

Graduation Project  
DEPI\_1\_CAI1\_ISS4\_G1e AWS Cloud  
Solution Admin & Architect

**Deploying a Highly Available Web Application with  
Auto Scaling using IAC**

Supervised by:

**Dr. Ibrahim Gomaa**

Eng. Tarek Elkabani

Eng Mohamed Kababi

Student Name: Bassam Amr Fathy

Student Number: 1122141562

Email Address: [bassam11287@gmail.com](mailto:bassam11287@gmail.com)

## **Project Objective**

To deploy a highly available web application on AWS using Terraform for easy deployment, with an Application Load Balancer (ALB) distributing traffic across instances in an Auto Scaling Group (ASG). The deployment includes a bastion host for SSH access, S3 for backup storage, and secure network configurations using public and private subnets with a NAT gateway for outbound internet access from private instances.

## **Why Terraform:**

- Declarative configuration
- Support multi cloud providers
- Reusable infrastructure code
- Support version control
- Resource management

## **Infrastructure Setup**

### **1.AWS Provider:**

- Configured for the us-east-1 region to deploy the infrastructure.

### **2.VPC and Subnets:**

- A VPC with CIDR block 192.168.0.0/16 is created.
- Four subnets are defined: two for public resources (e.g., ALB, bastion host) and two for private instances (e.g., ASG instances).

### **3.Internet Gateway & NAT Gateway:**

- An Internet Gateway allows public internet access for the public subnets.
- A NAT Gateway provides outbound internet access to private instances via the NAT gateway in the public subnet.

### **4.Security Groups:**

- Security group for ALB and instances allows HTTP (port 80) and SSH (port 22) traffic from the internet.
- A dedicated security group for the bastion host allows SSH access, enabling the management of private instances through the bastion host.

### **5.Application Load Balancer (ALB):**

- The ALB listens on port 80 and forwards traffic to an Auto Scaling Group (ASG) through a target group. Health checks are performed on the root path to ensure instance availability.

## 6.Auto Scaling Group (ASG):

- The ASG dynamically scales between 1 to 3 instances, ensuring high availability. Each instance hosts a simple web server using apache that serves a web page for the project.
- The ASG is configured to use an EC2 Launch Configuration, which installs updates, including apache server.

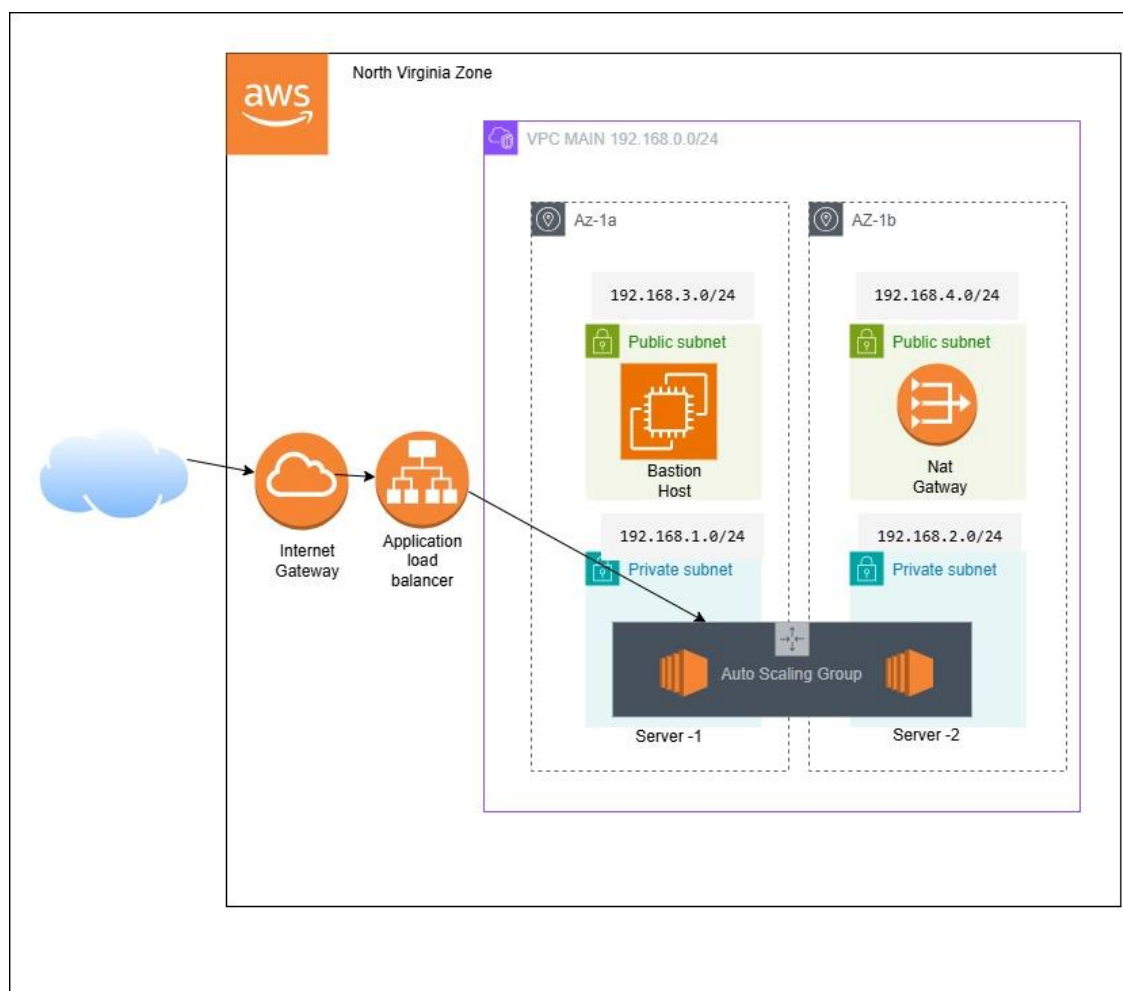
## 7.Bastion Host:

- A bastion host (EC2 instance) is deployed in the public subnet to securely SSH into the private EC2 instances.
- Users upload an SSH key to the bastion host for connecting to private instances.

## 8.Private Subnet EC2 Access and Internet Connectivity:

- From the bastion host, SSH access is established to instances in private subnets.
- Private instances can connect to the internet through the NAT Gateway to download necessary software and updates.

## Project Architecture



### Screenshot before project Deployment:

As shown below, before deployment the code stack all resources are empty except for the default of AWS lab learner

## Resources

EC2 Global View

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

### Security Groups (1) [Info](#)

**Actions** **Export security groups to CSV** **Create security group**

< 1 >

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	<a href="#">sg-09f05ff7a796f87ff</a>	default	<a href="#">vpc-04db7e2344e86161e</a>	default VPC se

[EC2](#) > Load balancers

### Load balancers

**Actions** **Create load balancer**

< 1 >

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
No load balancers						
You don't have any load balancers in us-east-1						

Create load balancer

## **Code Stack:**

### **VPC Configuration**

```
resource "aws_vpc" "main" {  
  cidr_block = "192.168.0.0/16"  
  
  tags = {  
    Name = "main-vpc"  
  }  
}
```

- **Resource: aws\_vpc.main**
- **Description: Creates a Virtual Private Cloud (VPC) with a CIDR block of 192.168.0.0/16.**
- **Tags: Names the VPC as main-vpc.**

### **Subnet Configuration**

```
resource "aws_subnet" "subnet1" {  
  vpc_id      = aws_vpc.main.id  
  cidr_block   = "192.168.1.0/24"  
  availability_zone = "us-east-1a"  
  
  tags = {  
    Name = "priv_subnet-1"  
  }  
}  
  
resource "aws_subnet" "subnet2" {  
  vpc_id      = aws_vpc.main.id  
  cidr_block   = "192.168.2.0/24"  
  availability_zone = "us-east-1b"  
  
  tags = {  
    Name = "priv_subnet-2"  
  }  
}  
  
resource "aws_subnet" "subnet3" {  
  vpc_id      = aws_vpc.main.id  
  cidr_block   = "192.168.3.0/24"
```

```

availability_zone = "us-east-1a"

tags = {
  Name = "pub_subnet-1"
}

resource "aws_subnet" "subnet4" {
  vpc_id      = aws_vpc.main.id
  cidr_block   = "192.168.4.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "pub_subnet-2"
  }
}

```

- **Resources:** aws\_subnet.subnet1, aws\_subnet.subnet2, aws\_subnet.subnet3, aws\_subnet.subnet4
- **Description:**
  - **Private Subnets:**
    - subnet1: 192.168.1.0/24 in us-east-1a
    - subnet2: 192.168.2.0/24 in us-east-1b
  - **Public Subnets:**
    - subnet3: 192.168.3.0/24 in us-east-1a
    - subnet4: 192.168.4.0/24 in us-east-1b

