Descargo Azure Cli

```
lobal options (use these before the subcommand, if any):
 -chdir=DIR    Switch to a different working directory before executing th
               given subcommand.
 -help         Show this help output, or the help for a specified subcomma
.
 -version      An alias for the "version" subcommand.
S C:\Users\palonso> Invoke-WebRequest -Uri https://aka.ms/installazurecliw
dows -OutFile .\AzureCLI.msi; Start-Process msiexec.exe -Wait -ArgumentLis
'/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

Configurar los permisos de la cuenta

az login

```
PS C:\Users\palonso> az login
A web browser has been opened at https://login.microsoftonline.com,
web browser is available or if the web browser fails to open, use
[
  {
    "cloudName": "AzureCloud",
    "user": {
      "name": "palonsosuela@gmail.com",
      "type": "user"
    }
  }
]
```

Dentro del json busco la columna que representa la subscripcion

Y establezco la cuenta para el CLI de azure.

az account set --subscription "id:"

Ahora crearemos una identidad del servicio para poder indicarle a terraform los token.

az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/<SUBSCRIPTION_ID>"

```
Creating 'Contributor' role assignment under scope '/subscriptions/a1494415-
3f3d-47df-b50a-7234b631bda2'
The output includes credentials that you must protect. Be sure that you do n
ot include these credentials in your code or check the credentials into your
