What are we going to see in this session?

- Providers
- Lets create out first terraform file
  - Provider
  - Resource
- Terraform file final look
- How to run the terraform file?
- Output plan to a file

Providers

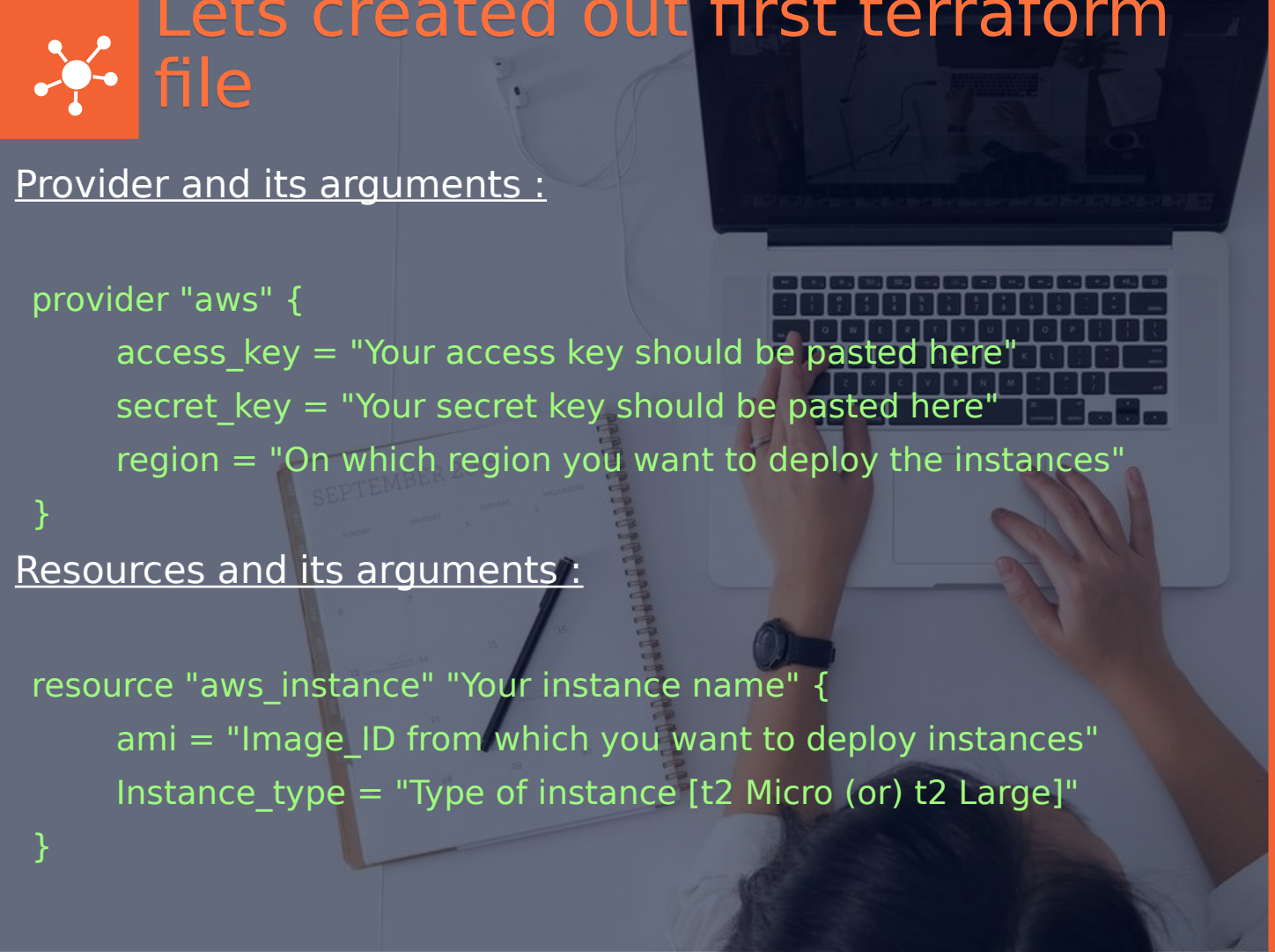
Providers are responsible to enable API communication to Infrastructure service providers.
Providers generally are

- Public cloud : AWS, GCP, Azure, Digital ocean etc..
- Private solutions : Open stack & VMware.

AWS

```
provider "aws" {
  access_key = “xxxxx”
  secret_key = “xxxxx”
  region = “xxxxx”
}
```

Lets created out first terraform file



Provider and its arguments :

```
provider "aws" {
        access_key = "Your access key should be pasted here"
        secret_key = "Your secret key should be pasted here"
        region = "On which region you want to deploy the instances"
}
```

Resources and its arguments :

```
resource "aws_instance" "Your instance name" {
        ami = "Image_ID from which you want to deploy instances"
        Instance_type = "Type of instance [t2 Micro (or) t2 Large]"
}
```

