



BÀI TẬP THỰC HÀNH 01

Dùng Terraform và CloudFormation để quản lý và triển khai hạ tầng AWS

Môn học: Công nghệ DevOps và Ứng dụng

Mã MH: NT548

GIẢNG VIÊN HƯỚNG DẪN:

ThS. Lê Anh Tuấn

Dùng Terraform và CloudFormation để quản lý và triển khai hạ tầng AWS **2**

A. YÊU CẦU THỰC HÀNH

Dùng Terraform và CloudFormation để quản lý và triển khai tự động hạ tầng

AWS.

1. Các dịch vụ cần triển khai:

- **VPC:** Tạo một VPC chứa các thành phần sau (**3 điểm**):
 - Subnets: Bao gồm cả Public Subnet (kết nối với Internet Gateway) và Private Subnet (sử dụng NAT Gateway để kết nối ra ngoài).
 - Internet Gateway: Kết nối với Public Subnet để cho phép các tài nguyên bên trong có thể truy cập Internet.
 - Default Security Group: Tạo Security Group mặc định cho VPC
- **Route Tables:** Tạo Route Tables cho Public và Private Subnet (**2 điểm**):
 - Public Route Table: Định tuyến lưu lượng Internet thông qua Internet Gateway.
 - Private Route Table: Định tuyến lưu lượng Internet thông qua NAT Gateway.
- **NAT Gateway:** Cho phép các tài nguyên trong Private Subnet có thể kết nối Internet mà vẫn bảo đảm tính bảo mật (**1 điểm**).
- **EC2:** Tạo các instance trong Public và Private Subnet, đảm bảo Public instance có thể truy cập từ Internet, còn Private instance chỉ có thể truy cập từ Public instance thông qua SSH hoặc các phương thức bảo mật khác (**2 điểm**).
- **Security Groups:** Tạo các Security Groups để kiểm soát lưu lượng vào/ra của EC2 instances (**2 điểm**):
 - Public EC2 Security Group: Chỉ cho phép kết nối SSH (port 22) từ một IP cụ thể

(hoặc IP của người dùng).

- Private EC2 Security Group: Cho phép kết nối từ Public EC2 instance thông qua port cần thiết (SSH hoặc các port khác nếu có nhu cầu).

2. Yêu cầu:

- Các dịch vụ phải được viết dưới dạng module.
- Phải đảm bảo an toàn bảo mật cho EC2 (thiết lập Security Groups).
- Kết quả:
 - Báo cáo Word (theo mẫu đi kèm ở dưới).
 - Link GitHub (source code và file README hướng dẫn cách chạy mã nguồn).

3. Lưu ý:

- Sinh viên cần thường xuyên cập nhật mã nguồn lên GitHub.
- Phải có các test cases để kiểm tra từng dịch vụ được triển khai thành công.

BÁO CÁO THỰC HÀNH 01

Dùng Terraform và CloudFormation để quản lý và triển khai hạ tầng AWS

Môn học: Công nghệ DevOps và Ứng dụng

Lớp: NT548.P21

THÀNH VIÊN THỰC HIỆN (Nhóm 17):

STT	Họ và tên	MSSV
1	Phạm Đức Huy Hoàng	22521344
2	Nguyễn Thị Thanh Tuyền	22521632
3	Trần Văn Thân	22521322
4	Huỳnh Nhật Duy	22520314

Điểm tự đánh giá
10

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	2 tiếng
Phân chia công việc	Hoàng + Tuyền: Terraform Thân + : Cloudformation
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

