

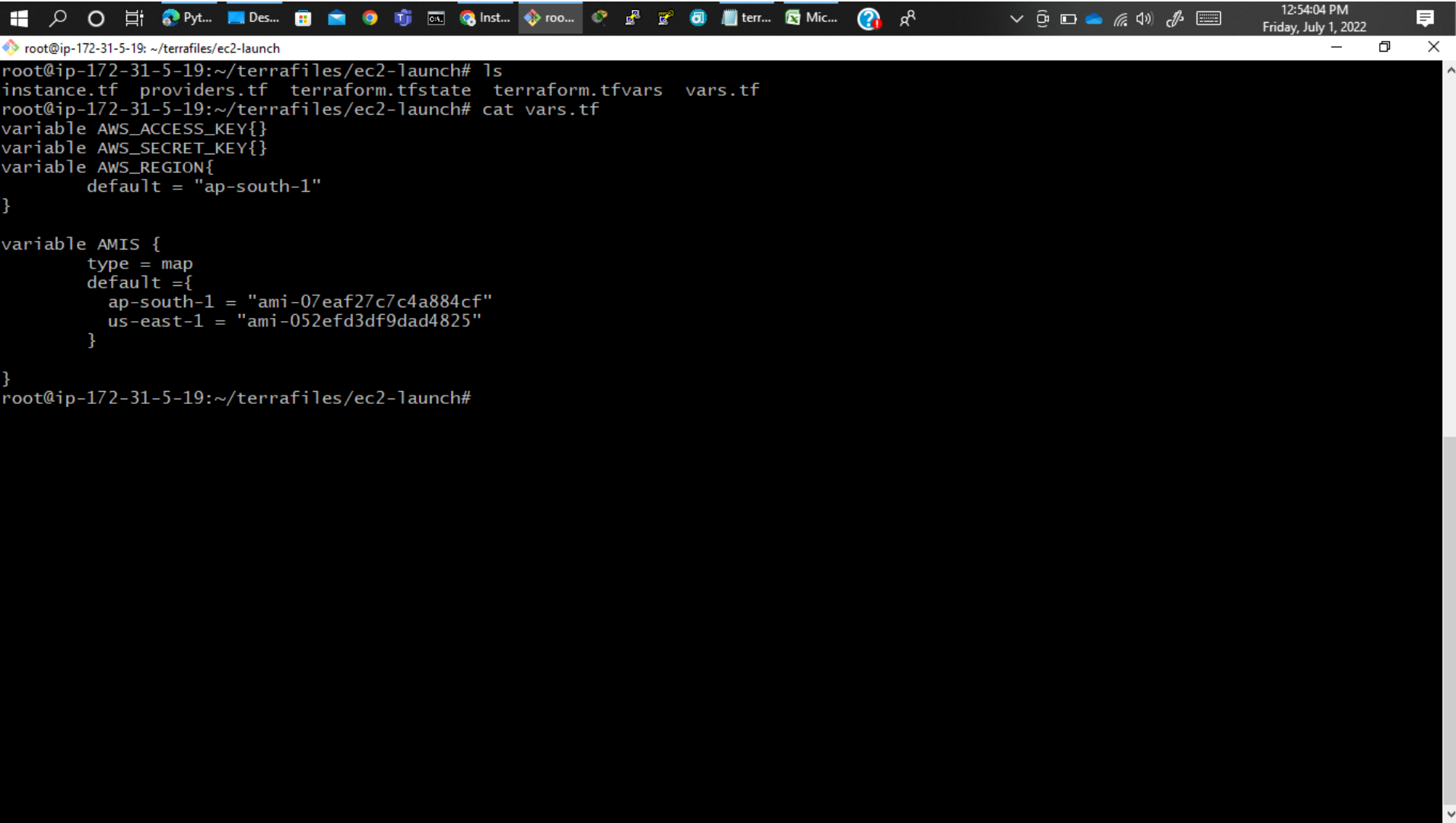
HashiCorp

Terraform



***Create some files for launch ec2 instance using terraform

Providers.tf ... vars.tf...instance.tf...terraform.tfvars



A terminal window with a dark background and light-colored text. The window title bar shows the path ~/terrafiles/ec2-launch. The terminal output shows the following commands and their results:

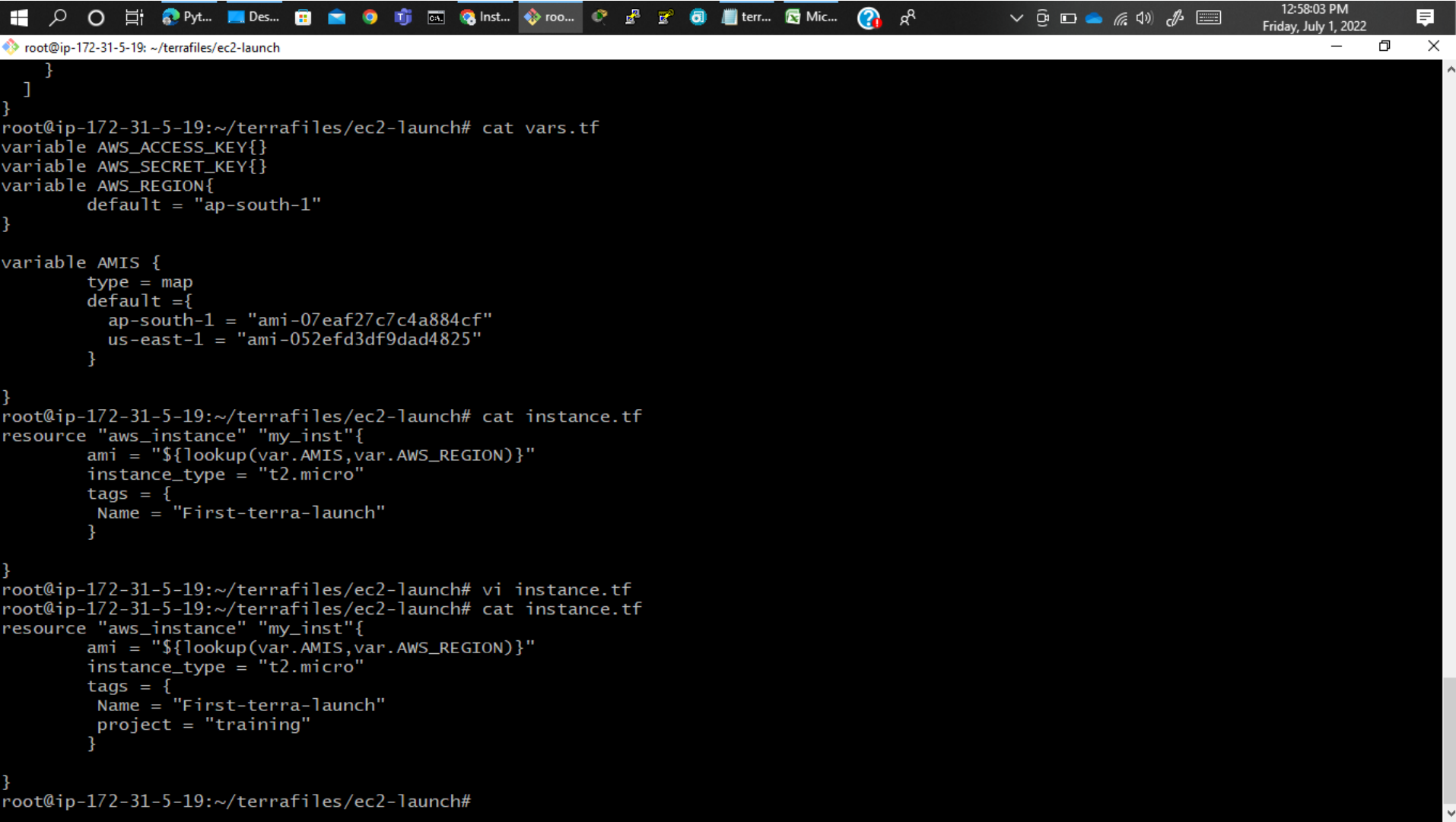
```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch# ls
instance.tf  providers.tf  terraform.tfstate  terraform.tfvars  vars.tf
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat vars.tf
variable AWS_ACCESS_KEY{}
variable AWS_SECRET_KEY{}
variable AWS_REGION{
    default = "ap-south-1"
}

variable AMIS {
    type = map
    default = {
        ap-south-1 = "ami-07eaf27c7c4a884cf"
        us-east-1 = "ami-052efd3df9dad4825"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch#
```

