Download and add terraform into "environment variables"

| TEMP | C:\Users\USUARIO\AppData\Local\Temp |
| terraform | D:\Descargas\terraform\terraform.exe |
| TMP | C:\Users\USUARIO\AppData\Local\Temp |

```
C:\Users\USUARIO>terraform -v
Terraform v1.1.9
on windows_amd64

C:\Users\USUARIO>
```

Then check the aws cli

```
C:\Users\USUARIO>aws --version
aws-cli/2.5.8 Python/3.9.11 Windows/10 exe/AMD64 prompt/off
```

Then add the credentials form aws:

```
C:\Users\USUARIO>aws configure
AWS Access Key ID [None]: AKIATSHNFA7UTEA3Q4XE
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
```

Then créate the file

```
main.tf
 1    terraform {
 2        required_providers {
 3            aws = {
 4                source = "hashicorp/aws"
 5                version = "~> 3.5.0"
 6            }
 7        }
 8    }
 9
10    provider "aws" {
11        region = "us-east-1"
12    }
13
14    resource "aws_budgets_budget" "new-test" {
15        name               = "monthly-budget"
16        budget_type        = "COST"
17        limit_amount       = "500.0"
18        limit_unit         = "USD"
19        time_unit          = "MONTHLY"
20        time_period_start  = "2022-04-01_00:01"
21    }
22
```

Then launch with terraform init

```
D:\terraform-homework>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 3.5.0"...
- Installing hashicorp/aws v3.5.0...
- Installed hashicorp/aws v3.5.0 (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.

D:\terraform-homework>
```

Then check the info with: terraform validate

```
D:\terraform-homework>terraform fmt
main.tf

D:\terraform-homework>terraform validate
Success! The configuration is valid.


D:\terraform-homework>
```

Then créate the plan with: terraform plan

```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_budgets_budget.like-and-subscribe will be created
  + resource "aws_budgets_budget" "like-and-subscribe" {
      + account_id        = (known after apply)
      + budget_type       = "COST"
      + cost_filters      = (known after apply)
      + id                = (known after apply)
      + limit_amount      = "500.0"
      + limit_unit        = "USD"
      + name              = "monthly-budget"
      + name_prefix       = (known after apply)
      + time_period_end   = "2087-06-15_00:00"
      + time_period_start = "2020-09-01_00:01"
      + time_unit         = "MONTHLY"

      + cost_types {
          + include_credit              = (known after apply)
          + include_discount            = (known after apply)
          + include_other_subscription  = (known after apply)
          + include_recurring           = (known after apply)
          + include_refund              = (known after apply)
          + include_subscription        = (known after apply)
          + include_support             = (known after apply)
          + include_tax                 = (known after apply)
          + include_upfront             = (known after apply)
          + use_amortized               = (known after apply)
          + use_blended                 = (known after apply)
        }
    }

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

Then run the command: terraform apply

```
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

aws_budgets_budget.like-and-subscribe: Creating...
aws_budgets_budget.like-and-subscribe: Creation complete after 1s [id=245310490601:monthly-budget]

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