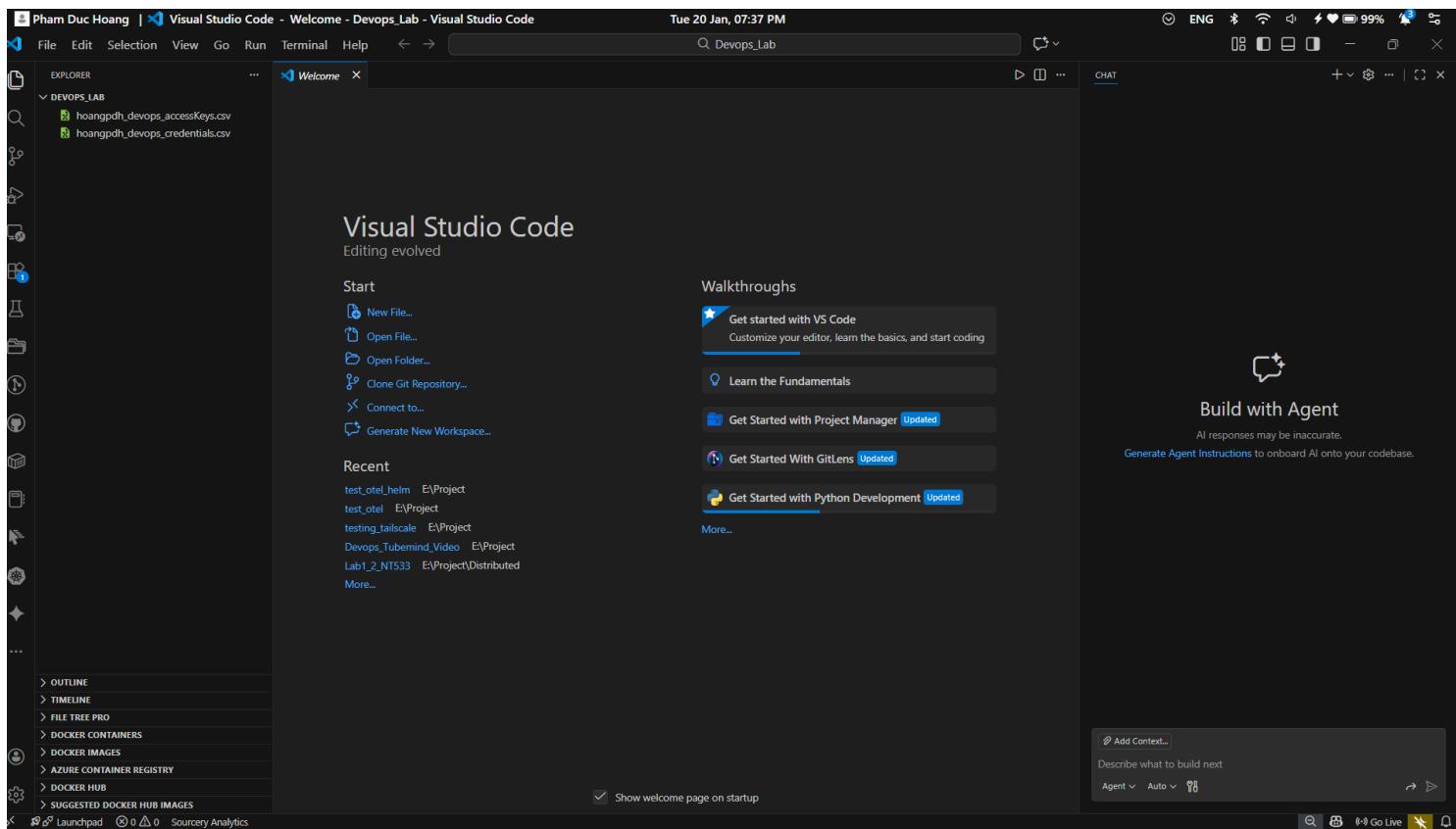
B. BÁO CÁO CHI TIẾT

Câu 1: Terraform

1. Tạo user trên AWS để lấy access key và secret access key

The screenshot shows the AWS IAM 'Create user' success page. At the top, there's a green banner with the message 'User created successfully'. Below it, a note says 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' A 'View user' button is present. On the left, a vertical navigation bar shows steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password), with Step 4 being the current active step. The main content area is titled 'Retrieve password' and contains 'Console sign-in details' with fields for 'Console sign-in URL' (https://hoangpdh.signin.aws.amazon.com/console), 'User name' (hoangpdh_devops), and 'Console password' (redacted). A 'Email sign-in instructions' button is also visible.

The screenshot shows the AWS IAM 'Create access key' page. At the top, there's a green banner with the message 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below it, a vertical navigation bar shows steps: Step 1 (Access key best practices & alternatives), Step 2 - optional (Set description tag), Step 3 (Retrieve access keys), with Step 3 being the current active step. The main content area is titled 'Retrieve access keys' and contains an 'Access key' section with a note about losing the key. It shows an 'Access key' field (AKIAQSTSKNPN62JL5IO3) and a 'Secret access key' field (redacted). A 'Show' link is available for the secret key. Below this is an 'Access key best practices' section with a bulleted list: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' A 'Download .csv file' and 'Done' button are at the bottom.



2. Cài đặt công cụ

Nhóm đã tiến hành cài đặt các công cụ cần thiết cho bài lab bao gồm:

- **AWS CLI:** Để quản lý AWS services qua dòng lệnh.
- **Terraform:** Công cụ IaC để triển khai và quản lý hạ tầng tự động.

