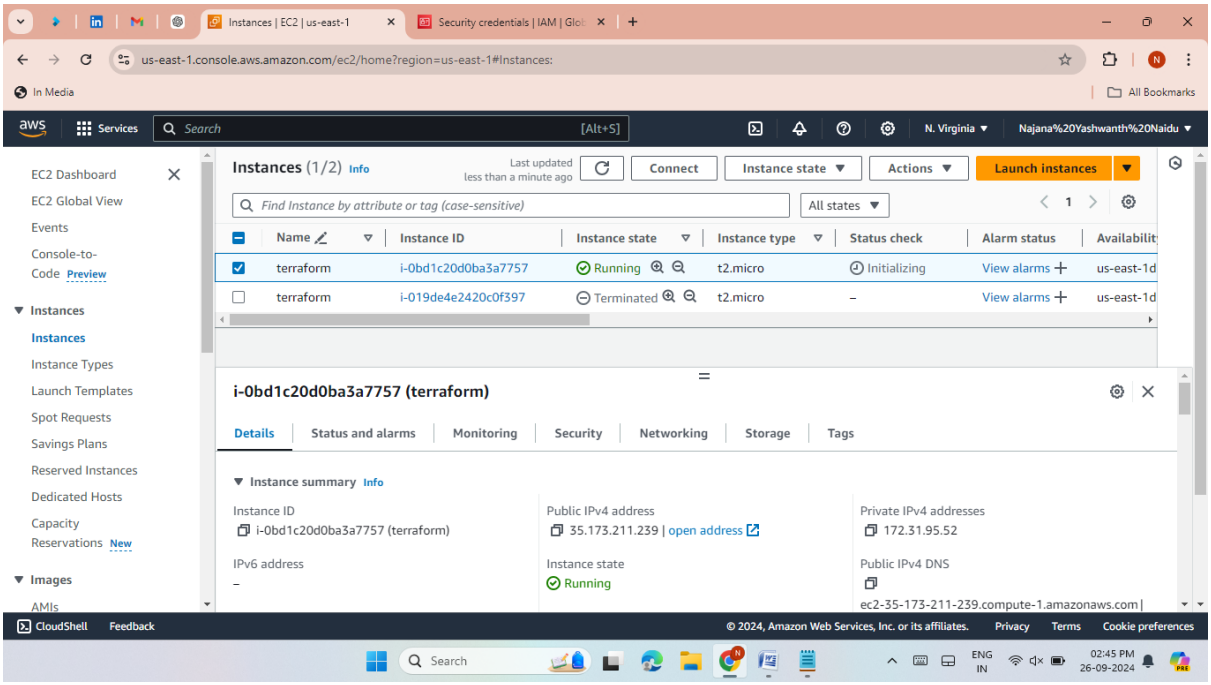
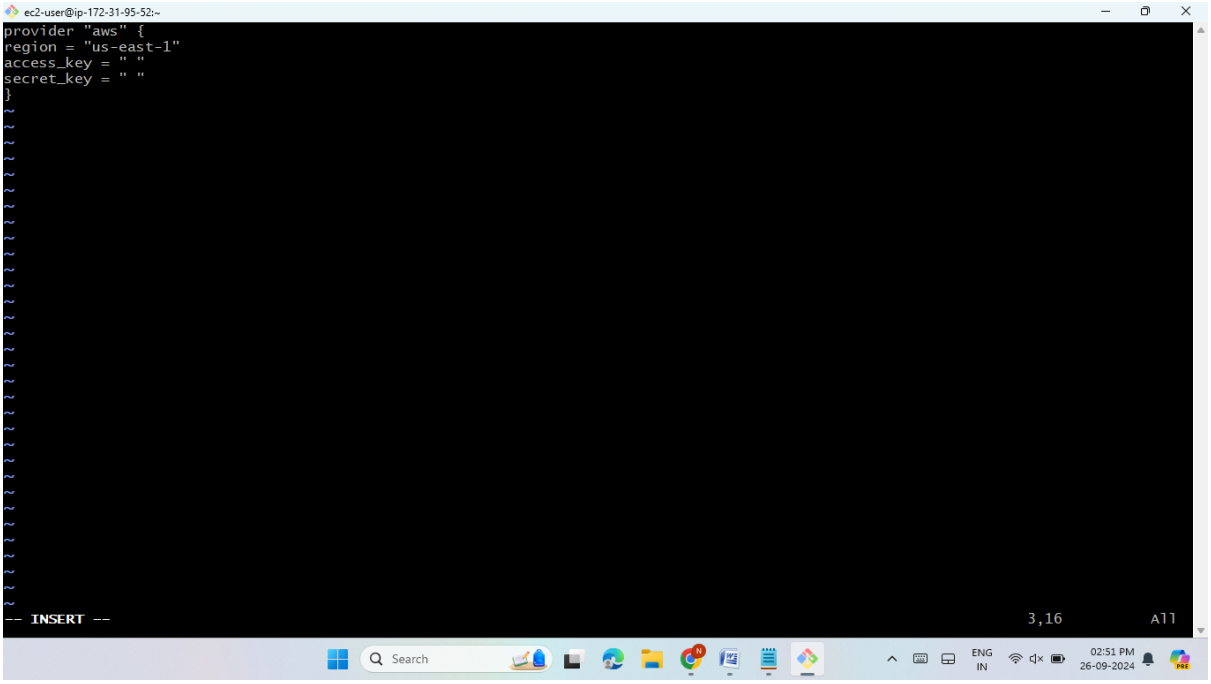
Project

DEPLOY THREE-TIRE ARCHITECTURE IN AWS USING TERRAFORM

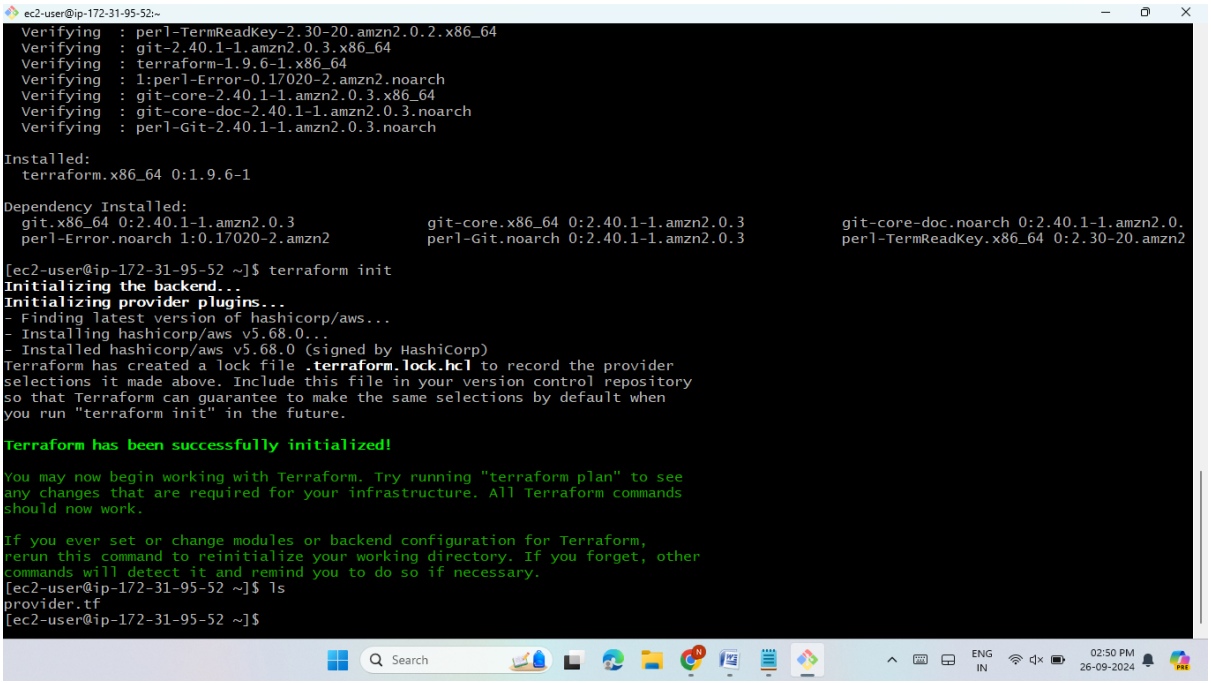
- Terraform is an infrastructure-as-code software tool created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language (HCL).
- Create a EC2 instance then install tarraform.



- Provide access keys and secret key.
- A software element known as a Terraform provider enables Terraform to communicate with a particular infrastructure platform. The resource kinds and data sources that Terraform can handle for that platform must be implemented by providers.Cloud platforms, data centres, network devices, databases, and other resources inside the target infrastructure or service can all be defined, configured, and managed by Terraform providers.

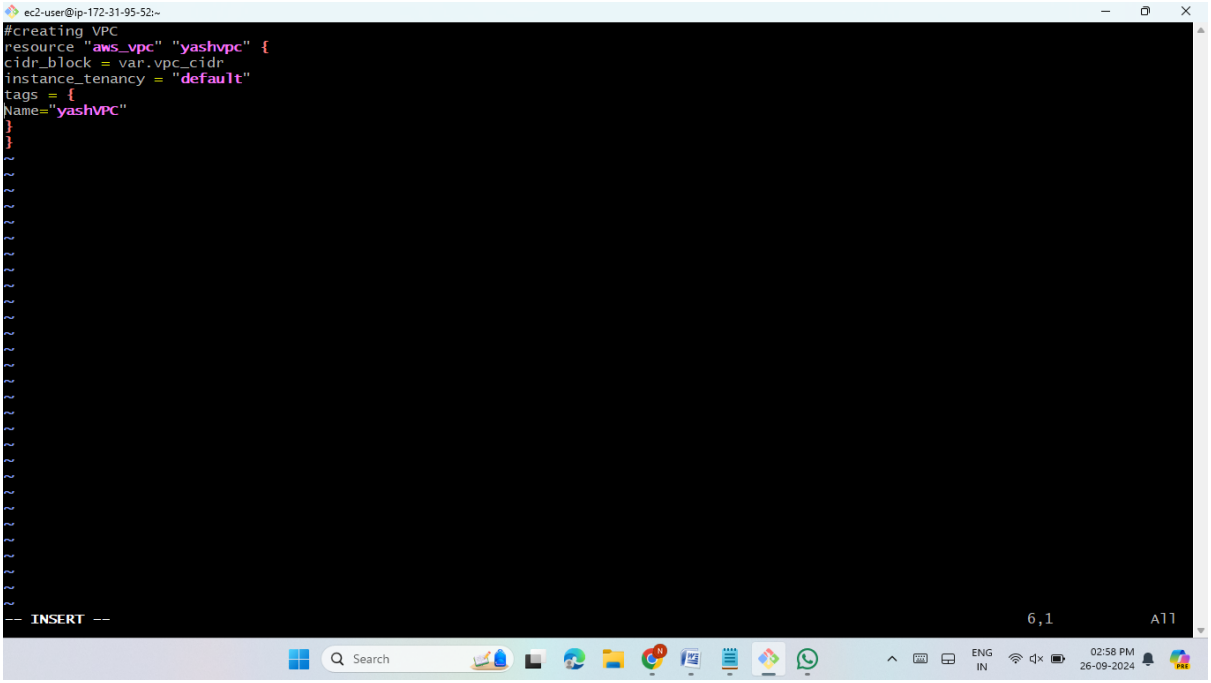


- Enter the teraform init command to initialized the terraform
- Terraform init command initializes a Terraform working directory by downloading and installing any required plugins and dependencies. It should be run before any other Terraform commands.

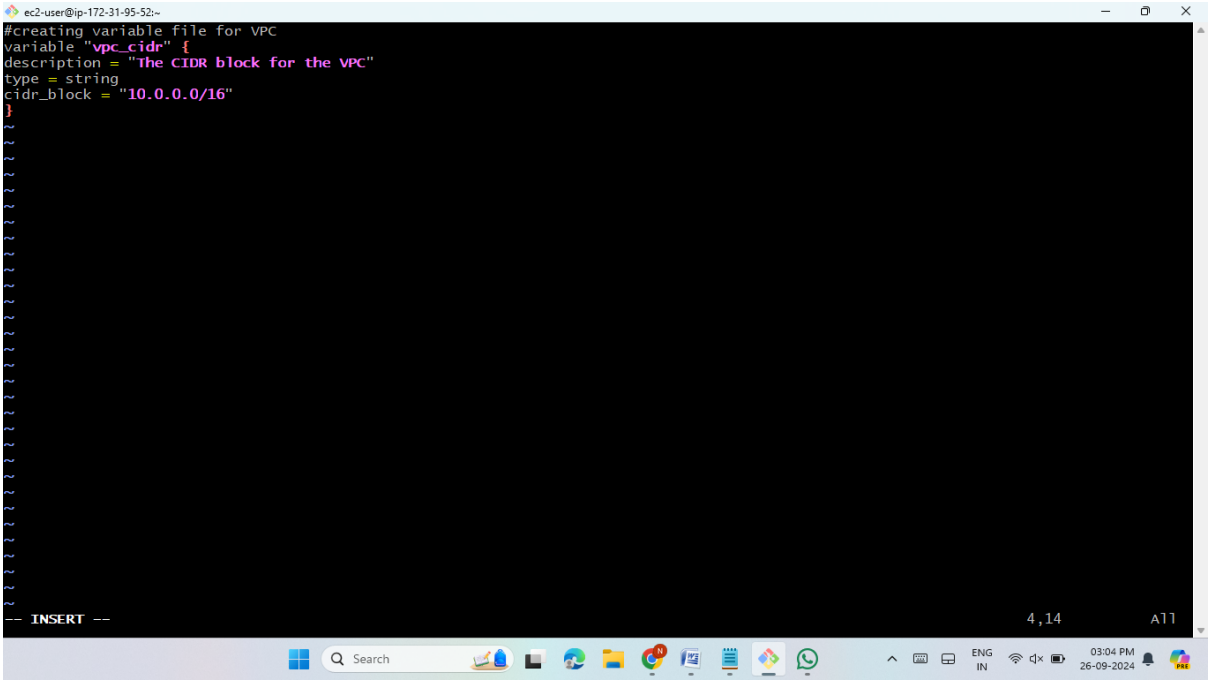


Create a file for the VPC

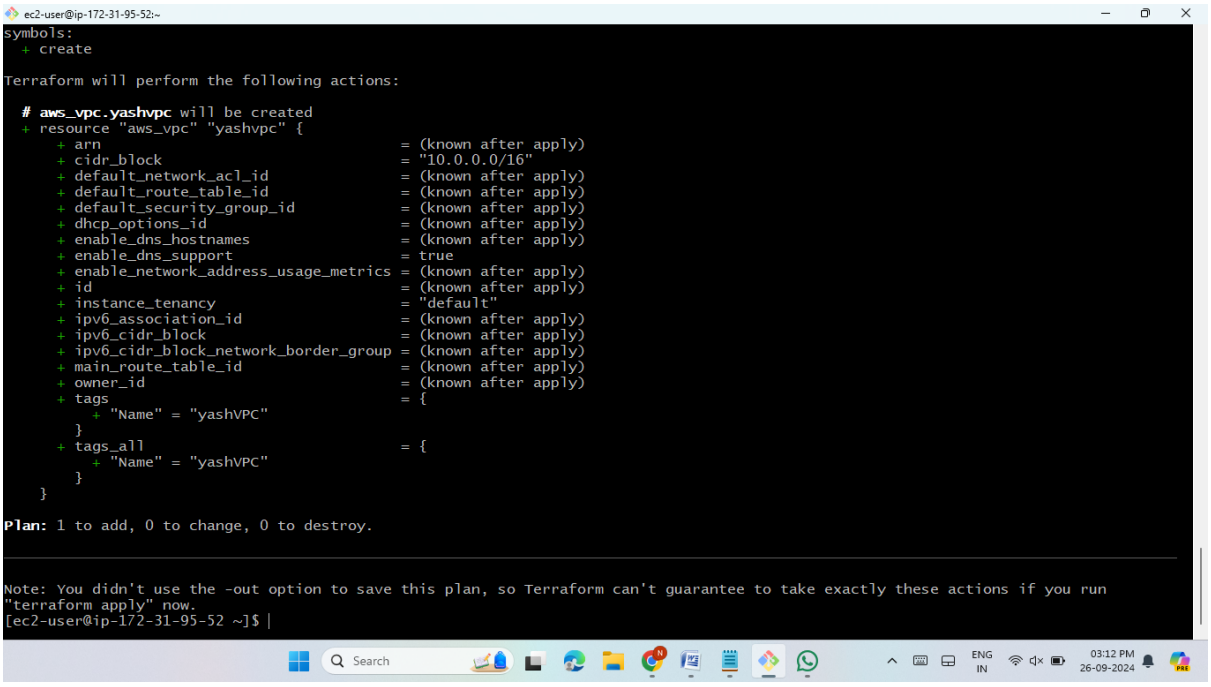
- Here Creating vpc.tf file
- **.tf** file in Terraform is a text file that contains the configuration code for Terraform to provision infrastructure. These files are written in **HashiCorp Configuration Language (HCL)**



- Here the creating variable file for the VPC.
- A **variable file** in Terraform is a file used to define and assign values to variables in your Terraform configuration.

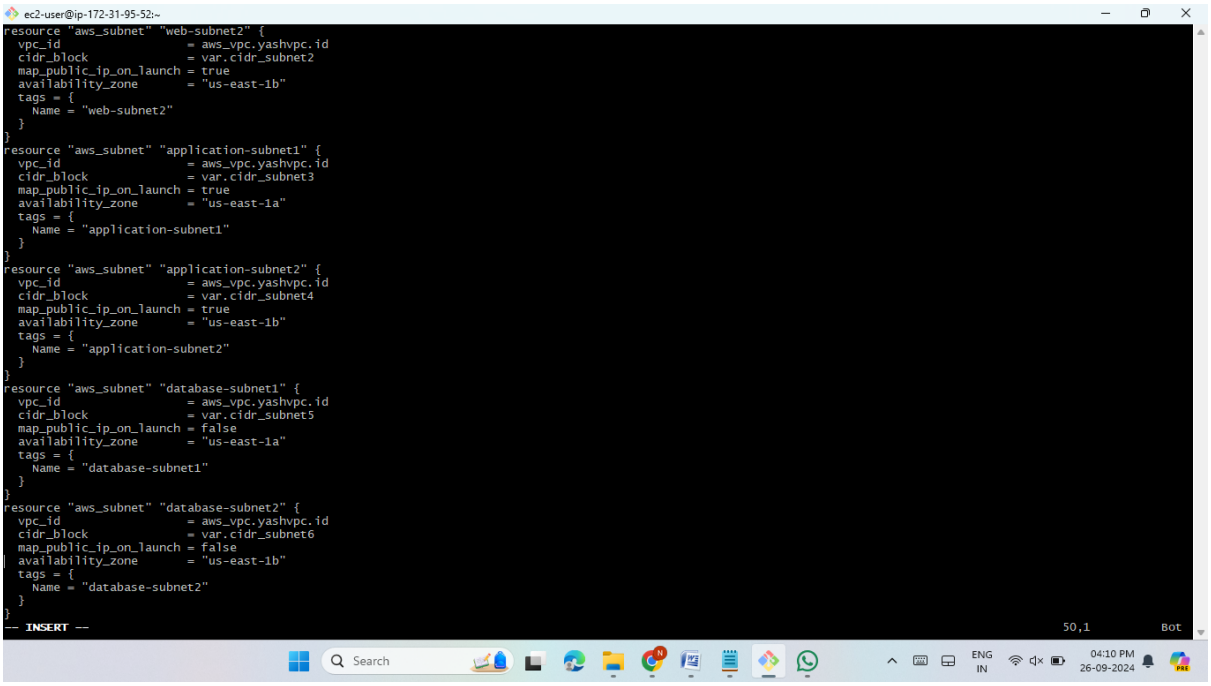


- Type the following command.
- **terraform fmt** (used to automatically format Terraform configuration files)
- **terraform validate** (used to check whether a Terraform configuration is syntactically valid and internally consistent. It verifies that the configuration files are structured correctly and that all resource definitions and configurations comply with Terraform's internal logic, but it doesn't actually interact with infrastructure or execute any changes.)
- **terraform plan** (used to create an execution plan, showing what actions Terraform will take to reach the desired state of your infrastructure based on your configuration files. It does not make any changes to the infrastructure but rather provides a preview of what Terraform will do when you apply the changes.)
- **terraform apply** (used to apply the changes required to reach the desired state of the configuration. It executes the planned actions, such as creating, modifying, or destroying infrastructure resources, based on the configuration defined in .tf files and the current state of the infrastructure.)
- We can see the vpc actions.