## **Internet Gateway and NAT Gateway**

```
resource "aws_internet_gateway" "main" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "main-gateway"
  }
}

resource "aws_eip" "nat_eip" {
  tags = {
    Name = "nat-eip"
  }
}

resource "aws_nat_gateway" "nat_gw" {
  allocation_id = aws_eip.nat_eip.id
  subnet_id     = aws_subnet.subnet4.id

  tags = {
    Name = "nat-gateway"
  }
}
```

- **Resources:**
  - **aws\_internet\_gateway.main: Creates an Internet Gateway attached to the VPC.**
  - **aws\_eip.nat\_eip: Allocates an Elastic IP for the NAT Gateway.**
  - **aws\_nat\_gateway.nat\_gw: Creates a NAT Gateway in subnet4 using the allocated EIP.**

## **Route Tables**

```
resource "aws_route_table" "public_RT" {
  vpc_id = aws_vpc.main.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.main.id
  }

  tags = {
    Name = "public_RT"
  }
}

resource "aws_route_table" "private_RT" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "private_RT"
  }
}

resource "aws_route" "private_route" {
  route_table_id      = aws_route_table.private_RT.id
  destination_cidr_block = "0.0.0.0/0"
  nat_gateway_id      = aws_nat_gateway.nat_gw.id
}
```

- **Resources:**
  - **aws\_route\_table.public\_RT: Route table for public subnets with a default route to the Internet Gateway.**
  - **aws\_route\_table.private\_RT: Route table for private subnets.**
  - **aws\_route.private route: Adds a default route to the NAT Gateway for private subnets.**



## **Security Groups**

```
resource "aws_security_group" "HTTP_SG" {
  vpc_id = aws_vpc.main.id

  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "HTTP-SG"
  }
}

resource "aws_security_group" "bastion_SG" {
  vpc_id = aws_vpc.main.id

  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
```

```

from_port = 0
to_port   = 0
protocol  = "-1"
cidr_blocks = ["0.0.0.0/0"]
}

tags = {
  Name = "bastion-SG"
}
}

```

- **Resources:**
  - aws\_security\_group.HTTP\_SG: Allows HTTP (port 80) and SSH (port 22) inbound traffic from anywhere. Allows all outbound traffic.
  - aws\_security\_group.bastion\_SG: Allows SSH (port 22) inbound traffic from anywhere. Allows all outbound traffic.

## **EC2 Instances**

```

resource "aws_instance" "bastion_host" {
  ami           = "ami-0fff1b9a61dec8a5f"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.subnet3.id
  vpc_security_group_ids = [aws_security_group.bastion_SG.id]
  associate_public_ip_address = true
  key_name      = "vockey"

  tags = {
    Name = "bastion-host"
  }

  depends_on = [aws_security_group.bastion_SG]
}

```

- **Resource:** aws\_instance.bastion\_host
- **Description:** Launches a bastion host in the public subnet (subnet3).
- **Configuration:**
  - AMI: ami-0fff1b9a61dec8a5f
  - Instance Type: t2.micro
  - Security Group: Attached to bastion\_SG
  - Public IP: Associated for internet access
  - SSH Key: Uses vockey for SSH access
- **Tags:** Named bastion-host.
- **Dependencies:** Depends on the creation of the bastion\_SG security group.

## **Load Balancer**

```
resource "aws_lb" "test" {
  name           = "bassam-alb"
  internal       = false
  load_balancer_type = "application"
  security_groups = [aws_security_group.HTTP_SG.id]
  subnets       = [aws_subnet.subnet3.id, aws_subnet.subnet4.id]

  enable_deletion_protection = false

  tags = {
    Name = "bassam-alb"
  }
}
```

- **Resource: aws\_lb.test**
- **Description: Creates an Application Load Balancer (ALB) named bassam-alb.**
- **Configuration:**
  - **Type: Application Load Balancer**
  - **Security Groups: Attached to HTTP\_SG**
  - **Subnets: Deployed in both public subnets (subnet3 and subnet4)**
  - **Deletion Protection: Disabled**
- **Tags: Named bassam-alb.**