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> aws --version
aws-cli/2.31.22 Python/3.13.7 Windows/11 exe/AMD64
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> terraform --version
Terraform v1.13.4
on windows_amd64

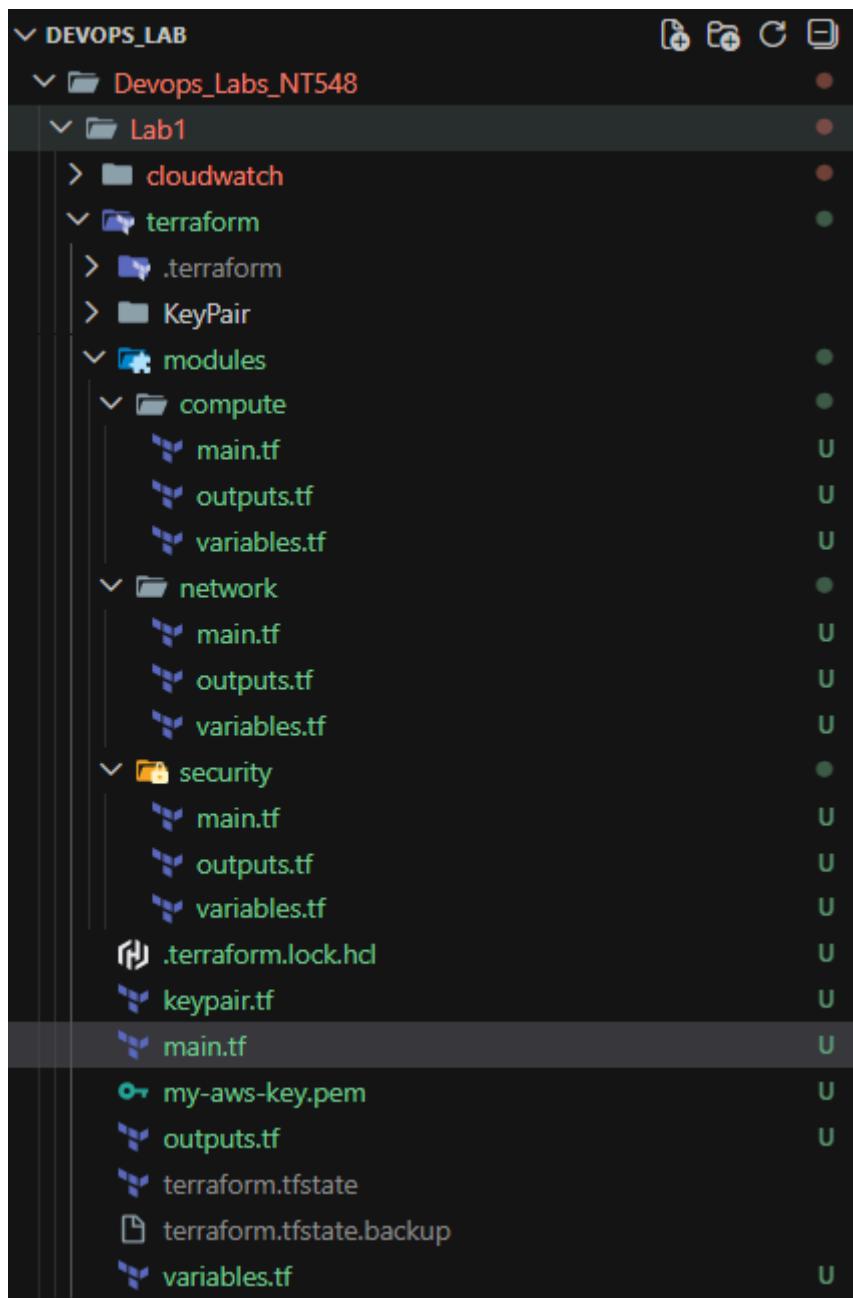
Your version of Terraform is out of date! The latest version
is 1.14.3. You can update by downloading from https://developer.hashicorp.com/terraform/install
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> |
```

3. Cấu hình dự án

Nhóm đã tiến hành viết các file terraform để triển khai tài nguyên trên AWS bao gồm:

1. **Keypair (Để SSH vào ec2 khi thiết lập xong)**
2. **Network (Module này xử lý VPC, Subnets, IGW, NAT và Route Tables)**
3. **Security Groups (Module này kiểm soát truy cập vào EC2)**
4. **Compute (Module này tạo các thực thể EC2)**

Mỗi module sẽ có 3 file main.tf, outputs.tf, và variables.tf để cấu hình, quản lý và xuất output mong muốn để có thể control và kiểm soát cho việc scale và viết code khi thêm nhiều tài nguyên hơn.



4. Các bước thực hiện

Bước 1: terraform init (Khởi tạo project và tải các provider/module cần thiết)

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> terraform init
Initializing the backend...
Initializing modules...
- compute in modules\compute
- network in modules\network
- security in modules\security
Initializing provider plugins...
- Finding latest version of hashicorp/tls...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/local...
- Installing hashicorp/tls v4.1.0...
- Installed hashicorp/tls v4.1.0 (signed by HashiCorp)
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
- Installing hashicorp/local v2.6.1...
- Installed hashicorp/local v2.6.1 (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.
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> |
```

