DevOps Certification Training

IntelliPaat

Terraform Assignment - 5

You have been asked to:

- Destroy the previous deployments
- Create a script to install apache2
- Run this script on a newly created EC2 instance
- Print the IP address of the instance in a file on the local, once deployed

Download Attachment

```
ubuntu@ip-172-31-93-95:~$ mkdir test
ubuntu@ip-172-31-93-95:~$ cd test/
ubuntu@ip-172-31-93-95:~/test$ nano instance.tf
ubuntu@ip-172-31-93-95:~/test$ nano install_nginx.sh
ubuntu@ip-172-31-93-95:~/test$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.8.0...
- Installed hashicorp/aws v4.8.0 (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.
ubuntu@ip-172-31-93-95:~/test$ terraform plan

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

ubuntu@ip-172-31-93-95: ~/test
ubuntu@ip-172-31-93-95:~$ ls
test
ubuntu@ip-172-31-93-95:~$ cd test/
ubuntu@ip-172-31-93-95:~/test$ ls
install_nginx.sh instance.tf terraform.tfstate terraform.tfstate.backup
ubuntu@ip-172-31-93-95:~/test$
17:47
03/04/2022

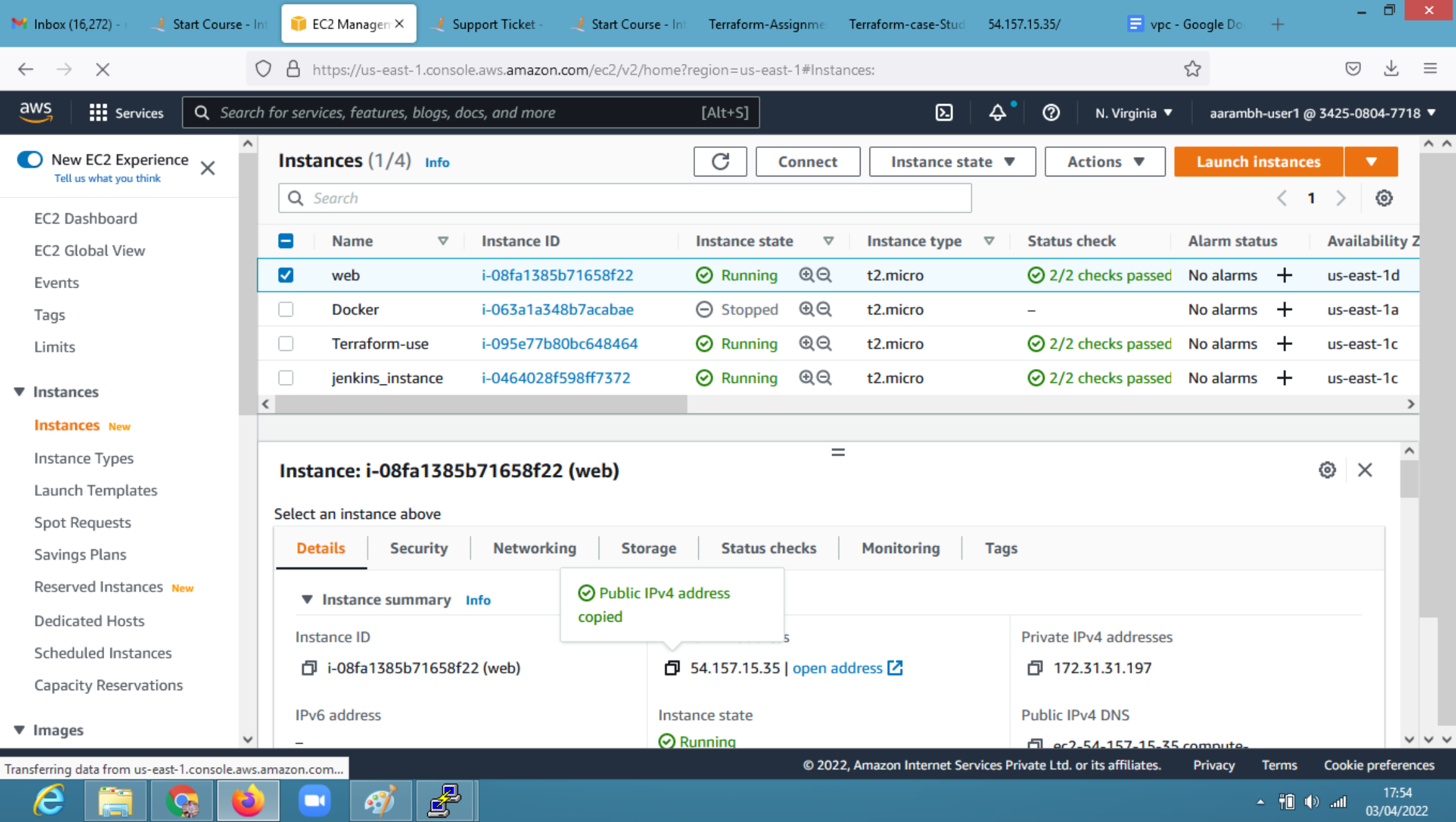
```
ubuntu@ip-172-31-93-95:~/test$ vi instance.tf
provider "aws" {
    access_key = "AKIAU7PYSBVTHSC6XYNM"
    secret_key = "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"
    region     = "us-east-1"
}
resource "aws_instance" "web" {
   ami= "ami-04505e74c0741db8d"
   instance_type = "t2.micro"
   key_name= "hotfix"
   user_data       = "${file("install_nginx.sh")}"
  tags = {
    Name = "web"
  }
}
output "IP" {
  value = aws_instance.web.public_ip
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

