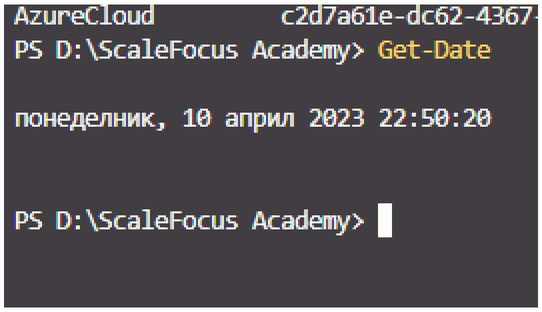
Terraform Exercise

Task 1: Install terraform and Azure CLI

1. Use official guidelines to install the latest version of terraform and Azure CLI
    - I have already installed both, Azure CLI and Terraform previously, so here I am only checking with the Terraform version using the command ***terraform -v***
2. Authenticate with Azure CLI
    - ***az login*** - browser window pops up where I need to log in with my Microsoft credentials to get access to Azure
3. Set the exercise subscription as default for Azure CLI
    - ***az account set –subscription 'Azure Pass - Sponsorship'*** - here, we can either set the subscription ID or the name of the subscription as default (either option is fine)
4. Provide console print screen:

4.1 Time and date when the exercise was worked 4.2 Output of the terraform command that will print out the Terraform version installed

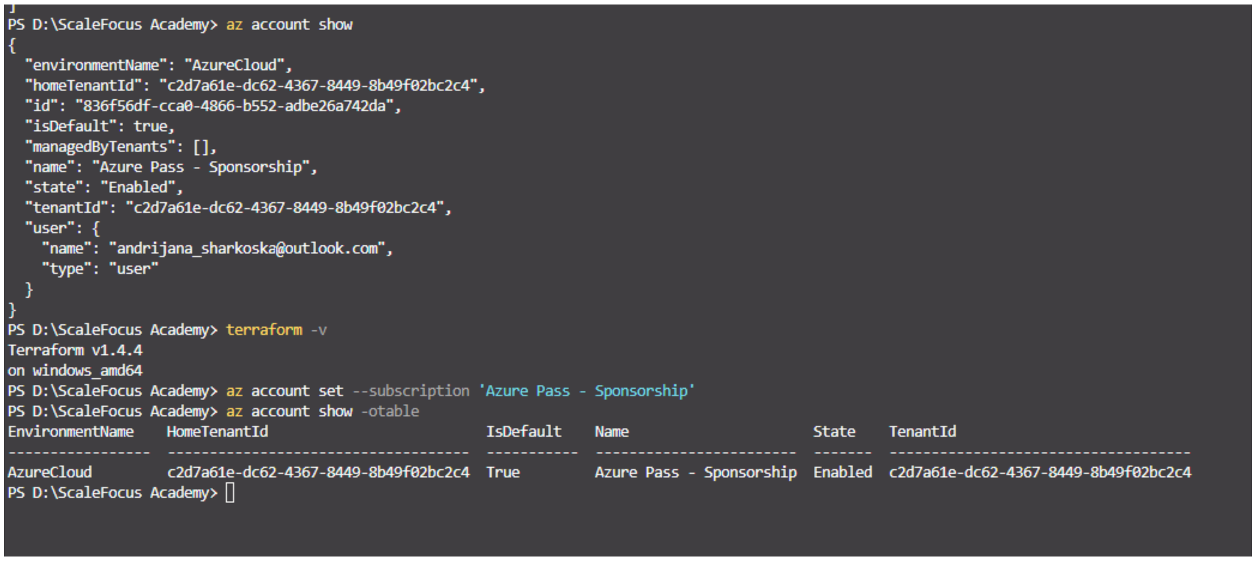
- We use the command ***Get-Date***



```
AzureCloud        c2d7a61e-dc62-4367-
PS D:\ScaleFocus Academy> Get-Date

понеделник, 10 април 2023 22:50:20


PS D:\ScaleFocus Academy>
```

4.3 Azure CLI output of the current subscription

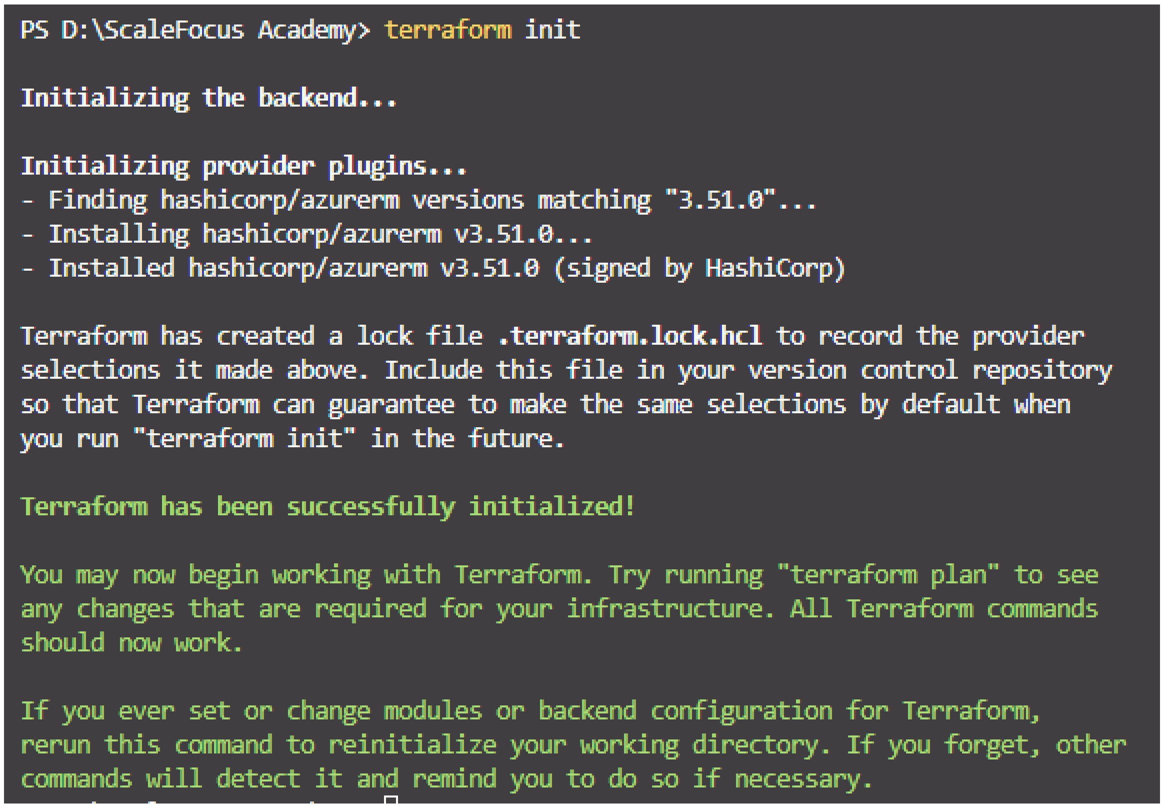
- ***az account show -otable*** to get a tabular representation of our Azure environment, with all of the subscription data provided.
- ***az account show*** it's basically the same as the previous, with an object-like visual representation of the subscription information.



```
PS D:\ScaleFocus Academy> az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": "c2d7a61e-dc62-4367-8449-8b49f02bc2c4",
  "id": "836f56df-cca0-4866-b552-adbe26a742da",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure Pass - Sponsorship",
  "state": "Enabled",
  "tenantId": "c2d7a61e-dc62-4367-8449-8b49f02bc2c4",
  "user": {
    "name": "andrijana_sharkoska@outlook.com",
    "type": "user"
  }
}
PS D:\ScaleFocus Academy> terraform -v
Terraform v1.4.4
on windows_amd64
PS D:\ScaleFocus Academy> az account set --subscription 'Azure Pass - Sponsorship'
PS D:\ScaleFocus Academy> az account show -otable
EnvironmentName    HomeTenantId                          IsDefault    Name                      State    TenantId
-----------------  ------------------------------------  -----------  ------------------------  -------  ------------------------------------
AzureCloud         c2d7a61e-dc62-4367-8449-8b49f02bc2c4  True         Azure Pass - Sponsorship  Enabled  c2d7a61e-dc62-4367-8449-8b49f02bc2c4
PS D:\ScaleFocus Academy>
```