Bước 2: terraform plan (Xem trước các thay đổi mà terraform sẽ áp dụng)

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> 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_key_pair.deployer will be created
+ resource "aws_key_pair" "deployer" {
    + arn          = (known after apply)
    + fingerprint  = (known after apply)
    + id          = (known after apply)
    + key_name     = "lab-key"
    + key_name_prefix = (known after apply)
    + key_pair_id   = (known after apply)
    + key_type      = (known after apply)
    + public_key    = (known after apply)
    + region        = "us-east-1"
    + tags_all      = (known after apply)
}

# local_file.ssh_key will be created
+ resource "local_file" "ssh_key" {
    + content          = (sensitive value)
    + content_base64sha256 = (known after apply)
    + content_base64sha512 = (known after apply)
    + content_md5      = (known after apply)
    + content_sha1      = (known after apply)
    + content_sha256     = (known after apply)
    + content_sha512     = (known after apply)
    + directory_permission = "0777"
    + file_permission     = "0600"
```

```

+ cidr_blocks      = [
+   "0.0.0.0/0",
]
+ from_port       = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol        = "tcp"
+ security_groups = []
+ self            = false
+ to_port         = 22
# (1 unchanged attribute hidden)
},
]
+ name             = (known after apply)
+ name_prefix     = (known after apply)
+ owner_id        = (known after apply)
+ region          = "us-east-1"
+ revoke_rules_on_delete = false
+ tags_all        = (known after apply)
+ vpc_id          = (known after apply)
}

```

Plan: 17 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```

+ private_ip = (known after apply)
+ public_ip  = (known after apply)

```

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> |

Bước 3: terraform apply (Bắt đầu tạo tài nguyên)

```

(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> terraform apply -auto-approve

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_key_pair.deployer will be created
+ resource "aws_key_pair" "deployer" {
  + arn           = (known after apply)
  + fingerprint   = (known after apply)
  + id            = (known after apply)
  + key_name      = "lab-key"
  + key_name_prefix = (known after apply)
  + key_pair_id    = (known after apply)
  + key_type       = (known after apply)
  + public_key     = (known after apply)
  + region          = "us-east-1"
  + tags_all       = (known after apply)
}

# local_file.ssh_key will be created
+ resource "local_file" "ssh_key" {
  + content          = (sensitive value)
  + content_base64sha256 = (known after apply)
  + content_base64sha512 = (known after apply)
  + content_md5      = (known after apply)
  + content_sha1     = (known after apply)
  + content_sha256   = (known after apply)
  + content_sha512   = (known after apply)
  + directory_permission = "0777"
  + file_permission   = "0600"
}
```

Apply complete! Resources: 17 added, 0 changed, 0 destroyed.

Outputs:

```

private_ip = "10.0.2.41"
public_ip = "98.92.81.184"
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> |

```

Ở đây sau khi apply thành công, output sẽ trả về 2 ip của 2 ec2, một cái là:

- Public EC2: 98.92.81.184
- Private EC2: 10.0.2.41

5. Kiểm tra tài nguyên trên AWS Console

Dưới đây là các tài nguyên đã được triển khai trên AWS thông qua terraform

The screenshot shows the AWS VPC dashboard with the title "Your VPCs". There are two VPC entries listed:

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR
Lab-VPC	vpc-070c103668349d546	Available	-	-	Off	10.0.0.0/24
Lab-VPC	vpc-0d2a1f71497e11c67	Available	-	-	Off	10.0.1.0/24

A message at the top right indicates "Last updated less than a minute ago".

