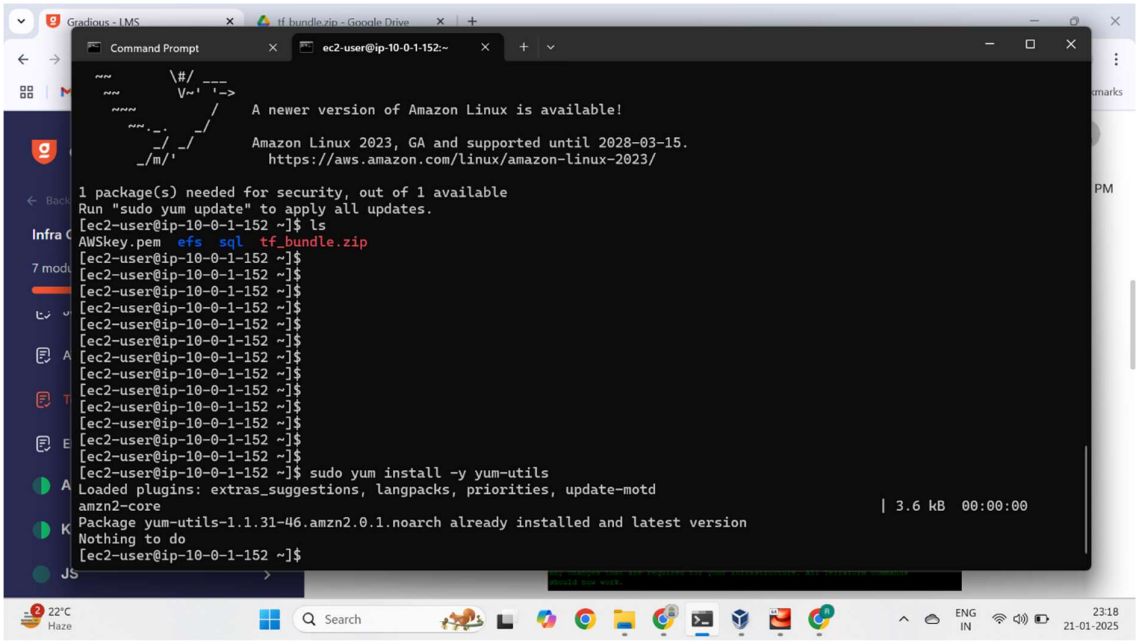
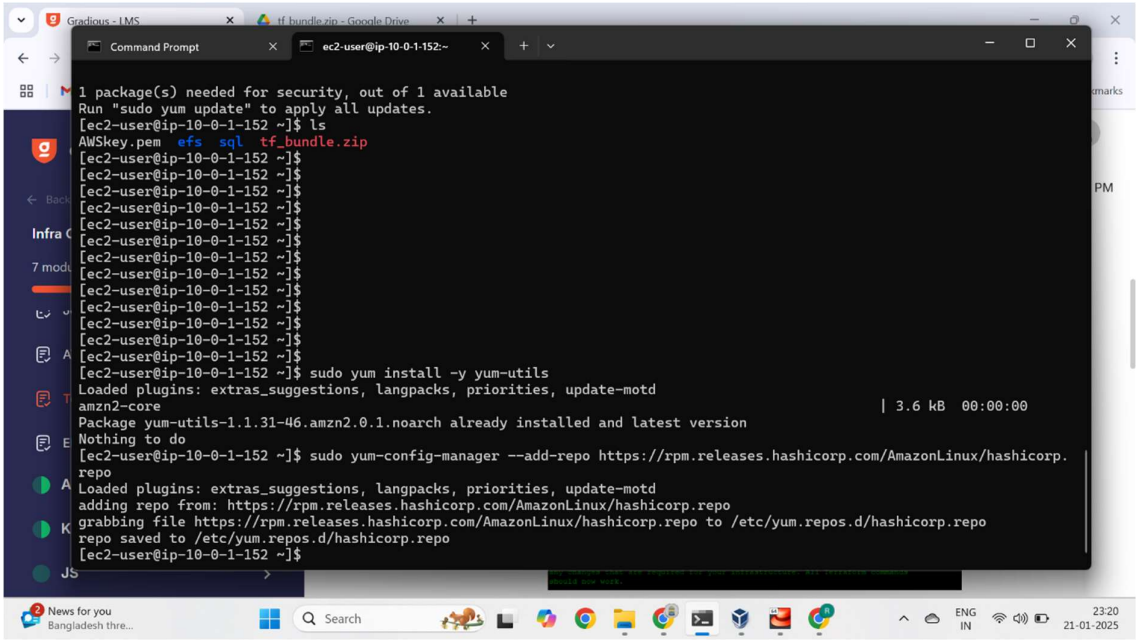
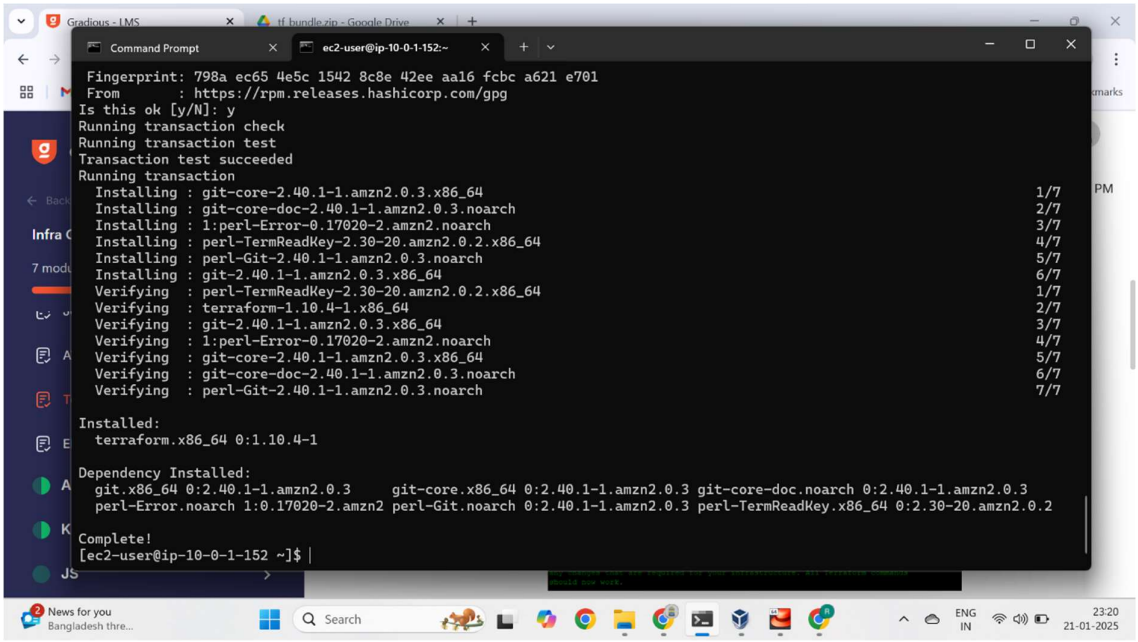
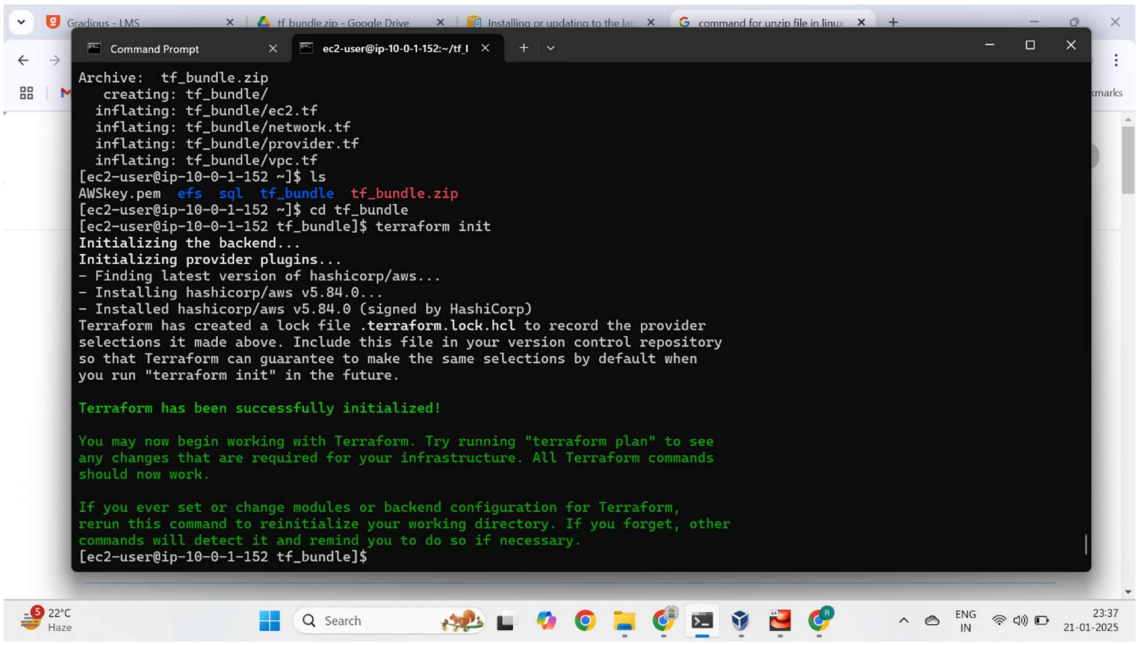
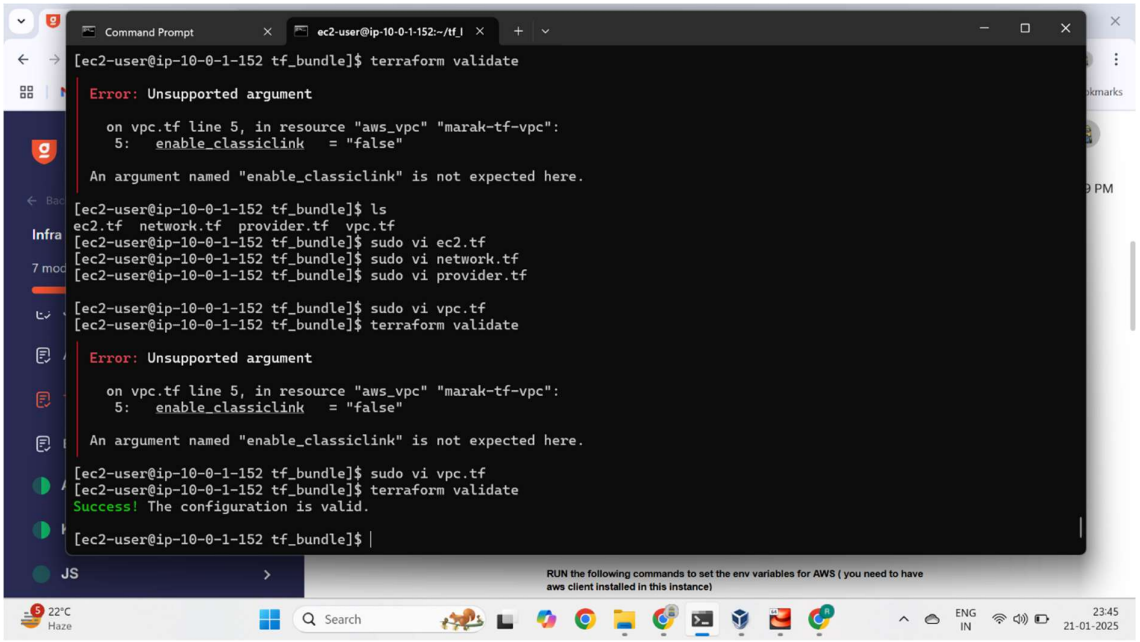
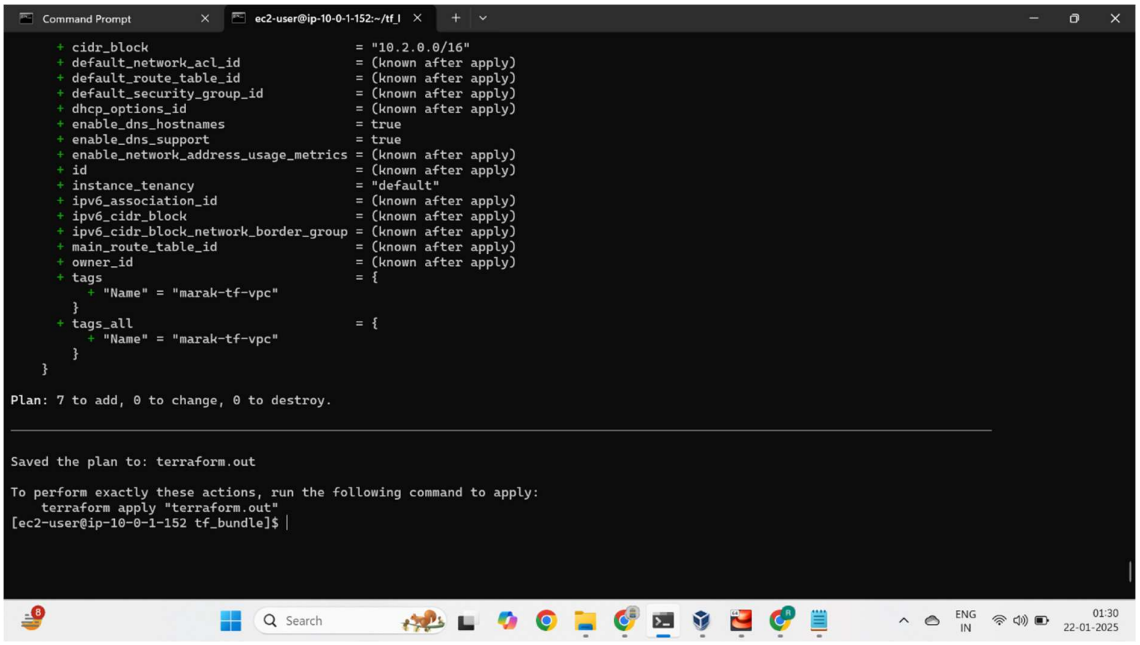

Command Prompt
ec2-user@ip-10-0-1-152:~
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-152 ~]$ ls
AWSkey.pem efs sql tf_bundle.zip
[ec2-user@ip-10-0-1-152 ~]$ sudo yum install -y yum-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
Package yum-utils-1.1.31-46.amzn2.0.1.noarch already installed and latest version
Nothing to do
[ec2-user@ip-10-0-1-152 ~]$

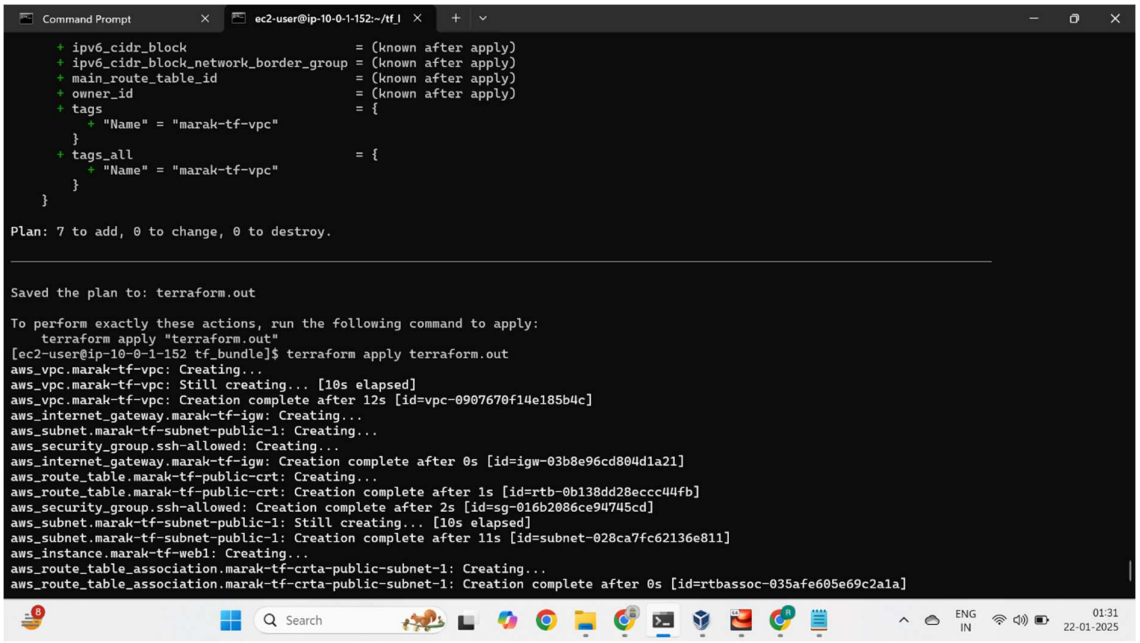

Command Prompt
ec2-user@ip-10-0-1-152:~
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-152 ~]$ ls
