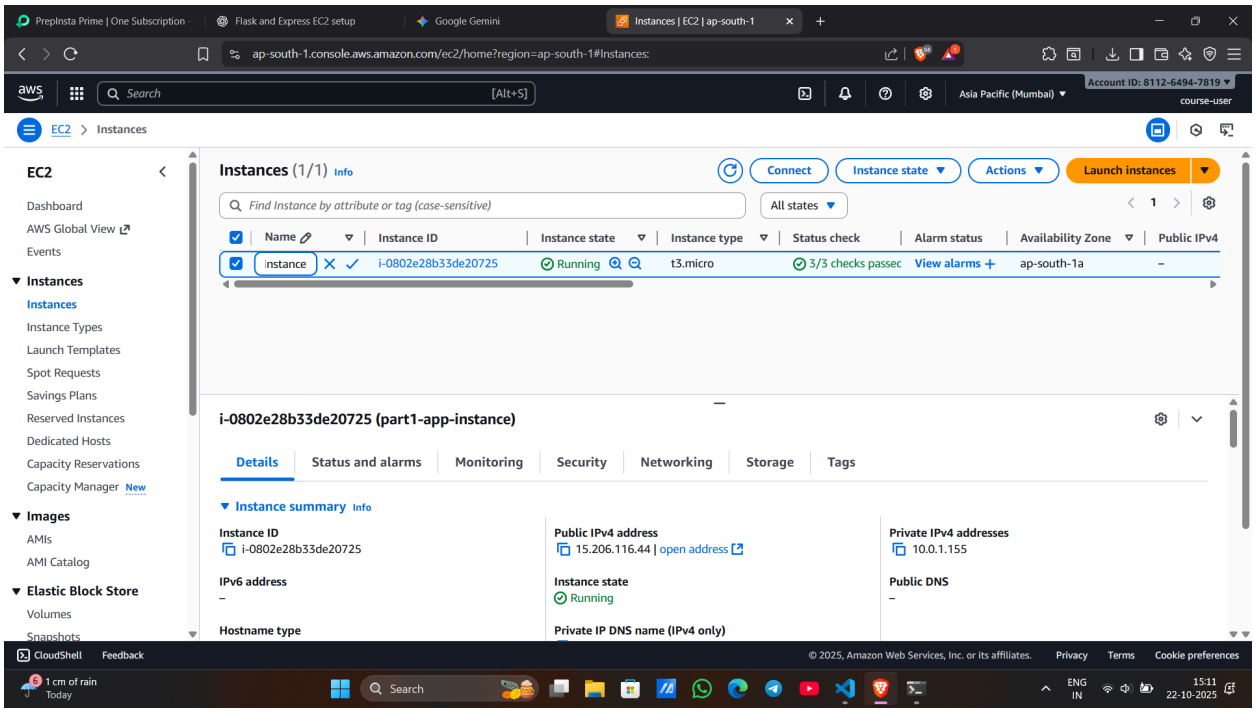
