Pre-Requisites:

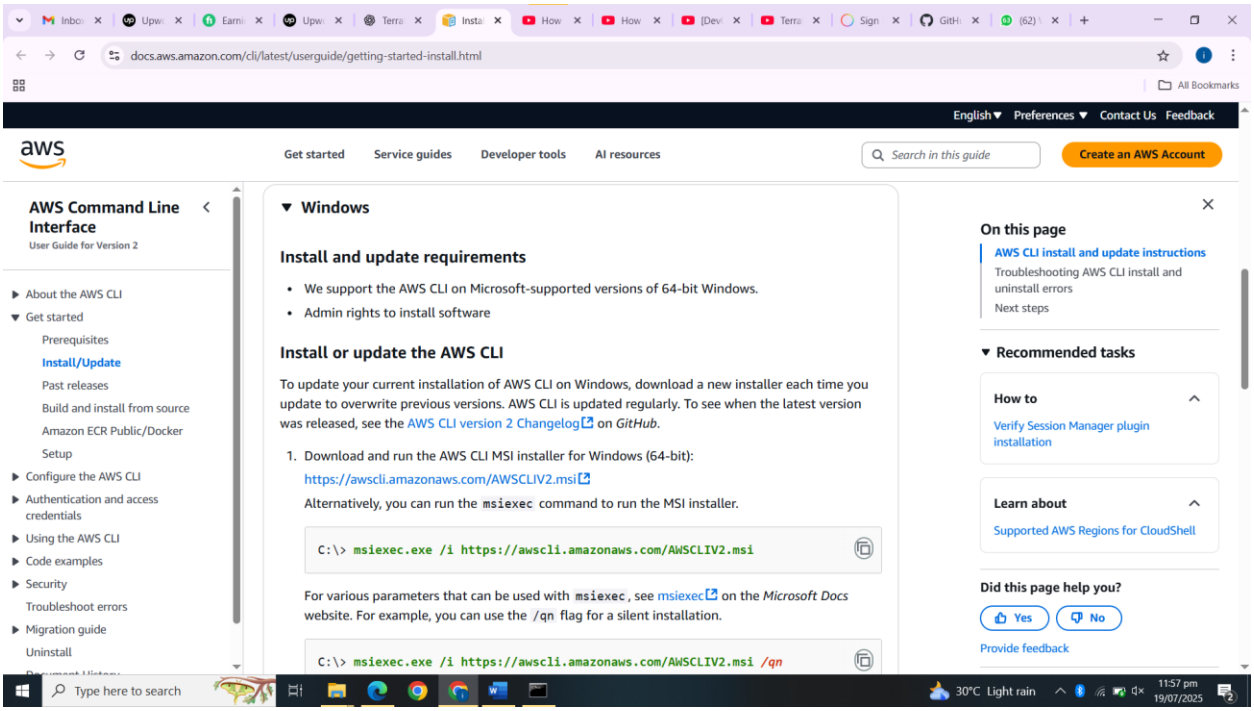
Git bash, node, npm, docker desktop already installed

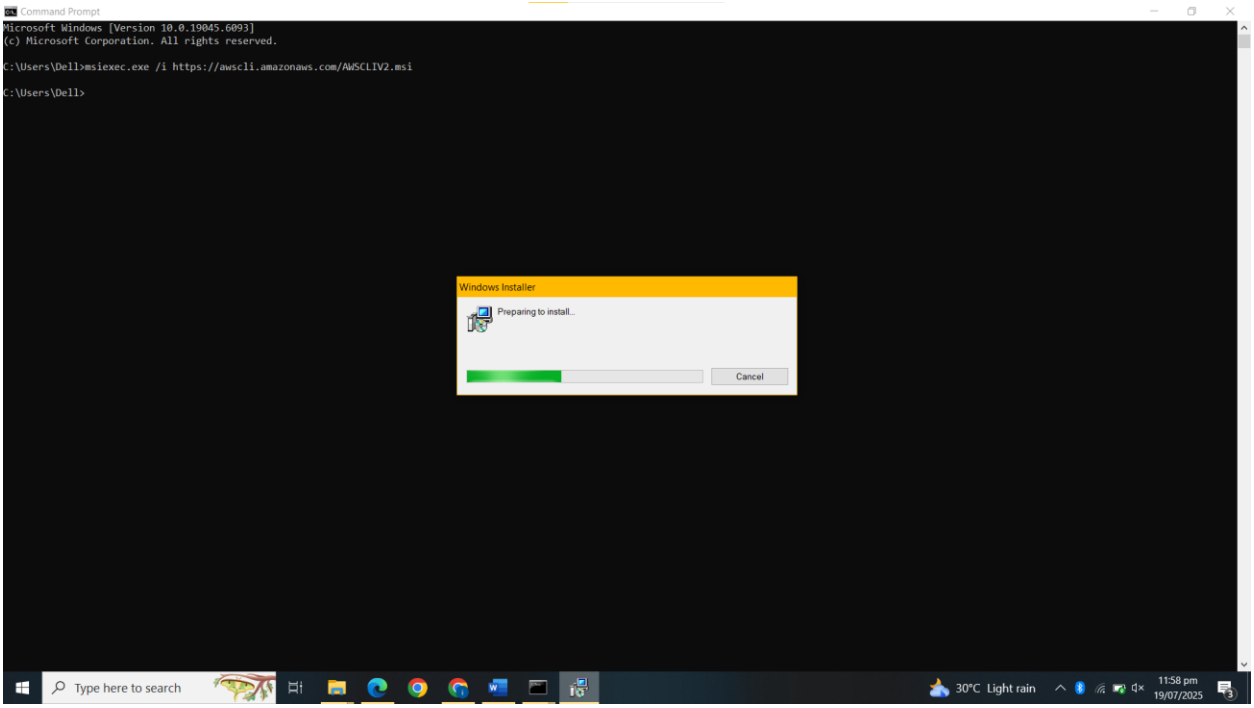
TASK#1:

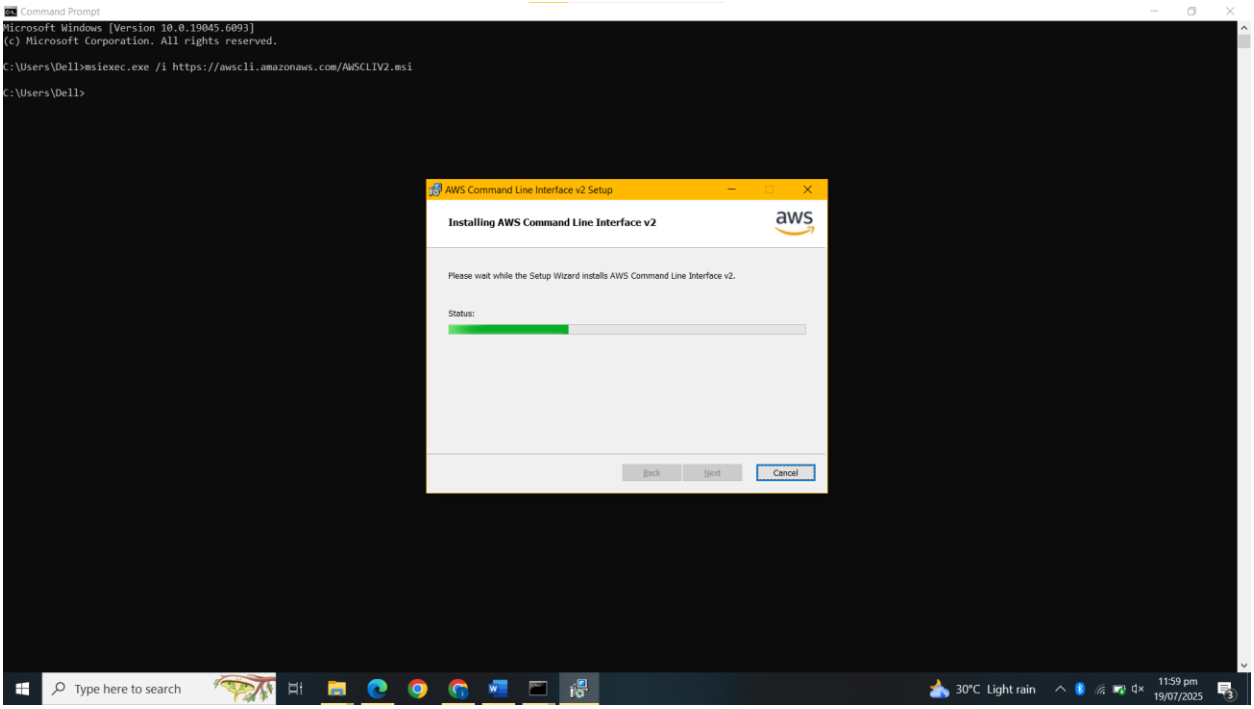
Task 1: Get Started with Terraform and AWS EC2
Goal: Set up Terraform and provision a basic EC2 instance.
Why: This helps you understand the core Terraform workflow and the AWS provider.
What You'll Learn: Installing Terraform, configuring AWS CLI, writing your first .tf file, and connecting to EC2.
• Install Terraform on your local machine.
• Set up AWS CLI with IAM credentials (learn about access keys).
• Create a Terraform script to launch a t2.micro EC2 instance.
• Use Terraform outputs to display the public IP.
• SSH into the instance and install Nginx to verify connectivity.

installing aws cli

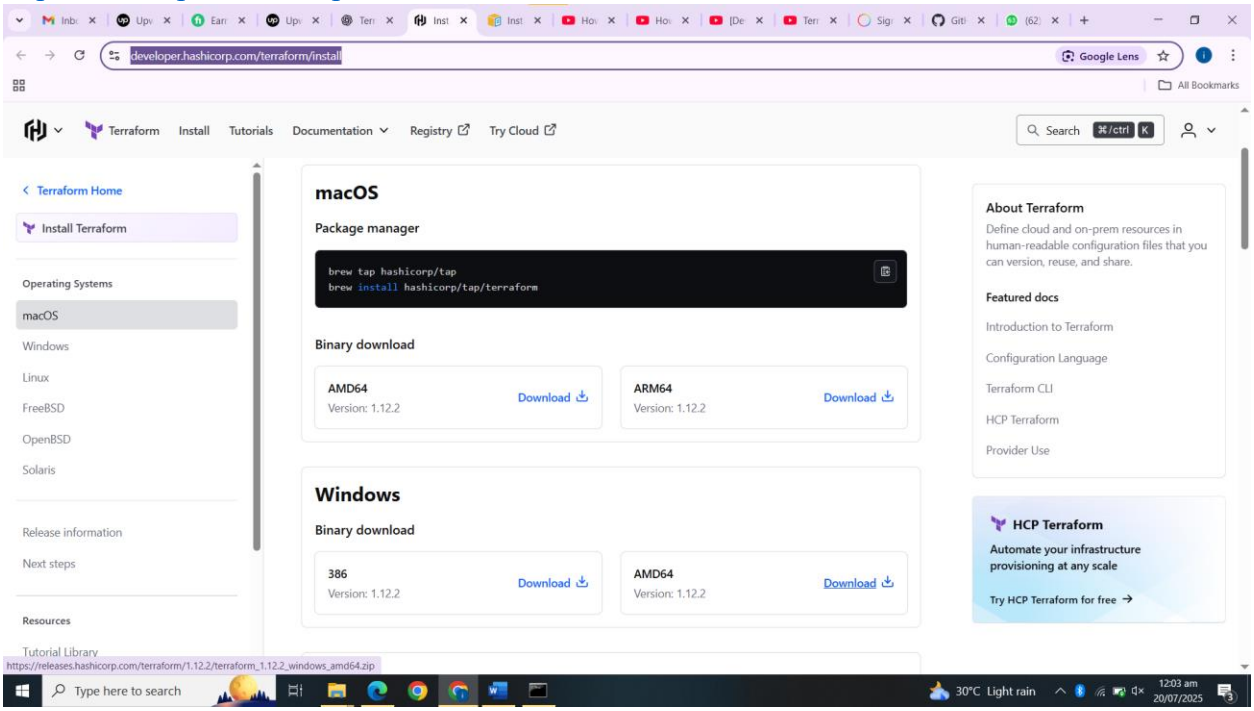
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html



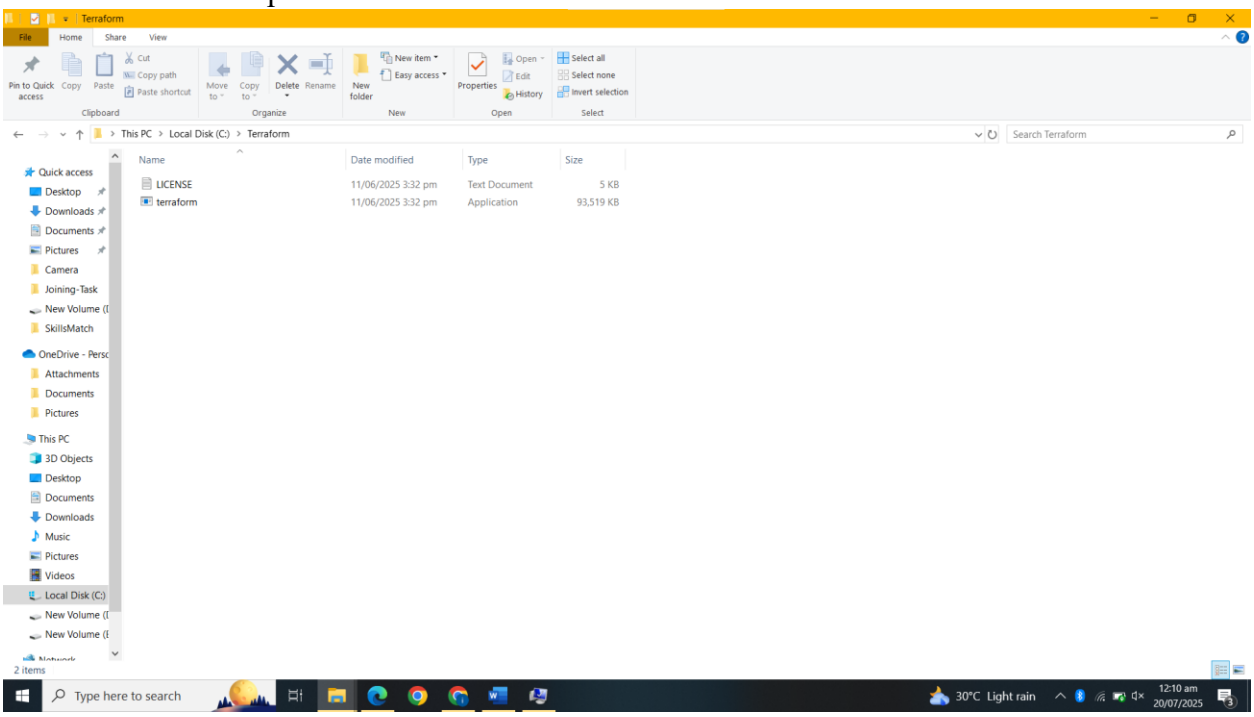

Command Prompt
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Dell>msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
C:\Users\Dell>
Windows Installer
Preparing to install...
Cancel
Type here to search
30°C Light rain
11:58 pm
19/07/2025


Command Prompt
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Dell>msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
C:\Users\Dell>
AWS Command Line Interface v2 Setup
Installing AWS Command Line Interface v2
aws
Please wait while the Setup Wizard installs AWS Command Line Interface v2.
Status:
Back
Next
Cancel
Type here to search
30°C Light rain
11:59 pm
19/07/2025

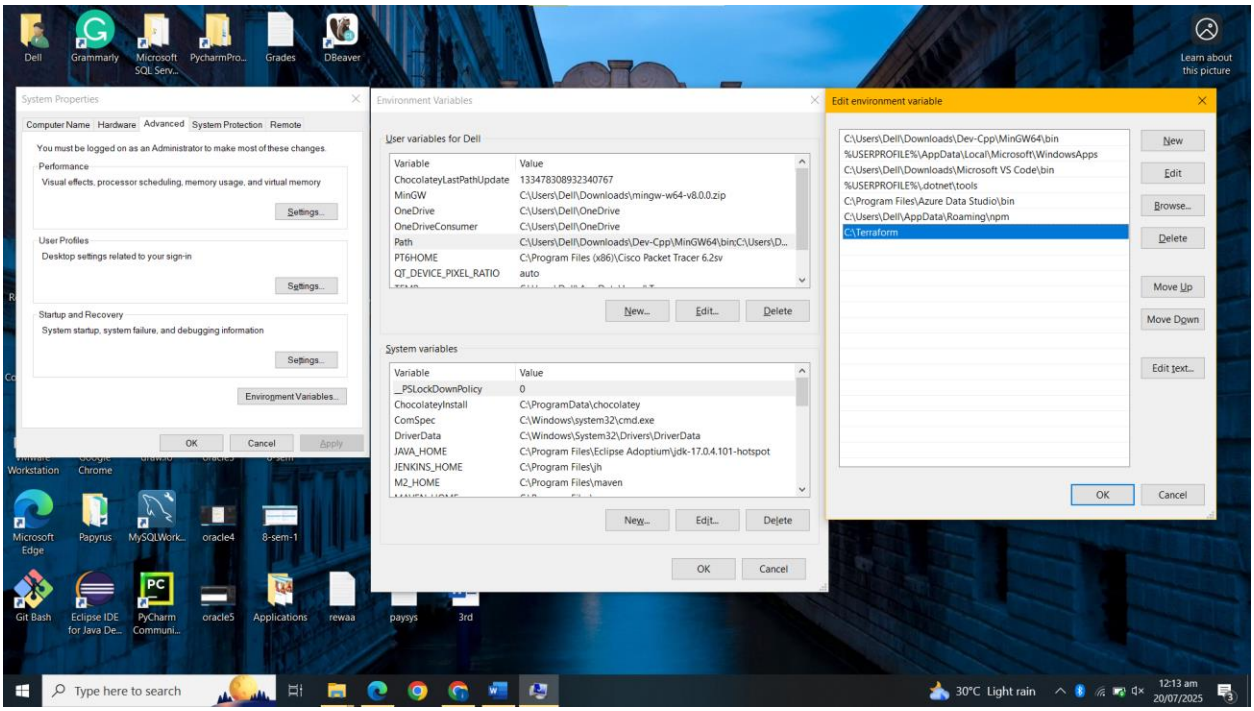
Download terraform
https://developer.hashicorp.com/terraform/install



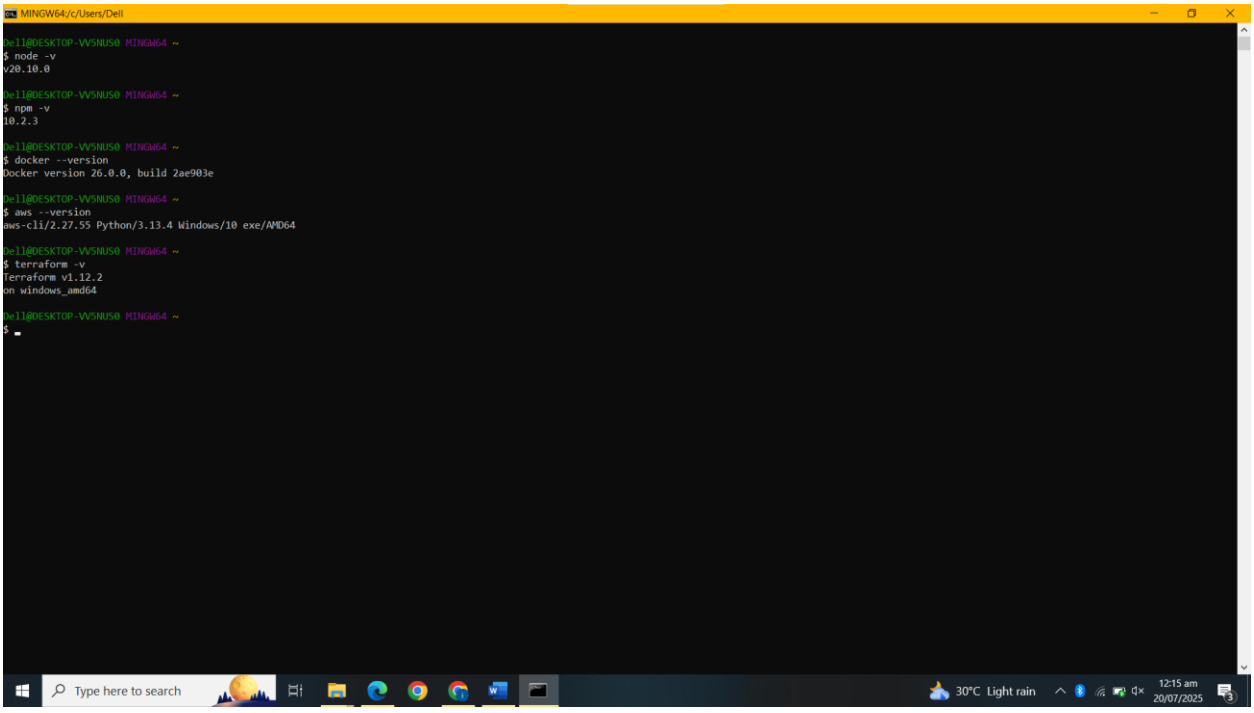
Extract the files and put it in terraform folder like this in the c drive:



Add it in the environment variables in the path



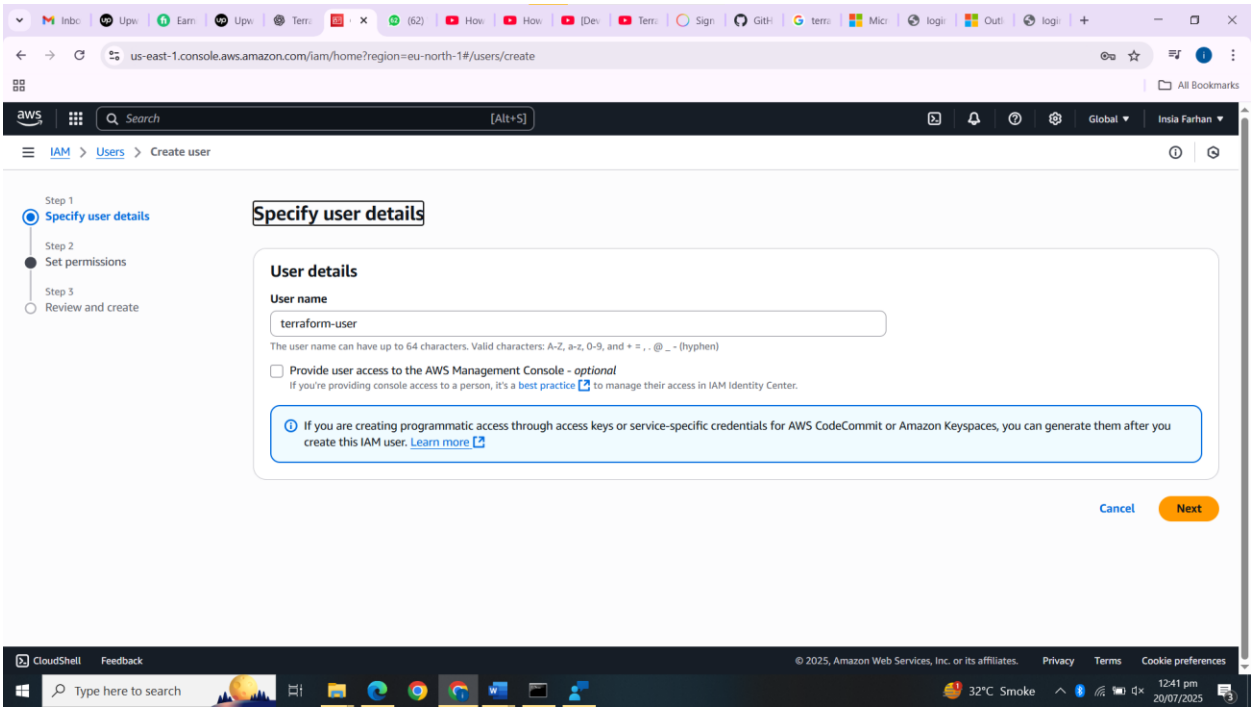
Ensure version of everything on git bash