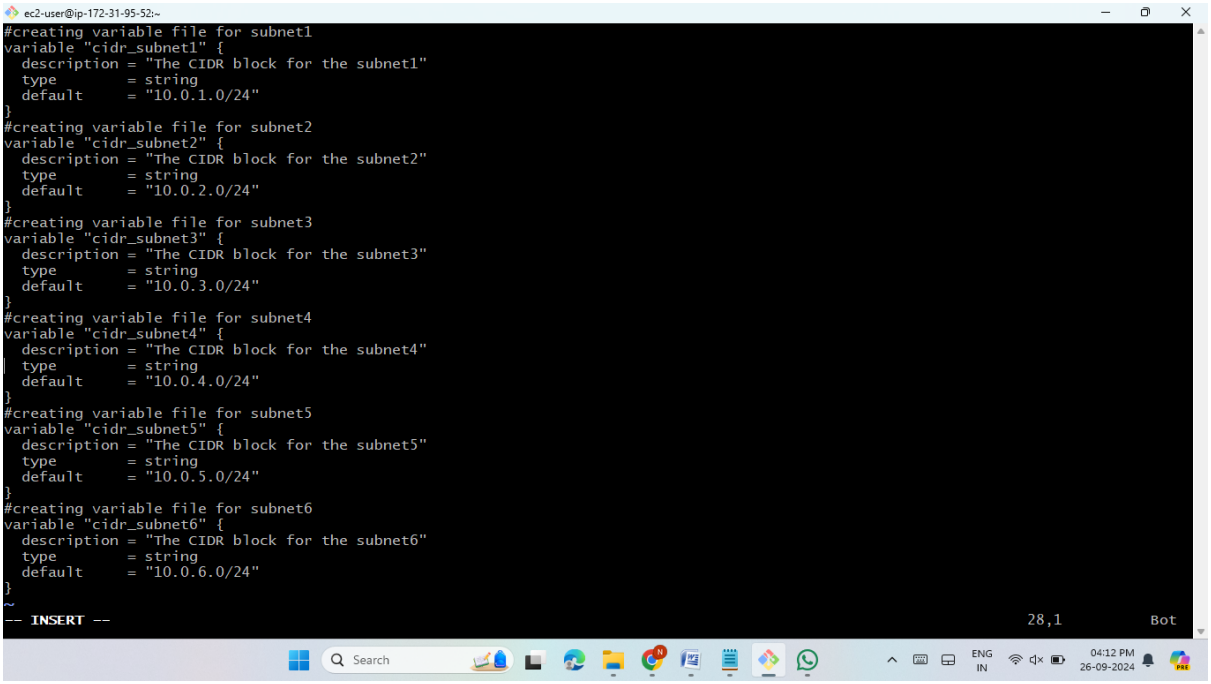
Create a file for the subnet

- In this project, I will create total 6 subnets for the front-end tier and back mixture of 4public & 2private subnet.
- Here Creating subnet.tf file



```
ec2-user@ip-172-31-95-52:~
resource "aws_subnet" "web-subnet2" {
  vpc_id                  = aws_vpc.yashvpc.id
  cidr_block              = var.cidr_subnet2
  map_public_ip_on_launch = true
  availability_zone       = "us-east-1b"
  tags = {
    Name = "web-subnet2"
  }
}
resource "aws_subnet" "application-subnet1" {
  vpc_id                  = aws_vpc.yashvpc.id
  cidr_block              = var.cidr_subnet3
  map_public_ip_on_launch = true
  availability_zone       = "us-east-1a"
  tags = {
    Name = "application-subnet1"
  }
}
resource "aws_subnet" "application-subnet2" {
  vpc_id                  = aws_vpc.yashvpc.id
  cidr_block              = var.cidr_subnet4
  map_public_ip_on_launch = true
  availability_zone       = "us-east-1b"
  tags = {
    Name = "application-subnet2"
  }
}
resource "aws_subnet" "database-subnet1" {
  vpc_id                  = aws_vpc.yashvpc.id
  cidr_block              = var.cidr_subnet5
  map_public_ip_on_launch = false
  availability_zone       = "us-east-1a"
  tags = {
    Name = "database-subnet1"
  }
}
resource "aws_subnet" "database-subnet2" {
  vpc_id                  = aws_vpc.yashvpc.id
  cidr_block              = var.cidr_subnet6
  map_public_ip_on_launch = false
  availability_zone       = "us-east-1b"
  tags = {
    Name = "database-subnet2"
  }
}
-- INSERT --                                                    50,1          Bot
```

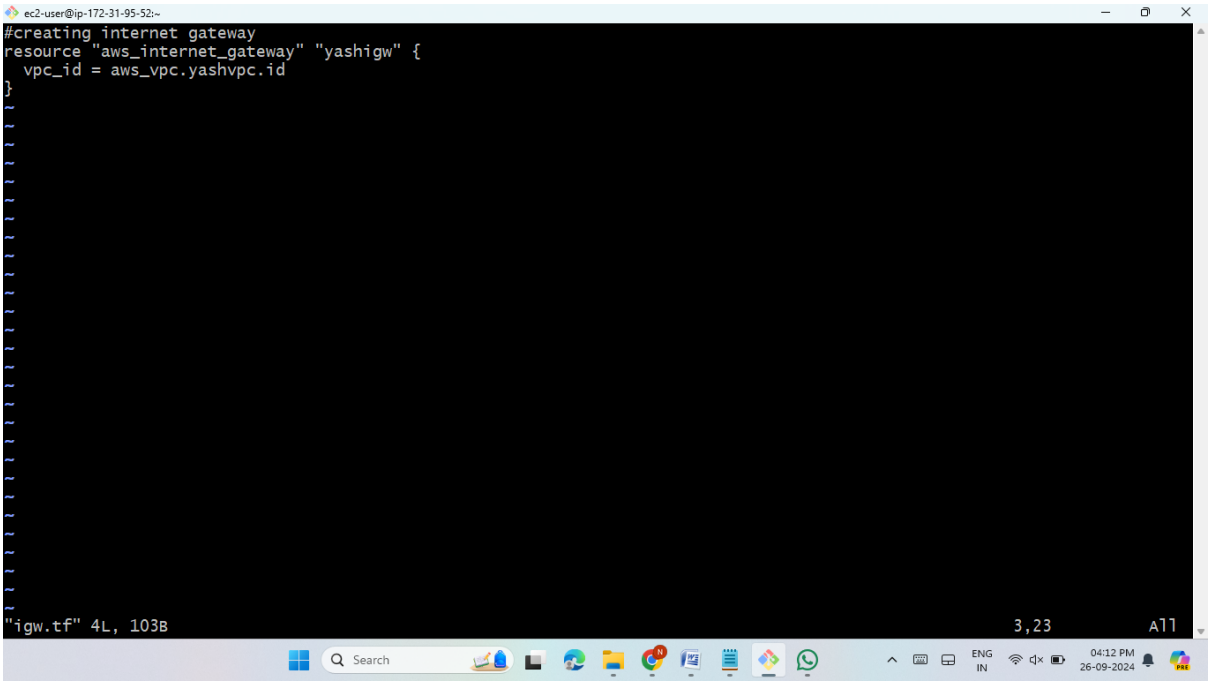
- Here Creating variable.tf file for 6 subnets



```
ec2-user@ip-172-31-95-52:~
#creating variable file for subnet1
variable "cidr_subnet1" {
  description = "The CIDR block for the subnet1"
  type        = string
  default     = "10.0.1.0/24"
}
#creating variable file for subnet2
variable "cidr_subnet2" {
  description = "The CIDR block for the subnet2"
  type        = string
  default     = "10.0.2.0/24"
}
#creating variable file for subnet3
variable "cidr_subnet3" {
  description = "The CIDR block for the subnet3"
  type        = string
  default     = "10.0.3.0/24"
}
#creating variable file for subnet4
variable "cidr_subnet4" {
  description = "The CIDR block for the subnet4"
  type        = string
  default     = "10.0.4.0/24"
}
#creating variable file for subnet5
variable "cidr_subnet5" {
  description = "The CIDR block for the subnet5"
  type        = string
  default     = "10.0.5.0/24"
}
#creating variable file for subnet6
variable "cidr_subnet6" {
  description = "The CIDR block for the subnet6"
  type        = string
  default     = "10.0.6.0/24"
}
~
-- INSERT --                                                    28,1          Bot
```

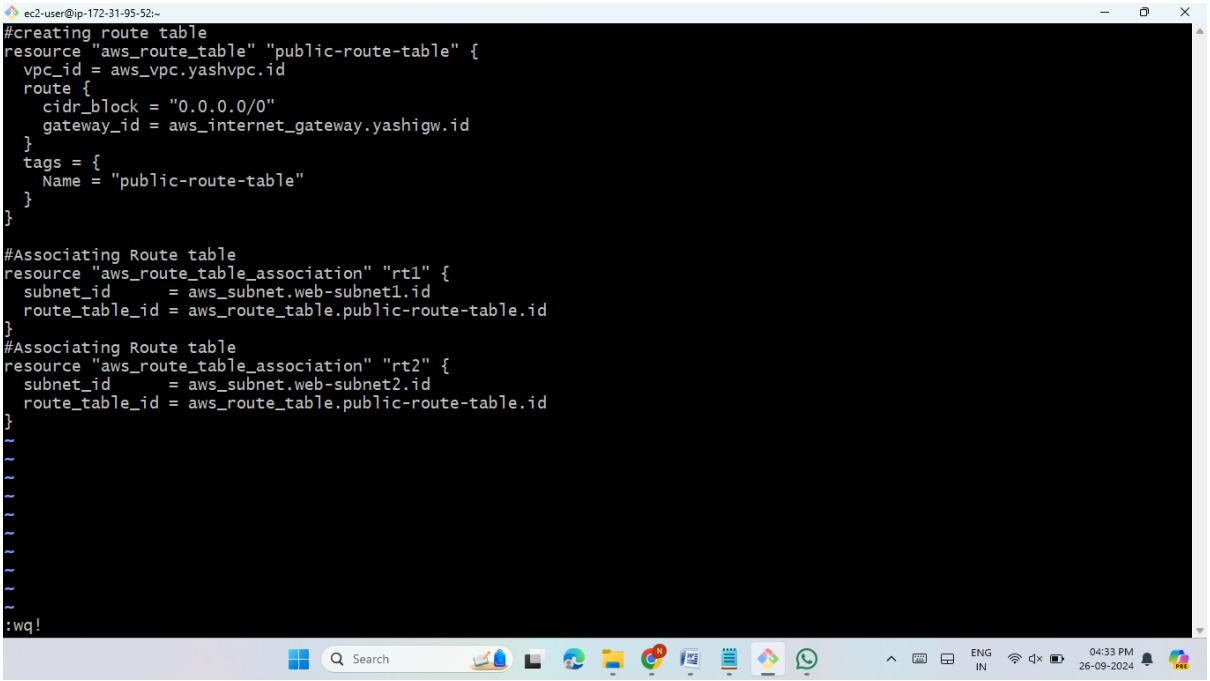
Create a file for Internet gateway

- Here Creating igw.tf file.



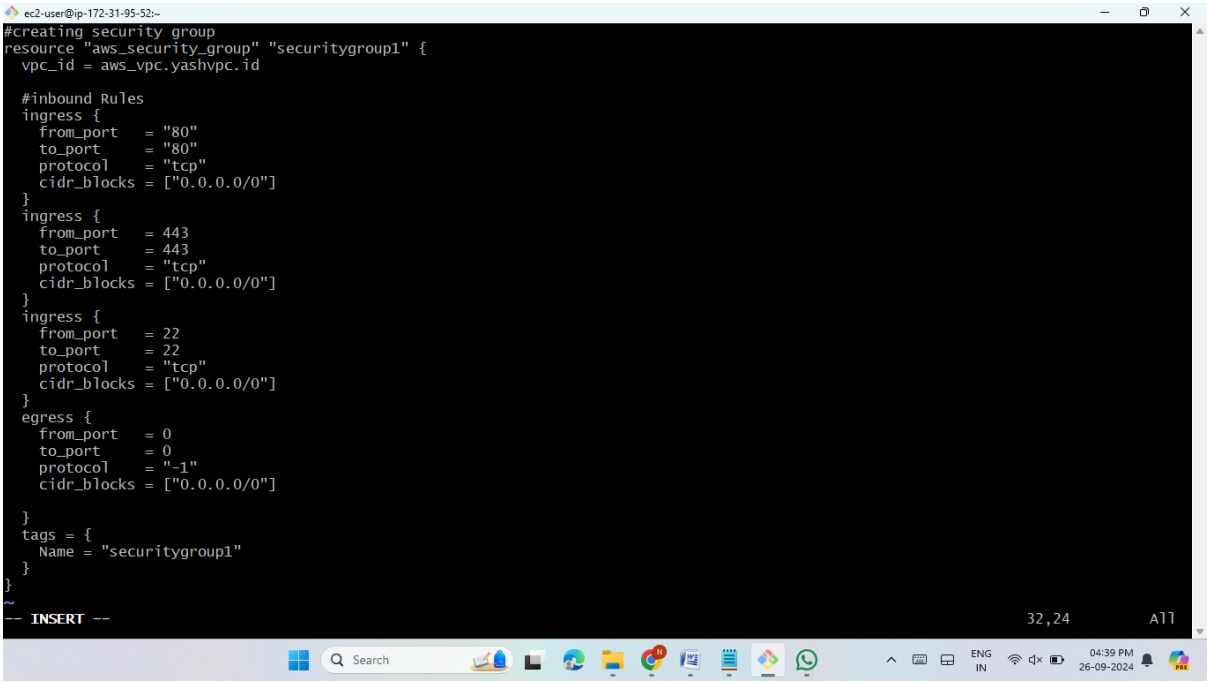
Create a file for the Route table

- Here Creating public- route-table.tf file.



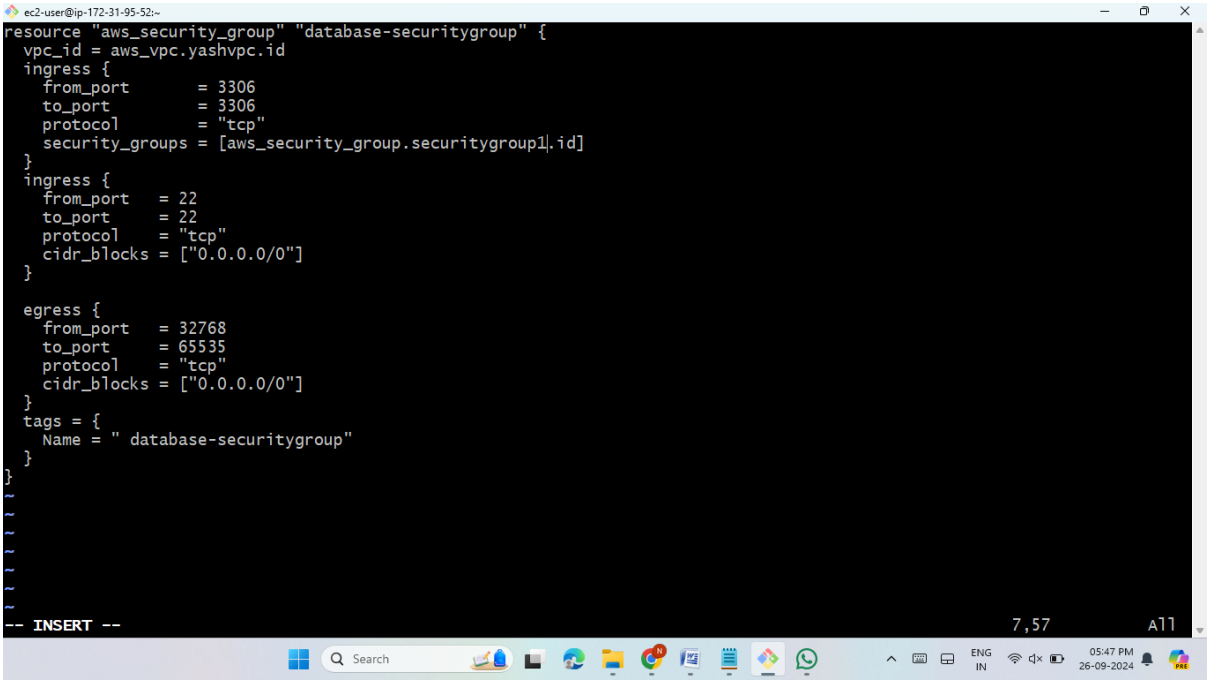
Create a file for security group

- Here Creating securitygroup1.tf file for instance1 and instance2 to launch the websites.



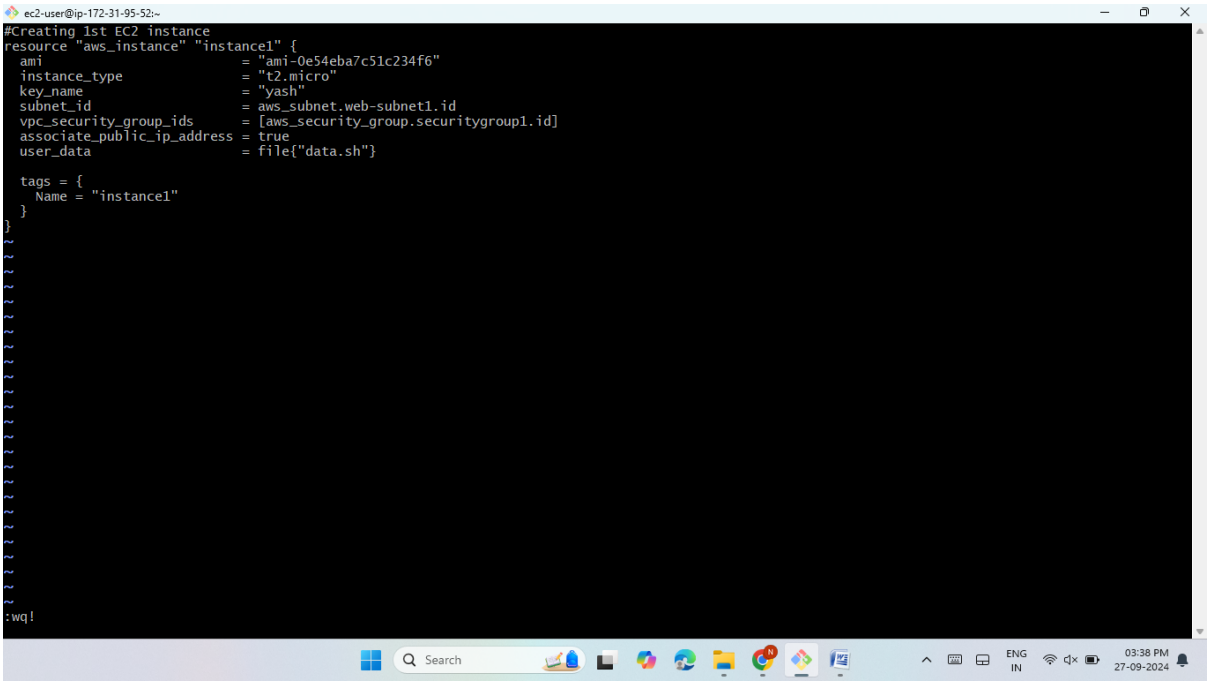
Create a file for security group for the database

- Create database-securitygroup.tf file for the RDS Database.

