

1. Use TF to create infra: VPC/Public subnet/Security group/EC2. Use Ansible to make the EC2 install Docker, pull image from your DockerHub and run a container from the image.

- All dir

```
duongtn1512@ansible1:~/CICD/Terraform$ tree
.
├── ansible
│   ├── inventory
│   │   └── lap
│   └── playbook
│       ├── install-docker.yml
│       ├── install-jenkins-container.yml
│       └── install-nginx.yml
├── inventory.tf
├── key
│   ├── key5.pem
│   └── key5.pub
├── main.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── variables.tf

4 directories, 12 files
```

- File main.tf

```
ssh powershell x main.tf U x inventory.tf U bucketS3.tf 3, U ! Docker.yml U
CICD > final > main.tf > resource "aws_security_group" "example" > description
1  # Define the AWS provider
2  provider "aws" {
3    region = "ap-southeast-1"
4  }
5  # Create a VPC
6  resource "aws_vpc" "lap" {
7    cidr_block = "10.0.0.0/16"
8  }
9  # Create a public subnet
10 resource "aws_subnet" "public" {
11   vpc_id = aws_vpc.lap.id
12   cidr_block = "10.0.1.0/24"
13   map_public_ip_on_launch = true
14 }
15 # Create SG
16 resource "aws_security_group" "example" {
17   vpc_id = aws_vpc.lap.id
18   name = "example-sg"
19   description = "Security group for the EC2 instance"
20
21   # Allow SSH, HTTP, and HTTPS traffic:
22   ingress {
23     from_port = 22
24     to_port = 22
25     protocol = "tcp"
26     cidr_blocks = ["0.0.0.0/0"]
27   }
28   ingress {
29     from_port = 80
30     to_port = 80
31     protocol = "tcp"
32     cidr_blocks = ["0.0.0.0/0"]
33   }
34 }
```

```

32     cidr_blocks = ["0.0.0.0/0"]
33 }
34 ingress {
35     from_port = 443 # Port 443 for HTTPS
36     to_port   = 443
37     protocol  = "tcp"
38     cidr_blocks = ["0.0.0.0/0"]
39 }
40 }
41
42
43 # Tạo máy chủ EC2
44 resource "aws_instance" "main" {
45     ami           = "ami-0df7a207adb9748c7" # AMI ID của Ubuntu
46     instance_type = "t2.medium" # Instance type của tôi
47     key_name      = aws_key_pair.fast.key_name # Key pair của tôi
48     security_groups = [aws_security_group.example.id]
49     subnet_id     = aws_subnet.public.id
50     tags = {
51         Name = "Terraform1"
52     }
53 }
54
55 # Để tạo key pair, ta sử dụng resource aws_key_pair trong tệp Terraform.
56 resource "aws_key_pair" "fast" {
57     key_name   = "key5.pub" # Tên key pair
58     public_key = file("key/key5.pub") # Đường dẫn đến file public key
59 }
60

```

Terraform used the selected providers to generate the following execution plan.
+ create

Terraform will perform the following actions:

```

# aws_instance.main will be created
+ resource "aws_instance" "main" {
+   ami                  = "ami-0df7a207adb9748c7"
+   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)

```

```

# aws_key_pair.fast will be created
+ resource "aws_key_pair" "fast" {
  + arn                = (known after apply)
  + fingerprint       = (known after apply)
  + id                 = (known after apply)
  + key_name           = "key5.pub"
  + key_name_prefix    = (known after apply)
  + key_pair_id        = (known after apply)
  + key_type           = (known after apply)
  + public_key         = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC8lN/AC2H5jPex1Dse6g4X
4Dg2xKVYgiwKisFOmmReFredwRSBYhEbtHS4oRCRnT4bCJ7YFaG4ugKLziFR6QJjTiFhWedj0jJAFIa0o08/IGF
QnMLQj6I9tVUog8aG+zZHN41xvRfCbAkPydMgvU3HS7iA08qIod/HfKzprodcPRA4B8wT9NNAHU10gD+EAE21pg
  + tags_all           = (known after apply)
}

# aws_security_group.example will be created
+ resource "aws_security_group" "example" {
  + arn                = (known after apply)
  + description        = "Security group for the EC2 instance"
  + egress              = (known after apply)
  + id                 = (known after apply)
  + ingress             = [
    + {
      + cidr_blocks      = [
        + "0.0.0.0/0",
      ]
      + description      = ""
      + from_port         = -1
      + ipv6_cidr_blocks = [
        + ":::/0",
      ]
    }
  ]
}