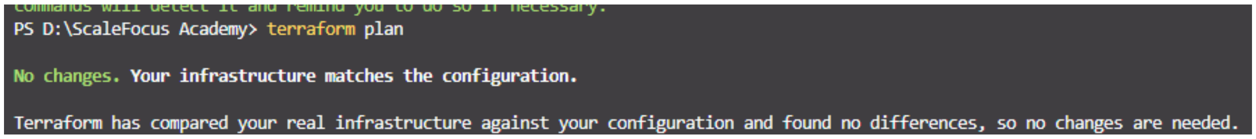
Task 2: Define your first terraform infrastructure code

- We initialize the terraform config file with ***terraform init***



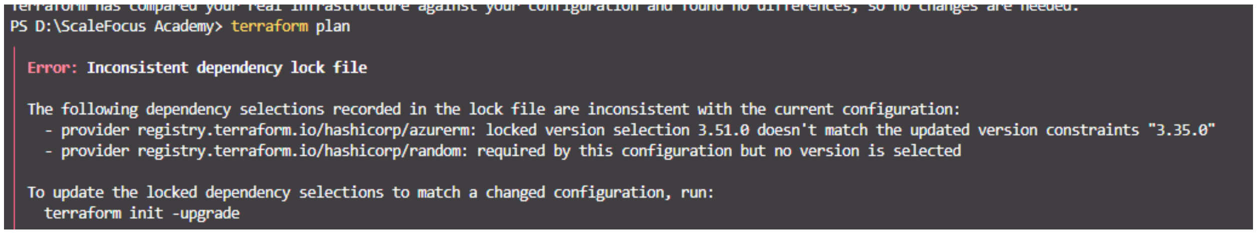
```
PS D:\ScaleFocus Academy> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "3.51.0"...
- Installing hashicorp/azurerm v3.51.0...
- Installed hashicorp/azurerm v3.51.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- Executing ***terraform plan*** command:



```
PS D:\ScaleFocus Academy> terraform plan

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
```

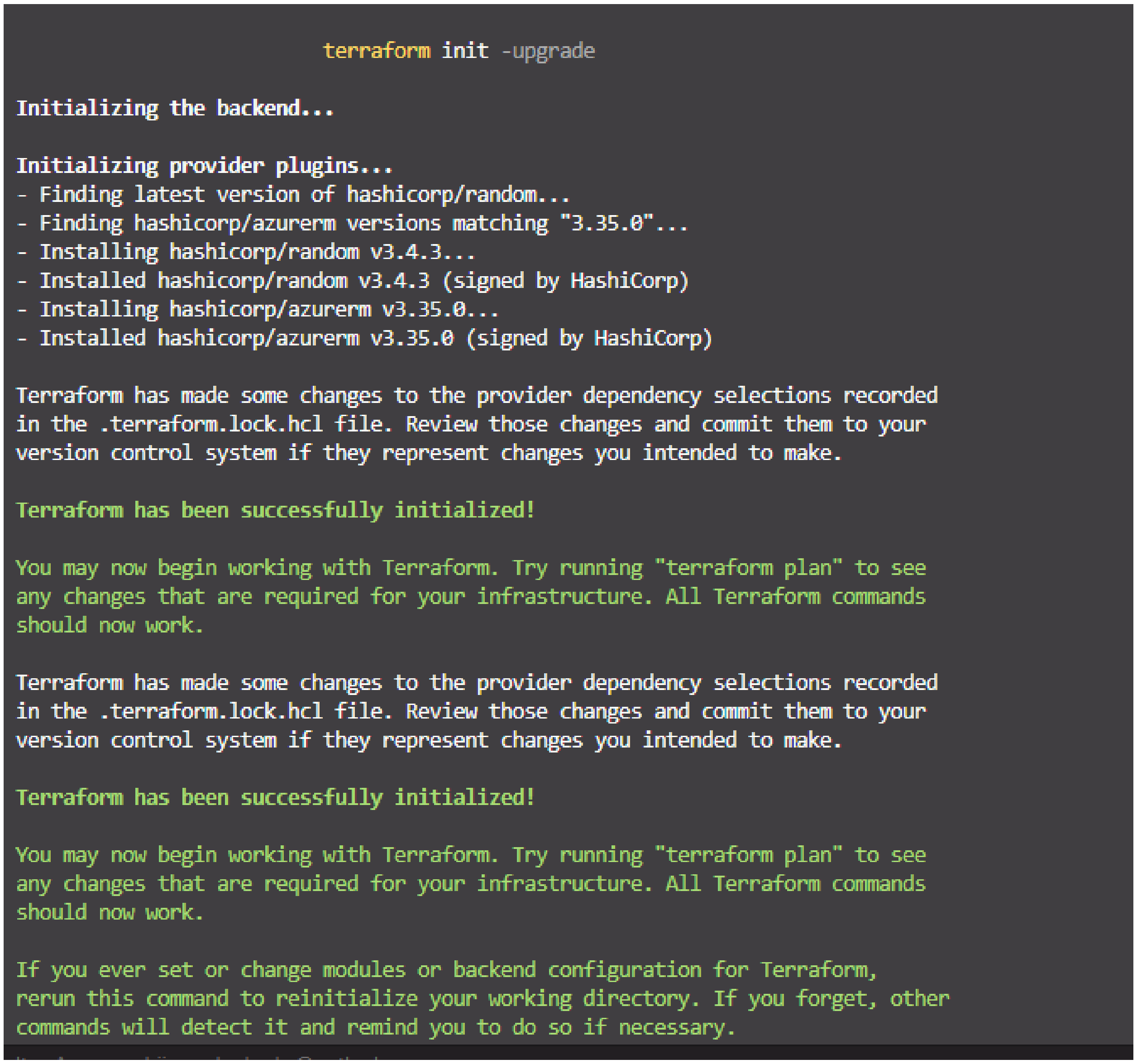
After the resource configuration has been added, we run again the command ***terraform plan***:



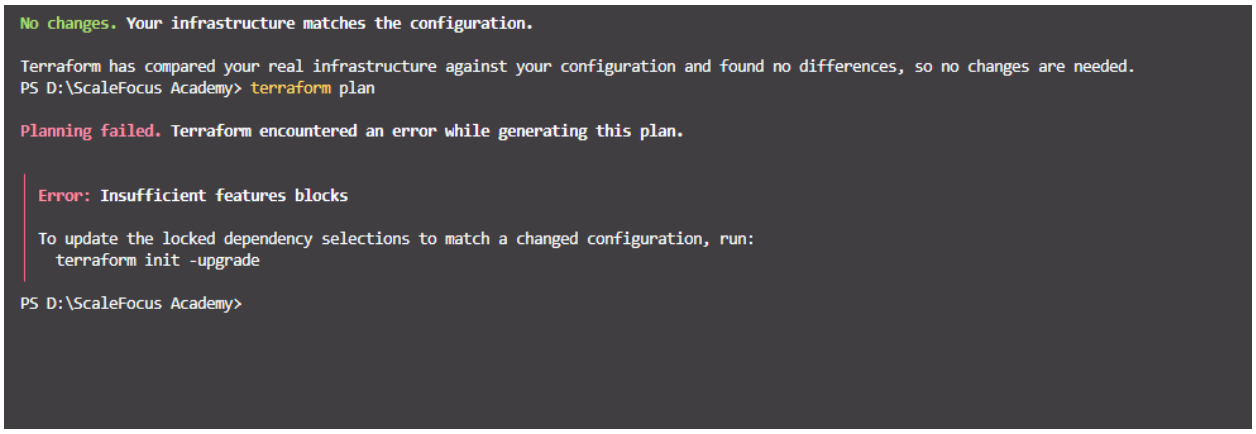
```
PS D:\ScaleFocus Academy> terraform plan