## **Auto Scaling Group**

```
resource "aws_launch_configuration" "app" {
  name           = "app-launch-configuration"
  image_id       = "ami-0fff1b9a61dec8a5f"
  instance_type  = "t2.micro"
  key_name       = "vockey"
  security_groups = [aws_security_group.HTTP_SG.id]

  user_data = <<-EOF
    #!/bin/bash
    sudo yum update -y
    sudo yum install httpd -y
    sudo systemctl start httpd
    sudo systemctl enable httpd
  EOF
}

resource "aws_autoscaling_group" "app" {
  launch_configuration = aws_launch_configuration.app.id
  min_size             = 1
  max_size             = 3
  desired_capacity      = 2
  vpc_zone_identifier   = [aws_subnet.subnet1.id, aws_subnet.subnet2.id]

  target_group_arns = [aws_lb_target_group.test.arn]

  tag {
    key      = "Name"
    value    = "ASG_Instance"
    propagate_at_launch = true
  }

  lifecycle {
    ignore_changes = [desired_capacity]
  }
}
```

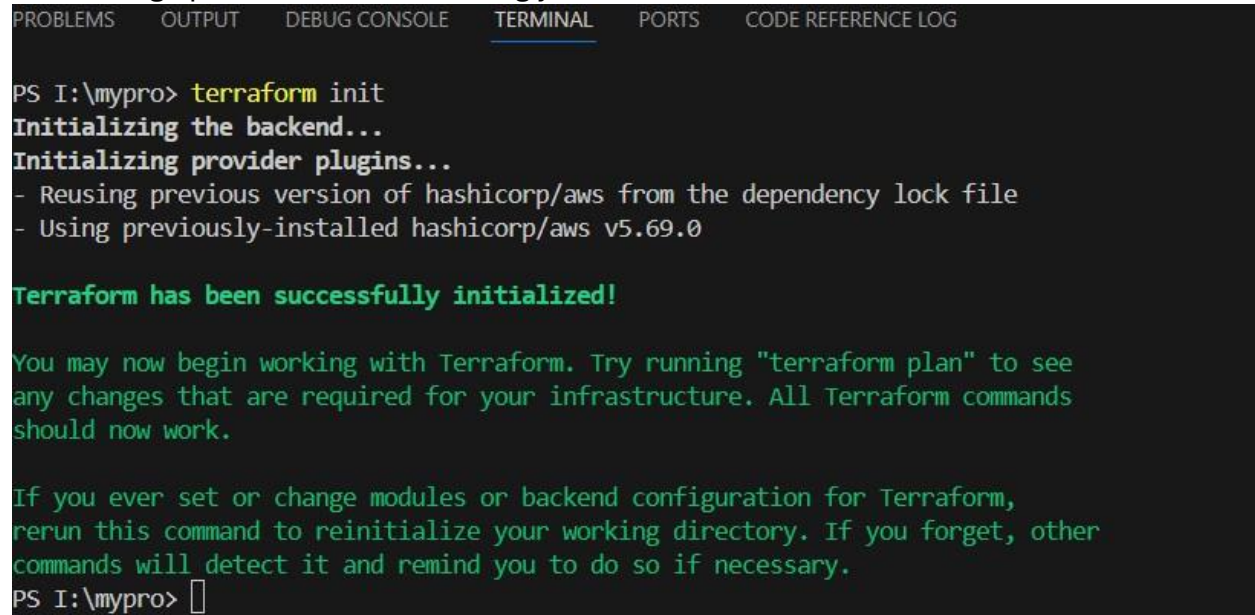
- **Resources:**
  - **aws\_launch\_configuration.app: Defines the launch configuration for the ASG.**
  - **aws\_autoscaling\_group.app: Creates an Auto Scaling Group with specified parameters.**
- **Configuration:**
  - **Launch Configuration:**
    - **AMI: ami-0fff1b9a61dec8a5f**
    - **Instance Type: t2.micro**
    - **SSH Key: vockey**
    - **Security Groups: Attached to HTTP\_SG**
    - **User Data: Installs and starts Apache HTTP server**
  - **Auto Scaling Group:**
    - **Size: Minimum 1, Maximum 3, Desired 2**
    - **Subnets: Deployed in private subnets (subnet1 and subnet2)**
    - **Target Group: Associated with TG-bassam**
    - **Tags: Instances tagged as ASG\_Instance**
    - **Lifecycle: Ignores changes to desired\_capacity to allow manual scaling**