```

```

# aws_subnet.public will be created
+ resource "aws_subnet" "public" {
  + arn                = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone   = (known after apply)
  + availability_zone_id = (known after apply)
  + cidr_block          = "10.0.1.0/24"
  + enable_dns64        = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                  = (known after apply)
  + ipv6_cidr_block_association_id = (known after apply)
  + ipv6_native          = false
  + map_public_ip_on_launch = true
  + owner_id             = (known after apply)
  + private_dns_hostname_type_on_launch = (known after apply)
  + tags_all             = (known after apply)
  + vpc_id               = (known after apply)
}

```

```

# aws_vpc.lap will be created
+ resource "aws_vpc" "lap" {
  + arn                = (known after apply)
  + cidr_block          = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
}

```

```
# local_file.ansible_inventory will be created
+ resource "local_file" "ansible_inventory" {
  + content          = (known after apply)
  + 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   = "0777"
  + filename          = "./ansible/inventory/lap"
  + id                = (known after apply)
}

# null_resource.playbook_exec will be created
+ resource "null_resource" "playbook_exec" {
  + id          = (known after apply)
  + triggers = {
    + "key" = (known after apply)
  }
}
```

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ aws_instance_main_private_ip = [
  + (known after apply),
]
```

- File inventory

```
CICD > final > inventory.tf > resource "local_file" "ansible_inventory" > content
1  # Ghi lại ip của ec2 mỗi tạo vào file inventory/lab của ansible
2  resource "local_file" "ansible_inventory" {
3    filename = var.ansible_host_path
4    content  = <<-EOT
5    [lap]
6    %{for ip in aws_instance.main.*.public_ip~}
7    ${ip} ansible_host=${ip} ansible_user=${var.ansible_user} \
8    ansible_ssh_private_key_file=${var.ansible_wsl_private_key_path} \
9    ansible_ssh_common_args='-o StrictHostKeyChecking=no' ansible_ssh_connection=ssh
10   %{endfor~}
11   EOT
12 }
13 # Kích hoạt cho ansible chạy playbook với host ip mới lấy được
14 resource "null_resource" "playbook_exec" {
15   triggers = {
16     key = uuid()
17   }
18
19   provisioner "local-exec" {
20     command = <<EOF
21     ansible-playbook ${var.ansible_command} -i ${var.ansible_wsl_host_path}
22     EOF
23   }
24   depends_on = [aws_instance.main, local_file.ansible_inventory]
25 }
26
```

- File variable

```
CICD > final > variables.tf > ...
6   }
7
8   variable "ansible_ssh_private_key_file" {
9     type           = string
10    description = "ssh key file to use for ansible_user"
11    default      = "../key/key5.pem"
12  }
13
14
15  variable "ansible_ssh_public_key_file" {
16    type           = string
17    description = "ssh public key in server authorized_keys"
18    default      = "../key/key5.pub"
19  }
20
21  variable "ansible_host_path" {
22    type           = string
23    description = "path to ansible inventory host"
24    default      = "../ansible/inventory/lap"
25  }
26
27  variable "ansible_command" {
28    default      = "../ansible/playbook/install-docker.yml"
29    description = "Command for container lab hosts"
30  }
31  variable "ansible_wsl_host_path" {
32    default      = "../ansible/inventory/lap"
33    description = "Command for container lab hosts"
34  }
35  variable "ansible_wsl_private_key_path" {
36    default      = "../key/key5.pem"
37    description = "Command for container lab hosts"
38  }
```

- All dir

```
duongtn1512@ansible1:~/CICD/Terraform$ tree
.
├── ansible
│   ├── inventory
│   │   └── lap
│   └── playbook
│       ├── install-docker.yml
│       ├── install-jenkins-container.yml
│       └── install-nginx.yml
├── inventory.tf
├── key
│   ├── key5.pem
│   └── key5.pub
├── main.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── variables.tf

