МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ ім. М. Є. Жуковського «Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокумунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

Лабораторна робота №8

З дициплiни: «Технології DevOps»

Виконав:

студент 5 курсу групи №555 ім

Напряму підготовки

125 Кібербезпека та захист інформації

ст. Орлов Станіслав Валерійович

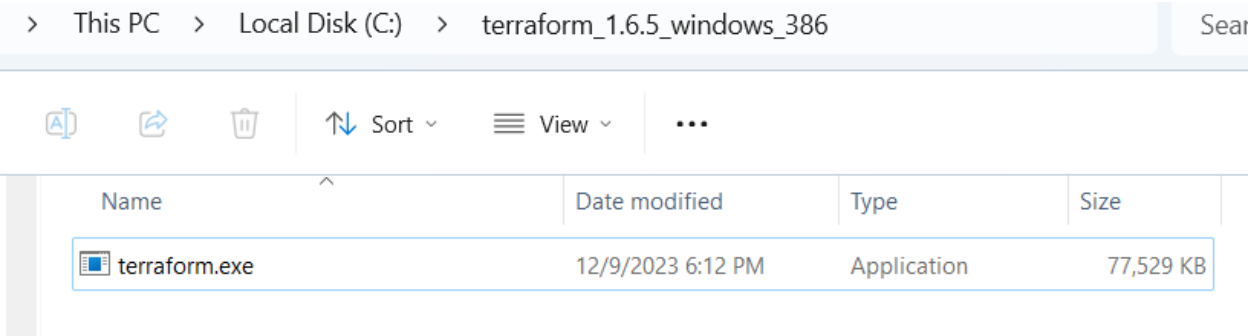
Прийняв:

