C H A P T E R 3

**Setting Up Hashicorp Vault**

This chapter of the book will provide intro of vault, architecture and hands on steps to setup open source vault:

• Introduction to Vault opensource

• Understanding Vault Architecture

• Using Vault Security Model

• Installing Vault Cli

# Introduction to Vault:

Vault is basically use for store and access secrets in a secure manner. We can access the secrets using API keys and password. At that time, it is very important to secure our credentials. Secrets are defined as any form of sensitive credentials that need to be controlled and can be used to unlock sensitive information. Secrets could be store in the type of password, API keys, SSH Keys. Vault stores secrets for authentication and authorization.  
  
A modern system requires tightly access to a secrets, Hashicorp vault makes it very easy to manage and access by API and Interface. We can monitor detailed logs and fetch the details about who accesses what.   
  
Authentication can either be via password or using dynamic values to generate temporary tokens that allows you to access particular path. Policies written using (HCL) Hashicorp Configuration Language are used to determine who gets what access.

Following are the key feature of Vault:

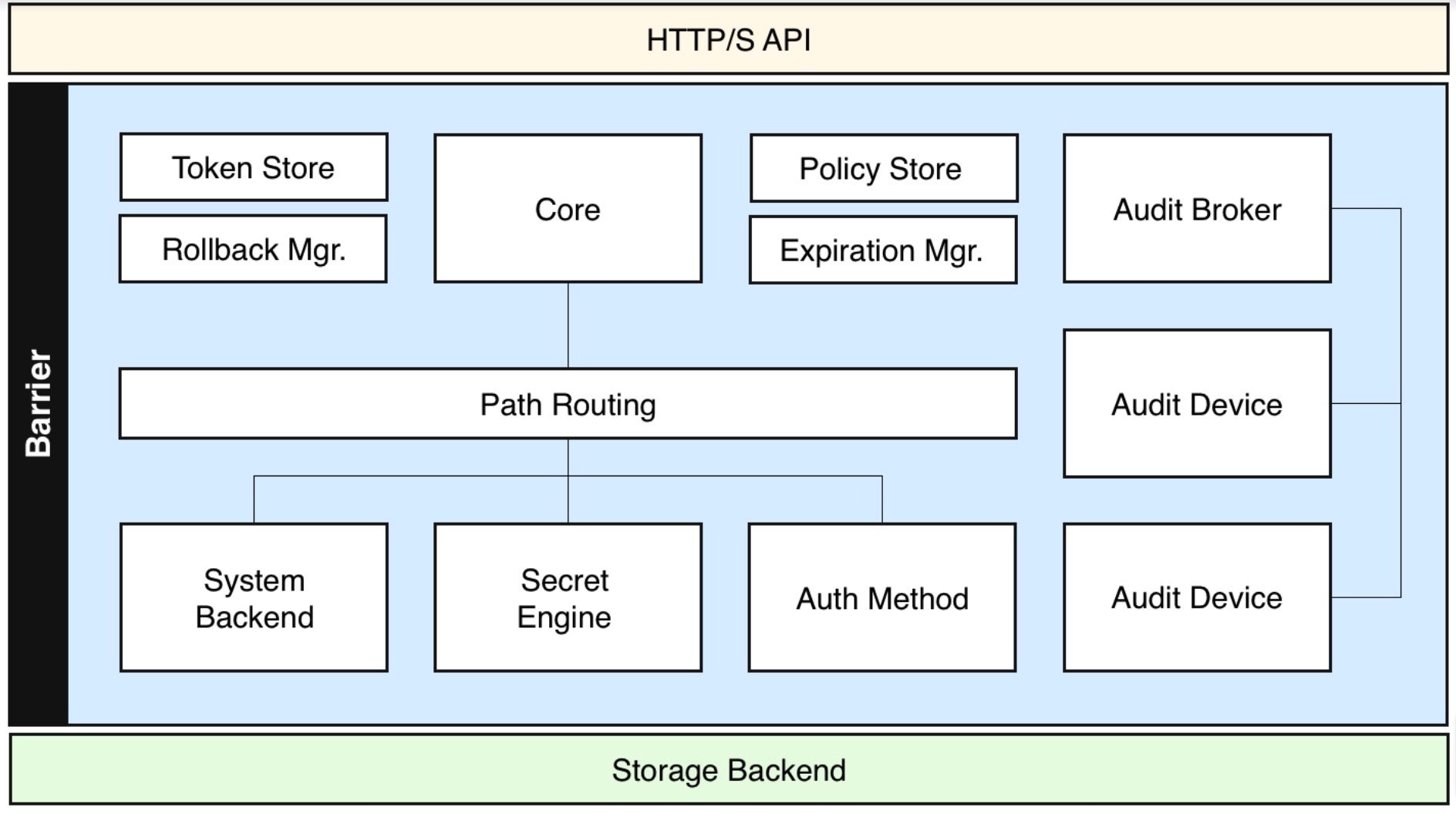
1) **Data Encryption:** Vault can easily encrypt and decrypt the credentials. Vault can`t store the data when encrypt and decrypt the credentials. Vault allows security to define encryption parameters and developer store encrypted data in a location SQL without having to design their own encryption methods.

2) **Revocation:** Vault will have feature of revoke the credentials after some time (768 hours). If we want vault revoke the credentials early for that we can set the policy on the basis of time period. Revocation assists in key rolling as well as locking down systems in the case of intrusion.  
The leasing method of dynamic credentials means Vault Knows which secrets each client has. This makes it possible to revoke specific leases in the case of any attacked/hacked.  
  