│ Error: Inconsistent dependency lock file
│
│ The following dependency selections recorded in the lock file are inconsistent with the current configuration:
│   - provider registry.terraform.io/hashicorp/azurerm: locked version selection 3.51.0 doesn't match the updated version constraints "3.35.0"
│   - provider registry.terraform.io/hashicorp/random: required by this configuration but no version is selected
│
│ To update the locked dependency selections to match a changed configuration, run:
│   terraform init -upgrade
```

Now we have problem that terraform is initialized with different provider version and we must reinitialize with “upgrade” terraform to get the assigned version.

- We have to upgrade the version after changing it to 3.35.0

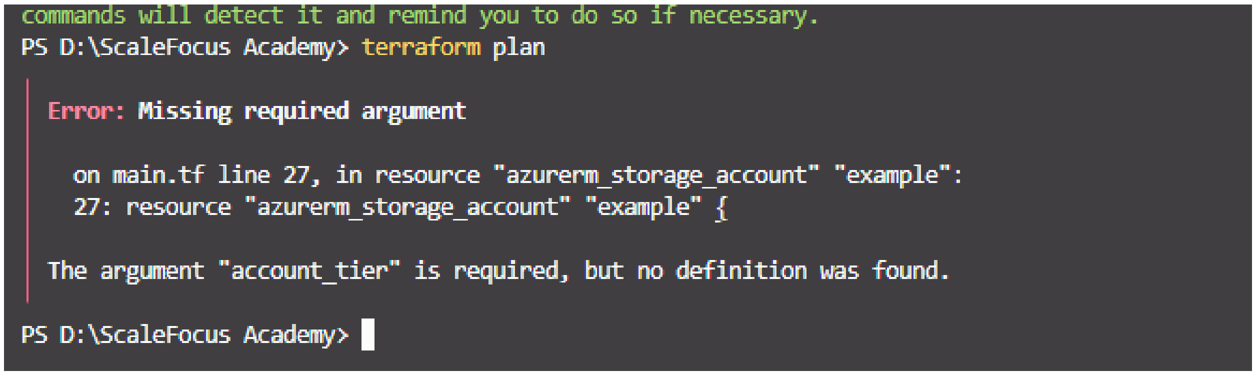


```
                           terraform init -upgrade

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding hashicorp/azurerm versions matching "3.35.0"...
- Installing hashicorp/random v3.4.3...
- Installed hashicorp/random v3.4.3 (signed by HashiCorp)
- Installing hashicorp/azurerm v3.35.0...
- Installed hashicorp/azurerm v3.35.0 (signed by HashiCorp)

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

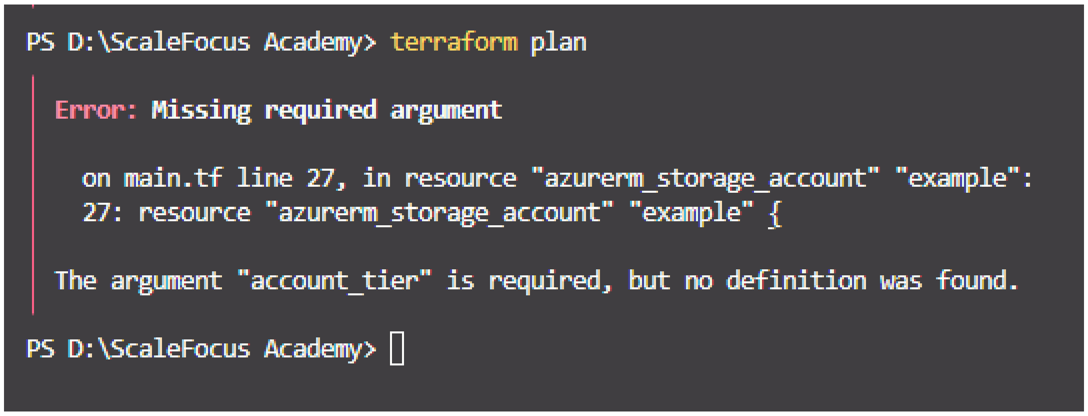


```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
PS D:\ScaleFocus Academy> terraform plan

Planning failed. Terraform encountered an error while generating this plan.

│
│ Error: Insufficient features blocks
│
│ To update the locked dependency selections to match a changed configuration, run:
│   terraform init -upgrade
│
PS D:\ScaleFocus Academy>
```

- However, we still get an issue with the restore_policy feature in the blob_properties section
- It was a feature introduced in the terraform version 3.36.0, and we are using an older version in this case.



```
commands will detect it and remind you to do so if necessary.
PS D:\ScaleFocus Academy> terraform plan

 Error: Missing required argument

   on main.tf line 27, in resource "azurerm_storage_account" "example":
   27: resource "azurerm_storage_account" "example" {

 The argument "account_tier" is required, but no definition was found.

PS D:\ScaleFocus Academy> 
```

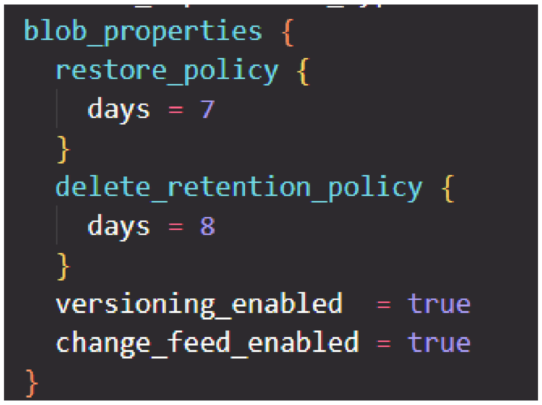


```
PS D:\ScaleFocus Academy> terraform plan

 Error: Missing required argument

   on main.tf line 27, in resource "azurerm_storage_account" "example":
   27: resource "azurerm_storage_account" "example" {

 The argument "account_tier" is required, but no definition was found.

PS D:\ScaleFocus Academy> 
```

- Now we are getting a different error. The cause of this error is:

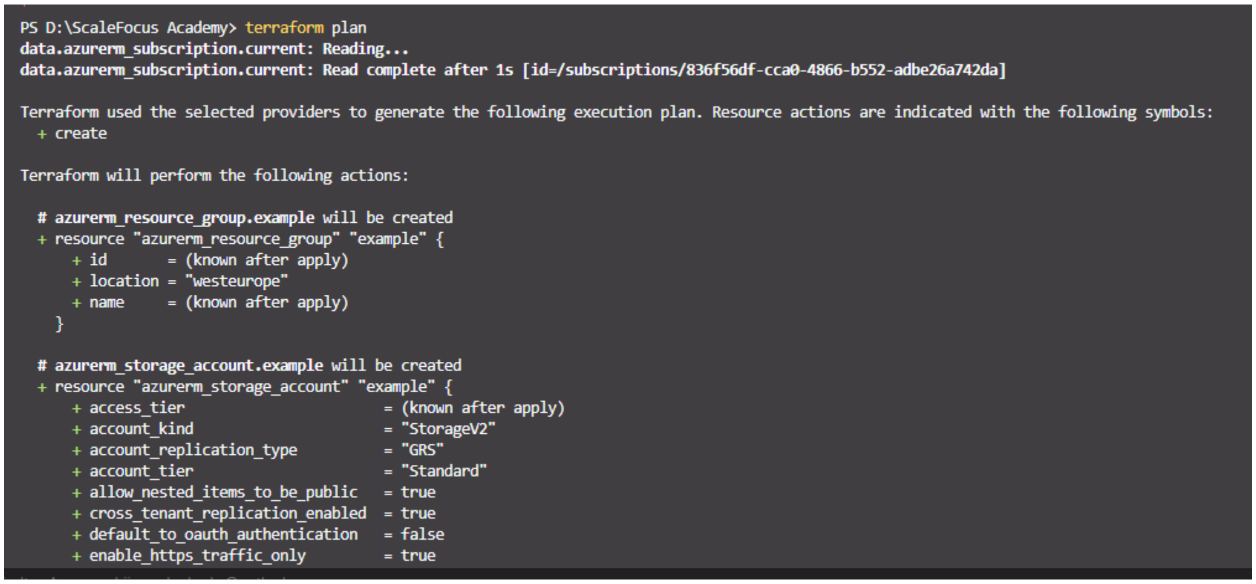
restore_policy - (Optional) A restore_policy block as defined below. This must be used together with delete_retention_policy set, versioning_enabled and change_feed_enabled set to true.

So, we must use these features if we are setting the restore_policy block.

