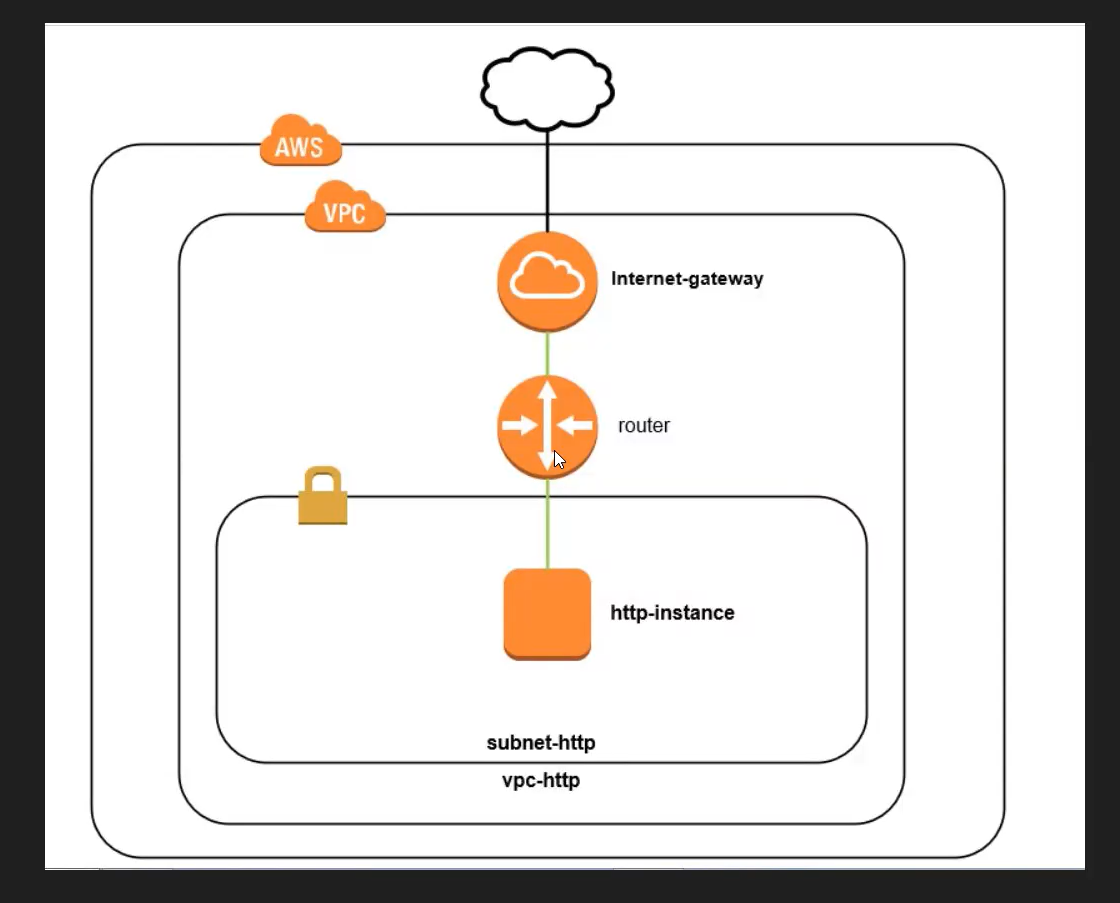
