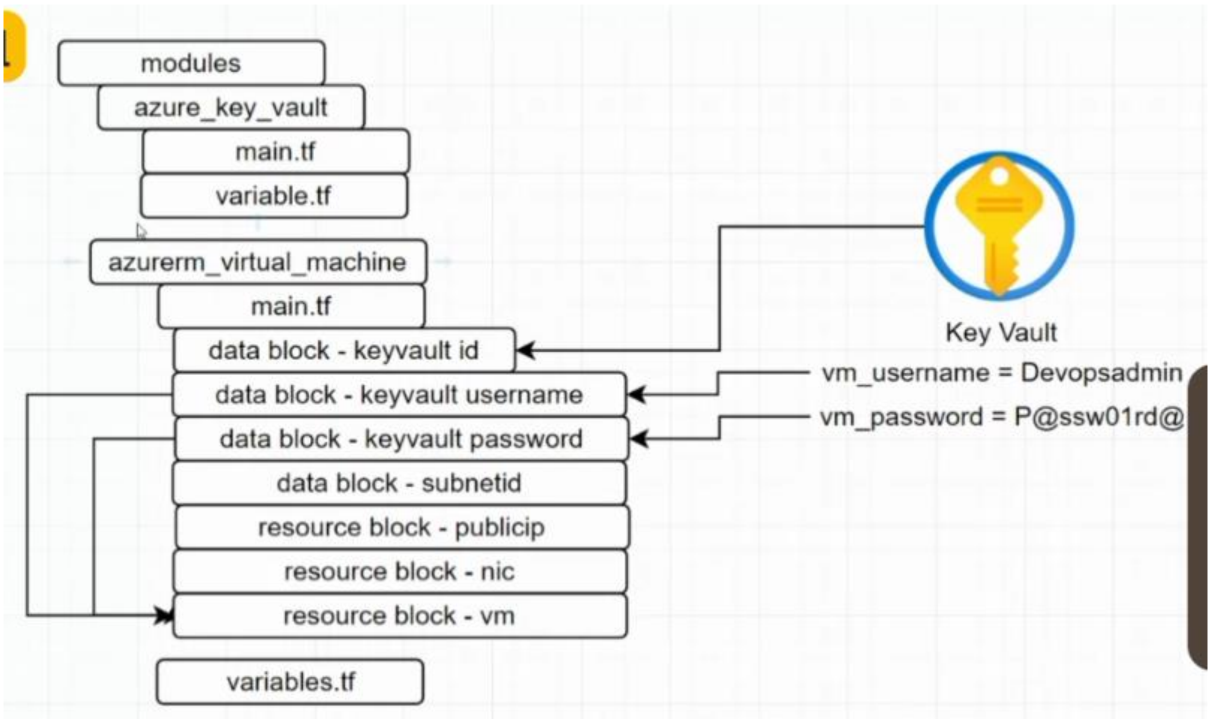
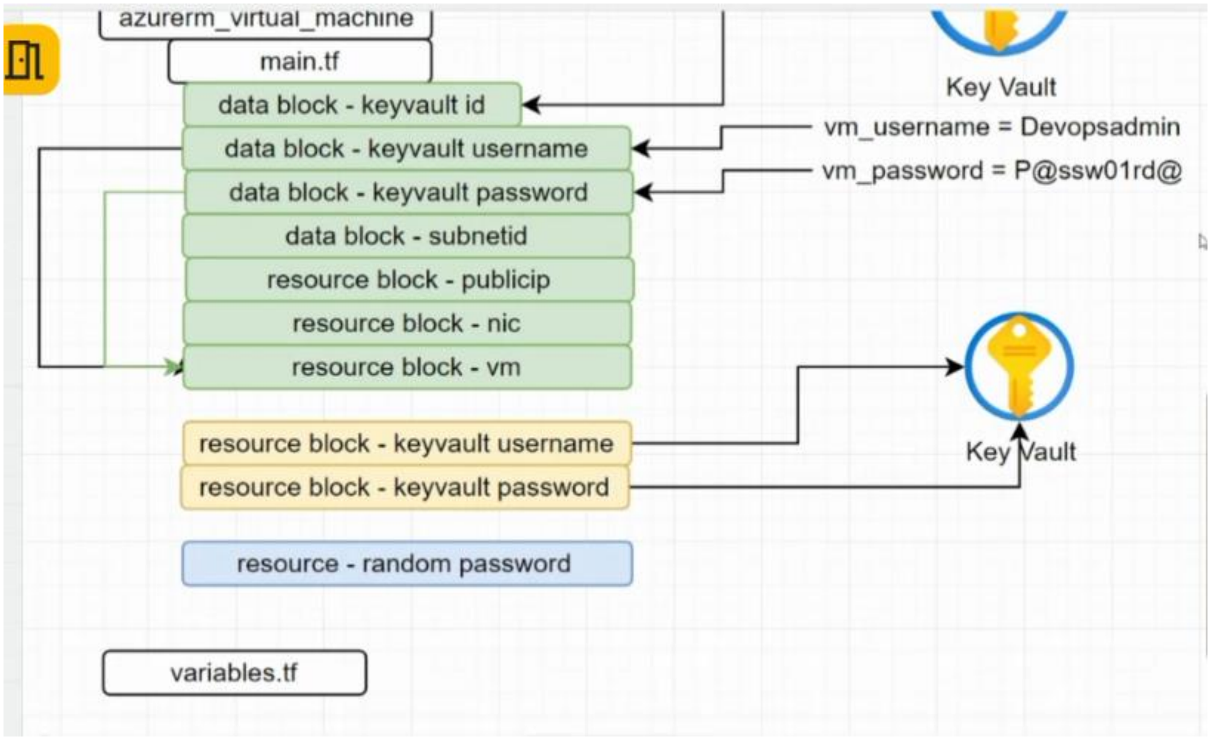
AGENDA – Deploying keyvault with terraform

