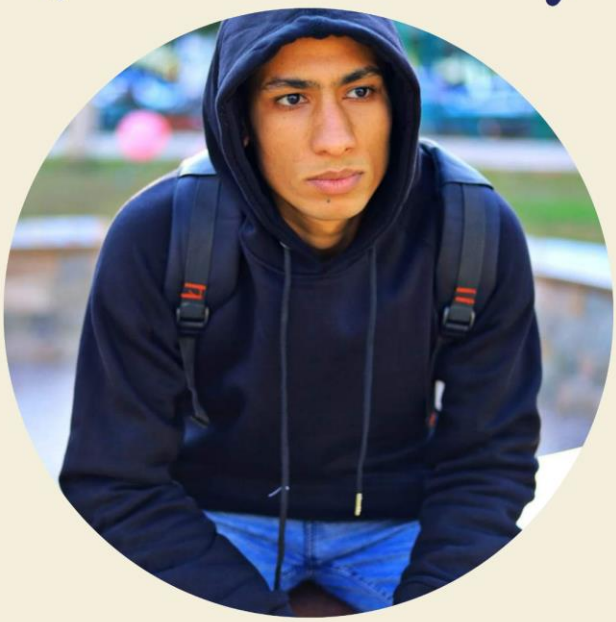
TERRAFORM

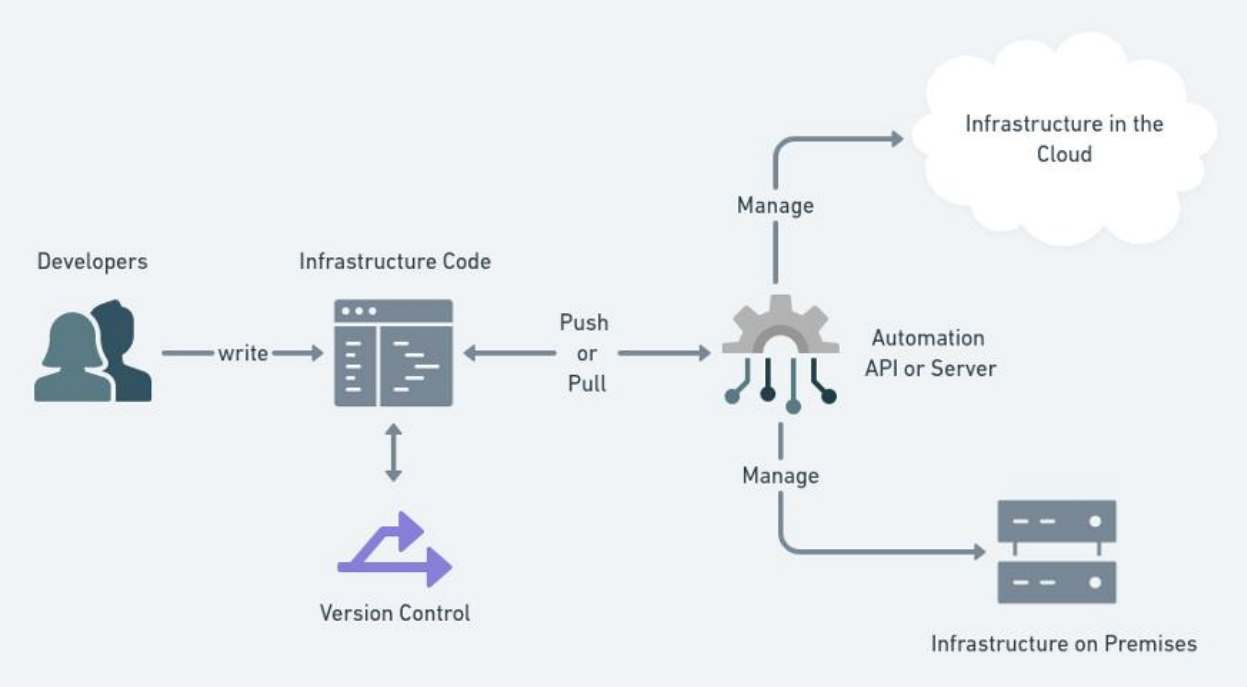


By:Mostafa Mahmoud Bahgat

LinkedIn:https://www.linkedin.com/in/mostafamahmoudbahgat

Terraform

Infrastructure-as-a code(IAC) : هو Software بيمد ال Resource بتاعتي سواء كانت on-perm or Cloud بمعني تاني اني اقدر ابني واتتحكم في Infra كامله من خلال ال Coding



الأنواع :

Scripting-1 : وسيلة بسيطة جدا بعمل من خلالها Task او مجموعات Tasks بسيطة

Configuration Management Tools-2 : هس أدوات بتعمل Implement بشكل Automation وConfig install and ال Servers وهو مصمم لل Tasks المعقدة

Provisioning Tools-3 : هي ال بتعمل createg لل Infra من البداية

Containers and Templating Tools-4 : مثل ال Docker – K8s بتقدر تعمل Formulate لل Templates من خلال image

---

في طريقتين واحنا بنعمل Automating Infrastructure

Declarative(=Function)-1 : تقدر ت Define ال Desired State عشان أوصل ل State معينه والطريقة دي الأكثر شيوعا

مثل(K8s-Docker-Terraform)

Imperative-2 : بتحدد الخطوات Step By Step وبتعمل execute عشان ت Reach ال Desired Solution ال عاوز توصله

مثل (Ansible-Chef-Puppet)

Terraform : هي احد ادوت ال Infra as a code بتسمحلك انك تعمل Define سواء ع ال cloud او on-prem في ال resources بتاعتك عن طريق Config File – بتسخدم لغة HashiCorp Config Language(HCL) - هي Declarative Language

وكمان ميزتها انها بتتعامل مع كل ال infra Providers علي عكس ال CloudFormation الخاصة ب AWS فقط

ازاي بعمل Deploy من خلال ال terraform :