AWSkey.pem efs sql tf_bundle.zip
[ec2-user@ip-10-0-1-152 ~]$ sudo yum install -y yum-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
Package yum-utils-1.1.31-46.amzn2.0.1.noarch already installed and latest version
Nothing to do
[ec2-user@ip-10-0-1-152 ~]$ sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
grabbing file https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo to /etc/yum.repos.d/hashicorp.repo
repo saved to /etc/yum.repos.d/hashicorp.repo
[ec2-user@ip-10-0-1-152 ~]$

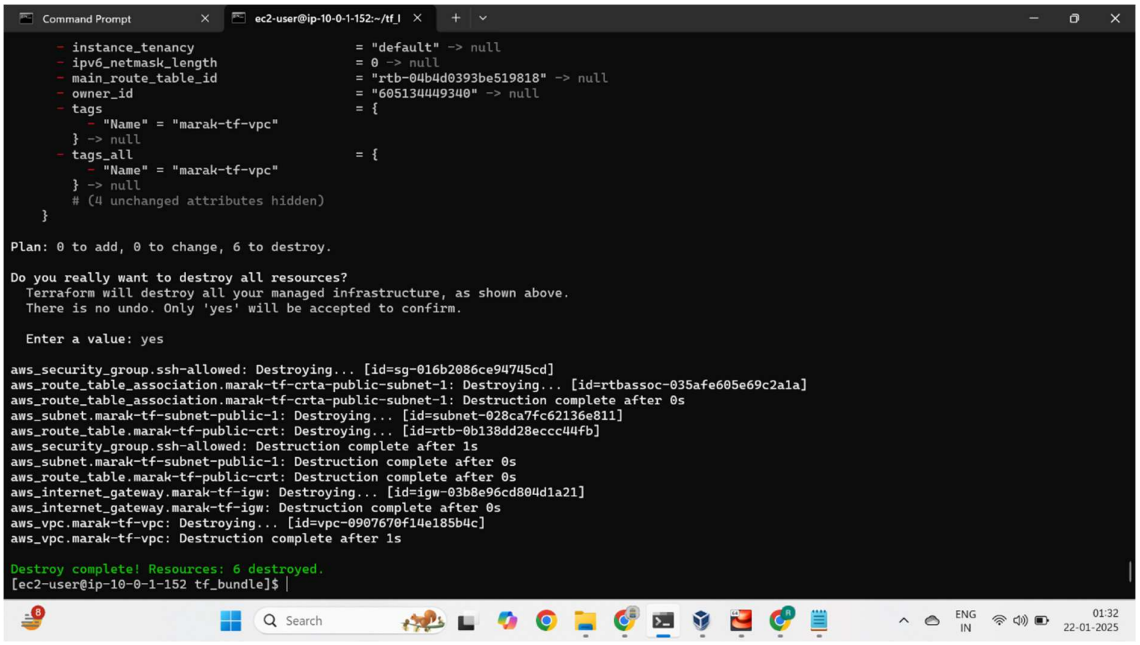

Command Prompt
ec2-user@ip-10-0-1-152:~
Fingerprint: 798a ec65 4e5c 1542 8c8e 42ee aa16 fcbc a621 e701
From : https://rpm.releases.hashicorp.com/gpg
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : git-core-2.40.1-1.amzn2.0.3.x86_64 1/7