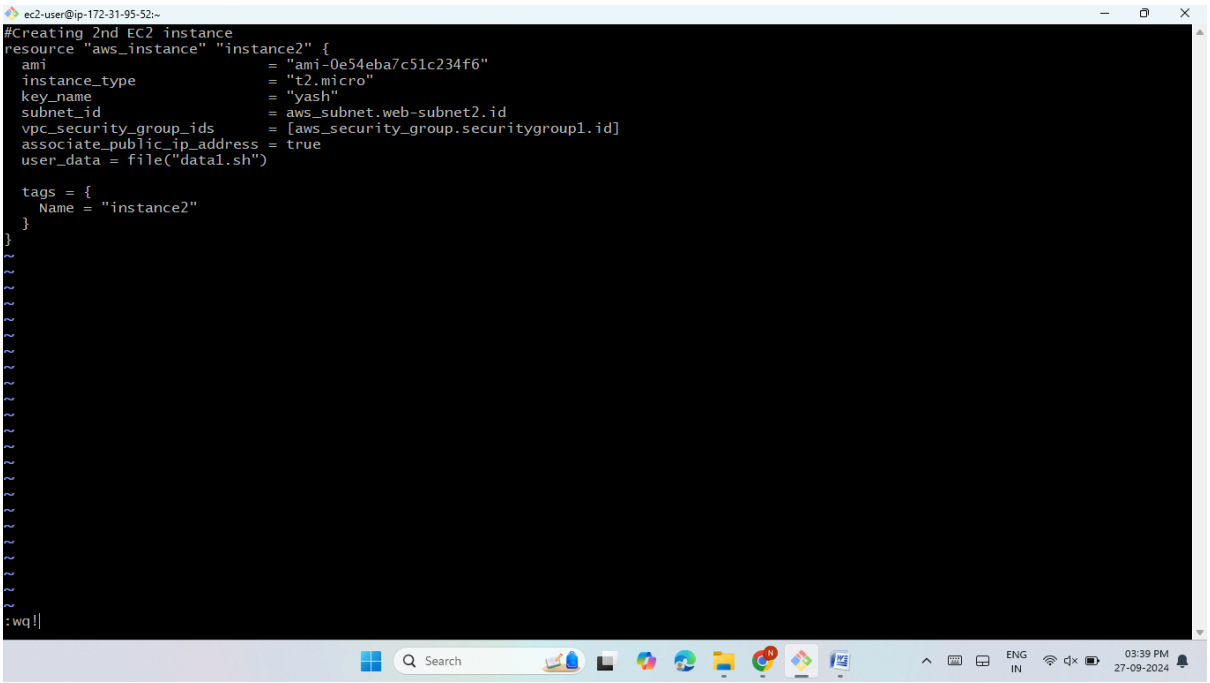


Create a file for the EC2 instances

- Here Creating instance1.tf file for hosting website.

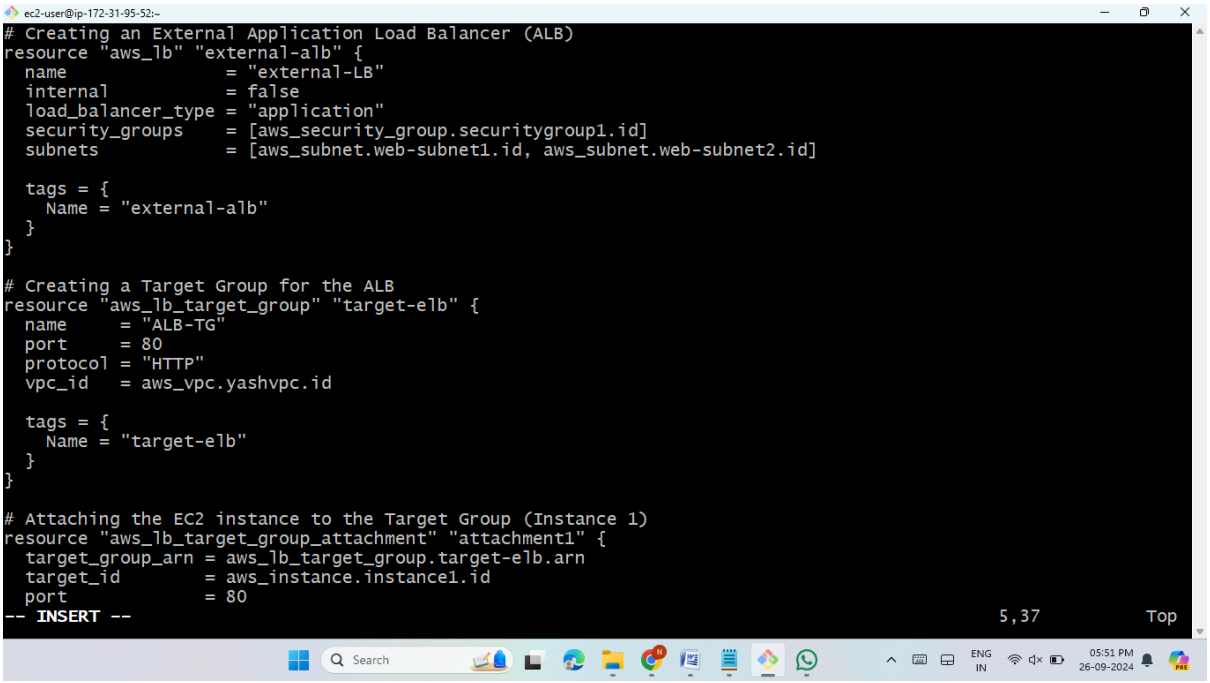


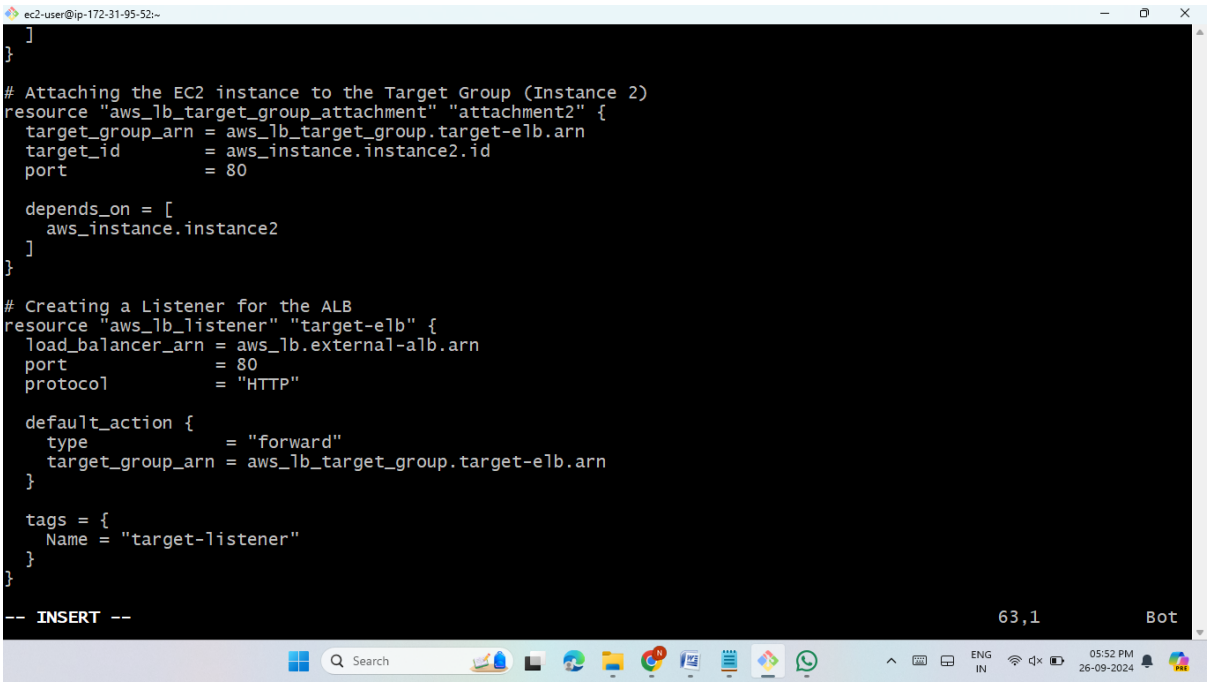
- Here Creating instance2.tf file for hosting website.



Create a file for application load balancer

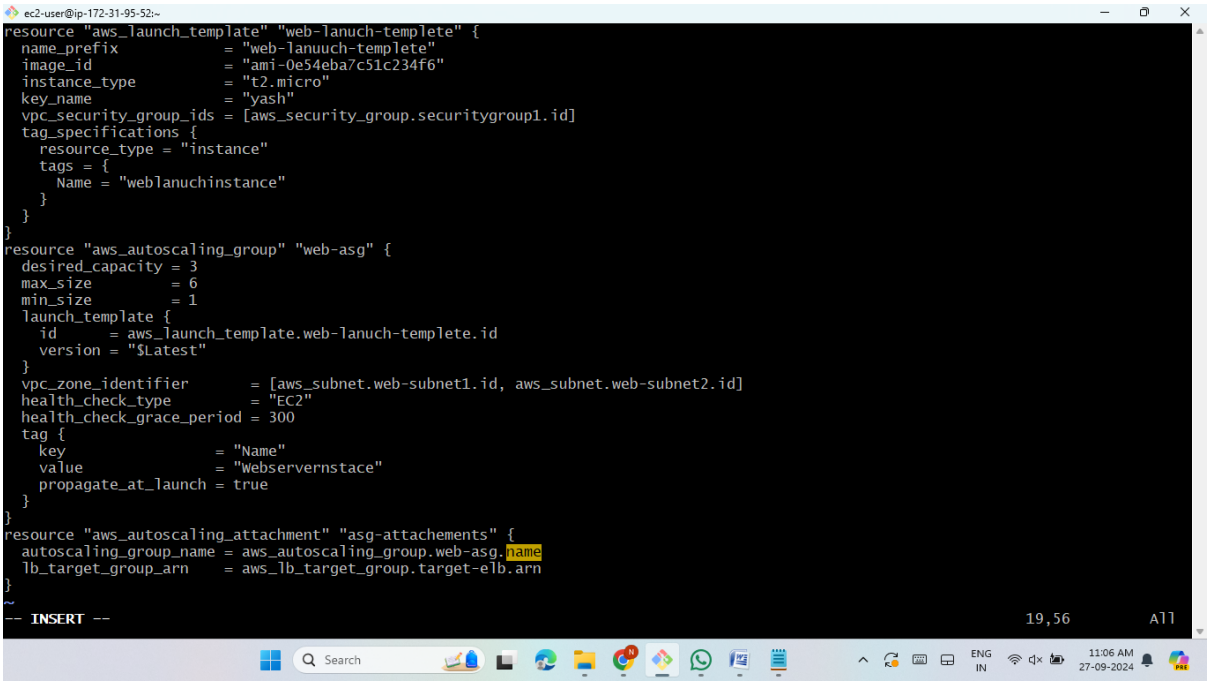
- Here creating application load balancer and target group file.





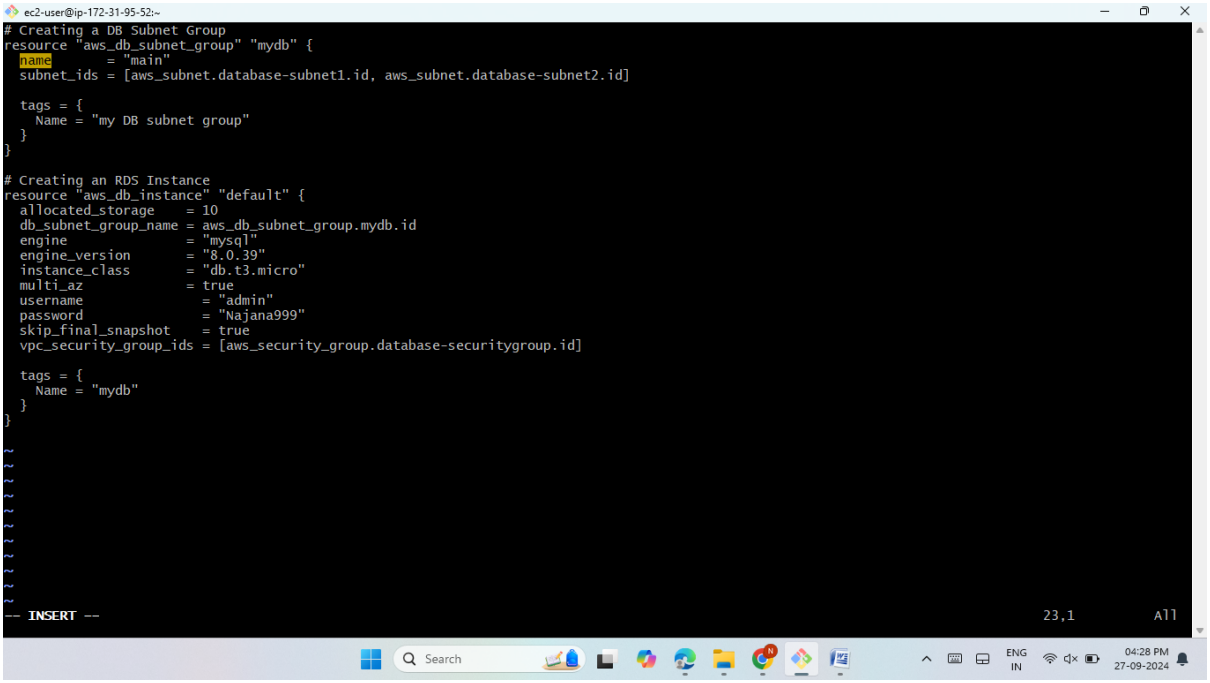
Create a file for Auto Scaling Group

- Here Creating auto scaling group.



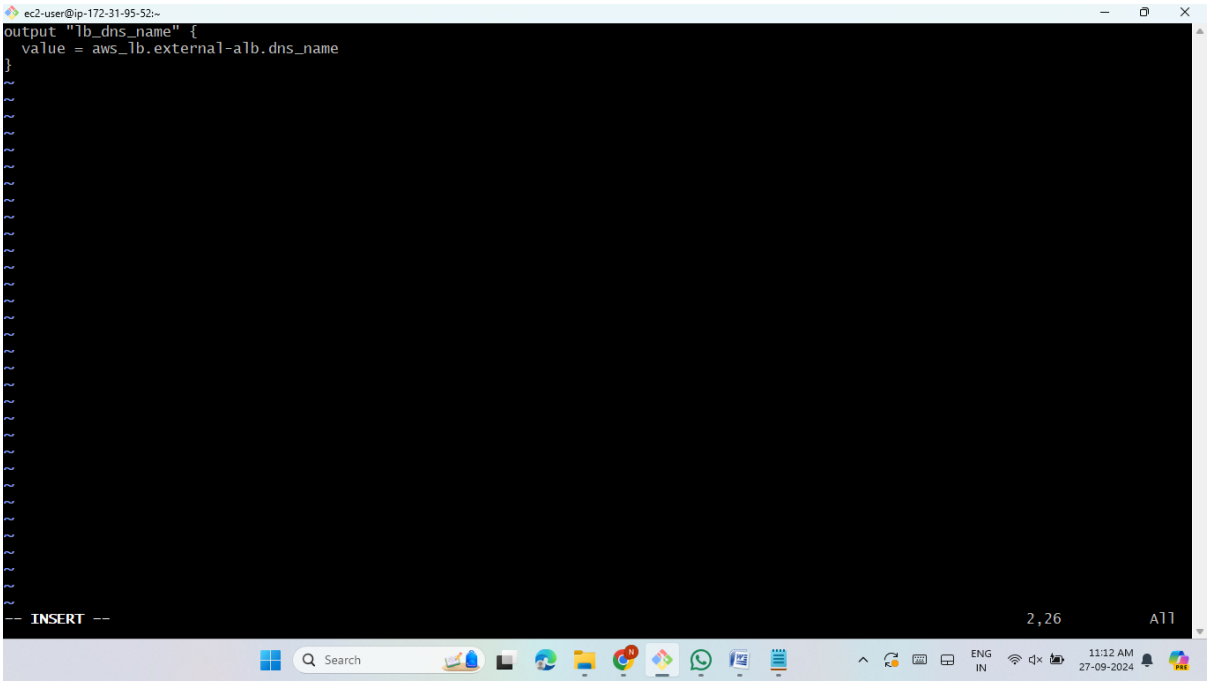
Create a file for the RDS instance

- Here Creating RDS file with using MYSQL Database.



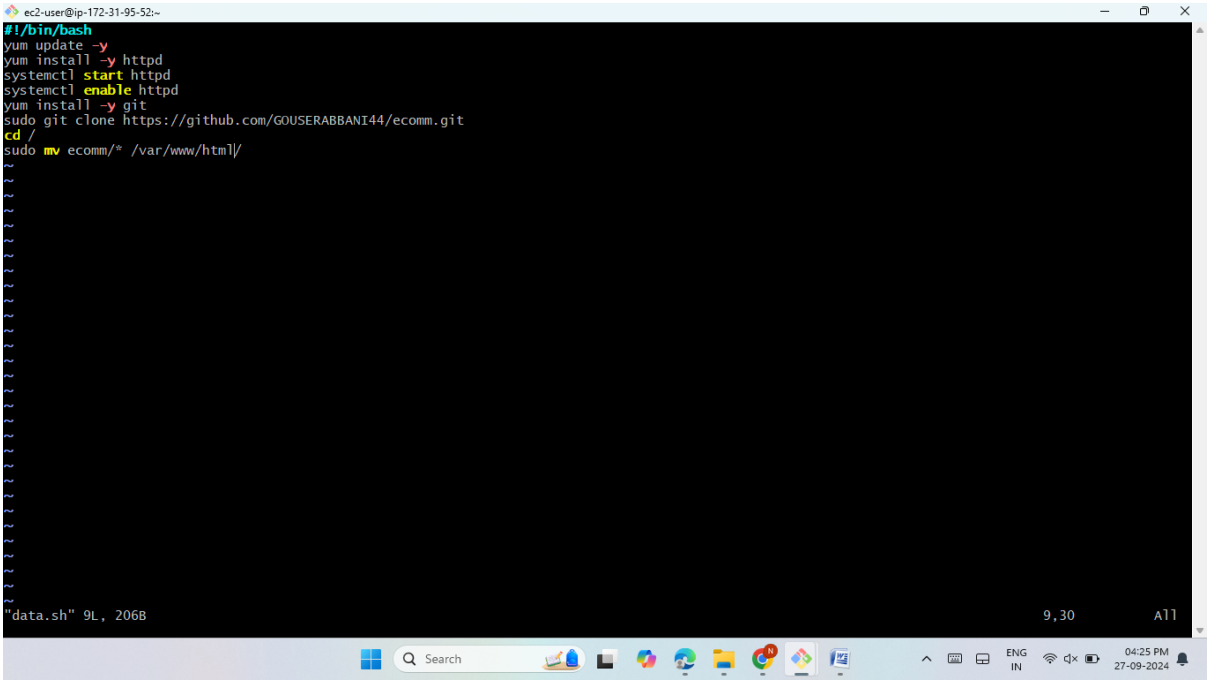
Create a file for output

- Creating output.tf file for host websites using DNS.

