**Output.tf**

**output "instance\_id" {**

**description = "ID of the EC2 instance"**

**value = aws\_instance.app\_server.id**

**}**

**output "instance\_public\_ip" {**

**description = "Public IP address of the EC2 instance"**

**value = aws\_instance.app\_server.public\_ip**

**}**

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

aws\_instance.app\_server: Refreshing state... [id=i-0a19ec3e6d3e22485]

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.app\_server will be updated in-place

~ resource "aws\_instance" "app\_server" {

id = "i-0a19ec3e6d3e22485"

~ tags = {

~ "Name" = "YetAnotherName" -> "aws-instance"

}

~ tags\_all = {

~ "Name" = "YetAnotherName" -> "aws-instance"

}

# (25 unchanged attributes hidden)

# (5 unchanged blocks hidden)

}

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

Changes to Outputs:

+ instance\_id = "i-0a19ec3e6d3e22485"

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: Modifying... [id=i-0a19ec3e6d3e22485]

aws\_instance.app\_server: Modifications complete after 1s [id=i-0a19ec3e6d3e22485]

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

Outputs:

instance\_id = "i-0a19ec3e6d3e22485"

instance\_public\_ip = ""

C:\Users\user\Downloads\terraform\_1.1.9\_windows\_amd64>terraform output

instance\_id = "i-0a19ec3e6d3e22485"

instance\_public\_ip = ""

C:\Users\user\Downloads\terraform\_1.1.9\_windows\_amd64>terraform output instance\_id

"i-0a19ec3e6d3e22485"

C:\Users\user\Downloads\terraform\_1.1.9\_windows\_amd64>terraform output instance\_public\_id

╷

│ Error: Output "instance\_public\_id" not found

│

│ The output variable requested could not be found in the state file. If you recently added this to your configuration,

│ be sure to run `terraform apply`, since the state won't be updated with new output variables until that command is

│ run.

╵

C:\Users\user\Downloads\terraform\_1.1.9\_windows\_amd64>terraform destroy

aws\_instance.app\_server: Refreshing state... [id=i-0a19ec3e6d3e22485]

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

following symbols:

- destroy

Terraform will perform the following actions:

# aws\_instance.app\_server will be destroyed

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

- ami = "ami-0f2e255ec956ade7f" -> null

- arn = "arn:aws:ec2:ap-south-1:730064952606:instance/i-0a19ec3e6d3e22485" -> null

- associate\_public\_ip\_address = false -> null

- availability\_zone = "ap-south-1b" -> null

- cpu\_core\_count = 1 -> null

- cpu\_threads\_per\_core = 1 -> null

- disable\_api\_termination = false -> null

- ebs\_optimized = false -> null

- get\_password\_data = false -> null

- hibernation = false -> null

- id = "i-0a19ec3e6d3e22485" -> null

- instance\_initiated\_shutdown\_behavior = "stop" -> null

- instance\_state = "stopped" -> null

- instance\_type = "t2.micro" -> null

- ipv6\_address\_count = 0 -> null

- ipv6\_addresses = [] -> null

- monitoring = false -> null

- primary\_network\_interface\_id = "eni-0d90ca6d6552aae04" -> null

- private\_dns = "ip-172-31-15-176.ap-south-1.compute.internal" -> null

- private\_ip = "172.31.15.176" -> null

- secondary\_private\_ips = [] -> null

- security\_groups = [

- "default",

] -> null

- source\_dest\_check = true -> null

- subnet\_id = "subnet-03deb119a50582552" -> null

- tags = {

- "Name" = "aws-instance"

} -> null

- tags\_all = {

- "Name" = "aws-instance"

} -> null

- tenancy = "default" -> null

- vpc\_security\_group\_ids = [

- "sg-08700754d8dbb2b5e",

] -> null

- capacity\_reservation\_specification {

- capacity\_reservation\_preference = "open" -> null

}

- credit\_specification {

- cpu\_credits = "standard" -> null

}

- enclave\_options {

- enabled = false -> null

}

- metadata\_options {

- http\_endpoint = "enabled" -> null

- http\_put\_response\_hop\_limit = 1 -> null

- http\_tokens = "optional" -> null

- instance\_metadata\_tags = "disabled" -> null

}

- root\_block\_device {

- delete\_on\_termination = true -> null

- device\_name = "/dev/sda1" -> null

- encrypted = false -> null

- iops = 100 -> null

- tags = {} -> null

- throughput = 0 -> null

- volume\_id = "vol-0143530189610c95f" -> null

- volume\_size = 8 -> null

- volume\_type = "gp2" -> null

}

}

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

Changes to Outputs:

- instance\_id = "i-0a19ec3e6d3e22485" -> null

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.

There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws\_instance.app\_server: Destroying... [id=i-0a19ec3e6d3e22485]

aws\_instance.app\_server: Still destroying... [id=i-0a19ec3e6d3e22485, 10s elapsed]

aws\_instance.app\_server: Destruction complete after 11s

Destroy complete! Resources: 1 destroyed.