1)

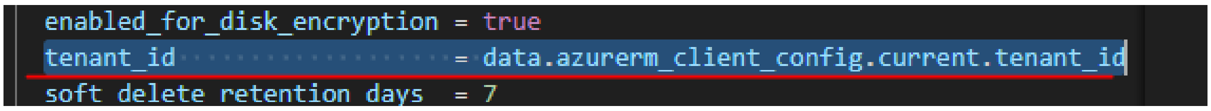


2) Now we will write code in a way that username and password will not be fetched from data block but it will be auto generated.

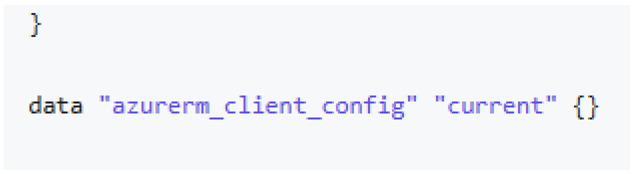


azurerm_virtual_machine
main.tf
data block - keyvault id
data block - keyvault username
data block - keyvault password
data block - subnetid
resource block - publicip
resource block - nic
resource block - vm
resource block - keyvault username
resource block - keyvault password
resource - random password
variables.tf
Key Vault
vm_username = Devopsadmin
vm_password = P@ssw01rd@
Key Vault

3) We have a data block to fetch tenant id

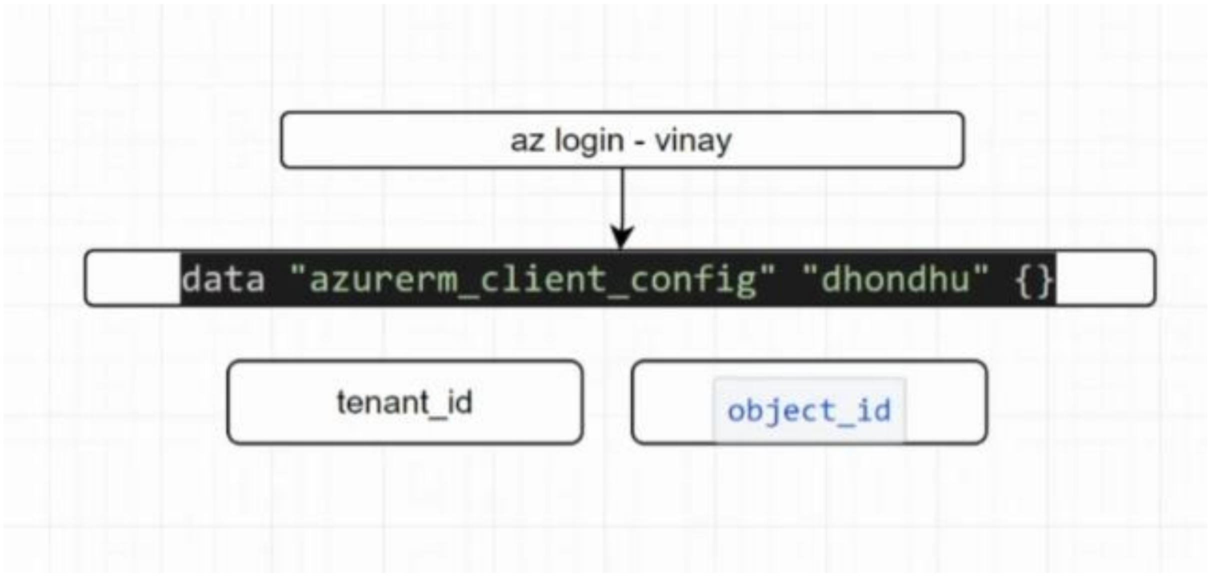


```
enabled_for_disk_encryption = true
tenant_id                   = data.azurerm_client_config.current.tenant_id
soft_delete_retention_days  = 7
```