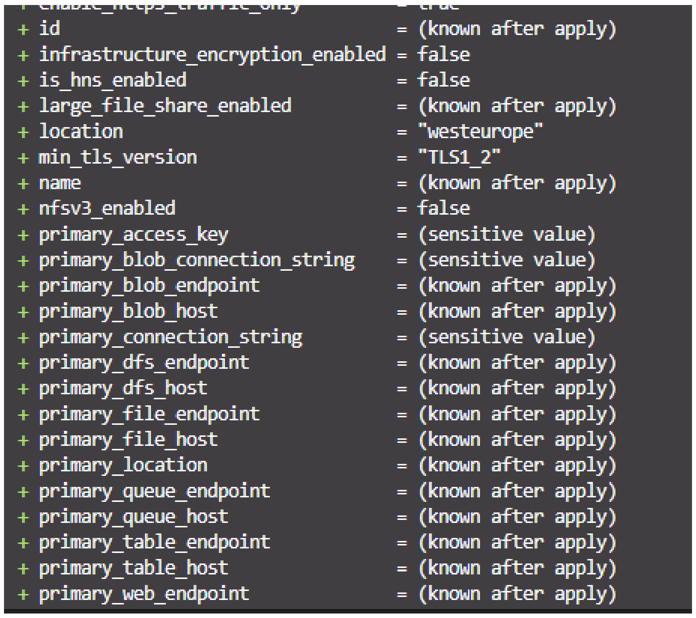


```
blob_properties {
  restore_policy {
    days = 7
  }
  delete_retention_policy {
    days = 8
  }
  versioning_enabled  = true
  change_feed_enabled = true
}
```

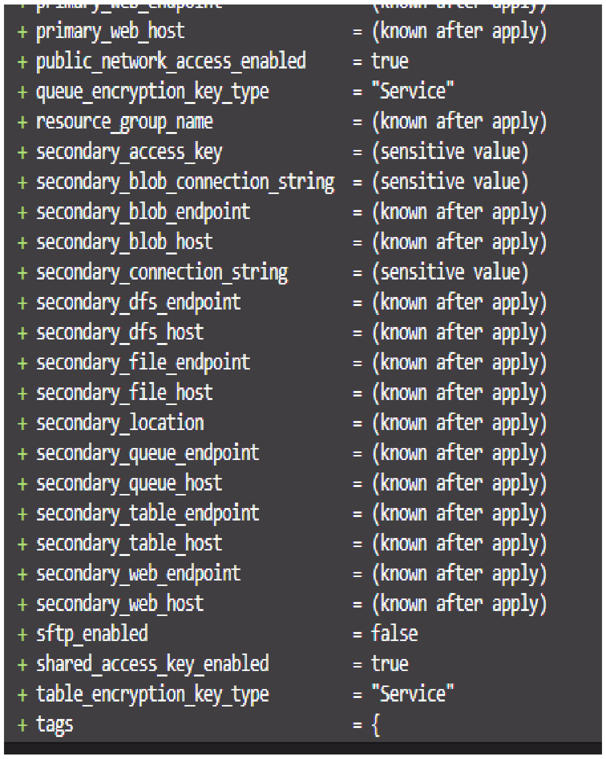
We run the command ***terraform plan*** once again with all of the configuration set up:



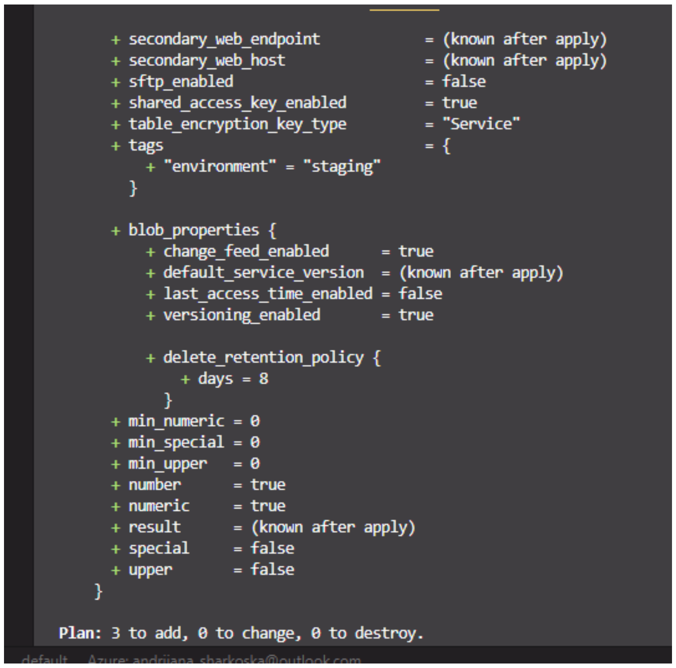
```
PS D:\ScaleFocus Academy> terraform plan
data.azurerm_subscription.current: Reading...
data.azurerm_subscription.current: Read complete after 1s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.example will be created
  + resource "azurerm_resource_group" "example" {
      + id       = (known after apply)
      + location = "westeurope"
      + name     = (known after apply)
    }

  # azurerm_storage_account.example will be created
  + resource "azurerm_storage_account" "example" {
      + access_tier                        = (known after apply)
      + account_kind                       = "StorageV2"
      + account_replication_type           = "GRS"
      + account_tier                       = "Standard"
      + allow_nested_items_to_be_public    = true
      + cross_tenant_replication_enabled   = true
      + default_to_oauth_authentication    = false
      + enable_https_traffic_only          = true
```



```
+ enable_https_traffic_only          = true
+ id                                 = (known after apply)
+ infrastructure_encryption_enabled  = false
+ is_hns_enabled                     = false
+ large_file_share_enabled           = (known after apply)
+ location                           = "westeurope"
+ min_tls_version                    = "TLS1_2"
+ name                               = (known after apply)
+ nfsv3_enabled                      = false
+ primary_access_key                 = (sensitive value)
+ primary_blob_connection_string     = (sensitive value)
+ primary_blob_endpoint              = (known after apply)
+ primary_blob_host                  = (known after apply)
+ primary_connection_string          = (sensitive value)
+ primary_dfs_endpoint               = (known after apply)
+ primary_dfs_host                   = (known after apply)
+ primary_file_endpoint              = (known after apply)
+ primary_file_host                  = (known after apply)
+ primary_location                   = (known after apply)
+ primary_queue_endpoint             = (known after apply)
+ primary_queue_host                 = (known after apply)
+ primary_table_endpoint             = (known after apply)
+ primary_table_host                 = (known after apply)
+ primary_web_endpoint               = (known after apply)
```



```
+ primary_web_endpoint               = (known after apply)
+ primary_web_host                   = (known after apply)
+ public_network_access_enabled      = true
+ queue_encryption_key_type          = "Service"
+ resource_group_name                = (known after apply)
+ secondary_access_key               = (sensitive value)
+ secondary_blob_connection_string   = (sensitive value)
+ secondary_blob_endpoint            = (known after apply)
+ secondary_blob_host                = (known after apply)
+ secondary_connection_string        = (sensitive value)
+ secondary_dfs_endpoint             = (known after apply)
+ secondary_dfs_host                 = (known after apply)
+ secondary_file_endpoint            = (known after apply)
+ secondary_file_host                = (known after apply)
+ secondary_location                 = (known after apply)
+ secondary_queue_endpoint           = (known after apply)
+ secondary_queue_host               = (known after apply)
+ secondary_table_endpoint           = (known after apply)
+ secondary_table_host               = (known after apply)
+ secondary_web_endpoint             = (known after apply)
+ secondary_web_host                 = (known after apply)
+ sftp_enabled                       = false
+ shared_access_key_enabled          = true
+ table_encryption_key_type          = "Service"
+ tags                               = {
```



```
      + secondary_web_endpoint            = (known after apply)
      + secondary_web_host                = (known after apply)
      + sftp_enabled                      = false
      + shared_access_key_enabled         = true
      + table_encryption_key_type         = "Service"
      + tags                              = {
          + "environment" = "staging"
        }

      + blob_properties {
          + change_feed_enabled       = true
          + default_service_version   = (known after apply)
          + last_access_time_enabled  = false
          + versioning_enabled        = true

          + delete_retention_policy {
              + days = 8
            }
      + min_numeric = 0
      + min_special = 0
      + min_upper   = 0
      + number      = true
      + numeric     = true
      + result      = (known after apply)
      + special     = false
      + upper       = false
    }

Plan: 3 to add, 0 to change, 0 to destroy.
```