4 directories, 12 files
```

- File install-docker.yml

```
1
2 - name: Install Common
3   hosts: all
4   gather_facts: true
5   become: true
6   tasks:
7     # Install Docker
8     - name: Install Container Engine
9       shell: |
10         sudo apt-get update
11         sudo apt-get install -y ca-certificates curl gnupg lsb-release
12         curl -fsSL https://download.docker.com/linux/debian/gpg | sudo
13         echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/sha
14
15         sudo apt-get install docker.io docker-compose -y
16         systemctl enable docker.service
17         systemctl start docker.service
18
19         cat <<EOF | sudo tee /etc/docker/daemon.json
20         {
21           "exec-opts": ["native.cgroupdriver=systemd"],
22           "log-driver": "json-file",|
23           "log-opts": {
24             "max-size": "100m"
25           },
26           "storage-driver": "overlay2"
27         }
28         EOF
29         sudo systemctl enable docker
30         sudo systemctl daemon-reload
31         sudo systemctl restart docker
32         sudo chmod 777 /var/run/docker.sock
```

```
34   # Run a image
35   - name: Pull image form dockerhub repo
36     shell: |
37       docker pull duongtn1512/random_game:1cf3f88
38       docker run -d --name exe1 -p 8080:80 duongtn1512/random_game:1cf3f88
39
```



```

null_resource.playbook_exec: Provisioning with 'local-exec'...
null_resource.playbook_exec (local-exec): Executing: ["/bin/sh" "-c" "      ansible-playbo
"]

null_resource.playbook_exec (local-exec): PLAY [Install Common] *****

null_resource.playbook_exec (local-exec): TASK [Gathering Facts] *****
null_resource.playbook_exec (local-exec): ok: [18.136.197.37]

null_resource.playbook_exec (local-exec): TASK [Install Container Engine] *****
null_resource.playbook_exec: Still creating... [10s elapsed]
null_resource.playbook_exec: Still creating... [20s elapsed]
null_resource.playbook_exec: Still creating... [30s elapsed]
null_resource.playbook_exec: Still creating... [40s elapsed]
null_resource.playbook_exec: Still creating... [50s elapsed]
null_resource.playbook_exec (local-exec): changed: [18.136.197.37]

null_resource.playbook_exec (local-exec): TASK [Pull image form dockerhub repo] *****
null_resource.playbook_exec: Still creating... [1m0s elapsed]
null_resource.playbook_exec: Still creating... [1m10s elapsed]
null_resource.playbook_exec (local-exec): changed: [18.136.197.37]

null_resource.playbook_exec (local-exec): PLAY RECAP *****
null_resource.playbook_exec (local-exec): 18.136.197.37      : ok=3    changed=2

null_resource.playbook_exec: Creation complete after 1m11s [id=9068924301385412419]

```

```

ec2_global_ips = [
    [
        "18.136.197.37",
    ],
]

```

```

duongtn1512@ansible1:~/CICD/Terraform$ ssh -i key/key5.pem ubuntu@18.136.197.37
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

```

```

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

```

System information as of Mon Oct 9 13:02:23 UTC 2023

```

System load:  0.1796875      Processes:            122
Usage of /:   30.1% of 7.57GB Users logged in:          0
Memory usage: 9%             IPv4 address for docker0: 172.17.0.1
Swap usage:   0%             IPv4 address for eth0:   172.31.20.183

```

Expanded Security Maintenance for Applications is not enabled.

130 updates can be applied immediately.
73 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Mon Oct 9 13:00:24 2023 from 116.96.46.78

```

ubuntu@ip-172-31-20-183:~$ docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
S				
e5b4bb1515e1	duongtn1512/random_game:1cf3f88	"/docker-entrypoint...."	About a minute ago	Up About a minute

```

ubuntu@ip-172-31-20-183:~$

```

2. Use TF to add S3 resource and then push an index.html onto the S3 (using TF code to push, not the GUI) without destroying the previous resources and also add IAM role for to EC2 to access the S3's data. Use Ansible to stop the running container, install nginx, config the default index.html to the one stored on S3.