## **Terraform Preparation**

In this figures below we initialize ,formatting , validate and plan for the code stack

### **Terraform Initialization**

initializes a working directory and downloads the necessary provider plugins and modules and setting up the backend for storing your infrastructure's state



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

PS I:\mypro> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.69.0

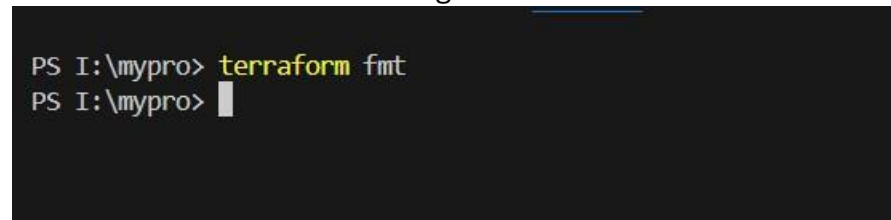
Terraform has been successfully initialized!

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 I:\mypro> 
```

### **Terraform Format**

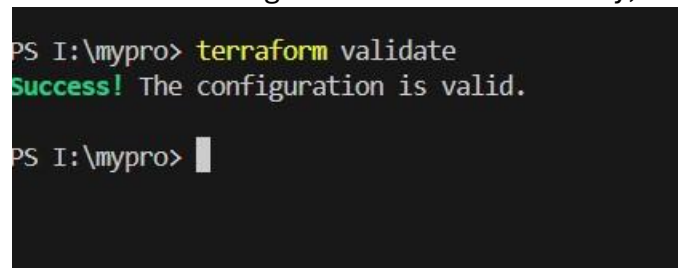
used to rewrite Terraform configuration files to a canonical format and style.



```
PS I:\mypro> terraform fmt
PS I:\mypro> 
```

### **Terraform Validate:**

validates the configuration files in a directory, referring only to the configuration



```
PS I:\mypro> terraform validate
Success! The configuration is valid.

PS I:\mypro> 
```

## Terraform Plan:

creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure.

```
PS I:\mypro> 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_autoscaling_group.app will be created
+ resource "aws_autoscaling_group" "app" {
  + arn                    = (known after apply)
  + availability_zones     = (known after apply)
  + default_cooldown       = (known after apply)
  + desired_capacity       = 2
  + force_delete           = false
  + force_delete_warm_pool = false
  + health_check_grace_period = 300
  + health_check_type      = (known after apply)
  + id                     = (known after apply)
  + ignore_failed_scaling_activities = false
  + launch_configuration    = (known after apply)
  + load_balancers          = (known after apply)
  + max_size               = 3
  + metrics_granularity     = "1Minute"
  + min_size               = 1
  + name                   = (known after apply)
  + name_prefix             = (known after apply)
  + predicted_capacity      = (known after apply)
  + protect_from_scale_in  = false
  + service_linked_role_arn = (known after apply)
  + target_group_arns       = (known after apply)
  + vpc_zone_identifier     = (known after apply)
  + wait_for_capacity_timeout = "10m"
}
```

## Terraform Apply

executes planned actions, creating, updating, or deleting infrastructure resources to match the new state outlined in your IaC.

```
Plan: 23 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: 
```

```

aws_lb.test: Still creating... [2m10s elapsed]
aws_lb.test: Still creating... [2m20s elapsed]
aws_lb.test: Still creating... [2m30s elapsed]
aws_lb.test: Still creating... [2m40s elapsed]
aws_lb.test: Still creating... [2m50s elapsed]
aws_lb.test: Still creating... [3m0s elapsed]
aws_lb.test: Still creating... [3m10s elapsed]
aws_lb.test: Creation complete after 3m18s [id=arn:aws:elasticloadbalancing:us-east-1:248193779364:loadbalancer/ap
aws_lb_listener.test: Creating...
aws_lb_listener.test: Creation complete after 1s [id=arn:aws:elasticloadbalancing:us-east-1:248193779364:listener/b1eca8324092f]