Installing : git-core-doc-2.40.1-1.amzn2.0.3.noarch 2/7
Installing : 1:perl-Error-0.17020-2.amzn2.noarch 3/7
Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 4/7
Installing : perl-Git-2.40.1-1.amzn2.0.3.noarch 5/7
Installing : git-2.40.1-1.amzn2.0.3.x86_64 6/7
Verifying : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 1/7
Verifying : terraform-1.10.4-1.x86_64 2/7
Verifying : git-2.40.1-1.amzn2.0.3.x86_64 3/7
Verifying : 1:perl-Error-0.17020-2.amzn2.noarch 4/7
Verifying : git-core-2.40.1-1.amzn2.0.3.x86_64 5/7
Verifying : git-core-doc-2.40.1-1.amzn2.0.3.noarch 6/7
Verifying : perl-Git-2.40.1-1.amzn2.0.3.noarch 7/7
Installed:
terraform.x86_64 0:1.10.4-1
Dependency Installed:
git.x86_64 0:2.40.1-1.amzn2.0.3 git-core.x86_64 0:2.40.1-1.amzn2.0.3 git-core-doc.noarch 0:2.40.1-1.amzn2.0.3
perl-Error.noarch 1:0.17020-2.amzn2 perl-Git.noarch 0:2.40.1-1.amzn2.0.3 perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2
Complete!
[ec2-user@ip-10-0-1-152 ~]$
News for you
Search
ENG
IN
23:20
21-01-2025


Command Prompt
ec2-user@ip-10-0-1-152:~/tf_l
Archive: tf_bundle.zip
creating: tf_bundle/
inflating: tf_bundle/ec2.tf
inflating: tf_bundle/network.tf
inflating: tf_bundle/provider.tf
inflating: tf_bundle/vpc.tf
[ec2-user@ip-10-0-1-152 ~]$ ls
AWSkey.pem efs sql tf_bundle tf_bundle.zip
[ec2-user@ip-10-0-1-152 ~]$ cd tf_bundle
[ec2-user@ip-10-0-1-152 tf_bundle]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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.
[ec2-user@ip-10-0-1-152 tf_bundle]$