- IAM role

```
42 resource "aws_iam_role" "s3_access_role" {
43   name = "EC2S3AccessRole"
44
45   assume_role_policy = jsonencode({
46     Version = "2012-10-17",
47     Statement = [
48       {
49         Action = "sts:AssumeRole",
50         Effect = "Allow",
51         Principal = {
52           Service = "ec2.amazonaws.com"
53         }
54       }
55     ]
56   })
57 }
58
59 # Tạo máy chủ EC2
60 resource "aws_instance" "main" {
61   ami           = "ami-0df7a207adb9748c7" # AMI ID của Ubuntu
62   instance_type = "t2.medium" # Instance type của tôi
63   key_name      = aws_key_pair.fast.key_name # Key pair của tôi
64   security_groups = [aws_security_group.example.id]
65   subnet_id     = aws_subnet.public.id
66   iam_instance_profile = aws_iam_role.s3_access_role.name
67   tags = {
68     Name = "Terraform1"
69   }
70 }
```

- S3 terraform

```
41 # Create an S3 bucket
42 resource "aws_s3_bucket" "example" {
43   bucket = "lap-final" # Replace with a unique bucket name
44   acl    = "public-read" # Set the ACL to allow public read access (for hosting a website)
45 }
46
47 # Upload the index.html file to the S3 bucket
48 resource "aws_s3_bucket_object" "index" {
49   bucket = aws_s3_bucket.example.id
50   key    = "index.html"
51   source = "./data/index.html" # Replace with the path to your index.html file
52   content_type = "text/html"
53   acl        = "public"
54 }
```


- Nginx-site-config.conf.j2 store at data

```
CICD > final > data > nginx-site-config.conf.j2
1  server {
2      listen 80 default_server;
3      server_name _;
4
5      location / {
6          proxy_pass https://lap-final.s3.ap-southeast-1.amazonaws.com/index.html
7          proxy_set_header Host $host;
8          proxy_set_header X-Real-IP $remote_addr;
9          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
10     }
11 }
12
```

- New playbook to Use Ansible to stop the running container, install nginx, configthe default index.html to the one stored on S3.

```
CICD > final > ansible > playbook > ! Nginx-s3.yml > {} 0 > [ ] tasks > {} 1
1  ---
2  - name: Stop Container, Install Nginx, and Configure Index.html
3    hosts: your_ec2_instance
4    become: yes
5    tasks:
6      - name: Stop the running container (replace with your container name)
7        command: docker stop your_container_name
8        ignore_errors: yes
9        async: 60 # Give it some time to stop gracefully
10       poll: 0
11
12      - name: Wait for the container to stop
13        async_status:
14          jid: "{{ ansible_job_id }}"
15        register: job_result
16        until: job_result.finished
17        retries: 30
18        delay: 10
19
20      - name: Install Nginx
21        apt:
22          name: nginx
23          state: present
24
25      - name: Configure Nginx default site to use S3-hosted index.html
26        template:
27          src: /data/nginx-site-config.conf.j2
28          dest: /etc/nginx/sites-available/default
29        notify:
30          - Reload Nginx
31
```

```

19 |
20 |     - name: Install Nginx
21 |       apt:
22 |         name: nginx
23 |         state: present
24 |
25 |     - name: Configure Nginx default site to use S3-hosted index.html
26 |       template:
27 |         src: /data/nginx-site-config.conf.j2
28 |         dest: /etc/nginx/sites-available/default
29 |       notify:
30 |         - Reload Nginx
31 |
32 |     - name: Ensure Nginx is started and enabled
33 |       service:
34 |         name: nginx
35 |         state: started
36 |         enabled: yes
37 |
38 |     handlers:
39 |       - name: Reload Nginx
40 |         service:
41 |           name: nginx
42 |           state: reloaded
43 |

```

- Dir

```
duongtn1512@ansible1:~/CICD/Terraform$ tree
```

```

.
├── ansible
│   ├── inventory
│   │   └── lap
│   └── playbook
│       ├── install-docker.yml
│       ├── install-jenkins-container.yml
│       ├── install-nginx.yml
│       └── Nginx-s3.yml
├── data
│   ├── index.html
│   └── nginx-site-config.conf.j2
├── inventory.tf
├── key
│   ├── key5.pem
│   └── key5.pub
├── main.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── variables.tf

```

5 directories, 15 files

```
duongtn1512@ansible1:~/CICD/Terraform$
```

- Change in variable.tf

```

variable "ansible_command" {
  default      = "./ansible/playbook/Nginx-s3.yml"
  description = "Command for container lab hosts"
}