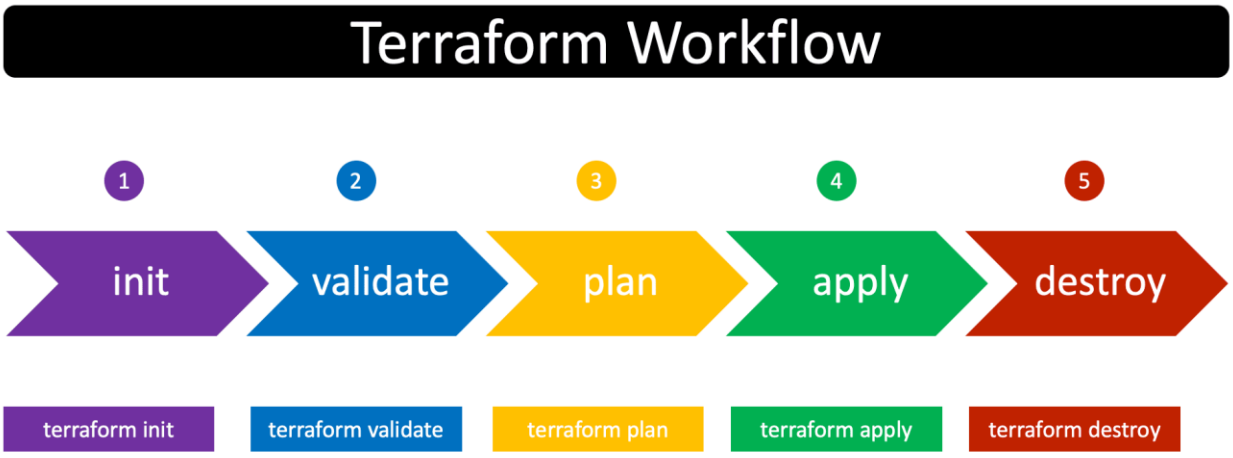
scop-1 بشوف أي ال انا محتاجة وابدا احفظه

Author-2 بكتب الحاجة ال حفظتها دي ال عاوز انشاءها في config File

Initialize-3 بعمل Install لل Plugins ال هتحتاجها الحاجة ال عاوز انشاءها

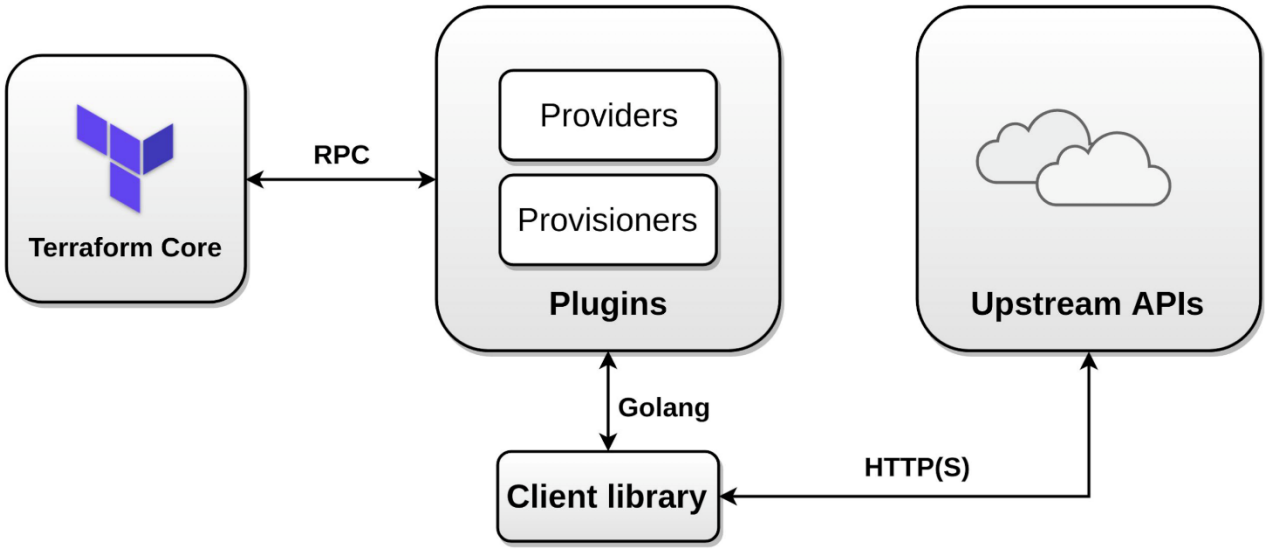
Plan-4 بيعمل Preview لكل حاجه انا هعملها

Apply-5 هنا هينفذ كل حاجه انا كتبها



ودول هيتم شرحهم بالتفاصيل

:Terraform Architecture



- Terraform Core :

مسؤولة عن ال code ال انت بتكتبه هي ال بتقراه يعني – مسؤولة عن ال State Management – مسؤولة عن ال Resource Graph – مسؤولة عن ال Plan – بتعمل Communication مع ال Plugins عن طريق ال RPC(remote procedure calls)

عشان تاخد ال Plugins ال محتاجها

- Terraform Plugins :