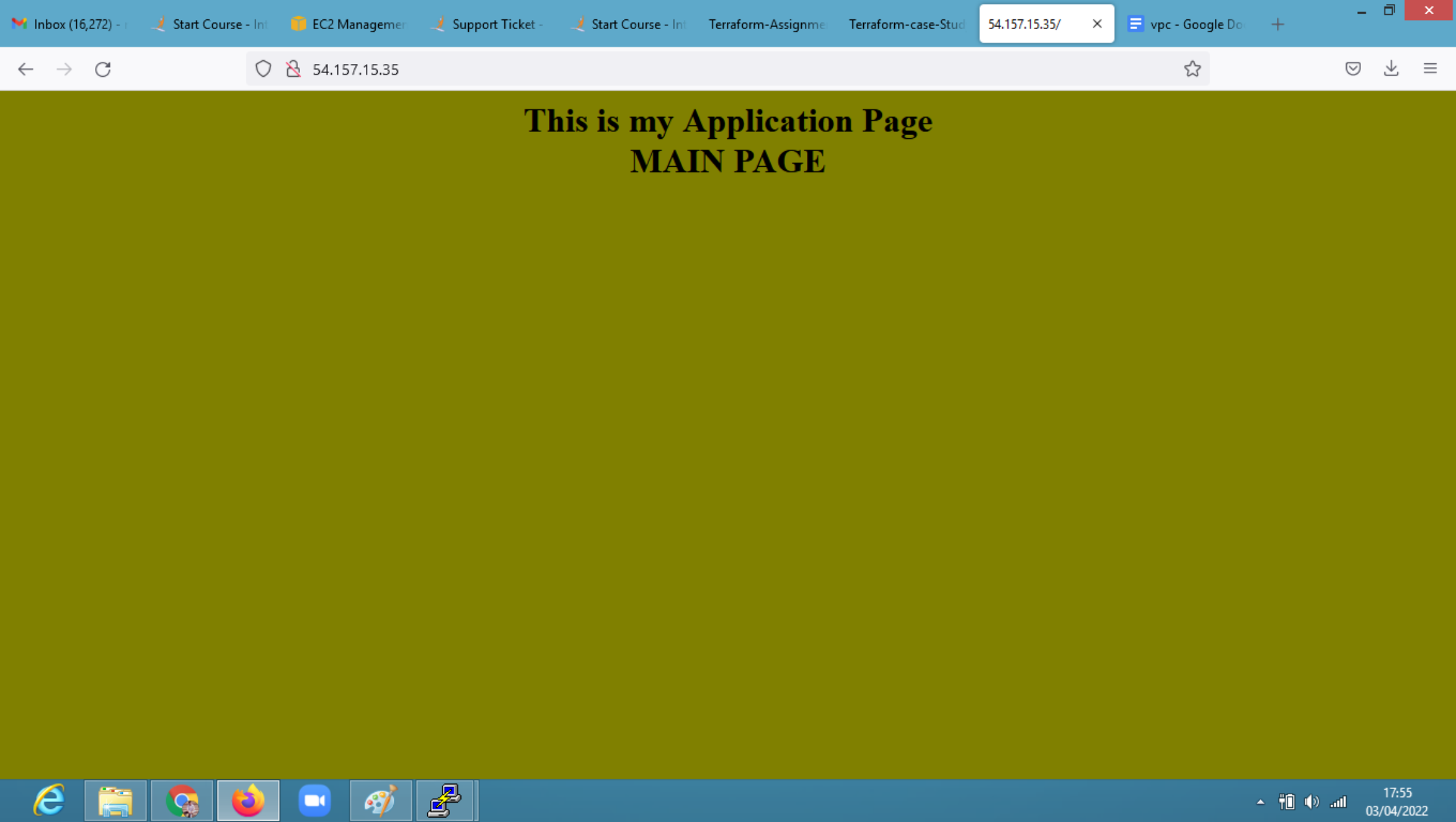
```
ubuntu@ip-172-31-93-95:~/test$ vi install_nginx.sh
#!/bin/bash
sudo apt-get update
sudo apt-get install -y nginx
sudo su
echo "<html><body bgcolor="olive"><h1 align="center"> This is my Application Page<br>MAIN PAGE<br></h1></body></html>"> /v
ar/www/html/index.html
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
ubuntu@ip-172-31-93-95:~/test$ 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.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-04505e74c0741db8d"
      + 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_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (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                             = "hotfix1"
      + 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)
```

```
ubuntu@ip-172-31-93-95:~/test$ nano instance.tf
ubuntu@ip-172-31-93-95:~/test$ terraform apply

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

Terraform will perform the following actions:

  # aws_instance.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-04505e74c0741db8d"
      + 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_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (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                             = "hotfix"
      + 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)
```

```
          + iops                = (known after apply)
          + kms_key_id          = (known after apply)
          + tags                = (known after apply)
          + throughput          = (known after apply)
          + volume_id           = (known after apply)
          + volume_size         = (known after apply)
          + volume_type         = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + IP = (known after apply)

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.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
aws_instance.web: Creation complete after 32s [id=i-08fa1385b71658f22]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

IP = "54.157.15.35"
ubuntu@ip-172-31-93-95:~/test$
```