The screenshot shows the AWS Subnets dashboard with the title "Subnets". There are two subnet entries listed:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Private-Subnet	subnet-0f345e7a96c421fde	Available	vpc-0d2a1f71497e11c67 Lab-VPC	Off	10.0.2.0/24
Public-Subnet	subnet-0d0db299217cb5f0	Available	vpc-0d2a1f71497e11c67 Lab-VPC	Off	10.0.1.0/24

A message at the top right indicates "Last updated less than a minute ago".

AWS Search [Alt+S] United States (N. Virginia) hoangpdh_vpbank (0399-6820-5787) hoangpdh_vpbank

VPC > Route tables

VPC dashboard AWS Global View Filter by VPC: Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only Internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers Security Network ACLs Security groups

Route tables (4) Info Last updated 1 minute ago Actions Create route table

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-05ab0de91a04c0627	subnet-0d00db299217cb...	-	No	vpc-0d2a1f71497e11c67 Lab-..
-	rtb-0e6a24cdf1607ff8c	subnet-0f345e7a96c421f...	-	No	vpc-0d2a1f71497e11c67 Lab-..
-	rtb-0d476db389e6fc02	-	-	Yes	vpc-070c103668349d546 Lab-..
-	rtb-075ba5bef49dddbfc	-	-	Yes	vpc-0d2a1f71497e11c67 Lab-..

Select a route table

CloudShell Feedback Console mobile app © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Search [Alt+S] United States (N. Virginia) hoangpdh_vpbank (0399-6820-5787) hoangpdh_vpbank

VPC > Internet gateways

VPC dashboard AWS Global View Filter by VPC: Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only Internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers Security Network ACLs Security groups

Internet gateways (1) Info Actions Create internet gateway

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0d37c6937998847ed	Attached	vpc-0d2a1f71497e11c67 Lab-VPC	039968205787

Select an internet gateway above

CloudShell Feedback Console mobile app © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Global View 

NAT gateways (3) Info

Name	NAT gateway ID	Connectivity...	State	State message	Availability ...	Route table ID
-	nat-07666bf31d41d1ec0	Public	Available	-	Zonal	-
-	nat-0839125eeba006e09	Public	Deleted	-	Zonal	-
-	nat-053aca247d79286d0	Public	Deleted	-	Zonal	-

Select a NAT gateway

CloudShell Feedback Console mobile app © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Global View 

Elastic IP addresses (1) Info

Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record	Assoc
-	100.50.254.75	Public IP	eipalloc-0df4c22591e3c8898	-	-

Select an elastic IP address

View IP address usage and recommendations to release unused IPs with [Public IP insights](#). 

CloudShell Feedback Console mobile app © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area shows a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. Two instances are listed: Private-EC2 (i-0f8bf346a2607bbb) and Public-EC2 (i-032b71583374a6cd7), both in the 'Running' state. A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a filter bar says 'Instance state = running'. Buttons for 'Connect', 'Actions', and 'Launch instances' are at the top right.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	Private-EC2	i-0f8bf346a2607bbb	Running	t3.small	3/3 checks passed	View alarms +	us-east-1f	-
<input type="checkbox"/>	Public-EC2	i-032b71583374a6cd7	Running	t3.small	3/3 checks passed	View alarms +	us-east-1f	ec2-98-92-8

6. Kiểm tra 2 instance và SSH thử

The screenshot shows the AWS EC2 Instances page. The instance summary for 'i-0f8bf346a26076bbb' is displayed. Key details include:

- Instance ID:** i-0f8bf346a26076bbb
- Public IPv4 address:** -
- IPv6 address:** -
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-2-41.ec2.internal
- Instance type:** t3.small
- VPC ID:** vpc-0d2a1f71497e11c67 (Lab-VPC)
- Subnet ID:** subnet-0f345e7a96c421fde (Private-Subnet)
- Instance ARN:** arn:aws:ec2:us-east-1:039968205787:instance/i-0f8bf346a26076bbb
- Managed:** false

On the left sidebar, the 'Instances' section is expanded, showing options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager.

