Following are the steps which should be followed for making an object of one module to be available for the other module at a high level: First, an output variable to be defined in a resource configuration. Till you do not declare resource configuration details, the scope of local and to a module. Now, you have to declare the output variable of module\_A to be used in other module's configuration. A brand new and latest key name should be created by you and the value should be kept equivalent to the module\_A’s output variable.

Now, for module\_B you have to create a file variable.tf. Establish an input variable inside this file having exactly the same name as was in the key defined by you in module\_B. In a module, this particular variable enables the resource's dynamic configuration. For making this variable available to some other module also, replicate the process. This is because the particular variable established here have its scope restricted to module\_B.

vault auth eable aws

vault write auth/aws/config/client

secretkey

accesskey

vault policy write "db-policy"

path

capabilities

vault write

auth/aws/role/app-db-role

auth\_type=ec2policies=db-policy

bound\_ami\_id=amiid

database secrets using

plugin

roles

create policies and apply to users

created users

uisng vault scenario workflow:

I worked on automating the creation of database secrets in vault using database secret engine and associated Auth methods and policies and users to it using terraform and jenkins. when the application team wants to store database credentials in vault I provide them terrafrom modules to create secrets, and some documentation on how to use the modules.once they commit changes in the code repository using webhooks which triggers jenkins to execute the build and build is approved by the teams admin and then secrets will be created in vault. skip this(In here jenkins uses the aws auth method to interact with vault) Later i create the

users and policies and some documentation on how to use the modules. once they make the changes and push it to masterbranch using webhooks a jenkins build triggers and the terraform plan and apply and secrets are created in vault and a role with a policy