3) **On demand secrets:** Vault can generate on-demand secrets for some methods. i.e: AWS and SQL database. Vault can handle the problems of “Dynamic Secrets”. A dynamic secret is generated on demand and secret id is unique to a end user. Dynamic secrets are more secure instead of static secrets, which is defined ahead of time and shared. Vault revoke the access when the lease expires.

4) **Renewal:** Vault has renewal feature, As per revocation feature vault will revoke the credentials and after the revocation end user renew the secrets through the renew API.  
By attaching a lease to secrets vault have information about maximum time to live and after that secrets are being automatically rotated. This feature improve security.

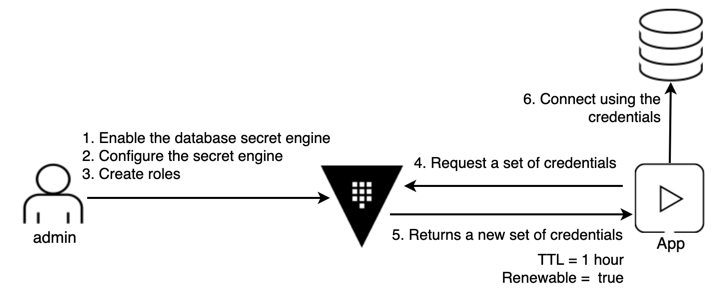
5) **Secret management:** Secret management is one of the main feature of Vault. Hashicorp vault can be used to store any type of credentials, including sensitive environment variable, API keys, database etc. Using vault allows you to take full control of any sensitive secrets.

C H A P T E R 3

 Figure 3.1 Vault Functional architecture

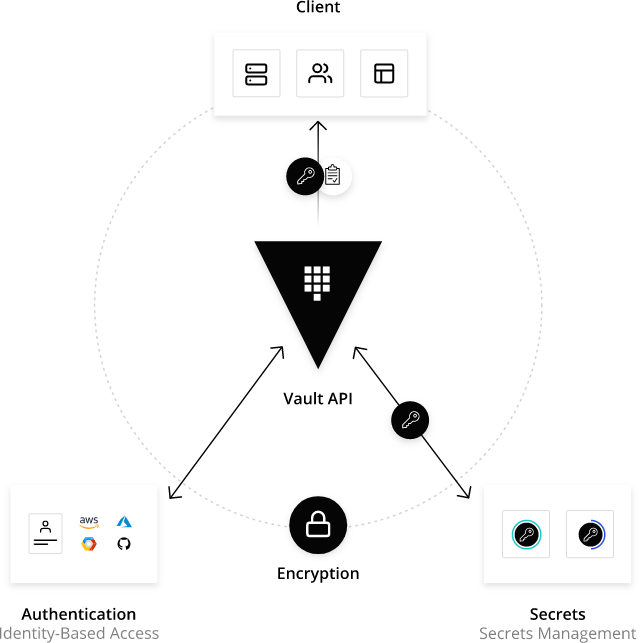
Storage Backend: Storage backend used for store encrypted data/secrets.   
  
Barrier: Barrier is basically use for all data flows between vault and backend storage through the barrier.  
  
Client Token: Client token is nothing, it is used for vault to verify the identity of the end user.  
  
Secret: Secret is anything that you want to control access. i.e: password of database etc   
  
Auth Method: Auth method is used for authenticate application and user are connecting to vault.

C H A P T E R 3

 Figure 3.2 Application & vault architecture

Vault basically use for management of secrets; we can secure our credentials with vault.  
It can be used to access sensitive data and at the same time generate access for any application on lease.

# Vault Security Model:

Vault manages sensitive data in a secure manner. The vault provides very high security in relation to data storage. Vault Security model is to provide security information, authentication, availability, integrity to secure sensitive data.  
In that case if the end user wants to access the data from the vault, then the client needs the correct authentication and authorization to access the data or modify various policies.  
  
    
   
 Figure 3.3 Manage Secrets and Protect Sensitive Data  
  
  
**Threat Model:**  
  
Vault Communication or Eavesdropping. End user and application communication should be secure. Eavesdropping is the act of secretly or stealthily listening to the private conversation or communications of others without their consent.  
  
Tampering with data at rest or in transit. If someone tapped the data or sensitive, any tampering should be detected and the vault should cancel the transaction process.  
  
  
Access to data or control without authentication or authorization. Applications and end users access data from the storage of the vault, then all requests must be pursued by applicable security policies.

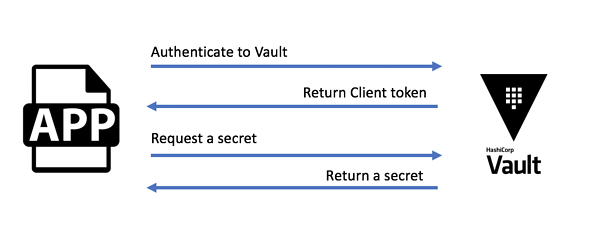
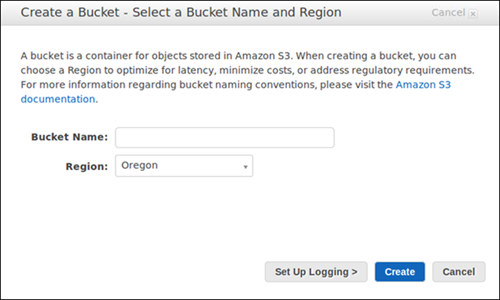
Privacy of stored secrets. Any data that leaves the vault to rest in the storage backend must be protected from eavesdropping. In practice, all data and secrets must be encrypted.  
  