The terminal window is part of a desktop environment. The taskbar at the top shows various application icons including PyCharm, Desktop, File Explorer, Google Chrome, and several instances of the terminal. The system clock in the top right corner indicates the time is 12:54:04 PM on Friday, July 1, 2022.

Terraform Files

A terminal window with a dark background and light-colored text. The window title bar shows the path 'root@ip-172-31-5-19: ~/terrafiles/ec2-launch'. The terminal displays the contents of three Terraform files: 'vars.tf', 'instance.tf', and 'main.tf'. The user has executed 'cat' and 'vi' commands to view and edit these files. The window includes a taskbar at the top with various application icons and a system tray on the right showing the time and date.

```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch
}
]
}
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat vars.tf
variable AWS_ACCESS_KEY{}
variable AWS_SECRET_KEY{}
variable AWS_REGION{
    default = "ap-south-1"
}

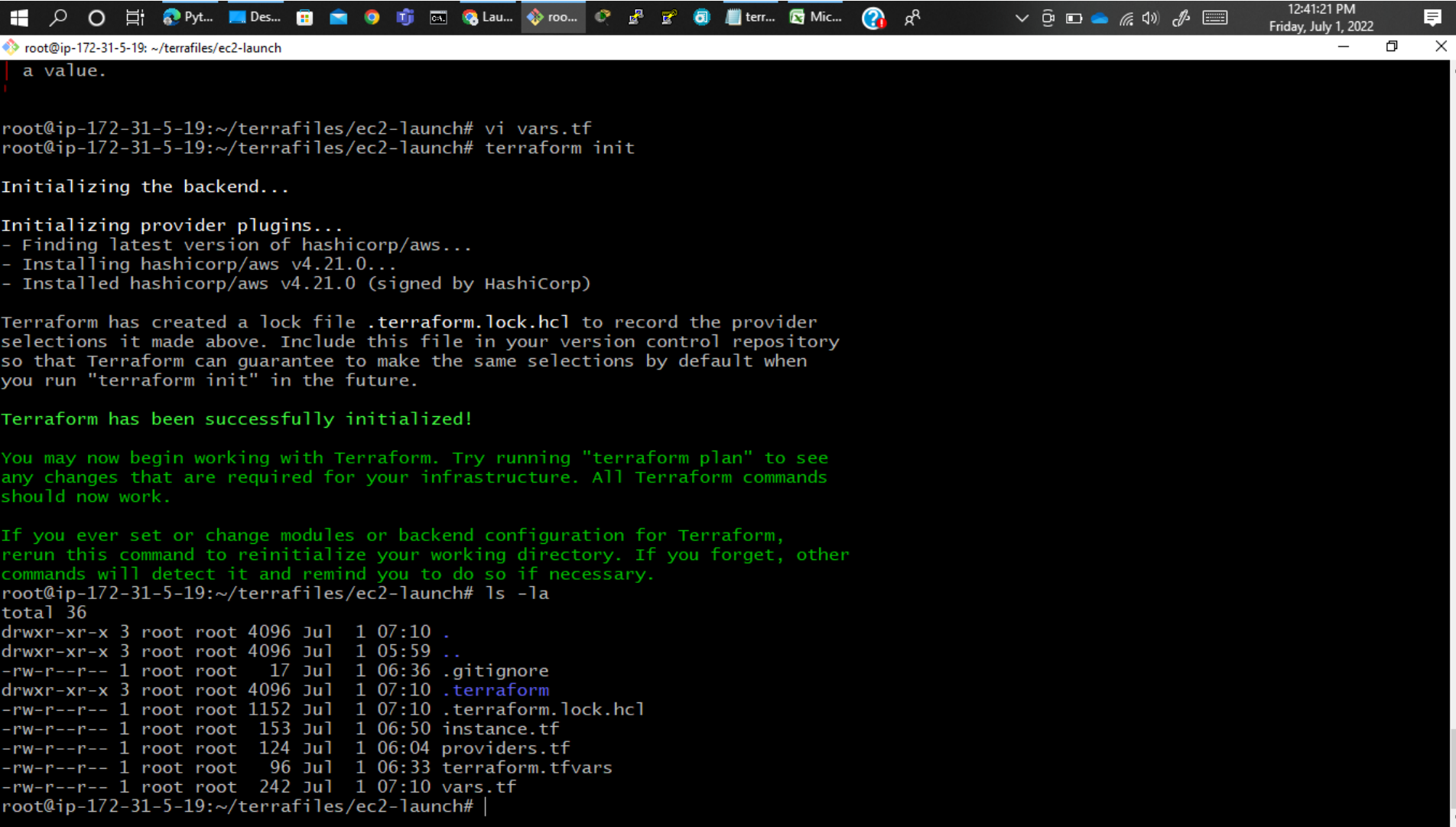
variable AMIS {
    type = map
    default = {
        ap-south-1 = "ami-07eaf27c7c4a884cf"
        us-east-1 = "ami-052efd3df9dad4825"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat instance.tf
resource "aws_instance" "my_inst"{
    ami = "${lookup(var.AMIS,var.AWS_REGION)}"
    instance_type = "t2.micro"
    tags = {
        Name = "First-terra-launch"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch# vi instance.tf
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat instance.tf
resource "aws_instance" "my_inst"{
    ami = "${lookup(var.AMIS,var.AWS_REGION)}"
    instance_type = "t2.micro"
    tags = {
        Name = "First-terra-launch"
        project = "training"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch#
```

