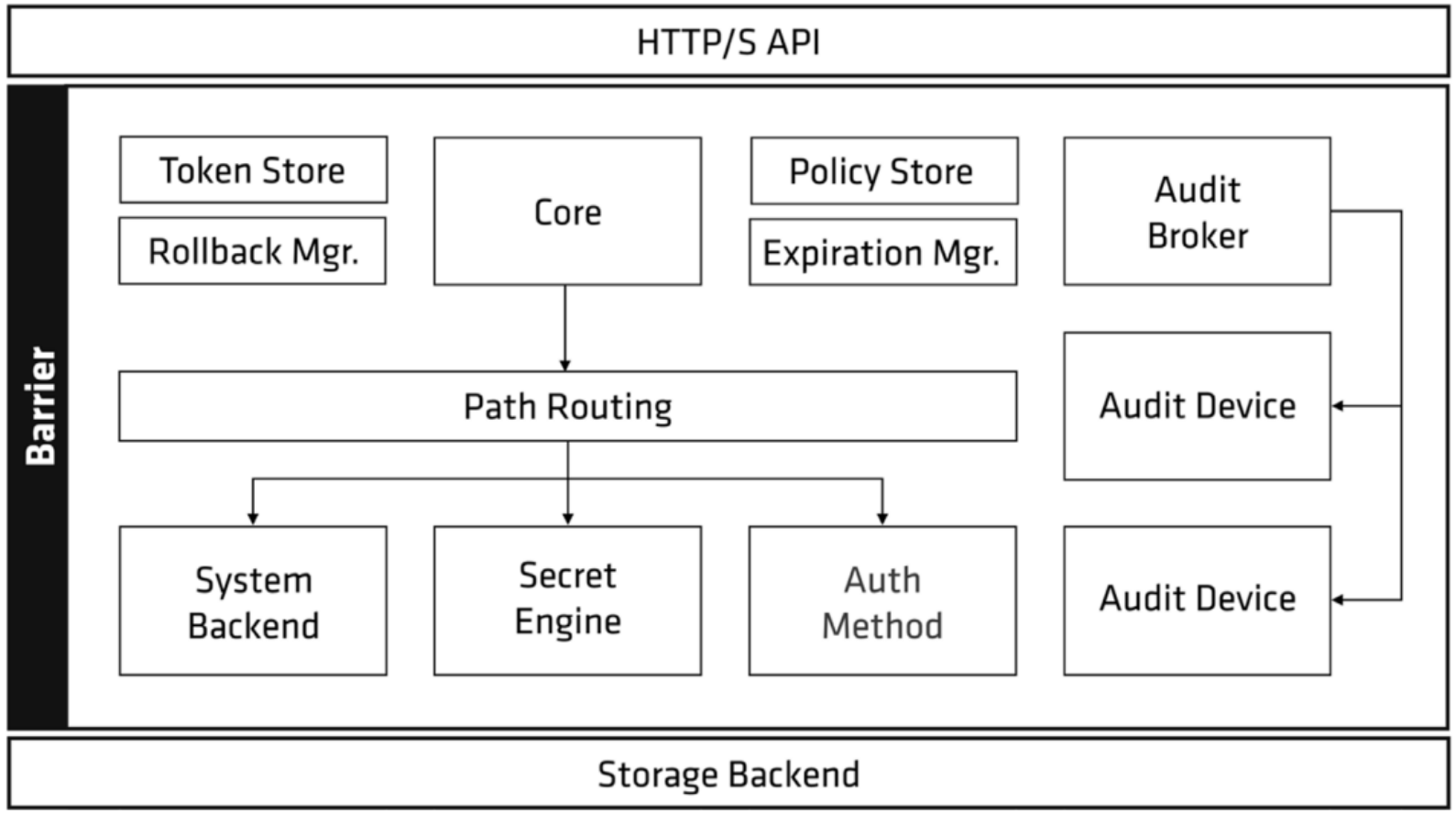
Before Vault



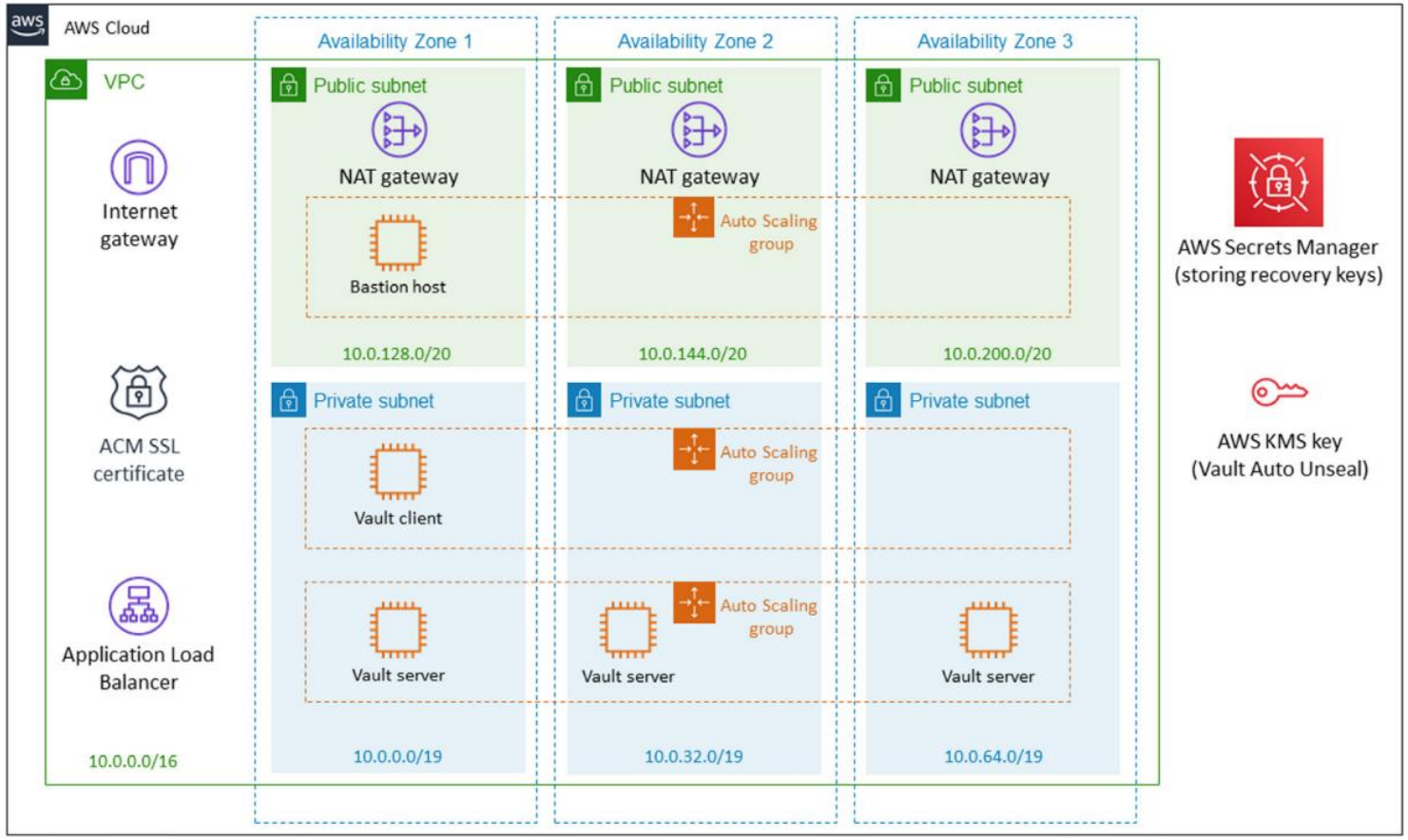
After Vault


Vault recognizes Alex's different authentication identities
Creates credentials dynamically based on identity and access policies