Instance summary for i-0f8bf346a26076bbb (Private-EC2) [Info](#)

Updated less than a minute ago

Instance ID i-0f8bf346a26076bbb	Public IPv4 address -	Private IPv4 addresses 10.0.2.41
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-2-41.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-2-41.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t3.small	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendation s. Learn more
Auto-assigned IP address -	VPC ID vpc-0d2a1f71497e11c67 (Lab-VPC)	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0f345e7a96c421fde (Private-Subnet)	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:1039968205787:instance/i-0f8bf346a26076bbb	

Sau khi kiểm tra trên AWS Console và thấy tài nguyên EC2 Public đã tồn tại, nhóm bắt đầu test để đảm bảo hơn:

- **Test case 1:** Kiểm tra khả năng truy cập Public EC2 (sử dụng file .pem tạo ra bởi moduel KeyPair)

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> ssh -i my-aws-key.pem ec2-user@98.92.81.184
,      #
~\_ ##### Amazon Linux 2023
~~ \#####|
~~ \###| |
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~   /_
~~*_ _/_/
~~/_/_/
~~/_m/'[ec2-user@ip-10-0-1-145 ~]$ |
```

Kết quả: Đã SSH vào được

- **Test case 2:** Kiểm tra khả năng kết nối nội bộ (Public -> Private) (copy key.pem và mang vào EC2 Public để có thể ssh vào private thông qua public)

```
[ec2-user@ip-10-0-1-145 ~]$ vi key.pem
[ec2-user@ip-10-0-1-145 ~]$ chmod 400 key.pem
[ec2-user@ip-10-0-1-145 ~]$ ssh -i key.pem ec2-user@10.0.2.41
The authenticity of host '10.0.2.41 (10.0.2.41)' can't be established.
ED25519 key fingerprint is SHA256:bk1DIKvcLY9VfD3qQ0/HuRBtUdmoScMlWONRGLqRnnc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: ^C
[ec2-user@ip-10-0-1-145 ~]$ ssh -i key.pem ec2-user@10.0.2.41 -o StrictHostKeyChecking=no
Warning: Permanently added '10.0.2.41' (ED25519) to the list of known hosts.

,
#_
~\_ #####_      Amazon Linux 2023
~~ \#####\
~~ \###|
~~  \#/ ____  https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~   /_
~~.~.  _/
~/ _/
~/m/'[ec2-user@ip-10-0-2-41 ~]$ |
```

Kết quả: Truy cập thành công từ máy Public và máy Private

- **Test case 3:** Kiểm tra Internet từ máy Private (NAT Gateway) (Thực hiện bằng cách đứng ở máy Private và ping ra google.com)

```
[ec2-user@ip-10-0-1-145 ~]$ vi key.pem
[ec2-user@ip-10-0-1-145 ~]$ chmod 400 key.pem
[ec2-user@ip-10-0-1-145 ~]$ ssh -i key.pem ec2-user@10.0.2.41
The authenticity of host '10.0.2.41 (10.0.2.41)' can't be established.
ED25519 key fingerprint is SHA256:bk1DIKvcLY9VfD3qQ0/HuRBtUdmoScMlWONRGLqRnnc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: ^C
[ec2-user@ip-10-0-1-145 ~]$ ssh -i key.pem ec2-user@10.0.2.41 -o StrictHostKeyChecking=no
Warning: Permanently added '10.0.2.41' (ED25519) to the list of known hosts.

,
#_
~\_ #####_      Amazon Linux 2023
~~ \#####\
~~ \###|
~~  \#/ ____  https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~   /_
~~.~.  _/
~/ _/
~/m/'[ec2-user@ip-10-0-2-41 ~]$ ping -c 4 google.com
PING google.com (142.251.163.100) 56(84) bytes of data.
64 bytes from wv-in-f100.1e100.net (142.251.163.100): icmp_seq=1 ttl=101 time=2.06 ms
64 bytes from wv-in-f100.1e100.net (142.251.163.100): icmp_seq=2 ttl=101 time=1.60 ms
64 bytes from wv-in-f100.1e100.net (142.251.163.100): icmp_seq=3 ttl=101 time=1.59 ms
64 bytes from wv-in-f100.1e100.net (142.251.163.100): icmp_seq=4 ttl=101 time=1.59 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.587/1.710/2.061/0.202 ms
[ec2-user@ip-10-0-2-41 ~]$ |
```

- **Test case 4:** Kiểm tra tính bảo mật (Chặn truy cập trực tiếp)

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> ssh -i my-aws-key.pem ec2-user@10.0.2.41
ssh: connect to host 10.0.2.41 port 22: Connection timed out
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1> |
```