Terraform init



```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch
a value.

root@ip-172-31-5-19:~/terrafiles/ec2-launch# vi vars.tf
root@ip-172-31-5-19:~/terrafiles/ec2-launch# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.21.0...
- Installed hashicorp/aws v4.21.0 (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.
root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls -la
total 36
drwxr-xr-x 3 root root 4096 Jul 1 07:10 .
drwxr-xr-x 3 root root 4096 Jul 1 05:59 ..
-rw-r--r-- 1 root root 17 Jul 1 06:36 .gitignore
drwxr-xr-x 3 root root 4096 Jul 1 07:10 .terraform
-rw-r--r-- 1 root root 1152 Jul 1 07:10 .terraform.lock.hcl
-rw-r--r-- 1 root root 153 Jul 1 06:50 instance.tf
-rw-r--r-- 1 root root 124 Jul 1 06:04 providers.tf
-rw-r--r-- 1 root root 96 Jul 1 06:33 terraform.tfvars
-rw-r--r-- 1 root root 242 Jul 1 07:10 vars.tf
root@ip-172-31-5-19:~/terrafiles/ec2-launch#
```

After terraform init you get .terraform file

root@ip-172-31-5-19: ~/terrafiles/ec2-launch

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.

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls -la

```
total 36
drwxr-xr-x 3 root root 4096 Jul  1 07:10 .
drwxr-xr-x 3 root root 4096 Jul  1 05:59 ..
-rw-r--r-- 1 root root   17 Jul  1 06:36 .gitignore
drwxr-xr-x 3 root root 4096 Jul  1 07:10 .terraform
-rw-r--r-- 1 root root 1152 Jul  1 07:10 .terraform.lock.hcl
-rw-r--r-- 1 root root  153 Jul  1 06:50 instance.tf
-rw-r--r-- 1 root root  124 Jul  1 06:04 providers.tf
-rw-r--r-- 1 root root   96 Jul  1 06:33 terraform.tfvars
-rw-r--r-- 1 root root  242 Jul  1 07:10 vars.tf
```

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform

providers

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform/

providers

root@ip-172-31-5-19:~/terrafiles/ec2-launch# cd providers

-su: cd: providers: No such file or directory

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform/providers

registry.terraform.io

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform/providers/registry.terraform.io

hashicorp

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform/providers/registry.terraform.io/hashicorp

aws

root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform/providers/registry.terraform.io/hashicorp/aws

4.21.0

root@ip-172-31-5-19:~/terrafiles/ec2-launch#

Terraform plan

```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch
ls: cannot access '.terraform.lock.hcl': Not a directory
.terraform.lock.hcl
root@ip-172-31-5-19:~/terrafiles/ec2-launch# ls .terraform.lock.hcl/.terraform.lock.hcl
ls: cannot access '.terraform.lock.hcl/.terraform.lock.hcl': Not a directory
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat .terraform.lock.hcl/.terraform.lock.hcl
cat: .terraform.lock.hcl/.terraform.lock.hcl: Not a directory
root@ip-172-31-5-19:~/terrafiles/ec2-launch# clear
root@ip-172-31-5-19:~/terrafiles/ec2-launch# 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.my_inst will be created