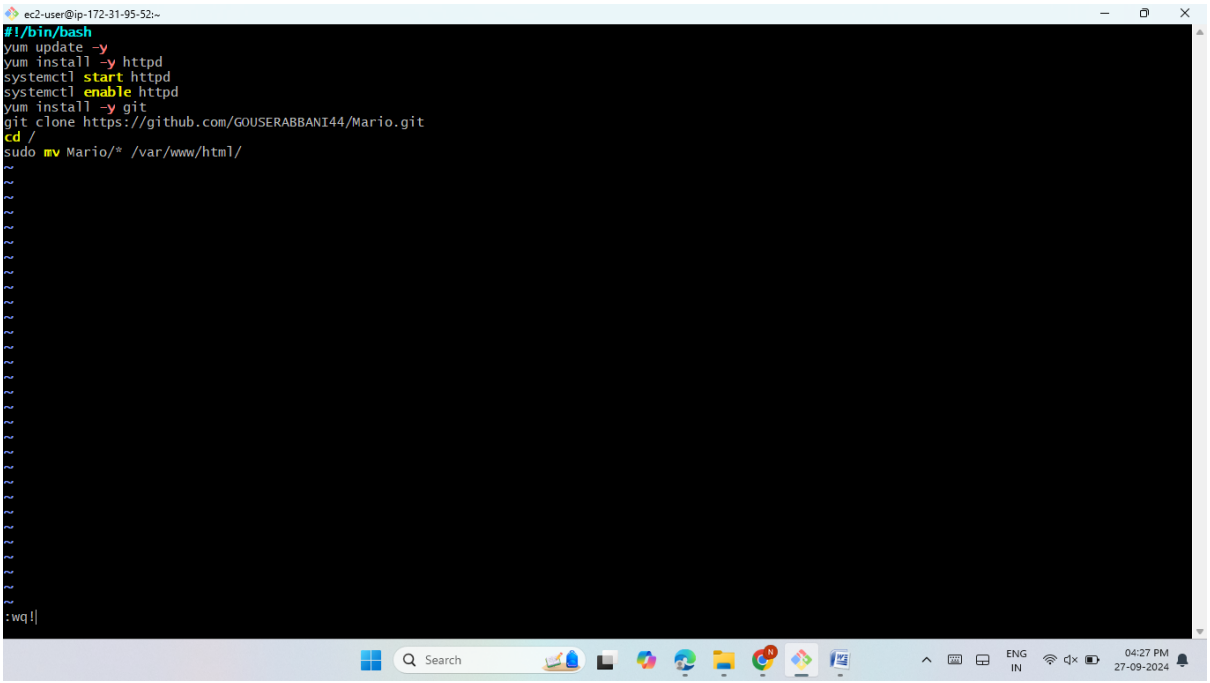


Create a file for user data

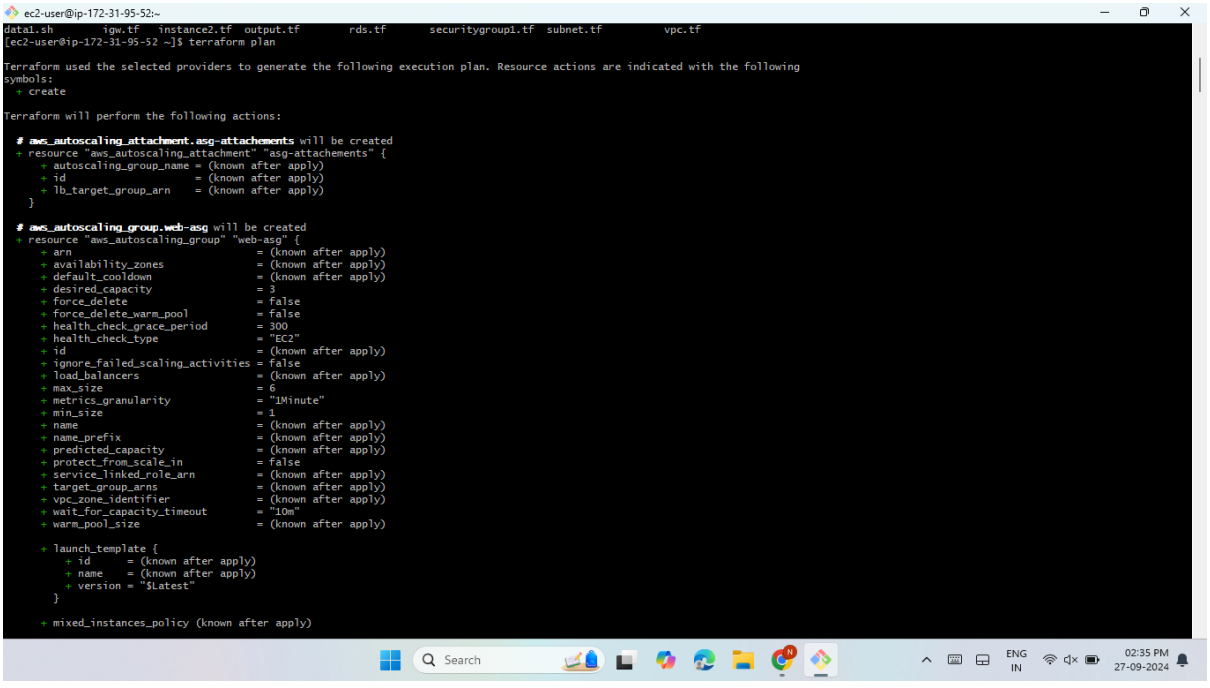
- Here Creating data.sh file for ecomm.git website.

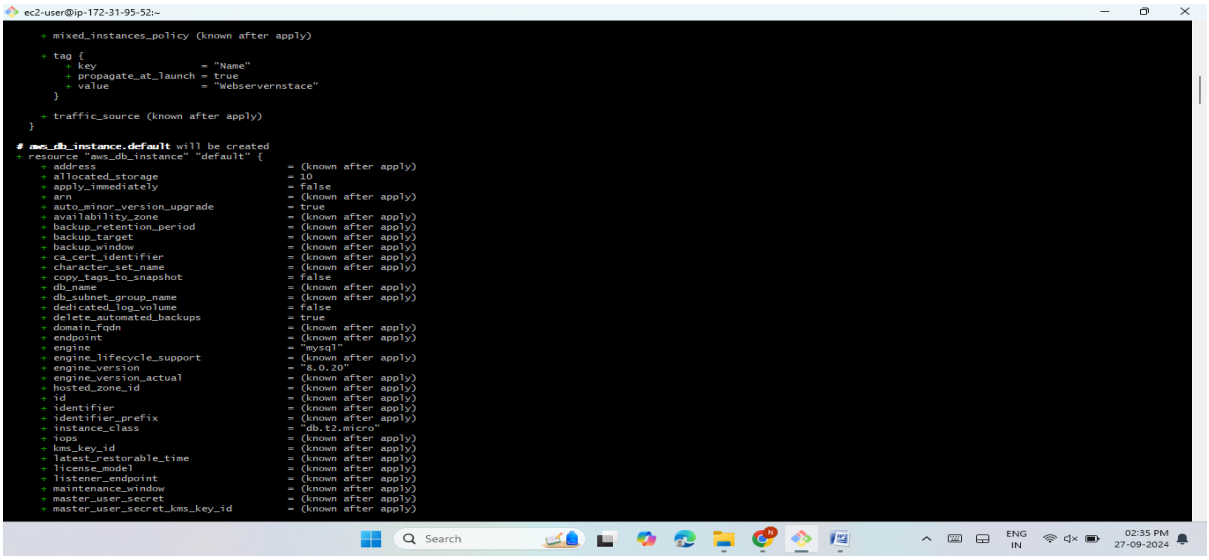


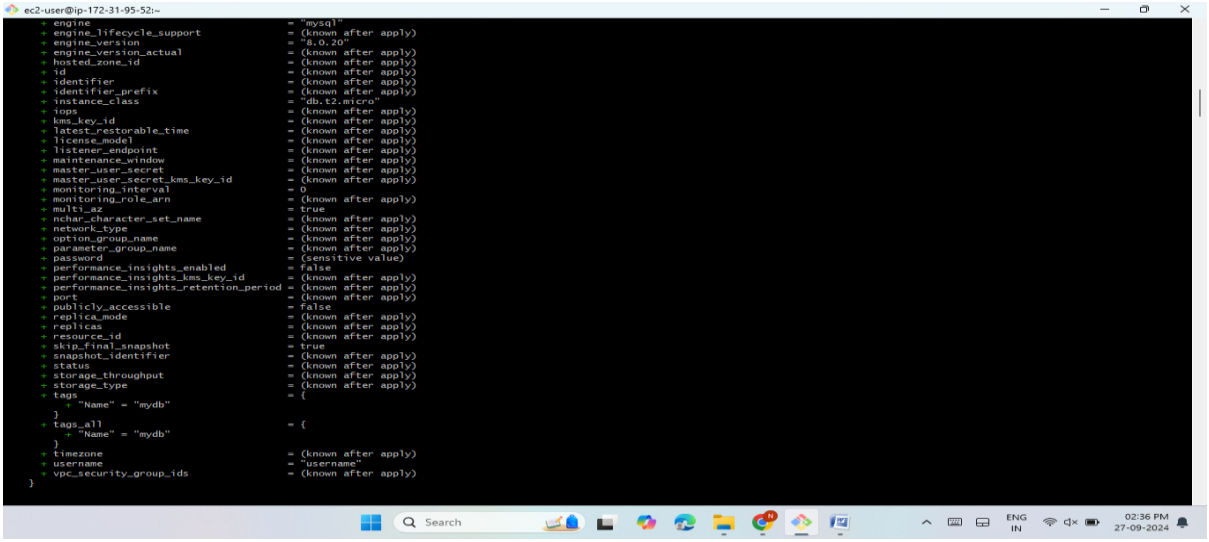
- Here Creating data1.sh file for Mario.git website.

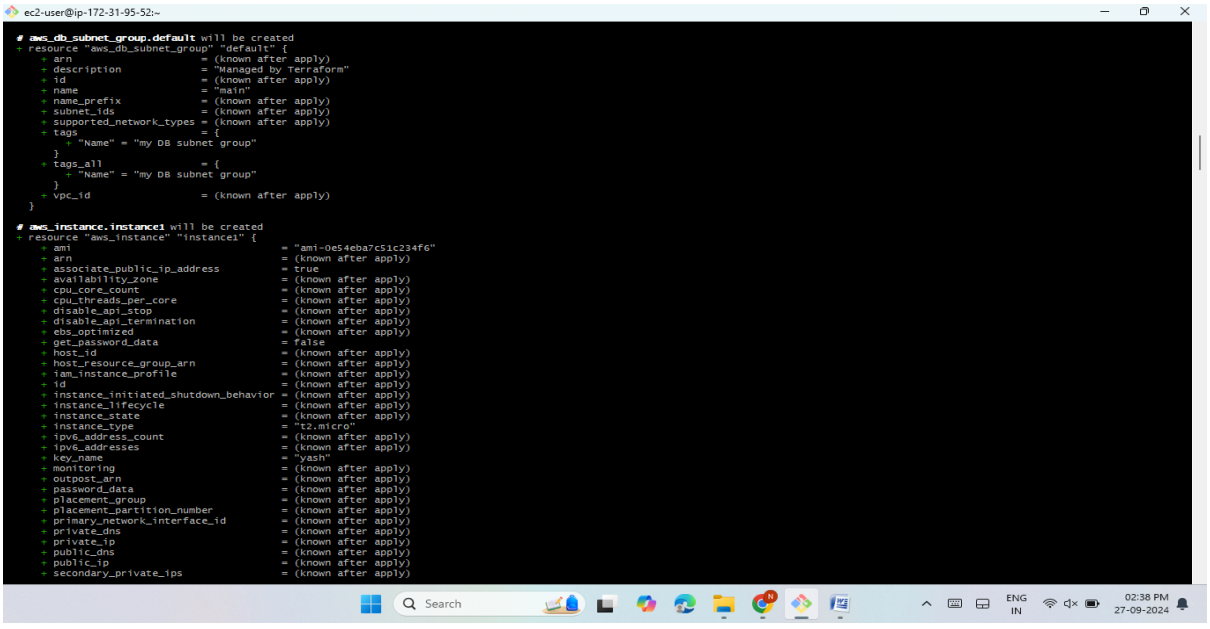


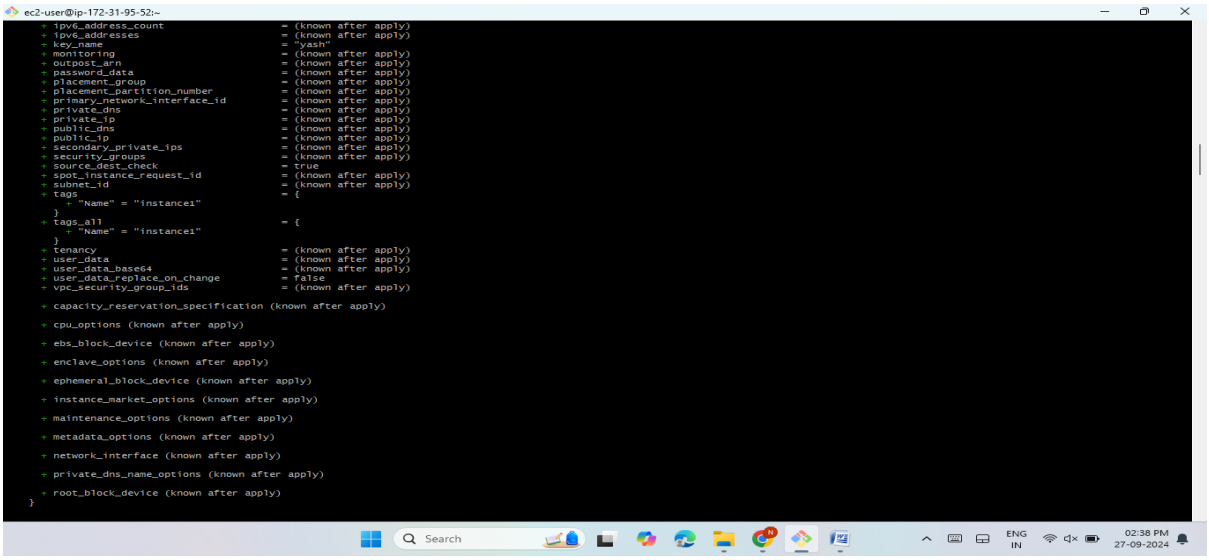
- Here I am using plan command to show the execution plan.
- We can see all the resources with the single command.

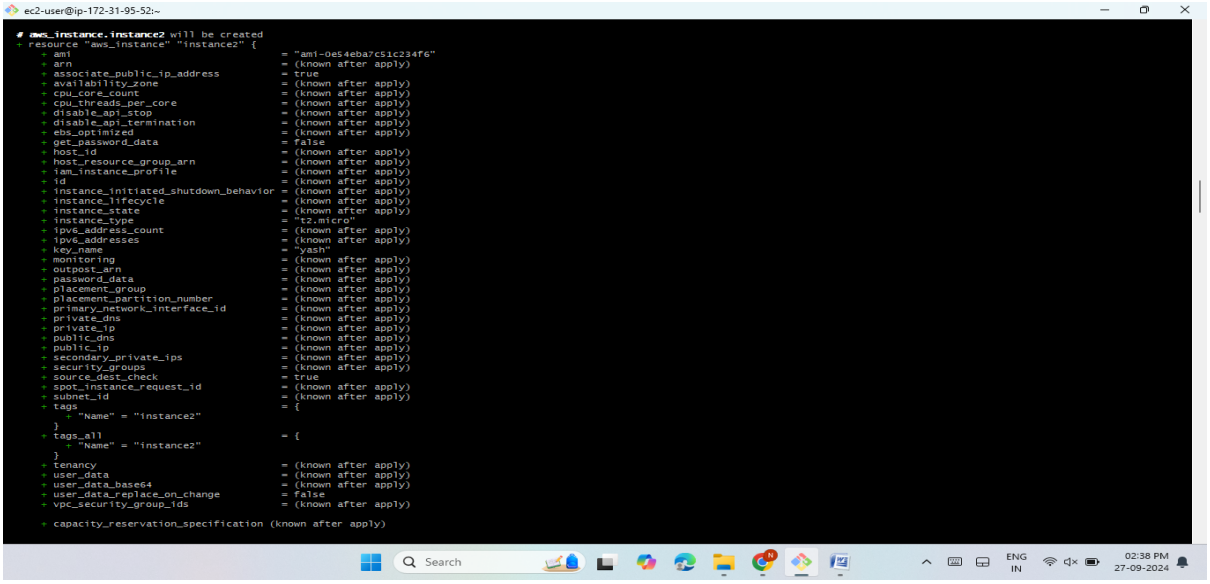


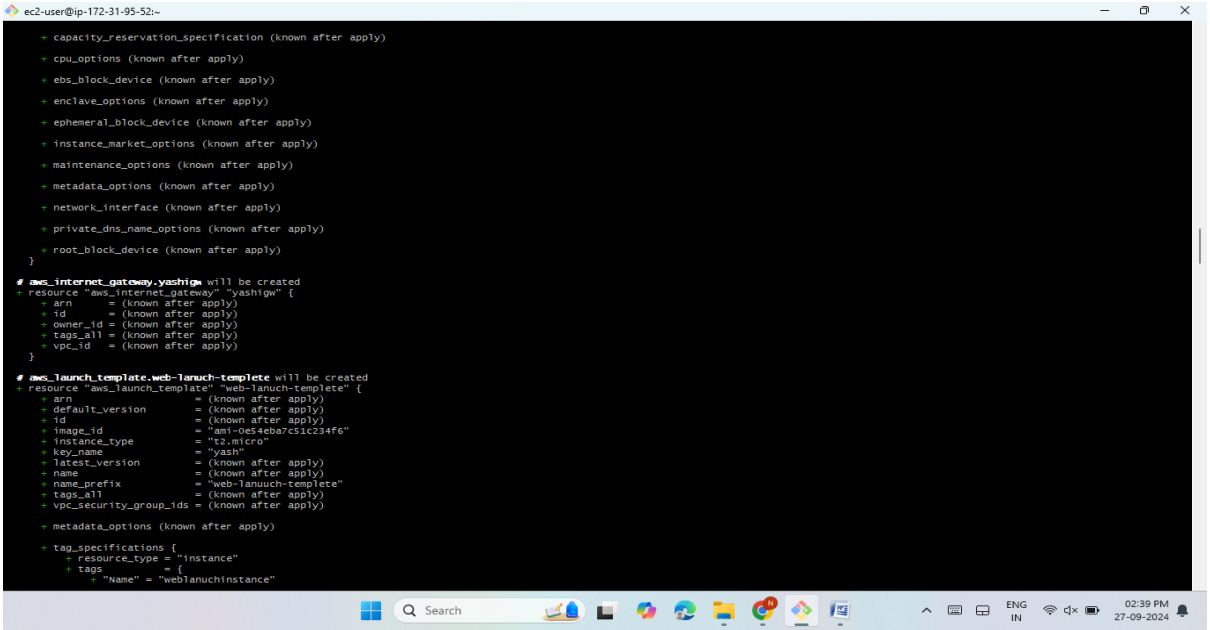

ec2-user@ip-172-31-95-52:~
+ mixed_instances_policy (known after apply)
+ tag {
+ key = "Name"
+ propagate_at_launch = true
+ value = "Webservernstace"
}
+ traffic_source (known after apply)
}
# aws_db_instance.default will be created
+ resource "aws_db_instance" "default" {
+ address = (known after apply)
+ allocated_storage = 10
+ apply_immediately = false
+ arn = (known after apply)
+ auto_minor_version_upgrade = true
+ availability_zone = (known after apply)
+ backup_retention_period = (known after apply)
+ backup_target = (known after apply)
+ backup_window = (known after apply)
+ ca_cert_identifier = (known after apply)
+ character_set_name = (known after apply)
+ copy_tags_to_snapshot = false
+ db_name = (known after apply)
+ db_subnet_group_name = (known after apply)
+ dedicated_log_volume = false
+ delete_automated_backups = true
+ domain_fqdn = (known after apply)
+ endpoint = (known after apply)
+ engine = "mysql"
+ engine_lifecycle_support = (known after apply)
+ engine_version = "8.0.20"
+ engine_version_actual = (known after apply)
+ hosted_zone_id = (known after apply)
+ id = (known after apply)
+ identifier = (known after apply)
+ identifier_prefix = (known after apply)
+ instance_class = "db.t2.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)
Search
ENG
IN
02:35 PM
27-09-2024

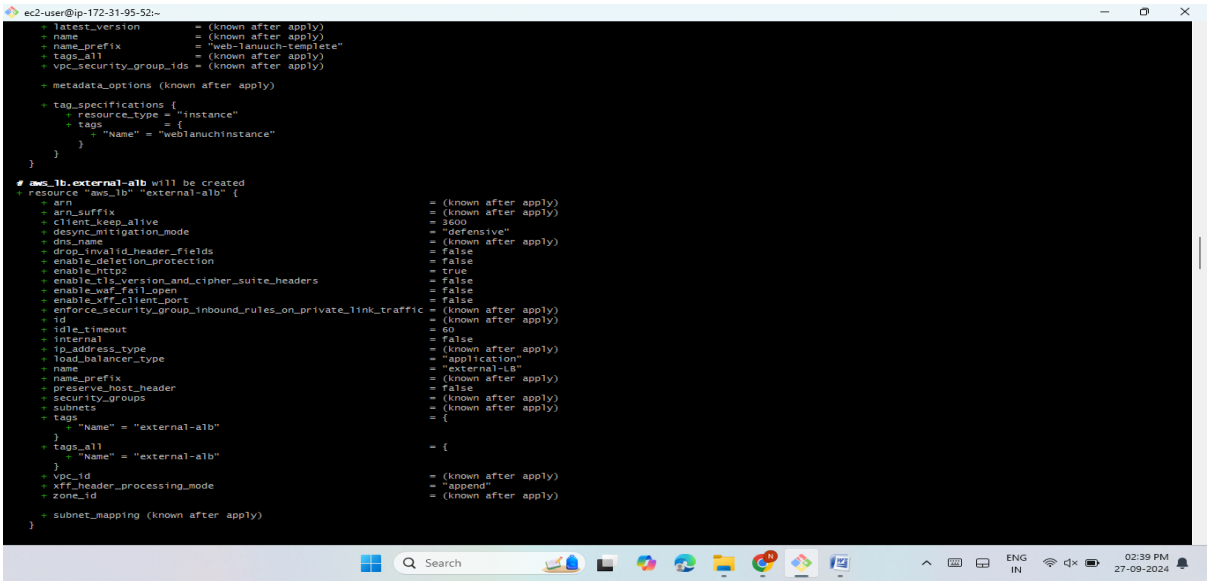

ec2-user@ip-172-31-95-52:~
+ engine = "mysql"
+ engine_lifecycle_support = (known after apply)
+ engine_version = "8.0.20"
+ engine_version_actual = (known after apply)
+ hosted_zone_id = (known after apply)
+ id = (known after apply)
+ identifier = (known after apply)
+ identifier_prefix = (known after apply)
+ instance_class = "db.t2.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 = true
+ 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 = true
+ snapshot_identifier = (known after apply)
+ status = (known after apply)
+ storage_throughput = (known after apply)
+ storage_type = (known after apply)
+ tags = {
+ "Name" = "mydb"
}
+ tags_all = {
+ "Name" = "mydb"
}
+ timezone = (known after apply)
+ username = "username"
+ vpc_security_group_ids = (known after apply)
}
Search
ENG
IN
02:36 PM
27-09-2024