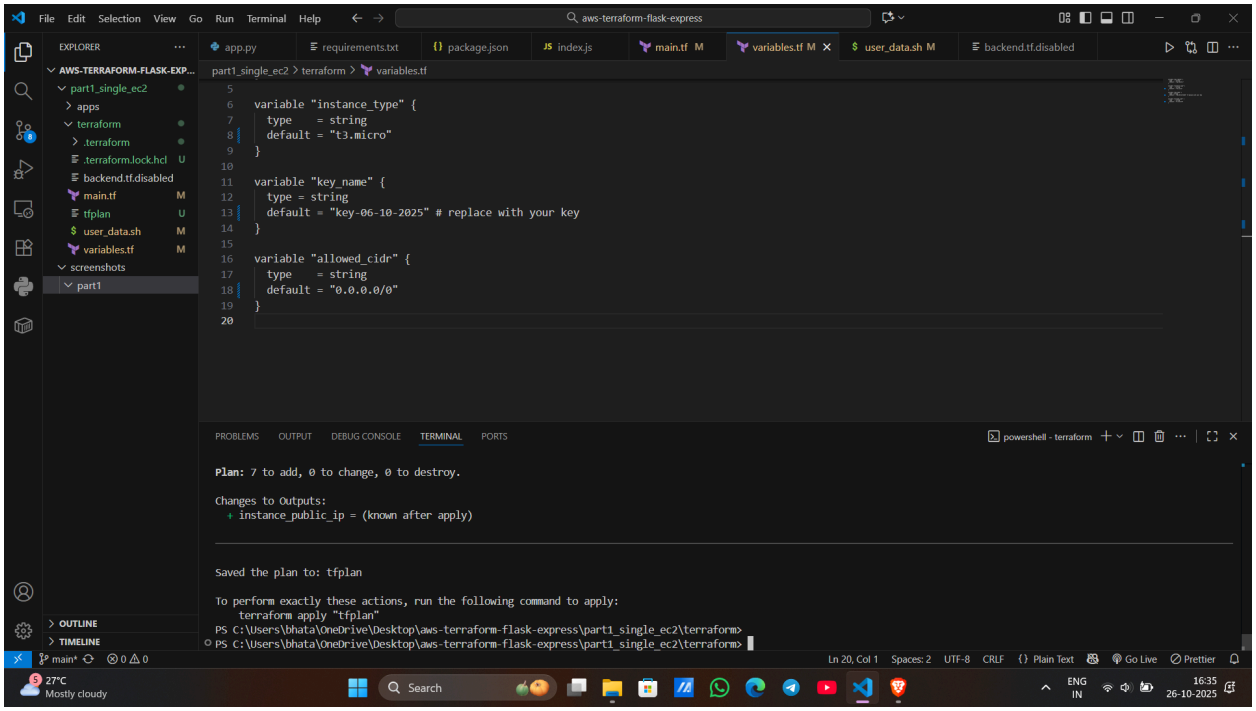
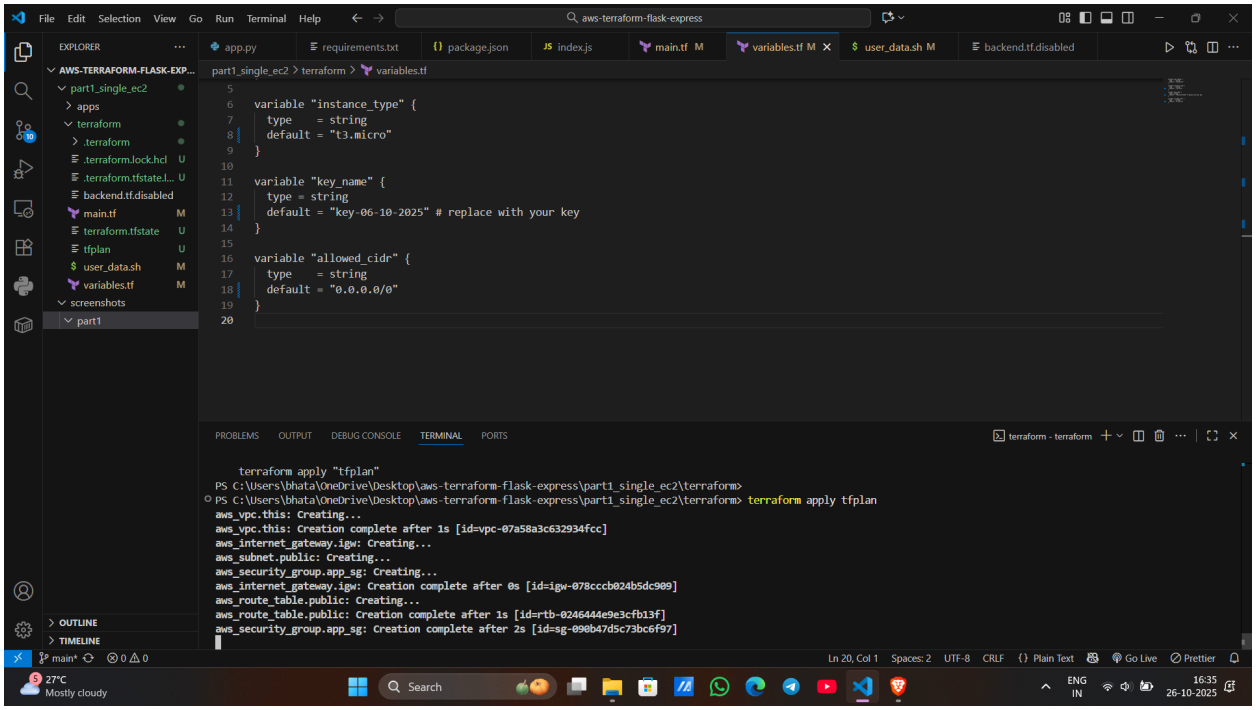
