CONNECTING EC-2 INSTANCE THROUGH MOBAXTERM

login as: ec2-user

switch to root user

[ec2-user@ip-172-31-26-108 ~]$ sudo bash

sudo yum install -y yum-utils is run to ensure that you have all the necessary tools to manage repositories and dependencies effectively, which might be required for the smooth installation of Terraform.

[root@ip-172-31-26-108 ec2-user]# sudo yum install -y yum-utils

Last metadata expiration check: 0:42:28 ago on Wed Aug 28 13:18:10 2024.

Package dnf-utils-4.3.0-13.amzn2023.0.4.noarch is already installed.

Dependencies resolved.

Nothing to do.

Complete!

This command will have Access to HashiCorp Packages, Easier Installation, Automatic Updates from the official source, Repository Configurations of terraform installation.

[root@ip-172-31-26-108 ec2-user]# sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

Adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

Installation of terraform (automatically without confirmation input for installation because of -y)

[root@ip-172-31-26-108 ec2-user]# sudo yum -y install terraform

Complete!

#

[root@ip-172-31-26-108 ec2-user]# terraform --version

Terraform v1.9.5

on linux_amd64

[root@ip-172-31-26-108 ec2-user]# mkdir terraform

[root@ip-172-31-26-108 ec2-user]# cd terraform

```
[root@ip-172-31-26-108 terraform]# terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
[root@ip-172-31-26-108 terraform]# vim main.tf
[root@ip-172-31-26-108 terraform]# cat main.tf
provider "aws" {
region = "ap-southeast-2"
}

resource "aws_vpc" "vpc-1" {
cidr_block = "10.0.0.0/22"

tags = {
Name = "my-vpc-1"
}
}
[root@ip-172-31-26-108 terraform]# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.64.0...
- Installed hashicorp/aws v5.64.0 (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.
[root@ip-172-31-26-108 terraform]# terraform validate
Success! The configuration is valid.

[root@ip-172-31-26-108 terraform]# 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_vpc.vpc-1 will be created
  + resource "aws_vpc" "vpc-1" {
      + arn   = (known after apply)
      + cidr_block   = "10.0.0.0/22"
      + 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                               = {
        + "Name" = "my-vpc-1"
      }
    + tags_all                           = {
        + "Name" = "my-vpc-1"
      }
  }

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.

[root@ip-172-31-26-108 terraform]# 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_vpc.vpc-1 will be created
```

```
 + resource "aws_vpc" "vpc-1" {
    + arn                              = (known after apply)
    + cidr_block                       = "10.0.0.0/22"
    + 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                             = {
      + "Name" = "my-vpc-1"
     }
    + tags_all                         = {
      + "Name" = "my-vpc-1"
     }
  }

Plan: 1 to add, 0 to change, 0 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_vpc.vpc-1: Creating...
aws_vpc.vpc-1: Creation complete after 4s [id=vpc-01d788a16032f72a5]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[root@ip-172-31-26-108 terraform]# vim main.tf
[root@ip-172-31-26-108 terraform]# cat main.tf
provider "aws" {
region = "eu-west-2"
}

resource "aws_vpc" "vpc-1" {
cidr_block = "10.0.0.0/20"

tags = {
Name = "my-vpc-1"
}
}

resource "aws_subnet" "sub-1" {
  vpc_id     = aws_vpc.vpc-1.id
  cidr_block = "10.0.0.0/24"

  tags = {
    Name = "subnet-1"
  }
}
[root@ip-172-31-26-108 terraform]# terraform validate
Success! The configuration is valid.
```

[root@ip-172-31-26-108 terraform]# terraform apply

aws_vpc.vpc-1: Refreshing state... [id=vpc-01d788a16032f72a5]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected

this plan:

```
 # aws_vpc.vpc-1 has been deleted
 - resource "aws_vpc" "vpc-1" {
     - id                               = "vpc-01d788a16032f72a5" -> null
       tags                             = {
           "Name" = "my-vpc-1"
       }
       # (19 unchanged attributes hidden)
   }
```

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the

following plan may include actions to undo or respond to these changes.