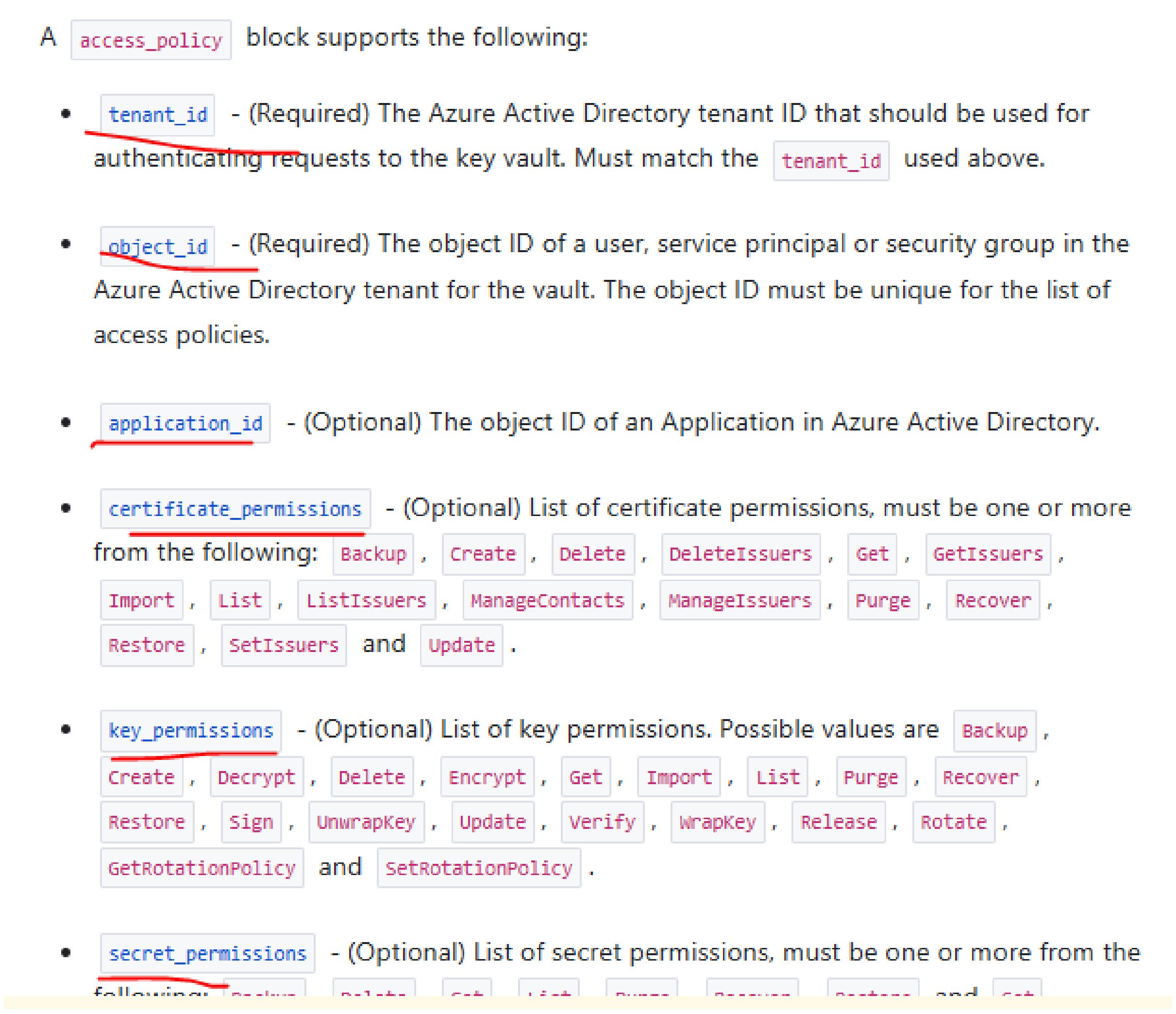


```
}

data "azurerm_client_config" "current" {}
```