22°C
Haze
Search
ENG
IN
23:37
21-01-2025


Command Prompt
ec2-user@ip-10-0-1-152:~/tf_l
[ec2-user@ip-10-0-1-152 tf_bundle]$ terraform validate
Error: Unsupported argument
on vpc.tf line 5, in resource "aws_vpc" "marak-tf-vpc":
5: enable_classiclink = "false"
An argument named "enable_classiclink" is not expected here.
[ec2-user@ip-10-0-1-152 tf_bundle]$ ls
ec2.tf network.tf provider.tf vpc.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ sudo vi ec2.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ sudo vi network.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ sudo vi provider.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ sudo vi vpc.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ terraform validate
Error: Unsupported argument
on vpc.tf line 5, in resource "aws_vpc" "marak-tf-vpc":
5: enable_classiclink = "false"
An argument named "enable_classiclink" is not expected here.
[ec2-user@ip-10-0-1-152 tf_bundle]$ sudo vi vpc.tf
[ec2-user@ip-10-0-1-152 tf_bundle]$ terraform validate
Success! The configuration is valid.
[ec2-user@ip-10-0-1-152 tf_bundle]$
RUN the following commands to set the env variables for AWS (you need to have aws client installed in this instance)
Search
23:45
21-01-2025


Command Prompt
ec2-user@ip-10-0-1-152:~/tf_l
+ cidr_block = "10.2.0.0/16"
+ 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 = true
+ 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" = "marak-tf-vpc"
}
+ tags_all = {
+ "Name" = "marak-tf-vpc"
}
}
Plan: 7 to add, 0 to change, 0 to destroy.