Secret and data availability in case of failure. The vault supports running in highly available configurations to avoid the unavailability of secrets.  
  
  
  
**External Threat:**  
  
Looking at the architecture of the vault, there are 3 different systems that we are concerned with for the vault. There is a client, speaking for a vault on an API. The vault or server is more accurate, providing APIs and serving requests. Finally, storage is the backend, which the server is using to read and write data.  
  
There is no mutual trust between the vault client and the server. Clients use TLS to verify the server's identity and establish a secure communication channel. The server requires that the client provide a client token for each request, which is used to identify the client. A client that does not provide its own token is only allowed to make login requests.   
Depending on the backend used, the vault can communicate with the backend over TLS to provide an additional layer of security.  
  
  
**Internal Threat:**  
  
End users worry that attackers hack the system's data. Within the vault system, a key security concern is an attacker's attempt to gain access to secret material that they are not authorized.   
If someone or attackers have any level of access to the vault and is able to be authenticate to the vault system, **it is an internal threat**.  
  
When a client first authenticates with the vault, a auth method is used to verify the client's identity and return a list of associated ACL policies. This association is prematurely configured by the operators of the vault.  
For example if an application wants to communicate with vault sensitive data, the vault creates a token that is randomly generated for authentication.  


Figure 3.4 App integrate with Vault Data

Application sends a token on each request for communication to the vault. Vaults checks the validity of the token for communication and generates an ACL based on the associated policies. On the basis of ACL policy rule, application perform the number of action.

## Installing Vault CLI and Vault Integrate with S3 and KMS keys

Hashicorp officially maintains the package for the following distribution:  
Ubuntu/Debian  
Centos/Rhel  
Fedora  
Amazon Linux  
  
**Pre-requisites for vault integrate with S3 and KMS keys:**  
**-** S3 Bucket  
- KMS keys with IAM user   
  
 **S3 bucket:** You need to create an Amazon S3 bucket where you will store your objects.  
  
1. Sign into the AWS Management Console  
2. Under Storage and content Delivery, choose S3 to open the Amazon S3 console.  
3. From the Amazon S3 console dashboard, Choose Create Bucket and Create empty S3 bucket for Vault storage.  
   
   
   
 Figure 3.5 Create S3 Bucket   
  
  
**KMS KEYS:** We assuming you have KMS keys with IAM user permission to integrate with Vault setup.

Let’s start setup on Centos machine with following steps:

Install yum-utils on server, yum utils use for manage repositories

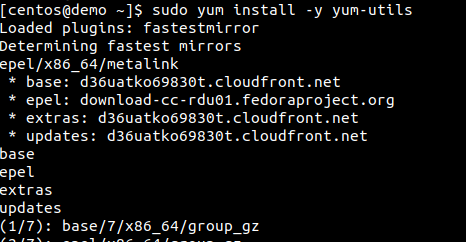


Figure 3.6 Install yum-utils

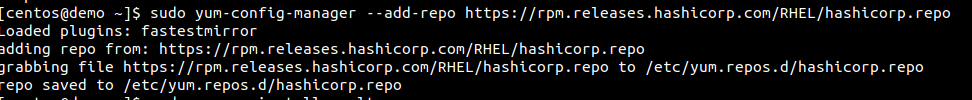
Add hashicorp repository to install vault using yum-config manager  
  


Figure 3.7 Add Hashicorp repository

Install vault on the server using yum install command

## C:\Users\anita.ty\Pictures\vault3.png

Figure 3.8 Install Vault

Verifying the Vault installation using vault command:

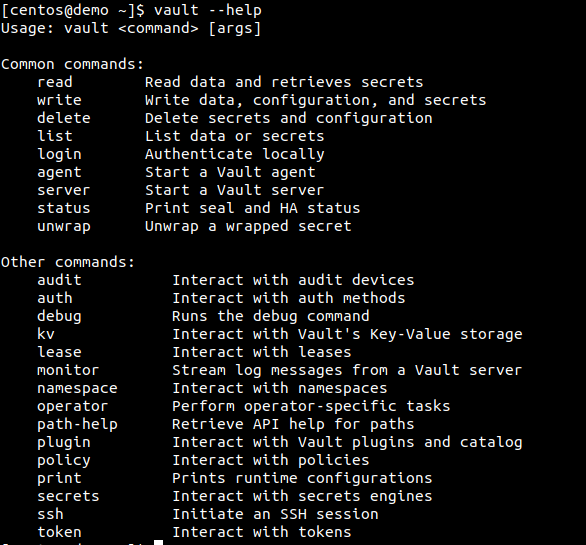
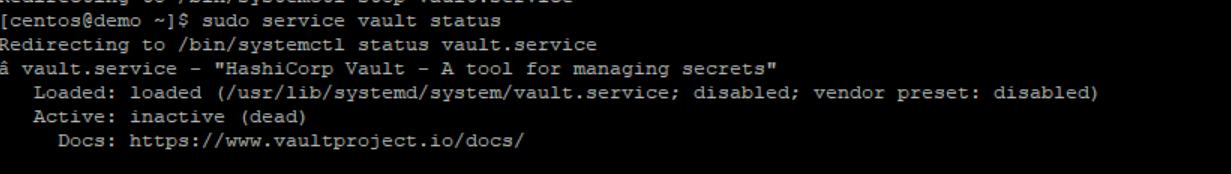
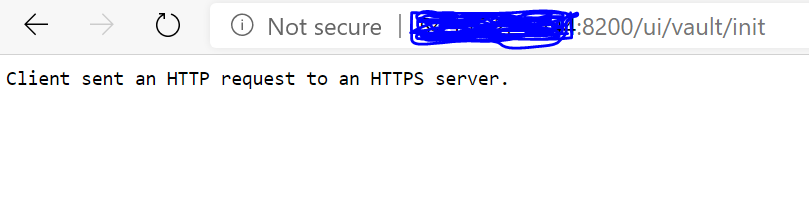
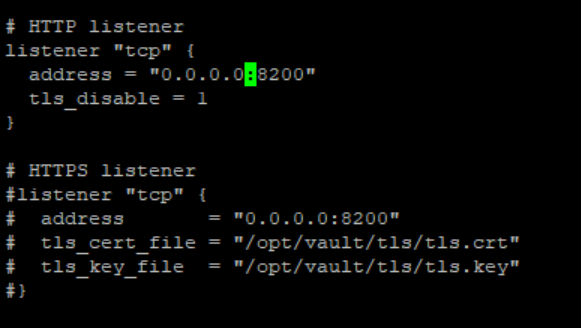