+ resource "aws_instance" "my_inst" {
  + ami              = "ami-07eaf27c7c4a884cf"
  + 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)
  + 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)
```

Terraform plan it shows the whats the plans for launch ec2

Terraform Apply :-apply the changes

```
root@ip-172-31-5-19: ~/terraform/ec2-launch
+ http_put_response_hop_limit = (known after apply)
+ http_tokens                  = (known after apply)
+ instance_metadata_tags      = (known after apply)
}

+ network_interface {
+   delete_on_termination = (known after apply)
+   device_index          = (known after apply)
+   network_card_index    = (known after apply)
+   network_interface_id  = (known after apply)
}

+ private_dns_name_options {
+   enable_resource_name_dns_a_record    = (known after apply)
+   enable_resource_name_dns_aaaa_record = (known after apply)
+   hostname_type                       = (known after apply)
}

+ root_block_device {
+   delete_on_termination = (known after apply)
+   device_name            = (known after apply)
+   encrypted              = (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.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value:
```

root@ip-172-31-5-19: ~/terraformfiles/ec2-launch

```
+ http_put_response_hop_limit = (known after apply)
+ http_tokens                  = (known after apply)
+ instance_metadata_tags      = (known after apply)
}

+ network_interface {
+   delete_on_termination = (known after apply)
+   device_index          = (known after apply)
+   network_card_index    = (known after apply)
+   network_interface_id  = (known after apply)
}

+ private_dns_name_options {
+   enable_resource_name_dns_a_record    = (known after apply)
+   enable_resource_name_dns_aaaa_record = (known after apply)
+   hostname_type                        = (known after apply)
}

+ root_block_device {
+   delete_on_termination = (known after apply)
+   device_name            = (known after apply)
+   encrypted              = (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.

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value:

Apply changes :--Yes

```
root@ip-172-31-5-19: ~/terraform/ec2-launch
+ network_interface {
+   delete_on_termination = (known after apply)
+   device_index          = (known after apply)
+   network_card_index    = (known after apply)
+   network_interface_id  = (known after apply)
+ }

+ private_dns_name_options {
+   enable_resource_name_dns_a_record    = (known after apply)
+   enable_resource_name_dns_aaaa_record = (known after apply)
+   hostname_type                        = (known after apply)
+ }

+ root_block_device {
+   delete_on_termination = (known after apply)
+   device_name            = (known after apply)
+   encrypted              = (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.