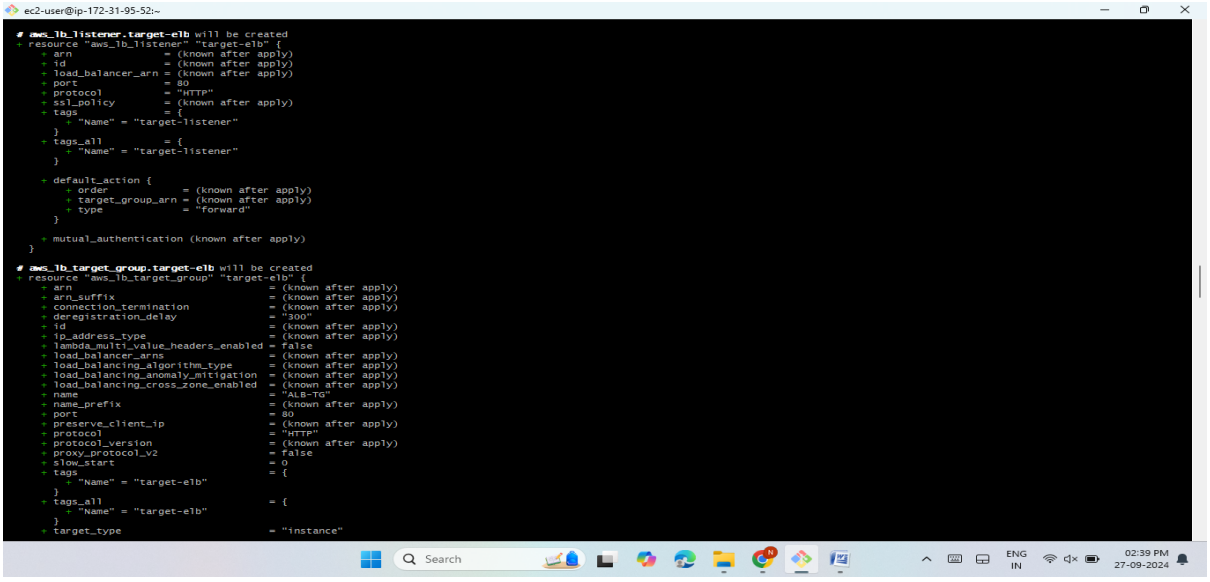

ec2-user@ip-172-31-95-52:~
# aws_db_subnet_group.default will be created
+ resource "aws_db_subnet_group" "default" {
+ arn = (known after apply)
+ description = "Managed by Terraform"
+ id = (known after apply)
+ name = "main"
+ name_prefix = (known after apply)
+ subnet_ids = (known after apply)
+ supported_network_types = (known after apply)
+ tags = {
+ "Name" = "my DB subnet group"
}
+ tags_all = {
+ "Name" = "my DB subnet group"
}
+ vpc_id = (known after apply)
}
# aws_instance.instance1 will be created
+ resource "aws_instance" "instance1" {
+ ami = "ami-0e54eba7c51c234f6"
+ arn = (known after apply)
+ associate_public_ip_address = true
+ 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
+ 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 = "yash"
+ 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)
Search
ENG
IN
02:38 PM
27-09-2024

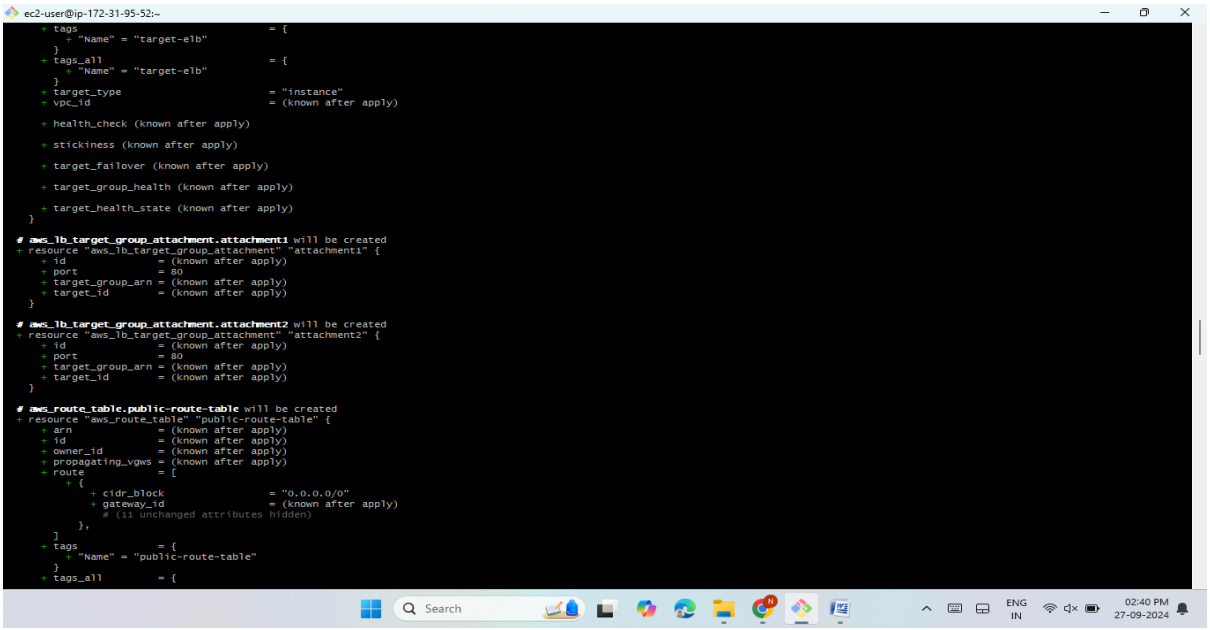

ec2-user@ip-172-31-95-52:~
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "yash"
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (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 = {
+ "Name" = "instance1"
}
+ tags_all = {
+ "Name" = "instance1"
}
+ 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)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
Search
ENG
IN
02:38 PM
27-09-2024

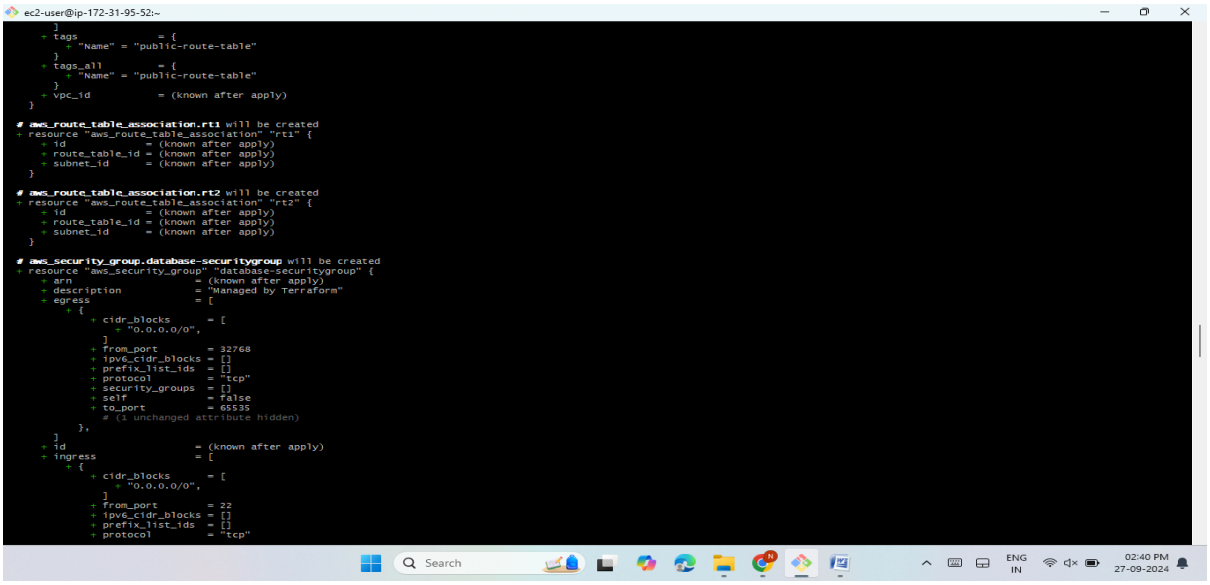

ec2-user@ip-172-31-95-52:~
# aws_instance.instance2 will be created
+ resource "aws_instance" "instance2" {
+ ami = "ami-0e54eba7c51c234f6"
+ arn = (known after apply)
+ associate_public_ip_address = true
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ get_password_data = false
+ 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 = "yash"
+ 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 = {
+ "Name" = "instance2"
}
+ tags_all = {
+ "Name" = "instance2"
}
+ 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)
+ capacity_reservation_specification (known after apply)
Search
ENG
IN
02:38 PM
27-09-2024

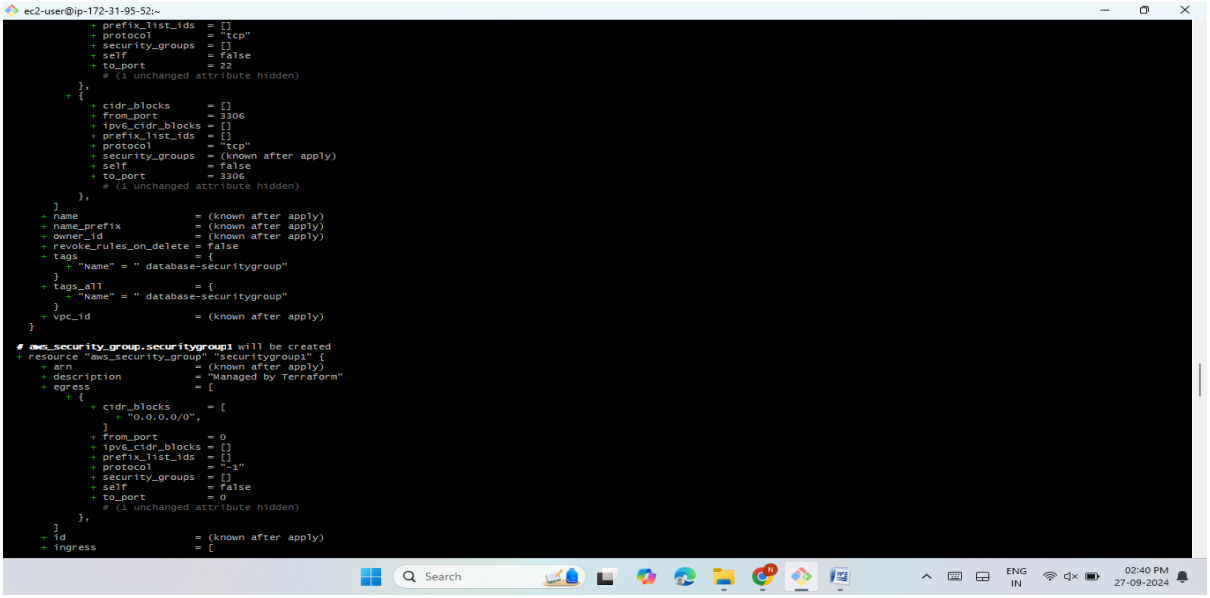

ec2-user@ip-172-31-95-52:~
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
# aws_internet_gateway.yashigw will be created
+ resource "aws_internet_gateway" "yashigw" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
# aws_launch_template.web-lanuch-templete will be created
+ resource "aws_launch_template" "web-lanuch-templete" {
+ arn = (known after apply)
+ default_version = (known after apply)
+ id = (known after apply)
+ image_id = "ami-0e54eba7c51c234f6"
+ instance_type = "t2.micro"
+ key_name = "yash"
+ latest_version = (known after apply)
+ name = (known after apply)
+ name_prefix = "web-lanuuch-templete"
+ tags_all = (known after apply)
+ vpc_security_group_ids = (known after apply)
+ metadata_options (known after apply)
+ tag_specifications {
+ resource_type = "instance"
+ tags = {
+ "Name" = "weblanuchinstance"
Search
ENG
IN
02:39 PM
27-09-2024

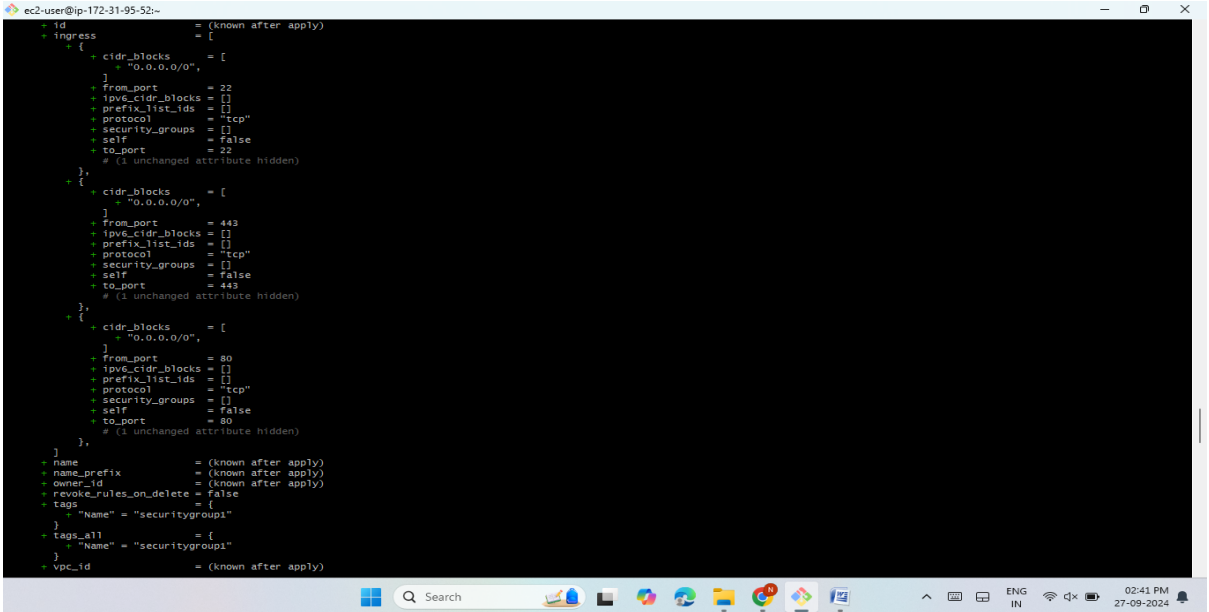

ec2-user@ip-172-31-95-52:~
+ latest_version = (known after apply)
+ name = (known after apply)
+ name_prefix = "web-lanuuch-templete"
+ tags_all = (known after apply)
+ vpc_security_group_ids = (known after apply)
+ metadata_options (known after apply)
+ tag_specifications {
+ resource_type = "instance"
+ tags = {
+ "Name" = "weblanuchinstance"
}
}
}
# aws_lb.external-alb will be created
+ resource "aws_lb" "external-alb" {
+ arn = (known after apply)
+ arn_suffix = (known after apply)
+ client_keep_alive = 3600
+ desync_mitigation_mode = "defensive"
+ dns_name = (known after apply)
+ drop_invalid_header_fields = false
+ enable_deletion_protection = false
+ enable_http2 = true
+ enable_tls_version_and_cipher_suite_headers = false
+ enable_waf_fail_open = false
+ enable_xff_client_port = false
+ enforce_security_group_inbound_rules_on_private_link_traffic = (known after apply)
+ id = (known after apply)
+ idle_timeout = 60
+ internal = false
+ ip_address_type = (known after apply)
+ load_balancer_type = "application"
+ name = "external-LB"
+ name_prefix = (known after apply)
+ preserve_host_header = false
+ security_groups = (known after apply)
+ subnets = (known after apply)
+ tags = {
+ "Name" = "external-alb"
}
+ tags_all = {
+ "Name" = "external-alb"
}
+ vpc_id = (known after apply)
+ xff_header_processing_mode = "append"
+ zone_id = (known after apply)
+ subnet_mapping (known after apply)
}
Search
ENG
IN
02:39 PM
27-09-2024

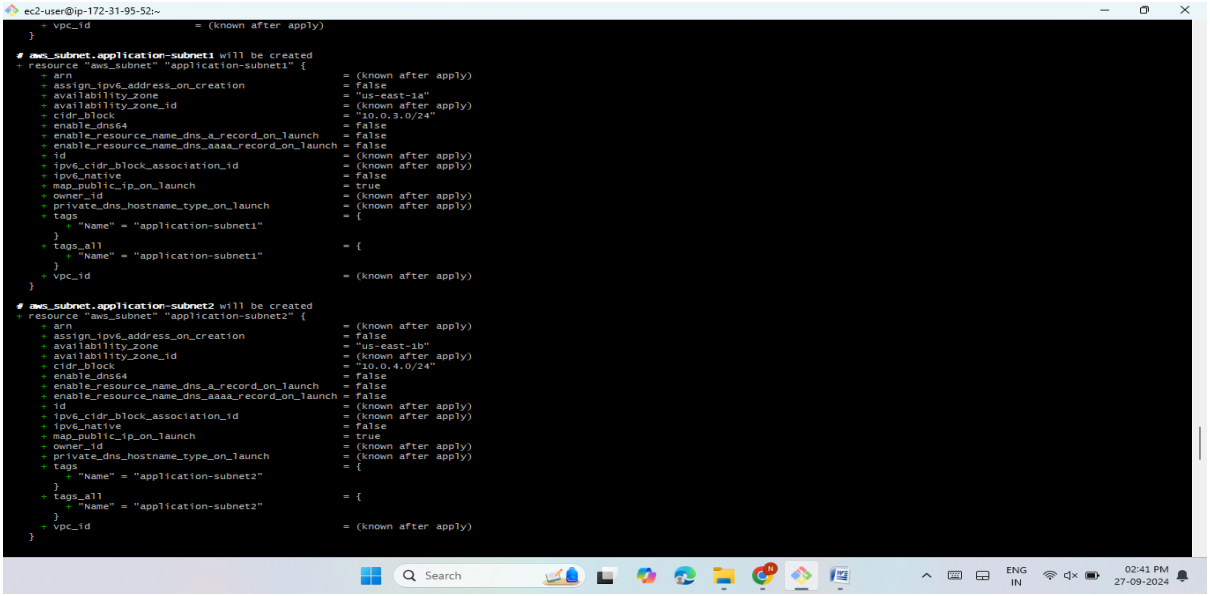

ec2-user@ip-172-31-95-52:~
# aws_lb_listener.target-elb will be created
+ resource "aws_lb_listener" "target-elb" {
+ arn = (known after apply)
+ id = (known after apply)
+ load_balancer_arn = (known after apply)
+ port = 80
+ protocol = "HTTP"
+ ssl_policy = (known after apply)
+ tags = {
+ "Name" = "target-listener"
}
+ tags_all = {
+ "Name" = "target-listener"
}
+ default_action {
+ order = (known after apply)
+ target_group_arn = (known after apply)
+ type = "forward"
}
+ mutual_authentication (known after apply)
}
# aws_lb_target_group.target-elb will be created
+ resource "aws_lb_target_group" "target-elb" {
+ arn = (known after apply)
+ arn_suffix = (known after apply)
+ connection_termination = (known after apply)
+ deregistration_delay = "300"
+ id = (known after apply)
+ ip_address_type = (known after apply)
+ lambda_multi_value_headers_enabled = false
+ load_balancer_arns = (known after apply)
+ load_balancing_algorithm_type = (known after apply)
+ load_balancing_anomaly_mitigation = (known after apply)
+ load_balancing_cross_zone_enabled = (known after apply)
+ name = "ALB-TG"
+ name_prefix = (known after apply)
+ port = 80
+ preserve_client_ip = (known after apply)
+ protocol = "HTTP"
+ protocol_version = (known after apply)
+ proxy_protocol_v2 = false
+ slow_start = 0
+ tags = {
+ "Name" = "target-elb"
}
+ tags_all = {
+ "Name" = "target-elb"
}
+ target_type = "instance"
Search
ENG
IN
02:39 PM
27-09-2024

