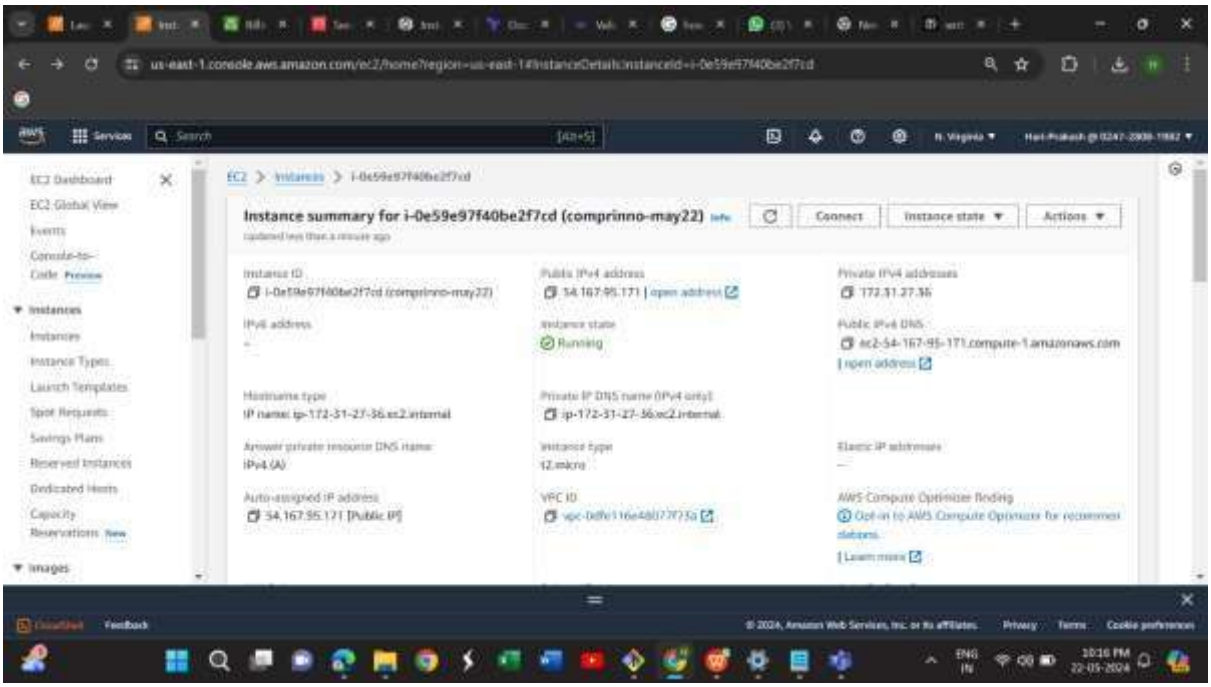
TERRAFORM TASK

WRITE A TERRAFORM SCRIPT TO LAUNCH A VPC ,4 SUBNETS ,2 PUBLIC SUBNET AND 2 PRIVATE SUBNET,1 EC2 IN A PUBLIC AND A RDS IN PRIVATE

