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 №6

З диципліни: «Теорія та технології розроблення безпечних розподільних систем»

Виконав:

студент 5 курсу групи №555 ім

Напряму підготовки

125 Кібербезпека та захист інформації
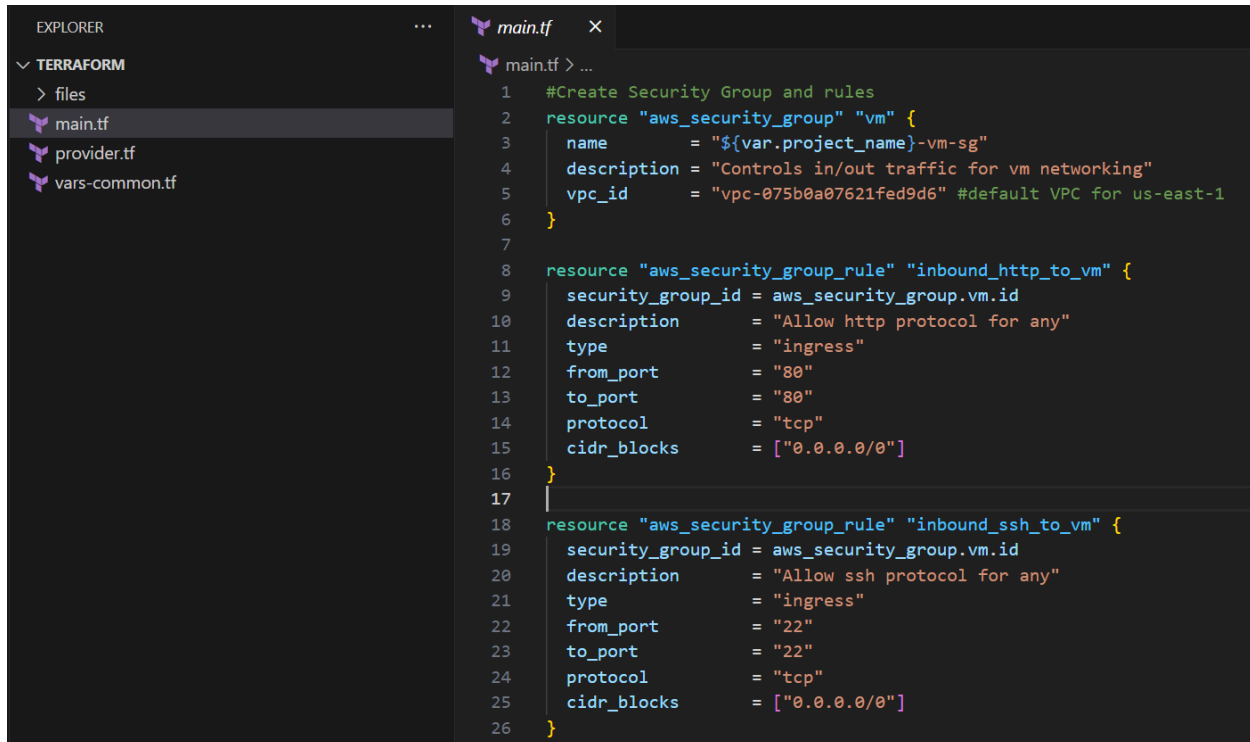
ст. Орлов Станіслав Валерійович

Прийняв:

аспірант

Карпенко Андрій Сергійович

Харків, 2023

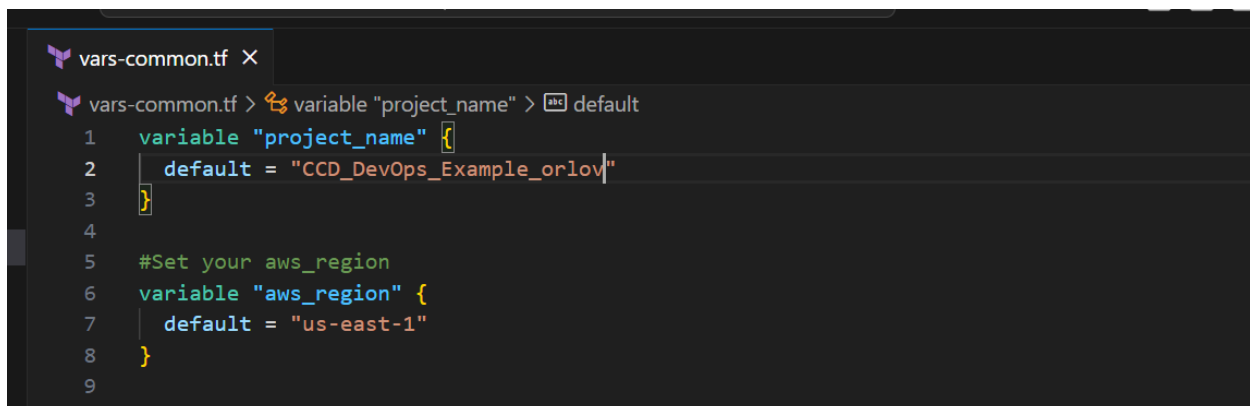# Step 1 Opening simple terraform project:

1. Download and unzip sample project.

Step 2 Understanding Terraform Code

2. Replace student id:

3. Update key VM settings with created key-pair name

	Name	Type	Created	Fingerprint	ID
	StanislavOrlov-key-pair	rsa	2023/11/20 16:34 GMT+2	4b:1f:7a:70:77:c8:f0:0b:35:cc:b1:c8:4...	key-000020bf3f8fc04dc
	Stanislav-Orlov-Linux-Key	rsa	2023/11/20 18:43 GMT+2	9e:5f:46:cc:8c:b1:d7:b5:38:73:c1:55:1...	key-02f17b0da31936ae3

Key pairs (2) Info

Actions ▼   Create key pair

Find Key Pair by attribute or tag

4. Launching a new EC2 instance

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

Distributed Systems - Orlov - Lab 6

Add additional tags

Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description
Amazon Linux 2023 AMI 2023.2.20231113.0 x86_64 HVM kernel-6.1

Architecture

64-bit (x86) ▼

AMI ID

ami-0230bd60aa48260c6   Verified provider

VPC - required   Info

vpc-00aa9ff9e57f9cab3   (default) ▼
172.31.0.0/16

Subnet   Info

subnet-0ec4614168a52d6ba
VPC: vpc-00aa9ff9e57f9cab3   Owner: 237907323765   Availability Zone: us-east-1f
IP addresses available: 4091   CIDR: 172.31.64.0/20

Create new subnet

Step 4 Configure AWS Credentials for AWS CLI

Step 5 Deploying AWS Virtual Infrastructure

terraform init

terraform validate

terraform plan

```
PS C:\Projects\XAI_навчання\Git\SecureDistibutedSystems\Lab6\Terraform(Lab-10)\Terraform> 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:

  # aws_instance.CCD_demo will be created
  + resource "aws_instance" "CCD_demo" {
      + ami                                  = "ami-0b0dcb5067f052a63"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "CCD2022"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
```

List of planned of actions to be performed

```
  + resource "aws_security_group_rule" "inbound_ssh_to_vm" {
      + cidr_blocks                = [
          + "0.0.0.0/0",
        ]
      + description                = "Allow ssh protocol for any"
      + from_port                  = 22
      + id                         = (known after apply)
      + protocol                   = "tcp"
      + security_group_id          = (known after apply)
      + security_group_rule_id     = (known after apply)
      + self                       = false
      + source_security_group_id   = (known after apply)
      + to_port                    = 22
      + type                       = "ingress"
    }

  # aws_security_group_rule.vm_outbound_any will be created
  + resource "aws_security_group_rule" "vm_outbound_any" {
      + cidr_blocks                = [
          + "0.0.0.0/0",
        ]
      + from_port                  = 0
      + id                         = (known after apply)
      + protocol                   = "-1"
      + security_group_id          = (known after apply)
      + security_group_rule_id     = (known after apply)
      + self                       = false
      + source_security_group_id   = (known after apply)
      + to_port                    = 0
      + type                       = "egress"
    }

Plan: 6 to add, 0 to change, 0 to destroy.
```

Applying infrastructure changes

```
PS C:\Projects\XAI_навчання\Git\SecureDistibutedSystems\Lab6\Terraform(Lab-10)\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:
```

Getting and error about invalid key pair

```
Error: creating EC2 Instance: InvalidKeyPair.NotFound: The key pair 'CCD2022' does not exist
        status code: 400, request id: ebc4526c-3145-4cd0-9a64-d034065139a4
```

Updating with personal Key Pair information

```
instance_type           =   t2.micro
user_data               = file("./files/template.tpl") #U
key_name                = "StanislavOrlov-key-pair"
```