к.т.н., доцент

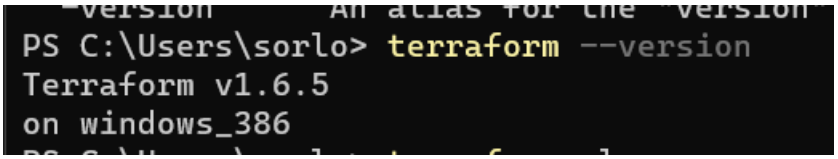
Узун Дмитро Дмитрович

Харків, 2023

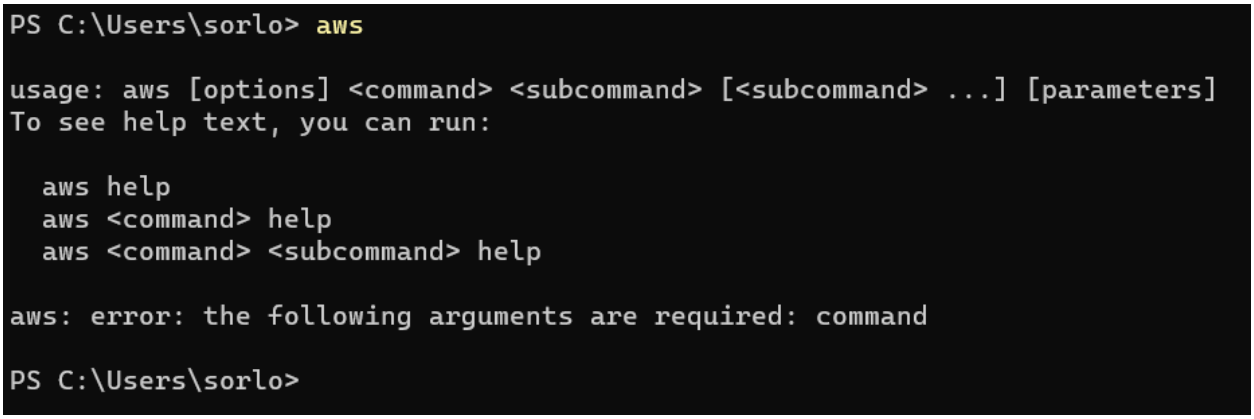
1. Install Terraform



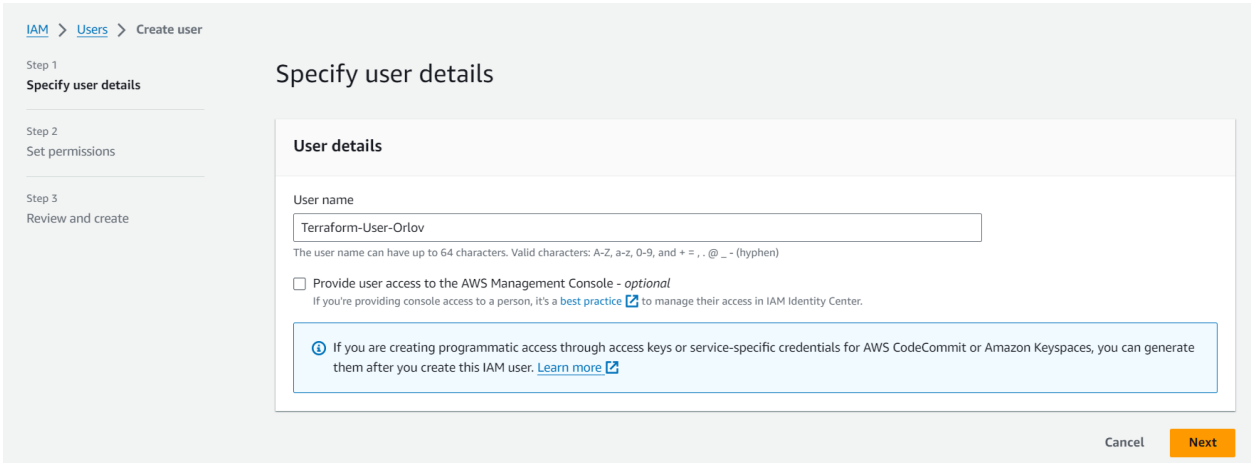
2. Adding terraform path to environment variables & verification.