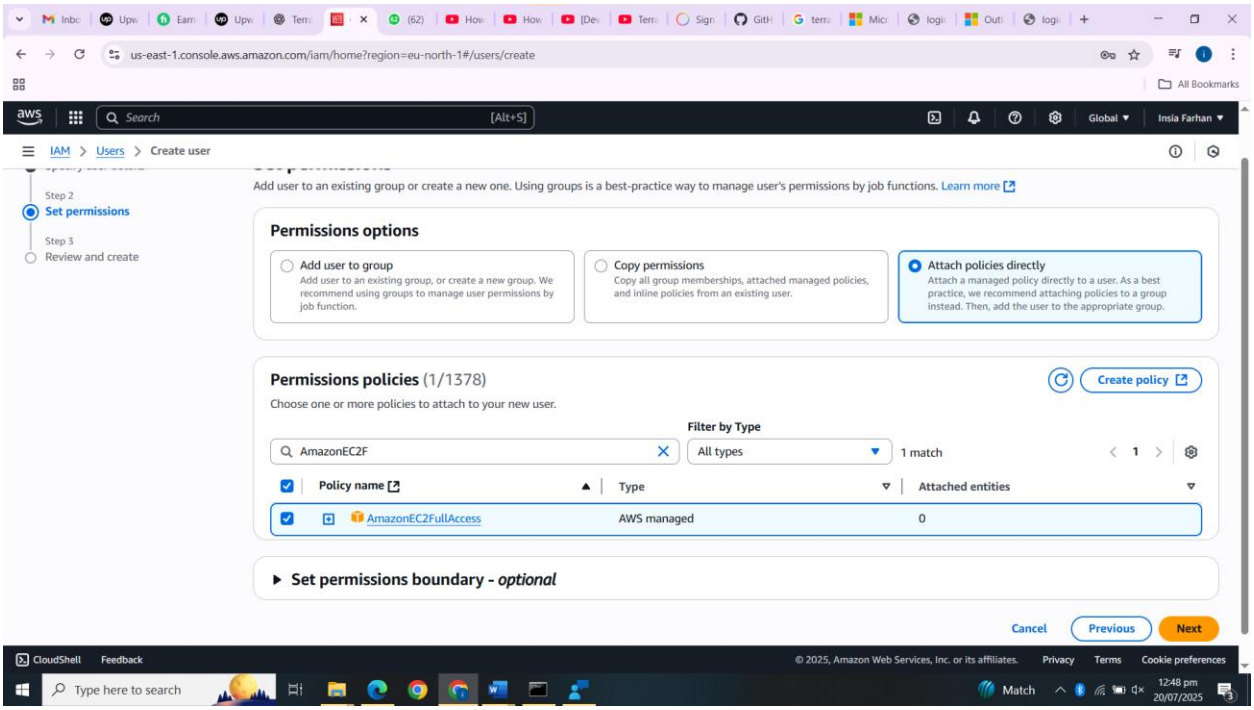


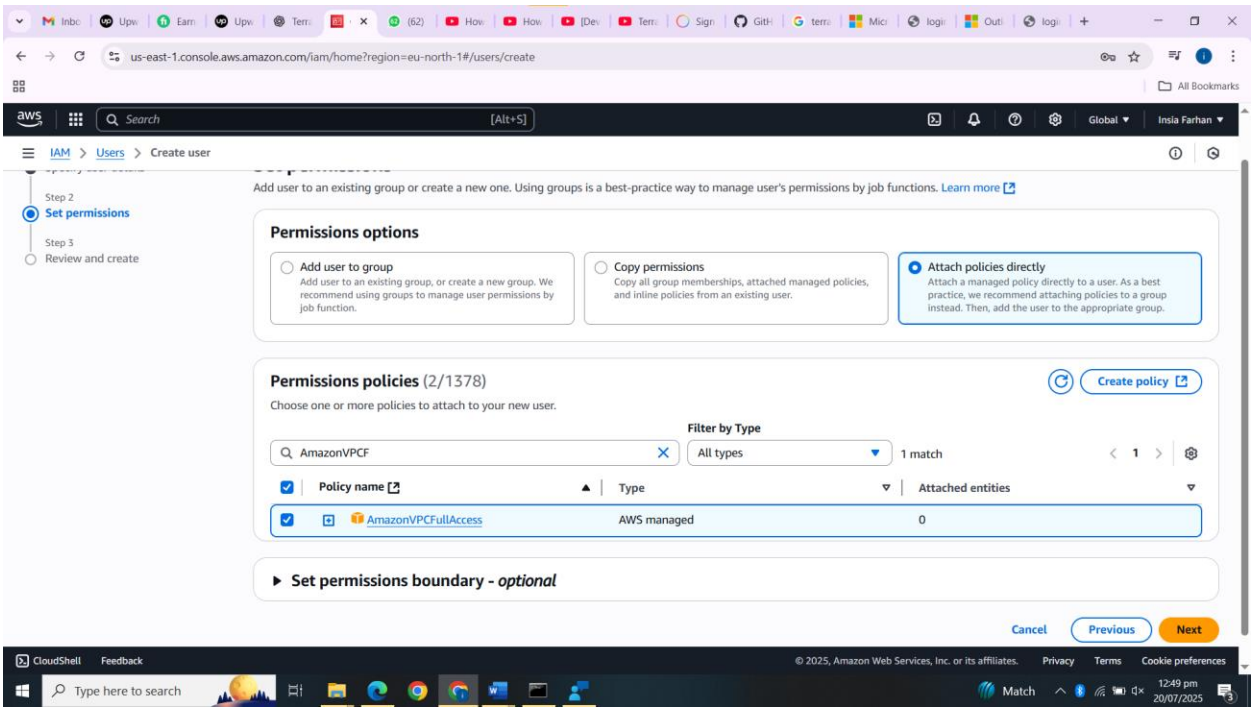
Create an AWS account and sign it:

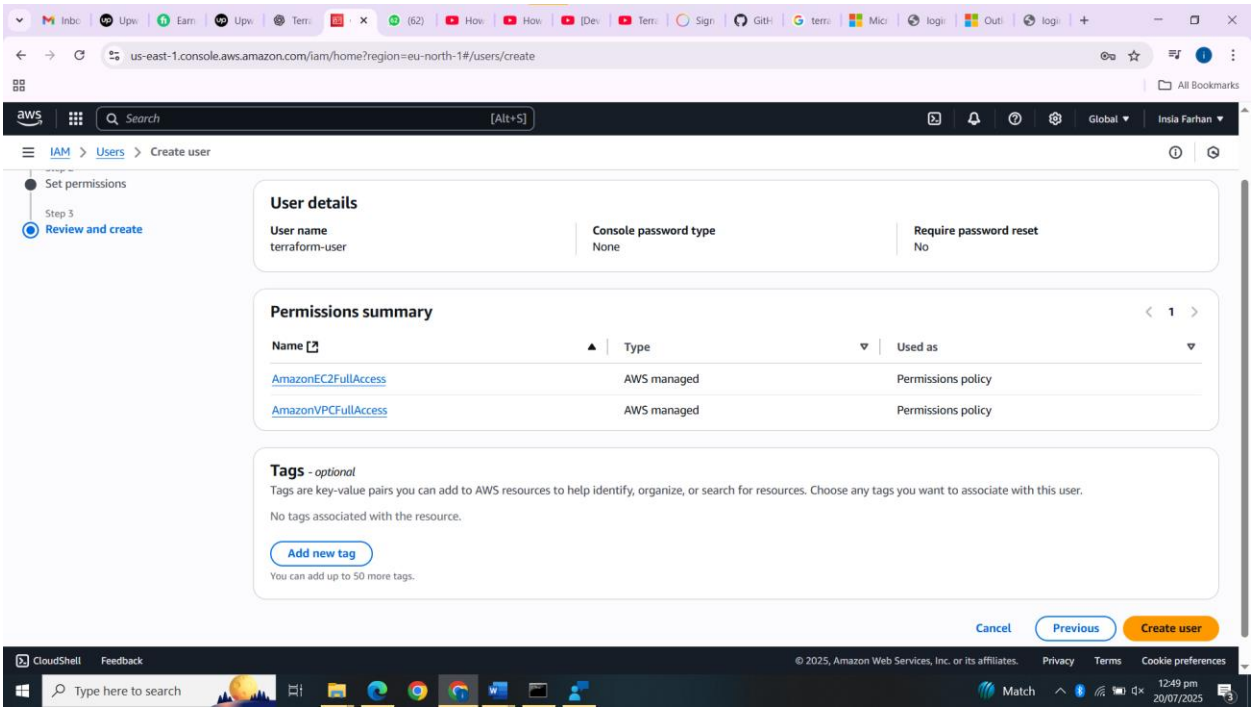
https://aws.amazon.com/

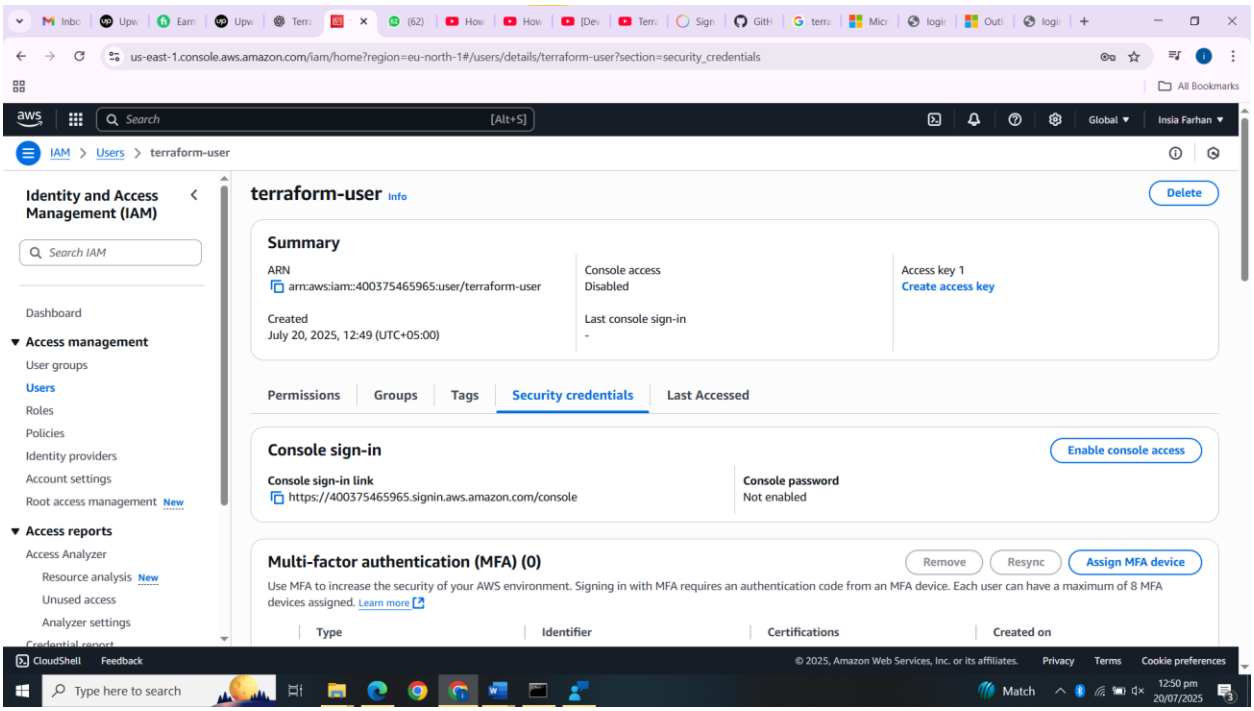
Create IAM User

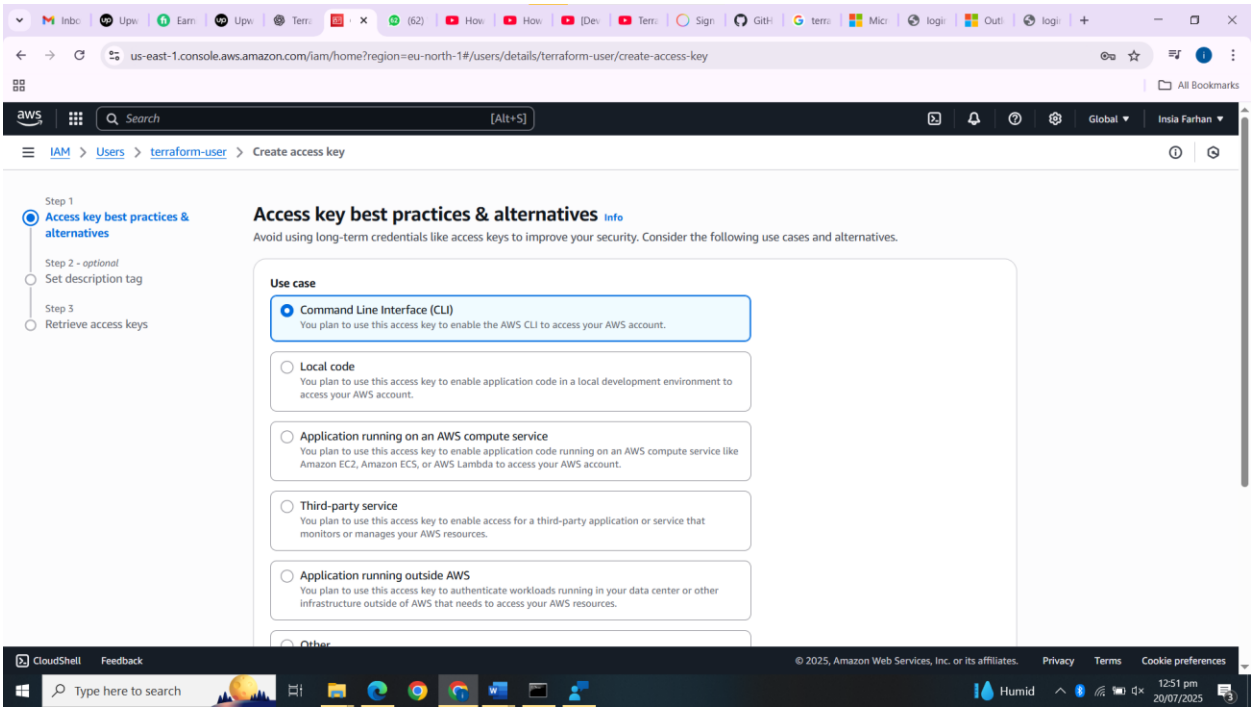


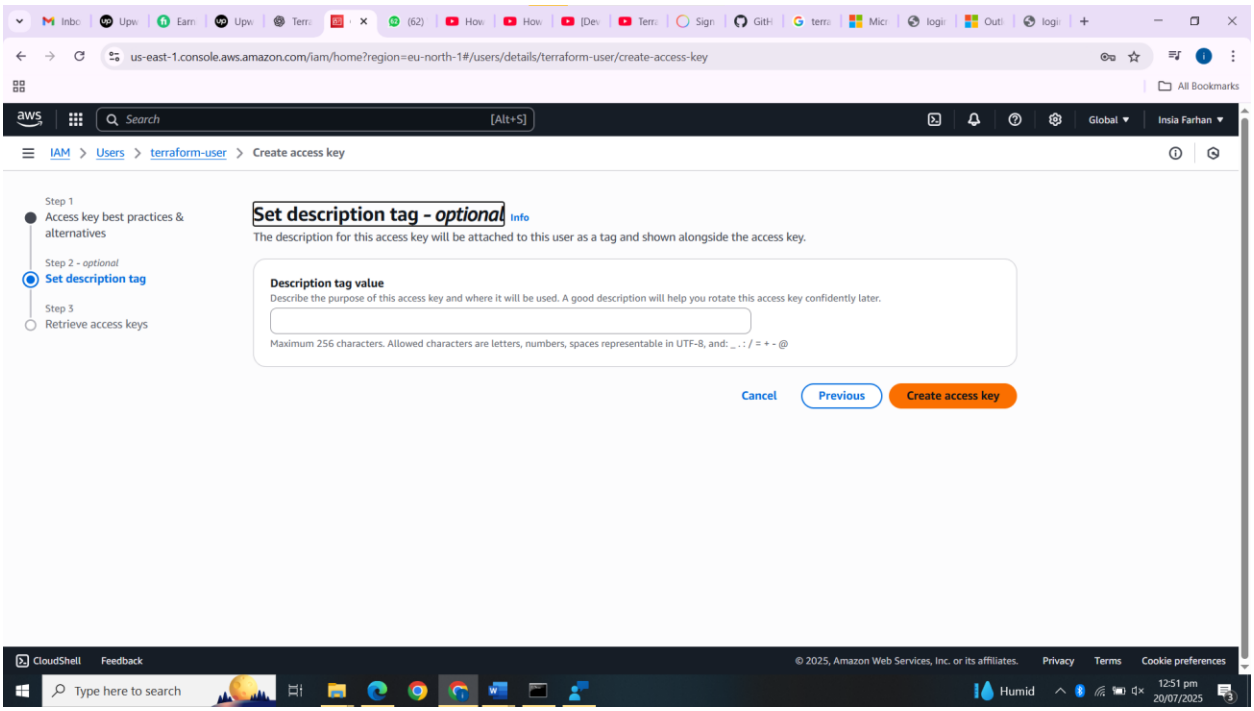


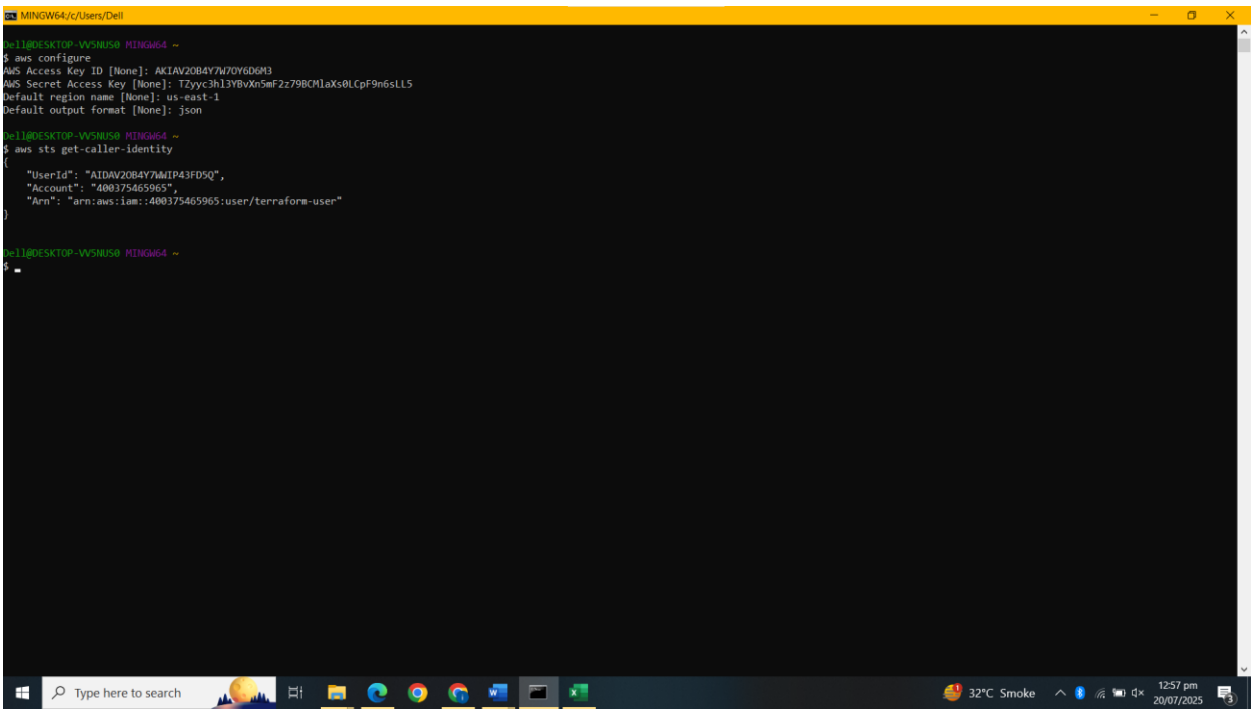

us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/create
Search
[Alt+S]
Global
Insia Farhan
IAM > Users > Create user
Step 2
Set permissions
Step 3
Review and create
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (2/1378)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
AmazonVPCF
All types
1 match
Policy name
Type
Attached entities
AmazonVPCFullAccess
AWS managed
0
Set permissions boundary - optional
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Match
12:49 pm
20/07/2025


us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/create
Search
[Alt+S]
Global
Insia Farhan
IAM > Users > Create user
Set permissions
Step 3
Review and create
User details
User name
terraform-user
Console password type
None
Require password reset
No
Permissions summary
Name
Type
Used as
AmazonEC2FullAccess
AWS managed
Permissions policy
AmazonVPCFullAccess
AWS managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Match
12:49 pm
20/07/2025


us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/terraform-user?section=security_credentials
Search [Alt+S]
Global
Insia Farhan
IAM > Users > terraform-user
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
terraform-user Info
Delete
Summary
ARN
arn:aws:iam::400375465965:user/terraform-user
Console access
Disabled
Access key 1
Create access key
Created
July 20, 2025, 12:49 (UTC+05:00)
Last console sign-in
-
Permissions
Groups
Tags
Security credentials
Last Accessed
Console sign-in
Enable console access
Console sign-in link
https://400375465965.signin.aws.amazon.com/console
Console password
Not enabled
Multi-factor authentication (MFA) (0)
Remove
Resync
Assign MFA device
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more
Type
Identifier
Certifications
Created on
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Match
12:50 pm
20/07/2025

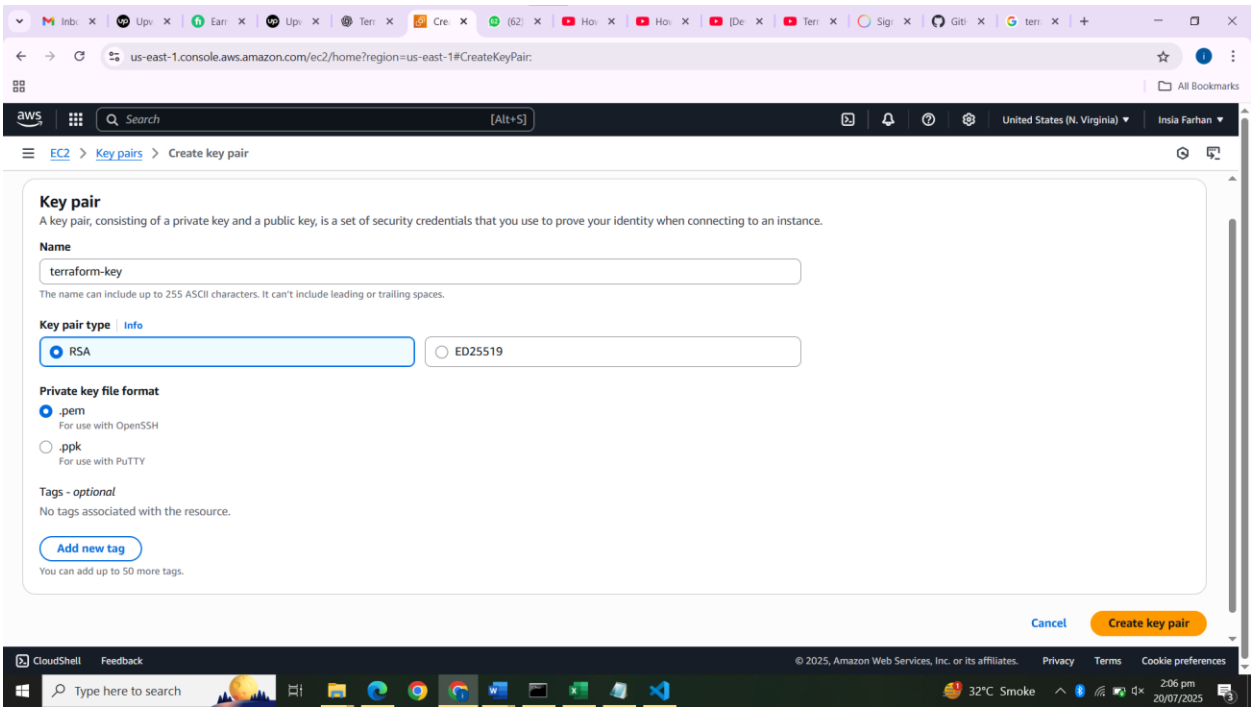

us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/terraform-user/create-access-key
Search [Alt+S]
Global
Insia Farhan
IAM > Users > terraform-user > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Access key best practices & alternatives Info
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.
Use case
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
Other
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Humid
12:51 pm
20/07/2025



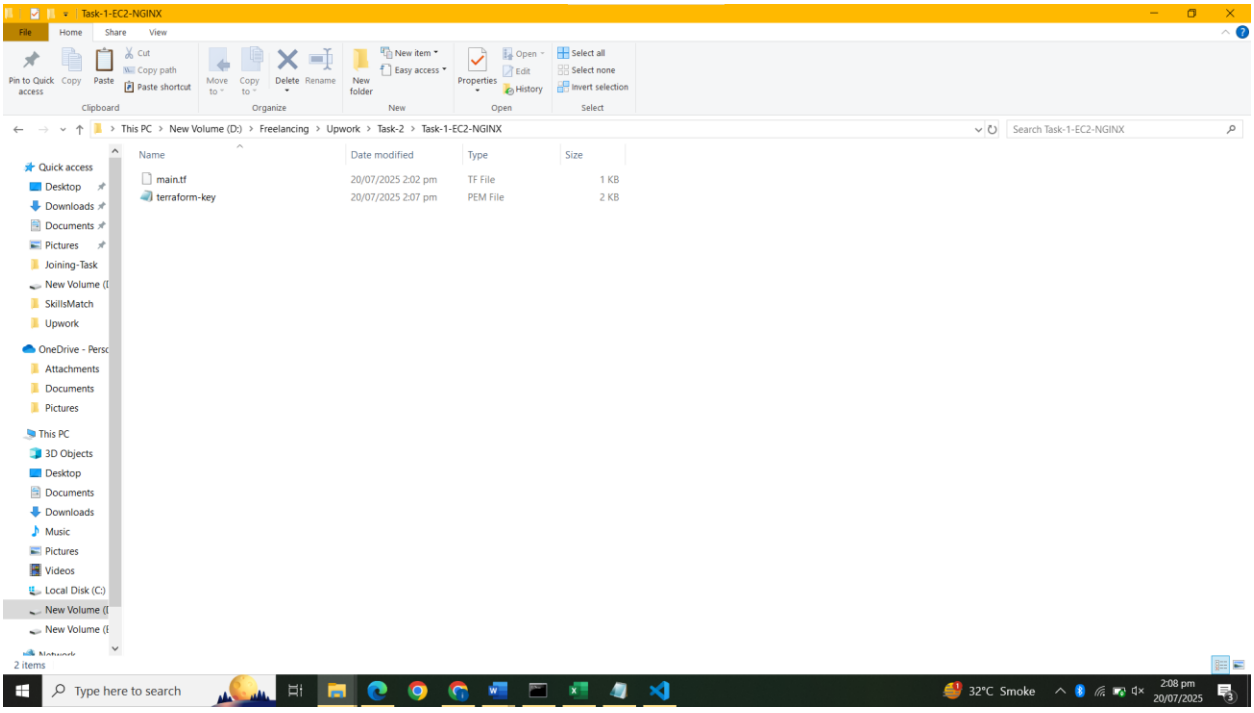


no error, it means **your AWS CLI is now connected to your account**!