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.my_inst: Creating...
aws_instance.my_inst: Still creating... [10s elapsed]
aws_instance.my_inst: Still creating... [20s elapsed]
```

Apply changes successfully

```
root@ip-172-31-5-19: ~/terraformfiles/ec2-launch
+ network_interface_id = (known after apply)
}

+ private_dns_name_options {
+   enable_resource_name_dns_a_record   = (known after apply)
+   enable_resource_name_dns_aaaa_record = (known after apply)
+   hostname_type                       = (known after apply)
}

+ root_block_device {
+   delete_on_termination = (known after apply)
+   device_name            = (known after apply)
+   encrypted              = (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.

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.my_inst: Creating...
aws_instance.my_inst: Still creating... [10s elapsed]
aws_instance.my_inst: Still creating... [20s elapsed]
aws_instance.my_inst: Still creating... [30s elapsed]
aws_instance.my_inst: Creation complete after 31s [id=i-0ddd97a0d4552ecf9]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-5-19:~/terraformfiles/ec2-launch#
```

Ec2 instance launched successfully

root@ip-172-31-5-19: ~/terraform/ec2-launch

Instances | EC2 Management Console | Docs overview | hashicorp/aws | +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:

aws Services Search for services, features, blogs, docs, and more [Alt+S] Mumbai Yaswanth

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
Capacity Reservations

Images

Instances (1/3) Info

Search

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	-	i-07752ec6df4febe11	Stopped	t2.micro	-	No alarms	ap-south-