Message regarding successful instance creation

```
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_instance.CCD_demo: Creating...
aws_instance.CCD_demo: Still creating... [10s elapsed]
aws_instance.CCD_demo: Still creating... [20s elapsed]
aws_instance.CCD_demo: Still creating... [30s elapsed]
aws_instance.CCD_demo: Creation complete after 35s [id=i-0f2dacca6987937e0]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Projects\XAI_навчання\Git\SecureDistibutedSystems\Lab6\Terraform(Lab-10)\Terraform>
```
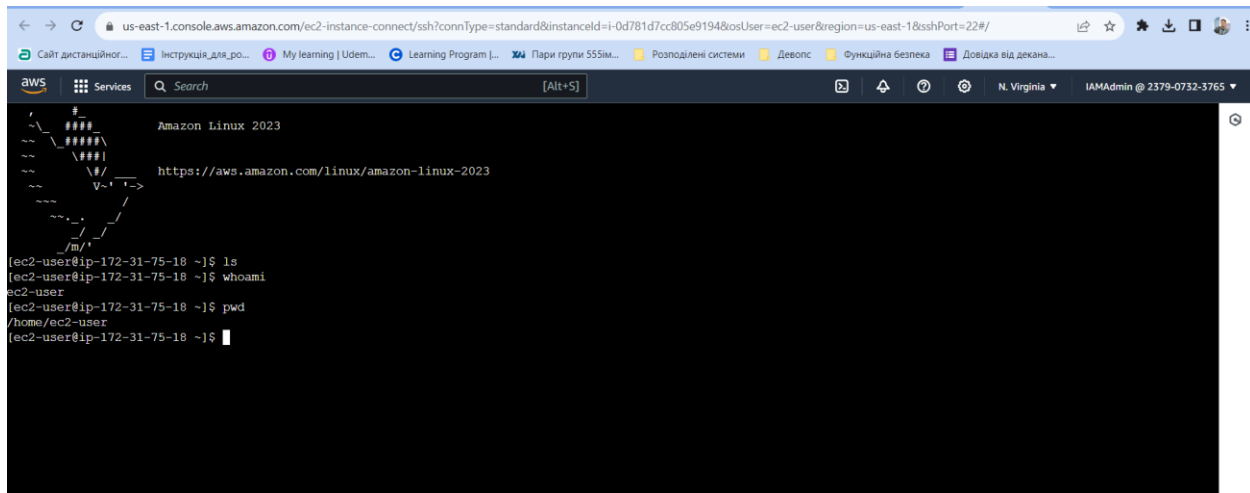
Step 6 Checking Deployed Resources

Connecting to deployed EC2 instance

Step 7 Destroying AWS EC2 Resources


```
PS C:\Projects\XAI_навчання\Git\SecureDistibutedSystems\Lab6\Terraform(Lab-10)\Terraform> terraform destroy
aws_security_group.vm: Refreshing state... [id=sg-087c5031dc37d09d3]
aws_security_group_rule.vm_outbound_any: Refreshing state... [id=sgrule-1960886935]
aws_security_group_rule.inbound_http_to_vm: Refreshing state... [id=sgrule-916124940]
aws_security_group_rule.inbound_ssh_to_vm: Refreshing state... [id=sgrule-2146391873]
aws_network_interface.main: Refreshing state... [id=eni-09f0c7b41d75b1473]
aws_instance.CCD_demo: Refreshing state... [id=i-0f2dacca6987937e0]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy
```

```
Plan: 0 to add, 0 to change, 6 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_security_group_rule.vm_outbound_any: Destroying... [id=sgrule-1960886935]
aws_security_group_rule.inbound_http_to_vm: Destroying... [id=sgrule-916124940]
aws_security_group_rule.inbound_ssh_to_vm: Destroying... [id=sgrule-2146391873]
aws_instance.CCD_demo: Destroying... [id=i-0f2dacca6987937e0]
aws_security_group_rule.vm_outbound_any: Destruction complete after 1s
aws_security_group_rule.inbound_ssh_to_vm: Destruction complete after 1s
aws_security_group_rule.inbound_http_to_vm: Destruction complete after 2s
aws_instance.CCD_demo: Still destroying... [id=i-0f2dacca6987937e0, 10s elapsed]
aws_instance.CCD_demo: Still destroying... [id=i-0f2dacca6987937e0, 20s elapsed]
aws_instance.CCD_demo: Still destroying... [id=i-0f2dacca6987937e0, 30s elapsed]
aws_instance.CCD_demo: Destruction complete after 31s
aws_network_interface.main: Destroying... [id=eni-09f0c7b41d75b1473]
aws_network_interface.main: Destruction complete after 0s
aws_security_group.vm: Destroying... [id=sg-087c5031dc37d09d3]
aws_security_group.vm: Destruction complete after 1s

Destroy complete! Resources: 6 destroyed.
PS C:\Projects\XAI_навчання\Git\SecureDistibutedSystems\Lab6\Terraform(Lab-10)\Terraform>
```

Висновки:

У ході виконання лабораторної роботи ознайомився та отримав практичні навички з Infrastructure-as-Code створивши хмарну інфрастуктуру AWS використовуючи фреймворк Terraform.

Посилання на GitHub - [SecureDistibutedSystems/Lab6 at master · stanislavorlov/SecureDistibutedSystems (github.com)](https://github.com)