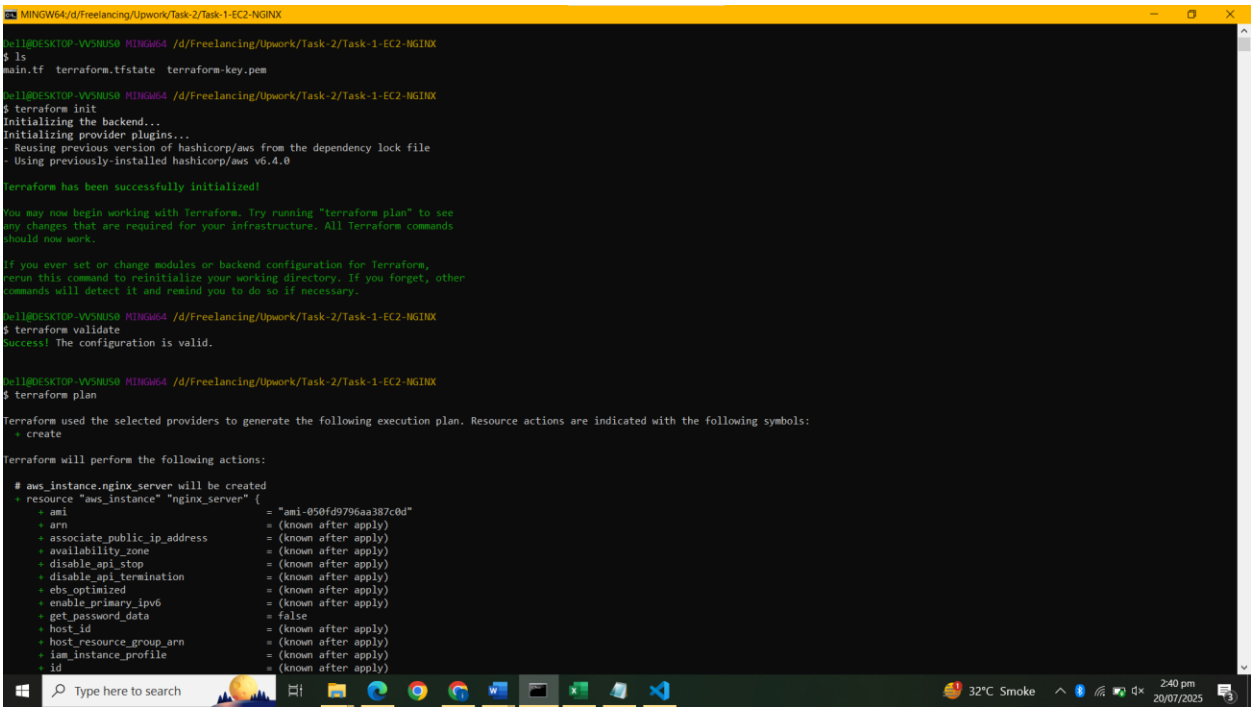
AWS CLI helps Terraform communicate with your AWS account using your IAM credentials.

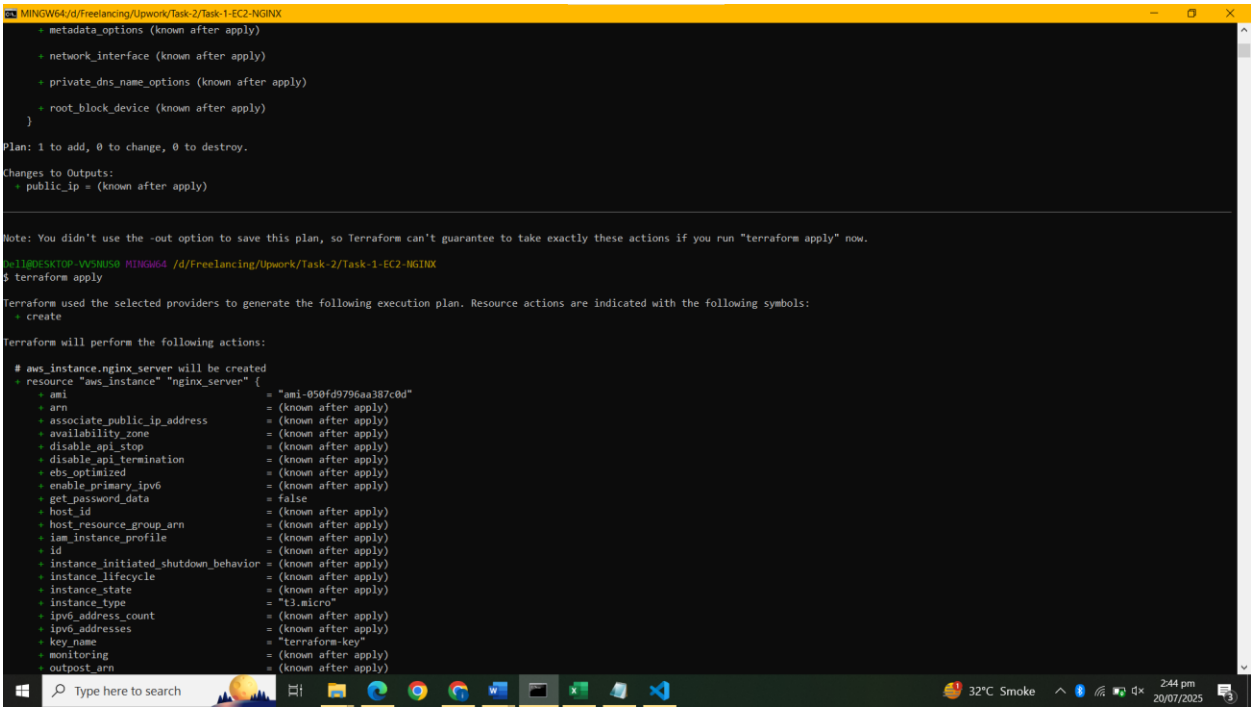


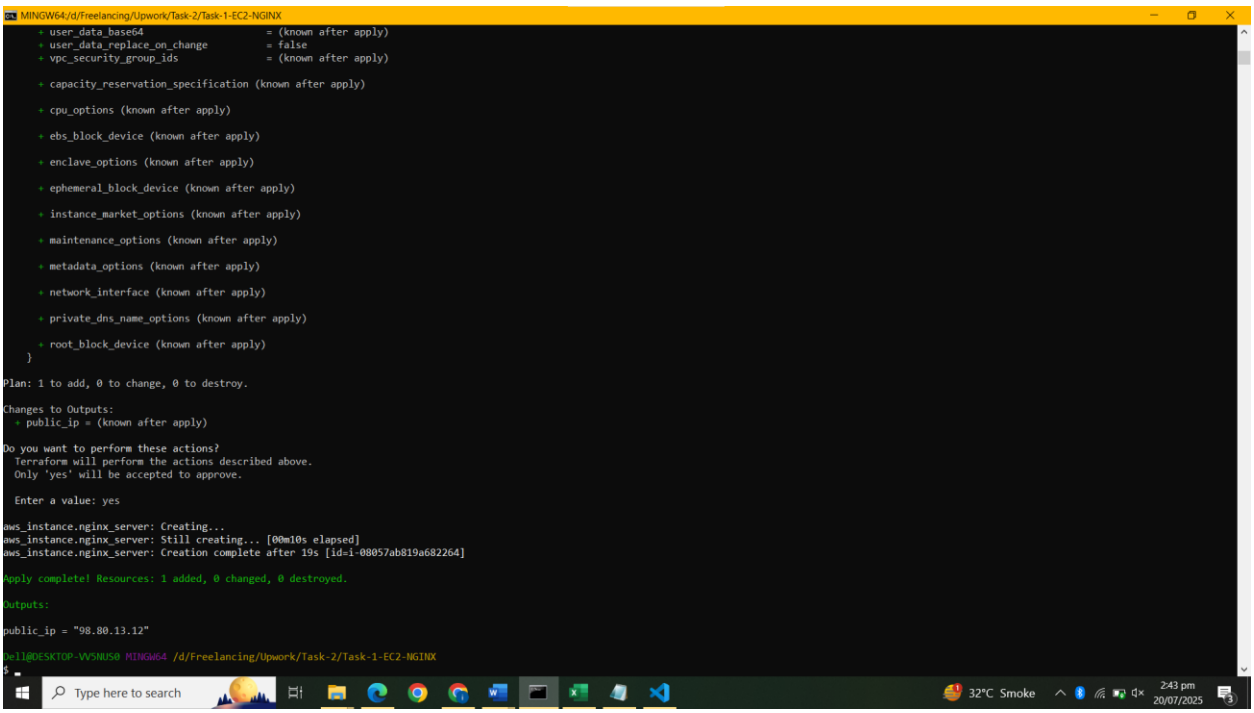
This .pem file allows you to **SSH (connect)** to the EC2 instance.

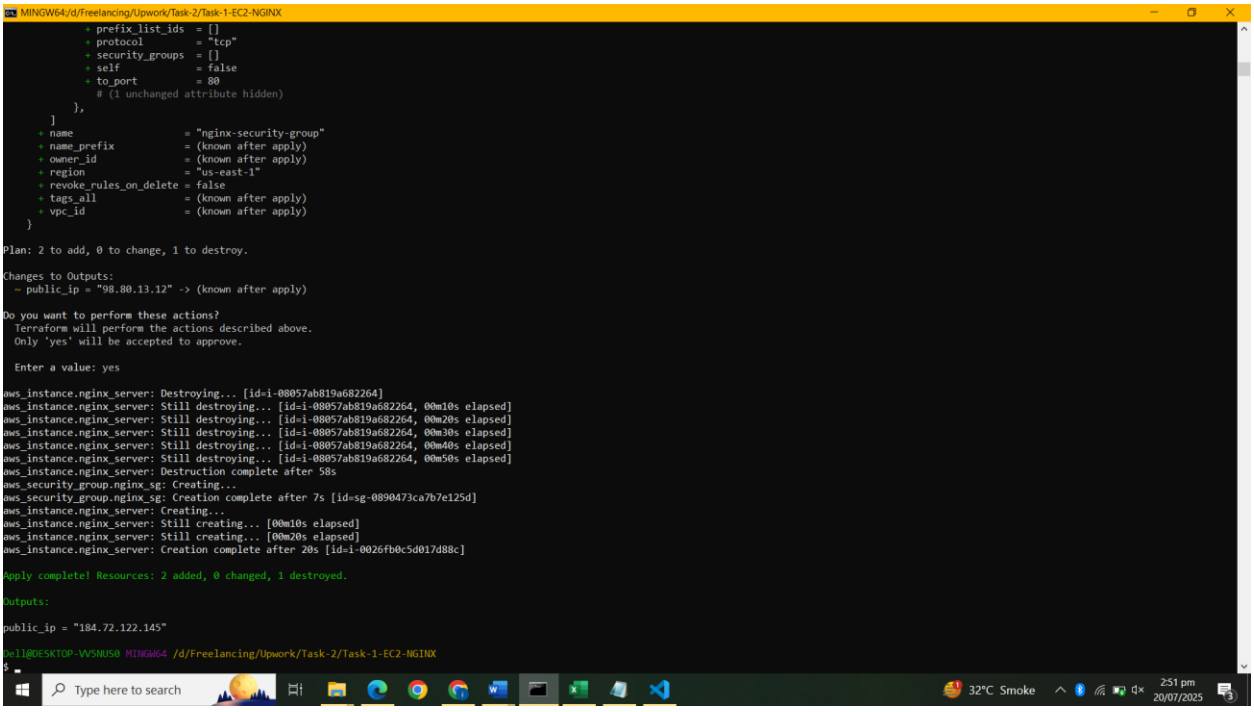


Use the accurate ami in the main.tf
checking with terraform plan to see a dry run, that is what will be applied

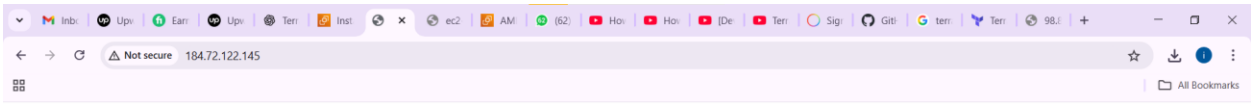

MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ ls
main.tf terraform.tfstate terraform-key.pem
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.4.0
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.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ terraform validate
Success! The configuration is valid.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ 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_instance.nginx_server will be created
+ resource "aws_instance" "nginx_server" {
+ ami = "ami-050fd9796aa387c0d"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (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)
Type here to search
32°C Smoke
2:40 pm
20/07/2025


MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
Plan: 1 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ public_ip = (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.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ 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_instance.nginx_server will be created
+ resource "aws_instance" "nginx_server" {
+ ami = "ami-050fd9796aa387c0d"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (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 = "t3.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "terraform-key"
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
Type here to search
32°C Smoke
2:44 pm
20/07/2025


MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
+ 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)
}
Plan: 1 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ public_ip = (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
aws_instance.nginx_server: Creating...
aws_instance.nginx_server: Still creating... [00m10s elapsed]
aws_instance.nginx_server: Creation complete after 19s [id=i-08057ab819a682264]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
public_ip = "98.80.13.12"
Dell@DESKTOP-VV5NU50 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
Type here to search
32°C Smoke
2:43 pm
20/07/2025


MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 80
# (1 unchanged attribute hidden)
},
]
+ name = "nginx-security-group"
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ region = "us-east-1"
+ revoke_rules_on_delete = false
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}
Plan: 2 to add, 0 to change, 1 to destroy.
Changes to Outputs:
~ public_ip = "98.80.13.12" -> (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
aws_instance.nginx_server: Destroying... [id=i-08057ab819a682264]
aws_instance.nginx_server: Still destroying... [id=i-08057ab819a682264, 00m10s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-08057ab819a682264, 00m20s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-08057ab819a682264, 00m30s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-08057ab819a682264, 00m40s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-08057ab819a682264, 00m50s elapsed]
aws_instance.nginx_server: Destruction complete after 58s
aws_security_group.nginx_sg: Creating...
aws_security_group.nginx_sg: Creation complete after 7s [id=sg-0890473ca7b7e125d]
aws_instance.nginx_server: Creating...
aws_instance.nginx_server: Still creating... [00m10s elapsed]
aws_instance.nginx_server: Still creating... [00m20s elapsed]
aws_instance.nginx_server: Creation complete after 20s [id=i-0026fb0c5d017d88c]
Apply complete! Resources: 2 added, 0 changed, 1 destroyed.
Outputs:
public_ip = "184.72.122.145"
Dell@DESKTOP-VV5NU50 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
Type here to search
32°C Smoke
2:51 pm
20/07/2025

Access via web, use http as didn’t do the ssl stuff for https



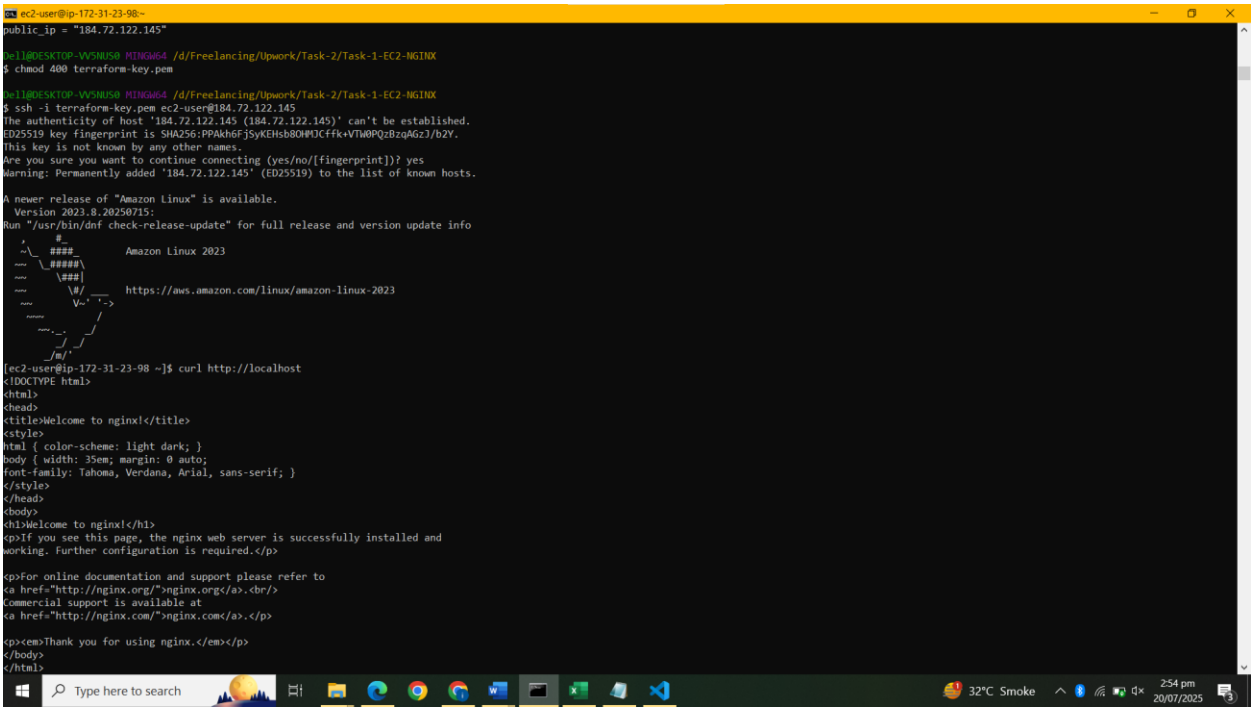
Welcome to nginx!

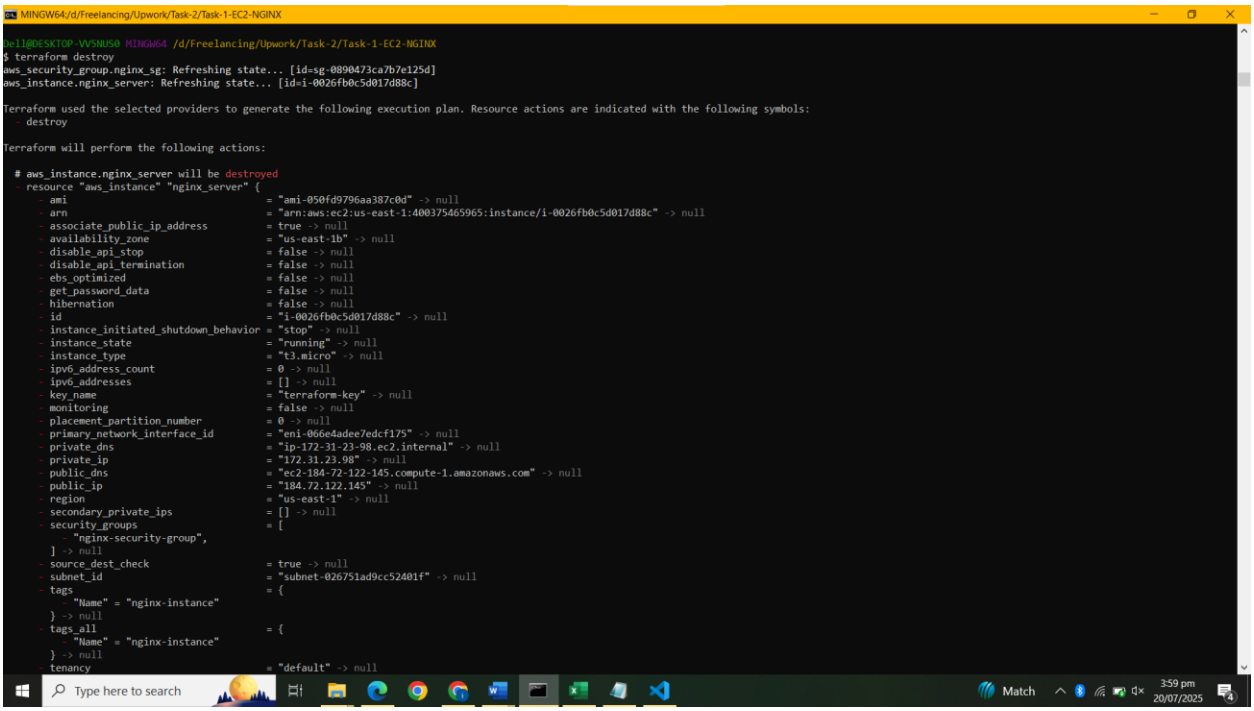
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

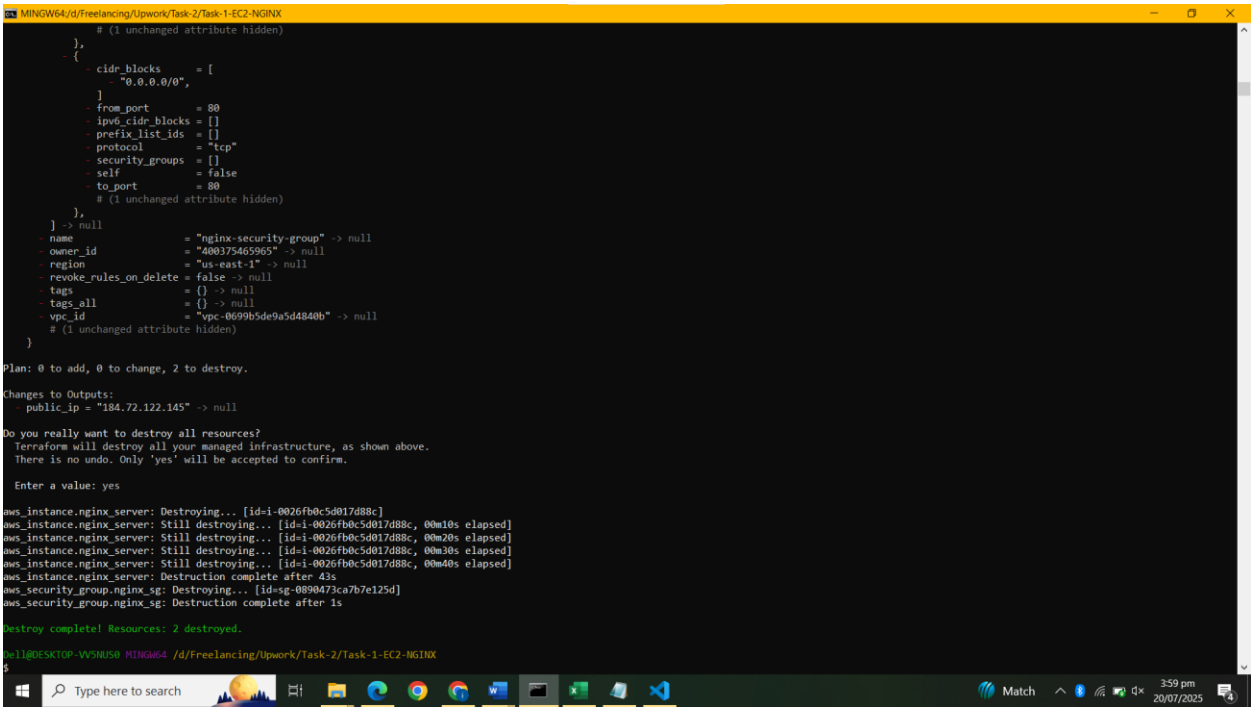
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Ssh success:



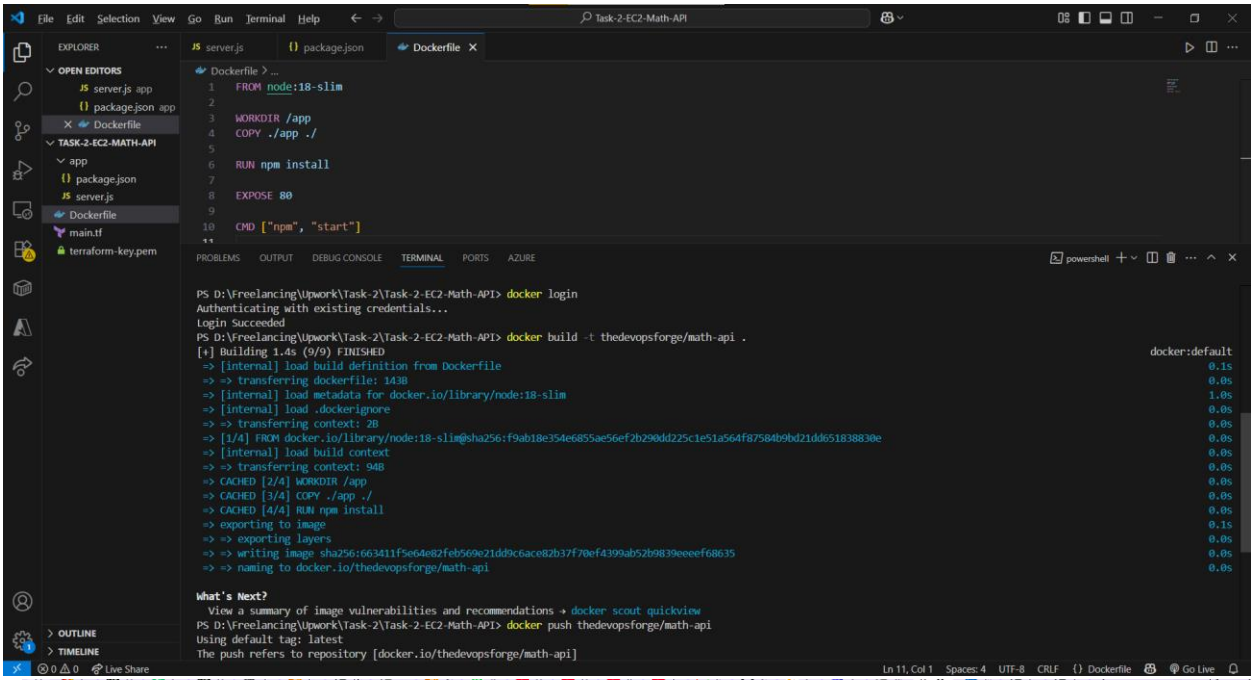

MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$ terraform destroy
aws_security_group.nginx_sg: Refreshing state... [id=sg-0890473ca7b7e125d]
aws_instance.nginx_server: Refreshing state... [id=i-0026fb0c5d017d88c]
Terraform 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_instance.nginx_server will be destroyed
- resource "aws_instance" "nginx_server" {
- ami = "ami-050fd9796aa387c0d" -> null
- arn = "arn:aws:ec2:us-east-1:400375465965:instance/i-0026fb0c5d017d88c" -> null
- associate_public_ip_address = true -> null
- availability_zone = "us-east-1b" -> null
- disable_api_stop = false -> null
- disable_api_termination = false -> null
- ebs_optimized = false -> null
- get_password_data = false -> null
- hibernation = false -> null
- id = "i-0026fb0c5d017d88c" -> null
- instance_initiated_shutdown_behavior = "stop" -> null
- instance_state = "running" -> null
- instance_type = "t3.micro" -> null
- ipv6_address_count = 0 -> null
- ipv6_addresses = [] -> null
- key_name = "terraform-key" -> null
- monitoring = false -> null
- placement_partition_number = 0 -> null
- primary_network_interface_id = "eni-066e4adee7edcf175" -> null
- private_dns = "ip-172-31-23-98.ec2.internal" -> null
- private_ip = "172.31.23.98" -> null
- public_dns = "ec2-184-72-122-145.compute-1.amazonaws.com" -> null
- public_ip = "184.72.122.145" -> null
- region = "us-east-1" -> null
- secondary_private_ips = [] -> null
- security_groups = [
- "nginx-security-group",
] -> null
- source_dest_check = true -> null
- subnet_id = "subnet-026751ad9cc52401f" -> null
- tags = {
- "Name" = "nginx-instance"
} -> null
- tags_all = {
- "Name" = "nginx-instance"
} -> null
- tenancy = "default" -> null
Type here to search
Match
3:59 pm
20/07/2025

