ubuntu@ip-172-31-24-238:~/terraform-ws$ sudo nano terraform-casestudy.tf

provider "aws" {

region = "us-east-2"

profile = "default"

}

resource "aws\_instance" "firstec2" {

ami = "ami-0629230e074c580f2"

instance\_type = "t2.micro"

user\_data = <<-EOL

#!/bin/bash -xe

sudo apt-get update

sudo apt-get install apache2

EOL

subnet\_id = "${aws\_subnet.first.id}"

tags = {

Name = "Terraform-casestudy"

}

}

resource "aws\_vpc" "main" {

cidr\_block = "192.168.0.0/16"

enable\_dns\_support = "1"

enable\_dns\_hostnames = "1"

tags = {

Name = "myfirstvpc"

}

}

resource "aws\_subnet" "first" {

availability\_zone = "us-east-2a"

cidr\_block = "192.168.1.0/24"

map\_public\_ip\_on\_launch = "1"

vpc\_id = "${aws\_vpc.main.id}"

tags = {

Name = "myfirstsubnet"

}

}

resource "aws\_default\_security\_group" "default\_myfirst" {

ingress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

vpc\_id = "${aws\_vpc.main.id}"

tags = {

Name = "myfirstsecuritygroup"

}

}

resource "aws\_internet\_gateway" "internet" {

vpc\_id = "${aws\_vpc.main.id}"

tags = {

Name = "myinternetgateway"

}

}

resource "aws\_route" "internet" {

route\_table\_id = "${aws\_vpc.main.default\_route\_table\_id}"

destination\_cidr\_block = "0.0.0.0/0"

gateway\_id = "${aws\_internet\_gateway.internet.id}"

}

resource "aws\_route\_table\_association" "a" {

subnet\_id = "${aws\_subnet.first.id}"

route\_table\_id = "${aws\_vpc.main.default\_route\_table\_id}"

}

resource "aws\_network\_interface" "first" {

subnet\_id = "${aws\_subnet.first.id}"

tags = {

Name = "mynetworkinterface"

}

}

resource "aws\_network\_interface\_attachment" "connection" {

instance\_id = "${aws\_instance.firstec2.id}"

network\_interface\_id = "${aws\_network\_interface.first.id}"

device\_index = 1

}

output "IPs" {

value = "Terraform-casestudy - ${aws\_instance.firstec2.public\_ip}"

}

ubuntu@ip-172-31-24-238:~/terraform-ws$ terraform init

Initializing the backend...

Initializing provider plugins...

- Reusing previous version of hashicorp/aws from the dependency lock file

- Using previously-installed hashicorp/aws v3.66.0

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.

ubuntu@ip-172-31-24-238:~/terraform-ws$ terraform validate

Success! The configuration is valid.

ubuntu@ip-172-31-24-238:~/terraform-ws$ terraform plan -out myplan2

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\_default\_security\_group.default\_myfirst will be created

+ resource "aws\_default\_security\_group" "default\_myfirst" {

+ arn = (known after apply)

+ description = (known after apply)

+ egress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 0

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "-1"

+ security\_groups = []

+ self = false

+ to\_port = 0

},

]

+ id = (known after apply)

+ ingress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 0

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "-1"

+ security\_groups = []

+ self = false

+ to\_port = 0

},

]

+ name = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "myfirstsecuritygroup"

}

+ tags\_all = {

+ "Name" = "myfirstsecuritygroup"

}

+ vpc\_id = (known after apply)

}

# aws\_instance.firstec2 will be created

+ resource "aws\_instance" "firstec2" {

+ ami = "ami-0629230e074c580f2"

+ 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 = (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" = "Terraform-casestudy"

}

+ tags\_all = {

+ "Name" = "Terraform-casestudy"

}

+ tenancy = (known after apply)

+ user\_data = "a3f5acd3c901b2e9c0a8bdb4bde4b313c1ea0a6d"

+ user\_data\_base64 = (known after apply)

+ vpc\_security\_group\_ids = (known after apply)

+ capacity\_reservation\_specification {

+ capacity\_reservation\_preference = (known after apply)

+ capacity\_reservation\_target {

+ capacity\_reservation\_id = (known after apply)

}

}

+ ebs\_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)

+ snapshot\_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)

}

+ enclave\_options {

+ enabled = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_put\_response\_hop\_limit = (known after apply)

+ http\_tokens = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = (known after apply)

+ device\_index = (known after apply)

+ network\_interface\_id = (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)

}

}

# aws\_internet\_gateway.internet will be created

+ resource "aws\_internet\_gateway" "internet" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "myinternetgateway"

}

+ tags\_all = {

+ "Name" = "myinternetgateway"

}

+ vpc\_id = (known after apply)

}

# aws\_network\_interface.first will be created

+ resource "aws\_network\_interface" "first" {

+ arn = (known after apply)

+ id = (known after apply)

+ interface\_type = (known after apply)

+ ipv4\_prefix\_count = (known after apply)

+ ipv4\_prefixes = (known after apply)

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ ipv6\_prefix\_count = (known after apply)

+ ipv6\_prefixes = (known after apply)

+ mac\_address = (known after apply)