مسؤولة عن عملية ال Calls بين ال Terraform و ال Cloud API

مسؤولة عن عملية ال Authentication مع ال Providers

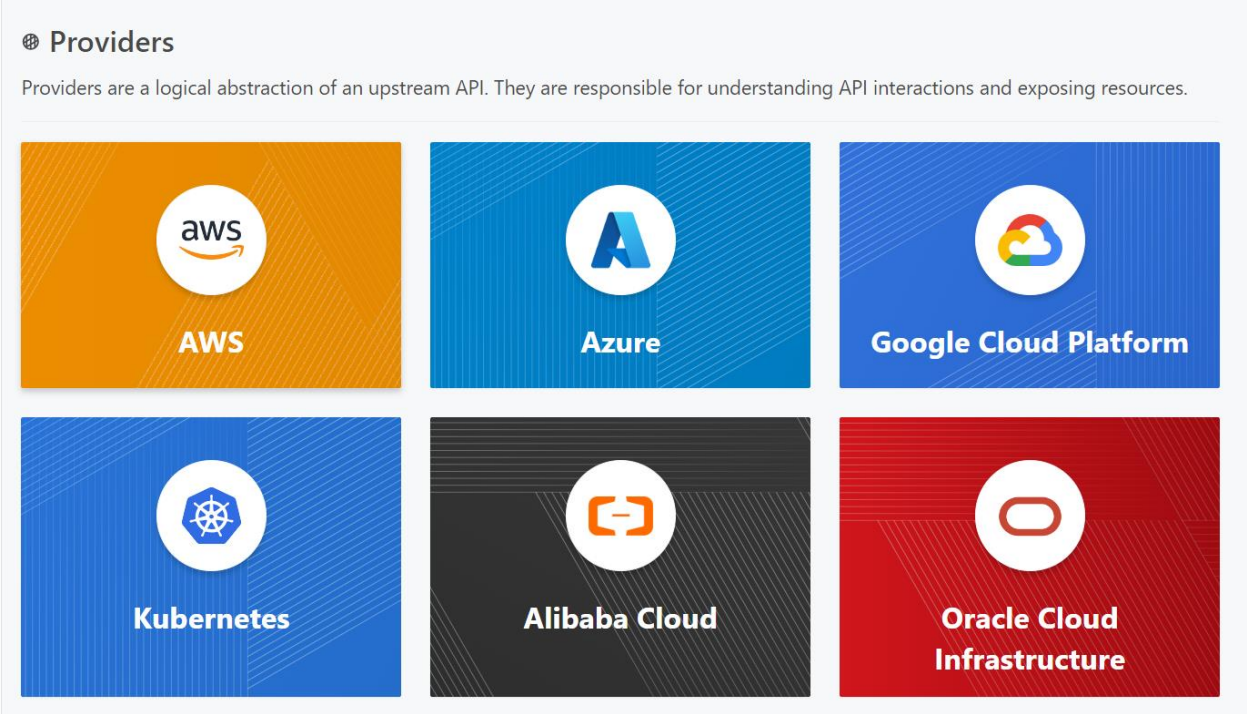
Install Terrform : علي حسب ال OS ال حابب تعمل عليه Install ودا الموقع الرسمي لل Terraform هتختار ال OS وهيوضحلك ال Steps ازاي تعمل Install

https://developer.hashicorp.com/terraform/install

: Terraform Concepts

عشان ابدا اشتغل علي ال Terraform بعمل Directory ثم بعمل Files بتكون ال Extension بتاعتها (.tf)

: Providers -1



هي اول خطوة لان هنا بحددله ال Provider ال هبدا اشتغل عليه زي AWS ولا Azure ولا Google Cloud وهكذا

وهنا هبدا اتعامل مع Azure Cloud :

ال Code بيكون بالطريقة دي :

```
[root@terraform ~]# cd tr_project/
```

اول خطوه عملت directory اسمه tr_project

```
[root@terraform tr_project]# vim tf_provider.tf
```

تاني خطوة عملت file اسمه provider اسمه tf_provider.tf (لازم يكون tf.)

وداخل ال file هبدا اكتب ال code الخاص بال provider

```
#Providers

terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "=3.0.0"
    }
  }
}
```

هنا بحددله اني هبدا اتعامل مع Azure Cloud

ال Version ال ف ال code هي ال Version الخاصة بال Plugin الخاصة ب azure

لو حابب تشتغل علي Provider تاني هتدخل علي اللينك داه تختار ال provider وتشوف ال code الخاص بيه

https://registry.terraform.io/browse/providers

طيب انا كدا حددله ال Cloud Providers ال هتعامل معه كدا فاضل حاجه مهمه وهي ال Authentication مع الاكونت بتاعي ال هبدا اعمل create عليه

وفي اكتر من طريقة عشان تعمل Authentication مع Azure :

Azure CLI-

Managed Identity-

Service Principal and a Client Certificate-

Service Principal and a Client Secret-

Service Principal and OpenID Connect-

AKS Workload Identity-

هنستخدم ال Azure CLI : هعمل install لل Azure Cli علي ال Host بتاعي

```
[root@terraform ~]# sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

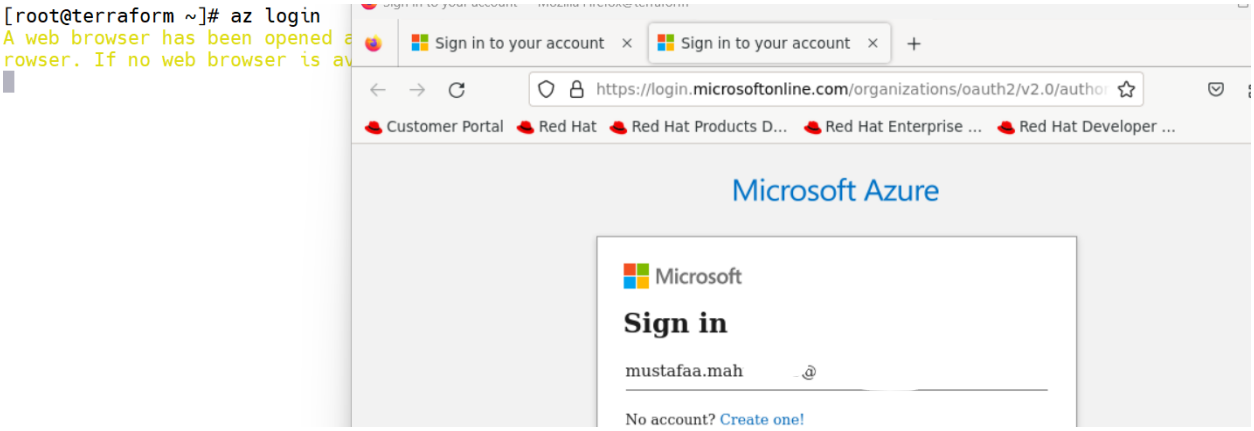
بعمل Import لل Packages

```
[root@terraform ~]# sudo tee /etc/yum.repos.d/azure-cli.repo <<EOF
[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
EOF
[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
```

تاني حاجه بعمل ال Repo الخاص بيها