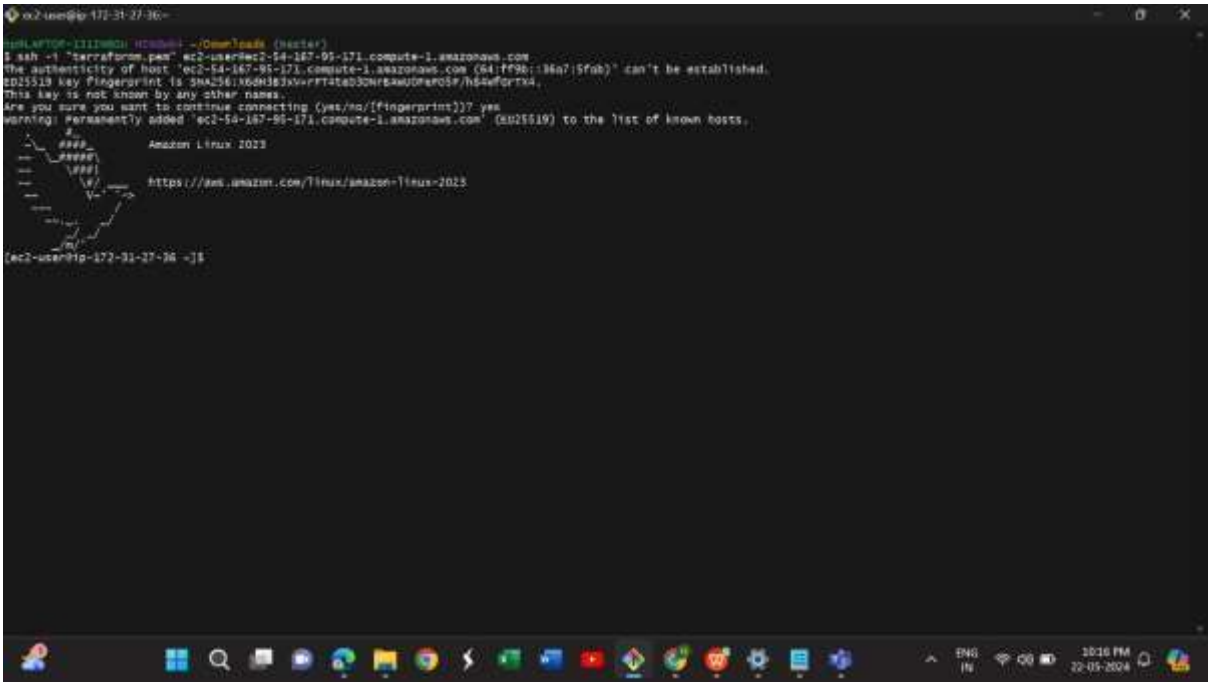
PREPARED BY

HARI PRAKASH.R

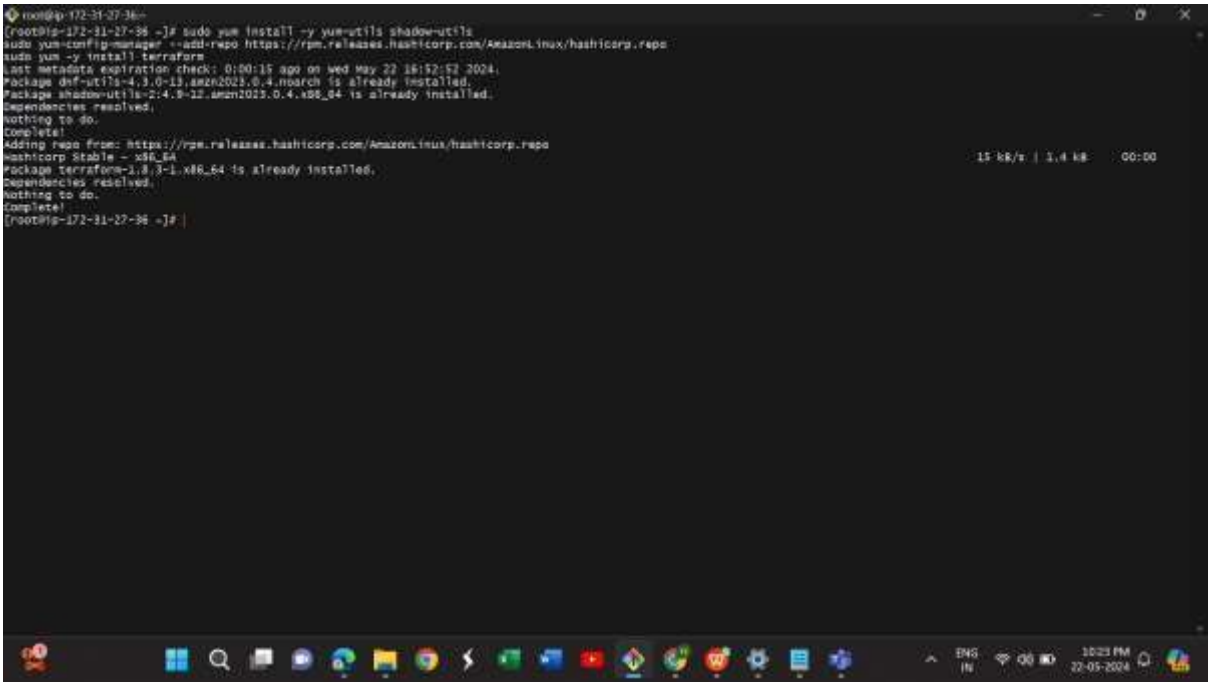


STEP 1:

In the AWS management Console and create the another EC2 instance.