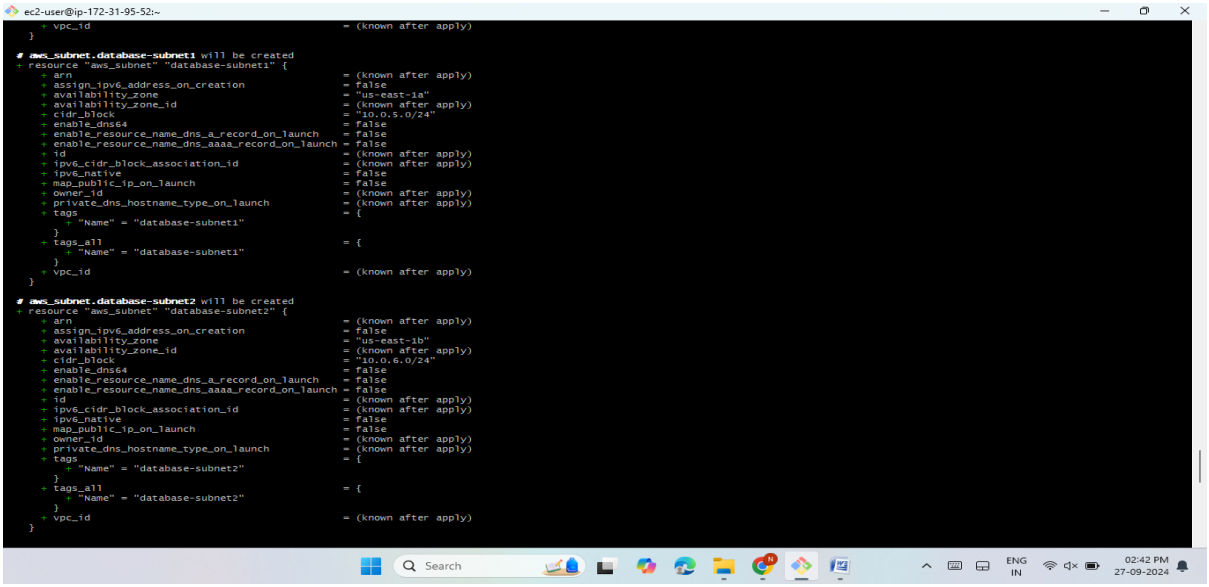

ec2-user@ip-172-31-95-52:~
+ tags = {
+ "Name" = "target-elb"
}
+ tags_all = {
+ "Name" = "target-elb"
}
+ target_type = "instance"
+ vpc_id = (known after apply)
+ health_check (known after apply)
+ stickiness (known after apply)
+ target_failover (known after apply)
+ target_group_health (known after apply)
+ target_health_state (known after apply)
}
# aws_lb_target_group_attachment.attachment1 will be created
+ resource "aws_lb_target_group_attachment" "attachment1" {
+ id = (known after apply)
+ port = 80
+ target_group_arn = (known after apply)
+ target_id = (known after apply)
}
# aws_lb_target_group_attachment.attachment2 will be created
+ resource "aws_lb_target_group_attachment" "attachment2" {
+ id = (known after apply)
+ port = 80
+ target_group_arn = (known after apply)
+ target_id = (known after apply)
}
# aws_route_table.public-route-table will be created
+ resource "aws_route_table" "public-route-table" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
+ propagating_vgws = (known after apply)
+ route = [
+ {
+ cidr_block = "0.0.0.0/0"
+ gateway_id = (known after apply)
# (11 unchanged attributes hidden)
},
]
+ tags = {
+ "Name" = "public-route-table"
}
+ tags_all = {
Search
ENG
IN
02:40 PM
27-09-2024

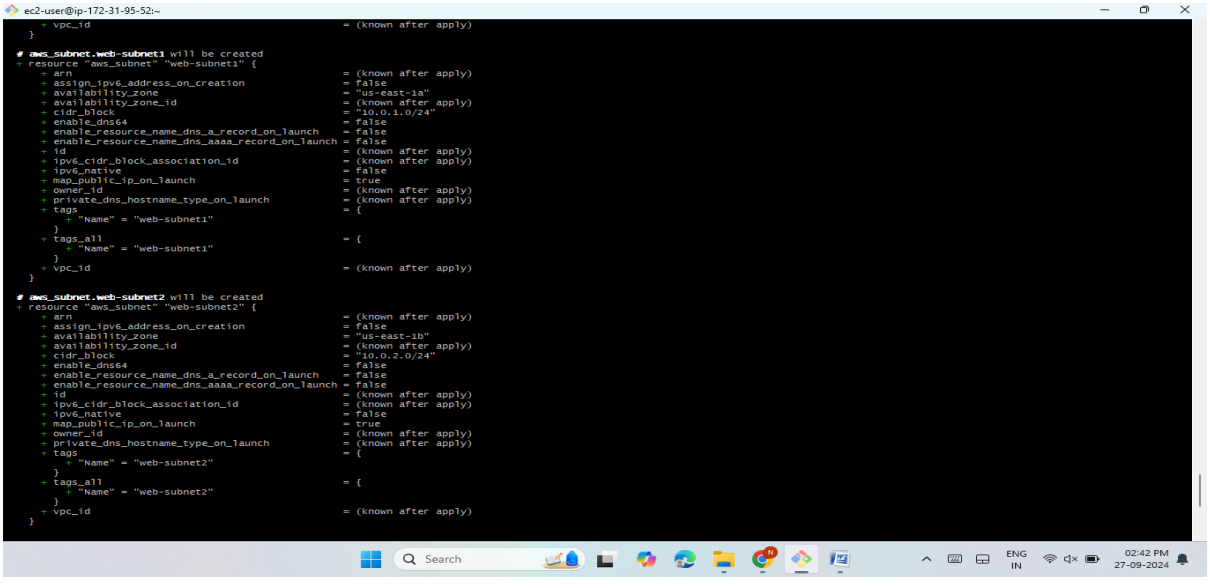

ec2-user@ip-172-31-95-52:~
aws_route_table_association.rt1 will be created
aws_route_table_association.rt2 will be created
aws_security_group.database-securitygroup will be created
Search
ENG
IN
02:40 PM
27-09-2024

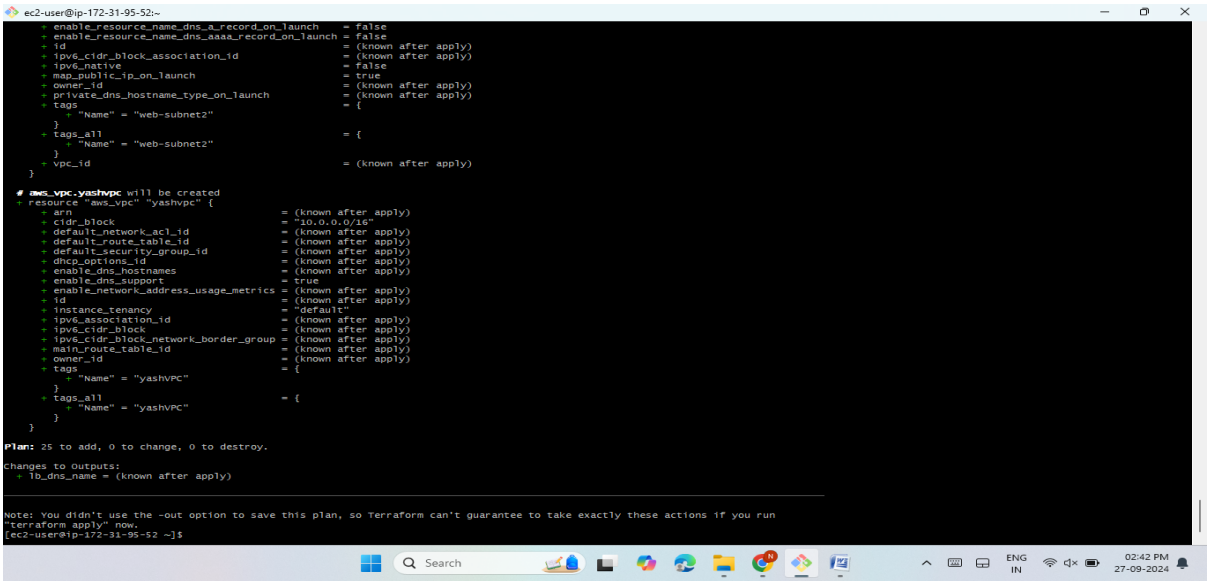

ec2-user@ip-172-31-95-52:~
aws_security_group.securitygroup1 will be created
Search
ENG
IN
02:40 PM
27-09-2024

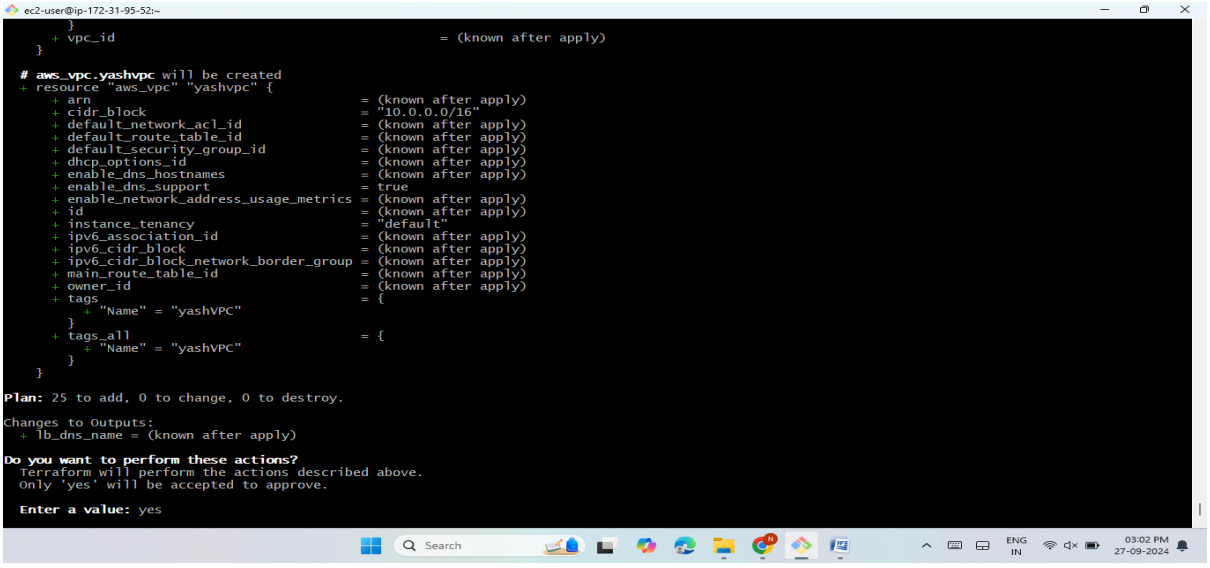

ec2-user@ip-172-31-95-52:~
Search
ENG
IN
02:41 PM
27-09-2024

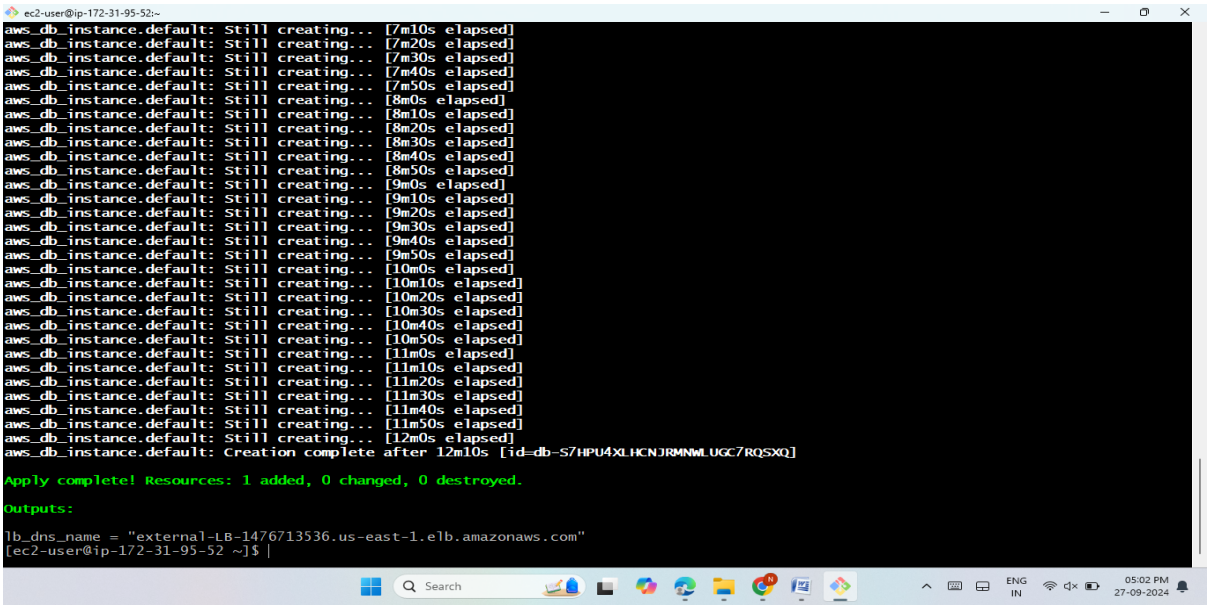

ec2-user@ip-172-31-95-52:~
+ vpc_id = (known after apply)
}
# aws_subnet.application-subnet1 will be created
+ resource "aws_subnet" "application-subnet1" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1a"
+ availability_zone_id = (known after apply)
+ 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 = true
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags = {
+ "Name" = "application-subnet1"
}
+ tags_all = {
+ "Name" = "application-subnet1"
}
+ vpc_id = (known after apply)
}
# aws_subnet.application-subnet2 will be created
+ resource "aws_subnet" "application-subnet2" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1b"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.4.0/24"
+ 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 = {
+ "Name" = "application-subnet2"
}
+ tags_all = {
+ "Name" = "application-subnet2"
}
+ vpc_id = (known after apply)
}
Search
ENG
IN
02:41 PM
27-09-2024


ec2-user@ip-172-31-95-52:~
+ vpc_id = (known after apply)
}
# aws_subnet.database-subnet1 will be created
+ resource "aws_subnet" "database-subnet1" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1a"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.5.0/24"
+ 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" = "database-subnet1"
}
+ tags_all = {
+ "Name" = "database-subnet1"
}
+ vpc_id = (known after apply)
}
# aws_subnet.database-subnet2 will be created
+ resource "aws_subnet" "database-subnet2" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1b"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.6.0/24"
+ 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" = "database-subnet2"
}
+ tags_all = {
+ "Name" = "database-subnet2"
}
+ vpc_id = (known after apply)
}
Search
ENG
IN
02:42 PM
27-09-2024


ec2-user@ip-172-31-95-52:~
+ vpc_id = (known after apply)
}
# aws_subnet.web-subnet1 will be created
+ resource "aws_subnet" "web-subnet1" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1a"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.1.0/24"
+ 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 = {
+ "Name" = "web-subnet1"
}
+ tags_all = {
+ "Name" = "web-subnet1"
}
+ vpc_id = (known after apply)
}
# aws_subnet.web-subnet2 will be created
+ resource "aws_subnet" "web-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 = {
+ "Name" = "web-subnet2"
}
+ tags_all = {
+ "Name" = "web-subnet2"
}
+ vpc_id = (known after apply)
}
Search
ENG
IN
02:42 PM
27-09-2024

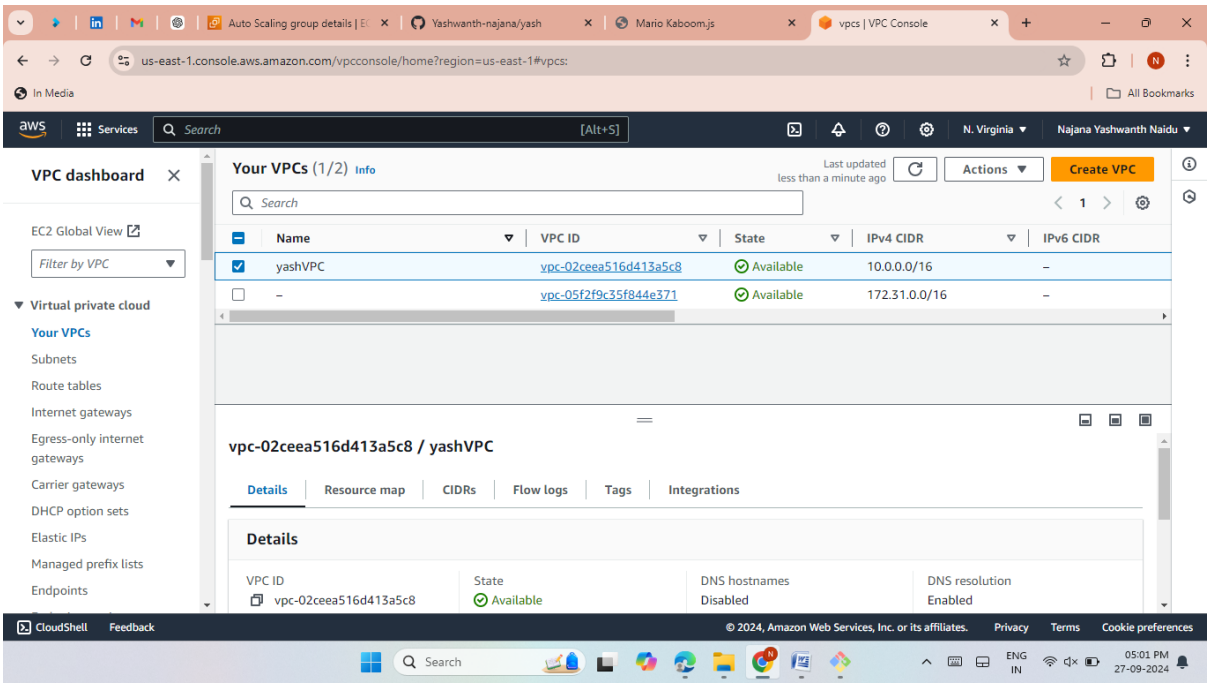

ec2-user@ip-172-31-95-52:~
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id = (known after apply)
+ ipv6_cidr_block_association_id = (known after apply)
+ ipv6_native = false
+ map_public_ip_on_launch = true
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags = {
+ "Name" = "web-subnet2"
}
+ tags_all = {
+ "Name" = "web-subnet2"
}
+ vpc_id = (known after apply)
}
# aws_vpc.yashvpc will be created
+ resource "aws_vpc" "yashvpc" {
+ 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 = {
+ "Name" = "yashVPC"
}
+ tags_all = {
+ "Name" = "yashVPC"
}
}
Plan: 25 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ lb_dns_name = (known after apply)
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.
[ec2-user@ip-172-31-95-52 ~]$
Search
ENG
IN
02:42 PM
27-09-2024


ec2-user@ip-172-31-95-52:~
}
+ vpc_id = (known after apply)
}
# aws_vpc.yashvpc will be created
+ resource "aws_vpc" "yashvpc" {
+ 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 = {
+ "Name" = "yashVPC"
}
+ tags_all = {
+ "Name" = "yashVPC"
}
}
Plan: 25 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ lb_dns_name = (known after apply)
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
Search
ENG
IN
03:02 PM
27-09-2024