Kết quả: Không thể ssh trực tiếp vào máy private từ mạng ngoài

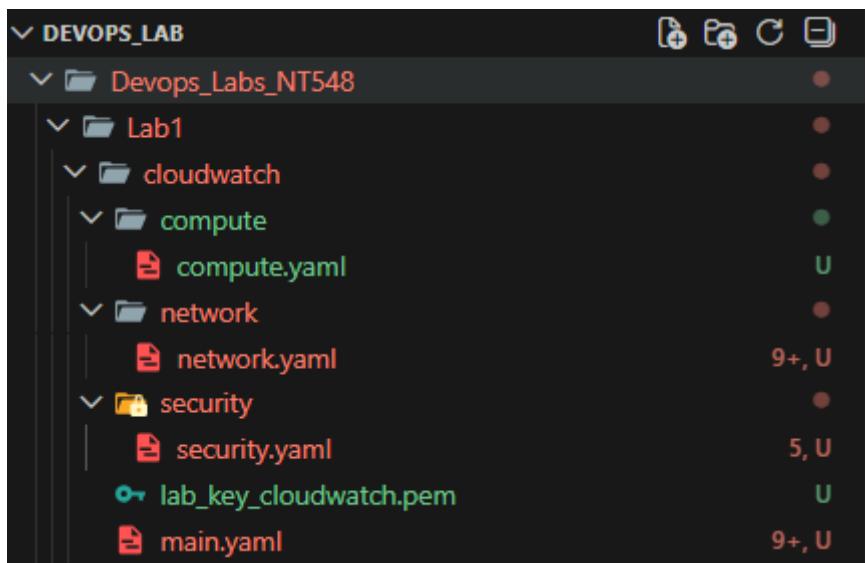
Câu 2: CloudFormation

Các bước thực hiện của nhóm bao gồm như sau:

1. Tạo thủ công Amazon S3 bucket (Để đẩy các file yaml cấu hình các module giống trên terraform lên AWS)

The screenshot shows the AWS S3 console interface. At the top, there's a green banner indicating "Successfully created bucket 'lab1-nt548-cfn-nhom17'". Below this, the main area displays "General purpose buckets (1)" with a single entry: "lab1-nt548-cfn-nhom17" from "US East (N. Virginia) us-east-1" created on "January 20, 2026, 20:49:03 (UTC+07:00)". To the right of the bucket list are two informational boxes: "Account snapshot" (updated daily) and "External access summary" (updated daily). At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App, along with standard footer links for Privacy, Terms, and Cookie preferences.

2. Cấu trúc thư mục khi sử dụng Cloudformation



3. Tạo stack và upload các file yaml lên bằng thủ công

- Create Stack
- Upload a template file
- Sửa parameters **S3BucketURL** với endpoint của S3 ở trên khi tạo thành công
“<https://lab1-nt548-cfn-nhom17.s3.us-east-1.amazonaws.com>”
- Nhấn Next và đợi các tài nguyên được tạo (Thành công thì nó sẽ có 4 stack tên là ComputeStack, SecurityStack, NetworkStack, và tên stack lab)
- Ở stack lab1-nt548-nhom17 mở tab output để có thể thấy được output cuối cùng khi các tài nguyên được tạo thành công)

Amazon S3

Upload succeeded
For more information, see the [Files and folders](#) table.

Summary

Destination
<s3://lab1-nt548-cfn-nhom17>

Succeeded
 3 files, 4.1 KB (100.00%)

Failed
 0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Files and folders (3 total, 4.1 KB)

Find by name							
Name	Folder	Type	Size	Status	Error		
security.yaml	-	-	922.0 B	Succeeded	-		
network.yaml	-	-	2.3 KB	Succeeded	-		
compute.yaml	-	-	943.0 B	Succeeded	-		

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Amazon S3

Upload succeeded

For more information, see the [Files and folders](#) table.

Summary

Destination
<s3://lab1-nt548-cfn-nhom17>

Succeeded
 3 files, 4.1 KB (100.00%)

Failed
 0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Files and folders (3 total, 4.1 KB)

Find by name							
Name	Folder	Type	Size	Status	Error		
security.yaml	-	-	922.0 B	Succeeded	-		
network.yaml	-	-	2.3 KB	Succeeded	-		
compute.yaml	-	-	943.0 B	Succeeded	-		

