CHANGE INFRASTRUCTURE

~ ami = "ami-079b5e5b3971bd10d" -> "ami-0f2e255ec956ade7f" # forces replacement

Here we’ve changed the ami from Linux to Ubuntu

C:\Users\Administrator\Downloads\terraform\_1.1.9\_windows\_amd64>terraform apply

aws\_instance.app\_server: Refreshing state... [id=i-096b84e0dc6137aa4]

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

-/+ destroy and then create replacement

Terraform will perform the following actions:

# aws\_instance.app\_server must be replaced

-/+ resource "aws\_instance" "app\_server" {

~ ami = "ami-079b5e5b3971bd10d" -> "ami-0f2e255ec956ade7f" # forces replacement

~ arn = "arn:aws:ec2:ap-south-1:589711277245:instance/i-096b84e0dc6137aa4" -> (known after apply)

~ associate\_public\_ip\_address = true -> (known after apply)

~ availability\_zone = "ap-south-1b" -> (known after apply)

~ cpu\_core\_count = 1 -> (known after apply)

~ cpu\_threads\_per\_core = 1 -> (known after apply)

~ disable\_api\_termination = false -> (known after apply)

~ ebs\_optimized = false -> (known after apply)

- hibernation = false -> null

+ host\_id = (known after apply)

~ id = "i-096b84e0dc6137aa4" -> (known after apply)

~ instance\_initiated\_shutdown\_behavior = "stop" -> (known after apply)

~ instance\_state = "running" -> (known after apply)

~ ipv6\_address\_count = 0 -> (known after apply)

~ ipv6\_addresses = [] -> (known after apply)

+ key\_name = (known after apply)

~ monitoring = false -> (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 = "eni-08dfb798348844585" -> (known after apply)

~ private\_dns = "ip-172-31-13-38.ap-south-1.compute.internal" -> (known after apply)

~ private\_ip = "172.31.13.38" -> (known after apply)

~ public\_dns = "ec2-3-109-207-48.ap-south-1.compute.amazonaws.com" -> (known after apply)

~ public\_ip = "3.109.207.48" -> (known after apply)

~ secondary\_private\_ips = [] -> (known after apply)

~ security\_groups = [

- "default",

] -> (known after apply)

~ subnet\_id = "subnet-08d87886393b54ee4" -> (known after apply)

tags = {

"Name" = "aws\_instance"

}

~ tenancy = "default" -> (known after apply)

+ user\_data = (known after apply)

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

~ vpc\_security\_group\_ids = [

- "sg-0e768ef13a6895c54",

] -> (known after apply)

# (5 unchanged attributes hidden)

~ capacity\_reservation\_specification {

~ capacity\_reservation\_preference = "open" -> (known after apply)

+ capacity\_reservation\_target {

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

+ capacity\_reservation\_resource\_group\_arn = (known after apply)

}

}

- credit\_specification {

- cpu\_credits = "standard" -> null

}

+ 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 = false -> (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

~ maintenance\_options {

~ auto\_recovery = "default" -> (known after apply)

}

~ metadata\_options {

~ http\_endpoint = "enabled" -> (known after apply)

~ http\_put\_response\_hop\_limit = 1 -> (known after apply)

~ http\_tokens = "optional" -> (known after apply)

~ instance\_metadata\_tags = "disabled" -> (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)

}

~ root\_block\_device {

~ delete\_on\_termination = true -> (known after apply)

~ device\_name = "/dev/xvda" -> (known after apply)

~ encrypted = false -> (known after apply)

~ iops = 100 -> (known after apply)

+ kms\_key\_id = (known after apply)

~ tags = {} -> (known after apply)

~ throughput = 0 -> (known after apply)

~ volume\_id = "vol-0b8f4b0a5be00e0de" -> (known after apply)

~ volume\_size = 8 -> (known after apply)

~ volume\_type = "gp2" -> (known after apply)

}

}

Plan: 1 to add, 0 to change, 1 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: Destroying... [id=i-096b84e0dc6137aa4]

aws\_instance.app\_server: Still destroying... [id=i-096b84e0dc6137aa4, 10s elapsed]

aws\_instance.app\_server: Still destroying... [id=i-096b84e0dc6137aa4, 20s elapsed]

aws\_instance.app\_server: Still destroying... [id=i-096b84e0dc6137aa4, 30s elapsed]

aws\_instance.app\_server: Still destroying... [id=i-096b84e0dc6137aa4, 40s elapsed]

aws\_instance.app\_server: Destruction complete after 40s

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 1m2s [id=i-03887159802136781]

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

C:\Users\Administrator\Downloads\terraform\_1.1.9\_windows\_amd64>aws --version

aws-cli/2.7.0 Python/3.9.11 Windows/10 exe/AMD64 prompt/off