default   Azure: andrijana.sharkoska@outlook.com

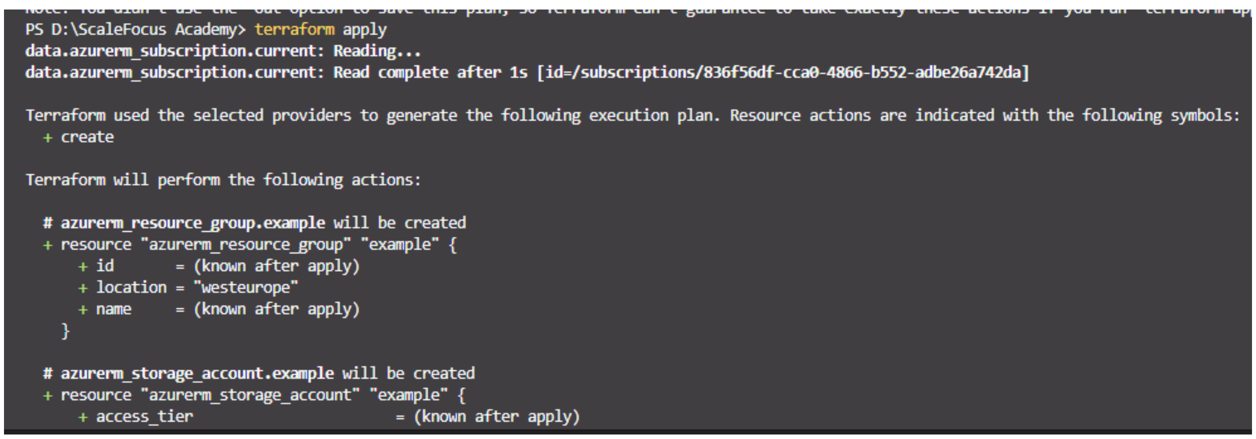
2.7.1 How many resources have you defined in your code and how many resources does the plan output show? Are they the same and why?

- I've defined 3 resources in my code: azure resource group, azure storage account and a random string generator, and got only one resource in the output.
- They are not the same, because some of the fields in the resources have been added later after we've run the command terraform plan, and furthermore, after the plan is created additional resources will be known after we apply the configuration plan.

2.7.2 What is the location of your resource group and what is the location of the storage account?

- The location of the resource group and the storage account is ***West Europe***.

2.8 Deploy your code to on the subscription and answer the questions bellow:



```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform ap
PS D:\ScaleFocus Academy> terraform apply
data.azurerm_subscription.current: Reading...
data.azurerm_subscription.current: Read complete after 1s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.example will be created
  + resource "azurerm_resource_group" "example" {
      + id       = (known after apply)
      + location = "westeurope"
      + name     = (known after apply)
    }

  # azurerm_storage_account.example will be created
  + resource "azurerm_storage_account" "example" {
      + access_tier                       = (known after apply)
```

- We deploy with the command ***terraform apply***

2.8.1 How many resources do you have on your subscription? (To list all resources, type “All resources” in the search bar on the top in Azure Portal)

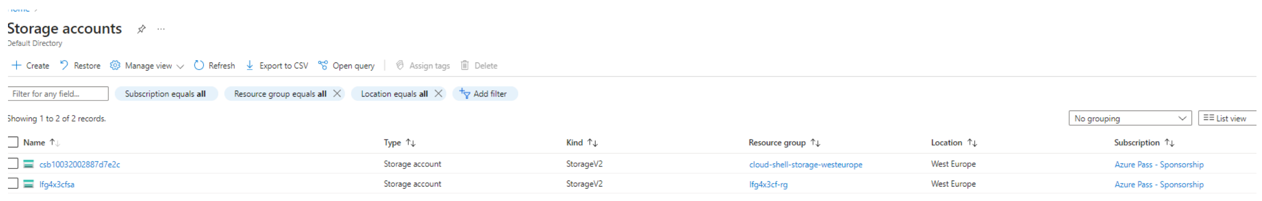
- No, they are not the same

2.8.2 Are the number of resources shown in the All resources portal window the same with the ones from your plan?

- No, some resources are not shown.

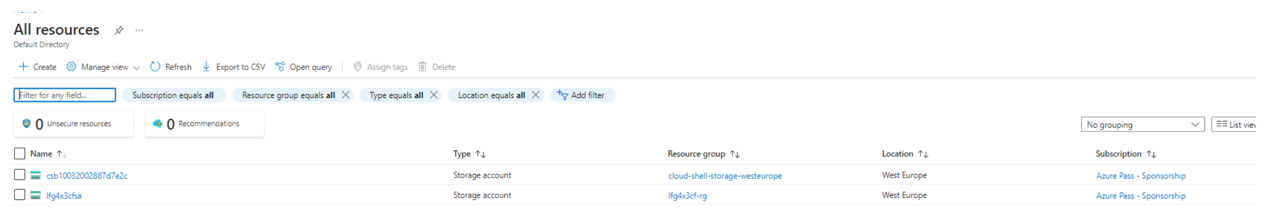
2.8.3 Give short explanation about the resources that are not shown?

2.8.4 Provide print screen of your portal with all resources.



Showing 1 to 2 of 2 records.

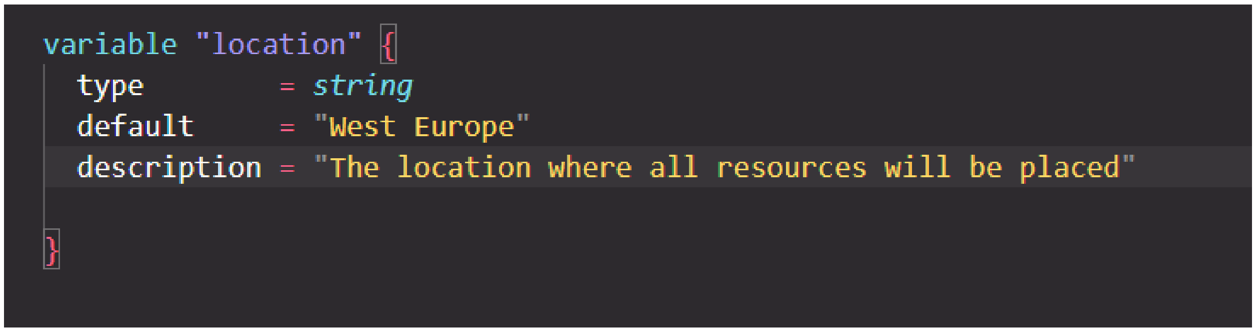
Name	Type	Kind	Resource group	Location	Subscription
csb10032002887d7e2c	Storage account	StorageV2	cloud-shell-storage-westeurope	West Europe	Azure Pass - Sponsorship
lfg4x3cfsa	Storage account	StorageV2	lfg4x3cf-rg	West Europe	Azure Pass - Sponsorship



Name	Type	Resource group	Location	Subscription
csb10032002887d7e2c	Storage account	cloud-shell-storage-westeurope	West Europe	Azure Pass - Sponsorship
lfg4x3cfsa	Storage account	lfg4x3cf-rg	West Europe	Azure Pass - Sponsorship

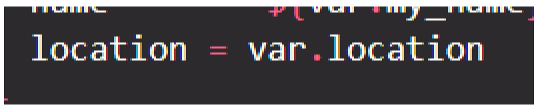
Task 3: Using variables and outputs

1.4. Define another variable of type string named “location” with default value of “West Europe” and description “The location where all resources will be placed.”. Use the definition of step 1.2 as reference and search in terraform documentation for defining a default value in a variable.



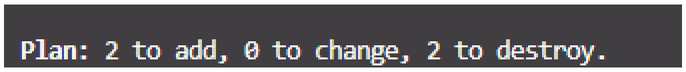
```
variable "location" {
  type        = string
  default     = "West Europe"
  description = "The location where all resources will be placed"

}
```

1.5. Reference the location variable as the location for the azurerm_resource_group resource. Use the example in step 1.2 of referencing the value but this time without the use of interpolation.



```
location = var.location
```

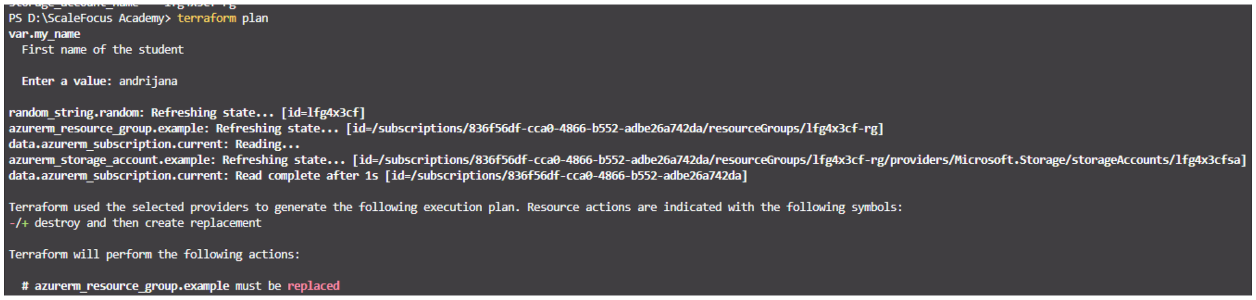
To allow automation we would need to have the non secret variables defined in a tfvars file named “inputs.tfvars” that you will create.



```
Plan: 2 to add, 0 to change, 2 to destroy.
```

Execute terraform plan command with the option –var-file=inputs.tfvars
The plan should try to destroy 2 resources and create 2 resources.