4)



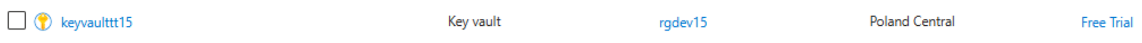
5) Our access_policy attribute permits below fields also



A `access_policy` block supports the following:

- `tenant_id` - (Required) The Azure Active Directory tenant ID that should be used for authenticating requests to the key vault. Must match the `tenant_id` used above.
- `object_id` - (Required) The object ID of a user, service principal or security group in the Azure Active Directory tenant for the vault. The object ID must be unique for the list of access policies.
- `application_id` - (Optional) The object ID of an Application in Azure Active Directory.
- `certificate_permissions` - (Optional) List of certificate permissions, must be one or more from the following: `Backup` , `Create` , `Delete` , `DeleteIssuers` , `Get` , `GetIssuers` , `Import` , `List` , `ListIssuers` , `ManageContacts` , `ManageIssuers` , `Purge` , `Recover` , `Restore` , `SetIssuers` and `Update` .
- `key_permissions` - (Optional) List of key permissions. Possible values are `Backup` , `Create` , `Decrypt` , `Delete` , `Encrypt` , `Get` , `Import` , `List` , `Purge` , `Recover` , `Restore` , `Sign` , `UnwrapKey` , `Update` , `Verify` , `WrapKey` , `Release` , `Rotate` , `GetRotationPolicy` and `SetRotationPolicy` .
- `secret_permissions` - (Optional) List of secret permissions, must be one or more from the