```
[root@terraform ~]# sudo dnf install -y azure-cli
az --version
Updating Subscription Management repositories.
Azure CLI
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Last metadata expiration check: 0:00:09 ago on Fri 31 May 2024 03:34:35 PM EEST.
Dependencies resolved.
================================================================================
 Package                         Architecture                    Version
================================================================================
Installing:
 azure-cli                       x86_64                          2.38.1-1.el7
```

تالت حاجه بعمل install لل azure cli



رابع حاجه بعمل login علي الاكونت بكتب command وهو az login بيفتحلي browser اعمل login

You have logged into Microsoft Azure!

You can close this window, or we will redirect you to the Azure CLI documents in 10 seconds.

بيعرضلي انه عمل login ولو فيه error بيظهرلك

---

init-2 : بعد م اكتب ال انا محتاجة ف Infra بعمل Run للكود دا ووظيفته انه بيعمل Initialize لل Working Directory

هبدا اعمل Create لل resource group علي Azure

```
resource "azurerm_resource_group" "TF-RG" {
  name     = "TF-RG"
  location = "West Europe"
}
```

Resource عشان ابد اعرف terraform اني عاوز اعمل resource وهنا ببدا اعمل كل ال Infra ال انا محتاجها سوا network-vm-firewall أي resource بيكون في ال section ال اسمه resource

"azurerm_resource_group" دا اسم ال plugin الخاصة ب ال resource group عند azure

Name : دا اسم ال resource group ال بعمله

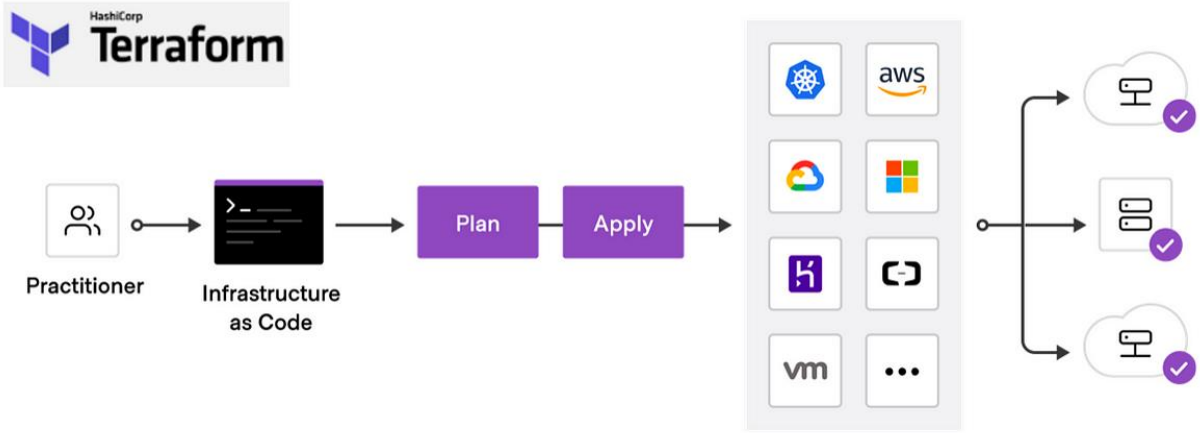
Location : دا ال location ال هيتعمل فيه create لل resource group بتاعي

بعد كدا ببدا انفذ ال command الخاص بال init

```
[root@terraform tr_project]# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "3.0.0"...
- Installing hashicorp/azurerm v3.0.0...
- Installed hashicorp/azurerm v3.0.0 (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.
[root@terraform tr_project]#
```

بيبدا يعمل Initialize ويعمل download لل plugin الخاصة بال Azure resource group

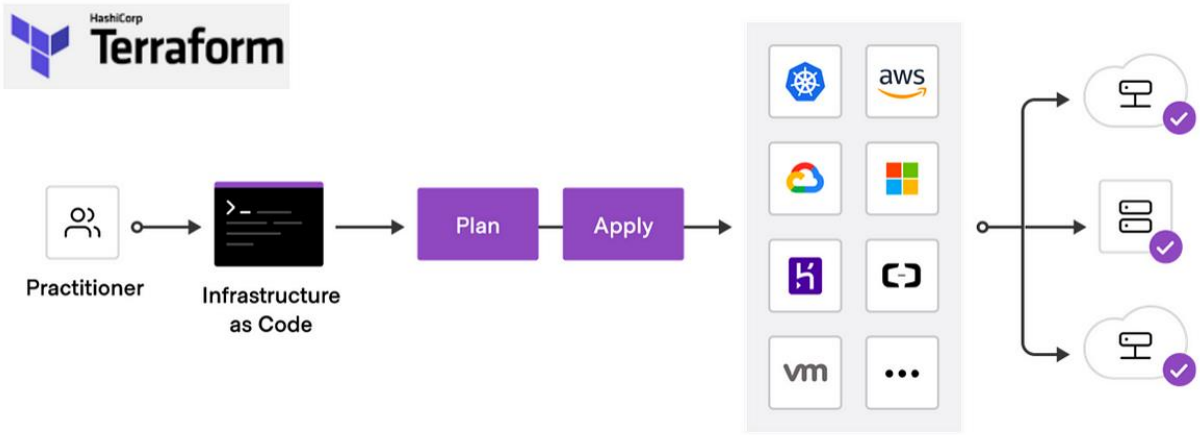
Plan-3 : بيعمل Preview لكل حاجه انا هعملها بناءا علي ال code ال انا كتبته




```
[root@terraform tr_project]# terraform plan

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

Plan: 1 to add, 0 to change, 0 to destroy.
```

اخر line بيقولك انه فيه plan هتحصل وشارحلك أي ال هيتم وهو انه هيعمل create لل RG وباقي التفاصيل

Apply-4 : بينفذلي كل ال انا كتبته ف ال code




```
[root@terraform tr_project]# terraform apply

Terraform used the selected providers to generate the
following symbols:
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