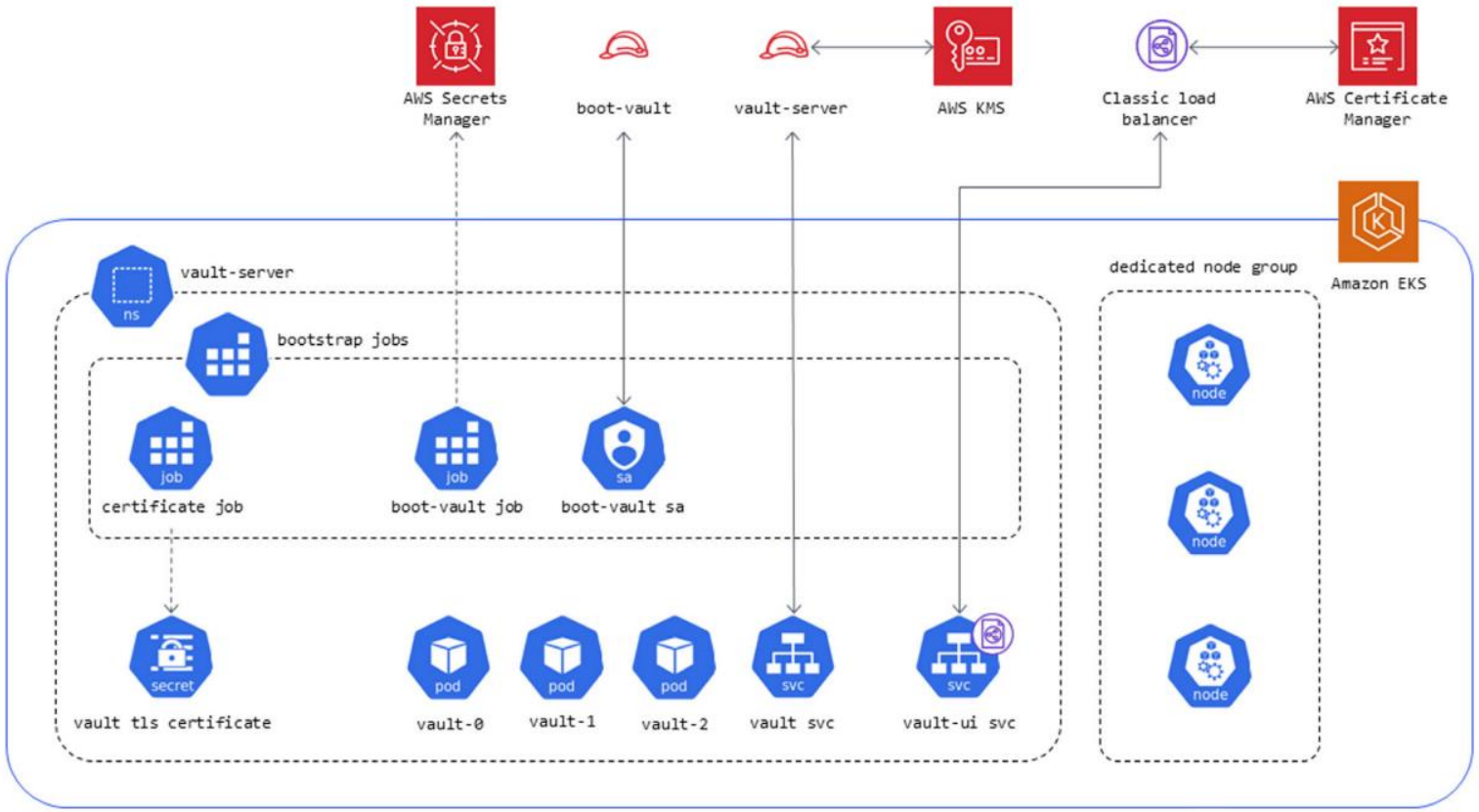
ARCHITECTURE AND DESIGN



Vault on AWS EC2



Vault on AWS EKS



Install Vault

Download Vault:

```
wget https://releases.hashicorp.com/vault/1.5.0/vault_1.5.0_linux_amd64.zip
```

Unpack Vault:

```
unzip vault_1.5.0_linux_amd64.zip
```

Move the Vault binary to the bin directory:

```
sudo mv vault /usr/bin
```

Labs and Activities

- https://github.com/Jaibw/hashicorp-vault


Q&A

THANK YOU