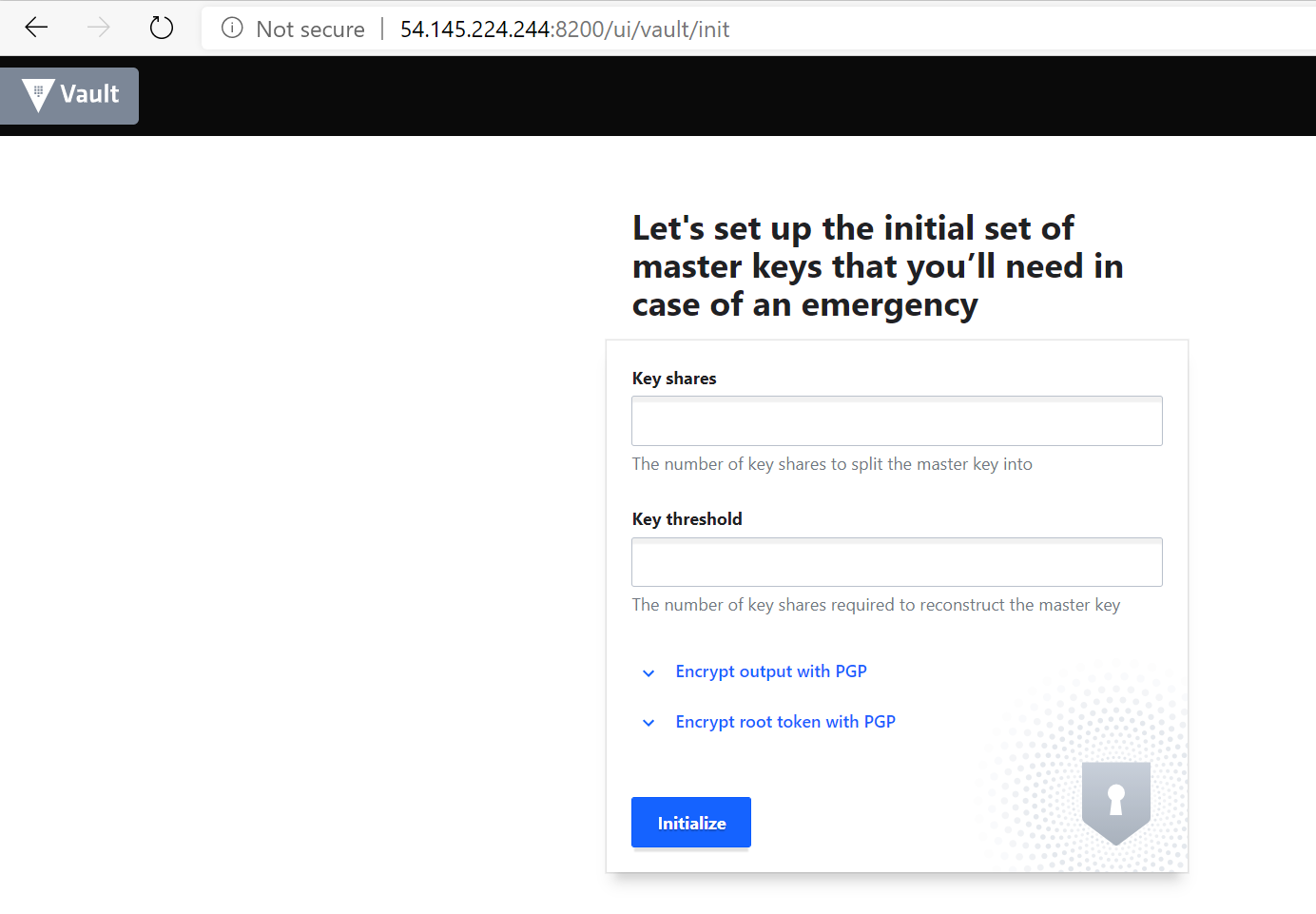
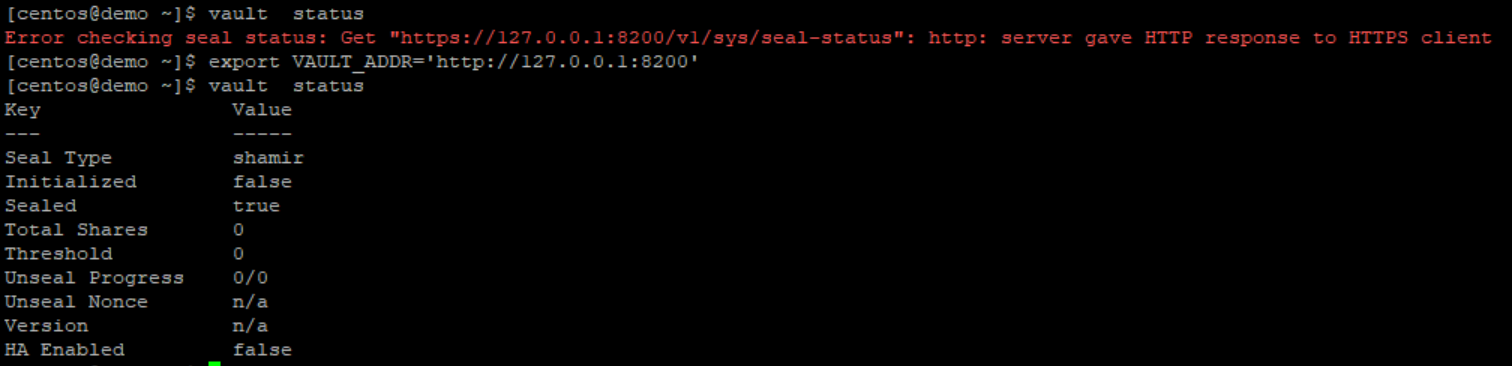
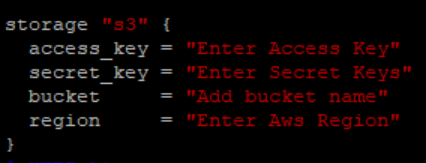
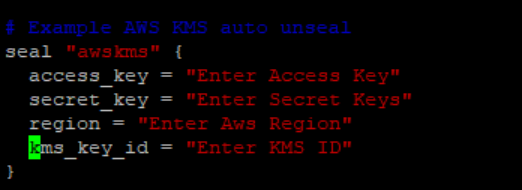
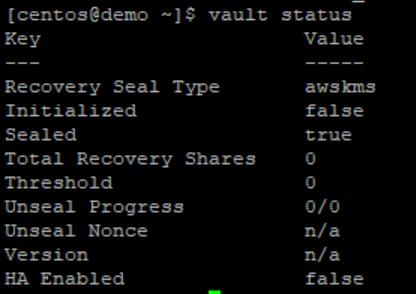
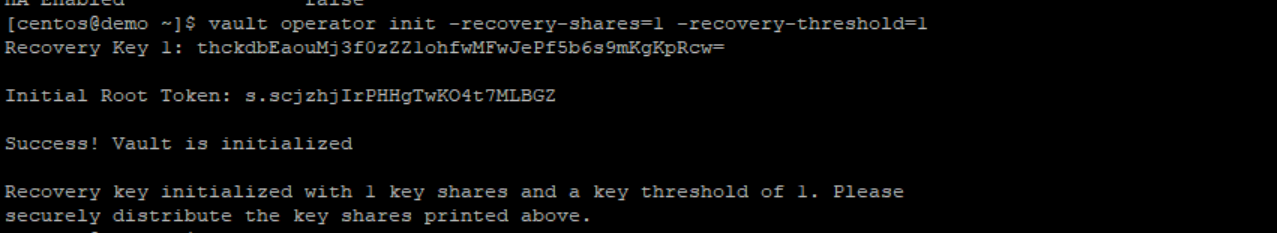
Figure 3.9 Verify Vault Installation