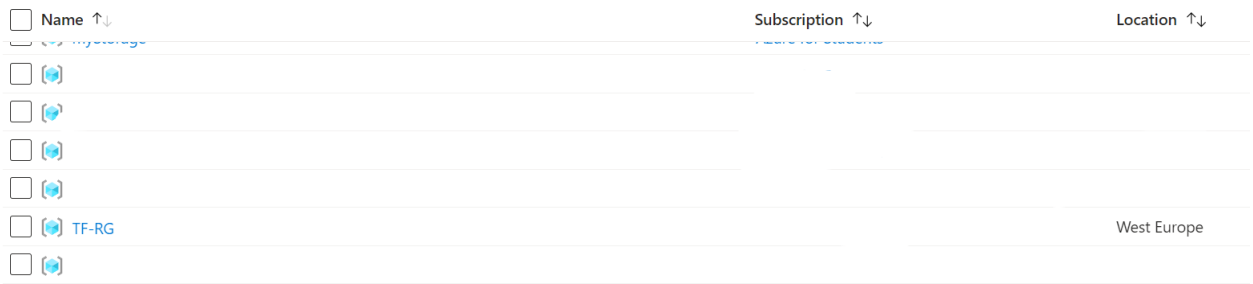
في البداية بيعرضلي نفس محتويات ال Plan لكن مع اختلاف انه بيسالك خلاص عاوز تنفذ ال code دا

--

```
[root@terraform tr_project]# terraform apply

Terraform used the selected providers to generate the followi
following symbols:
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

azurerm_resource_group.TF-RG: Creating...
azurerm_resource_group.TF-RG: Creation complete after 2s [id=
roups/TF-RG]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[root@terraform tr_project]#
```

ف لما أقوله Yes خلاص عاوز انفذ ال code بتاعي بيبدا ينفذ ال code وهنا هيبدا يعملي RG علي ال Azure

--



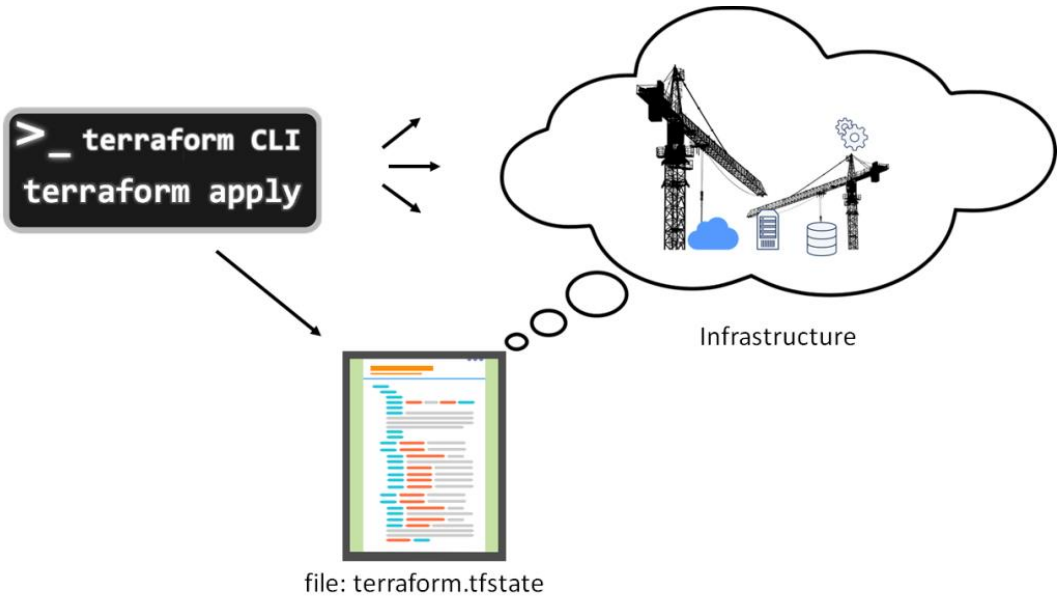
فتحت ال Account بتاعي ف لقيت ان خلاص ال RG بتاعي اتعمله Create

---

لو عملت Apply تاني بعد كدا من غير معمل أي تعديل علي ال code بتاعي مش بيديني أي Action فقط بيعرفني ان ال Recourse ال بحاول اعمله create هو موجود ومفيش أي تغيير حصل علي ال code عشان ينفذها

```
[root@terraform tr_project]# terraform apply
azurerm_resource_group.TF-RG: Refreshing state... [id=/subs
F-RG]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against you
needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
[root@terraform tr_project]#
```

ال State File :



دا file بيتعمله create بشكل Automatic بعد اول Apply

```
[root@terraform tr_project]# ls
                    terraform.tfstate
[root@terraform tr_project]#
```

File دا بيكون فيه كل التفاصيل بتاع ال Resources ال تم انشائها بالتفاصيل.

```
{
  "version": 4,
  "terraform_version": "1.8.4",
  "serial": 2,
  "lineage": "6376ace9-8f68-4b1
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "azurerm_resource
      "name": "TF-RG",
      "provider": "provider[\"r
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "id": "/subscriptio
            "location": "westeu
            "name": "TF-RG",
            "tags": {},
            "timeouts": null
          },
```

في كل مره بعد كدا تعمل فيها Apply بيبدأ يقارن بين ال Code بتاعك وبين ال State File دا هل في تغيير حصل في ال code مش موجود في ال State file ف هيبدا ينفذ التغيير دا بس لو مفيش أي تغيير حصل ف مش هينفذ حاجه

وبيقارن ال State file مع ال Account لان انا ممكن ادخل ع الاكونت واحذف ال RG ال عملتله Create ف ساعتها نفس ال Cdoe اه ونفس ال State file اه لكن مش موجود علي ال Account فهيبدا يعمله Create لانه مش موجود علي ال Account

State Management : هنتكلم هنا عن ال Backend and Lock

- في مشاكل ممكن تقابليني زي مثلا ان ال State File يتحذف ف دي مشكله

او ان او ان يكون في اكتر من عملية علي ال State File

ف ال Backend هو ال state file بتاعي لكن مش هيكون local هخزنه في مكان تاني وليكن علي ال Azure Storage او ال S3 علي AWS

وال Backend ليها Section لوحدها

وليكن هبدا اخزن ال State file علي ال Azure Storage

Name	Type	Kind	Resource group	Location
backend0	Storage account	StorageV2	Backend	East US