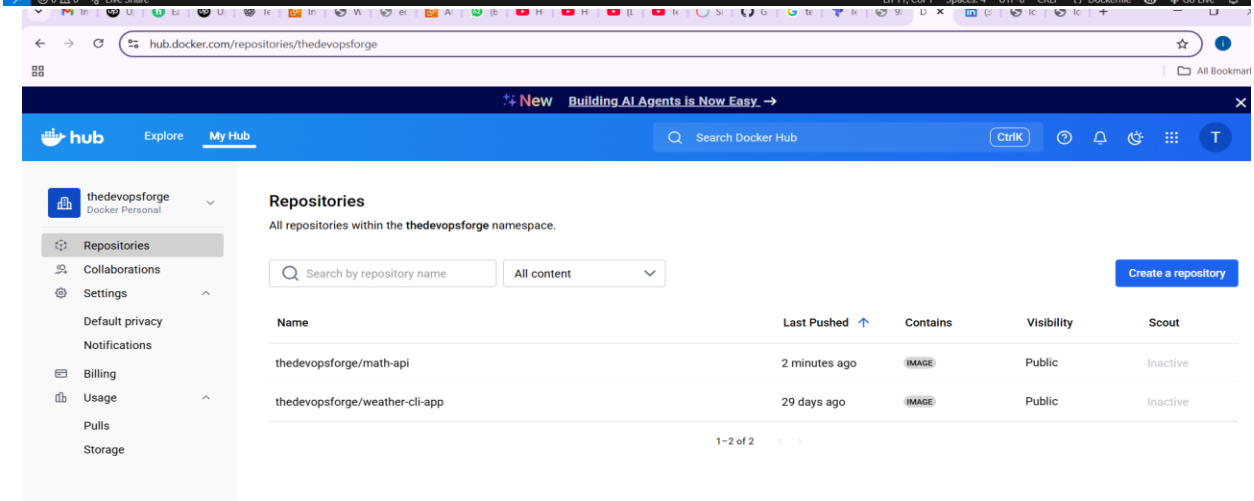

MINGW64:/d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
(1 unchanged attribute hidden)
},
{
- cidr_blocks = [
- "0.0.0.0/0",
]
- from_port = 80
- ipv6_cidr_blocks = []
- prefix_list_ids = []
- protocol = "tcp"
- security_groups = []
- self = false
- to_port = 80
(1 unchanged attribute hidden)
},
] -> null
- name = "nginx-security-group" -> null
- owner_id = "400375465965" -> null
- region = "us-east-1" -> null
- revoke_rules_on_delete = false -> null
- tags = {} -> null
- tags_all = {} -> null
- vpc_id = "vpc-0699b5de9a5d4840b" -> null
# (1 unchanged attribute hidden)
}
Plan: 0 to add, 0 to change, 2 to destroy.
Changes to Outputs:
- public_ip = "184.72.122.145" -> null
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_instance.nginx_server: Destroying... [id=i-0026fb0c5d017d88c]
aws_instance.nginx_server: Still destroying... [id=i-0026fb0c5d017d88c, 00m10s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-0026fb0c5d017d88c, 00m20s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-0026fb0c5d017d88c, 00m30s elapsed]
aws_instance.nginx_server: Still destroying... [id=i-0026fb0c5d017d88c, 00m40s elapsed]
aws_instance.nginx_server: Destruction complete after 43s
aws_security_group.nginx_sg: Destroying... [id=sg-0890473ca7b7e125d]
aws_security_group.nginx_sg: Destruction complete after 1s
Destroy complete! Resources: 2 destroyed.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-1-EC2-NGINX
$
Type here to search
Match
3:59 pm
20/07/2025

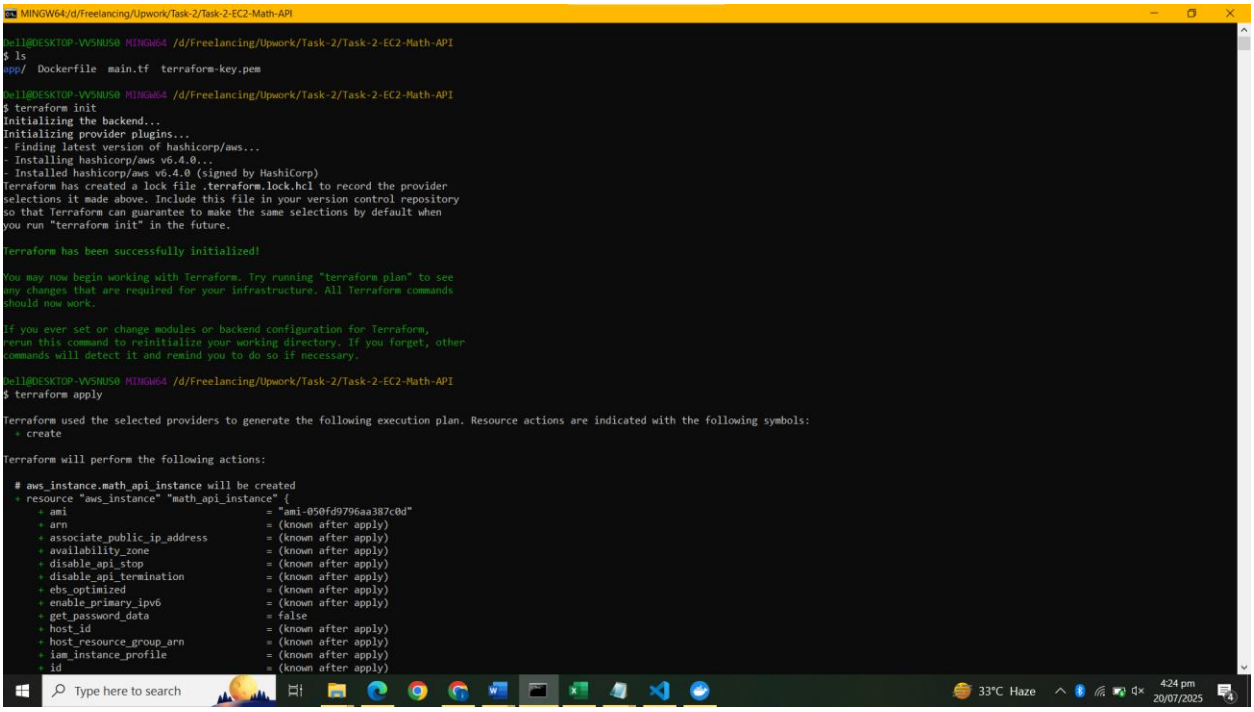
TASK#2:

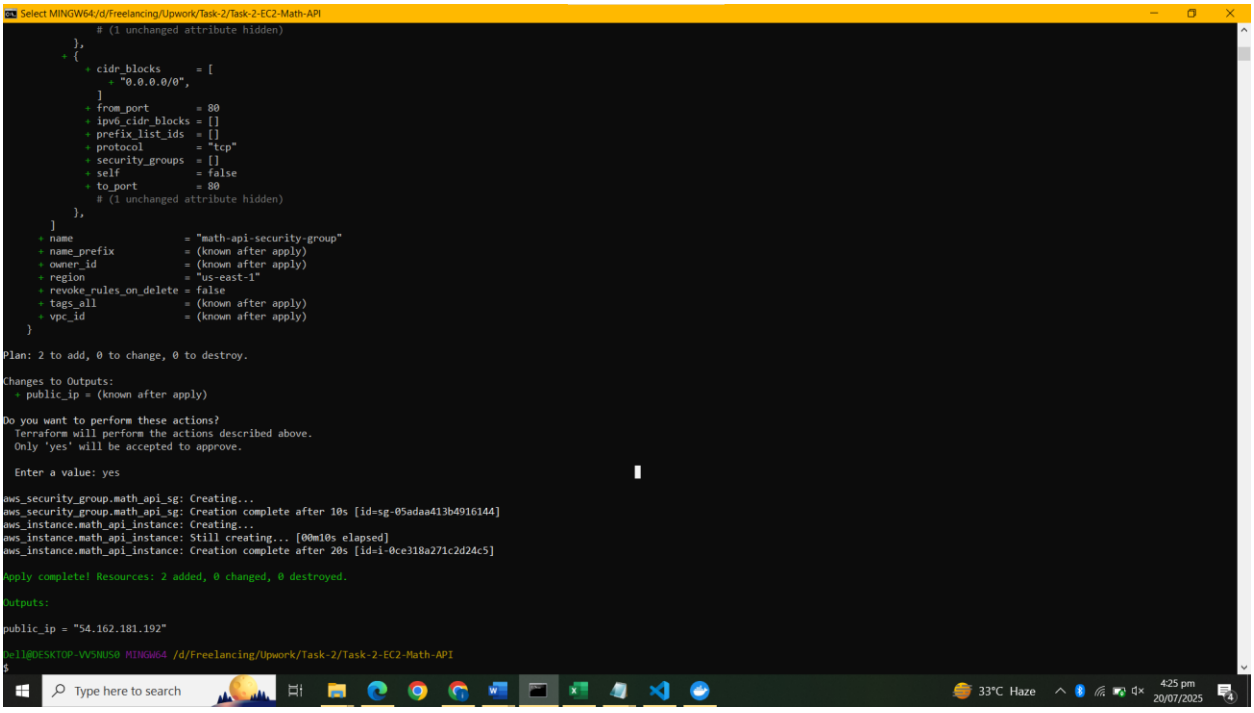
Task 2: Deploy Backend API on EC2 using Terraform
Goal: Automate API deployment with Docker on EC2 via Terraform.
Why: This teaches how to combine Terraform and Docker to automate app deployment.
What You'll Learn: Writing a Dockerfile, managing Docker Hub images, using Terraform user_data.

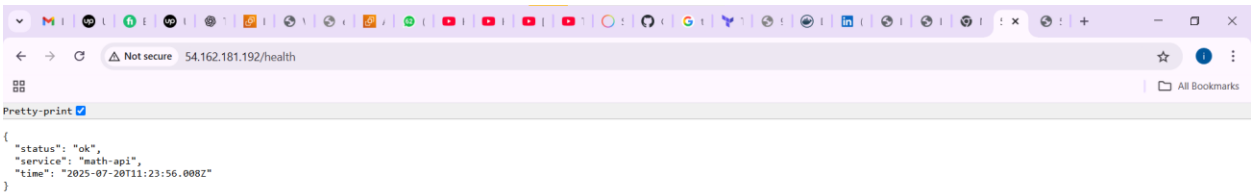
- *Write a Dockerfile for a Node.js or Flask API.*
- *Push your image to Docker Hub.*
- *Use Terraform to create an EC2 instance with Docker installed.*
- *Deploy the container using user_data.*
- *Validate via the /health endpoint.*



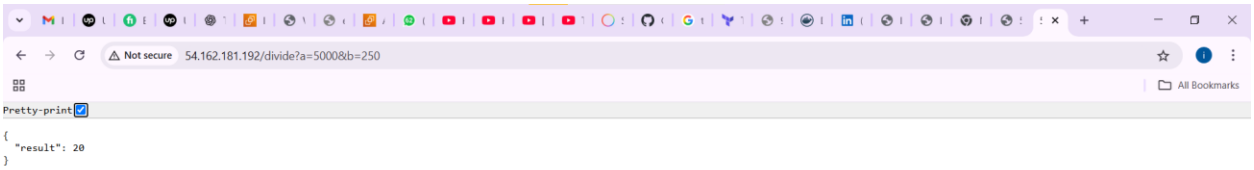



MINGW64:/d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
$ ls
app/ Dockerfile main.tf terraform-key.pem
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.4.0...
- Installed hashicorp/aws v6.4.0 (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.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
$ 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_instance.math_api_instance will be created
  + resource "aws_instance" "math_api_instance" {
      + ami   = "ami-050fd9796aa387c0d"
      + arn   = (known after apply)
      + associate_public_ip_address   = (known after apply)
      + availability_zone   = (known after apply)
      + disable_api_stop   = (known after apply)
      + disable_api_termination   = (known after apply)
      + ebs_optimized   = (known after apply)
      + enable_primary_ipv6   = (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)
Type here to search
33°C Haze
4:24 pm
20/07/2025

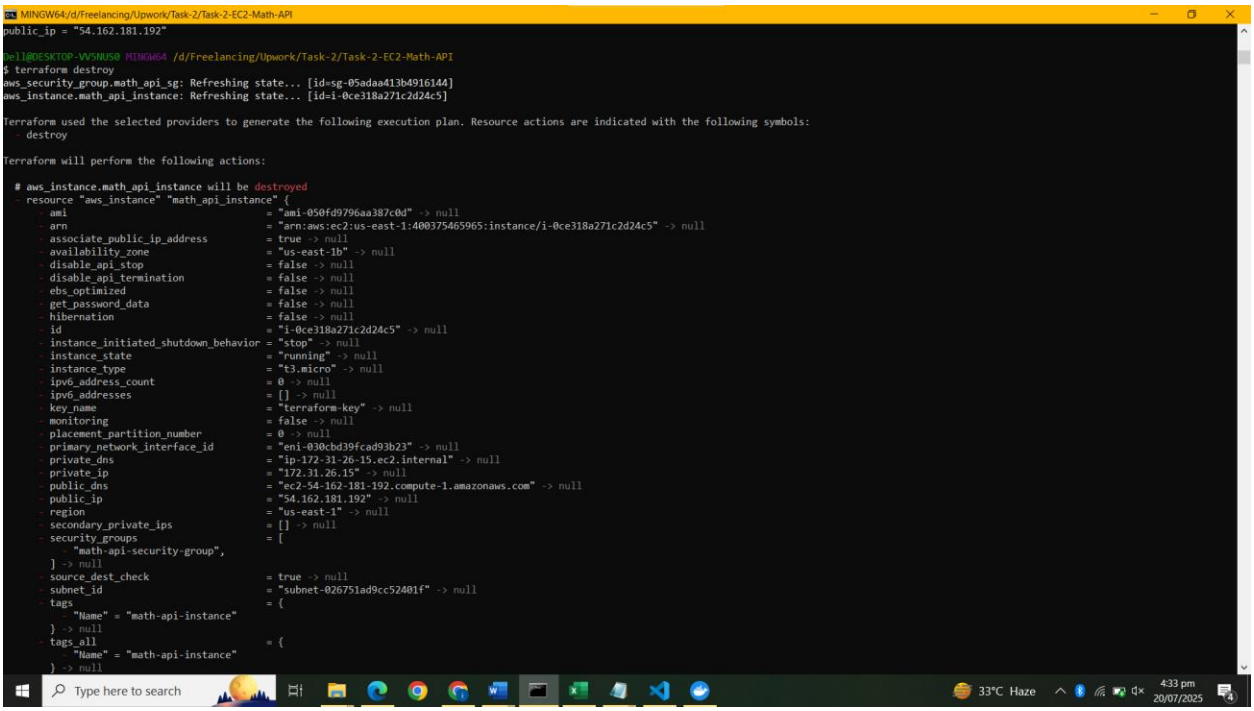

Select MINGW64:/d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
            # (1 unchanged attribute hidden)
          },
        + {
            + cidr_blocks   = [
                + "0.0.0.0/0",
              ]
            + from_port   = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids = []
            + protocol   = "tcp"
            + security_groups = []
            + self   = false
            + to_port   = 80
            # (1 unchanged attribute hidden)
          },
      ]
    + name   = "math-api-security-group"
    + name_prefix   = (known after apply)
    + owner_id   = (known after apply)
    + region   = "us-east-1"
    + revoke_rules_on_delete = false
    + tags_all   = (known after apply)
    + vpc_id   = (known after apply)
  }
Plan: 2 to add, 0 to change, 0 to destroy.
Changes to Outputs:
  + public_ip = (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
aws_security_group.math_api_sg: Creating...
aws_security_group.math_api_sg: Creation complete after 10s [id=sg-05adaa413b4916144]
aws_instance.math_api_instance: Creating...
aws_instance.math_api_instance: Still creating... [00m10s elapsed]
aws_instance.math_api_instance: Creation complete after 20s [id=i-0ce318a271c2d24c5]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Outputs:
public_ip = "54.162.181.192"
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-2-EC2-Math-API
$
Type here to search
33°C Haze
4:25 pm
20/07/2025


Not secure
54.162.181.192/health
All Bookmarks
Pretty-print
{
"status": "ok",
"service": "math-api",
"time": "2025-07-20T11:23:56.008Z"
}


Type here to search
33°C Haze
4:25 pm
20/07/2025

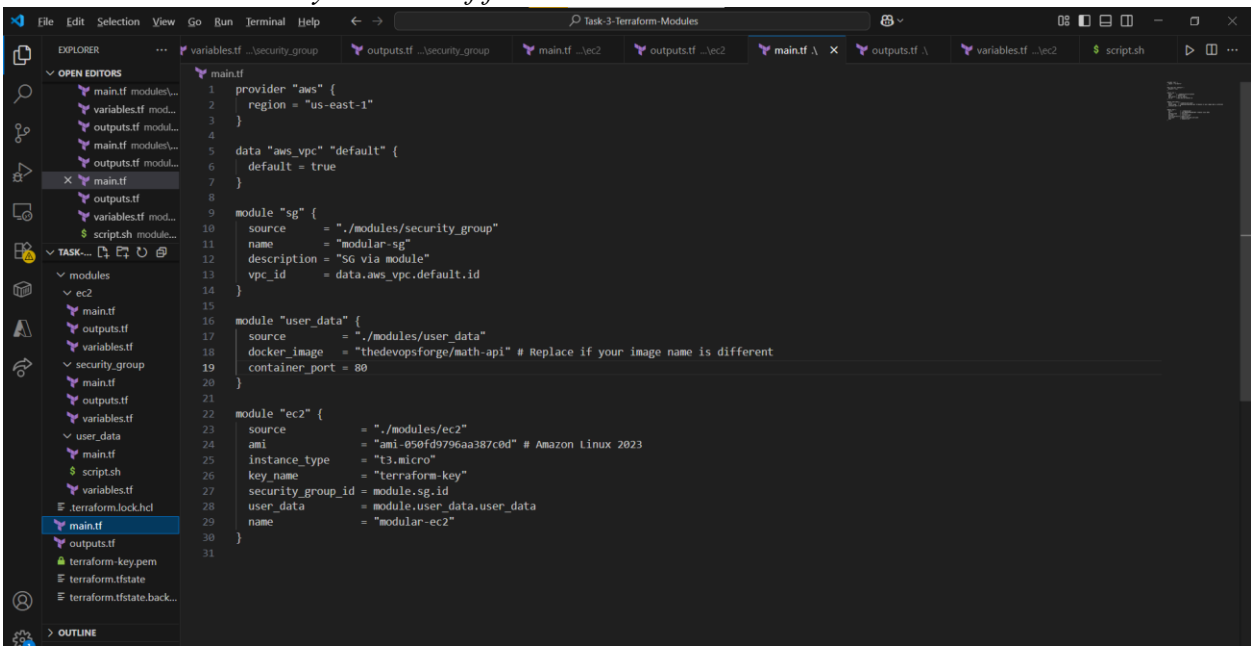

Not secure
54.162.181.192/divide?a=5000&b=250
All Bookmarks
Pretty-print
{
"result": 20
}

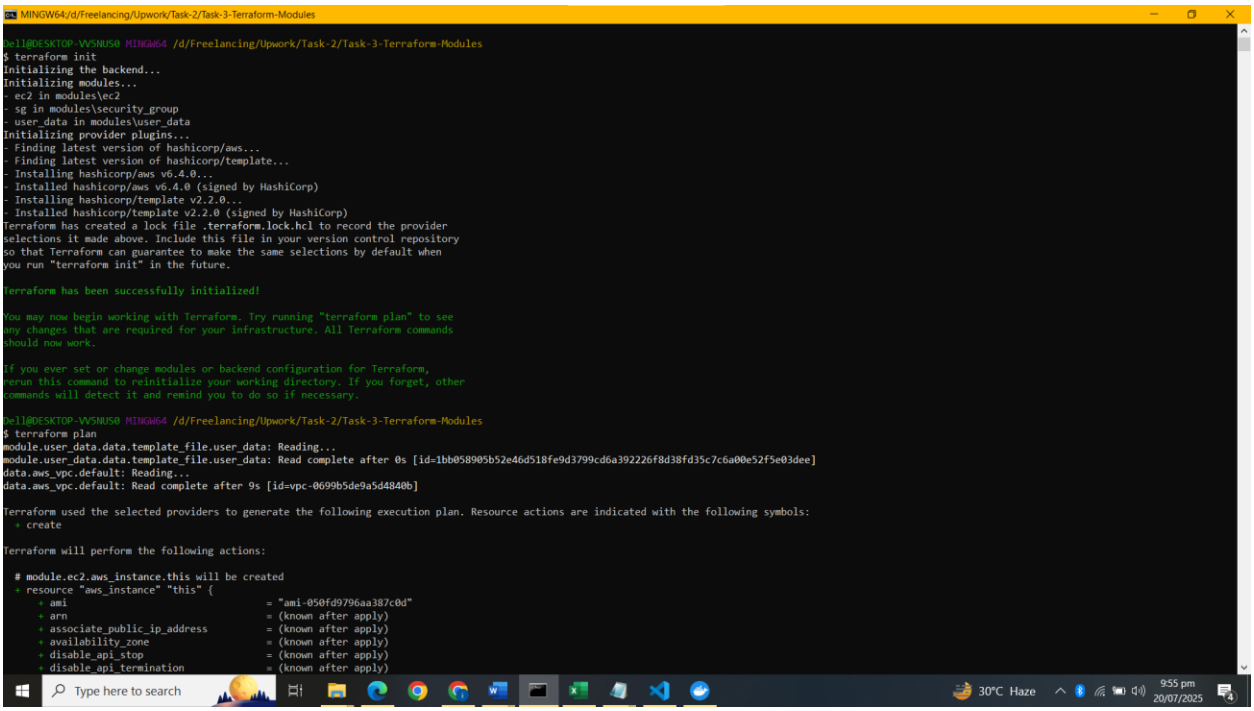

Type here to search
33°C Haze
4:25 pm
20/07/2025

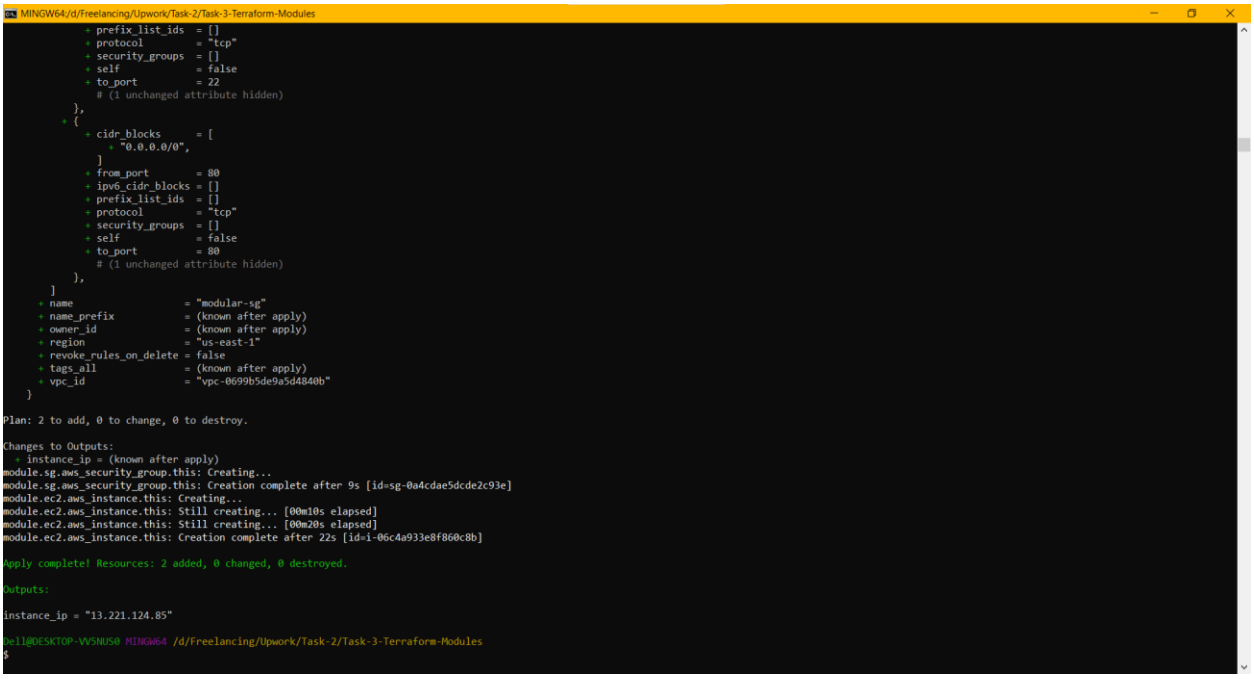


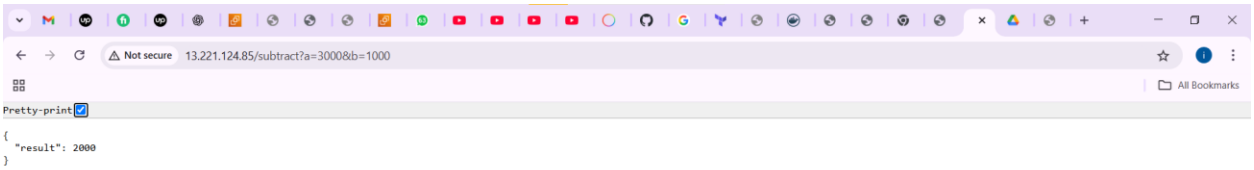
TASK#3:

Task 3: Use Terraform Modules
Goal: Modularize Terraform code for reusability.
Why: Modules help avoid repetition and improve code maintenance.
What You'll Learn: How to build and reuse Terraform modules.
- *Create modules for EC2, Security Group, and deployment logic.*
- *Use variables and outputs to pass data between modules.*
- *Call these modules in your main .tf files.*