```

- terraform apply --auto-approve

Terraform will perform the following actions:

```
# aws_iam_role.s3_access_role will be created
+ resource "aws_iam_role" "s3_access_role" {
  + arn                = (known after apply)
  + assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action    = "sts:AssumeRole"
          + Effect    = "Allow"
          + Principal = {
            + Service = "ec2.amazonaws.com"
          }
        },
      ]
    }
  + Version = "2012-10-17"
}

+ create_date      = (known after apply)
+ force_detach_policies = false
+ id               = (known after apply)
+ managed_policy_arns = (known after apply)
+ max_session_duration = 3600
+ name             = "EC2S3AccessRole"
+ name_prefix      = (known after apply)
+ path             = "/"
+ tags_all         = (known after apply)
+ unique_id        = (known after apply)
```

```
# aws_s3_bucket.example will be created
+ resource "aws_s3_bucket" "example" {
  + acceleration_status = (known after apply)
  + acl                 = "public-read"
  + arn                 = (known after apply)
  + bucket              = "your-unique-bucket-name"
  + bucket_domain_name = (known after apply)
  + bucket_prefix       = (known after apply)
  + bucket_regional_domain_name = (known after apply)
  + force_destroy       = false
  + hosted_zone_id      = (known after apply)
  + id                  = (known after apply)
  + object_lock_enabled = (known after apply)
  + policy              = (known after apply)
  + region              = (known after apply)
  + request_payer       = (known after apply)
  + tags_all            = (known after apply)
  + website_domain      = (known after apply)
  + website_endpoint    = (known after apply)
}
```

```
# aws_s3_bucket_object.index will be created
+ resource "aws_s3_bucket_object" "index" {
  + acl           = "public-read"
  + bucket        = (known after apply)
  + bucket_key_enabled = (known after apply)
  + content_type  = "text/html"
  + etag          = (known after apply)
  + force_destroy = false
  + id            = (known after apply)
  + key           = "index.html"
  + kms_key_id    = (known after apply)
  + server_side_encryption = (known after apply)
  + source        = "../data/index.html"
  + storage_class = (known after apply)
```

```
# null_resource.playbook_exec must be replaced
-/+ resource "null_resource" "playbook_exec" {
  ~ id          = "9068924301385412419" -> (known after apply)
  ~ triggers = { # forces replacement
    ~ "key" = "99339c0f-6c55-c513-297e-228326ec8622" -> (known after apply)
  }
}
```

Plan: 4 to add, 1 to change, 1 to destroy.

Changes to Outputs:

```
+ bucket_url          = (known after apply)
null_resource.playbook_exec: Destroying... [id=9068924301385412419]
null_resource.playbook_exec: Destruction complete after 0s
aws_iam_role.s3_access_role: Creating...
aws_s3_bucket.example: Creating...
aws_iam_role.s3_access_role: Creation complete after 5s [id=EC2S3AccessRole]
aws_instance.main: Modifying... [id=i-071bc7c895c589023]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 10s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 20s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 30s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 40s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 50s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m0s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m10s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m20s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m30s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m40s elapsed]
aws_instance.main: Still modifying... [id=i-071bc7c895c589023, 1m50s elapsed]
```

Amazon S3 > Buckets > lap-final

lap-final Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

<

1

>

	Name	Type	Last modified	Size	Storage class
	<div><div></div>index.html</div>	html	October 9, 2023, 20:46:18 (UTC+07:00)	1.5 KB	Standard

3. Destroy TF resource and rewrite TF code with the following requirements:

```
duongtn1512@ansible1:~/CICD/Terraform$ terraform destroy --auto-approve
aws_iam_role.s3_access_role: Refreshing state... [id=EC2S3AccessRole]
aws_key_pair.fast: Refreshing state... [id=key5.pub]
aws_instance.main: Refreshing state... [id=i-071bc7c895c589023]
local_file.ansible_inventory: Refreshing state... [id=baa03ae6021dfed8fd78a206f1877d8c227e71c1]
```

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

- destroy

Terraform will perform the following actions:

```
# aws_iam_role.s3_access_role will be destroyed
- resource "aws_iam_role" "s3_access_role" {
  - arn              = "arn:aws:iam::125777342244:role/EC2S3AccessRole" -> null
  - assume_role_policy = jsonencode(
    {
      - Statement = [
        - {
          - Action      = "sts:AssumeRole"
          - Effect      = "Allow"
          - Principal = {
            - Service = "ec2.amazonaws.com"
          }
        }
      ]
    }
  ) -> null
  - create_date      = "2023-10-09T13:25:15Z" -> null
  - force_detach_policies = false -> null
  - id                = "EC2S3AccessRole" -> null
  - managed_policy_arns = [] -> null
  - max_session_duration = 3600 -> null
```