ف عندي Account Storage اسمه backend0 و RG اسمه Backend وعملت Create ل container اسمه terraform-state

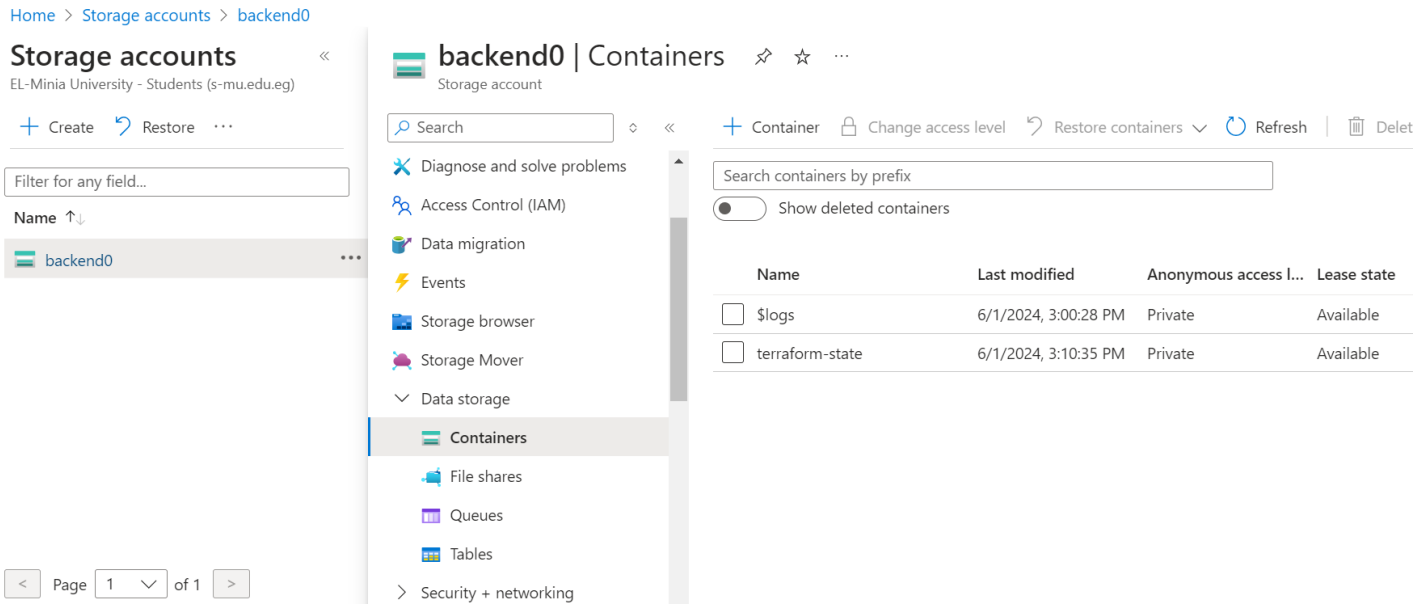
ودي المعلومات اللازمه عشان ابدا اعمل terraform backend وال Code بيكون بالشكل دا :

```
terraform {
  backend "azurerm" {
    resource_group_name  = "Backend"
    storage_account_name = "backend0"
    container_name       = "terraform-state"
    key                  = "terraform.tfstate"
  }
}
```

وال Key : دا اسم ال File ال هيكون في ال Container ف سميته ب نفس الاسم ال Default

بعد م يتم انشاء ال file علي ال cloud بيبدا تلقائيا بيسمح ال file ال local ويعمل واحد Backup فقط مش بيستخدمه في ال Run

```
[root@terraform tr_project]# ls
terraform.tfstate.backup
[root@terraform tr_project]#
```



دا ال storage وال container بتاعي

--

```
[root@terraform tr_project]# terraform init

Initializing the backend...
```

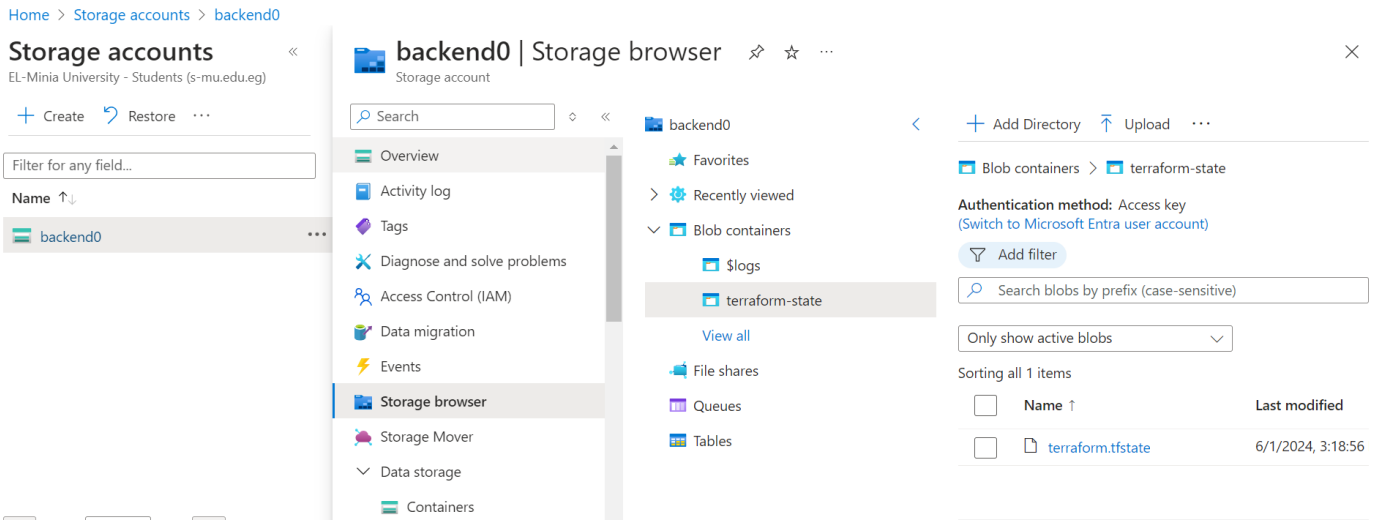
بيدا اعمل init واي تغيير بعد كدا لازم تعمل init