<input type="checkbox"/>	Terraform	i-0bdfec689bb11b1f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-
<input checked="" type="checkbox"/>	First-terra-lau...	i-0ddd97a0d4552ecf9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-

Instance: i-0ddd97a0d4552ecf9 (First-terra-launch)

Details Security Networking Storage Status checks Monitoring Tags

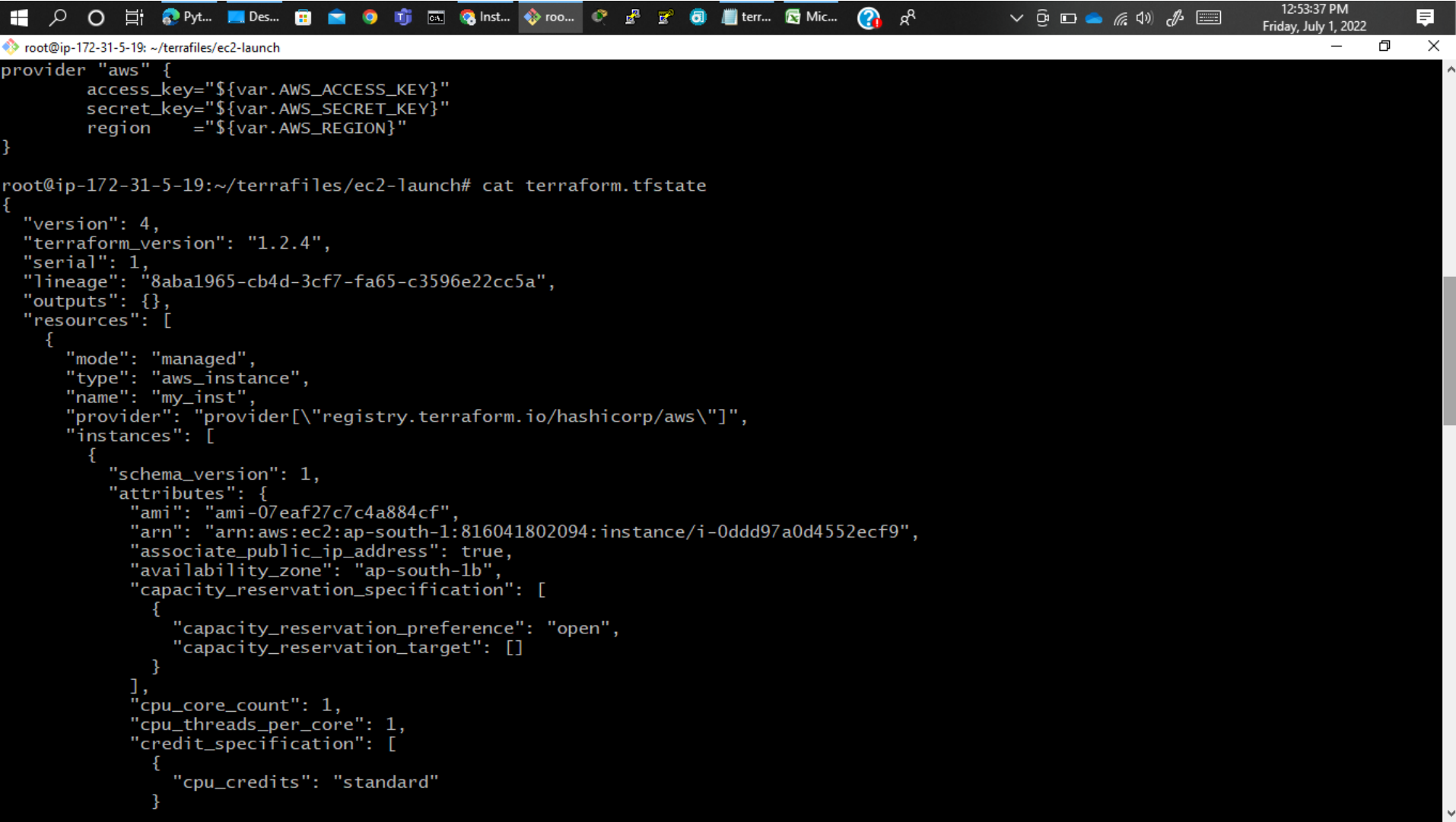
Instance summary Info

Instance ID i-0ddd97a0d4552ecf9 (First-terra-launch)	Public IPv4 address 3.110.45.253 open address	Private IPv4 addresses 172.31.0.231
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-110-45-253.ap-south-1.compute.amazonaws.com open address

https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#InstanceDetails:instanceId=i-0ddd97a0d4552ecf9

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

After launch new instance using terraform new file will be created :---terraform.tfstate

A terminal window screenshot showing the contents of a Terraform configuration file and its state file. The terminal has a dark background with white text. The top bar shows the system clock as 12:53:37 PM on Friday, July 1, 2022. The terminal prompt is root@ip-172-31-5-19: ~/terrafiles/ec2-launch. The first part shows the provider configuration for AWS. The second part shows the output of the 'cat terraform.tfstate' command, displaying the state file's JSON content. The state file includes metadata like version, serial, and lineage, and a list of resources. The resource 'aws_instance.my_inst' is shown with its attributes, including AMI ID, ARN, and capacity reservation details.