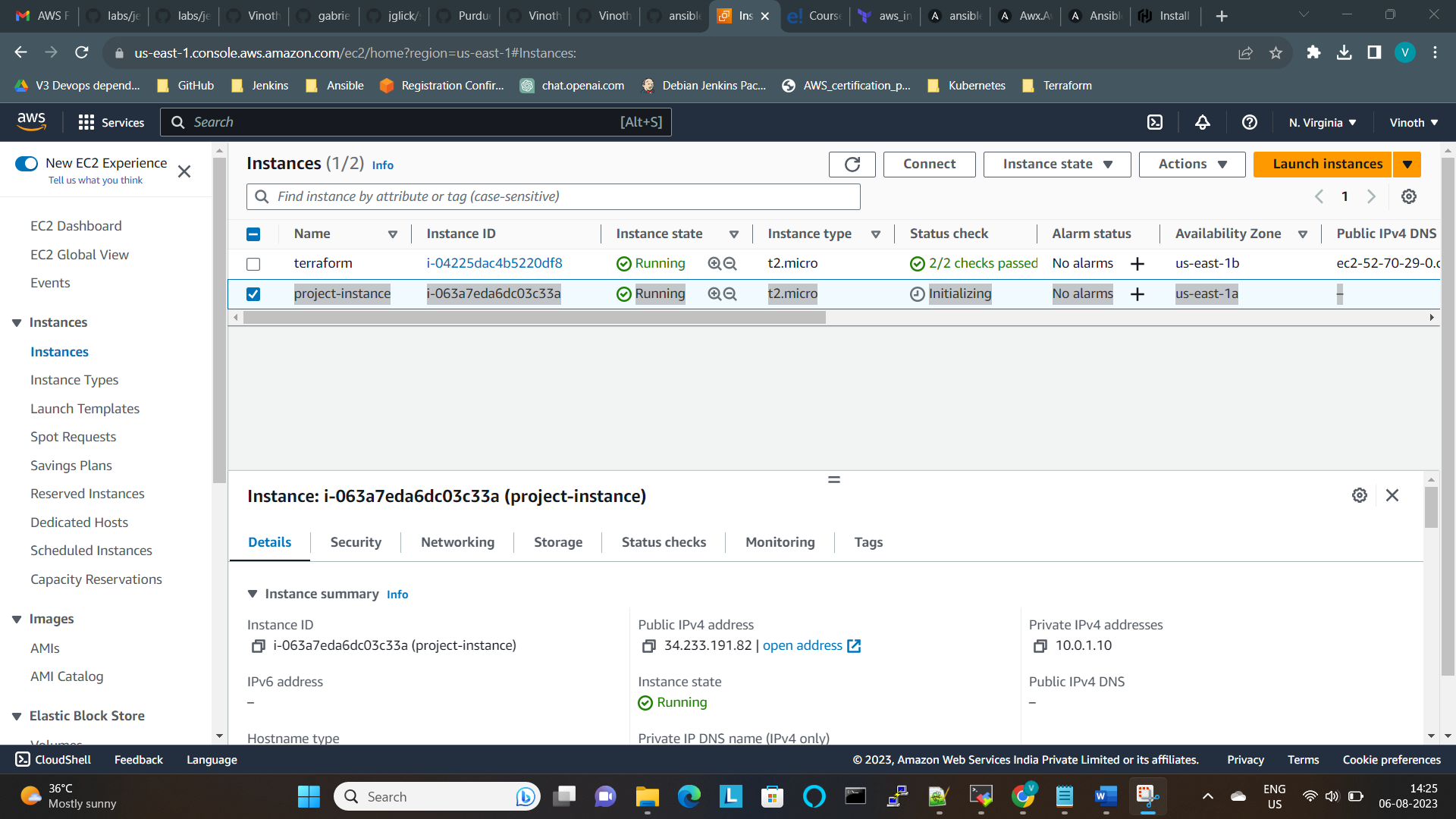
**PROJECT 1 – TERRAFORM**