STEP 2: Connect the EC2 instance with that ssh

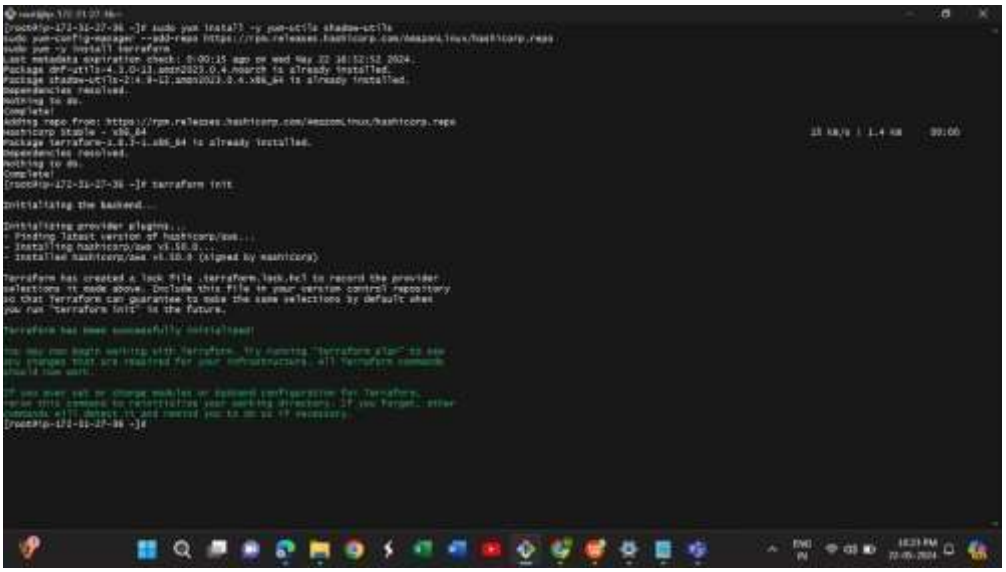


STEP 3: Install the Terraform with the below code

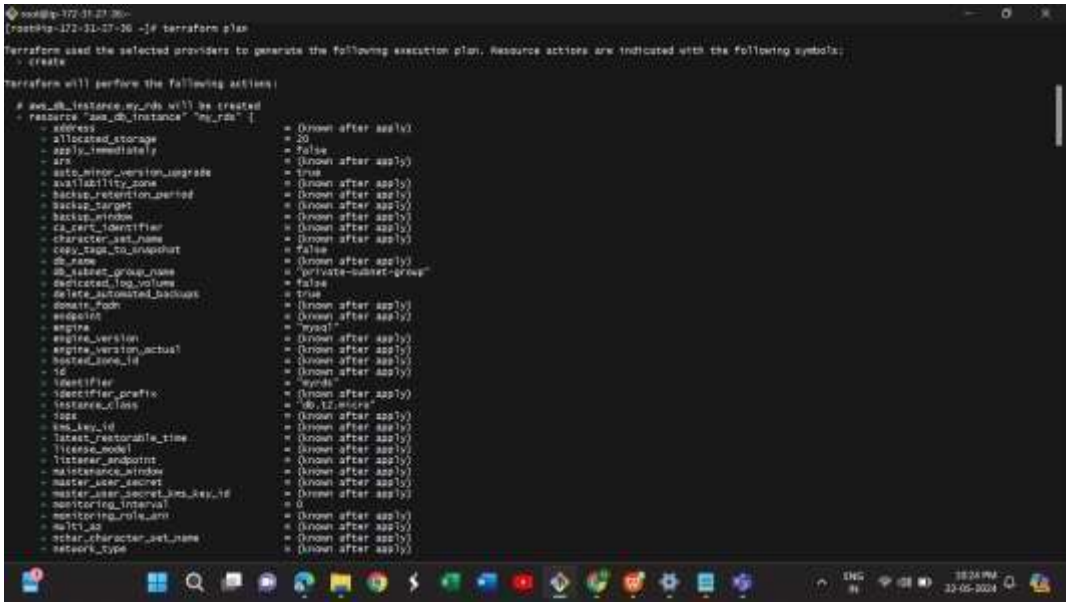
sudo yum install -y yum-utils shadow-utils

sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

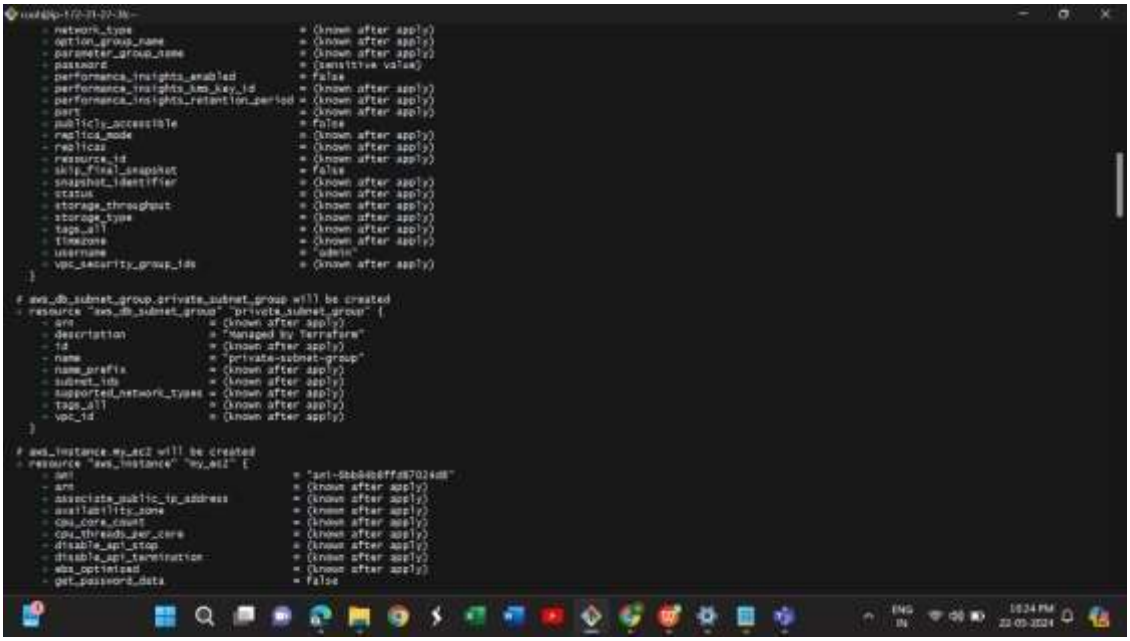
sudo yum -y install terraform

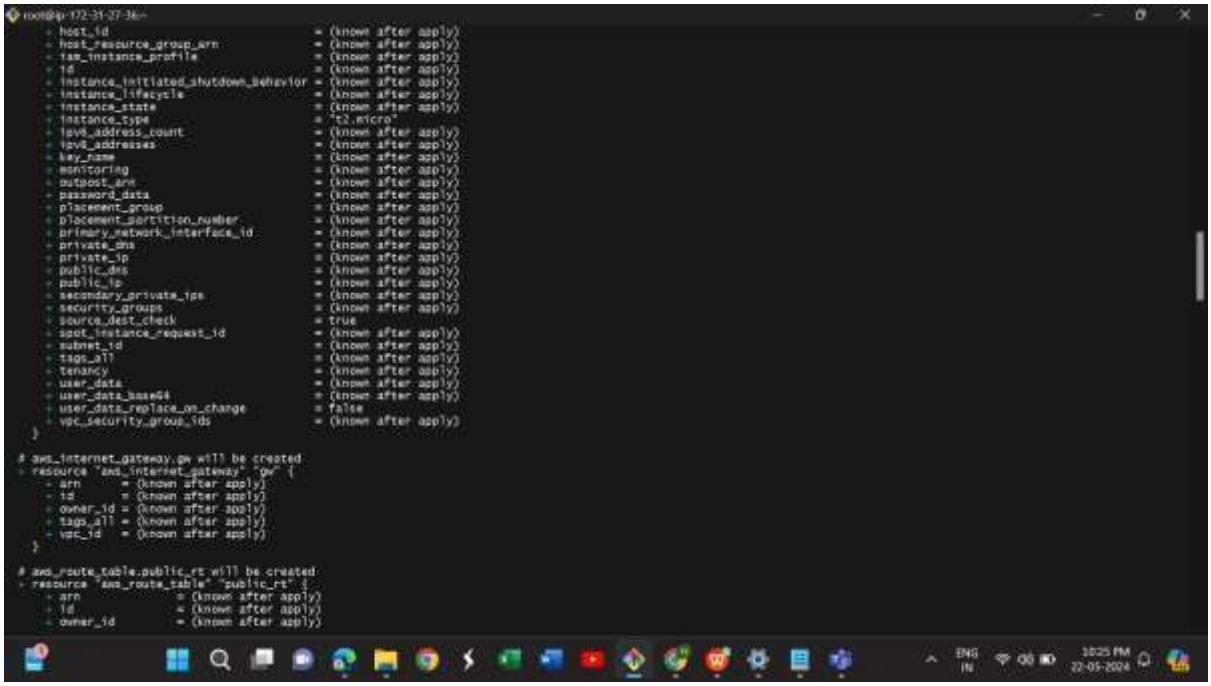


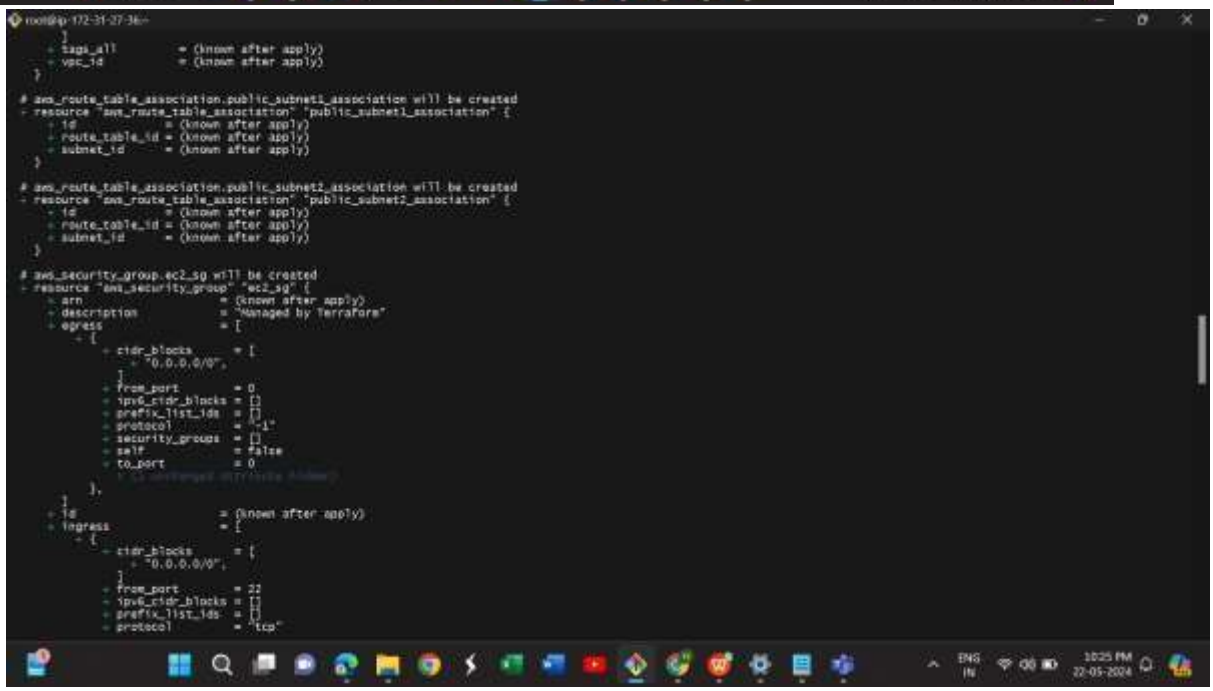
STEP 4 : I have attach the code in the main.tf