 source control. For more information, see https://aka.ms/azadsp-cli
```

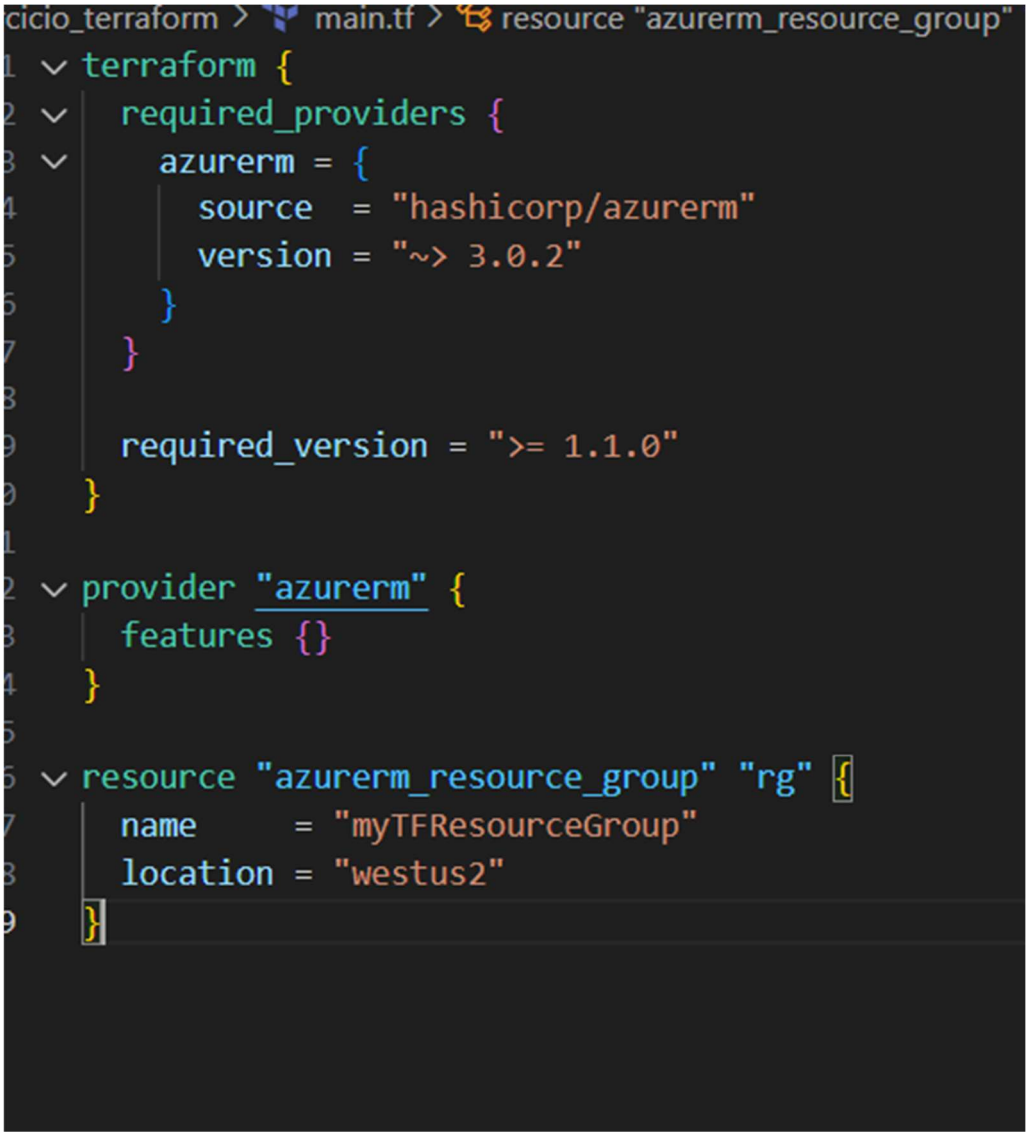
Actualizaremos con las variables de entorno

$ $Env:ARM_CLIENT_ID = "<APPID_VALUE>"

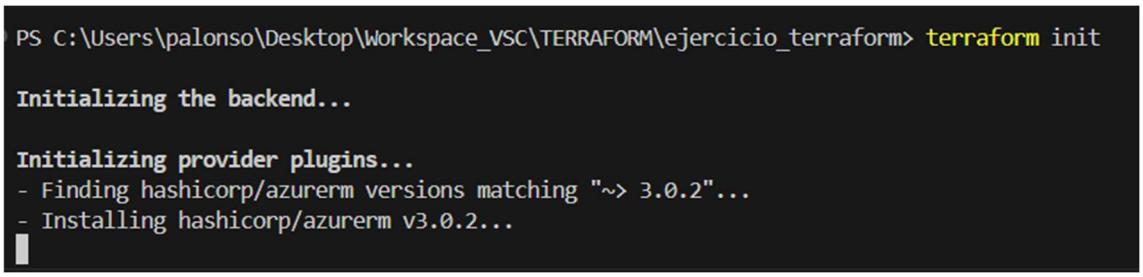
$ $Env:ARM_CLIENT_SECRET = "<PASSWORD_VALUE>"

$ $Env:ARM_SUBSCRIPTION_ID = "<SUBSCRIPTION_ID>"

$ $Env:ARM_TENANT_ID = "<TENANT_VALUE>"



Configuro en un main.tf el provider de azure e inicio terraform con terraform init:



```
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.
```

Formateamos la configuración en el caso de que hubiera algún error de formateo con:

terraform fmt

Y también comprobamos que sea válido

$ terraform validate

```
commands will detect it and remind you to do so if necessary.
PS C:\Users\palonso\Desktop\Workspace_VSC\TERRAFORM\ejercicio_terraform> terraform fmt
PS C:\Users\palonso\Desktop\Workspace_VSC\TERRAFORM\ejercicio_terraform> terraform validate
Success! The configuration is valid.

PS C:\Users\palonso\Desktop\Workspace_VSC\TERRAFORM\ejercicio_terraform>
```

Usamos terraform apply para aplicar la configuración:

```
PS C:\Users\palonso\Desktop\Workspace_VSC\TERRAFORM\ejercicio_terraform> terraform apply

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.rg will be created
  + resource "azurerm_resource_group" "rg" {
```

También tenemos estos comandos:

Ver el estado actual:
$ terraform show

Lista de recursos de terraform
$ terraform state list

Comandos disponibles para terraform:

$ terraform state