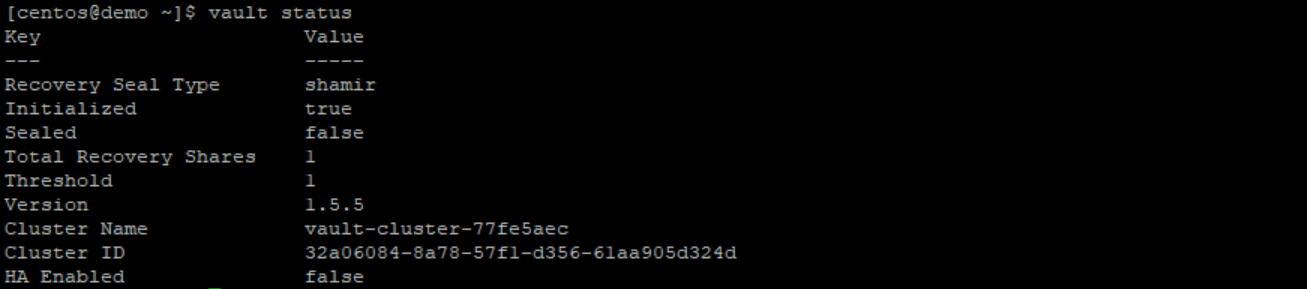
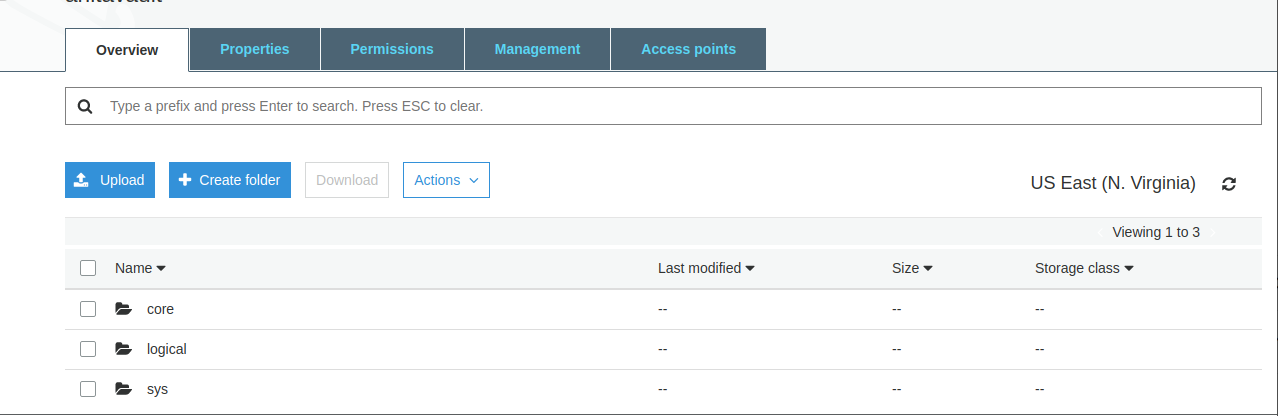
Check Vault version using vault –version command:  
  