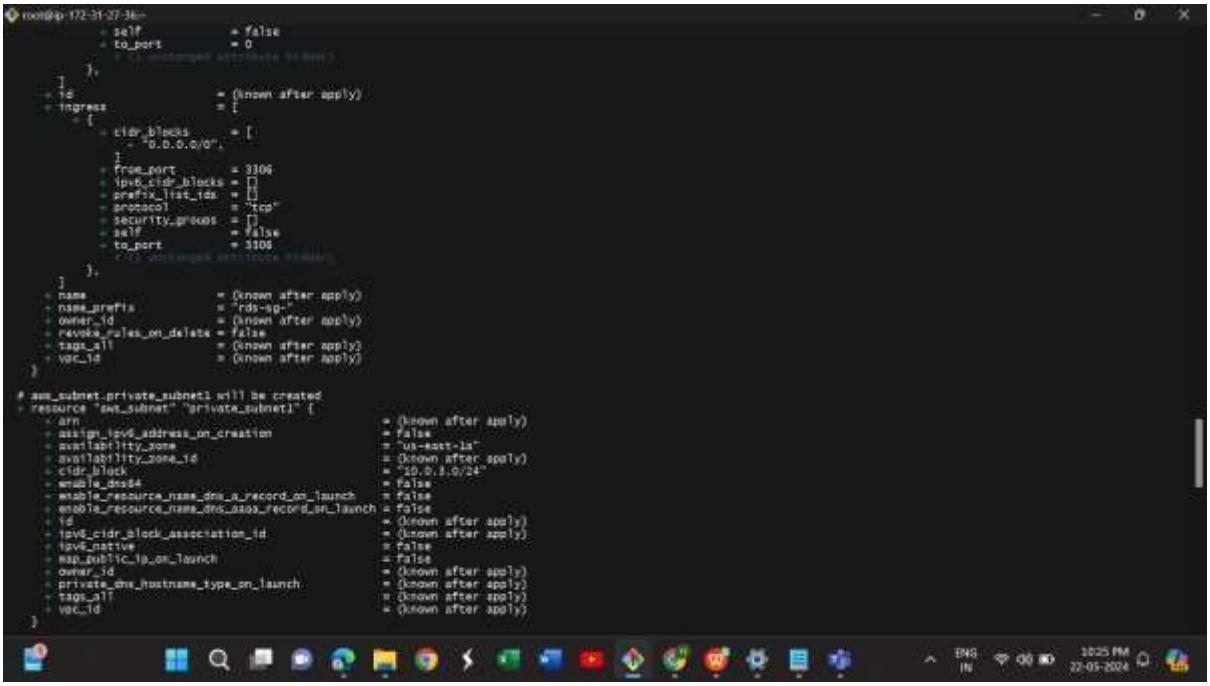


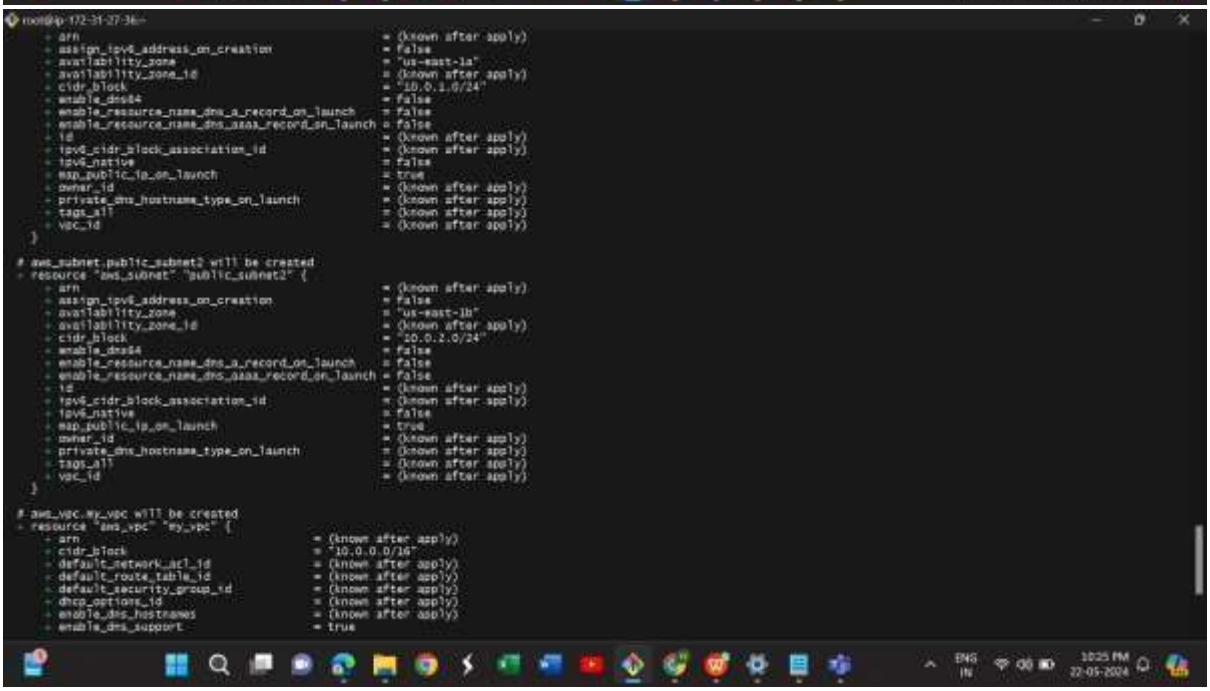
STEP 5: After uploading the terraform code in the main.tf ,Later that You want to execute with the Terraform Plan