- Nginx app running on t2-micro or t3-micro with custom index.html (stored on S3)

```
ssh  Nginx-s3.yml U  nginx-deploy.yml U X  Docker.yml U
CICD > final > ansible > playbook > ! nginx-deploy.yml > {} 0 > [ ] tasks > {} 1
1  ---
2  - name: Deploy Nginx with Custom Index.html from S3
3    hosts: all
4    become: yes
5
6    tasks:
7      - name: Update package cache and install Nginx
8        apt:
9          name: nginx
10         state: present
11         become: yes
12
13     - name: Stop Nginx service (if already running)
14       service:
15         name: nginx
16         state: stopped
17         ignore_errors: yes
18
19     - name: Configure Nginx default site to use S3-hosted index.html
20       template:
21         src: /data/nginx-site-config.conf.j2
22         dest: /etc/nginx/sites-available/default
23       notify:
24         - Restart Nginx
25
26     - name: Enable Nginx default site
27       file:
28         src: /etc/nginx/sites-available/default
29         dest: /etc/nginx/sites-enabled/default
30         state: link
31       notify:
32         - Restart Nginx
```

- nginx-site-config.conf.j2 to nginx server use s3 url

```
CICD > final > data > nginx-site-config.conf.j2
```

```
1  server {
2      listen 80 default_server;
3      server_name _;
4
5      location / {
6          proxy_pass https://lap-final.s3.ap-southeast-1.amazonaws.com/index.html
7          proxy_set_header Host $host;
8          proxy_set_header X-Real-IP $remote_addr;
9          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
10     }
11 }
12 |
```

- Application load-balancing for the web-servers

```
40  - name: Configure Application Load Balancer
41  hosts: localhost
42  gather_facts: no
43  tasks:
44  - name: Create Application Load Balancer
45    community.aws.elb_application_lb:
46      name: my-alb
47      state: present
48      subnets:
49        - subnet-1a2b3c4d
50        - subnet-5e6f7g8h
51      security_groups:
52        - sg-12345678
53      listeners:
54        - protocol: HTTP
55          port: 80
56          default_action:
57            type: fixed-response
58            fixed_response_type: content-type-plaintext
59            fixed_response_content_type: text/plain
60            fixed_response_message_body: "Hello from the ALB!"
61      wait: yes
62      register: alb_result
63
64  - name: Print ALB DNS Name
65    debug:
66      var: alb_result.elb_dns_name
67 |
```


- No userdata, only use Ansible.

```
61 resource "aws_iam_role" "s3_access_role" {
62   name = "EC2S3AccessRole"
63
64   assume_role_policy = jsonencode({
65     Version = "2012-10-17",
66     Statement = [
67       {
68         Action = "sts:AssumeRole",
69         Effect = "Allow",
70         Principal = {
71           Service = "ec2.amazonaws.com"
72         }
73       }
74     ]
75   })
76 }
77
78 # Tạo máy chủ EC2
79 resource "aws_instance" "main" {
80   ami           = "ami-0df7a207adb9748c7" # AMI ID của Ubuntu
81   instance_type = "t2.medium" # Instance type của tôi
82   key_name      = aws_key_pair.fast.key_name # Key pair của tôi
83   # security_groups = [aws_security_group.example.id]
84   # subnet_id      = aws_subnet.public.id
85   iam_instance_profile = aws_iam_role.s3_access_role.name
86   tags = {
87     Name = "Terraform1"
88   }
89 }
```