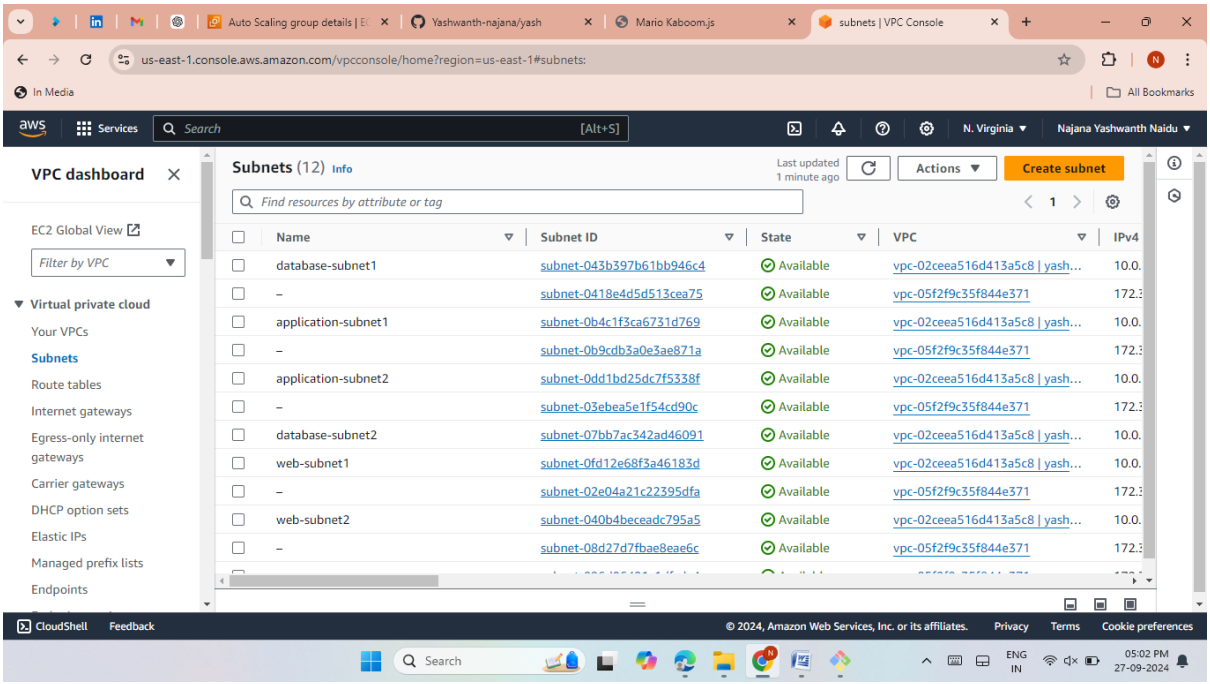

ec2-user@ip-172-31-95-52:~
aws_db_instance.default: Still creating... [7m10s elapsed]
aws_db_instance.default: Still creating... [7m20s elapsed]
aws_db_instance.default: Still creating... [7m30s elapsed]
aws_db_instance.default: Still creating... [7m40s elapsed]
aws_db_instance.default: Still creating... [7m50s elapsed]
aws_db_instance.default: Still creating... [8m0s elapsed]
aws_db_instance.default: Still creating... [8m10s elapsed]
aws_db_instance.default: Still creating... [8m20s elapsed]
aws_db_instance.default: Still creating... [8m30s elapsed]
aws_db_instance.default: Still creating... [8m40s elapsed]
aws_db_instance.default: Still creating... [8m50s elapsed]
aws_db_instance.default: Still creating... [9m0s elapsed]
aws_db_instance.default: Still creating... [9m10s elapsed]
aws_db_instance.default: Still creating... [9m20s elapsed]
aws_db_instance.default: Still creating... [9m30s elapsed]
aws_db_instance.default: Still creating... [9m40s elapsed]
aws_db_instance.default: Still creating... [9m50s elapsed]
aws_db_instance.default: Still creating... [10m0s elapsed]
aws_db_instance.default: Still creating... [10m10s elapsed]
aws_db_instance.default: Still creating... [10m20s elapsed]
aws_db_instance.default: Still creating... [10m30s elapsed]
aws_db_instance.default: Still creating... [10m40s elapsed]
aws_db_instance.default: Still creating... [10m50s elapsed]
aws_db_instance.default: Still creating... [11m0s elapsed]
aws_db_instance.default: Still creating... [11m10s elapsed]
aws_db_instance.default: Still creating... [11m20s elapsed]
aws_db_instance.default: Still creating... [11m30s elapsed]
aws_db_instance.default: Still creating... [11m40s elapsed]
aws_db_instance.default: Still creating... [11m50s elapsed]
aws_db_instance.default: Still creating... [12m0s elapsed]
aws_db_instance.default: Creation complete after 12m10s [id=db-S7HPU4XLHCNJRMNWLUGC7RQSXQ]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
lb_dns_name = "external-LB-1476713536.us-east-1.elb.amazonaws.com"
[ec2-user@ip-172-31-95-52 ~]$
Search
ENG
IN
05:02 PM
27-09-2024

Verify the resources

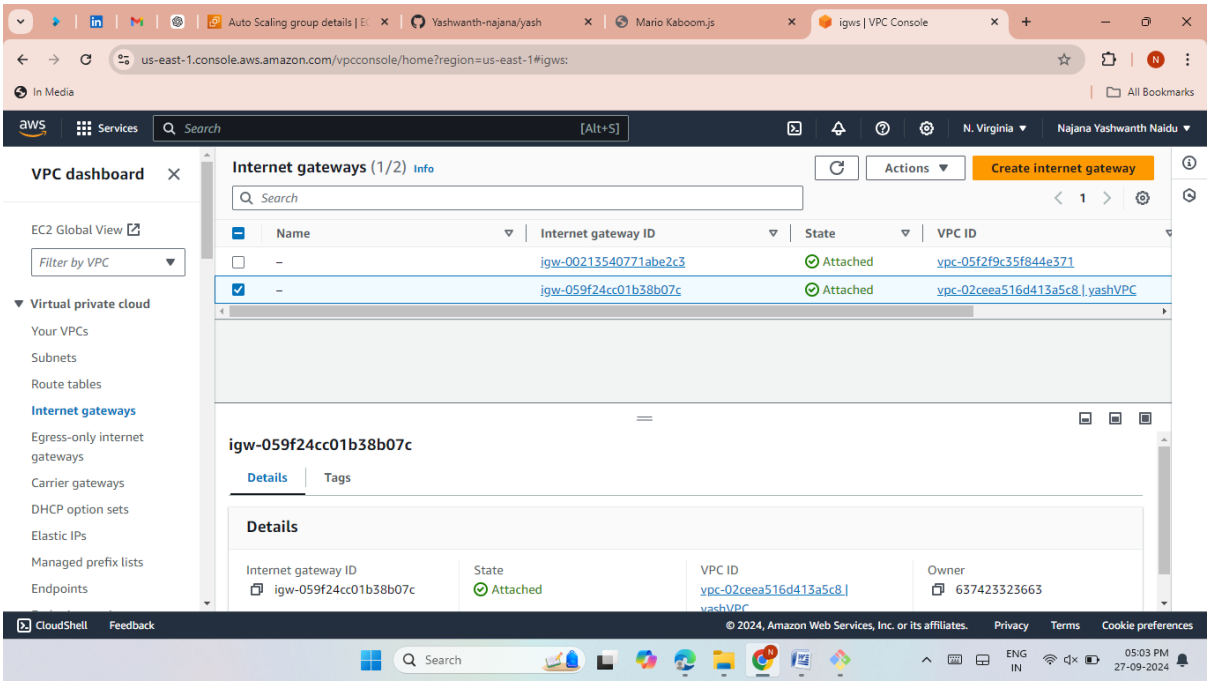
- Terraform will create all resources.
- Here the VPC was created.



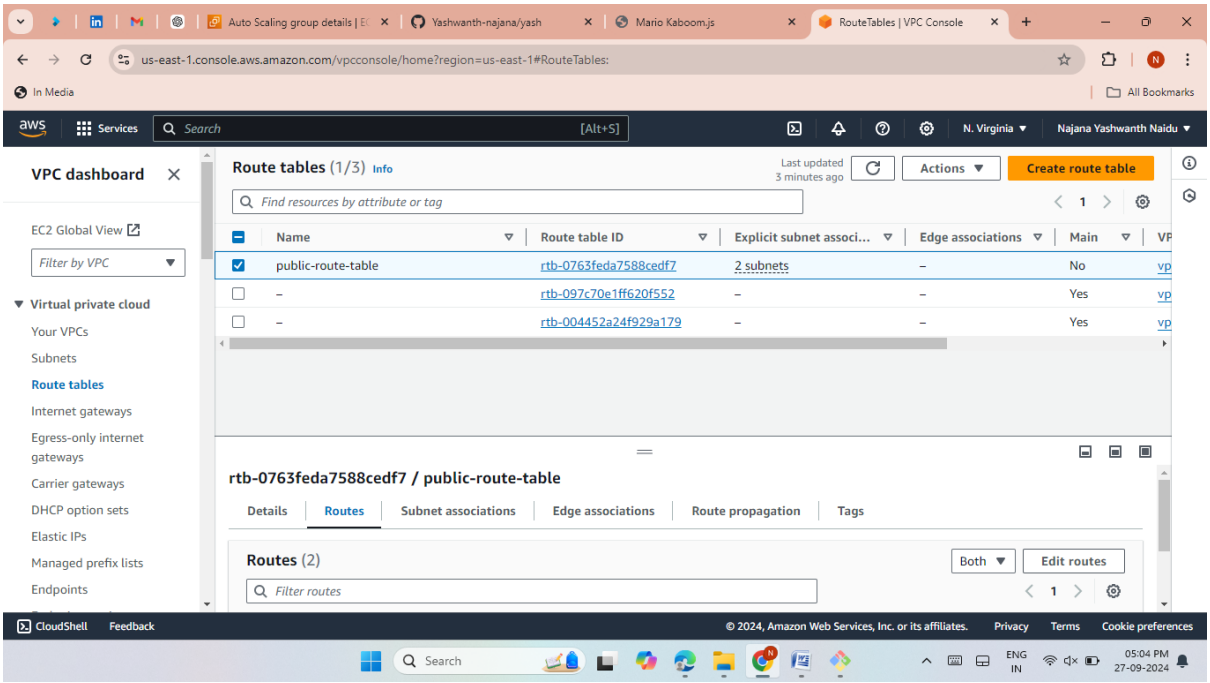
- Here Four public and two private subnets was created.



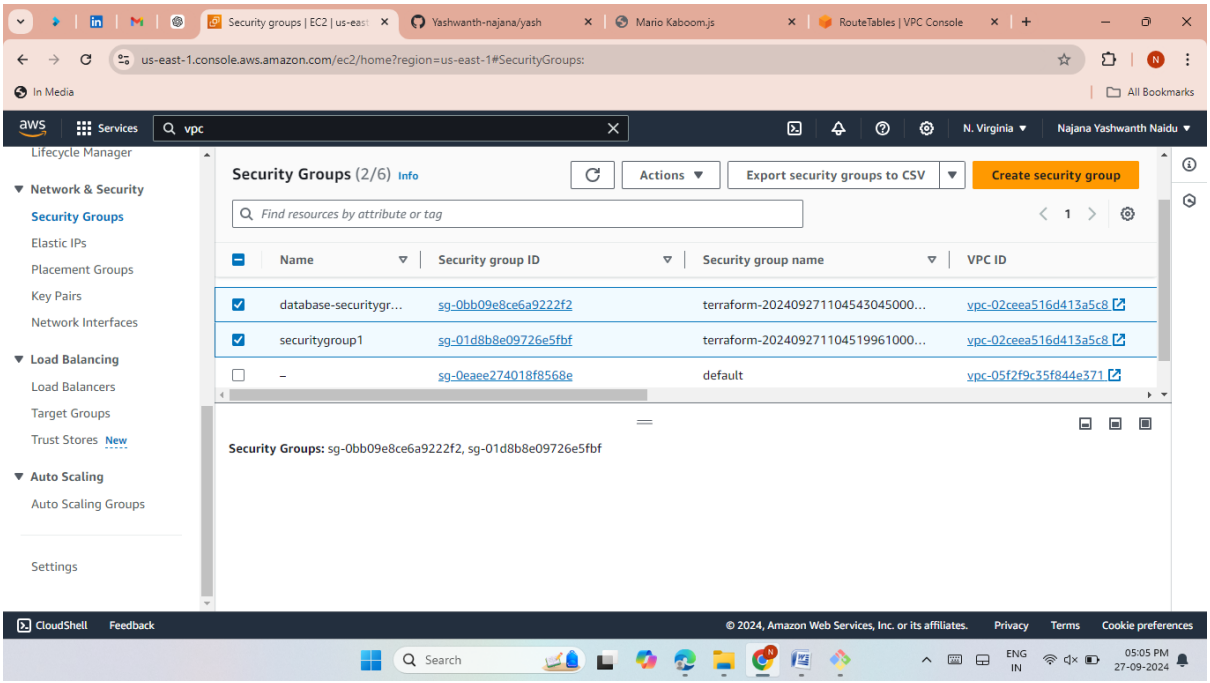
➢ Here Internet gateway was created.



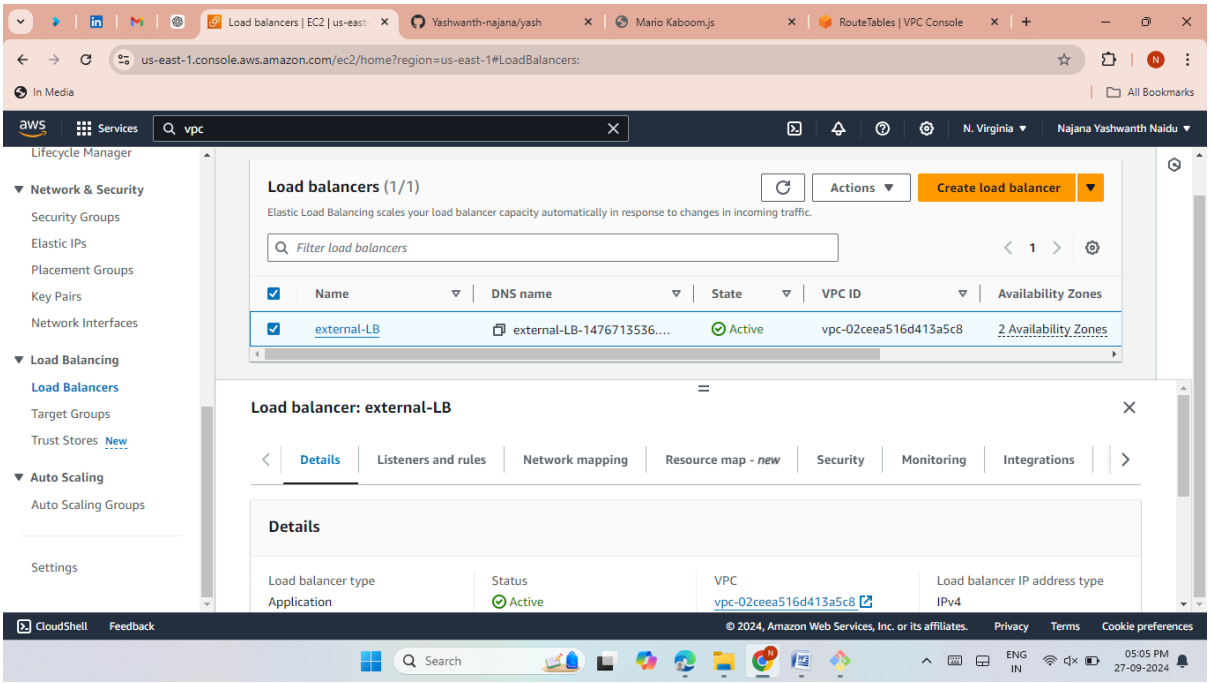
➢ Here the route table was created.

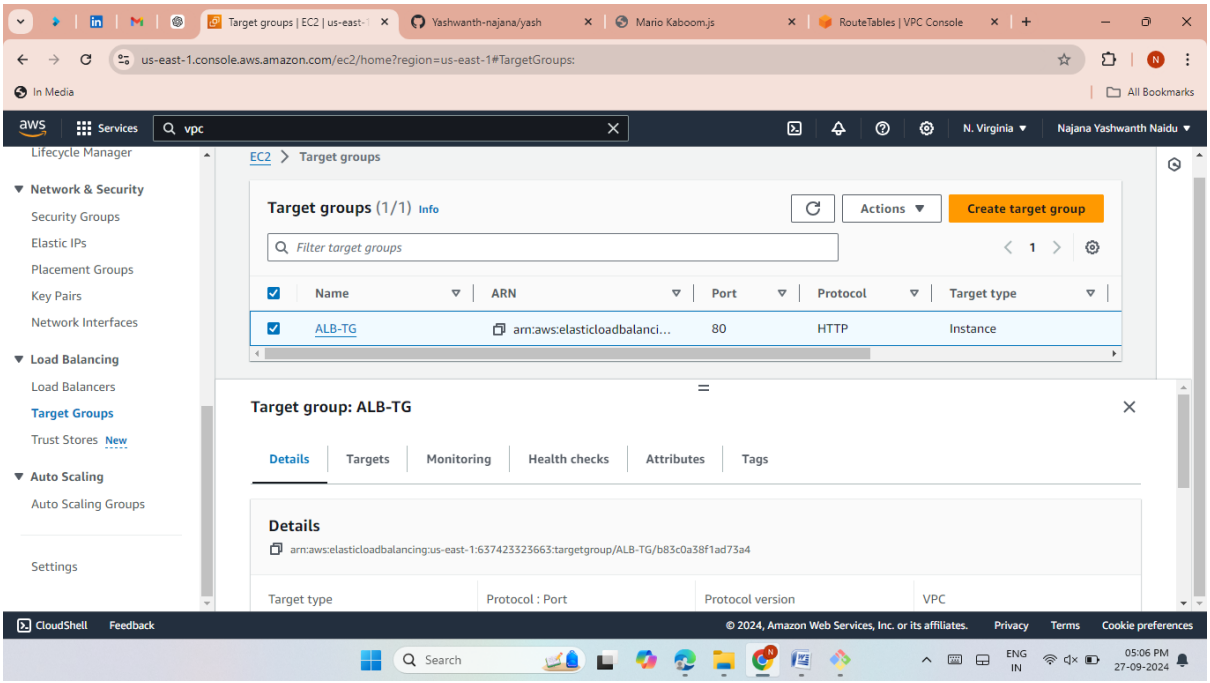


- Here we can see the two securitygroups was created.