--

```
[root@terraform tr_project]# terraform init

Initializing the backend...
Acquiring state lock. This may take a few moments...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "azurerm" backend. No existing state was found in the newly
  configured "azurerm" backend. Do you want to copy this state to the new "azurerm"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes

Releasing state lock. This may take a few moments...

Successfully configured the backend "azurerm"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Reusing previous version of hashicorp/azurerm from the dependency lock file
- Using previously-installed hashicorp/azurerm v3.0.0

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.
[root@terraform tr_project]#
```

بيسالك انه هيخزن ال state file دا علي Azure Storage

--



هنا خلاص عمل فعلا ال file وبنفس ال name ال كان في ال code

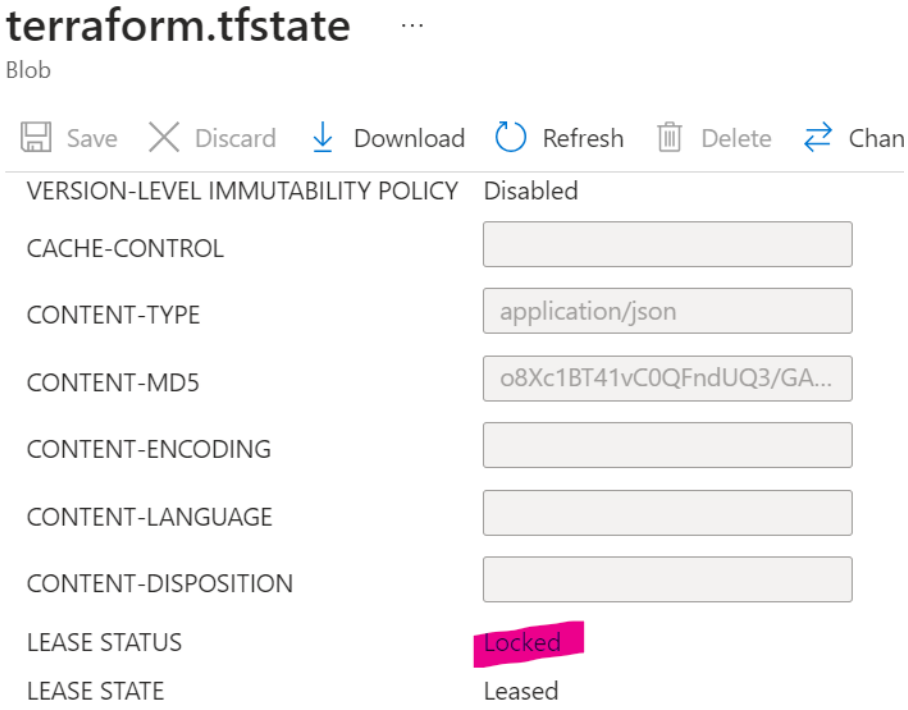
State Lock : دي ميزتها ان ال State File مينفعش يتعمله Run غير مرة واحدة في نفس الوقت يعني لو حد بيعمل Run لل State file دا وشخص تاني هيعمل Run لنفس ال State file في نفس الوقت ف هيطلعله error

وال Lock مع Azure موجود default مش محتاج تعمل حاجه عشان تفعله مثلا

```
[root@terraform tr_project]# terraform apply
Acquiring state lock. This may take a few moments...
azurerm_resource_group.TF-RG: Refreshing state... [id=
F-RG]

Terraform used the selected providers to generate the
following symbols:
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

طيب هنا بعمل Apply (run state file)

--



ف لو فتحت ال Container وشوفت ال File (state file)

هتلاقيه انها بقي Locked لان في عمليه Apply عليه دلوقت

--

```
[root@terraform tr_project]# terraform apply
Acquiring state lock. This may take a few moments...

  Error: Error acquiring the state lock

  Error message: state blob is already locked
  Lock Info:
    ID:        50c5d6bf-aab2-b533-d7b4-6d808e09380f
    Path:      terraform-state/terraform.tfstate
    Operation: OperationTypeApply
    Who:       root@terraform
    Version:   1.8.4
    Created:   2024-06-01 13:09:36.834742437 +0000 UTC
    Info:


  Terraform acquires a state lock to protect the state from being written
  by multiple users at the same time. Please resolve the issue above and try
  again. For most commands, you can disable locking with the "-lock=false"
  flag, but this is not recommended.

[root@terraform tr_project]#
```

ف لو شخص تاني عمل Apply في نفس الوقت هيطلعله Error

```
[root@terraform tr_project]# terraform fmt
backend.tf
[root@terraform tr_project]#
```

الامر داب يعمل format لل code بتاعي وينسقه بالطريقة المتعارف عليها

```
[root@terraform tr_project]# terraform destroy
```

الامر دا هعيمل delete لكل ال infra الي انا عملتها عندي

Terraform variable : عشان اقدر اعمل var في اكتر من طريقة

:1

-هعمل file اسمه variable.tf وهضع فيه المتغييرات

```
variable "location" {
  type = string
}
```

وهنا بعمل var خاص بال location

--

-وفي ال resource هعمل Reference لل Variables لل location بالطريقة دي:

```
resource "azurerm_resource_group" "TF-RG" {
  name     = "TF-RG"
  location = var.location
}
```

--

-بعد كدا هعمل apply

```
[root@terraform tr_project]# terraform apply
Acquiring state lock. This may take a few moments...
var.location
  Enter a value:
```

لما اعمل apply هيطلب مني ال value بتاع ال var ال اسمه location

والطريقة دي مش افضل حاجه لان ممكن يكون عندي اكتر من var

:2

-هعمل 2file

```
[root@terraform tr_project]# vim variables.tf
```

ودا هضع فيه المتغيرات بتاعتي

```
variable "location" {
  type        = string
  description = "The location where resources will be deployed"
}
```

--

والتاني هيكون .tfvars

```
[root@terraform tr_project]# vim variables.tf
```

هنا هدي قيمه للمتغير

```
location = "West Europe"
```

--

بعد كدا لما اعمل apply هتكون بالطريقة دي

```
[root@terraform tr_project]# terraform apply -var-file="prod.tfvars"
Acquiring state lock. This may take a few moments...
azurerm_resource_group.TF-RG: Refreshing state... [id=/subscriptions/7

Terraform used the selected providers to generate the following execut
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

اني بقوله وانت بتعمل apply هتستخدم file اسمه prod.tfvars ودا فيه قيمة المتغير بتاعي فمش هيسالك عن قيمة المتغير لانها موجودة في ال file ال انت كتبهوله .