- Terraform apply

```
duongtn1512@ansible1:~/CICD/Terraform$ 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_iam_role.s3_access_role will be created
+ resource "aws_iam_role" "s3_access_role" {
+   arn = (known after apply)
+   assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action   = "sts:AssumeRole"
          + Effect   = "Allow"
          + Principal = {
            + Service = "ec2.amazonaws.com"
          }
        },
      ]
    }
  )
+   create_date = (known after apply)
+   force_detach_policies = false
+   id = (known after apply)
```

```

# aws_instance.main will be created
+ resource "aws_instance" "main" {
  + ami                    = "ami-0df7a207adb9748c7"
  + 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    = "EC2S3AccessRole"
  + id                     = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle      = (known after apply)
  + instance_state          = (known after apply)
  + instance_type           = "t2.medium"
  + ipv6_address_count      = (known after apply)
  + ipv6_addresses          = (known after apply)
  + key_name                = "key5.pub"
  + 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)
  + public_dns              = (known after apply)
  + public_ip               = (known after apply)
  + secondary_private_ips   = (known after apply)
  + security_groups         = (known after apply)

```

```

# aws_key_pair.fast will be created
+ resource "aws_key_pair" "fast" {
  + arn                = (known after apply)
  + fingerprint       = (known after apply)
  + id                 = (known after apply)
  + key_name           = "key5.pub"
  + key_name_prefix    = (known after apply)
  + key_pair_id        = (known after apply)
  + key_type           = (known after apply)
  + public_key         = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA8IN/AC2H5jPex1Dse6g4Xv9HP
4Dg2xKVYgiWkIsFOmmReFredwRSBYhEbtHS4oRCRnT4bCJ7YFaG4ugKLziFR6QJjTiFhWedj0jJAFIaOo08/IGF1Cez
QnMLQj6I9tVUog8aG+zZHN41xvRFCbAkPydMgvU3HS7iaO8qIod/HfKzprodCPRa4B8wT9NNAHU10gD+EAE21pgal a
  + tags_all          = (known after apply)
}

```

```

# local_file.ansible_inventory will be created
+ resource "local_file" "ansible_inventory" {
  + content                = (known after apply)
  + 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       = "0777"
  + filename              = "./ansible/inventory/lap"
  + id                    = (known after apply)
}

```

```

# null_resource.playbook_exec will be created
+ resource "null_resource" "playbook_exec" {

```

```
# null_resource.playbook_exec will be created
+ resource "null_resource" "playbook_exec" {
  + id          = (known after apply)
  + triggers = {
    + "key" = (known after apply)
  }
}
```

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ aws_instance_main_private_ip = [
  + (known after apply),
]
+ aws_instance_main_public_ip  = [
  + (known after apply),
]
+ ec2_global_ips               = [
  + [
    + (known after apply),
  ],
]
```

aws_iam_role.s3_access_role: Creating...

aws_key_pair.fast: Creating...

aws_key_pair.fast: Creation complete after 2s [id=key5.pub]

aws_iam_role.s3_access_role: Creation complete after 3s [id=EC2S3AccessRole]

aws_instance.main: Creating...

aws_instance.main: Still creating... [10s elapsed]

aws_instance.main: Still creating... [20s elapsed]

aws_instance.main: Still creating... [30s elapsed]

aws_instance.main: Still creating... [40s elapsed]

- Đợi khá là lâu giờ còn 5'

```
aws_instance.main: Creating...
aws_instance.main: Still creating... [10s elapsed]
aws_instance.main: Still creating... [20s elapsed]
aws_instance.main: Still creating... [30s elapsed]
aws_instance.main: Still creating... [40s elapsed]
aws_instance.main: Still creating... [50s elapsed]
aws_instance.main: Still creating... [1m0s elapsed]
aws_instance.main: Still creating... [1m10s elapsed]
aws_instance.main: Still creating... [1m20s elapsed]
aws_instance.main: Still creating... [1m30s elapsed]
```

- Lỗi kết nối mạng

TASK [Gathering Facts] *****

fatal: [18.136.197.37]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 18.136.197.37 port 22: Connection timed out", "unreachable": true}

PLAY RECAP *****

18.136.197.37 : ok=0 changed=0 unreachable=1 failed=0 skipped=0 rescued=0 ignored=0

Error: local-exec provisioner error

```
with null_resource.playbook_exec,  
on inventory.tf line 17, in resource "null_resource" "playbook_  
17:   provisioner "local-exec" {
```

Error running command ' ansible-playbook ./ansible/playbook_
' : exit status 4. Output:

PLAY [Install Common] *****

TASK [Gathering Facts] *****

fatal: [18.136.197.37]: UNREACHABLE! => {"changed": false, "msg"
Connection timed out", "unreachable": true}

PLAY RECAP *****

18.136.197.37 : ok=0 changed=0 unreachable=1