

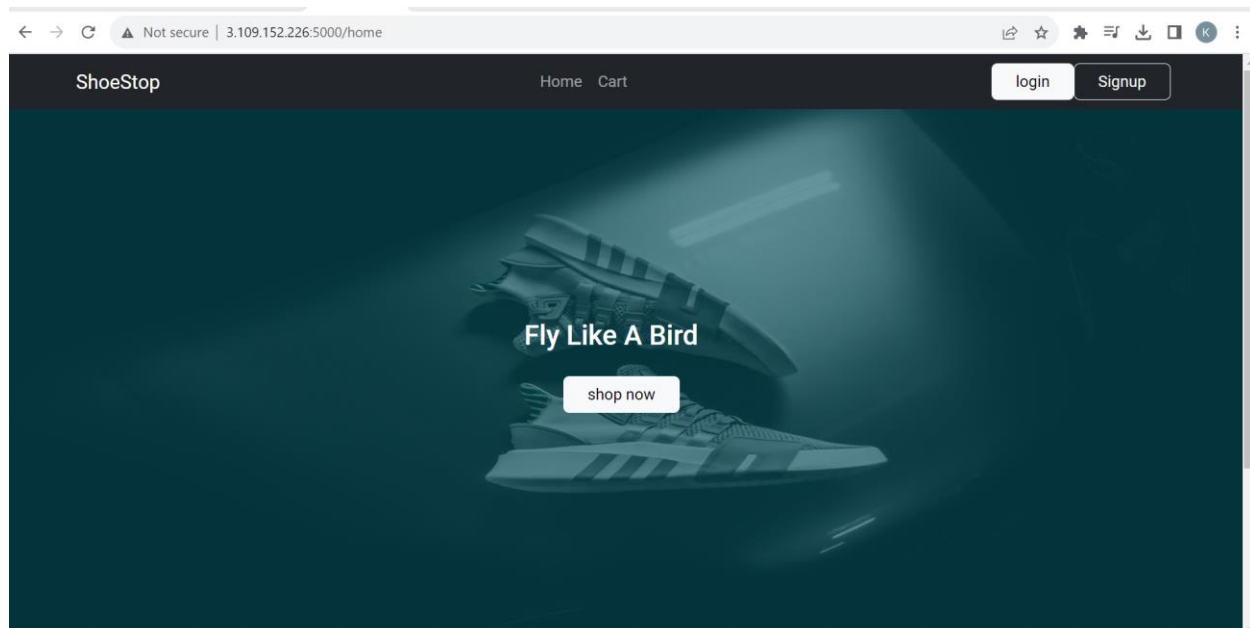
```

[root@ip-172-31-41-233 Hackathon-Application]# ll
total 16
drwxr-xr-x. 10 root root 16384 Aug 26 13:44 shoestop
[root@ip-172-31-41-233 Hackathon-Application]# cd shoestop/
[root@ip-172-31-41-233 shoestop]# l
-bash: l: command not found
[root@ip-172-31-41-233 shoestop]# ll
total 96
-rw-r--r--. 1 root root 704 Aug 26 13:44 Readme.md
drwxr-xr-x. 2 root root 111 Aug 26 13:44 controllers
drwxr-xr-x. 2 root root 26 Aug 26 13:44 db
drwxr-xr-x. 2 root root 31 Aug 26 13:44 middlewares
drwxr-xr-x. 2 root root 80 Aug 26 13:44 modal
drwxr-xr-x. 162 root root 16384 Aug 26 13:44 node_modules
-rw-r--r--. 1 root root 67125 Aug 26 13:44 package-lock.json
-rw-r--r--. 1 root root 509 Aug 26 13:44 package.json
drwxr-xr-x. 3 root root 54 Aug 26 13:44 public
drwxr-xr-x. 2 root root 51 Aug 26 13:44 routes
-rw-r--r--. 1 root root 996 Aug 26 13:44 server.js
drwxr-xr-x. 2 root root 24 Aug 26 13:44 utility
[root@ip-172-31-41-233 shoestop]# npm start

> jobs-server@1.0.0 start
> nodemon server.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
connected to db
server started at 5000

```



Directory: C:\Users\HP\terra

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|--------|---------|
| -a---- | 8/26/2023 11:20 PM | 0 | main.tf |

```
PS C:\Users\HP\terra>
PS C:\Users\HP\terra> aws configure
AWS Access Key ID [None]: AKIAH2I3VKPNJ4W4EX65
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]:
PS C:\Users\HP\terra>
PS C:\Users\HP\terra> ls
```

Directory: C:\Users\HP\terra

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|--------|---------|
| -a---- | 8/26/2023 11:20 PM | 0 | main.tf |

```
PS C:\Users\HP\terra> cat .\main.tf
PS C:\Users\HP\terra> ls
```

Directory: C:\Users\HP\terra

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|--------|---------|
| -a---- | 8/26/2023 11:54 PM | 371 | main.tf |

```

PS C:\Users\HP\terra> cat .\main.tf
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region = "us-east-2"
}

resource "aws_instance" "app_server" {
  ami = "ami-06f621d90fa29f6d0"
  instance_type = "t2.micro"

  tags = {
    Name = "EC2Instance"
  }
}

```

```

PS C:\Users\HP\terra>
PS C:\Users\HP\terra>
PS C:\Users\HP\terra> terraform init

```

Initializing the backend...