C:\Users\anita.ty\Pictures\vault4.png

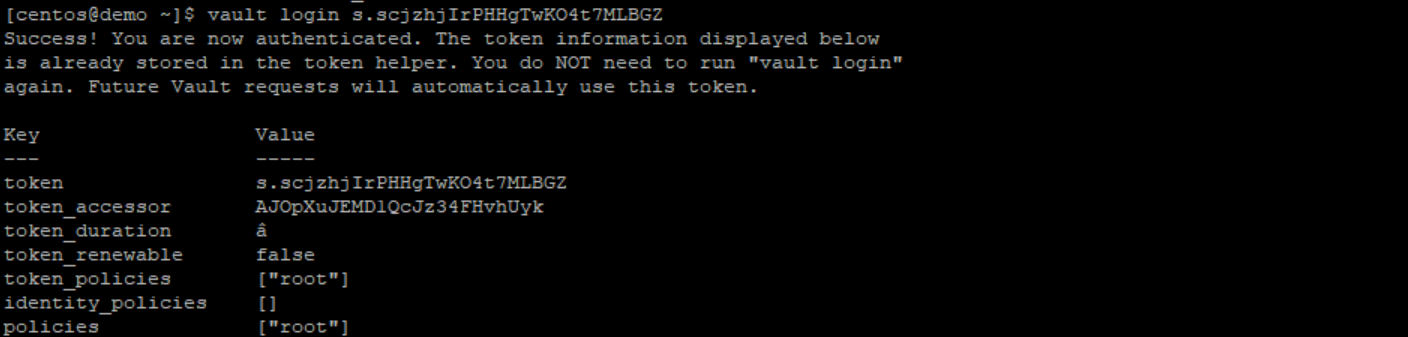
Figure 3.10 Check Vault version

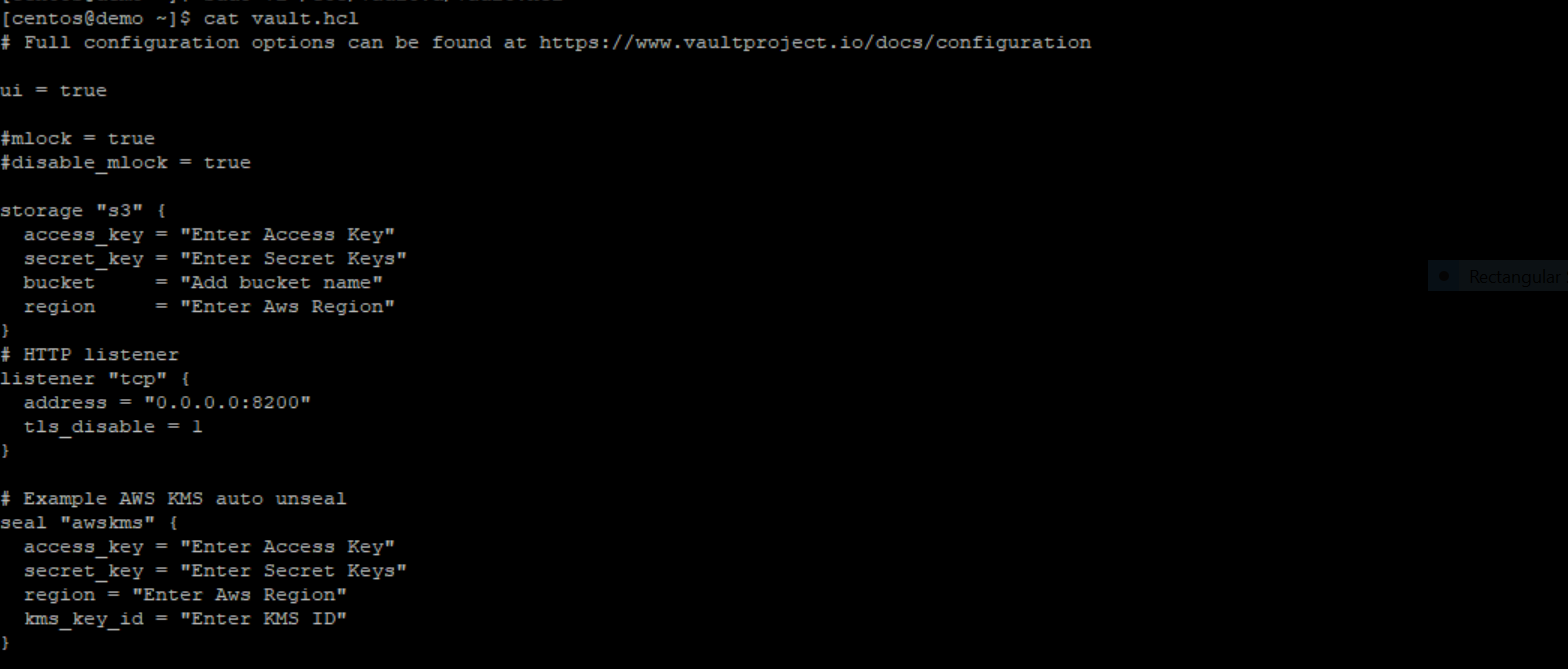
Check vault service status with below command  
  
   
 Figure 3.11 Check vault service status  
  
Now start vault service and check UI of Vault  
IP\_Address:8200  
  
   
  
Vault UI not working properly because of certificate, Now If you have certificate for Vault running on https, then upload certificate on the serevr or if you don’t have the certificate, you need to disable https in vault config file.