AWS CLI INSTALLATION

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

ACCESS KEY: AKIAQVVSC6FQJNFMEIXO

SECRET KEY: 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

TERRAFORM INSTALLATION

lsb\_release -a

wget https://releases.hashicorp.com/terraform/1.5.4/terraform\_1.5.4\_linux\_amd64.zip

ll

unzip terraform\_1.5.4\_linux\_amd64.zip

sudo apt install unzip

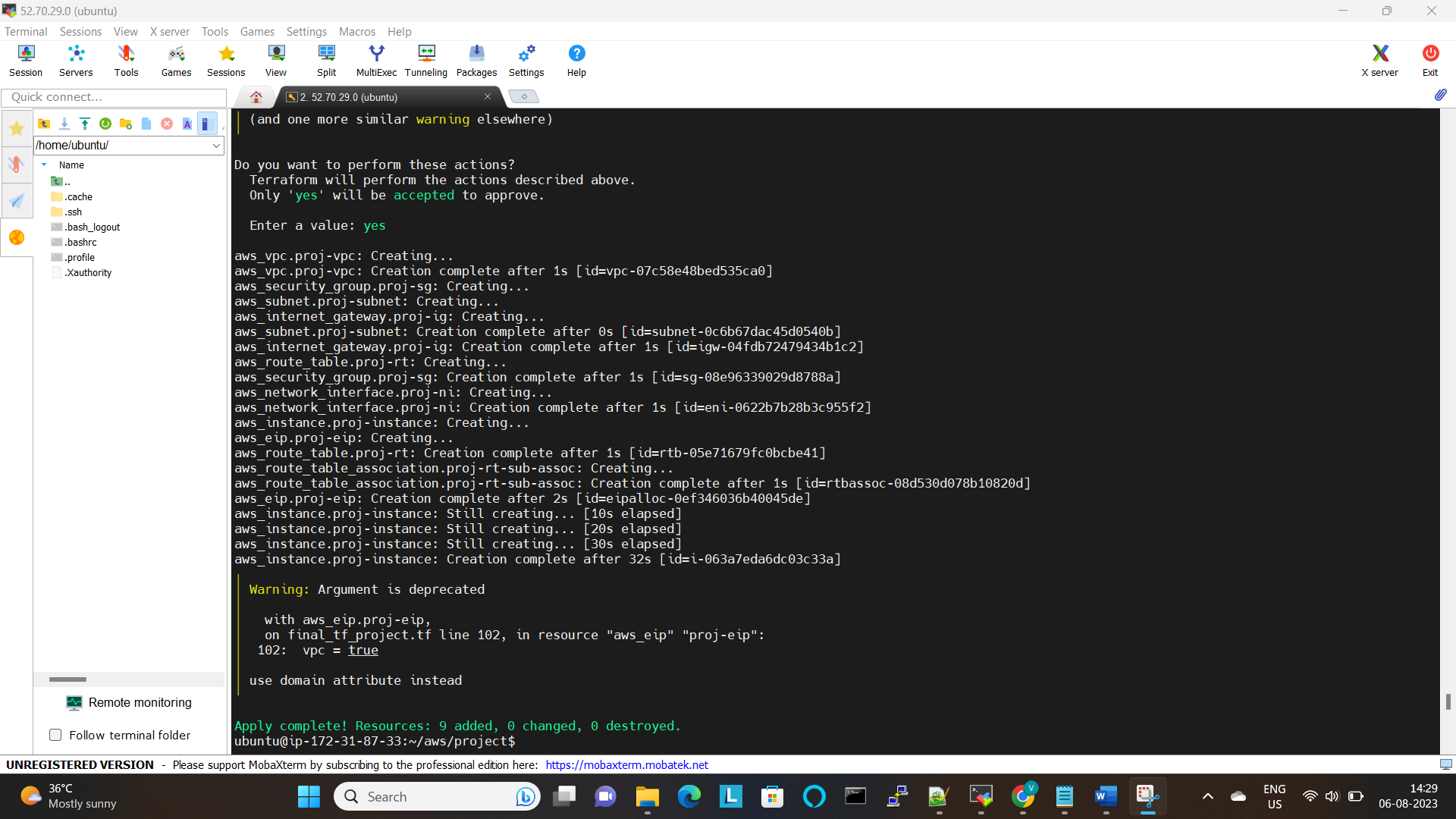
sudo unzip terraform\_1.5.4\_linux\_amd64.zip

ll

mv terraform /usr/local/bin/

sudo mv terraform /usr/local/bin/

which terraform



ubuntu@ip-172-31-87-33:~/aws/project$ terraform apply

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\_eip.proj-eip will be created