Saved the plan to: terraform.out
To perform exactly these actions, run the following command to apply:
terraform apply "terraform.out"
[ec2-user@ip-10-0-1-152 tf_bundle]$
Search
01:30
22-01-2025


Command Prompt
ec2-user@ip-10-0-1-152:~/tf_l
+ 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" = "marak-tf-vpc"
}
+ tags_all = {
+ "Name" = "marak-tf-vpc"
}
}
Plan: 7 to add, 0 to change, 0 to destroy.
Saved the plan to: terraform.out
To perform exactly these actions, run the following command to apply:
terraform apply "terraform.out"
[ec2-user@ip-10-0-1-152 tf_bundle]$ terraform apply terraform.out
aws_vpc.marak-tf-vpc: Creating...
aws_vpc.marak-tf-vpc: Still creating... [10s elapsed]
aws_vpc.marak-tf-vpc: Creation complete after 12s [id=vpc-0907670f14e185b4c]
aws_internet_gateway.marak-tf-igw: Creating...
aws_subnet.marak-tf-subnet-public-1: Creating...
aws_security_group.ssh-allowed: Creating...
aws_internet_gateway.marak-tf-igw: Creation complete after 0s [id=igw-03b8e96cd804d1a21]
aws_route_table.marak-tf-public-crt: Creating...
aws_route_table.marak-tf-public-crt: Creation complete after 1s [id=rtb-0b138dd28eccc44fb]
aws_security_group.ssh-allowed: Creation complete after 2s [id=sg-016b2086ce94745cd]
aws_subnet.marak-tf-subnet-public-1: Still creating... [10s elapsed]
aws_subnet.marak-tf-subnet-public-1: Creation complete after 11s [id=subnet-028ca7fc62136e811]
aws_instance.marak-tf-web1: Creating...
aws_route_table_association.marak-tf-crta-public-subnet-1: Creating...
aws_route_table_association.marak-tf-crta-public-subnet-1: Creation complete after 0s [id=rtbassoc-035afe605e69c2a1a]
Search
ENG
IN
01:31
22-01-2025


Command Prompt
ec2-user@ip-10-0-1-152:~/tf_l
- instance_tenancy = "default" -> null
- ipv6_netmask_length = 0 -> null
- main_route_table_id = "rtb-04b4d0393be519818" -> null
- owner_id = "605134449340" -> null
- tags = {
- "Name" = "marak-tf-vpc"
} -> null
- tags_all = {
- "Name" = "marak-tf-vpc"
} -> null
# (4 unchanged attributes hidden)
}
Plan: 0 to add, 0 to change, 6 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_security_group.ssh-allowed: Destroying... [id=sg-016b2086ce94745cd]
aws_route_table_association.marak-tf-crta-public-subnet-1: Destroying... [id=rtbassoc-035afe605e69c2a1a]
aws_route_table_association.marak-tf-crta-public-subnet-1: Destruction complete after 0s
aws_subnet.marak-tf-subnet-public-1: Destroying... [id=subnet-028ca7fc62136e811]
aws_route_table.marak-tf-public-crt: Destroying... [id=rtb-0b138dd28eccc44fb]
aws_security_group.ssh-allowed: Destruction complete after 1s
aws_subnet.marak-tf-subnet-public-1: Destruction complete after 0s
aws_route_table.marak-tf-public-crt: Destruction complete after 0s
aws_internet_gateway.marak-tf-igw: Destroying... [id=igw-03b8e96cd804d1a21]
aws_internet_gateway.marak-tf-igw: Destruction complete after 0s
aws_vpc.marak-tf-vpc: Destroying... [id=vpc-0907670f14e185b4c]
aws_vpc.marak-tf-vpc: Destruction complete after 1s
Destroy complete! Resources: 6 destroyed.
[ec2-user@ip-10-0-1-152 tf_bundle]$
Search
ENG
IN
01:32
22-01-2025