6) Now keyvault is created in azure portal by terraform

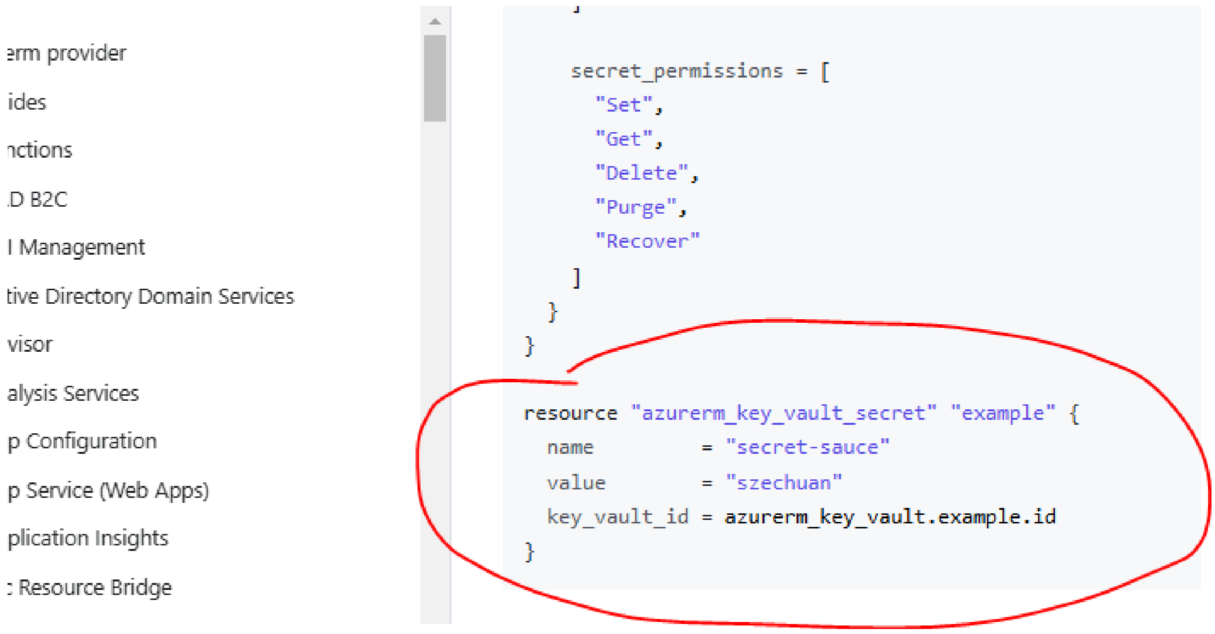


☐ keyvaulttt15 Key vault rgdev15 Poland Central Free Trial

+++

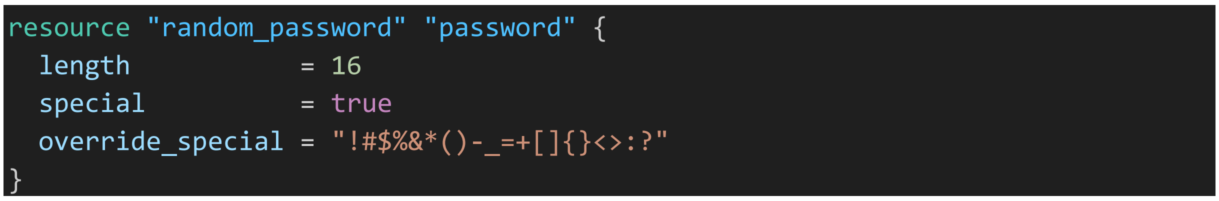
AGENDA – Whenever any keyvault is created then any random password should get generated and gets stored in created keyvault.

1) SEARCH – azurerm_keyvault_secret

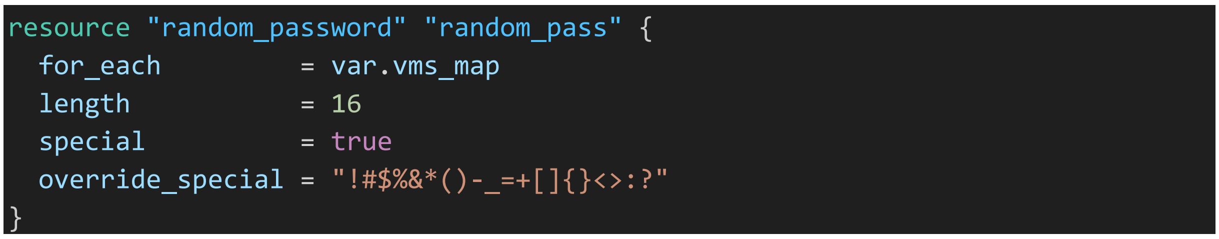


```
    ]

    secret_permissions = [
      "Set",
      "Get",
      "Delete",
      "Purge",
      "Recover"
    ]
  }
}

resource "azurerm_key_vault_secret" "example" {
  name         = "secret-sauce"
  value        = "szechuan"
  key_vault_id = azurerm_key_vault.example.id
}
```

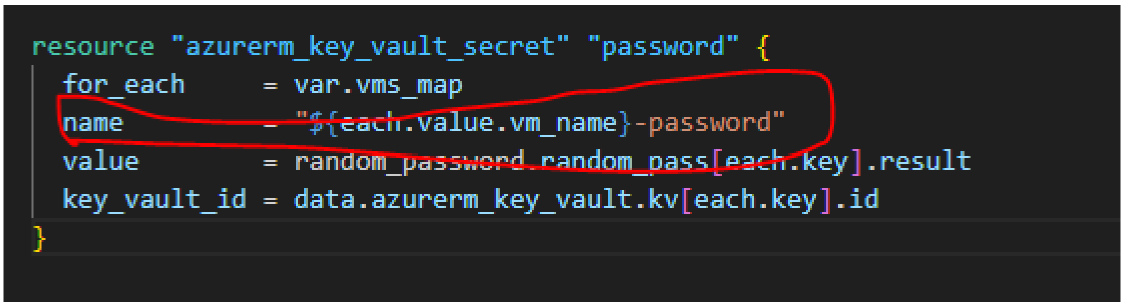
2) SEARCH – random password terraform



```
resource "random_password" "password" {
  length           = 16
  special          = true
  override_special = "!#$%&*()-_=+[]{}<>:?"
}
```