--

الطريقة الثالثة

انك بتعملي file

```
[root@terraform tr_project]# vim variables.tf
```

وتكتب فيه المتغيرات كلها بالقيمة بتاعتها

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

\

بعد كدا بتعملي apply

```
[root@terraform tr_project]# terraform apply
Acquiring state lock. This may take a few moments...
azurerm_resource_group.TF-RG: Refreshing state... [id=/subscri

Terraform used the selected providers to generate the followin
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

مش بيسالك علي أي قيمة او انك بتكتبله file ال فيه المتغيرات لانك كاتب المتغير بالقيمة بتاعته في ال file

---

Terraform Workspaces : هي اني بعمل اكتر من environments واقوله خلي ال Management بتاع ال State File منفصل

بمعني اني كل workspace بيكون ليه state file لوحده وليكن عندي workspace 2 واحده اسمها prod وواحده اسمها dev فلو انا داخل ال workspace ال اسمها dev في هستخدم ال statefile بتاع ال workspace دي ، طيب لو عاوز استخدم ال statefile بتاع ال pord يبقي هختار الأول اني ادخل في ال workspace ال prod عشان استخدم ال state بتاعها

--

```
[root@terraform tr_project]# terraform workspace new prod
Created and switched to workspace "prod"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
[root@terraform tr_project]#
```

بعمل create ل workspace اسمها prod

```
[root@terraform tr_project]# terraform workspace list
  default
* dev
  prod

[root@terraform tr_project]#
```

بعمل list لكل ال workspace الي عندي وكمان بيوضحلي انا واقف علي انهي workspace

--

```
[root@terraform tr_project]# terraform workspace select prod
Switched to workspace "prod".
[root@terraform tr_project]#
```

هنا بقوله اعملي switch من ال workspace ال انت واقف عليها لل workspace ال اسمها prod

---

Terraform output : هي attributes لل input الي كتبته في ال Resource

عشان اقدر اعمله

```
[root@terraform tr_project]# vim output.tf
```

-

هعمل file

--

```
output "resource_group_id" {
  description = "The ID of the resource group"

  value       = azurerm_resource_group.TF-RG-out.id

}
```

هكتب ال code الخاص ب ال output

اول حاجه بكتب output بعد كدا بكتب اسم ال resource بتاعي وهنا انا بستخدم resource group فكتبت اسم ال resource دي

بعد كدا لو حابب اكتب description

بعد كدا بكتب ال value وبتساوي اسم ال resource ال بستخدمه + الي الاسم الي انا مسميه بيه في ال file ال اسمه rescoure.tf ودا ال code بتاعه عشان توضح اكتر

```
resource "azurerm_resource_group" "TF-RG-out" {
  name     = "TF-RG-out"
  location = var.location
}
```

```
[root@terraform tr_project]# terraform output
resource_group_id = "/subscriptions/2f49d029-4                    resourceGroups/TF-RG-out"
[root@terraform tr_project]#
```

بعمل list بكل ال output ال عندي وهنا مش عندي غير واحد فقط .

--

```
[root@terraform tr_project]# terraform output resource_group_id
"/subscriptions/2f49d029-42e5-4                    /resourceGroups/TF-RG-out"
[root@terraform tr_project]#
```

هنا بحددله انا عاوز اعمل list لل output دا

Terraform Provisioners : بعمل بيها Run ل Commend معين سواء local او remote بتكتب مع ال Resources بيتم تنفيذه مع اول Apply.

```
resource "azurerm_resource_group" "TF-RG-out" {
  name     = "TF-RG-out"
  location = var.location
provisioner "local-exec" {
    command = "echo ${self.id} > resource_group_id.txt"
  }
}
```

بحددله section ال provisioner في ال file الخاص بال resource

وهنا بقوله هتعملي command هتطبعلي ال id بتاع ال RG ال انا هعمله Create وتضع ال Result دي في file اسمه resource_group_id.txt

```
[root@terraform tr_project]# terraform apply
Acquiring state lock. This may take a few moments...

Terraform used the selected providers to generate the following execution plan. Resource actions ar
  + create

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + resource_group_id = (known after apply)

Do you want to perform these actions in workspace "prod"?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

azurerm_resource_group.TF-RG-out: Creating...
azurerm_resource_group.TF-RG-out: Provisioning with 'local-exec'...
azurerm_resource_group.TF-RG-out (local-exec): Executing: ["/bin/sh" "-c" "echo /subscriptions/2f49
 resource_group_id.txt"]
azurerm_resource_group.TF-RG-out: Creation complete after 2s [id=/subscriptions/                  d7
Releasing state lock. This may take a few moments...

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

resource_group_id = "/subscriptions/2f49d029-4.                    ./resourceGroups/TF-RG-out"
```

بعمل apply وفيها بوضحلي ال provisioning ال هو هيعملها

--

```
[root@terraform tr_project]# ls
resource_group_id.txt
```

بعد كدا هيعمل create لل file ال انا قولتله عليه

--

```
[root@terraform tr_project]# cat resource_group_id.txt
/subscriptions/2f49d029-42e5-                    /resourceGroups/TF-RG-out
[root@terraform tr_project]#
```

لو عملت cat لل file هلاقيه نفس ال Result بتاع ال id الخاص بال RG

```
[root@terraform tr_project]# terraform output
resource_group_id = "/subscriptions/2f49d029-42
[root@terraform tr_project]#
```

دا ال id ونفس ال result

By: Mostafa Mahmoud Bahgat

LinkedIn:https://www.linkedin.com/in/mostafamahmoudbahgat