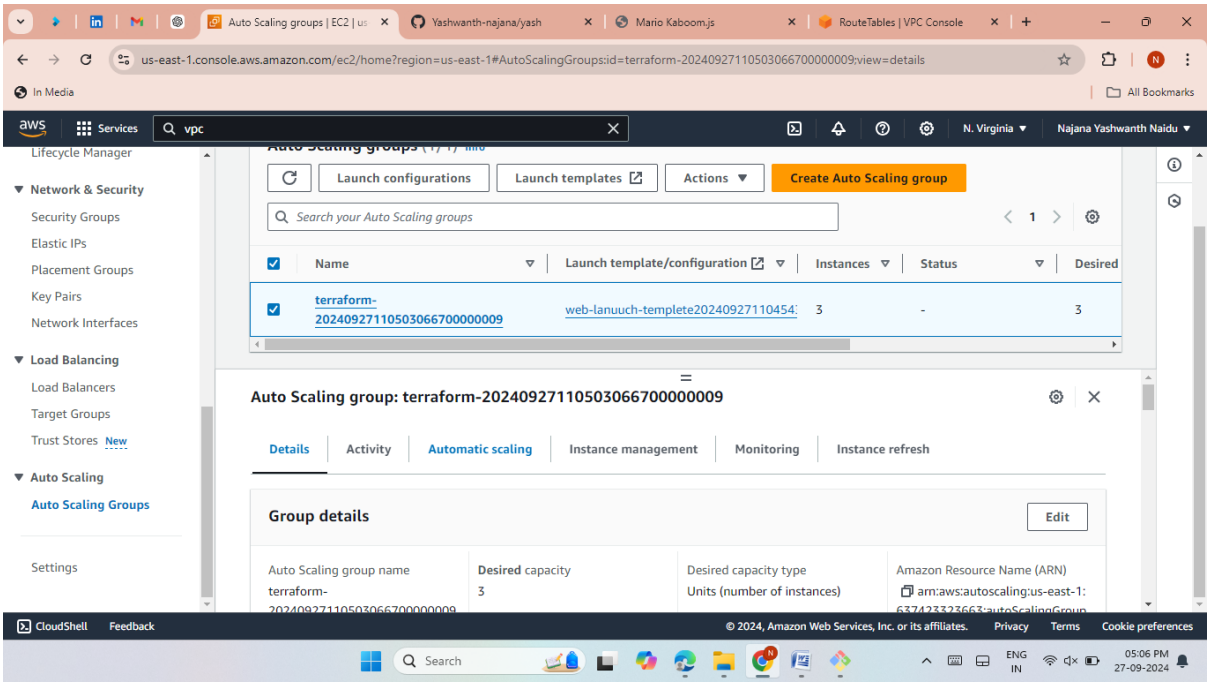


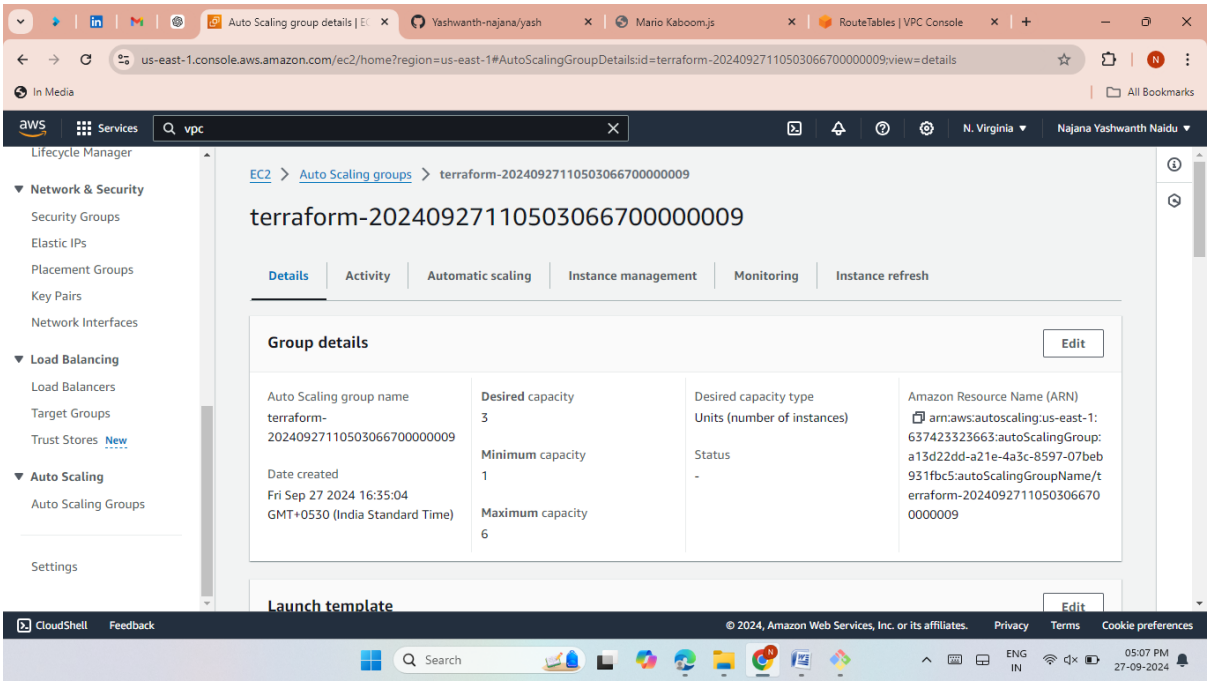
- Here the Load Balancer and Target Group was created.



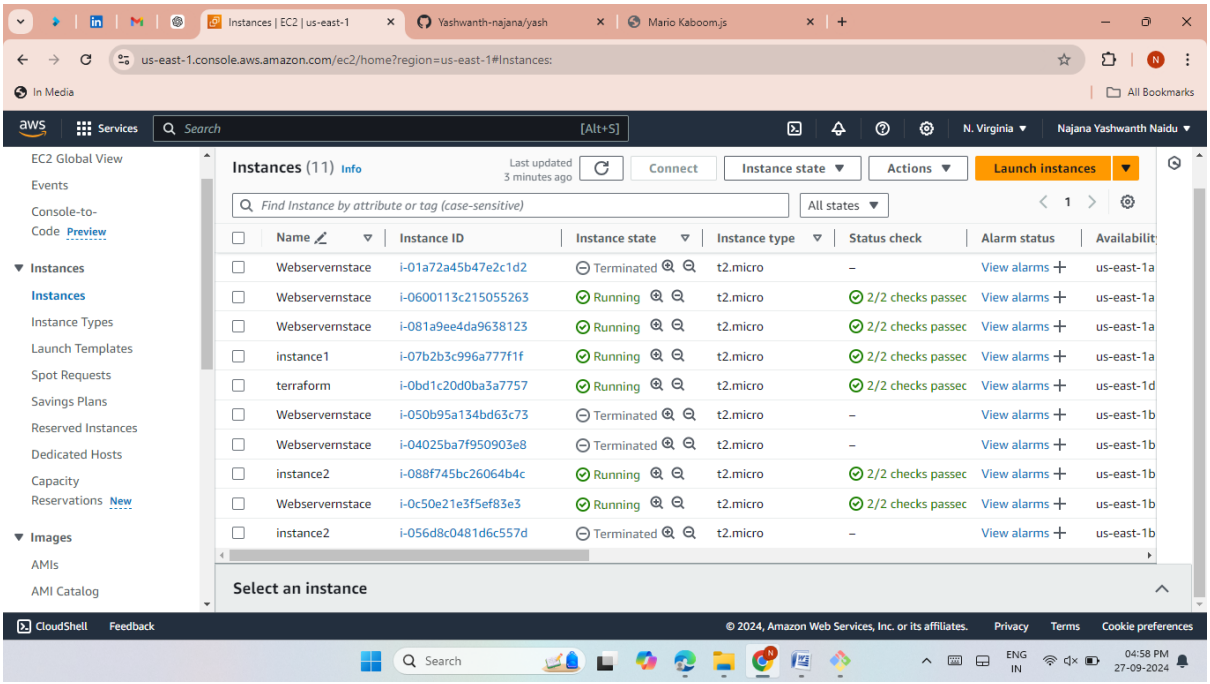


➢ Here the Auto Scaling Group was created.

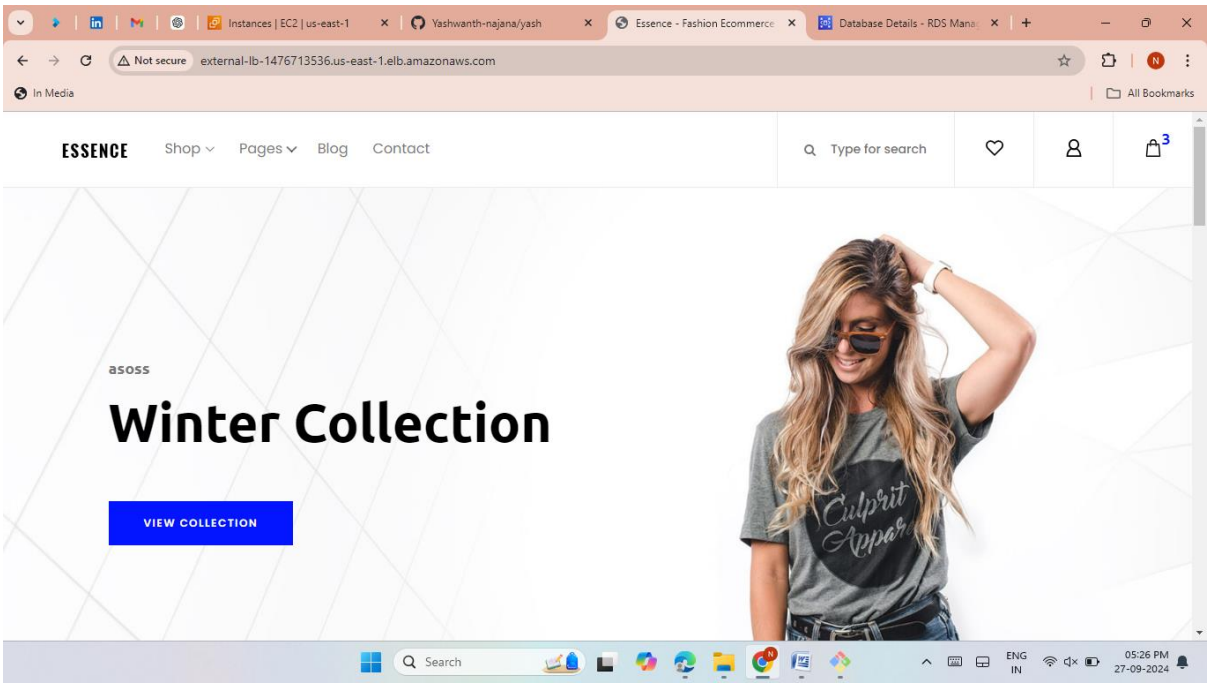


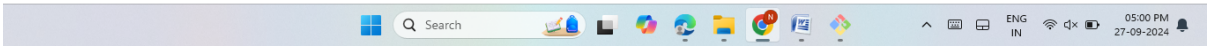


- Instance1 and Instance2 are created instances. Remaining instances are created by the Auto Scaling Group.

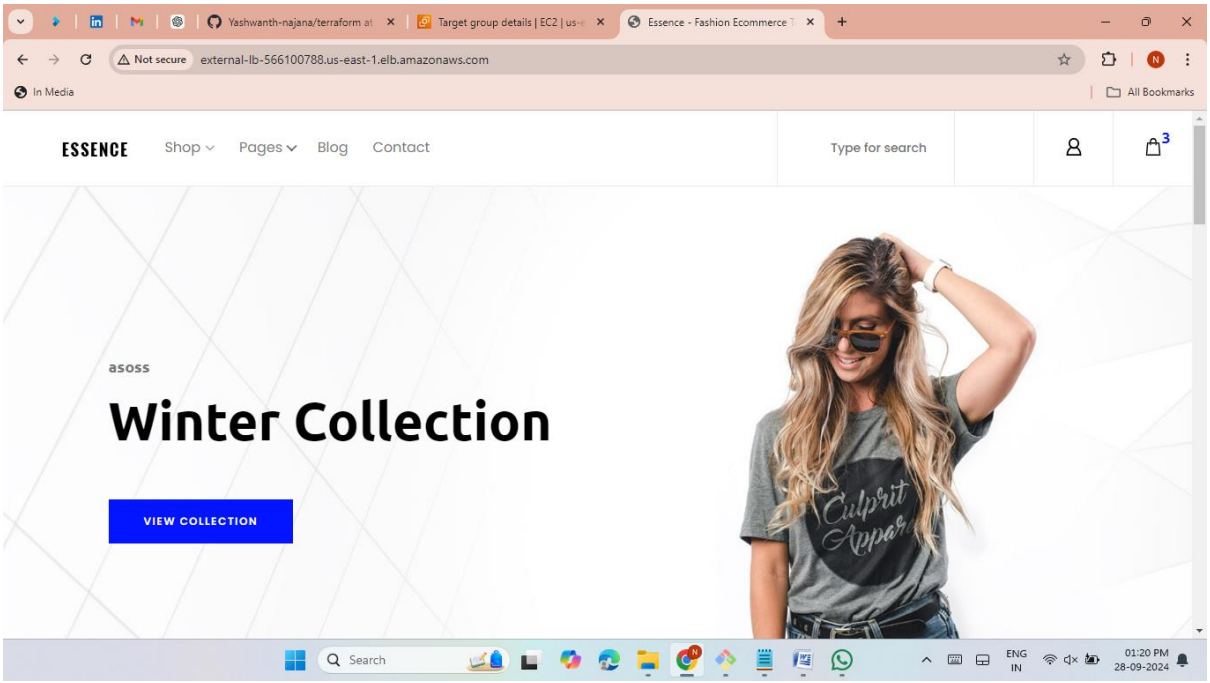


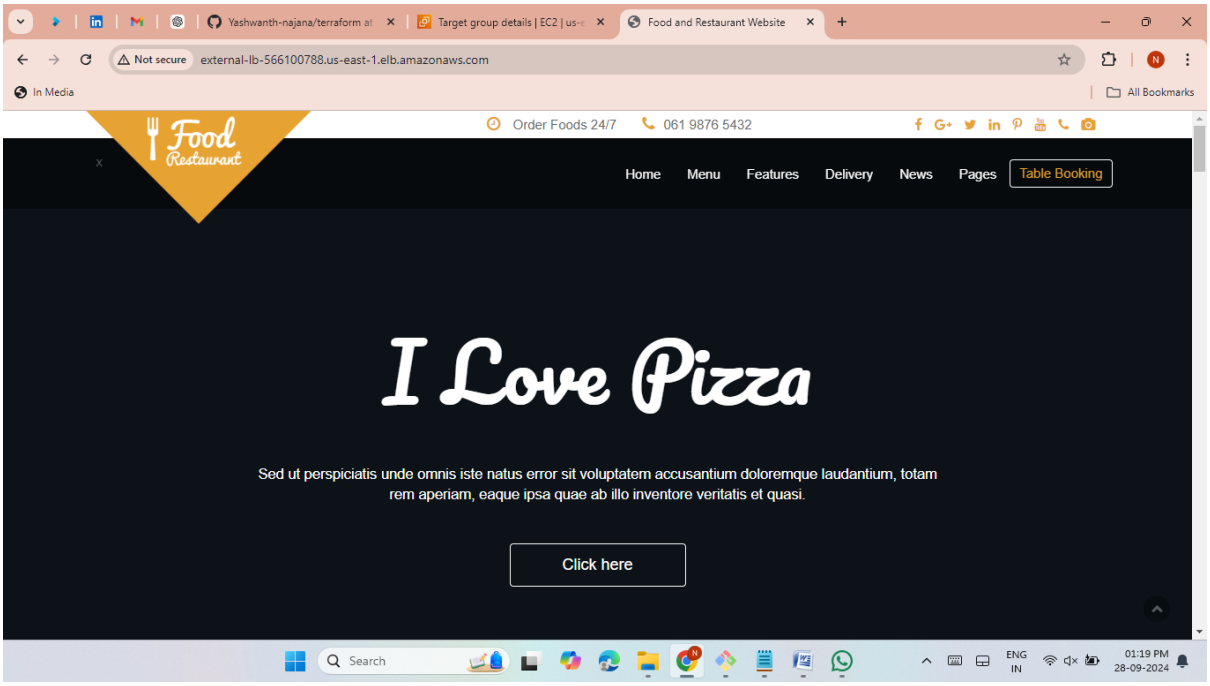
- Once the Resource creation finishes we can get the DNS of a load balancer and paste it into the browser you can see Load Balancer will send the request to two instances.
- Hosting and balancing the load with help of load balancer DNS.
- We can see the two websites in single DNS link.



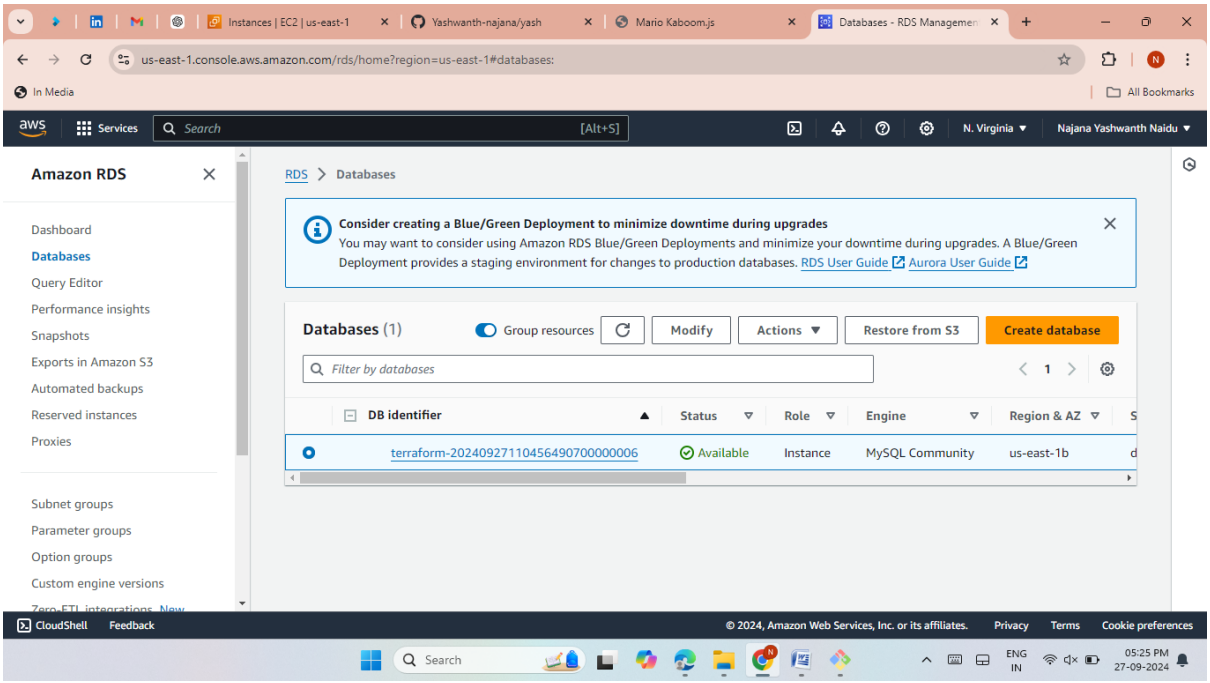


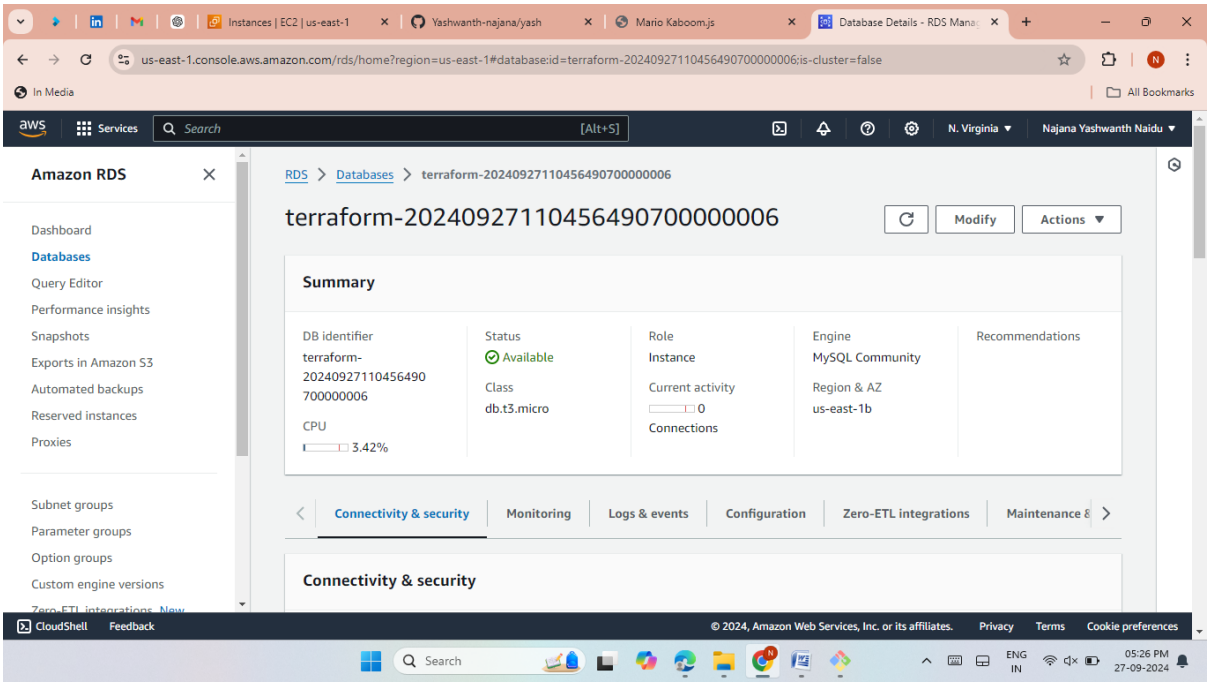
- Here I am hosting ecomm and food websites also.





- Here the RDS Database was created.





- Finally we can destroy all the resources with the single command **terraform destroy.**
- The terraform destroy command is used to completely remove all resources that were previously created and managed by Terraform in your infrastructure.