Apply complete! Resources: 23 added, 0 changed, 0 destroyed.
PS I:\mypro>

```

## OUTPUT

Resources
EC2 Global View

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	3	Auto Scaling Groups	1	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	1	Instances	3
Key pairs	1	Load balancers	1	Placement groups	0
Security groups	4	Snapshots	0	Volumes	3

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info
Launch configurations
Launch templates
Actions
Create Auto Scaling group

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	terraform-20241004121856223200000007	app-launch-configuration	2	-	2	1



<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	ASG_Instance	i-029b13e59315cafb0	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1a
<input type="checkbox"/>	bastion-host	i-041c41b4bd41627bd	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1a
<input type="checkbox"/>	ASG_Instance	i-0ac019653ff2ac812	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1b

Filter table to exclude running instances

i-029b13e59315cafb0 (ASG\_Instance)

Answer private resource DNS name

-

Auto-assigned IP address

-

IAM Role

-

Instance type

t2.micro

VPC ID

vpc-067d966cc3b1b43f1 (main-vpc)

Subnet ID

subnet-0aee3bd44e162f926 (priv\_subnet-1)

Elastic IP addresses

-

AWS Compute Optimizer finding

[Opt-in to AWS Compute Optimizer for recommendations.](#) | [Learn more](#)

Auto Scaling Group name

terraform-20241004121856223200000007

instances (1 / 2)

1 minute ago

COMPARE

INSTANCE STATE

ACTIONS

EXPORT INSTANCES

All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	ASG_Instance	i-029b13e59315cafb0	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1a
<input checked="" type="checkbox"/>	bastion-host	i-041c41b4bd41627bd	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1a
<input type="checkbox"/>	ASG_Instance	i-0ac019653ff2ac812	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-1b

i-041c41b4bd41627bd (bastion-host)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary [Info](#)

Instance ID

i-041c41b4bd41627bd (bastion-host)

IPv6 address

-

Hostname type

IP name: in-192.168.3.40 or 2 internal

Public IPv4 address

100.24.59.40 | [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

in-192.168.3.40 or 2 internal

Private IPv4 addresses

192.168.3.40

Public IPv4 DNS

-

```
Admin@DESKTOP-445DUV0 MINGW64 ~  
$ cd Downloads/  
  
Admin@DESKTOP-445DUV0 MINGW64 ~/Downloads  
$ ssh -i labsuser.p  
labsuser.pem labsuser.ppk  
  
Admin@DESKTOP-445DUV0 MINGW64 ~/Downloads  
$ ssh -i labsuser.pem ec2-user@100.24.59.40
```

```
Admin@DESKTOP-445DUV0 MINGW64 ~  
$ cd Downloads/  
  
Admin@DESKTOP-445DUV0 MINGW64 ~/Downloads  
$ ssh -i labsuser.p  
labsuser.pem labsuser.ppk  
  
Admin@DESKTOP-445DUV0 MINGW64 ~/Downloads  
$ ssh -i labsuser.pem ec2-user@100.24.59.40
```

```
[ec2-user@ip-192-168-3-40 ~]$ whoami  
ec2-user  
[ec2-user@ip-192-168-3-40 ~]$
```

## AWS GRADUATION PROJECT