1.6. Execute terraform plan.
1.6.1.You are seeing that the code is asking you to insert an input value. Type your first name in lowercase and press enter.

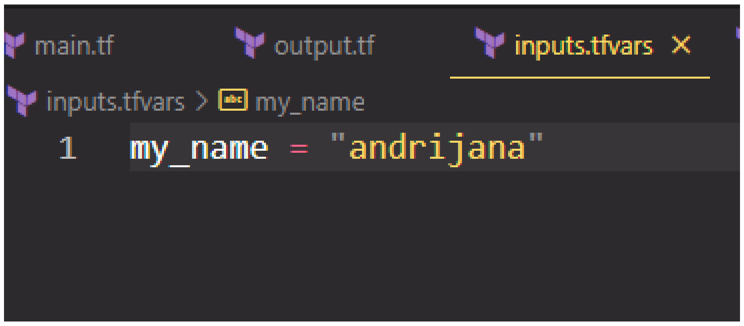


```
PS D:\ScaleFocus Academy> terraform plan
var.my_name
  First name of the student

  Enter a value: andrijana

random_string.random: Refreshing state... [id=lfg4x3cf]
azurerm_resource_group.example: Refreshing state... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg]
data.azurerm_subscription.current: Reading...
azurerm_storage_account.example: Refreshing state... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg/providers/Microsoft.Storage/storageAccounts/lfg4x3cfsa]
data.azurerm_subscription.current: Read complete after 1s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # azurerm_resource_group.example must be replaced
```

1.6.2.Please answer the following questions:
• How many variables do we have defined, and which are they?
 - We have created 2 variables, my_name and location.

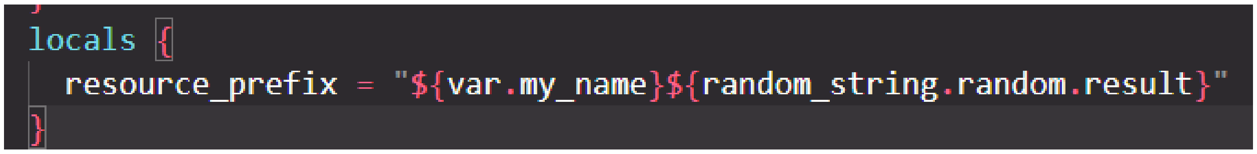
• Why did terraform asked us to input a value only for the my_name variable?
 - Because we have already declared a default value for the location variable.

1.7. Define input variable value in file
1.7.1.To allow automation we would need to have the non secret variables defined in a tfvars file named “inputs.tfvars” that you will create.
1.7.2.Define the value of the my_name variable inside the inputs.tfvars file like bellow:



```
my_name = "andrijana"
```

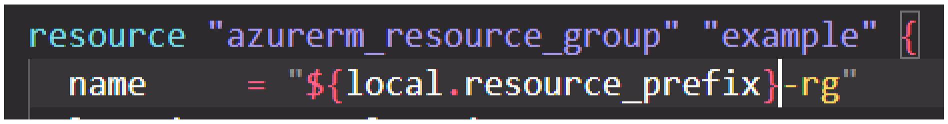
In main.tf before the data block create locals block where we will define a local value named resource_prefix where we will concatenate the input variable my_name with the generated value from the random string resource.



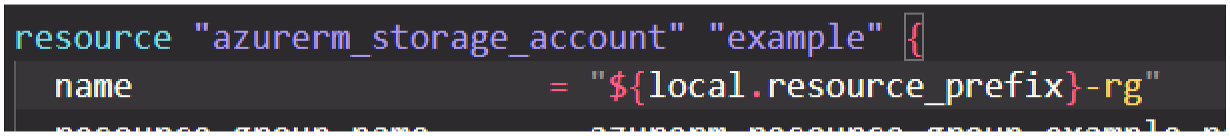
```
locals {
  resource_prefix = "${var.my_name}${random_string.random.result}"
}
```

1.7.3.Execute terraform plan command with the option –var-file=inputs.tfvars
1.7.4.The plan should try to destroy 2 resources and create 2 resources.

Add this resource_prefix as prefix of the name of the azurerm_resource_group and azurerm_storage_account resources.



```
resource "azurerm_resource_group" "example" {
  name     = "${local.resource_prefix}-rg"
```

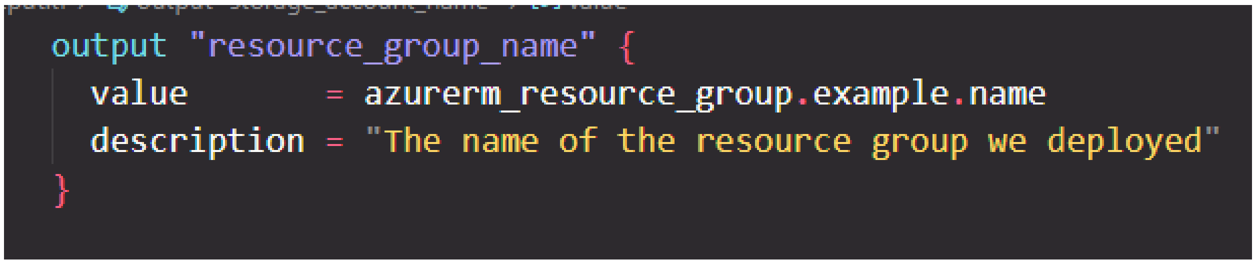


```
resource "azurerm_storage_account" "example" {
  name                     = "${local.resource_prefix}-rg"
```

2.3. Execute the terraform plan with the input variable file switch. It should show you again 2 resources for destroy and 2 resources to create.

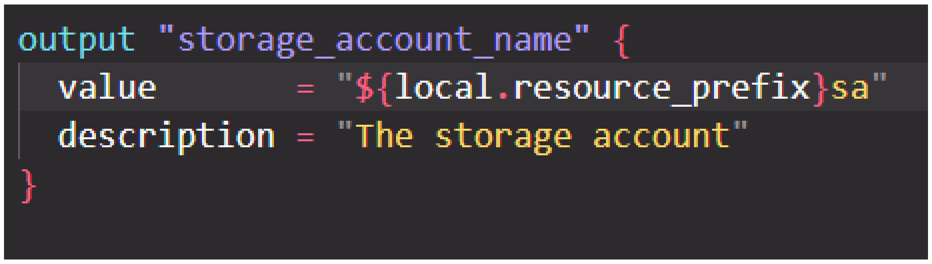
3. Using output values
3.1. To display some values from our resources we need to define the output values. For better visibility create an output.tf file where we will place all output values that we want to display after applying.