Inbox (16,272) -
Start Course - Int
EC2 Managen
Support Ticket -
Start Course - Int
Terraform-Assignme
Terraform-case-Stud
54.157.15.35/
vpc - Google Do
https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
aarambh-user1 @ 3425-0804-7718
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
Instances (1/4) Info
Connect
Instance state
Actions
Launch instances
Search
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Z
web i-08fa1385b71658f22 Running t2.micro 2/2 checks passed No alarms us-east-1d
Docker i-063a1a348b7acabae Stopped t2.micro – No alarms us-east-1a
Terraform-use i-095e77b80bc648464 Running t2.micro 2/2 checks passed No alarms us-east-1c
jenkins_instance i-0464028f598ff7372 Running t2.micro 2/2 checks passed No alarms us-east-1c
Instance: i-08fa1385b71658f22 (web)
Select an instance above
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Public IPv4 address copied
Instance summary Info
Instance ID
i-08fa1385b71658f22 (web)
54.157.15.35 | open address
Private IPv4 addresses
172.31.31.197
IPv6 address
Instance state
Running
Public IPv4 DNS
Transferring data from us-east-1.console.aws.amazon.com...
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
17:54
03/04/2022


Inbox (16,272) -
Start Course - Int
EC2 Managemer
Support Ticket -
Start Course - Int
Terraform-Assignme
Terraform-case-Stud
54.157.15.35/
vpc - Google Do
54.157.15.35
This is my Application Page
MAIN PAGE
17:55
03/04/2022