```

ec2-user@ip-192-168-1-156 ~]$
ec2-user@ip-192-168-1-156 ~]$
ec2-user@ip-192-168-1-156 ~]$
ec2-user@ip-192-168-1-156 ~]$ ifconfig
enx0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 192.168.1.156 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::10a7:9bff:fe94:d497 prefixlen 64 scopeid 0x20<link>
    ether 12:a7:9b:94:d4:97 txqueuelen 1000 (Ethernet)
    RX packets 22790 bytes 31572665 (30.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4215 bytes 327437 (319.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 1020 (1020.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 1020 (1020.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ec2-user@ip-192-168-1-156 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=2.54 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=1.30 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=1.33 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=1.49 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=1.55 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=2.13 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=116 time=1.57 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=116 time=1.77 ms
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7012ms
rtt min/avg/max/mdev = 1.304/1.709/2.536/0.398 ms
ec2-user@ip-192-168-1-156 ~]$

```

EC2 > Load balancers

**Load balancers (1)** Actions Create load balancer


Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	<a href="#">bassam-alb</a>	bassam-alb-1312631156...	Active	vpc-067d966cc3b1b4...	<a href="#">2 Availability Zones</a>	application

<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input checked="" type="checkbox"/>	<a href="#">terraform-20241009183622993600000007</a>	<a href="#">app-launch-configuration</a>	2	-	2	1

**Auto Scaling group: terraform-20241009183622993600000007**







Group details			
<a href="#">Edit</a>			
Auto Scaling group name	Desired capacity	Desired capacity type	Amazon Resource Name (ARN)
terraform-20241009183622993600000007	2	Units (number of instances)	 arn:aws:autoscaling:us-east-1:248193779364:autoScalingGroup:a89a7234-f2b4-4ecd-984f-35fb3f636df6:autoScalingGroupName/terraform-20241009183622993600000007
Date created	Minimum capacity	Status	
Wed Oct 09 2024 21:36:20 GMT+0300 (Eastern European Summer Time)	1	-	
	Maximum capacity		
	3		

Find Instance by attribute or tag (case-sensitive) All states ▼

Instance state = running X Clear filters

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
i-03af6abba8007f1f6	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
i-005c53055e955ba87	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b
i-0c79a824353085a1e	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a

	Name	Allocated IPv4 addr...	Type	Allocation ID
<input type="checkbox"/>	nat-eip	<a href="#">54.172.183.161</a>	Public IP	eipalloc-069f71d3c3924bbf2
<input type="checkbox"/>	main-gateway	<a href="#">igw-06288ae57e96f8c76</a>	Attached	<a href="#">vpc-0bf01c56105fd96e0</a>   <a href="#">main-vpc</a>

Details			
NAT gateway ID  nat-0788be50c9491bad1	Connectivity type Public	State  Available	State message –
NAT gateway ARN  arn:aws:ec2:us-east-1:248193779364:natgateway/nat-0788be50c9491bad1	Primary public IPv4 address 54.172.183.161	Primary private IPv4 address  192.168.4.133	Primary network interface ID eni-0b8070976ab00eb93 
VPC <a href="#">vpc-0bf01c56105fd96e0 / main-vpc</a>	Subnet <a href="#">subnet-0904e4b6cb63dcb7c / pub_subnet-2</a>	Created  Wednesday, October 9, 2024 at 21:36:14 GMT+3	Deleted –



<input type="checkbox"/>	Name ▾	Security group ID ▾	Security group name ▾	VPC ID ▾
<input type="checkbox"/>	-	<a href="#">sg-09f05ff7a796f87ff</a>	default	<a href="#">vpc-04db7e2344e86161e</a>
<input type="checkbox"/>	HTTP-SG	<a href="#">sg-0358b7f640f8e9ba6</a>	terraform-202410091836154780000...	<a href="#">vpc-0bf01c56105fd96e0</a>
<input type="checkbox"/>	-	<a href="#">sg-09fa3c82ae2782e04</a>	default	<a href="#">vpc-0bf01c56105fd96e0</a>
<input type="checkbox"/>	bastion-SG	<a href="#">sg-04f4b8315db6dc02c</a>	terraform-202410091836154702000...	<a href="#">vpc-0bf01c56105fd96e0</a>

<input checked="" type="checkbox"/>	Name ▾	Subnet ID ▾	State ▾	VPC ▾	IPv4 CIDR ▾
<input checked="" type="checkbox"/>	priv_subnet-1	<a href="#">subnet-090438304101c4a8e</a>	Available	<a href="#">vpc-0bf01c56105fd96e0</a>   main...	192.168.1.0/24
<input checked="" type="checkbox"/>	pub_subnet-2	<a href="#">subnet-0904e4b6cb63dcb7c</a>	Available	<a href="#">vpc-0bf01c56105fd96e0</a>   main...	192.168.4.0/24
<input checked="" type="checkbox"/>	pub_subnet-1	<a href="#">subnet-0b57975c517d51195</a>	Available	<a href="#">vpc-0bf01c56105fd96e0</a>   main...	192.168.3.0/24
<input type="checkbox"/>	-	<a href="#">subnet-0148f272f7c98d9af</a>	Available	<a href="#">vpc-04db7e2344e86161e</a>	172.31.0.0/20
<input checked="" type="checkbox"/>	priv_subnet-2	<a href="#">subnet-02335556e0690fc36</a>	Available	<a href="#">vpc-0bf01c56105fd96e0</a>   main...	192.168.2.0/24

<input checked="" type="checkbox"/>	Name ▾	Route table ID ▾	Explicit subnet associ... ▾
<input checked="" type="checkbox"/>	public_RT	<a href="#">rtb-0da6fa665aac0e497</a>	<u>2 subnets</u>
<input type="checkbox"/>	-	<a href="#">rtb-09aa7ec4f411cbc77</a>	-
<input type="checkbox"/>	-	<a href="#">rtb-0a35575a971d00249</a>	-
<input checked="" type="checkbox"/>	private_RT	<a href="#">rtb-08f6ba0d2a0b6cb16</a>	<u>2 subnets</u>

## Graduation Project DEPI\_1\_CAI1\_ISS4\_G1e AWS Cloud Solution Admin & Architect

### Deploying a Highly Available Web Application with Auto Scaling using IAC "Terraform"

Supervised by:  
**DR-Ibrahim Goma**  
 Eng.Mohamed Kababi  
 Eng.Tarek Elkabani

Student Name: Bassam Amr Fathy  
 Student Number: 1122141562  
 Email Address: [bassam11287@gmail.com](mailto:bassam11287@gmail.com)  
[Documentaion](#)

# AWS Infrastructure Deployment Documentation

[Home](#)

## Table of Contents

1. Introduction
2. Prerequisites
3. Architecture Overview
4. Terraform Configuration
  - VPC Configuration
  - Subnet Configuration
  - Internet Gateway and NAT Gateway
  - Route Tables
  - Security Groups
  - EC2 Instances
  - Load Balancer
  - Auto Scaling Group
5. Deployment Steps
6. Cleanup
7. Best Practices
8. Conclusion

## Stack Destroy