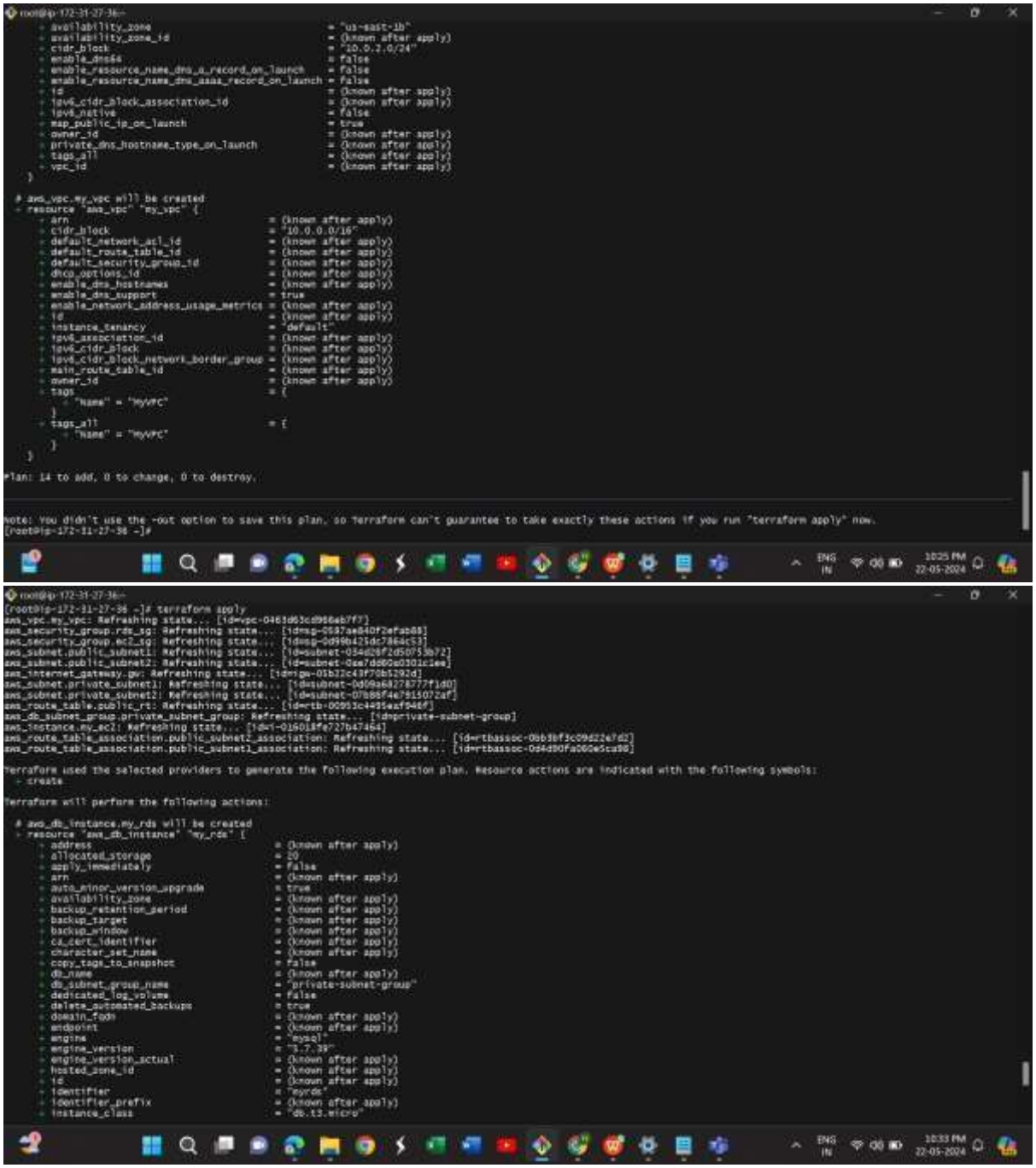

network_type = (known after apply)
option_group_name = (known after apply)
parameter_group_name = (known after apply)
password = (sensitive value)
performance_insights_enabled = false
performance_insights_kms_key_id = (known after apply)
performance_insights_retention_period = (known after apply)
port = (known after apply)
publicly_accessible = false
replica_mode = (known after apply)
replicas = (known after apply)
resource_id = (known after apply)
skip_final_snapshot = false
snapshot_identifier = (known after apply)
status = (known after apply)
storage_throughput = (known after apply)
storage_type = (known after apply)
tags_all = (known after apply)
timezone = (known after apply)
username = "admin"
vpc_security_group_ids = (known after apply)
# aws_db_subnet_group.private_subnet_group will be created
resource "aws_db_subnet_group" "private_subnet_group" {
arn = (known after apply)
description = "Managed by Terraform"
id = (known after apply)
name = "private-subnet-group"
name_prefix = (known after apply)
subnet_ids = (known after apply)
supported_network_types = (known after apply)
tags_all = (known after apply)
vpc_id = (known after apply)
# aws_instance.my_ec2 will be created
resource "aws_instance" "my_ec2" {
ami = "ami-0bb84b8ffd87024d8"
arn = (known after apply)
associate_public_ip_address = (known after apply)
availability_zone = (known after apply)
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


root@ip-172-31-27-36:~
+ 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 = (known after apply)
