**Source Codes**

1. **Module 1: Vault Setup and Configuration (Amazon Linux EC2)**

**Purpose:** Install, initialize, unseal, and run Vault on an EC2 instance.

**# Update the package list on the EC2 instance**

sudo apt-get update

**# Install the unzip utility required for extracting Vault**

sudo apt-get install -y unzip

**# Download the HashiCorp Vault binary from the official source**

wget https://releases.hashicorp.com/vault/1.9.0/vault\_1.9.0\_linux\_amd64.zip

**# Unzip the downloaded Vault archive**

unzip vault\_1.9.0\_linux\_amd64.zip

**# Move the Vault binary to a directory in the system's PATH**

sudo mv vault /usr/local/bin/

**# Initialize Vault to generate unseal keys and root token**

vault operator init

**# Unseal Vault using one of the generated unseal keys**

vault operator unseal <unseal\_key>

**# Start Vault in development mode (for testing purposes)**

vault server -dev

1. **Module 2: CI/CD Pipeline Integration (GitHub/Jenkins)**

**Purpose:** Set up Vault AppRole authentication for CI/CD tools.

**# Create a new AppRole named github-role and attach the github-policy**

vault write auth/approle/role/github-role policies="github-policy"

**# Retrieve the Role ID for github-role (used by CI/CD tools)**

vault read auth/approle/role/github-role/role-id

**# Generate a new Secret ID for the AppRole**

vault write -f auth/approle/role/github-role/secret-id

1. **Module 3: Database Credentials Management (DynamoDB via AWS Secrets** **Engine)**

**Purpose**: Configure Vault to dynamically generate and rotate DynamoDB credentials.

**# Enable the AWS secrets engine in Vault**

vault secrets enable aws

**# Configure the AWS root credentials for Vault to access AWS services**

vault write aws/config/root \

access\_key=<access\_key> \

secret\_key=<secret\_key> \

region=<your\_current\_region>

**# Create a read-only role for DynamoDB access using an IAM policy document**

vault write aws/roles/dynamodb-readonly \

policy\_document="{

\"Version\": \"2012-10-17\",

\"Statement\": [

{\"Effect\": \"Allow\", \"Action\": \"dynamodb:ListTables\", \"Resource\": \"\*\"},

{\"Effect\": \"Allow\", \"Action\": \"dynamodb:Query\", \"Resource\": \"arn:aws:dynamodb:us-west-2:123456789012:table/my-table\"}

]

}"

**# Generate temporary credentials for DynamoDB using the created role**

vault read aws/creds/dynamodb-readonly

1. **Module 4: Cloud-Native Application Secrets Management**

**Purpose:** Create a time-bound token for secure secret access in cloud-native apps.

**# Create a Vault token with a specific policy and expiration time (TTL)**

vault token create -policy="<your\_policy\_name>" -ttl="<hours>h"

**# (Use this token in your application code to securely fetch secrets)**

1. **Module 5: Secure Access Management for Microservices**

**Purpose:** Authenticate microservices securely using AppRole and Vault tokens.

**# Create a new AppRole for the microservice with specific policy access**

vault write auth/approle/role/micro-approle policies="microservice-a-policy"

**# Retrieve the Role ID for the microservice AppRole**

vault read auth/approle/role/micro-approle/role-id

**# Generate a new Secret ID to pair with the Role ID for login**

vault write -f auth/approle/role/micro-approle/secret-id

**# Log in to Vault using Role ID and Secret ID to get a client token**

vault write auth/approle/login \

role\_id="<role\_id>" \

secret\_id="<secret\_id>"