3.2. Inside the outputs.tf file define an output value named resource_group_name with the value of the name of the resource group that we create, like shown below:



```
output "resource_group_name" {
  value       = azurerm_resource_group.example.name
  description = "The name of the resource group we deployed"
}
```

3.3. Do the same for the output value named storage_account_name where the value will be the name of the storage account by using the example from step:



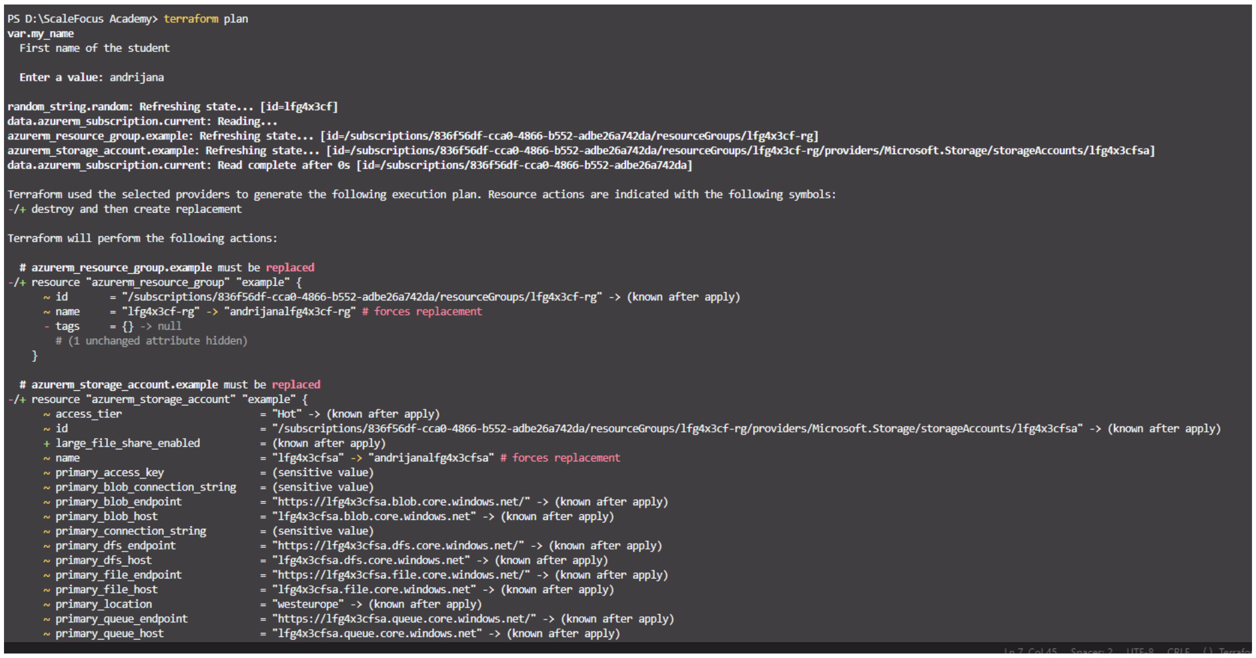
```
output "storage_account_name" {
  value       = "${local.resource_prefix}sa"
  description = "The storage account"
}
```

3.2 3.4. Execute the terraform plan with the input variable file switch. It should show you again 2 resources for destroy and 2 resources to create. You will also see at the bottom that there will be outputs.

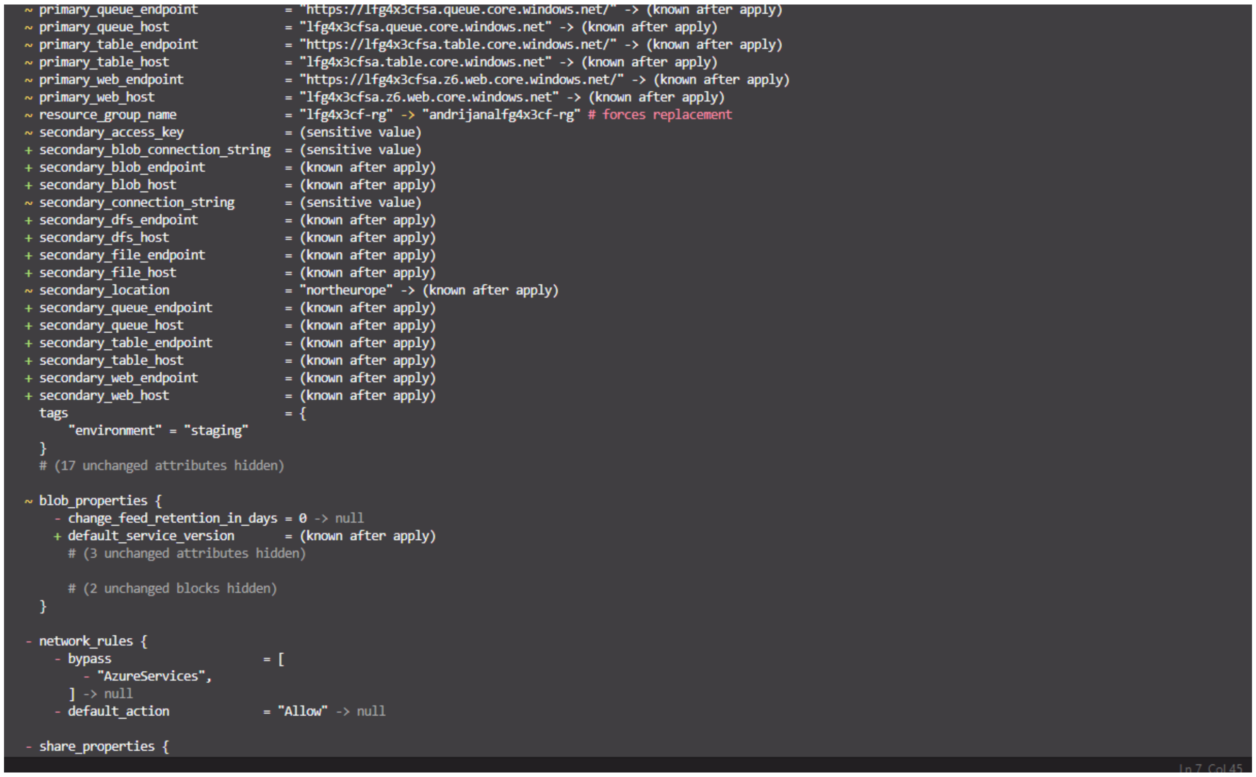
4. Understanding the reason why our resources are being destroyed

4.1. When you execute terraform plan it will give you information about the resources and parameters that are being created with “+”, destroyed and recreated with “-/+”, the ones destroyed with “–“ and the ones that will be modified with “~”.

4.2. In this task we will need to go over our terraform plan and identify the reasons why our resources are being replaced.



```
PS D:\ScaleFocus Academy> terraform plan
var.my_name
  First name of the student

  Enter a value: andrijana

random_string.random: Refreshing state... [id=lfg4x3cf]
data.azurerm_subscription.current: Reading...
azurerm_resource_group.example: Refreshing state... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg]
azurerm_storage_account.example: Refreshing state... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg/providers/Microsoft.Storage/storageAccounts/lfg4x3cfsa]
data.azurerm_subscription.current: Read complete after 0s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # azurerm_resource_group.example must be replaced
-/+ resource "azurerm_resource_group" "example" {
      ~ id       = "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg" -> (known after apply)
      ~ name     = "lfg4x3cf-rg" -> "andrijanalfg4x3cf-rg" # forces replacement
      - tags     = {} -> null
        # (1 unchanged attribute hidden)
    }

  # azurerm_storage_account.example must be replaced
-/+ resource "azurerm_storage_account" "example" {
      ~ access_tier                       = "Hot" -> (known after apply)
      ~ id                                = "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg/providers/Microsoft.Storage/storageAccounts/lfg4x3cfsa" -> (known after apply)
      + large_file_share_enabled          = (known after apply)
      ~ name                              = "lfg4x3cfsa" -> "andrijanalfg4x3cfsa" # forces replacement
      ~ primary_access_key                = (sensitive value)
      ~ primary_blob_connection_string    = (sensitive value)
      ~ primary_blob_endpoint             = "https://lfg4x3cfsa.blob.core.windows.net/" -> (known after apply)
      ~ primary_blob_host                 = "lfg4x3cfsa.blob.core.windows.net" -> (known after apply)
      ~ primary_connection_string         = (sensitive value)
      ~ primary_dfs_endpoint              = "https://lfg4x3cfsa.dfs.core.windows.net/" -> (known after apply)
      ~ primary_dfs_host                  = "lfg4x3cfsa.dfs.core.windows.net" -> (known after apply)
      ~ primary_file_endpoint             = "https://lfg4x3cfsa.file.core.windows.net/" -> (known after apply)
      ~ primary_file_host                 = "lfg4x3cfsa.file.core.windows.net" -> (known after apply)
      ~ primary_location                  = "westeurope" -> (known after apply)
      ~ primary_queue_endpoint            = "https://lfg4x3cfsa.queue.core.windows.net/" -> (known after apply)
      ~ primary_queue_host                = "lfg4x3cfsa.queue.core.windows.net" -> (known after apply)
```