MINGW64:/d/Freelancing/Upwork/Task-2/Task-3-Terraform-Modules
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-3-Terraform-Modules
$ terraform init
Initializing the backend...
Initializing modules...
- ec2 in modules\ec2
- sg in modules\security_group
- user_data in modules\user_data
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/template...
- Installing hashicorp/aws v6.4.0...
- Installed hashicorp/aws v6.4.0 (signed by HashiCorp)
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (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.
Dell@DESKTOP-VV5NUS0 MINGW64 /d/Freelancing/Upwork/Task-2/Task-3-Terraform-Modules
$ terraform plan
module.user_data.data.template_file.user_data: Reading...
module.user_data.data.template_file.user_data: Read complete after 0s [id=1bb058905b52e46d518fe9d3799cd6a392226f8d38fd35c7c6a00e52f5e03dee]
data.aws_vpc.default: Reading...
data.aws_vpc.default: Read complete after 9s [id=vpc-0699b5de9a5d4840b]
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:
# module.ec2.aws_instance.this will be created
+ resource "aws_instance" "this" {
+ ami = "ami-050fd9796aa387c0d"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
Type here to search
30°C Haze
9:55 pm
20/07/2025

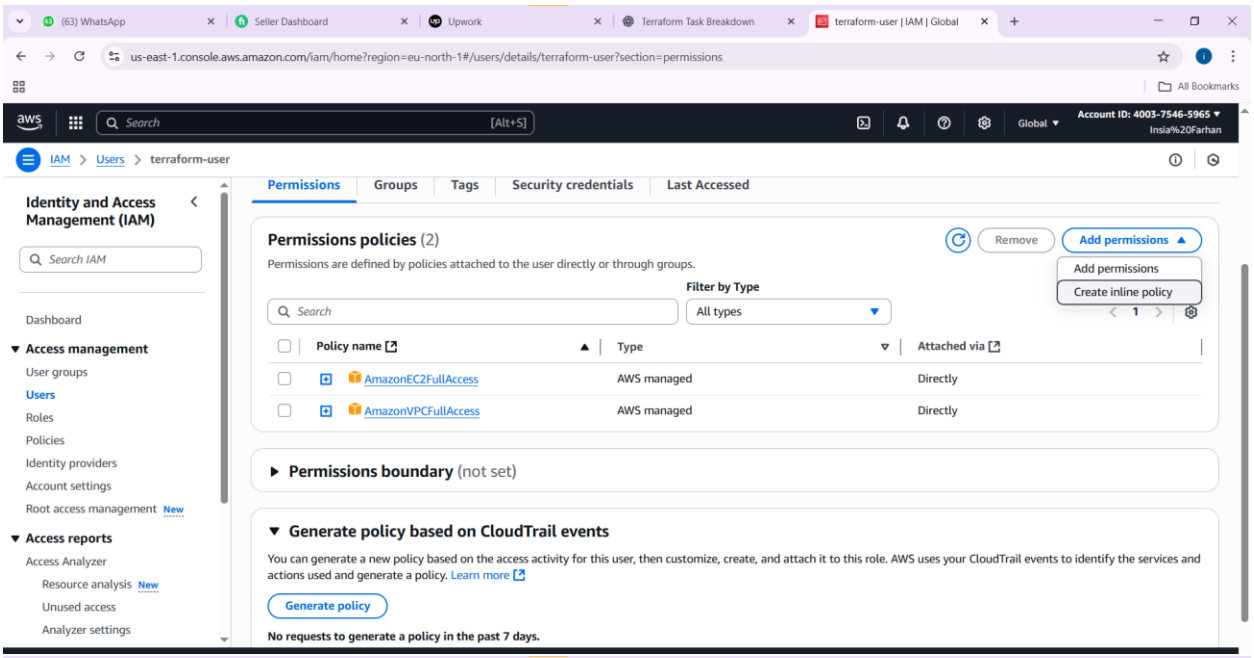

MINGW64:/d/Freelancing/Upwork/Task-2/Task-3-Terraform-Modules
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 22
# (1 unchanged attribute hidden)
},
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ from_port = 80
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 80
# (1 unchanged attribute hidden)
},
]
+ name = "modular-sg"
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ region = "us-east-1"
+ revoke_rules_on_delete = false
+ tags_all = (known after apply)
+ vpc_id = "vpc-0699b5de9a5d4840b"
}
Plan: 2 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ instance_ip = (known after apply)
module.sg.aws_security_group.this: Creating...
module.sg.aws_security_group.this: Creation complete after 9s [id=sg-0a4cdae5dcde2c93e]
module.ec2.aws_instance.this: Creating...
module.ec2.aws_instance.this: Still creating... [00m10s elapsed]
module.ec2.aws_instance.this: Still creating... [00m20s elapsed]
module.ec2.aws_instance.this: Creation complete after 22s [id=i-06c4a933e8f860c8b]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Outputs:
instance_ip = "13.221.124.85"
Dell@DESKTOP-VV5NU50 MINGW64 /d/Freelancing/Upwork/Task-2/Task-3-Terraform-Modules
$

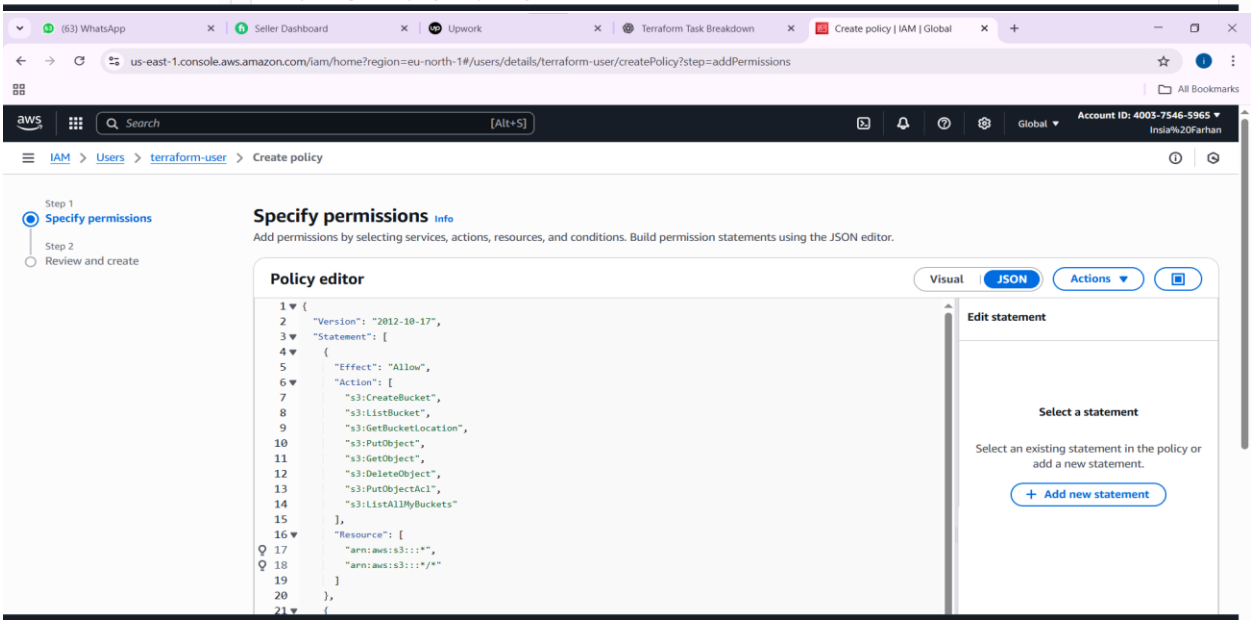

Not secure
13.221.124.85/subtract?a=3000&b=1000
All Bookmarks
Pretty-print
{
"result": 2000
}

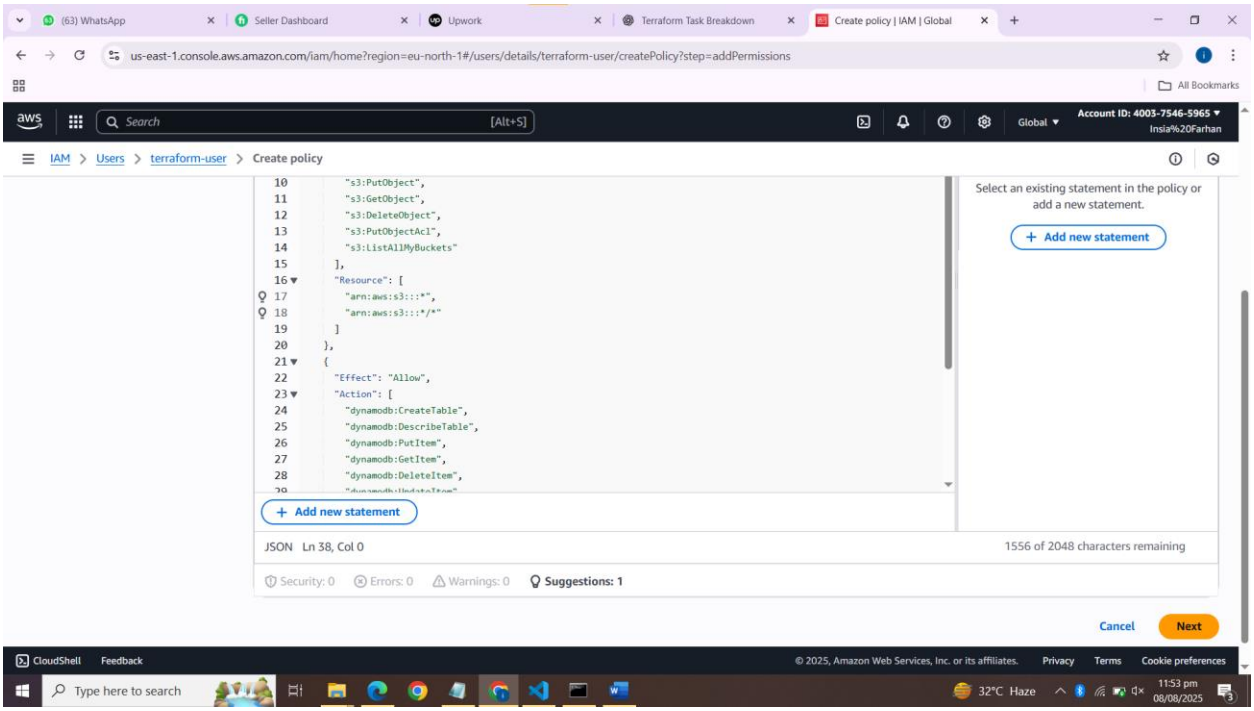

Type here to search
30°C Haze
10:25 pm
20/07/2025

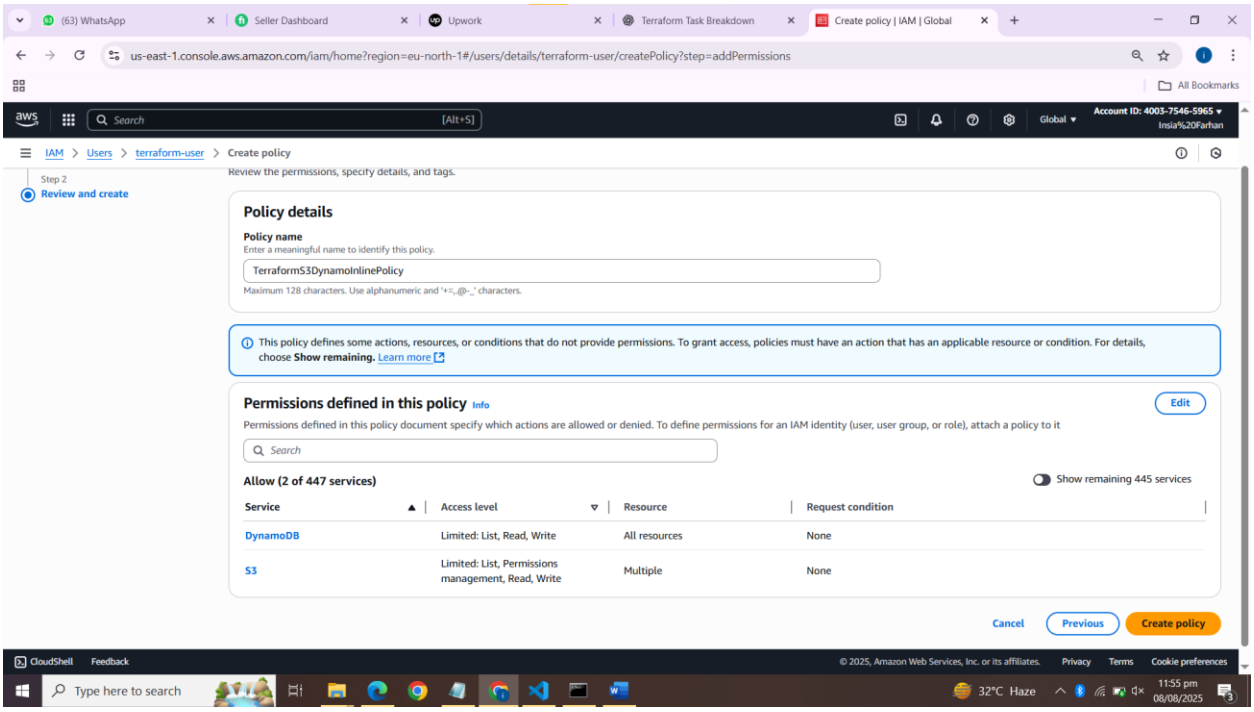
Task#4:

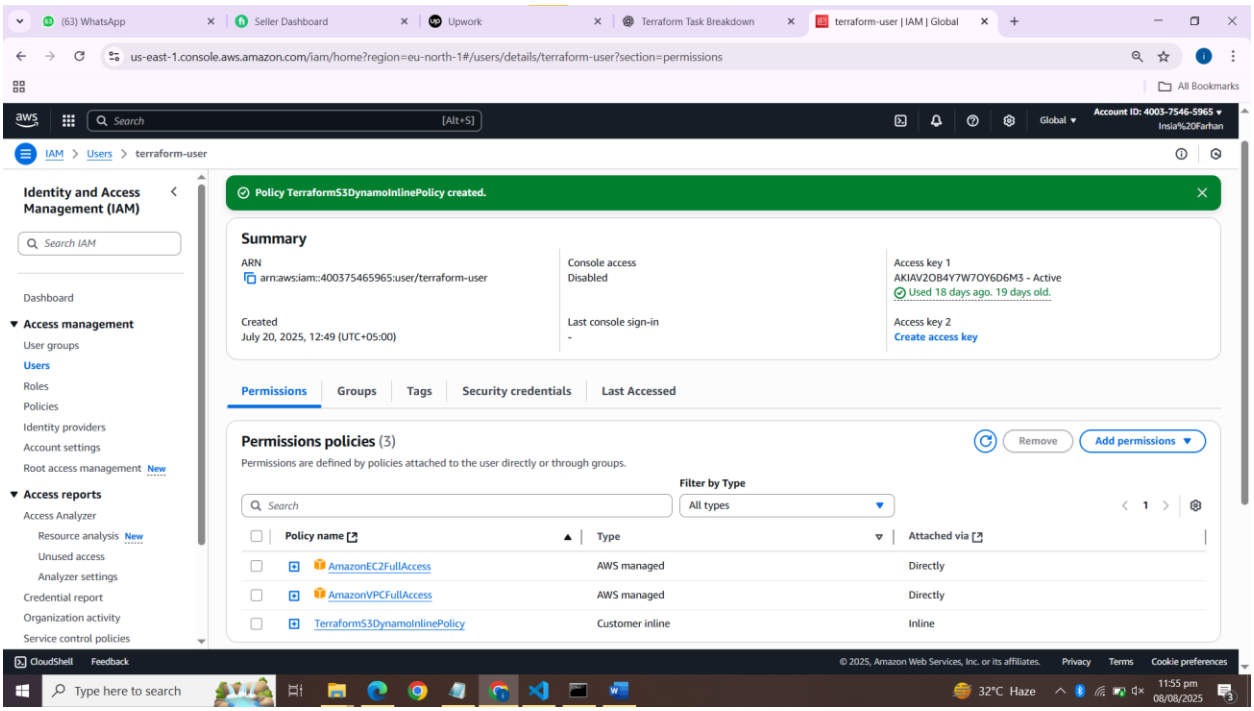
Task 4: Use S3 Backend and State Locking
Goal: Manage Terraform state remotely and securely.
Why: Remote state sharing is essential in team environments.
What You'll Learn: Setting up S3 backend and DynamoDB locking.
• Create an S3 bucket for storing .tfstate files.
• Create a DynamoDB table for state locking.
• Configure the backend block in Terraform.
• Test state locking with concurrent apply.

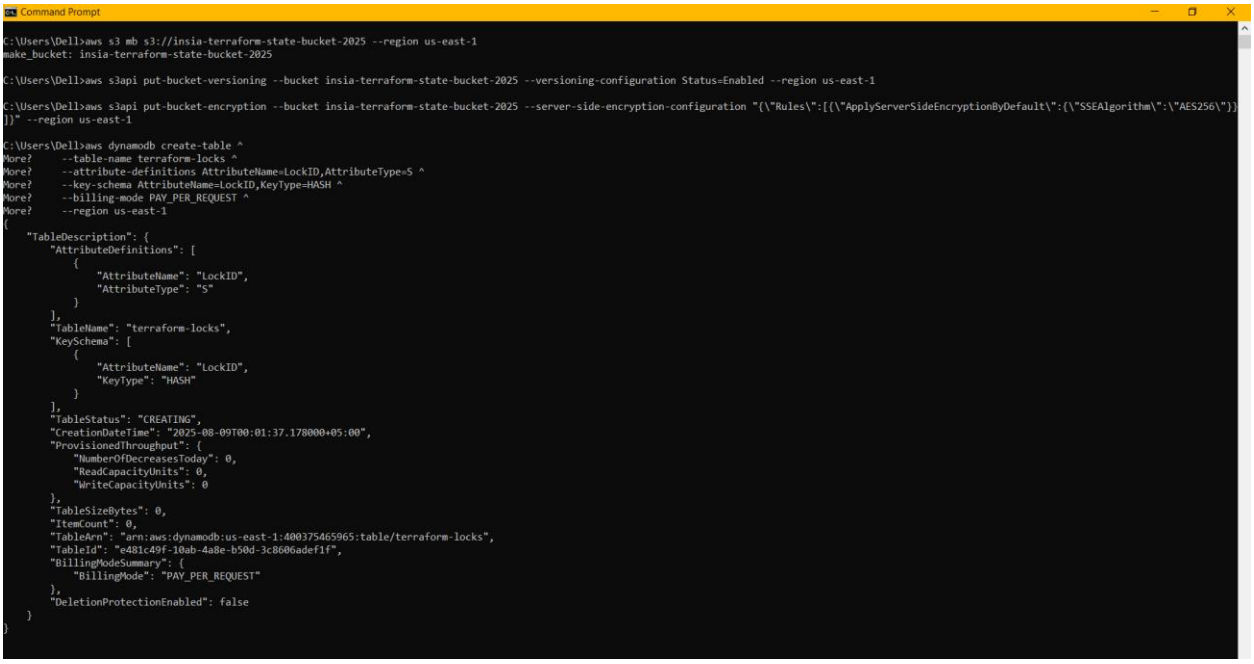


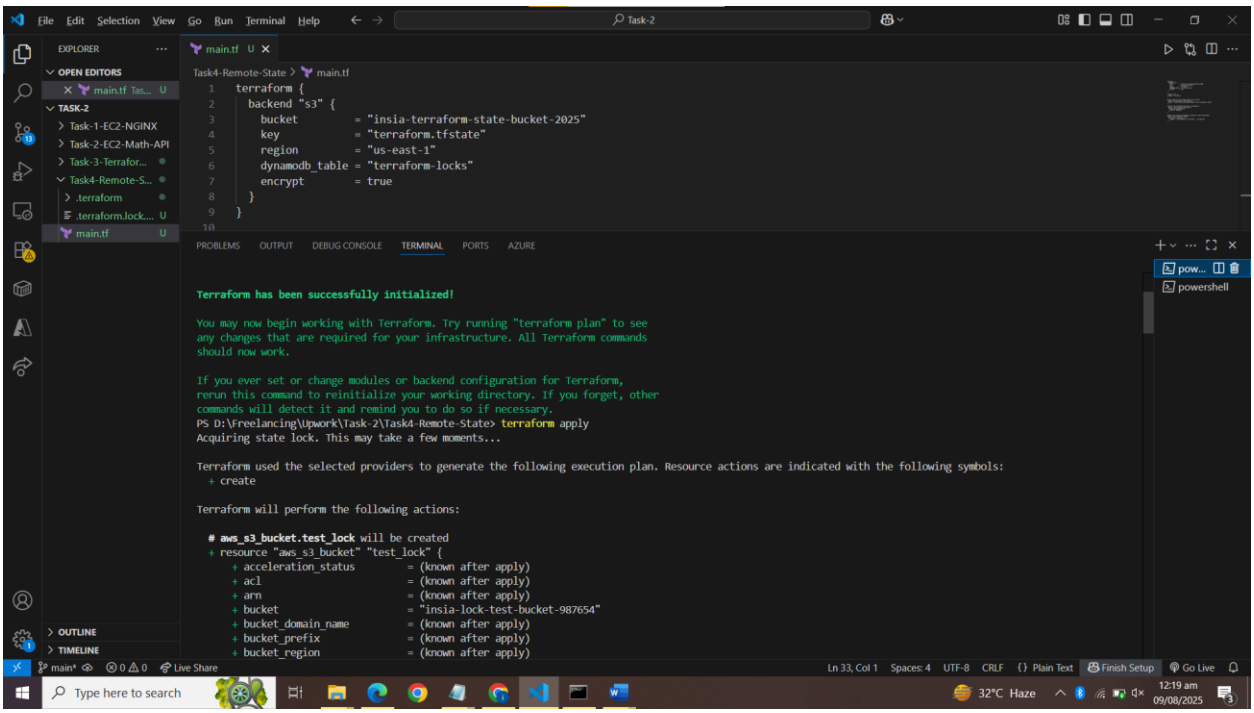


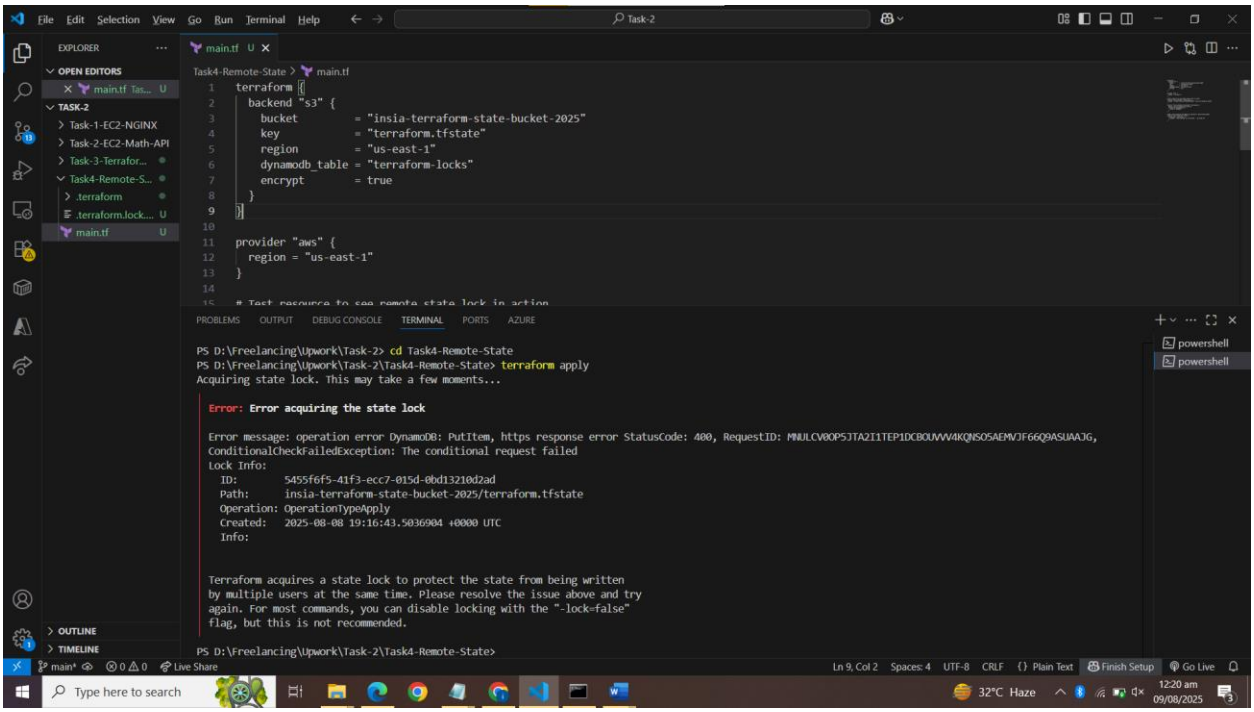

Create policy | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/terraform-user/createPolicy?step=addPermissions
IAM > Users > terraform-user > Create policy
"s3:PutObject",
"s3:GetObject",
"s3:DeleteObject",
"s3:PutObjectAcl",
"s3:ListAllMyBuckets"
"Resource": [
"arn:aws:s3:::*",
"arn:aws:s3:::*/*"
},
"Effect": "Allow",
"Action": [
"dynamodb:CreateTable",
"dynamodb:DescribeTable",
"dynamodb:PutItem",
"dynamodb:GetItem",
"dynamodb:DeleteItem",
Select an existing statement in the policy or add a new statement.
+ Add new statement
+ Add new statement
JSON Ln 38, Col 0
1556 of 2048 characters remaining
Security: 0 Errors: 0 Warnings: 0 Suggestions: 1
Cancel
Next
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Type here to search
32°C Haze
11:53 pm
08/08/2025