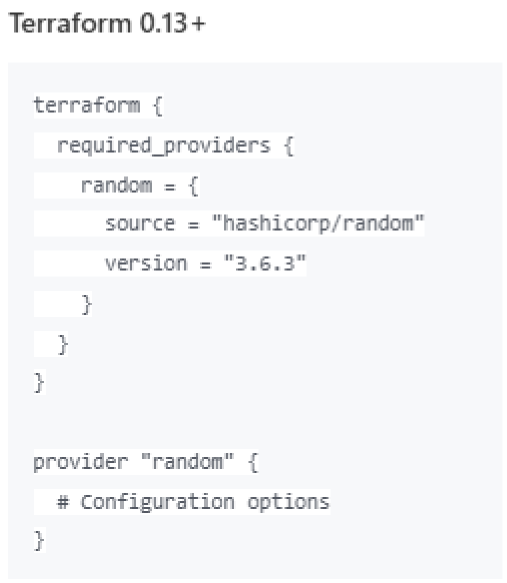


```
resource "random_password" "random_pass" {
  for_each         = var.vms_map
  length           = 16
  special          = true
  override_special = "!#$%&*()-_=+[]{}<>:?"
}
```



```
resource "azurerm_key_vault_secret" "password" {
  for_each     = var.vms_map
  name         = "${each.value.vm_name}-password"
  value        = random_password.random_pass[each.key].result
  key_vault_id = data.azurerm_key_vault.kv[each.key].id
}
```

3) Add random provider code also in providers.tf file

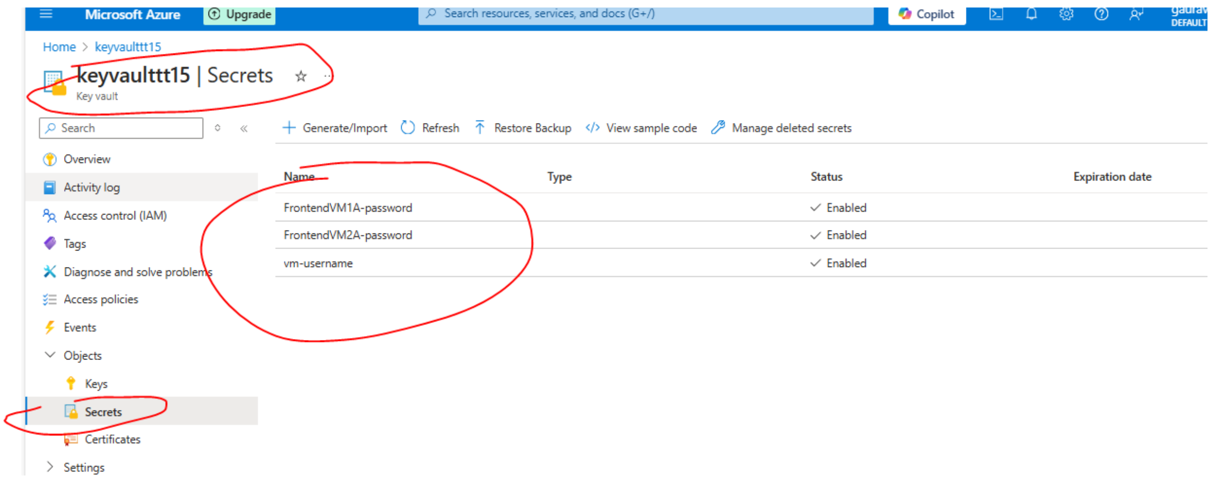


Terraform 0.13+

```
terraform {
  required_providers {
    random = {
      source = "hashicorp/random"
      version = "3.6.3"
    }
  }
}

provider "random" {
  # Configuration options
}
```

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "4.6.0"
    }
    random = {
      source = "hashicorp/random"
      version = "3.6.3"
    }
  }
}

provider "azurerm" {
  features {}
  subscription_id = "82812723-cb7c-49c5-b697-018df19bfc17"
}