Update vault config file and disable https  
Uncomment HTTP listener block & Commented HTTPS listener block.  
   
   
 Figure 3.12 Disable https listener   
   
Save the /etc/vault.d/vault.hcl file with above changes and restart vault service.  
  
Now check UI After restart the vault service.

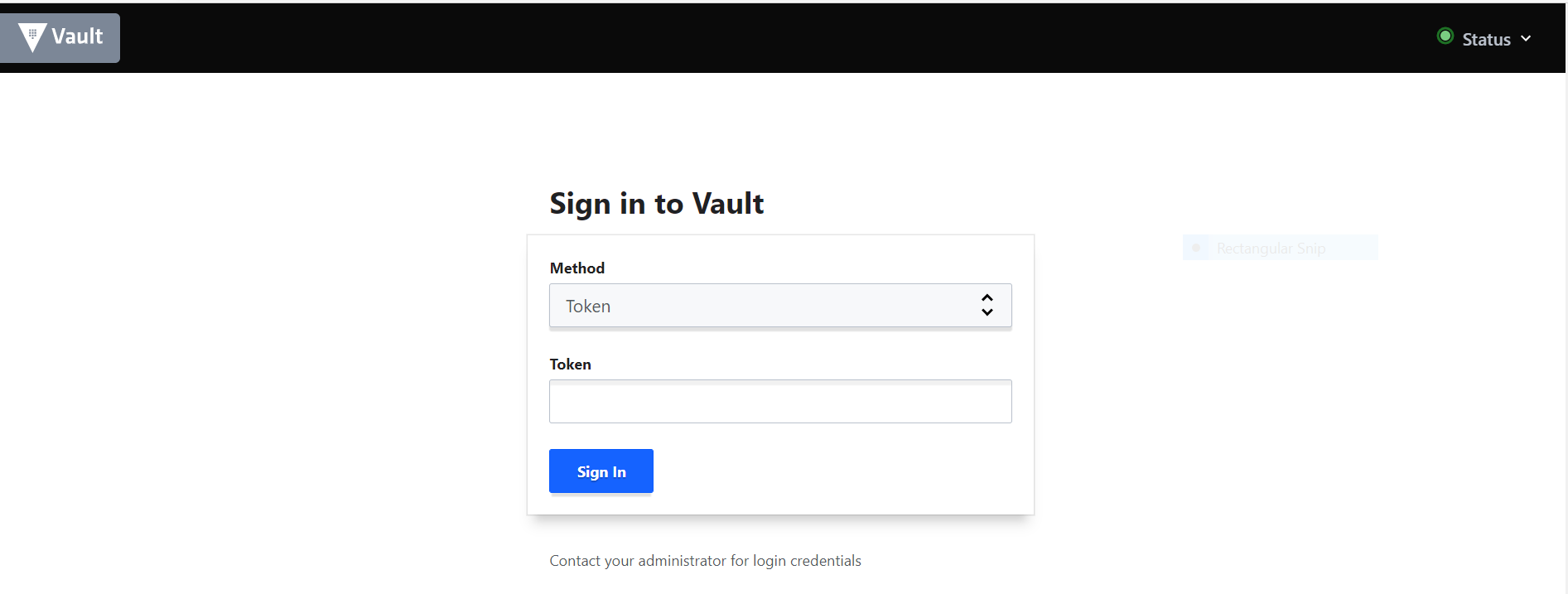
  
   
 Figure 3.13 Check Vault UI  
  
First, set the VAULT\_ADD environment variable:  
export VAULT\_ADDR=’http://127.0.0.1:8200’  
  
Now Check the vault server status  
  
  
  
 Figure 3.14 Check Vault status  
  
Before initialize the vault, you need to integrate S3 bucket as a backend and Update KMS keys in vault config(vault.hcl) file.  
  
Update **Access key,Secret key, bucket name and Region** in vault.hcl file for S3 storage.  
  
   
  
 Figure 3.15 Update config file for S3  
  
   
  
 Figure 3.16 Add KMS values  
  
Save the changes and restart vault service  
  
Run vault status command once done, make sure Recovery Seal Type **is awskms**  
You should see something similar to:  
   
   
 Figure 3.17 Check vault status   
  
 Run vault operator init command to initialize the vault server as follow:  
  
  
  
 Figure 3.18 Initialize Vault   
  
Now Run vault status command, once done make sure **Initialized is True & Sealed is False**.

  
  
 Figure 3.19 Check Vault Status  
  
Vault initialized once done, Check S3 bucket on AWS console and Vault automatically stored object in S3 bucket.  
  