+ resource "aws\_eip" "proj-eip" {

+ allocation\_id = (known after apply)

+ associate\_with\_private\_ip = "10.0.1.10"

+ association\_id = (known after apply)

+ carrier\_ip = (known after apply)

+ customer\_owned\_ip = (known after apply)

+ domain = (known after apply)

+ id = (known after apply)

+ instance = (known after apply)

+ network\_border\_group = (known after apply)

+ network\_interface = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ public\_ipv4\_pool = (known after apply)

+ tags\_all = (known after apply)

+ vpc = true

}

# aws\_instance.proj-instance will be created

+ resource "aws\_instance" "proj-instance" {

+ ami = "ami-053b0d53c279acc90"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = "us-east-1a"

+ 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)

+ host\_resource\_group\_arn = (known after apply)

+ iam\_instance\_profile = (known after apply)

+ id = (known after apply)

+ instance\_initiated\_shutdown\_behavior = (known after apply)

+ instance\_lifecycle = (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 = "demo"

+ 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)

+ spot\_instance\_request\_id = (known after apply)

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "project-instance"

}

+ tags\_all = {

+ "Name" = "project-instance"

}

+ tenancy = (known after apply)

+ user\_data = "1f89b4fde46ba68096c8d6eb801a3434fb72798e"

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

+ user\_data\_replace\_on\_change = false

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

+ network\_interface {

+ delete\_on\_termination = false

+ device\_index = 0

+ network\_card\_index = 0

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

}

}

# aws\_internet\_gateway.proj-ig will be created

+ resource "aws\_internet\_gateway" "proj-ig" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "gateway1"

}

+ tags\_all = {

+ "Name" = "gateway1"

}

+ vpc\_id = (known after apply)

}

# aws\_network\_interface.proj-ni will be created

+ resource "aws\_network\_interface" "proj-ni" {

+ 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\_address\_list = (known after apply)

+ ipv6\_address\_list\_enabled = false

+ 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\_ip\_list = (known after apply)

+ private\_ip\_list\_enabled = false

+ private\_ips = [

+ "10.0.1.10",

]

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

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tags\_all = (known after apply)

}

# aws\_route\_table.proj-rt will be created

+ resource "aws\_route\_table" "proj-rt" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ propagating\_vgws = (known after apply)

+ route = [

+ {

+ carrier\_gateway\_id = ""

+ cidr\_block = ""

+ core\_network\_arn = ""

+ destination\_prefix\_list\_id = ""

+ egress\_only\_gateway\_id = ""

+ gateway\_id = (known after apply)

+ ipv6\_cidr\_block = "::/0"

+ local\_gateway\_id = ""

+ nat\_gateway\_id = ""

+ network\_interface\_id = ""

+ transit\_gateway\_id = ""

+ vpc\_endpoint\_id = ""

+ vpc\_peering\_connection\_id = ""

},

+ {

+ carrier\_gateway\_id = ""

+ cidr\_block = "0.0.0.0/0"

+ core\_network\_arn = ""

+ destination\_prefix\_list\_id = ""

+ egress\_only\_gateway\_id = ""

+ gateway\_id = (known after apply)

+ ipv6\_cidr\_block = ""

+ local\_gateway\_id = ""

+ nat\_gateway\_id = ""

+ network\_interface\_id = ""

+ transit\_gateway\_id = ""

+ vpc\_endpoint\_id = ""

+ vpc\_peering\_connection\_id = ""

},

]

+ tags = {

+ "Name" = "rt1"

}

+ tags\_all = {

+ "Name" = "rt1"

}

+ vpc\_id = (known after apply)