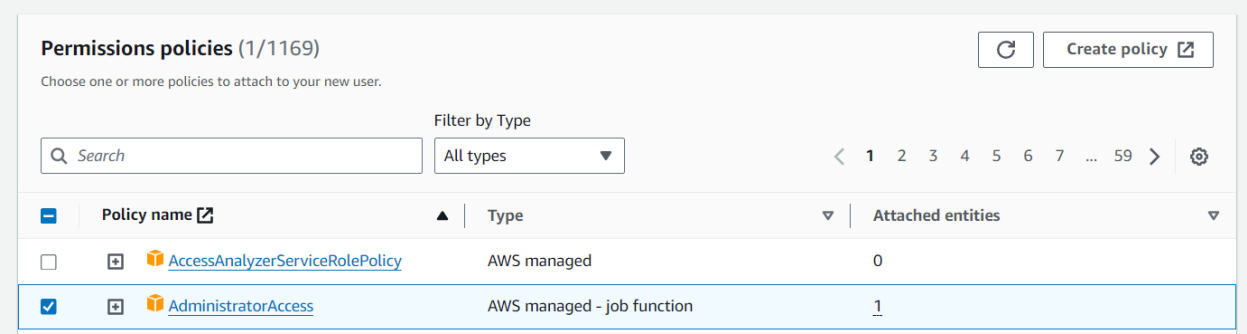
3. Installing AWS CLI



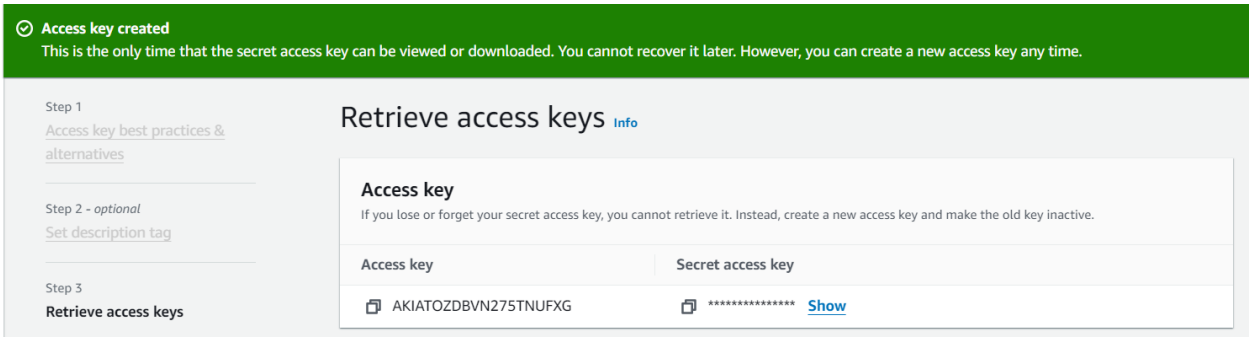
4. Creating new Terraform User



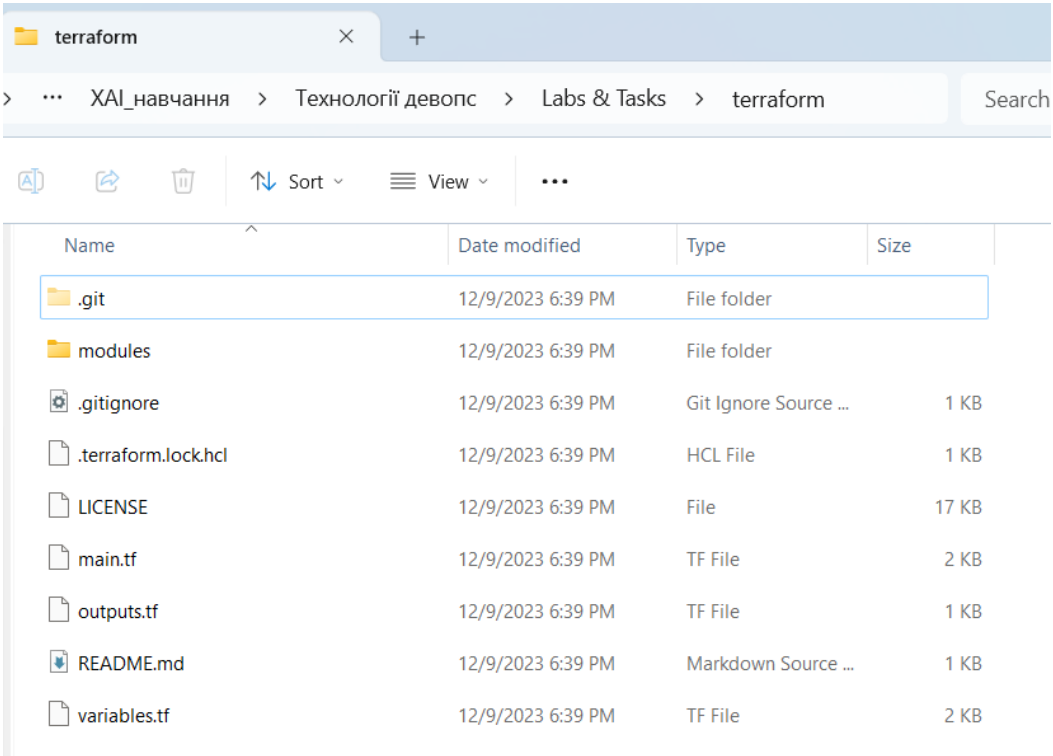
Setting up permissions



Retrieving access keys



5. Cloning up the repo - https://github.com/hashicorp/learn-terraform-modules.git



main.tf 1

C: > Projects > ХАІ_навчання > Технології девопс > Labs & Tasks > terraform > main.tf > resource "aws_eip" "ip"