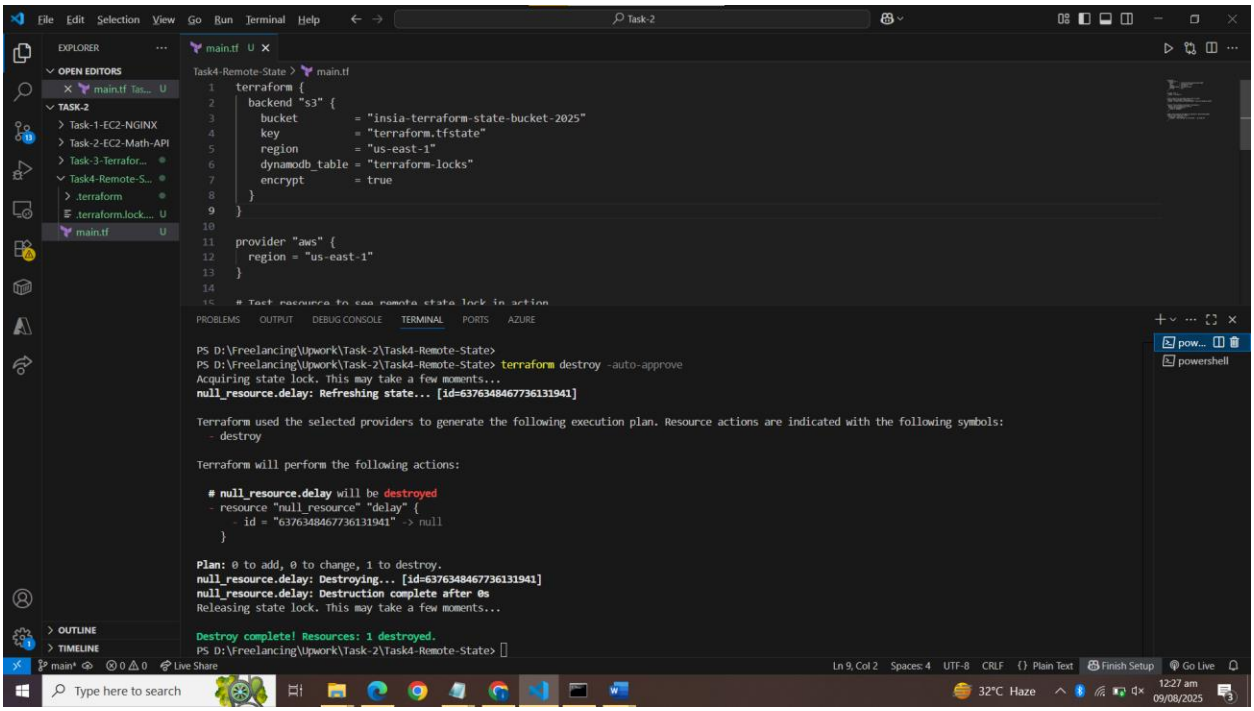

Create policy | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/terraform-user/createPolicy?step=addPermissions
IAM > Users > terraform-user > Create policy
Step 2
Review and create
Review the permissions, specify details, and tags.
Policy details
Policy name
Enter a meaningful name to identify this policy.
TerraformS3DynamoInlinePolicy
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.
This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose Show remaining. Learn more
Permissions defined in this policy Info
Edit
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it
Search
Allow (2 of 447 services)
Show remaining 445 services
Service
Access level
Resource
Request condition
DynamoDB
Limited: List, Read, Write
All resources
None
S3
Limited: List, Permissions management, Read, Write
Multiple
None
Cancel
Previous
Create policy
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Type here to search
32°C Haze
11:55 pm
08/08/2025

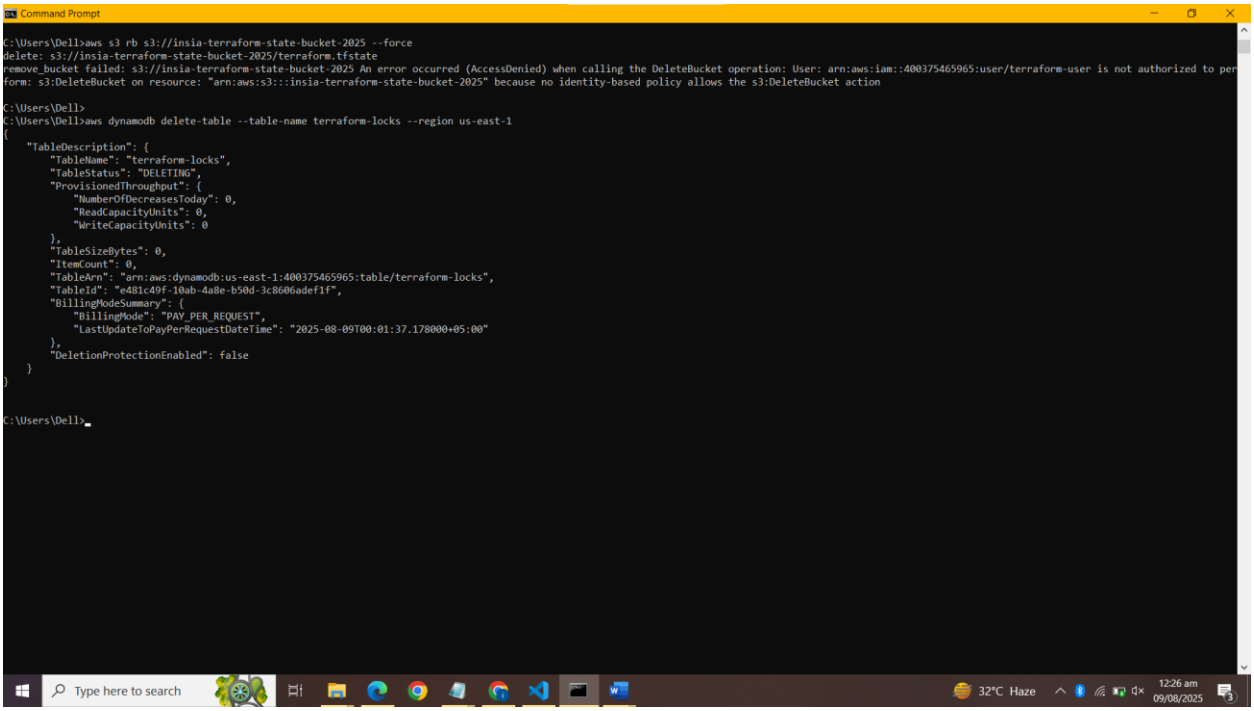

(63) WhatsApp
Seller Dashboard
Upwork
Terraform Task Breakdown
terraform-user | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/terraform-user?section=permissions
Search
[Alt+S]
Global
Account ID: 4003-7546-5965
Insia%20Farhan
IAM > Users > terraform-user
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Policy TerraformS3DynamoInlinePolicy created.
Summary
ARN
arn:aws:iam::400375465965:user/terraform-user
Console access
Disabled
Access key 1
AKIAV2OB4Y7W7OY6D6M3 - Active
Used 18 days ago. 19 days old.
Created
July 20, 2025, 12:49 (UTC+05:00)
Last console sign-in
-
Access key 2
Create access key
Permissions
Groups
Tags
Security credentials
Last Accessed
Permissions policies (3)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached via
AmazonEC2FullAccess
AWS managed
Directly
AmazonVPCFullAccess
AWS managed
Directly
TerraformS3DynamoInlinePolicy
Customer inline
Inline
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
32°C Haze
11:55 pm
08/08/2025


Command Prompt
C:\Users\Dell>aws s3 mb s3://insia-terraform-state-bucket-2025 --region us-east-1
make_bucket: insia-terraform-state-bucket-2025
C:\Users\Dell>aws s3api put-bucket-versioning --bucket insia-terraform-state-bucket-2025 --versioning-configuration Status=Enabled --region us-east-1
C:\Users\Dell>aws s3api put-bucket-encryption --bucket insia-terraform-state-bucket-2025 --server-side-encryption-configuration "{\"Rules\":[{\"ApplyServerSideEncryptionByDefault\":{\"SSEAlgorithm\":\"AES256\"}}]}" --region us-east-1
C:\Users\Dell>aws dynamodb create-table ^
More? --table-name terraform-locks ^
More? --attribute-definitions AttributeName=LockID,AttributeType=S ^
More? --key-schema AttributeName=LockID,KeyType=HASH ^
More? --billing-mode PAY_PER_REQUEST ^
More? --region us-east-1
{
"TableDescription": {
"AttributeDefinitions": [
{
"AttributeName": "LockID",
"AttributeType": "S"
}
],
"TableName": "terraform-locks",
"KeySchema": [
{
"AttributeName": "LockID",
"KeyType": "HASH"
}
],
"TableStatus": "CREATING",
"CreationDateTime": "2025-08-09T00:01:37.178000+05:00",
"ProvisionedThroughput": {
"NumberOfDecreasesToday": 0,
"ReadCapacityUnits": 0,
"WriteCapacityUnits": 0
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-east-1:400375465965:table/terraform-locks",
"TableId": "e481c49f-10ab-4a8e-b50d-3c8606adef1f",
"BillingModeSummary": {
"BillingMode": "PAY_PER_REQUEST"
},
"DeletionProtectionEnabled": false
}
}


File Edit Selection View Go Run Terminal Help
Task-2
EXPLORER
OPEN EDITORS
main.tf Tas... U
TASK-2
Task-1-EC2-NGINX
Task-2-EC2-Math-API
Task-3-Terrafor...
Task4-Remote-S...
.terraform
.terraform.lock.... U
main.tf U
main.tf U
Task4-Remote-State > main.tf
terraform {
backend "s3" {
bucket = "insia-terraform-state-bucket-2025"
key = "terraform.tfstate"
region = "us-east-1"
dynamodb_table = "terraform-locks"
encrypt = true
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
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.
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State> terraform apply
Acquiring state lock. This may take a few moments...
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_s3_bucket.test_lock will be created
+ resource "aws_s3_bucket" "test_lock" {
+ acceleration_status = (known after apply)
+ acl = (known after apply)
+ arn = (known after apply)
+ bucket = "insia-lock-test-bucket-987654"
+ bucket_domain_name = (known after apply)
+ bucket_prefix = (known after apply)
+ bucket_region = (known after apply)
pow...
powershell
OUTLINE
TIMELINE
main* Live Share
Ln 33, Col 1 Spaces: 4 UTF-8 CRLF {} Plain Text Finish Setup Go Live
Type here to search
32°C Haze
12:19 am
09/08/2025

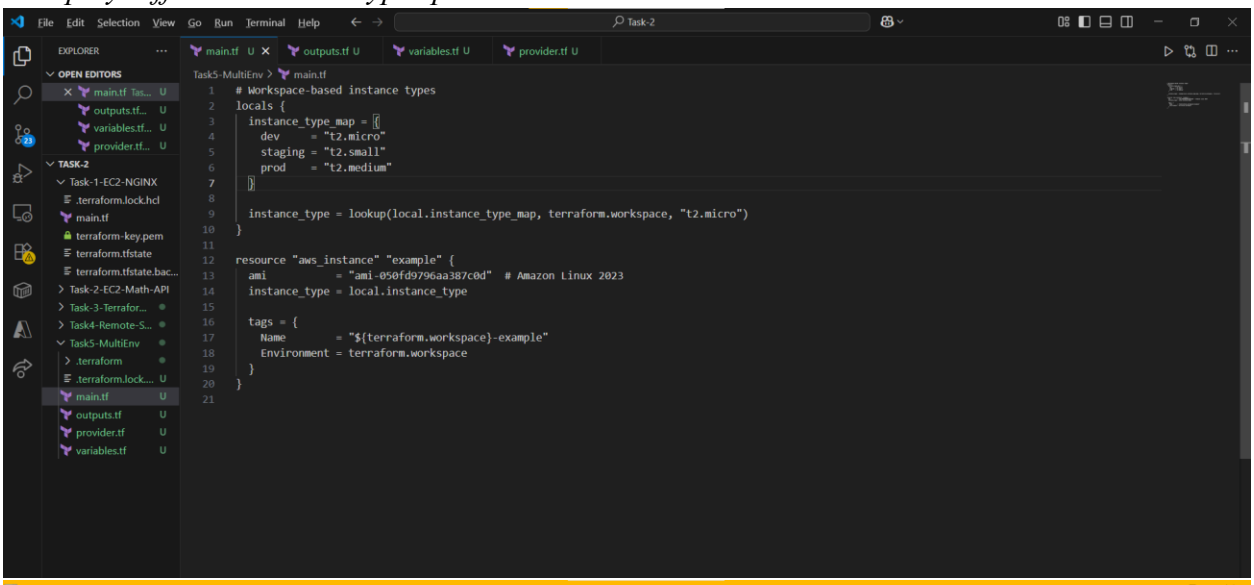

File Edit Selection View Go Run Terminal Help
Task-2
EXPLORER
OPEN EDITORS
main.tf Tas... U
TASK-2
Task-1-EC2-NGINX
Task-2-EC2-Math-API
Task-3-Terrafor...
Task4-Remote-S...
.terraform
.terraform.lock.... U
main.tf U
main.tf U
Task4-Remote-State > main.tf
terraform {
backend "s3" {
bucket = "insia-terraform-state-bucket-2025"
key = "terraform.tfstate"
region = "us-east-1"
dynamodb_table = "terraform-locks"
encrypt = true
}
}
provider "aws" {
region = "us-east-1"
}
# Test resource to see remote state lock in action
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS D:\Freelancing\Upwork\Task-2> cd Task4-Remote-State
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State> terraform apply
Acquiring state lock. This may take a few moments...
Error: Error acquiring the state lock
Error message: operation error DynamoDB: PutItem, https response error StatusCode: 400, RequestID: MNJLCV0OP5JTA2I1TEP1DCBOUVVV4KQNSO5AEMVJF66Q9ASUAAJG,
ConditionalCheckFailedException: The conditional request failed
Lock Info:
ID: 5455f6f5-41f3-ecc7-015d-0bd13210d2ad
Path: insia-terraform-state-bucket-2025/terraform.tfstate
Operation: OperationTypeApply
Created: 2025-08-08 19:16:43.5036904 +0000 UTC
Info:
Terraform acquires a state lock to protect the state from being written
by multiple users at the same time. Please resolve the issue above and try
again. For most commands, you can disable locking with the "-lock=false"
flag, but this is not recommended.
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State>
powershell
powershell
OUTLINE
TIMELINE
main* Live Share
Ln 9, Col 2 Spaces: 4 UTF-8 CRLF {} Plain Text Finish Setup Go Live
Type here to search
32°C Haze
12:20 am
09/08/2025

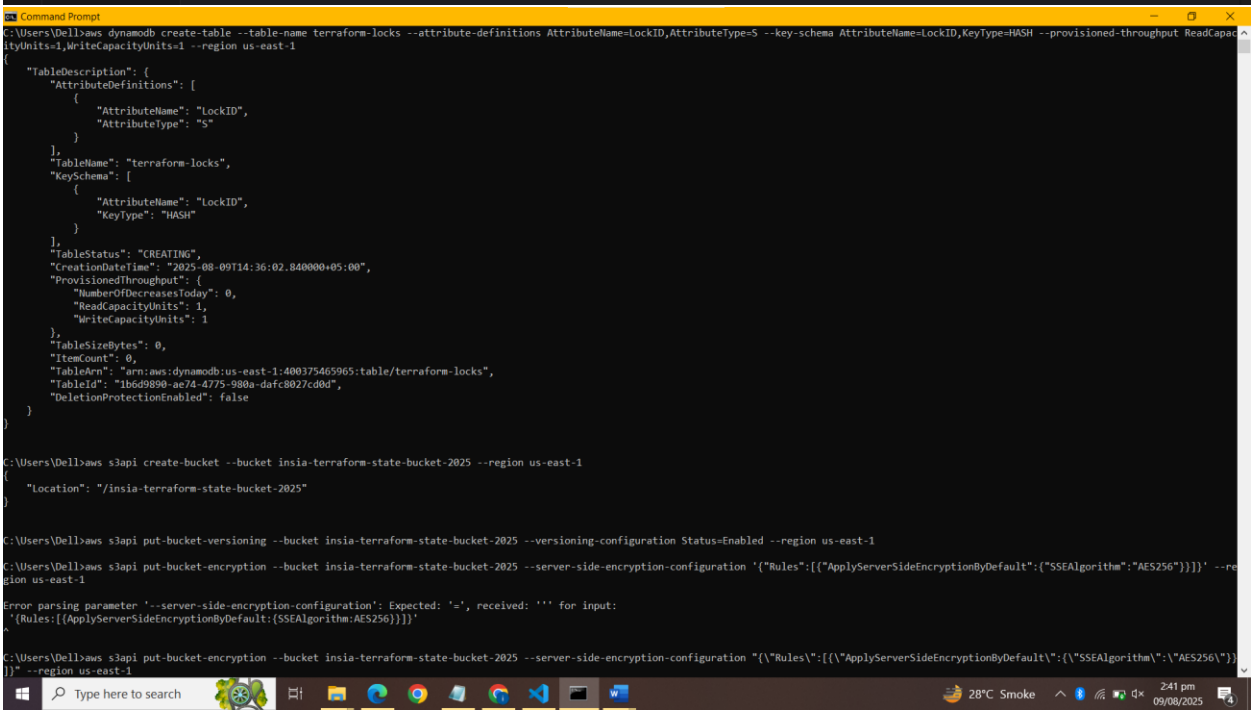

File Edit Selection View Go Run Terminal Help
Task-2
EXPLORER
OPEN EDITORS
main.tf Tas... U
TASK-2
Task-1-EC2-NGINX
Task-2-EC2-Math-API
Task-3-Terrafor...
Task4-Remote-S...
.terraform
.terraform.lock.... U
main.tf U
main.tf U
Task4-Remote-State > main.tf
1 terraform {
2 backend "s3" {
3 bucket = "insia-terraform-state-bucket-2025"
4 key = "terraform.tfstate"
5 region = "us-east-1"
6 dynamodb_table = "terraform-locks"
7 encrypt = true
8 }
9 }
10
11 provider "aws" {
12 region = "us-east-1"
13 }
14
15 # Test resource to see remote state lock in action
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State>
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State> terraform destroy -auto-approve
Acquiring state lock. This may take a few moments...
null_resource.delay: Refreshing state... [id=6376348467736131941]
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:
# null_resource.delay will be destroyed
- resource "null_resource" "delay" {
- id = "6376348467736131941" -> null
}
Plan: 0 to add, 0 to change, 1 to destroy.
null_resource.delay: Destroying... [id=6376348467736131941]
null_resource.delay: Destruction complete after 0s
Releasing state lock. This may take a few moments...
Destroy complete! Resources: 1 destroyed.
PS D:\Freelancing\Upwork\Task-2\Task4-Remote-State>
pow...
powershell
OUTLINE
TIMELINE
main* 0 0 Live Share
Ln 9, Col 2 Spaces: 4 UTF-8 CRLF {} Plain Text Finish Setup Go Live
Type here to search
32°C Haze
12:27 am
09/08/2025

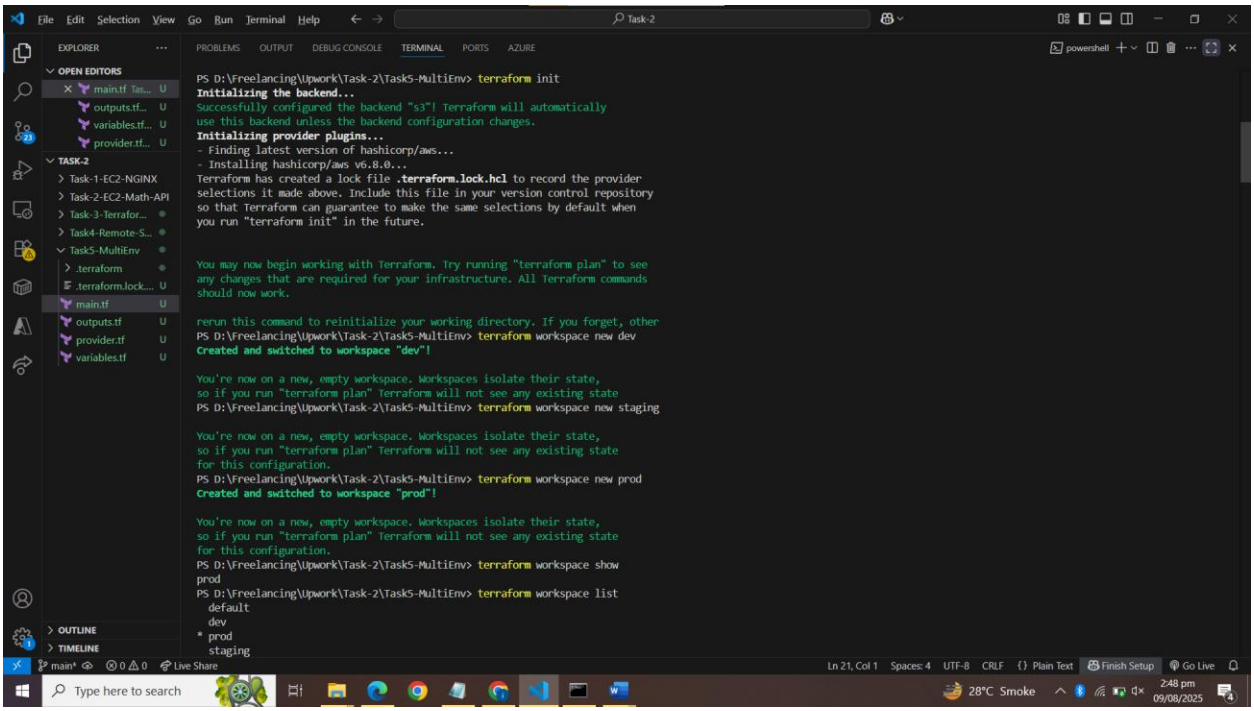

Command Prompt
C:\Users\Dell>aws s3 rb s3://insia-terraform-state-bucket-2025 --force
delete: s3://insia-terraform-state-bucket-2025/terraform.tfstate
remove_bucket failed: s3://insia-terraform-state-bucket-2025 An error occurred (AccessDenied) when calling the DeleteBucket operation: User: arn:aws:iam::400375465965:user/terraform-user is not authorized to per
form: s3:DeleteBucket on resource: "arn:aws:s3:::insia-terraform-state-bucket-2025" because no identity-based policy allows the s3:DeleteBucket action
C:\Users\Dell>
C:\Users\Dell>aws dynamodb delete-table --table-name terraform-locks --region us-east-1
{
"TableDescription": {
"TableName": "terraform-locks",
"TableStatus": "DELETING",
"ProvisionedThroughput": {
"NumberOfDecreasesToday": 0,
"ReadCapacityUnits": 0,
"WriteCapacityUnits": 0
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-east-1:400375465965:table/terraform-locks",
"TableId": "e481c49f-10ab-4a8e-b50d-3c8606adef1f",
"BillingModeSummary": {
"BillingMode": "PAY_PER_REQUEST",
"LastUpdateToPayPerRequestDateTime": "2025-08-09T00:01:37.178000+05:00"
},
"DeletionProtectionEnabled": false
}
}
C:\Users\Dell>
Type here to search
32°C Haze
12:26 am
09/08/2025

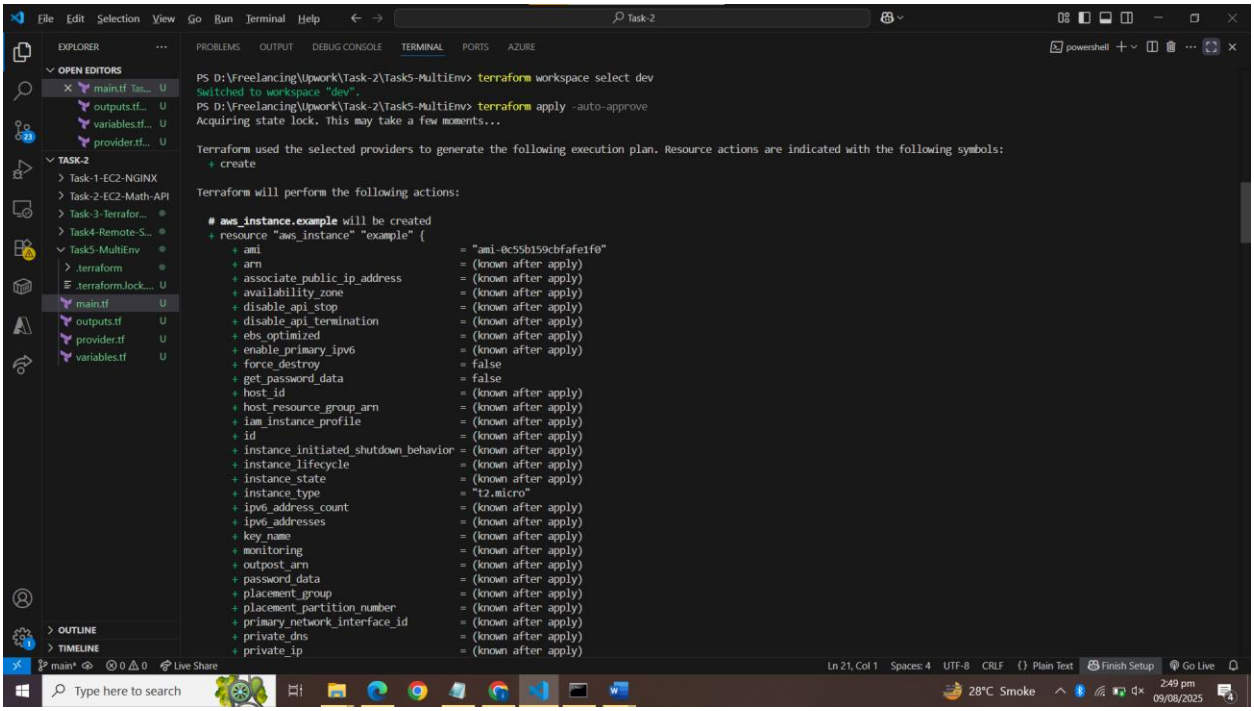
Task#5:

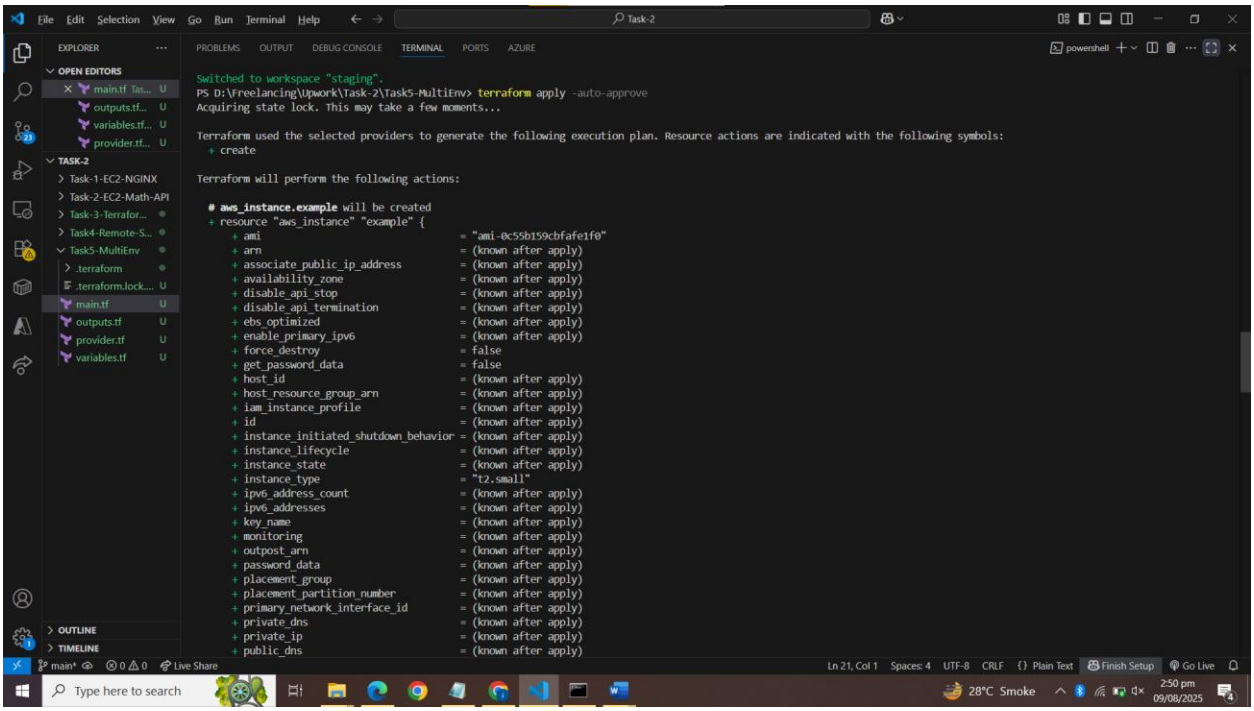
Task 5: Multi-Environment Support
Goal: Manage multiple deployment environments.
Why: Dev, staging, and prod environments require separation.
What You'll Learn: Using Terraform workspaces and conditional logic.
• Create workspaces for dev, staging, and prod.
• Use conditional variables to modify infrastructure per workspace.
• Deploy different instance types per environment.