+ outpost\_arn = (known after apply)

+ owner\_id = (known after apply)

+ private\_dns\_name = (known after apply)

+ private\_ip = (known after apply)

+ private\_ips = (known after apply)

+ private\_ips\_count = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "mynetworkinterface"

}

+ tags\_all = {

+ "Name" = "mynetworkinterface"

}

+ attachment {

+ attachment\_id = (known after apply)

+ device\_index = (known after apply)

+ instance = (known after apply)

}

}

# aws\_network\_interface\_attachment.connection will be created

+ resource "aws\_network\_interface\_attachment" "connection" {

+ attachment\_id = (known after apply)

+ device\_index = 1

+ id = (known after apply)

+ instance\_id = (known after apply)

+ network\_interface\_id = (known after apply)

+ status = (known after apply)

}

# aws\_route.internet will be created

+ resource "aws\_route" "internet" {

+ destination\_cidr\_block = "0.0.0.0/0"

+ gateway\_id = (known after apply)

+ id = (known after apply)

+ instance\_id = (known after apply)

+ instance\_owner\_id = (known after apply)

+ network\_interface\_id = (known after apply)

+ origin = (known after apply)

+ route\_table\_id = (known after apply)

+ state = (known after apply)

}

# aws\_route\_table\_association.a will be created

+ resource "aws\_route\_table\_association" "a" {

+ id = (known after apply)

+ route\_table\_id = (known after apply)

+ subnet\_id = (known after apply)

}

# aws\_subnet.first will be created

+ resource "aws\_subnet" "first" {

+ arn = (known after apply)

+ assign\_ipv6\_address\_on\_creation = false

+ availability\_zone = "us-east-2a"

+ availability\_zone\_id = (known after apply)

+ cidr\_block = "192.168.1.0/24"

+ id = (known after apply)

+ ipv6\_cidr\_block\_association\_id = (known after apply)

+ map\_public\_ip\_on\_launch = true

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "myfirstsubnet"

}

+ tags\_all = {

+ "Name" = "myfirstsubnet"

}

+ vpc\_id = (known after apply)

}

# aws\_vpc.main will be created

+ resource "aws\_vpc" "main" {

+ arn = (known after apply)

+ assign\_generated\_ipv6\_cidr\_block = false

+ cidr\_block = "192.168.0.0/16"

+ default\_network\_acl\_id = (known after apply)

+ default\_route\_table\_id = (known after apply)

+ default\_security\_group\_id = (known after apply)

+ dhcp\_options\_id = (known after apply)

+ enable\_classiclink = (known after apply)

+ enable\_classiclink\_dns\_support = (known after apply)

+ enable\_dns\_hostnames = true

+ enable\_dns\_support = true

+ id = (known after apply)

+ instance\_tenancy = "default"

+ ipv6\_association\_id = (known after apply)

+ ipv6\_cidr\_block = (known after apply)

+ main\_route\_table\_id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "myfirstvpc"

}

+ tags\_all = {

+ "Name" = "myfirstvpc"

}

}

Plan: 9 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ IPs = (known after apply)

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Saved the plan to: myplan2

To perform exactly these actions, run the following command to apply:

terraform apply "myplan2"

ubuntu@ip-172-31-24-238:~/terraform-ws$ terraform apply "myplan2"

aws\_vpc.main: Creating...

aws\_vpc.main: Still creating... [10s elapsed]

aws\_vpc.main: Creation complete after 11s [id=vpc-00ae182c786741a3e]

aws\_default\_security\_group.default\_myfirst: Creating...

aws\_subnet.first: Creating...

aws\_internet\_gateway.internet: Creating...

aws\_internet\_gateway.internet: Creation complete after 1s [id=igw-0e42c7493fb9299e5]

aws\_route.internet: Creating...

aws\_route.internet: Creation complete after 0s [id=r-rtb-0e4ad3b76e89f31821080289494]

aws\_default\_security\_group.default\_myfirst: Creation complete after 2s [id=sg-0796372dbc8bf9444]

aws\_subnet.first: Still creating... [10s elapsed]

aws\_subnet.first: Creation complete after 11s [id=subnet-0e9136991c1ad6196]

aws\_instance.firstec2: Creating...

aws\_route\_table\_association.a: Creating...

aws\_network\_interface.first: Creating...

aws\_route\_table\_association.a: Creation complete after 0s [id=rtbassoc-0e290909b1e7ce5ad]

aws\_network\_interface.first: Creation complete after 1s [id=eni-07703ce56e4263f50]

aws\_instance.firstec2: Still creating... [10s elapsed]

aws\_instance.firstec2: Still creating... [20s elapsed]

aws\_instance.firstec2: Still creating... [30s elapsed]

aws\_instance.firstec2: Creation complete after 31s [id=i-0f0651da8f802d979]

aws\_network\_interface\_attachment.connection: Creating...

aws\_network\_interface\_attachment.connection: Still creating... [10s elapsed]

aws\_network\_interface\_attachment.connection: Creation complete after 14s [id=eni-attach-0b95474042e3543e9]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

IPs = "Terraform-casestudy - 3.129.39.49"

ubuntu@ip-172-31-24-238:~/terraform-ws$