Terraform file final look

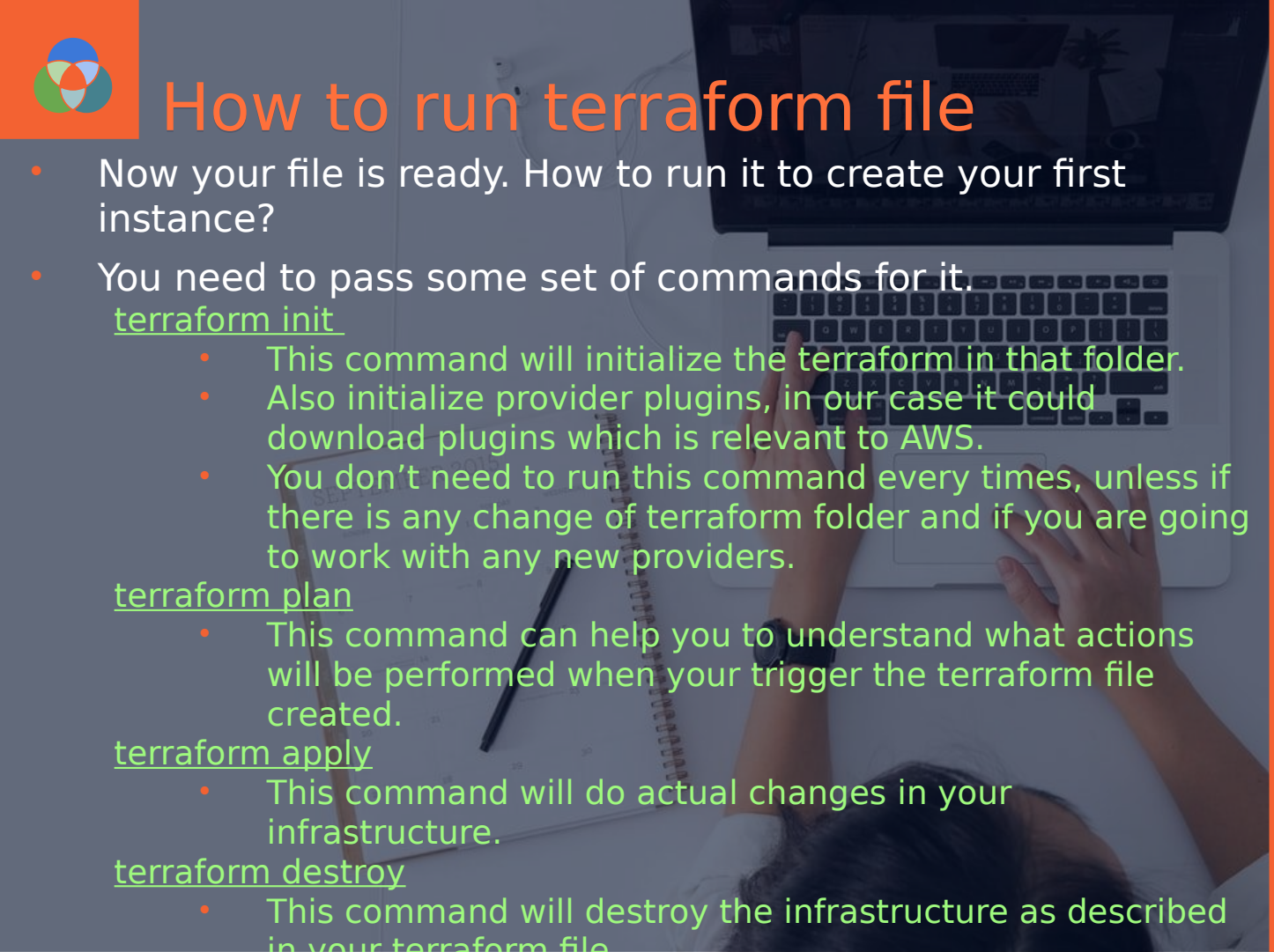
```
provider "aws" {
        access_key = ”xxxxxxxxxxxxxxx"
        secret_key = ”xxxxxxxxxxxxxxxx"
        region = "ap-south-1"
}

resource "aws_instance" "First_EC2" {
        ami = "ami-052c08d70def0ac62"
        instance_type = "t2.micro"
}
```

What is the disadvantage here?

Your credentials might get exposed when you are uploading this file in git. We shall see how to avoid such things in coming classes.

Code : git-repo [file path : Terraform/Codes/E1_without-vars]



How to run terraform file

- Now your file is ready. How to run it to create your first instance?
- You need to pass some set of commands for it.

terraform init

- This command will initialize the terraform in that folder.
- Also initialize provider plugins, in our case it could download plugins which is relevant to AWS.
- You don’t need to run this command every times, unless if there is any change of terraform folder and if you are going to work with any new providers.

terraform plan

- This command can help you to understand what actions will be performed when your trigger the terraform file created.

terraform apply

- This command will do actual changes in your infrastructure.

terraform destroy

- This command will destroy the infrastructure as described in your terraform file

Output plan to a file

Lets see some commands which can be helpful for terraform administrations.

terraform plan –out output.terraform

- This command will extract your plan into a file.
- However that file is not human readable.
- But why this is required?
- Lets say you have created terraform file with certain configuration and some of your colleague has made minor changes in terraform file which might cause the problem.
- In such case you can go ahead and apply the file you extracted at the time of its creation. [more like a time machine going back and front]

terraform apply output.terraform

- This command is going to apply the changes from the file.

terraform plan –out file name ; terraform apply filename

- Like this you can just combine those 2 command which eventually create the file and apply it as well.
- This is the best practice I would suggest to use in prod infrastructures.

End of this topic !

Any questions?