provider "random" {}
```

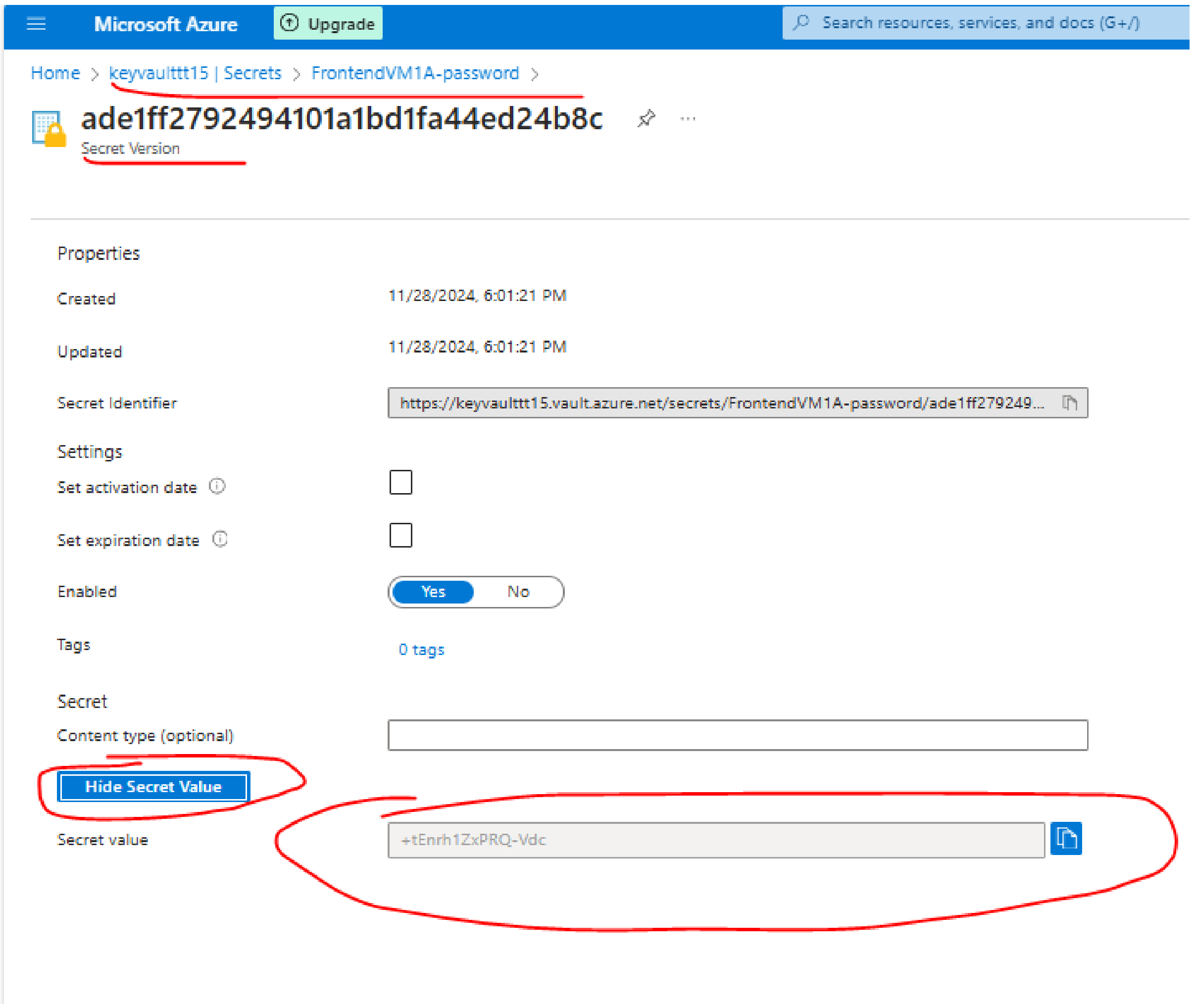
4) Now run terraform apply, so we can see in keyvault, under secret is vm username and password for individual vms is generated



keyvaulttt15 | Secrets
Key vault

Name	Type	Status	Expiration date
FrontendVM1A-password		✓ Enabled	
FrontendVM2A-password		✓ Enabled	
vm-username		✓ Enabled	

5) For vm1 generated password is shown below


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Home > keyvaulttt15 | Secrets > FrontendVM1A-password >
Hide Secret Value

ade1ff2792494101a1bd1fa44ed24b8c
Secret Version

Properties

Created	11/28/2024, 6:01:21 PM
Updated	11/28/2024, 6:01:21 PM
Secret Identifier	https://keyvaulttt15.vault.azure.net/secrets/FrontendVM1A-password/ade1ff279249...

Settings

Set activation date	☐
Set expiration date	☐
Enabled	[✓ Yes] No
Tags	0 tags

Secret

Content type (optional)	
Secret value	+tEnrh1ZxPRQ-Vdc

6)