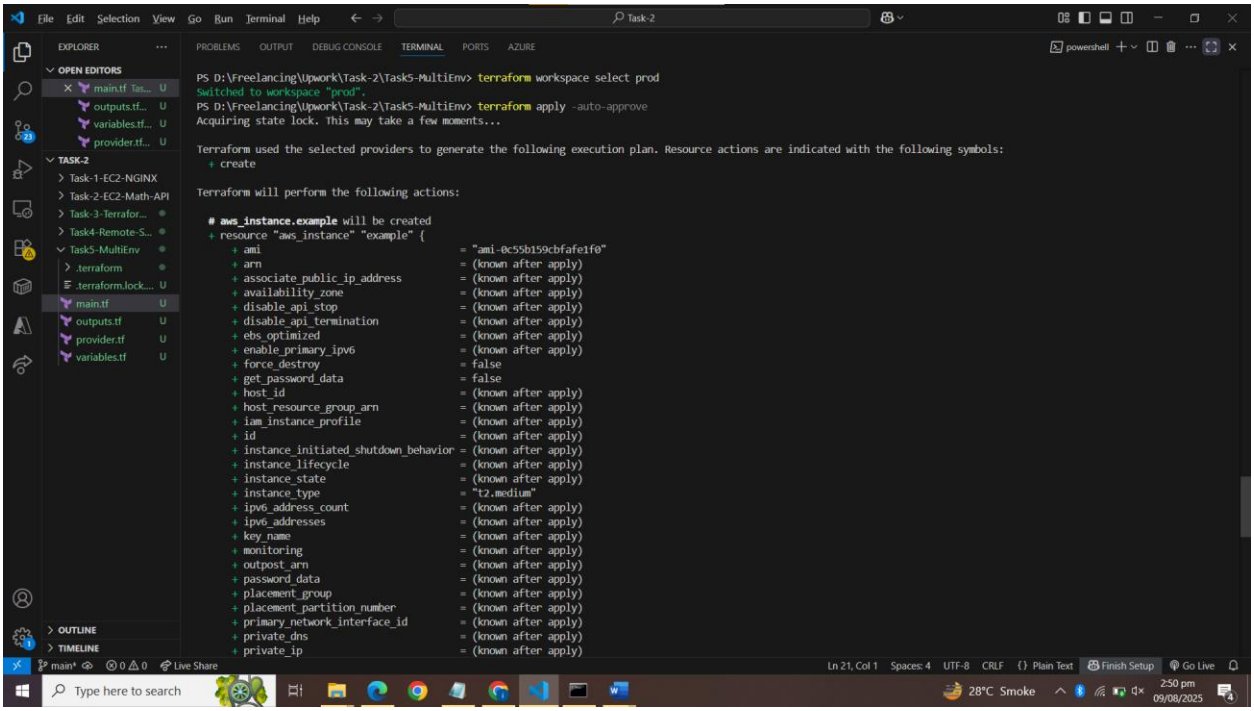


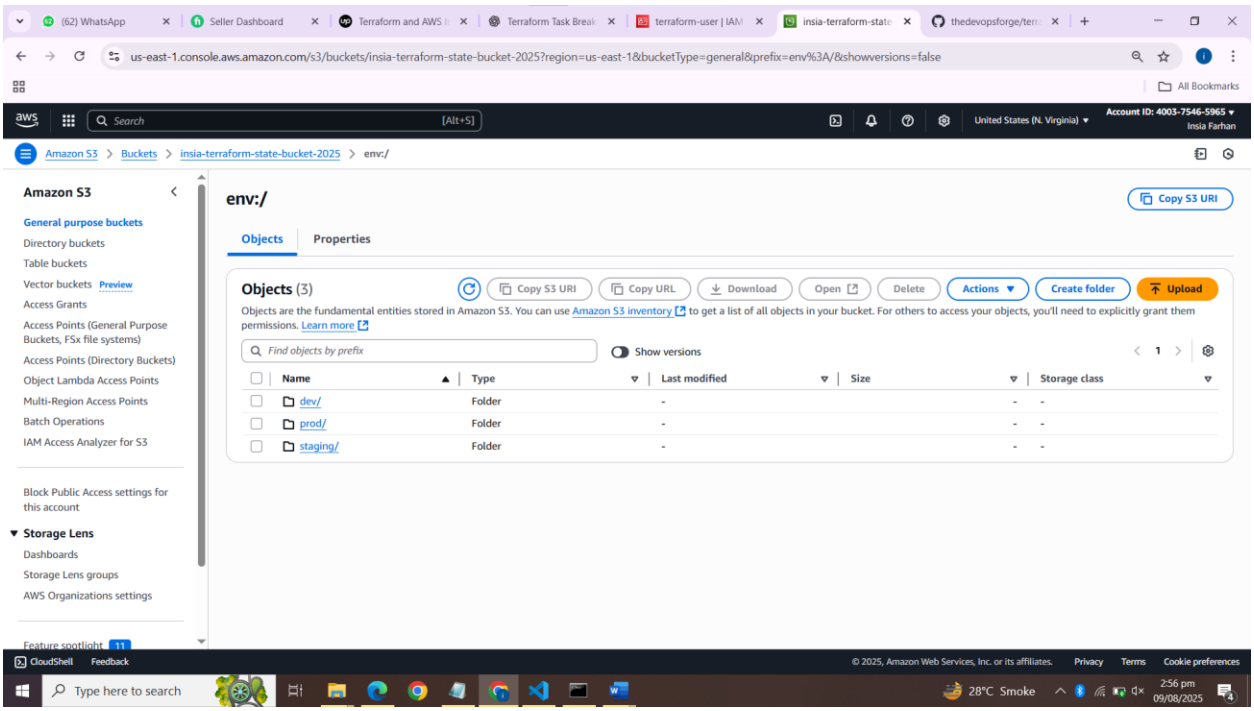


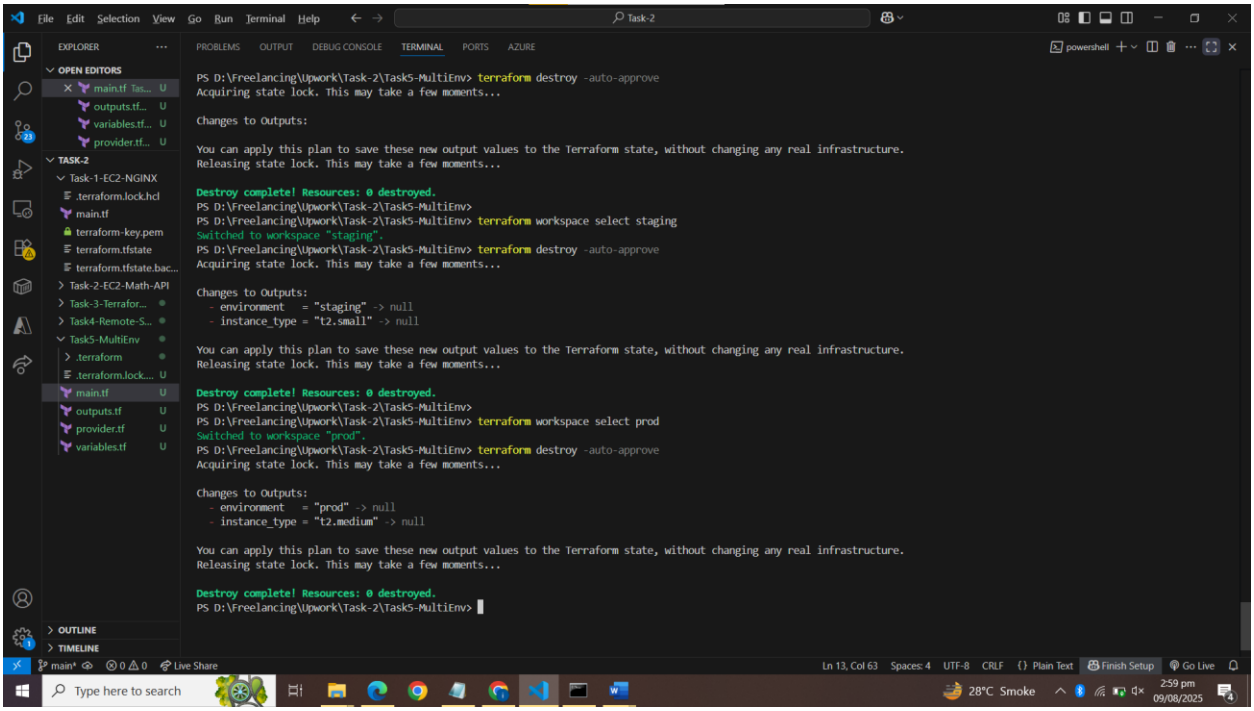

File Edit Selection View Go Run Terminal Help
Task-2
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
powershell
OPEN EDITORS
main.tf Tas... U
outputs.tf... U
variables.tf... U
provider.tf... U
TASK-2
Task-1-EC2-NGINX
Task-2-EC2-Math-API
Task-3-Terrafor...
Task4-Remote-S...
Task5-MultiEnv
.terraform
.terraform.lock.... U
main.tf U
outputs.tf U
provider.tf U
variables.tf U
OUTLINE
TIMELINE
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform init
Initializing the backend...
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.8.0...
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.
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.
rerun this command to reinitialize your working directory. If you forget, other
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace new dev
Created and switched to workspace "dev"!
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace new staging
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace new prod
Created and switched to workspace "prod"!
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace show
prod
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace list
default
dev
* prod
staging
main* 0 0 Live Share
Ln 21, Col 1 Spaces: 4 UTF-8 CRLF {} Plain Text Finish Setup Go Live
Type here to search
28°C Smoke
2:48 pm
09/08/2025


File Edit Selection View Go Run Terminal Help
Task-2
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
powershell
OPEN EDITORS
main.tf Tas... U
outputs.tf... U
variables.tf... U
provider.tf... U
TASK-2
Task-1-EC2-NGINX
Task-2-EC2-Math-API
Task-3-Terrafor...
Task4-Remote-S...
Task5-MultiEnv
.terraform
.terraform.lock.... U
main.tf U
outputs.tf U
provider.tf U
variables.tf U
OUTLINE
TIMELINE
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace select dev
Switched to workspace "dev".
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform apply -auto-approve
Acquiring state lock. This may take a few moments...
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_instance.example will be created
+ resource "aws_instance" "example" {
+ ami = "ami-0c55b159cbfafe1f0"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ force_destroy = false
+ 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 = (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)
main* 0 0 Live Share
Ln 21, Col 1 Spaces: 4 UTF-8 CRLF {} Plain Text Finish Setup Go Live
Type here to search
28°C Smoke
2:49 pm
09/08/2025


Switched to workspace "staging".
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform apply -auto-approve
Acquiring state lock. This may take a few moments...
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_instance.example will be created
+ resource "aws_instance" "example" {
+ ami = "ami-0c55b159cbfafe1f0"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ force_destroy = false
+ 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.small"
+ 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)


PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform workspace select prod
Switched to workspace "prod".
PS D:\Freelancing\Upwork\Task-2\Task5-MultiEnv> terraform apply -auto-approve
Acquiring state lock. This may take a few moments...
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_instance.example will be created
+ resource "aws_instance" "example" {
+ ami = "ami-0c55b159cbfafe1f0"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ force_destroy = false
+ 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.medium"
+ 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)





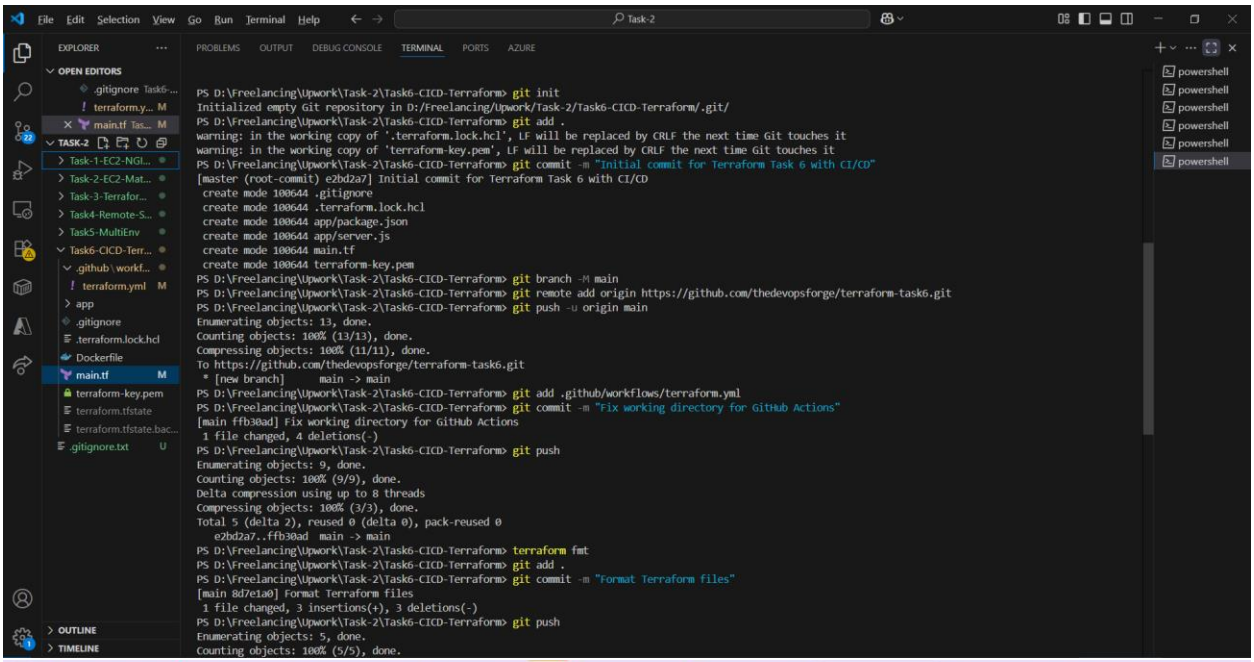
Destroy the S3 and DynamoDB as well.

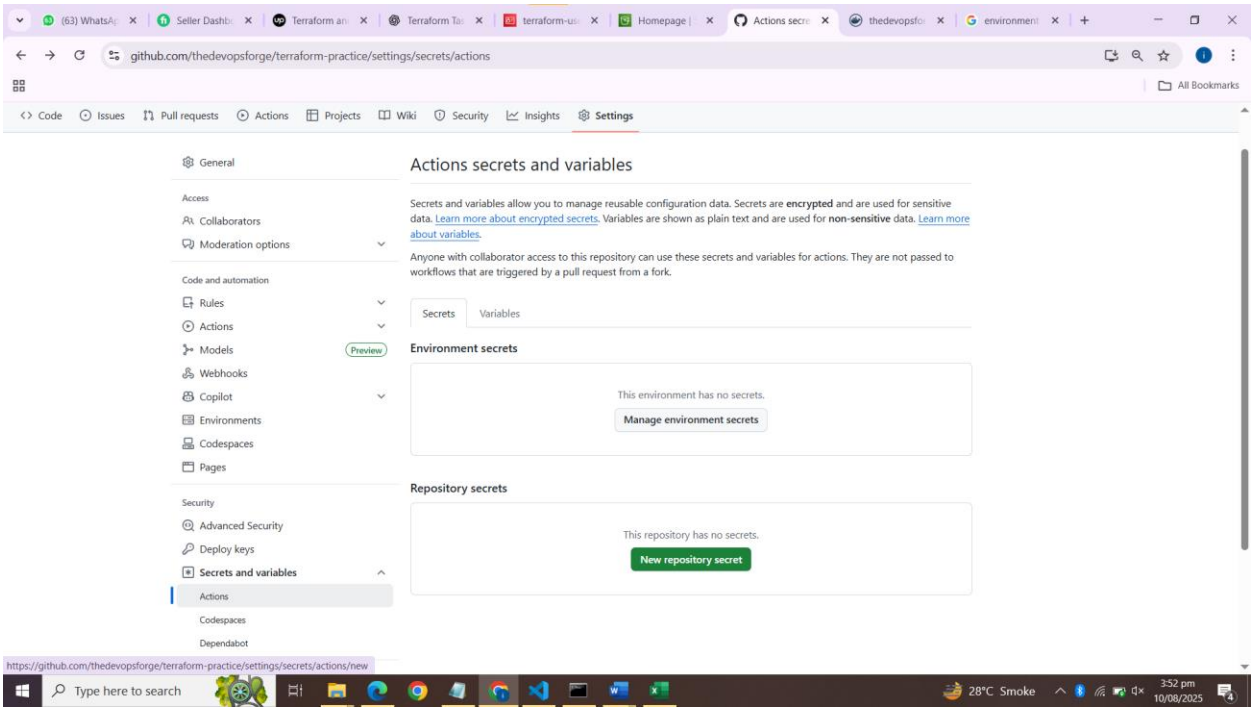
Task#6:

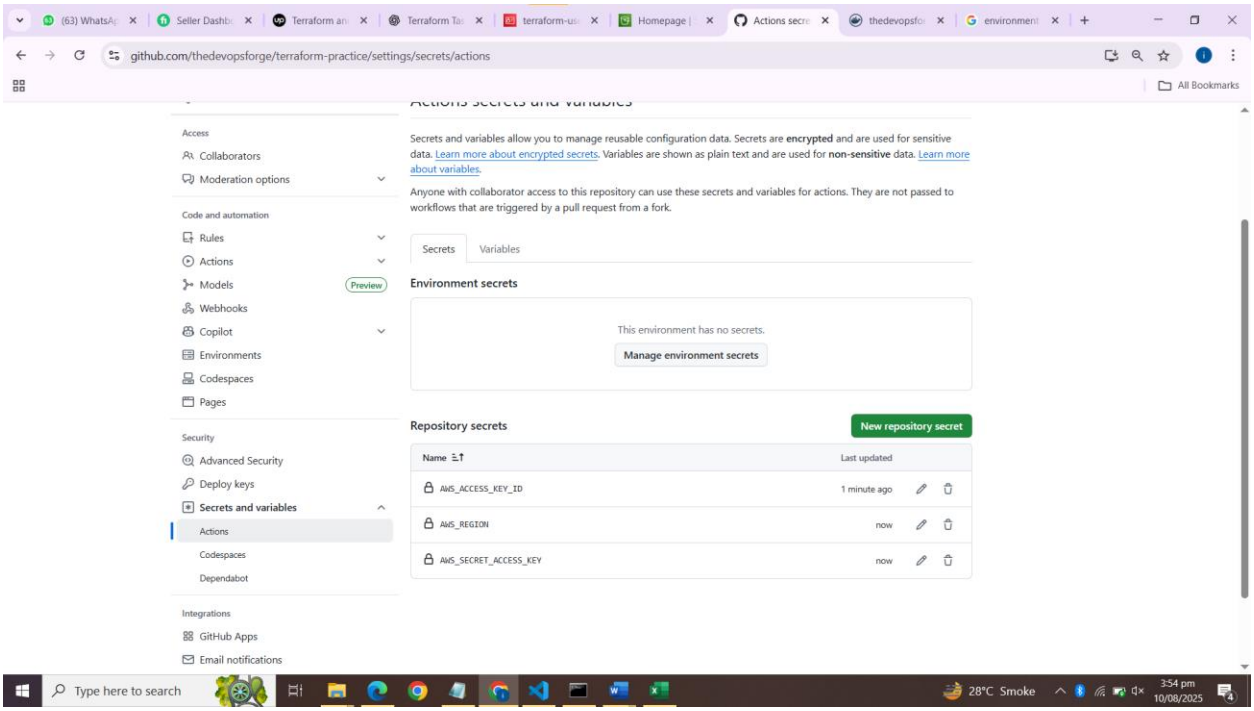
Task 6: CI/CD with GitHub Actions for Terraform
Goal: Automate your Terraform workflow with CI/CD.
Why: Automation ensures consistency and avoids manual errors.
What You'll Learn: Creating CI/CD pipelines, managing secrets, running plans and applies automatically.

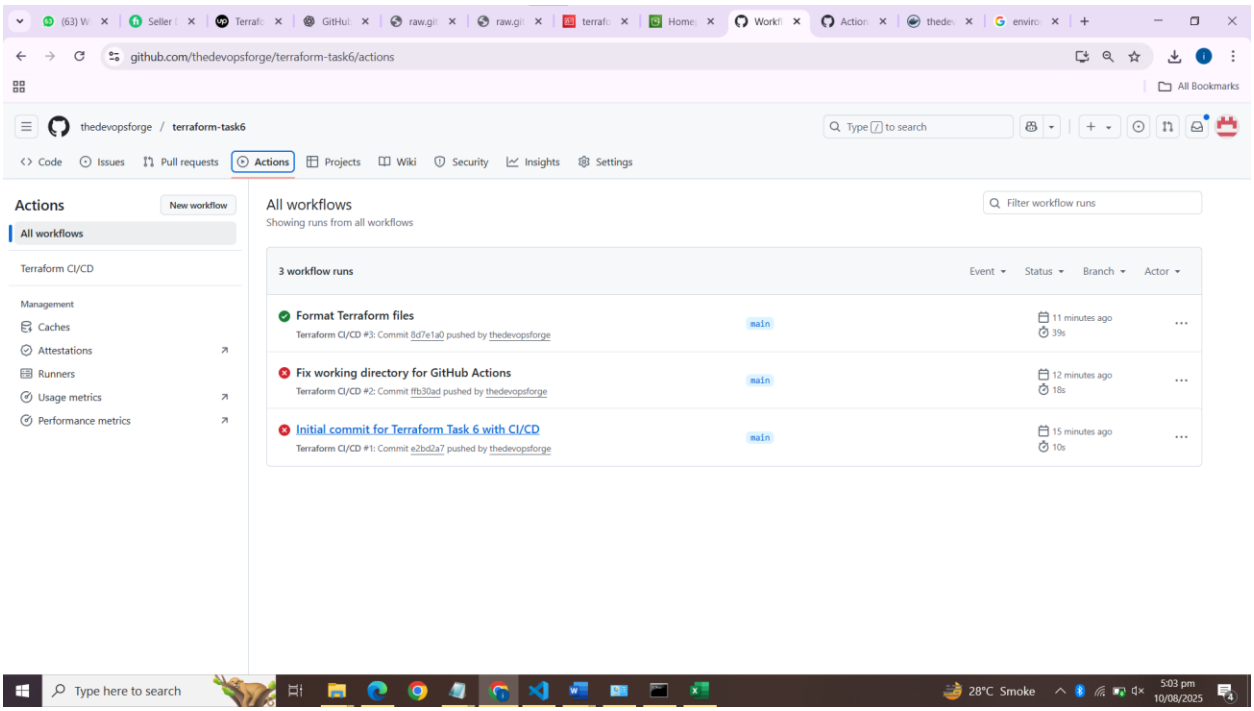
- *Write a GitHub Actions workflow to plan and apply Terraform scripts.*
- *Use GitHub Secrets to store credentials securely.*
- *Run Terraform plan on PRs, apply on main branch merges.*

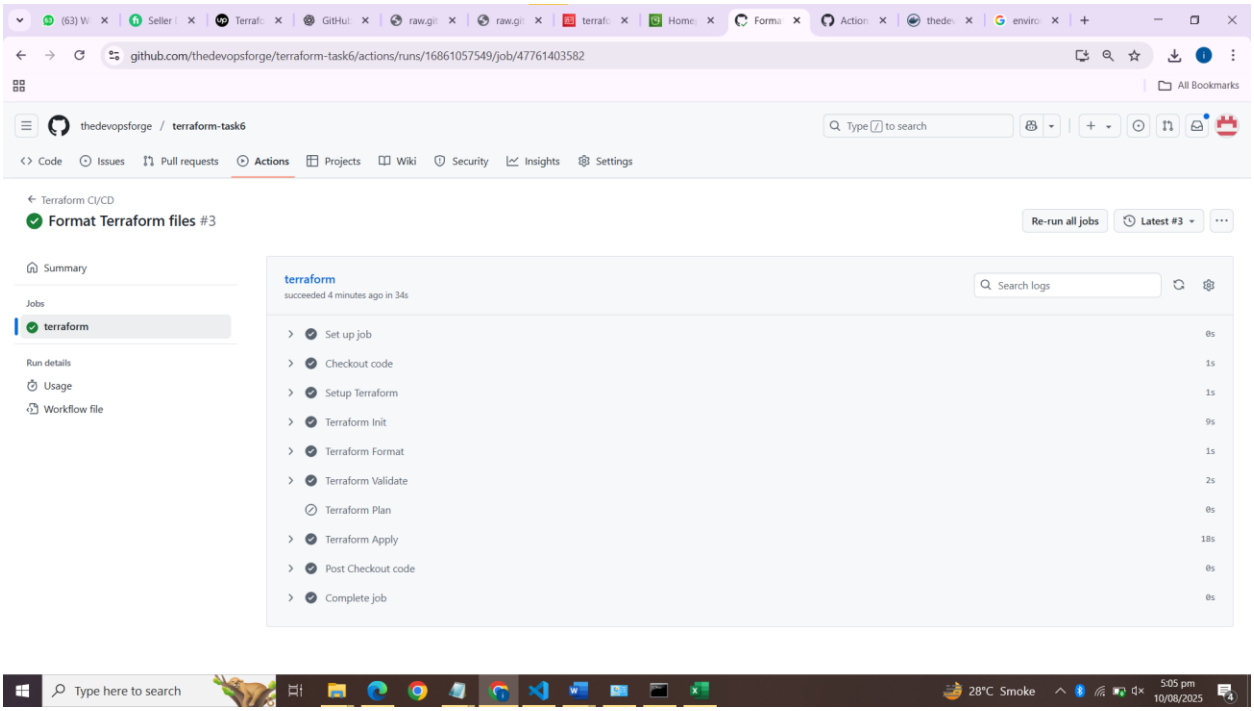
Using task 2s code in task6.