+ 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)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags_all = (known after apply)
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
# aws_internet_gateway.gw will be created
+ resource "aws_internet_gateway" "gw" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_route_table.public_rt will be created
+ resource "aws_route_table" "public_rt" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
ENG
IN
10:25 PM
22-05-2024

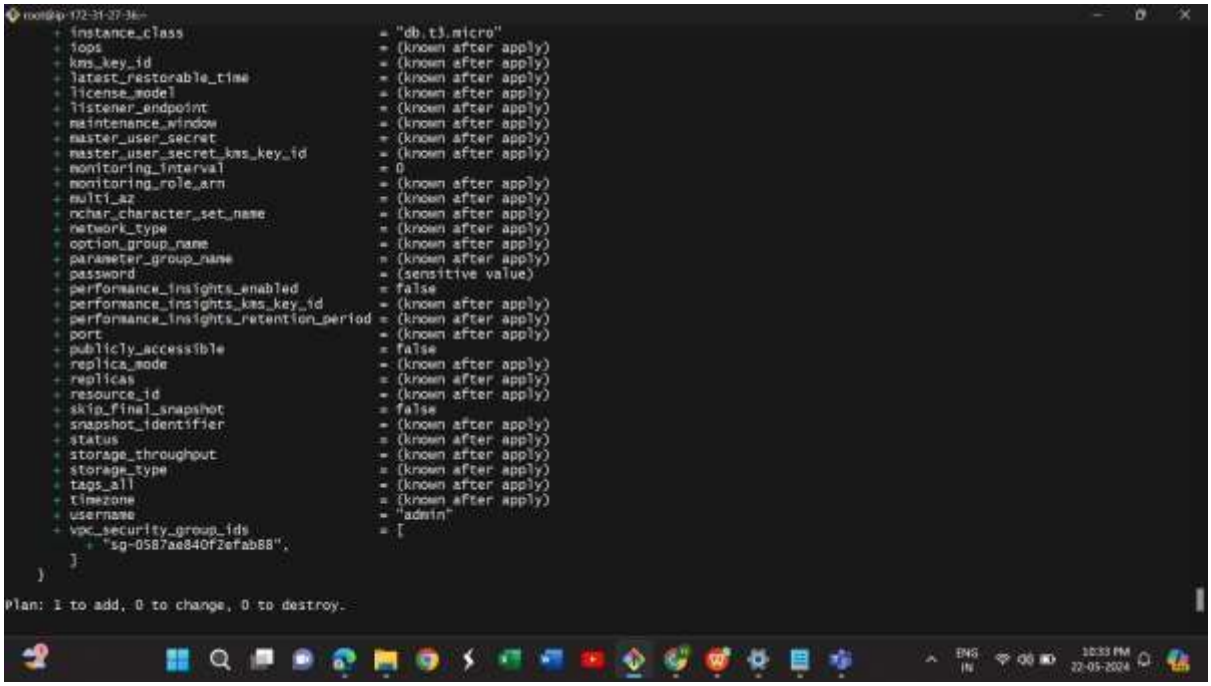

root@ip-172-31-27-36:~
}
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_route_table_association.public_subnet1_association will be created
+ resource "aws_route_table_association" "public_subnet1_association" {
+ id = (known after apply)
+ route_table_id = (known after apply)
+ subnet_id = (known after apply)
}
# aws_route_table_association.public_subnet2_association will be created
+ resource "aws_route_table_association" "public_subnet2_association" {
+ id = (known after apply)
+ route_table_id = (known after apply)
+ subnet_id = (known after apply)
}
# aws_security_group.ec2_sg will be created
+ resource "aws_security_group" "ec2_sg" {
+ arn = (known after apply)
+ description = "Managed by Terraform"
+ egress = [
- {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ from_port = 0
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "-1"
+ security_groups = []
+ self = false
+ to_port = 0
# (1 unchanged attribute hidden)
},
]
+ id = (known after apply)
+ ingress = [
- {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ from_port = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
ENG
IN
10:25 PM
22-05-2024