You should see something similar to:  
  
  
  
 Figure 3.20 S3 bucket with Initialize object  
  
  
  
  
  
  
  
Log into Vault using the generated initial root token:

  
  
 Figure 3.21 Vault Login  
  
  
Check Final vault configuration file

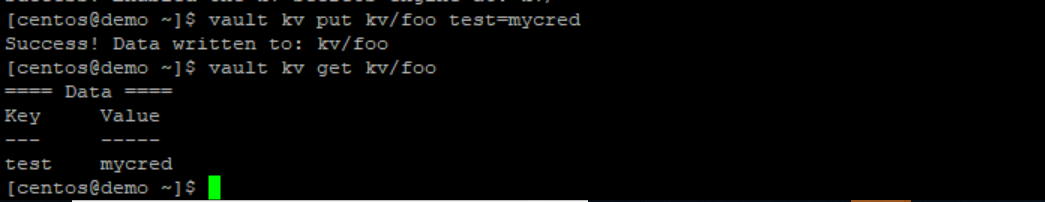
  
  
 Figure 3.22 Vault config File

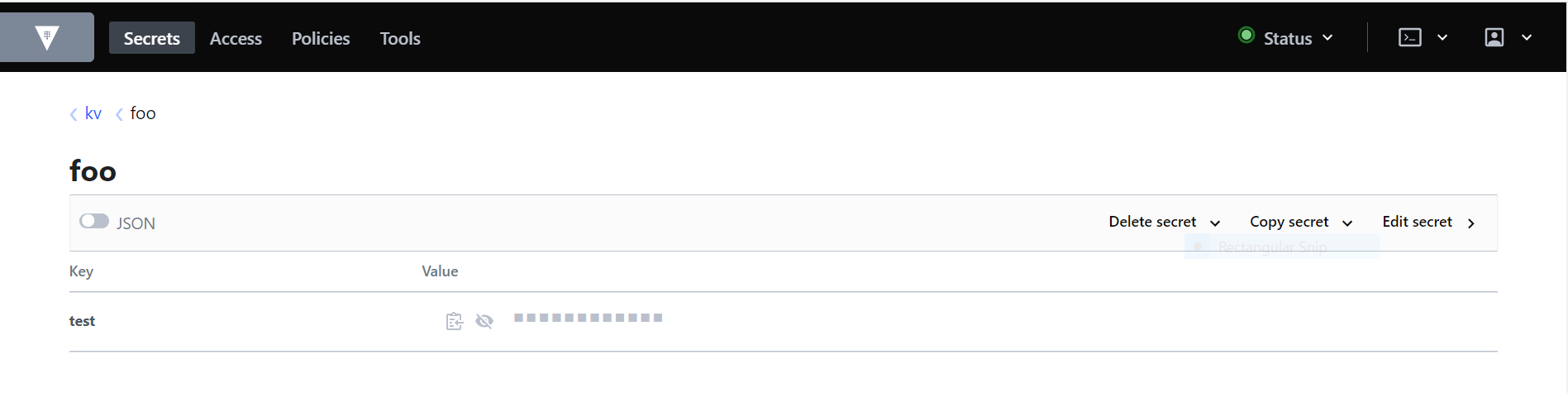
Check vault UI on IP\_Address:8200

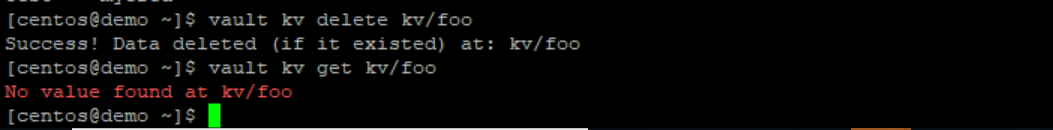
  
  
 Figure 3.23 Vault UI

Enter Root token for login the vault  
  
Enable Secrets engine

  
 Figure 3.24 Enable Secret Engine

Add some Credentials in Vault server for testing:  
  
  
  
 Figure 3.25 Enable Secret Engine  
  
Check created secrets on UI as well

  
  
 Figure 3.26 Check Secrets on Vault UI

Delete testing Credentials:  
  
  
  
 Figure 3.27 Delete Test Cred

# Cloud Vault:

Cloud vault is an online backup manager, A client-side backup application connecting to the off-site backup server. An intitutive interface allows non-experts to easily configure file backups. Native functionality for backing up Microsoft exchange, Microsoft SQL server, Mysql, Oracle and Lotus Notes is built-in at NO EXTRA COST.  
  
Cloud vault continuous backup to every 1 minutes. Continuous data backup technology and integration with ShadowProtect makes enterprise-grade recovery capability available to all.  
  
   
  
 Figure 3.27 Cloud Vault  
  
Advantages of Cloud Vault:

**Flexible retention policy:** You have complete control of how far back your data is available for restore. Every day for two weeks, every Friday for the last five weeks, the last day of each month for twelve months - you can even have all of the above at the same time if you wish.  
  
**Web Interface:** We do not need to install any software on any machine to restore data from Cloudvault. Just login to our web interface with credentials and choose what to restore. It's as easy as using your normal file manager.  
  
**Reduce disk usage:** The Cloudvault backup solution only needs to save the changes within your body of data. This means that your server is not reading redundant information every time you back up your files, prolonging the life of the disks and speeding up the backup process.  
  
  
**Fully encrypted storage:** Data stored on the Cloudvault server is fully encrypted before it even arrives and is meaningless without your password. The integrity of your data is checked regularly to ensure your data can be recovered on demand.

**Comprehensive reports:** At the end of every backup our system produces a detailed report which is compressed and emailed to your nominated member of staff. This gives you peace of mind knowing that individual files have been backed up, and how much storage is being used.

**Eliminate human error:** Analysis of backup failures has shown that approximately one third of all failures are due to human error. This includes failure to change the backup tape, failure to follow cleaning procedures, failure to store tapes correctly and failure to rotate tapes correctly.