```
}

# aws_vpc.main will be destroyed
- resource "aws_vpc" "main" {
  - arn = "arn:aws:ec2:us-east-1:248193779364:vpc/vpc-0bf01c56105fd96e0" -> null
  - assign_generated_ipv6_cidr_block = false -> null
  - cidr_block = "192.168.0.0/16" -> null
  - default_network_acl_id = "acl-0f451ed4649a8bc07" -> null
  - default_route_table_id = "rtb-0a35575a971d00249" -> null
  - default_security_group_id = "sg-09fa3c82ae2782e04" -> null
  - dhcp_options_id = "dopt-047a70bc12cb7ce8d" -> null
  - enable_dns_hostnames = false -> null
  - enable_dns_support = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id = "vpc-0bf01c56105fd96e0" -> null
  - instance_tenancy = "default" -> null
  - ipv6_netmask_length = 0 -> null
  - main_route_table_id = "rtb-0a35575a971d00249" -> null
  - owner_id = "248193779364" -> null
  - tags = {
    - "Name" = "main-vpc"
  } -> null
  - tags_all = {
    - "Name" = "main-vpc"
  } -> null
  # (4 unchanged attributes hidden)
}
```

**Plan:** 0 to add, 0 to change, 23 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_autoscaling_group.app: Destruction complete after 3m50s
aws_subnet.subnet1: Destroying... [id=subnet-090438304101c4a8e]
aws_subnet.subnet2: Destroying... [id=subnet-02335556e0690fc36]
aws_launch_configuration.app: Destroying... [id=app-launch-configuration]
aws_lb_target_group.test: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:248193779364:targetgroup/TG-bassam/a393b9122df0f83f]
aws_launch_configuration.app: Destruction complete after 0s
aws_security_group.HTTP_SG: Destroying... [id=sg-0358b7f640f8e9ba6]
aws_lb_target_group.test: Destruction complete after 1s
aws_subnet.subnet2: Destruction complete after 1s
aws_subnet.subnet1: Destruction complete after 1s
aws_security_group.HTTP_SG: Destruction complete after 2s
aws_vpc.main: Destroying... [id=vpc-0bf01c56105fd96e0]
aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 23 destroyed.
```

Instances (1/3) Info Last updated less than a minute ago Connect Instance state Actions Launch instances

All states < 1 > ⚙

<input checked="" type="checkbox"/>	Name <a href="#">↗</a>	Instance ID	Instance state <a href="#">↕</a>	Instance type <a href="#">↕</a>	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	ASG_Instance	i-029b13e59315cafb0	⌚ Shutting-down <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	-	<a href="#">View alarms +</a>	us-east-1a
<input type="checkbox"/>	bastion-host	i-041c41b4bd41627bd	⌚ Shutting-down <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	-	<a href="#">View alarms +</a>	us-east-1a
<input type="checkbox"/>	ASG_Instance	i-0ac019653ff2ac812	⌚ Shutting-down <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	-	<a href="#">View alarms +</a>	us-east-1b