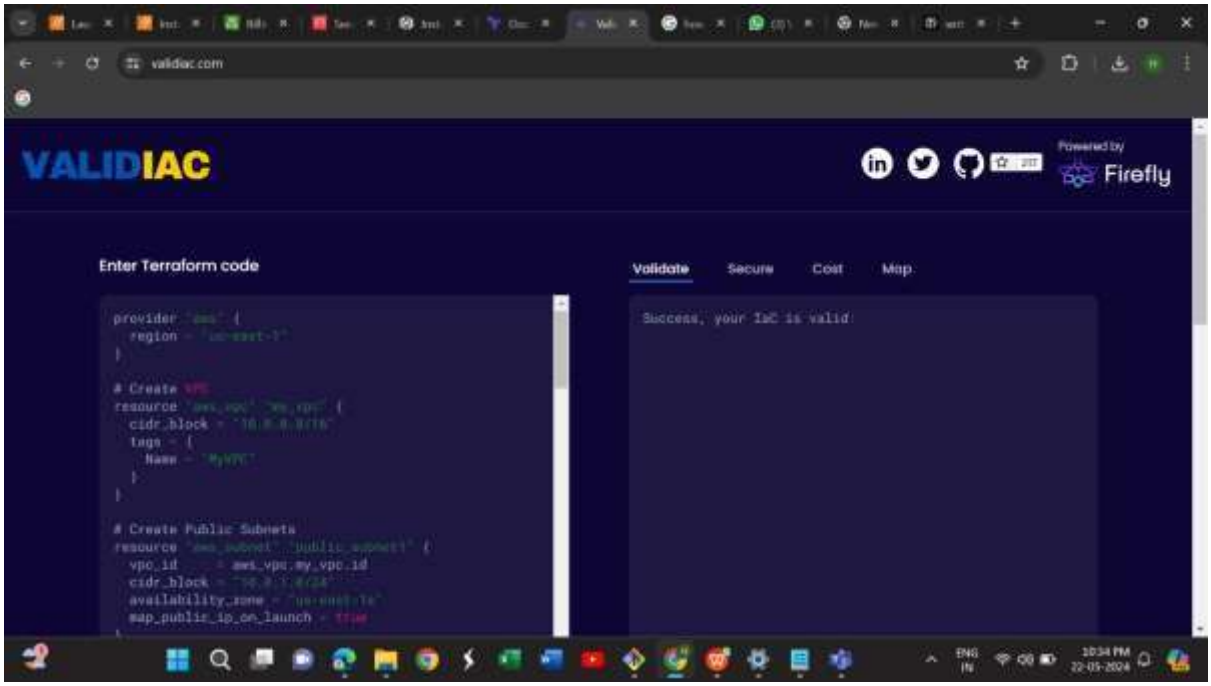

root@ip-172-31-27-36:~
+ self = false
+ to_port = 0
}
]
+ id = (known after apply)
+ ingress = [
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ from_port = 3306
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 3306
}
]
+ name = (known after apply)
+ name_prefix = "rds-sg-"
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_subnet.private_subnet1 will be created
+ resource "aws_subnet" "private_subnet1" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1a"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.3.0/24"
+ 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_all = (known after apply)
+ vpc_id = (known after apply)
}
ENG
IN
10:25 PM
22-05-2024


root@ip-172-31-27-36:~
+ 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 = true
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_subnet.public_subnet2 will be created
+ resource "aws_subnet" "public_subnet2" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1b"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.2.0/24"
+ 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 = true
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_vpc.my_vpc will be created
+ resource "aws_vpc" "my_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
ENG
IN
10:25 PM
22-05-2024



STEP 6: After that write the terraform apply


+ instance_class = "db.t3.micro"
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ latest_restorable_time = (known after apply)
+ license_model = (known after apply)
+ listener_endpoint = (known after apply)
+ maintenance_window = (known after apply)
+ master_user_secret = (known after apply)
+ master_user_secret_kms_key_id = (known after apply)
+ monitoring_interval = 0
+ monitoring_role_arn = (known after apply)
+ multi_az = (known after apply)
+ nchar_character_set_name = (known after apply)
+ network_type = (known after apply)
+ option_group_name = (known after apply)
+ parameter_group_name = (known after apply)
+ password = (sensitive value)
+ performance_insights_enabled = false
+ performance_insights_kms_key_id = (known after apply)
+ performance_insights_retention_period = (known after apply)
+ port = (known after apply)
+ publicly_accessible = false
+ replica_mode = (known after apply)
+ replicas = (known after apply)
+ resource_id = (known after apply)
+ skip_final_snapshot = false
+ snapshot_identifier = (known after apply)
+ status = (known after apply)
+ storage_throughput = (known after apply)
+ storage_type = (known after apply)
+ tags_all = (known after apply)
+ timezone = (known after apply)
+ username = "admin"
+ vpc_security_group_ids = [
+ "sg-0587ae840f2efab88",
]
}
Plan: 1 to add, 0 to change, 0 to destroy.


validiac.com
VALIDIAC
Powered by
Firefly
Enter Terraform code
Validate
Secure
Cost
Map
Success, your IaC is valid

STEP 7 : Later You push the code in the VALIDIAC in the code and you get the output code as a valid