```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch
provider "aws" {
    access_key="${var.AWS_ACCESS_KEY}"
    secret_key="${var.AWS_SECRET_KEY}"
    region     ="${var.AWS_REGION}"
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.2.4",
  "serial": 1,
  "lineage": "8aba1965-cb4d-3cf7-fa65-c3596e22cc5a",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "aws_instance",
      "name": "my_inst",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
            "ami": "ami-07eaf27c7c4a884cf",
            "arn": "arn:aws:ec2:ap-south-1:816041802094:instance/i-0ddd97a0d4552ecf9",
            "associate_public_ip_address": true,
            "availability_zone": "ap-south-1b",
            "capacity_reservation_specification": [
              {
                "capacity_reservation_preference": "open",
                "capacity_reservation_target": []
              }
            ],
            "cpu_core_count": 1,
            "cpu_threads_per_core": 1,
            "credit_specification": [
              {
                "cpu_credits": "standard"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Tfstate file give all info about your instace

Do some changes in vars.tf file

```
root@ip-172-31-5-19: ~/terrafiles/ec2-launch
}
]
}
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat vars.tf
variable AWS_ACCESS_KEY{}
variable AWS_SECRET_KEY{}
variable AWS_REGION{
    default = "ap-south-1"
}

variable AMIS {
    type = map
    default = {
        ap-south-1 = "ami-07eaf27c7c4a884cf"
        us-east-1 = "ami-052efd3df9dad4825"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat instance.tf
resource "aws_instance" "my_inst"{
    ami = "${lookup(var.AMIS,var.AWS_REGION)}"
    instance_type = "t2.micro"
    tags = {
        Name = "First-terra-launch"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch# vi instance.tf
root@ip-172-31-5-19:~/terrafiles/ec2-launch# cat instance.tf
resource "aws_instance" "my_inst"{
    ami = "${lookup(var.AMIS,var.AWS_REGION)}"
    instance_type = "t2.micro"
    tags = {
        Name = "First-terra-launch"
        project = "training"
    }
}

root@ip-172-31-5-19:~/terrafiles/ec2-launch#
```

After changes plan and apply Terraform plan

```
root@ip-172-31-5-19: ~/terraformfiles/ec2-launch
}
root@ip-172-31-5-19:~/terraformfiles/ec2-launch# terraform plan
aws_instance.my_inst: Refreshing state... [id=i-0ddd97a0d4552ecf9]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  ~ update in-place

Terraform will perform the following actions:

# aws_instance.my_inst will be updated in-place
~ resource "aws_instance" "my_inst" {
  id = "i-0ddd97a0d4552ecf9"
  ~ tags = {
    + "project" = "training"
    # (1 unchanged element hidden)
  }
  ~ tags_all = {
    + "project" = "training"
    # (1 unchanged element hidden)
  }
  # (29 unchanged attributes hidden)

  # (7 unchanged blocks hidden)
}

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

```
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.
root@ip-172-31-5-19:~/terraformfiles/ec2-launch#
```

Apply the changes :::Terraform apply

```
root@ip-172-31-5-19: ~/terraform/ec2-launch
~ update in-place

Terraform will perform the following actions:

# aws_instance.my_inst will be updated in-place
~ resource "aws_instance" "my_inst" {
  id = "i-0ddd97a0d4552ecf9"
  ~ tags = {
    + "project" = "training"
    # (1 unchanged element hidden)
  }
  ~ tags_all = {
    + "project" = "training"
    # (1 unchanged element hidden)
  }
  # (29 unchanged attributes hidden)

  # (7 unchanged blocks hidden)
}

Plan: 0 to add, 1 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.my_inst: Modifying... [id=i-0ddd97a0d4552ecf9]
aws_instance.my_inst: Modifications complete after 0s [id=i-0ddd97a0d4552ecf9]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
root@ip-172-31-5-19:~/terraform/ec2-launch#
```

Changes will be applied successfully!!!!

Here is the proof

root@ip-172-31-5-19: ~/terraform/ec2-launch

Instances | EC2 Management Console | Docs overview | hashicorp/aws | +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:

aws Services Search for services, features, blogs, docs, and more [Alt+S] Mumbai Yaswanth

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
Capacity Reservations

Images

Instances (1/3) Info

Search

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	-	i-07752ec6df4febe11	Stopped	t2.micro	-	No alarms	ap-south-
<input type="checkbox"/>	Terraform	i-0bdfecc689bb11b1f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-
<input checked="" type="checkbox"/>	First-terra-lau...	i-0ddd97a0d4552ecf9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-

Instance: i-0ddd97a0d4552ecf9 (First-terra-launch)

Tags

Manage tags

Key	Value
Name	First-terra-launch
project	training

Waiting for ap-south-1.console.aws.amazon.com...

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



P.Narmada
padigapatinarmadareddy@gmail.com