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

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

aws-cli/1.18.69 Python/3.8.10 Linux/5.11.0-1020-aws botocore/1.16.19

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

example.tf terraform\_1.0.10\_linux\_amd64.zip

ubuntu@ip-172-31-24-238:~/terraform-ws$ vi example.tf

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

provider "aws" {

profile = "default"

region = "us-east-2"

}

resource "aws\_instance" "example" {

ami = "ami-0d718c3d715cec4a7"

instance\_type = "t2.micro"

tags = {

Name = "myinstance"

}

}

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

example.tf terraform\_1.0.10\_linux\_amd64.zip

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

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

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...

- Installing hashicorp/aws v3.66.0...

- Installed hashicorp/aws v3.66.0 (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-172-31-24-238:~/terraform-ws$ terraform validate

Success! The configuration is valid.

ubuntu@ip-172-31-24-238:~/terraform-ws$ 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.example will be created

+ resource "aws\_instance" "example" {

+ ami = "ami-0d718c3d715cec4a7"

+ 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" = "myinstance"

}

+ tags\_all = {

+ "Name" = "myinstance"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ 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)

}

}

Plan: 1 to add, 0 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.

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

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.example will be created

+ resource "aws\_instance" "example" {

+ ami = "ami-0d718c3d715cec4a7"

+ 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" = "myinstance"

}

+ tags\_all = {

+ "Name" = "myinstance"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ 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)

}

}

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

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

Saved the plan to: myplan

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

terraform apply "myplan"

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

aws\_instance.example: Creating...

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

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

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

aws\_instance.example: Creation complete after 32s [id=i-0f1a2304f86068501]

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

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

# aws\_instance.example:

resource "aws\_instance" "example" {

ami = "ami-0d718c3d715cec4a7"

arn = "arn:aws:ec2:us-east-2:626284337142:instance/i-0f1a2304f86068501"

associate\_public\_ip\_address = true

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

cpu\_core\_count = 1

cpu\_threads\_per\_core = 1

disable\_api\_termination = false

ebs\_optimized = false

get\_password\_data = false

hibernation = false

id = "i-0f1a2304f86068501"

instance\_initiated\_shutdown\_behavior = "stop"

instance\_state = "running"

instance\_type = "t2.micro"

ipv6\_address\_count = 0

ipv6\_addresses = []

monitoring = false

primary\_network\_interface\_id = "eni-091bb3ee60d5697c3"

private\_dns = "ip-172-31-40-0.us-east-2.compute.internal"

private\_ip = "172.31.40.0"

public\_dns = "ec2-3-144-127-17.us-east-2.compute.amazonaws.com"

public\_ip = "3.144.127.17"

secondary\_private\_ips = []

security\_groups = [

"default",

]

source\_dest\_check = true

subnet\_id = "subnet-931c37df"

tags = {

"Name" = "myinstance"

}

tags\_all = {

"Name" = "myinstance"

}

tenancy = "default"

vpc\_security\_group\_ids = [

"sg-51c0a61a",

]

capacity\_reservation\_specification {

capacity\_reservation\_preference = "open"

}

credit\_specification {

cpu\_credits = "standard"

}

enclave\_options {

enabled = false

}

metadata\_options {

http\_endpoint = "enabled"

http\_put\_response\_hop\_limit = 1

http\_tokens = "optional"

}

root\_block\_device {

delete\_on\_termination = true

device\_name = "/dev/xvda"

encrypted = false

iops = 100

tags = {}

throughput = 0

volume\_id = "vol-01eeea3c5e37f62dc"

volume\_size = 8

volume\_type = "gp2"

}

}