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

© 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Filter status

Search by stack name

Active ▾

View nested

◀ 1 ▶ | ⚙

Stacks

- NESTED**
lab1-nt548-nhom17-ComputeStack-V14ZQTBZ50ZO
2026-01-20 21:00:21 UTC+0700
✓ CREATE_COMPLETE
- NESTED**
lab1-nt548-nhom17-SecurityStack-16Y6FGY0L6ZQS
2026-01-20 20:59:46 UTC+0700
✓ CREATE_COMPLETE
- NESTED**
lab1-nt548-nhom17-NetworkStack-Z2OSJHYAXSHQ
2026-01-20 20:57:07 UTC+0700
✓ CREATE_COMPLETE
- lab1-nt548-nhom17**
2026-01-20 20:57:05 UTC+0700
✓ CREATE_COMPLETE

aws | Search [Alt+S] 🔍

United States (N. Virginia) hoangpdh_vpbank (0399-6820-5787) hoangpdh_vpbank

☰ CloudFormation > Stacks > lab1-nt548-nhom17

Filter status

Search by stack name

Active ▾

View nested

Stacks

- NESTED**
lab1-nt548-nhom17-ComputeStack-V14ZQTBZ50ZO
2026-01-20 21:00:21 UTC+0700
✓ CREATE_COMPLETE
- NESTED**
lab1-nt548-nhom17-SecurityStack-16Y6FGY0L6ZQS
2026-01-20 20:59:46 UTC+0700
✓ CREATE_COMPLETE
- NESTED**
lab1-nt548-nhom17-NetworkStack-Z2OSJHYAXSHQ
2026-01-20 20:57:07 UTC+0700
✓ CREATE_COMPLETE
- lab1-nt548-nhom17**
2026-01-20 20:57:05 UTC+0700
✓ CREATE_COMPLETE

Outputs (2)

Key	Value	Description	Export name
PrivateIP	10.0.2.66	-	-
PublicIP	98.93.204.28	-	-

4. Thực hiện các testcase giống như Terraform để kiểm tra khả năng truy cập Public EC2, ping Public -> Private và truy cập google.com từ Private

Ở đây nhóm em không thực hiện kiểm tra các tài nguyên trên AWS Console giống như Terraform mà chỉ thực hiện các testcase bởi vì thực hiện thành công các test case tương đương với thiết lập các tài nguyên thành công trên AWS Console.

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1\cloudwatch> ssh -i .\lab_key_cloudwatch.pem ec2-user@98.93.204.28
,      #
~\_ #####      Amazon Linux 2023
~~ \#####\
~~  \###|
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~     / 
~~.~.~ / 
~/ ~/
~/m'
Last login: Tue Jan 20 14:04:20 2026 from 222.253.48.126
[ec2-user@ip-10-0-1-93 ~]$ |
```

```
(base) PS E:\Project\Devops_Lab\Devops_Labs_NT548\Lab1\cloudwatch> ssh -i .\lab_key_cloudwatch.pem ec2-user@98.93.204.28
,      #
~\_ #####      Amazon Linux 2023
~~ \#####\
~~  \###|
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~     / 
~~.~.~ / 
~/ ~/
~/m'
Last login: Tue Jan 20 14:07:09 2026 from 222.253.48.126
[ec2-user@ip-10-0-1-93 ~]$ vi key.pem
[ec2-user@ip-10-0-1-93 ~]$ chmod 400 key.pem
[ec2-user@ip-10-0-1-93 ~]$ ssh -i key.pem ec2-user@10.0.2.66
,      #
~\_ #####      Amazon Linux 2023
~~ \#####\
~~  \###|
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '->
~~     / 
~~.~.~ / 
~/ ~/
~/m'
Last login: Tue Jan 20 14:06:47 2026 from 10.0.1.93
[ec2-user@ip-10-0-2-66 ~]$ |
```

```
[ec2-user@ip-10-0-2-66 ~]$ ping -c 4 google.com
PING google.com (142.251.179.102) 56(84) bytes of data.
64 bytes from pd-in-f102.1e100.net (142.251.179.102): icmp_seq=1 ttl=105 time=2.72 ms
64 bytes from pd-in-f102.1e100.net (142.251.179.102): icmp_seq=2 ttl=105 time=2.16 ms
64 bytes from pd-in-f102.1e100.net (142.251.179.102): icmp_seq=3 ttl=105 time=2.15 ms
64 bytes from pd-in-f102.1e100.net (142.251.179.102): icmp_seq=4 ttl=105 time=2.15 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.148/2.294/2.719/0.244 ms
[ec2-user@ip-10-0-2-66 ~]$ |
```

Kết quả:

- SSH vào được máy Public thông qua keypair được tạo thủ công trên AWS Console (Do cloudformation không tạo được keypair bằng code như Terraform)
- Ping thành công từ máy Public -> Private thông qua key.pem
- Từ máy private ping ra ngoài internet thành công (ping ra google.com)

Link github: https://github.com/hein-nkhh/Devops_Labs_NT548/tree/main/Lab1