Initializing provider plugins...

```

- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.67.0...
- Installed hashicorp/aws v4.67.0 (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.

```

PS C:\Users\HP\terra>

```

```

PS C:\Users\HP\terra> ls

```

Directory: C:\Users\HP\terra

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|--------|---------------------|
| d---- | 8/26/2023 11:54 PM | | .terraform |
| -a---- | 8/26/2023 11:54 PM | 1407 | .terraform.lock.hcl |
| -a---- | 8/26/2023 11:54 PM | 371 | main.tf |

```

PS C:\Users\HP\terra>

```

```

PS C:\Users\HP\terra> 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.app_server will be created
+ resource "aws_instance" "app_server" {
  + ami                  = "ami-06f621d90fa29f6d0"
  + 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_state         = (known after apply)
  + instance_type          = "t2.micro"
  + ipv6_address_count      = (known after apply)
  + ipv6_addresses         = (known after apply)
  + key_name                = (known after apply)
  + 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)
  + source_dest_check       = true
  + subnet_id              = (known after apply)
  + tags                    = {
    + "Name" = "EC2Instance"
  }
  + tags_all                = {
    + "Name" = "EC2Instance"
  }
  + tenancy                  = (known after apply)
  + user_data                = (known after apply)
  + user_data_base64         = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids   = (known after apply)
}

```

```

information.
PS C:\Users\HP\terra> terraform fmt
main.tf
PS C:\Users\HP\terra> terraform validate
Success! The configuration is valid.

PS C:\Users\HP\terra>

```

C:\Windows\system32\cmd.exe - powershell

```
+ ami = "ami-06f621d90fa29f6d0"
+ 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_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ 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)
+ source_dest_check = true
+ subnet_id = (known after apply)
+ tags = {
  + "Name" = "EC2Instance"
}
+ tags_all = {
  + "Name" = "EC2Instance"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
```

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

```

+ subnet_id           = (known after apply)
+ tags                = {
+   + "Name" = "EC2Instance"
+ }
+ tags_all            = {
+   + "Name" = "EC2Instance"
+ }
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64    = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

```

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.app_server: Creating...
aws_instance.app_server: Still creating... [10s elapsed]
aws_instance.app_server: Still creating... [20s elapsed]
aws_instance.app_server: Still creating... [30s elapsed]
aws_instance.app_server: Still creating... [40s elapsed]
aws_instance.app_server: Still creating... [50s elapsed]
aws_instance.app_server: Still creating... [1m0s elapsed]
aws_instance.app_server: Creation complete after 1m7s [id=i-0aad11183d7a5a398]

```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
 PS C:\Users\HP\terra>

```

- enclave_options {
  - enabled = false -> null
}

- maintenance_options {
  - auto_recovery = "default" -> null
}

- metadata_options {
  - http_endpoint           = "enabled" -> null
  - http_put_response_hop_limit = 1 -> null
  - http_tokens              = "optional" -> null
  - instance_metadata_tags    = "disabled" -> null
}

- private_dns_name_options {
  - enable_resource_name_dns_a_record    = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type                        = "ip-name" -> null
}

- root_block_device {
  - delete_on_termination = true -> null
  - device_name            = "/dev/sda1" -> null
  - encrypted              = false -> null
  - iops                   = 0 -> null
  - tags                   = {} -> null
  - throughput             = 0 -> null
  - volume_id              = "vol-0d738bb6ec495d72b" -> null
  - volume_size            = 8 -> null
  - volume_type            = "standard" -> null
}
}

```

Plan: 0 to add, 0 to change, 1 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

Plan: 0 to add, 0 to change, 1 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_instance.app_server: Destroying... [id=i-0aad11183d7a5a398]
aws_instance.app_server: Still destroying... [id=i-0aad11183d7a5a398, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-0aad11183d7a5a398, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-0aad11183d7a5a398, 30s elapsed]
aws_instance.app_server: Destruction complete after 32s

```

Destroy complete! Resources: 1 destroyed.

PS C:\Users\HP\terra>

```
PS C:\Users\HP\terra>
PS C:\Users\HP\terra> ls
```

Directory: C:\Users\HP\terra

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|--------|--------------------------|
| d----- | 8/26/2023 11:54 PM | | .terraform |
| -a---- | 8/26/2023 11:54 PM | 1407 | .terraform.lock.hcl |
| -a---- | 8/27/2023 12:06 AM | 361 | main.tf |
| -a---- | 8/27/2023 12:12 AM | 180 | terraform.tfstate |
| -a---- | 8/27/2023 12:12 AM | 4626 | terraform.tfstate.backup |

```
PS C:\Users\HP\terra>
PS C:\Users\HP\terra>
```