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "2.69.0"
    }
  }
}
provider aws {
  region = "us-west-1"
}
data "aws_ami" "amazon_linux" {
  most_recent = true
  owners = ["amazon"]
  filter {
    name = "name"
    values = ["amzn2-ami-hvm-*-x86_64-gp2"]
  }
}
resource "aws_instance" "example_a" {
  ami = data.aws_ami.amazon_linux.id
  instance_type = "t2.micro"
}
resource "aws_instance" "example_b" {
  ami = data.aws_ami.amazon_linux.id
  instance_type = "t2.micro"
}
resource "aws_eip" "ip" {
  vpc = true
  instance = aws_instance.example_a.id
```

6. Terraform init

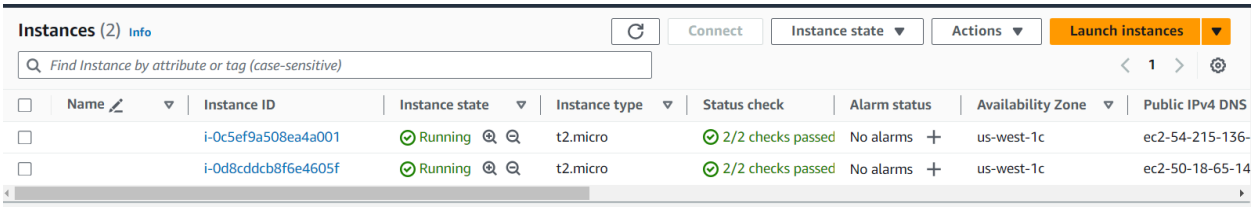
```
PS C:\Projects\ХАІ_навчання\Технології девопс\Labs & Tasks\terraform> terraform init

Initializing the backend...
Initializing modules...
Downloading registry.terraform.io/terraform-aws-modules/ec2-instance/aws 2.12.0 for ec2_instances...
- ec2_instances in .terraform\modules\ec2_instances
Downloading registry.terraform.io/terraform-aws-modules/vpc/aws 2.21.0 for vpc...
- vpc in .terraform\modules\vpc
- website_s3_bucket in modules\aws-s3-static-website-bucket

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Installing hashicorp/aws v3.25.0...
- Installed hashicorp/aws v3.25.0 (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.

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.
PS C:\Projects\ХАІ_навчання\Технології девопс\Labs & Tasks\terraform>
```

7. Applying terraform script

```
commands will detect it and remind you to do so if necessary.
PS C:\Projects\XAI_навчання\Технології девопс\Labs & Tasks\terraform> terraform apply
data.aws_ami.amazon_linux: Reading...
data.aws_ami.amazon_linux: Read complete after 1s [id=ami-0839bf007aad25236]

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:

  # aws_eip.ip will be created
  + resource "aws_eip" "ip" {
      + allocation_id     = (known after apply)
      + association_id    = (known after apply)
      + customer_owned_ip = (known after apply)
      + domain            = (known after apply)
      + id                = (known after apply)
      + instance          = (known after apply)
      + network_interface = (known after apply)
      + private_dns       = (known after apply)
      + private_ip        = (known after apply)
      + public_dns        = (known after apply)
      + public_ip         = (known after apply)
      + public_ipv4_pool  = (known after apply)
      + vpc               = true
    }
```

```
Plan: 3 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_instance.example_b: Creating...
aws_instance.example_a: Creating...
aws_instance.example_a: Still creating... [10s elapsed]
aws_instance.example_b: Still creating... [10s elapsed]
aws_instance.example_a: Still creating... [20s elapsed]
aws_instance.example_b: Still creating... [20s elapsed]
aws_instance.example_b: Still creating... [30s elapsed]
aws_instance.example_a: Still creating... [30s elapsed]
aws_instance.example_b: Still creating... [40s elapsed]
aws_instance.example_a: Still creating... [40s elapsed]
aws_instance.example_a: Creation complete after 40s [id=i-0d8cddcb8f6e4605f]
aws_eip.ip: Creating...
aws_instance.example_b: Creation complete after 40s [id=i-0c5ef9a508ea4a001]
aws_eip.ip: Creation complete after 4s [id=eipalloc-00618154fa3c39504]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Projects\XAI_навчання\Технології девопс\Labs & Tasks\terraform>
```

8. Observe created resources via AWS Amazon console.



Висновки

У ході виконання лабораторної роботи оволодів основними навичками роботи з інфраструктурої у вигляді коду Terraform. Створив та запустив тераформа-скрипт для створення та підняття двох EC2 інстансів у клауді Amazon.