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_subnet.sub-1 will be created
+ resource "aws_subnet" "sub-1" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = (known after apply)
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.0.0/24"
    + 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" = "subnet-1"
      }
    + tags_all                                       = {
        + "Name" = "subnet-1"
      }
    + vpc_id                                         = (known after apply)
  }

# aws_vpc.vpc-1 will be created
+ resource "aws_vpc" "vpc-1" {
    + arn                              = (known after apply)
    + cidr_block                       = "10.0.0.0/20"
    + 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                               = {
        + "Name" = "my-vpc-1"
      }
    + tags_all                           = {
        + "Name" = "my-vpc-1"
      }
  }

Plan: 2 to add, 0 to change, 0 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_vpc.vpc-1: Creating...
aws_vpc.vpc-1: Creation complete after 1s [id=vpc-0ac8a7fc2dfff521d]
```

aws_subnet.sub-1: Creating...

aws_subnet.sub-1: Creation complete after 0s [id=subnet-0114b5f31f2d72c62]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

[root@ip-172-31-26-108 terraform]# terraform destroy

aws_vpc.vpc-1: Refreshing state... [id=vpc-0ac8a7fc2dfff521d]

aws_subnet.sub-1: Refreshing state... [id=subnet-0114b5f31f2d72c62]

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_subnet.sub-1 will be destroyed

- resource "aws_subnet" "sub-1" {

- arn = "arn:aws:ec2:eu-west-2:381492190780:subnet/subnet-0114b5f31f2d72c62" -> null

- assign_ipv6_address_on_creation = false -> null

- availability_zone = "eu-west-2b" -> null

- availability_zone_id = "euw2-az3" -> null

- cidr_block = "10.0.0.0/24" -> null

- enable_dns64 = false -> null

- enable_lni_at_device_index = 0 -> null

- enable_resource_name_dns_a_record_on_launch = false -> null

- enable_resource_name_dns_aaaa_record_on_launch = false -> null

- id = "subnet-0114b5f31f2d72c62" -> null

- ipv6_native = false -> null

- map_customer_owned_ip_on_launch = false -> null

- map_public_ip_on_launch = false -> null

- owner_id = "381492190780" -> null

```
    - private_dns_hostname_type_on_launch          = "ip-name" -> null
    - tags                                         = {
        - "Name" = "subnet-1"
      } -> null
    - tags_all                                     = {
        - "Name" = "subnet-1"
      } -> null
    - vpc_id                                       = "vpc-0ac8a7fc2dfff521d" -> null
      # (4 unchanged attributes hidden)
  }

 # aws_vpc.vpc-1 will be destroyed
 - resource "aws_vpc" "vpc-1" {
    - arn                            = "arn:aws:ec2:eu-west-2:381492190780:vpc/vpc-0ac8a7fc2dfff521d" ->
null
    - assign_generated_ipv6_cidr_block     = false -> null
    - cidr_block                       = "10.0.0.0/20" -> null
    - default_network_acl_id             = "acl-06a96e7e7d3e34300" -> null
    - default_route_table_id             = "rtb-059c5f986f02fe7ec" -> null
    - default_security_group_id          = "sg-0c967e62963c298b3" -> null
    - dhcp_options_id                    = "dopt-086bc95bf596e27a2" -> null
    - enable_dns_hostnames               = false -> null
    - enable_dns_support                 = true -> null
    - enable_network_address_usage_metrics = false -> null
    - id                               = "vpc-0ac8a7fc2dfff521d" -> null
    - instance_tenancy                   = "default" -> null
    - ipv6_netmask_length                = 0 -> null
    - main_route_table_id                = "rtb-059c5f986f02fe7ec" -> null
    - owner_id                         = "381492190780" -> null
    - tags                             = {
        - "Name" = "my-vpc-1"
```

```
      } -> null
    - tags_all                        = {
        - "Name" = "my-vpc-1"
      } -> null
      # (4 unchanged attributes hidden)
  }

Plan: 0 to add, 0 to change, 2 to destroy.

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_subnet.sub-1: Destroying... [id=subnet-0114b5f31f2d72c62]
aws_subnet.sub-1: Destruction complete after 0s
aws_vpc.vpc-1: Destroying... [id=vpc-0ac8a7fc2dfff521d]
aws_vpc.vpc-1: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
[root@ip-172-31-26-108 terraform]# terraform show
The state file is empty. No resources are represented.
[root@ip-172-31-26-108 terraform]# terraform version
Terraform v1.9.5
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v5.64.0
[root@ip-172-31-26-108 terraform]#
```