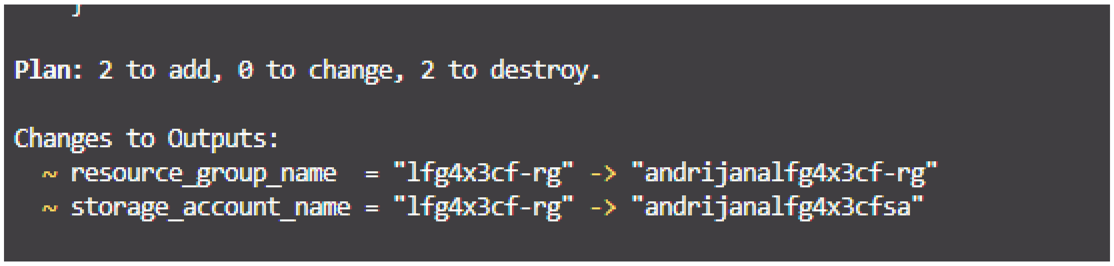


```
  ~ primary_queue_endpoint            = "https://lfg4x3cfsa.queue.core.windows.net/" -> (known after apply)
  ~ primary_queue_host                = "lfg4x3cfsa.queue.core.windows.net" -> (known after apply)
  ~ primary_table_endpoint            = "https://lfg4x3cfsa.table.core.windows.net/" -> (known after apply)
  ~ primary_table_host                = "lfg4x3cfsa.table.core.windows.net" -> (known after apply)
  ~ primary_web_endpoint              = "https://lfg4x3cfsa.z6.web.core.windows.net/" -> (known after apply)
  ~ primary_web_host                  = "lfg4x3cfsa.z6.web.core.windows.net" -> (known after apply)
  ~ resource_group_name               = "lfg4x3cf-rg" -> "andrijanalfg4x3cf-rg" # forces replacement
  ~ secondary_access_key              = (sensitive value)
  + secondary_blob_connection_string  = (sensitive value)
  + secondary_blob_endpoint           = (known after apply)
  + secondary_blob_host               = (known after apply)
  ~ secondary_connection_string       = (sensitive value)
  + secondary_dfs_endpoint            = (known after apply)
  + secondary_dfs_host                = (known after apply)
  + secondary_file_endpoint           = (known after apply)
  + secondary_file_host               = (known after apply)
  ~ secondary_location                = "northeurope" -> (known after apply)
  + secondary_queue_endpoint          = (known after apply)
  + secondary_queue_host              = (known after apply)
  + secondary_table_endpoint          = (known after apply)
  + secondary_table_host              = (known after apply)
  + secondary_web_endpoint            = (known after apply)
  + secondary_web_host                = (known after apply)
    tags                              = {
        "environment" = "staging"
    }
    # (17 unchanged attributes hidden)

  ~ blob_properties {
      - change_feed_retention_in_days = 0 -> null
      + default_service_version       = (known after apply)
        # (3 unchanged attributes hidden)

        # (2 unchanged blocks hidden)
    }

  - network_rules {
      - bypass                     = [
          - "AzureServices",
        ] -> null
      - default_action             = "Allow" -> null

  - share_properties {
```

4.2.1.Search for the term “forces replacement” and node the resource name and the parameter that forces replacement. Describe the reason behind it

- We can see in 3 places the term ***forces replacement***.

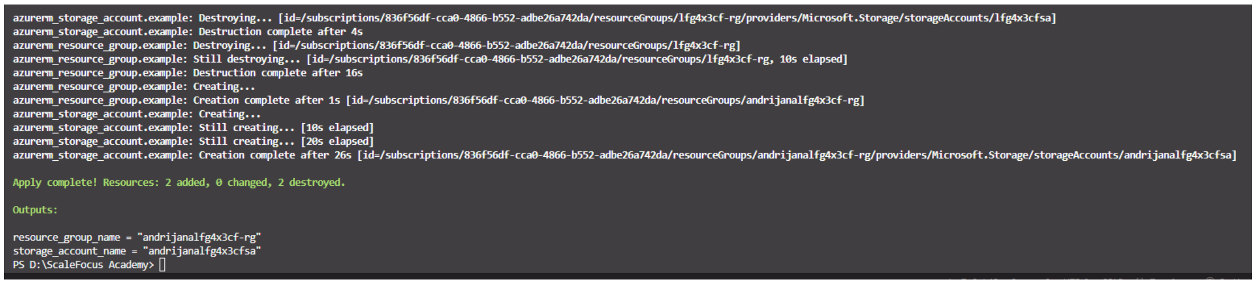
5. Apply the terraform code and write down the outputs



```
    }

Plan: 2 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  ~ resource_group_name  = "lfg4x3cf-rg" -> "andrijanalfg4x3cf-rg"
  ~ storage_account_name = "lfg4x3cf-rg" -> "andrijanalfg4x3cfsa"
```

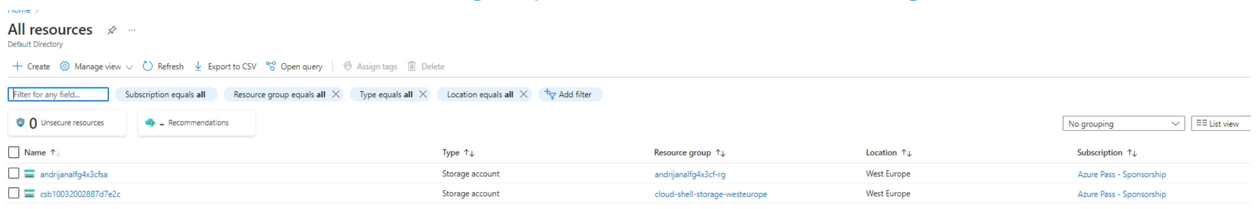
- I’ve been changing the variable values and the code, experimenting a bit to see the output (and I realized I forgot to change the **-rg suffix** to **sa** for the storage account output- which I later fixed, as you can see)

- Here is the final output:



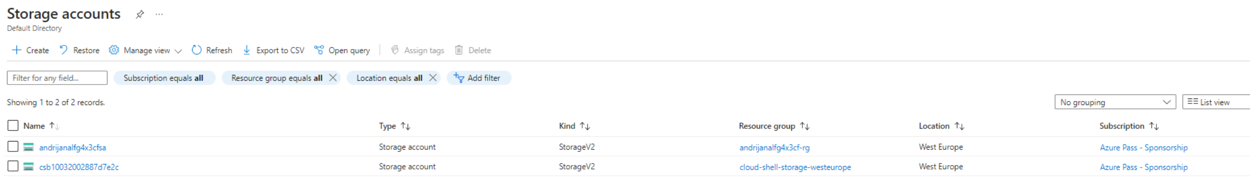
```
azurerm_storage_account.example: Destroying... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg/providers/Microsoft.Storage/storageAccounts/lfg4x3cfsa]
azurerm_storage_account.example: Destruction complete after 4s
azurerm_resource_group.example: Destroying... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg]
azurerm_resource_group.example: Still destroying... [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/lfg4x3cf-rg, 10s elapsed]
azurerm_resource_group.example: Destruction complete after 16s
azurerm_resource_group.example: Creating...
azurerm_resource_group.example: Creation complete after 1s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/andrijanalfg4x3cf-rg]
azurerm_storage_account.example: Creating...
azurerm_storage_account.example: Still creating... [10s elapsed]
azurerm_storage_account.example: Still creating... [20s elapsed]
azurerm_storage_account.example: Creation complete after 26s [id=/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/andrijanalfg4x3cf-rg/providers/Microsoft.Storage/storageAccounts/andrijanalfg4x3cfsa]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.

Outputs:

resource_group_name = "andrijanalfg4x3cf-rg"
storage_account_name = "andrijanalfg4x3cfsa"
PS D:\ScaleFocus Academy> 
```

- The name of the resource group in Azure resources changed as well:



Name	Type	Resource group	Location	Subscription
andrijanalfg4x3cfsa	Storage account	andrijanalfg4x3cf-rg	West Europe	Azure Pass - Sponsorship
csb10032002887d7e2c	Storage account	cloud-shell-storage-westeurope	West Europe	Azure Pass - Sponsorship

- And the name of the storage account:



Showing 1 to 2 of 2 records.

Name	Type	Kind	Resource group	Location	Subscription
andrijanalfg4x3cfsa	Storage account	StorageV2	andrijanalfg4x3cf-rg	West Europe	Azure Pass - Sponsorship
csb10032002887d7e2c	Storage account	StorageV2	cloud-shell-storage-westeurope	West Europe	Azure Pass - Sponsorship