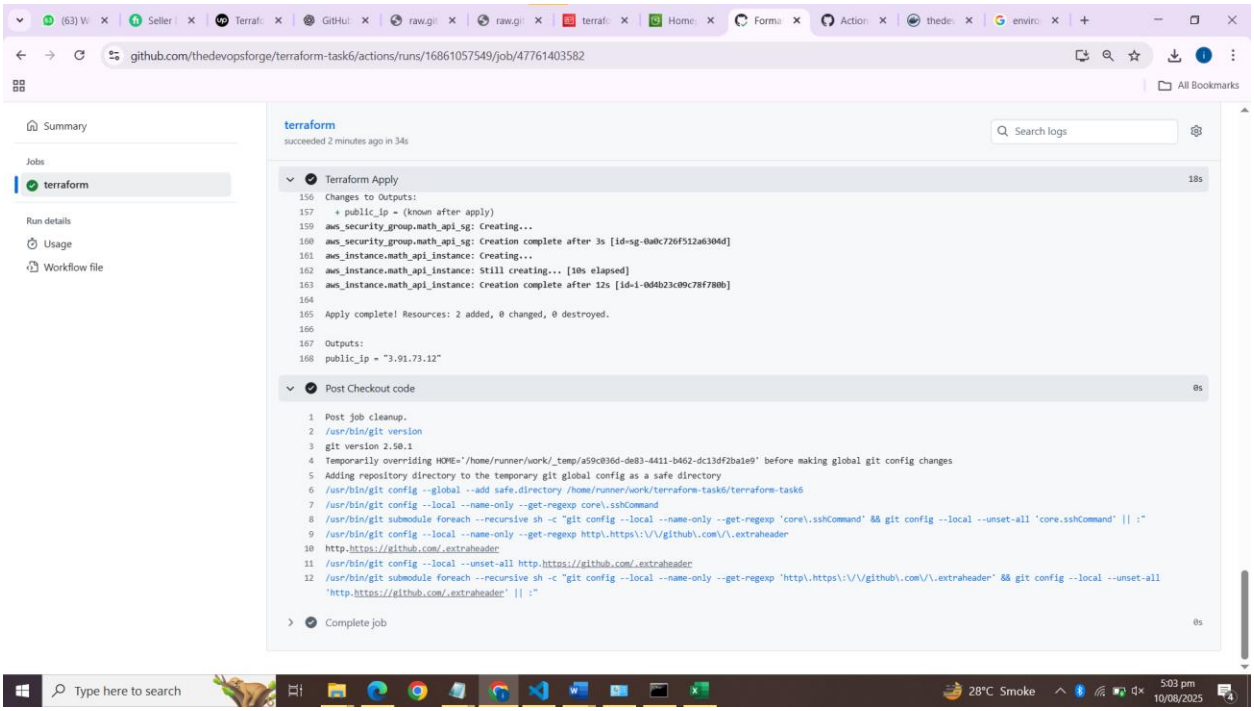


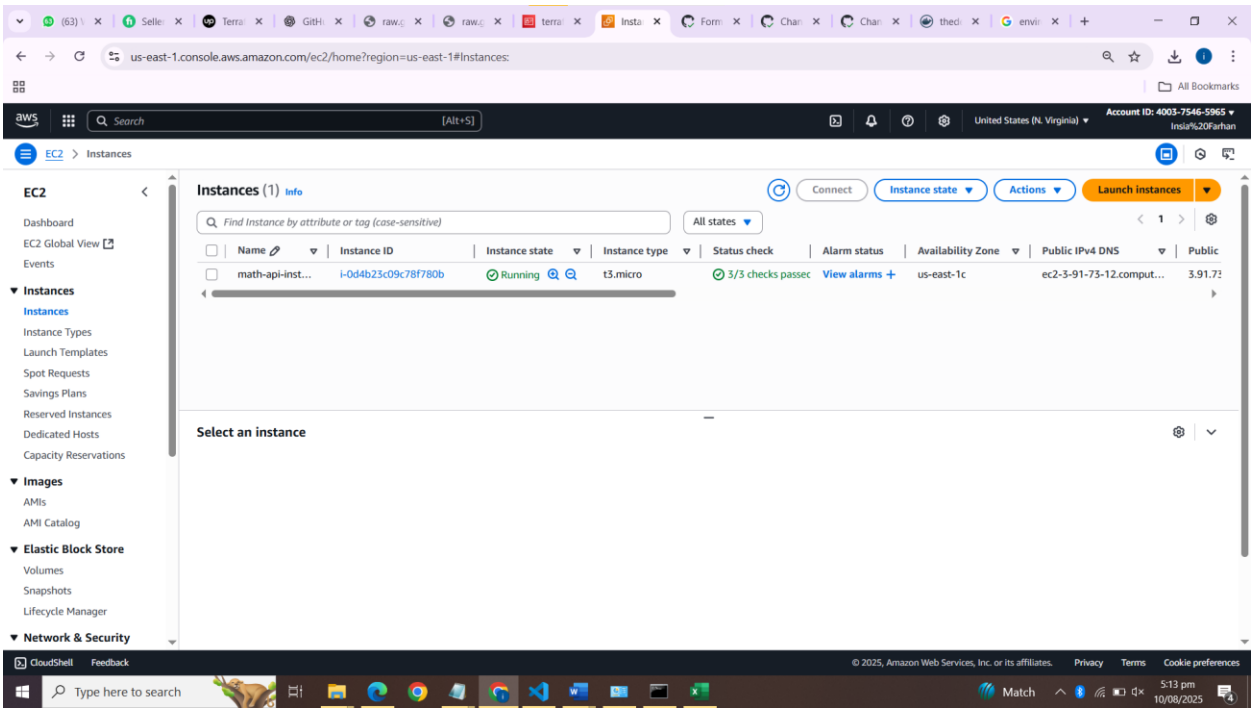

github.com/thedevopsforge/terraform-practice/settings/secrets/actions
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Rules
Actions
Models
Preview
Webhooks
Copilot
Environments
Codespaces
Pages
Security
Advanced Security
Deploy keys
Secrets and variables
Actions
Codespaces
Dependabot
Actions secrets and variables
Secrets and variables allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for non-sensitive data. Learn more about variables.
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.
Secrets
Variables
Environment secrets
This environment has no secrets.
Manage environment secrets
Repository secrets
This repository has no secrets.
New repository secret
https://github.com/thedevopsforge/terraform-practice/settings/secrets/actions/new
Type here to search
28°C Smoke
3:52 pm
10/08/2025

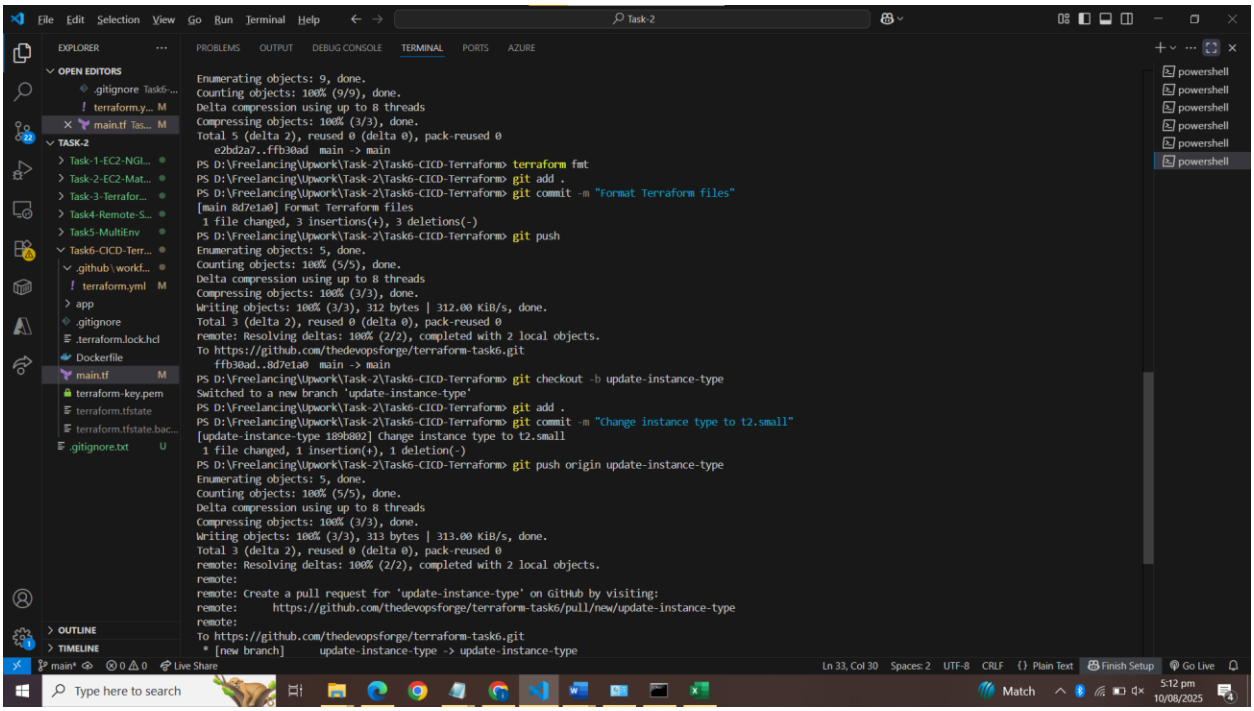

github.com/thedevopsforge/terraform-practice/settings/secrets/actions
Access
Collaborators
Moderation options
Code and automation
Rules
Actions
Models
Preview
Webhooks
Copilot
Environments
Codespaces
Pages
Security
Advanced Security
Deploy keys
Secrets and variables
Actions
Codespaces
Dependabot
Integrations
GitHub Apps
Email notifications
Secrets and variables allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for non-sensitive data. Learn more about variables.
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.
Secrets
Variables
Environment secrets
This environment has no secrets.
Manage environment secrets
Repository secrets
New repository secret
Name
Last updated
AWS_ACCESS_KEY_ID
1 minute ago
AWS_REGION
now
AWS_SECRET_ACCESS_KEY
now
Type here to search
28°C Smoke
3:54 pm
10/08/2025

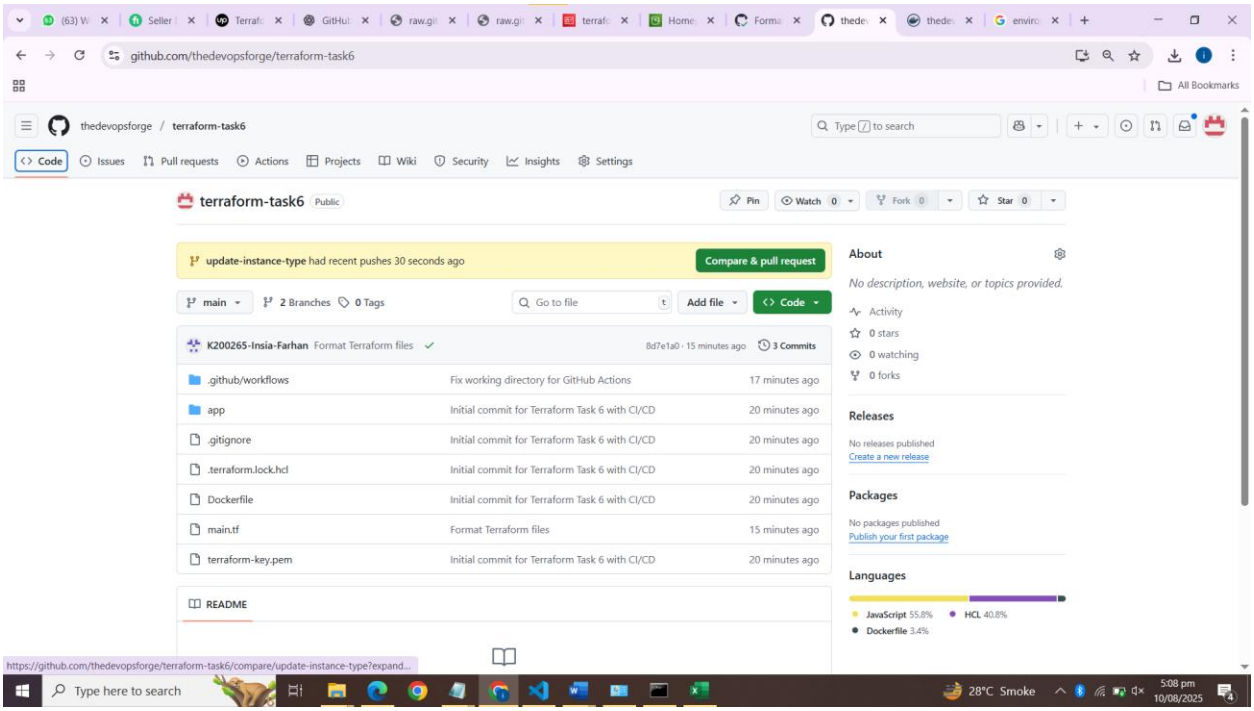

github.com/thedevopsforge/terraform-task6/actions
thedevopsforge / terraform-task6
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
Terraform CI/CD
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Filter workflow runs
3 workflow runs
Event
Status
Branch
Actor
Format Terraform files
Terraform CI/CD #3: Commit 8d7e1a0 pushed by thedevopsforge
main
11 minutes ago
39s
Fix working directory for GitHub Actions
Terraform CI/CD #2: Commit ffb30ad pushed by thedevopsforge
main
12 minutes ago
18s
Initial commit for Terraform Task 6 with CI/CD
Terraform CI/CD #1: Commit e2bd2a7 pushed by thedevopsforge
main
15 minutes ago
10s
Type here to search
28°C Smoke
5:03 pm
10/08/2025

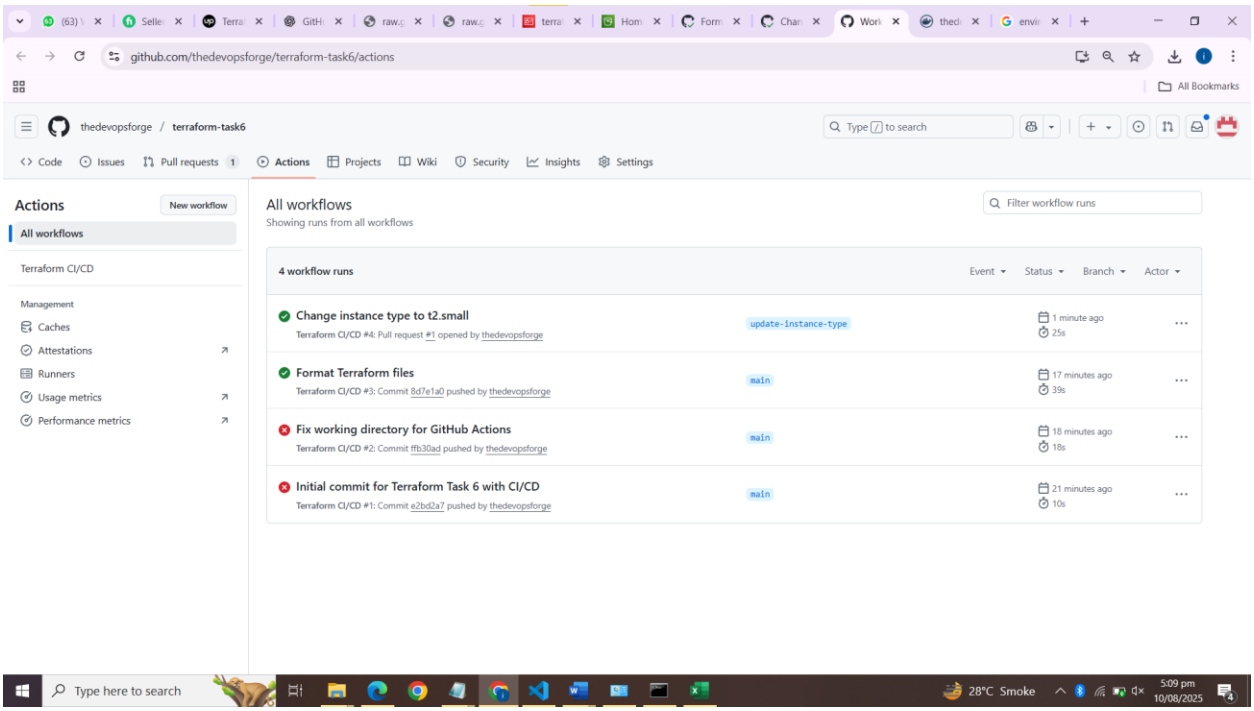

github.com/thedevopsforge/terraform-task6/actions/runs/16861057549/job/47761403582
thedevopsforge / terraform-task6
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Terraform CI/CD
Format Terraform files #3
Re-run all jobs
Latest #3
Summary
Jobs
terraform
Run details
Usage
Workflow file
terraform
succeeded 4 minutes ago in 34s
Search logs
Set up job 0s
Checkout code 1s
Setup Terraform 1s
Terraform Init 9s
Terraform Format 1s
Terraform Validate 2s
Terraform Plan 0s
Terraform Apply 18s
Post Checkout code 0s
Complete job 0s
Type here to search
28°C Smoke
5:05 pm
10/08/2025

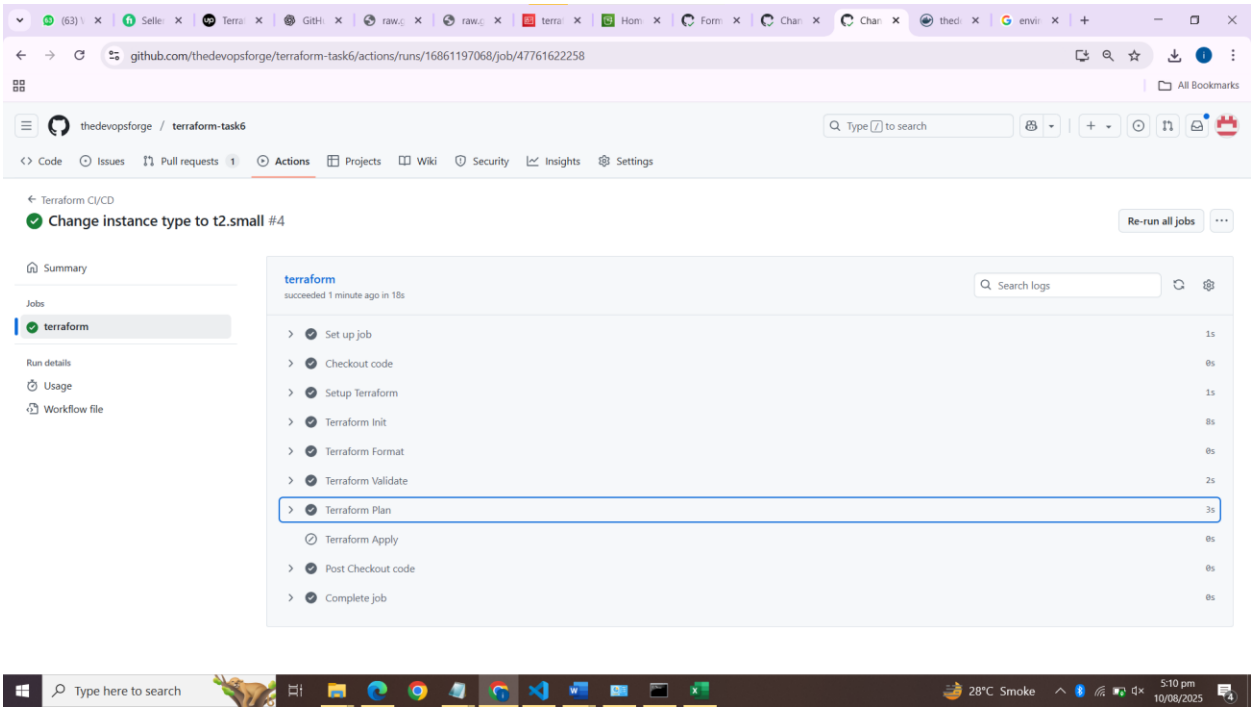

github.com/thedevopsforge/terraform-task6/actions/runs/16861057549/job/47761403582
Summary
Jobs
terraform
Run details
Usage
Workflow file
terraform
succeeded 2 minutes ago in 34s
Search logs
Terraform Apply 18s
Changes to Outputs:
+ public_ip = (known after apply)
aws_security_group.math_api_sg: Creating...
aws_security_group.math_api_sg: Creation complete after 3s [id=sg-0a0c726f512a6304d]
aws_instance.math_api_instance: Creating...
aws_instance.math_api_instance: Still creating... [10s elapsed]
aws_instance.math_api_instance: Creation complete after 12s [id=i-0d4b23c09c78f780b]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Outputs:
public_ip = "3.91.73.12"
Post Checkout code 0s
Post job cleanup.
/usr/bin/git version
git version 2.50.1
Temporarily overriding HOME='/home/runner/work/_temp/a59c036d-de83-4411-b462-dc13df2ba1e9' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/terraform-task6/terraform-task6
/usr/bin/git config --local --name-only --get-regexp core\.sshCommand
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core\.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
/usr/bin/git config --local --name-only --get-regexp http\.https\:\/\/github\.com\/\.extraheader
http.https://github.com/.extraheader
/usr/bin/git config --local --unset-all http.https://github.com/.extraheader
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http\.https\:\/\/github\.com\/\.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"
Complete job 0s
Type here to search
28°C Smoke
5:03 pm
10/08/2025


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Search [Alt+S]
United States (N. Virginia)
Account ID: 4003-7546-5965
Insia%20Farhan
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public
math-api-inst...
i-0d4b23c09c78f780b
Running
t3.micro
3/3 checks passed
View alarms
us-east-1c
ec2-3-91-73-12.comput...
3.91.7
Select an instance
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Match
5:13 pm
10/08/2025


Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
e2bd2a7..ffb30ad main -> main
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> terraform fmt
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git add .
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git commit -m "Format Terraform files"
[main 8d7e1a0] Format Terraform files
1 file changed, 3 insertions(+), 3 deletions(-)
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/thedevopsforge/terraform-task6.git
ffb30ad..8d7e1a0 main -> main
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git checkout -b update-instance-type
Switched to a new branch 'update-instance-type'
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git add .
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git commit -m "Change instance type to t2.small"
[update-instance-type 189b802] Change instance type to t2.small
1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\Freelancing\Upwork\Task-2\Task6-CICD-Terraform> git push origin update-instance-type
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 313 bytes | 313.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'update-instance-type' on GitHub by visiting:
remote: https://github.com/thedevopsforge/terraform-task6/pull/new/update-instance-type
remote:
To https://github.com/thedevopsforge/terraform-task6.git
* [new branch] update-instance-type -> update-instance-type


github.com/thedevopsforge/terraform-task6
thedevopsforge / terraform-task6
terraform-task6
Public
update-instance-type had recent pushes 30 seconds ago
Compare & pull request
About
No description, website, or topics provided.
K200265-Insia-Farhan Format Terraform files
3 Commits
.github/workflows
app
.gitignore
.terraform.lock.hcl
Dockerfile
main.tf
terraform-key.pem
README
Releases
No releases published
Packages
No packages published
Languages
JavaScript 55.8%
HCL 40.8%
Dockerfile 3.4%


github.com/thedevopsforge/terraform-task6/actions
thedevopsforge / terraform-task6
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
Terraform CI/CD
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Filter workflow runs
4 workflow runs
Event
Status
Branch
Actor
Change instance type to t2.small
Terraform CI/CD #4: Pull request #1 opened by thedevopsforge
update-instance-type
1 minute ago
25s
Format Terraform files
Terraform CI/CD #3: Commit 8d7e1a0 pushed by thedevopsforge
main
17 minutes ago
39s
Fix working directory for GitHub Actions
Terraform CI/CD #2: Commit ffb30ad pushed by thedevopsforge
main
18 minutes ago
18s
Initial commit for Terraform Task 6 with CI/CD
Terraform CI/CD #1: Commit e2bd2a7 pushed by thedevopsforge
main
21 minutes ago
10s
Type here to search
28°C Smoke
5:09 pm
10/08/2025


github.com/thedevopsforge/terraform-task6/actions/runs/16861197068/job/47761622258
thedevopsforge / terraform-task6
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
Terraform CI/CD
Change instance type to t2.small #4
Re-run all jobs
Summary
Jobs
terraform
Run details
Usage
Workflow file
terraform
succeeded 1 minute ago in 18s
Search logs
Set up job 1s
Checkout code 0s
Setup Terraform 1s
Terraform Init 8s
Terraform Format 0s
Terraform Validate 2s
Terraform Plan 3s
Terraform Apply 0s
Post Checkout code 0s
Complete job 0s
Type here to search
28°C Smoke
5:10 pm
10/08/2025