}

# aws\_route\_table\_association.proj-rt-sub-assoc will be created

+ resource "aws\_route\_table\_association" "proj-rt-sub-assoc" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# aws\_security\_group.proj-sg will be created

+ resource "aws\_security\_group" "proj-sg" {

+ arn = (known after apply)

+ description = "Enable web traffic for the project"

+ egress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 0

+ ipv6\_cidr\_blocks = [

+ "::/0",

]

+ 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 = "HTTP traffic"

+ from\_port = 80

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 80

},

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = "HTTPS traffic"

+ from\_port = 443

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 443

},

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = "SSH port"

+ from\_port = 22

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 22

},

]

+ name = "proj-sg"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "proj-sg1"

}

+ tags\_all = {

+ "Name" = "proj-sg1"

}

+ vpc\_id = (known after apply)

}

# aws\_subnet.proj-subnet will be created

+ resource "aws\_subnet" "proj-subnet" {

+ arn = (known after apply)

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

+ availability\_zone = "us-east-1a"

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

+ cidr\_block = "10.0.1.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "subnet1"

}

+ tags\_all = {

+ "Name" = "subnet1"

}

+ vpc\_id = (known after apply)

}

# aws\_vpc.proj-vpc will be created

+ resource "aws\_vpc" "proj-vpc" {

+ arn = (known after apply)

+ cidr\_block = "10.0.0.0/16"

+ 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\_dns\_hostnames = (known after apply)

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

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

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

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

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

+ owner\_id = (known after apply)

+ tags\_all = (known after apply)

}

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

╷

│ Warning: Argument is deprecated

│

│ with aws\_eip.proj-eip,

│ on final\_tf\_project.tf line 102, in resource "aws\_eip" "proj-eip":

│ 102: vpc = true

│

│ use domain attribute instead

│

│ (and one more similar warning elsewhere)

╵

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\_vpc.proj-vpc: Creating...

aws\_vpc.proj-vpc: Creation complete after 1s [id=vpc-07c58e48bed535ca0]

aws\_security\_group.proj-sg: Creating...

aws\_subnet.proj-subnet: Creating...

aws\_internet\_gateway.proj-ig: Creating...

aws\_subnet.proj-subnet: Creation complete after 0s [id=subnet-0c6b67dac45d0540b]

aws\_internet\_gateway.proj-ig: Creation complete after 1s [id=igw-04fdb72479434b1c2]

aws\_route\_table.proj-rt: Creating...

aws\_security\_group.proj-sg: Creation complete after 1s [id=sg-08e96339029d8788a]

aws\_network\_interface.proj-ni: Creating...

aws\_network\_interface.proj-ni: Creation complete after 1s [id=eni-0622b7b28b3c955f2]

aws\_instance.proj-instance: Creating...

aws\_eip.proj-eip: Creating...

aws\_route\_table.proj-rt: Creation complete after 1s [id=rtb-05e71679fc0bcbe41]

aws\_route\_table\_association.proj-rt-sub-assoc: Creating...

aws\_route\_table\_association.proj-rt-sub-assoc: Creation complete after 1s [id=rtbassoc-08d530d078b10820d]

aws\_eip.proj-eip: Creation complete after 2s [id=eipalloc-0ef346036b40045de]

aws\_instance.proj-instance: Still creating... [10s elapsed]

aws\_instance.proj-instance: Still creating... [20s elapsed]

aws\_instance.proj-instance: Still creating... [30s elapsed]

aws\_instance.proj-instance: Creation complete after 32s [id=i-063a7eda6dc03c33a]

╷

│ Warning: Argument is deprecated

│

│ with aws\_eip.proj-eip,

│ on final\_tf\_project.tf line 102, in resource "aws\_eip" "proj-eip":

│ 102: vpc = true

│

│ use domain attribute instead

╵

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

ubuntu@ip-172-31-87-33:~/aws/project$