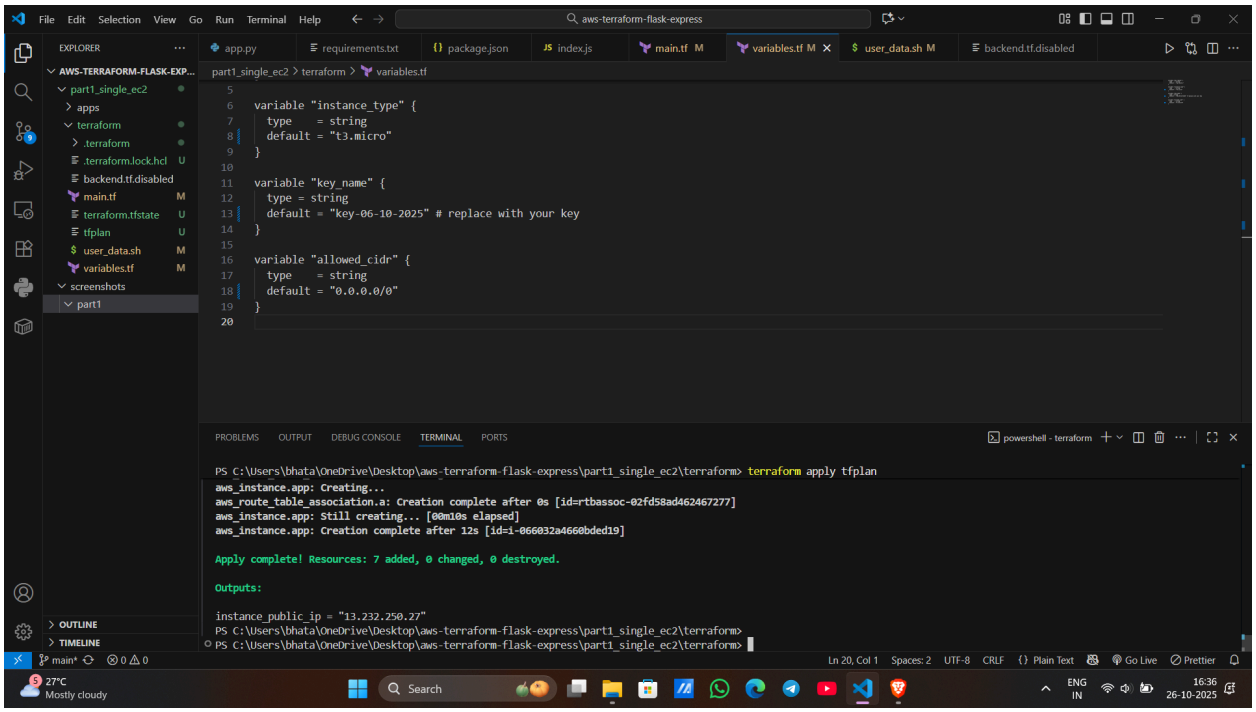
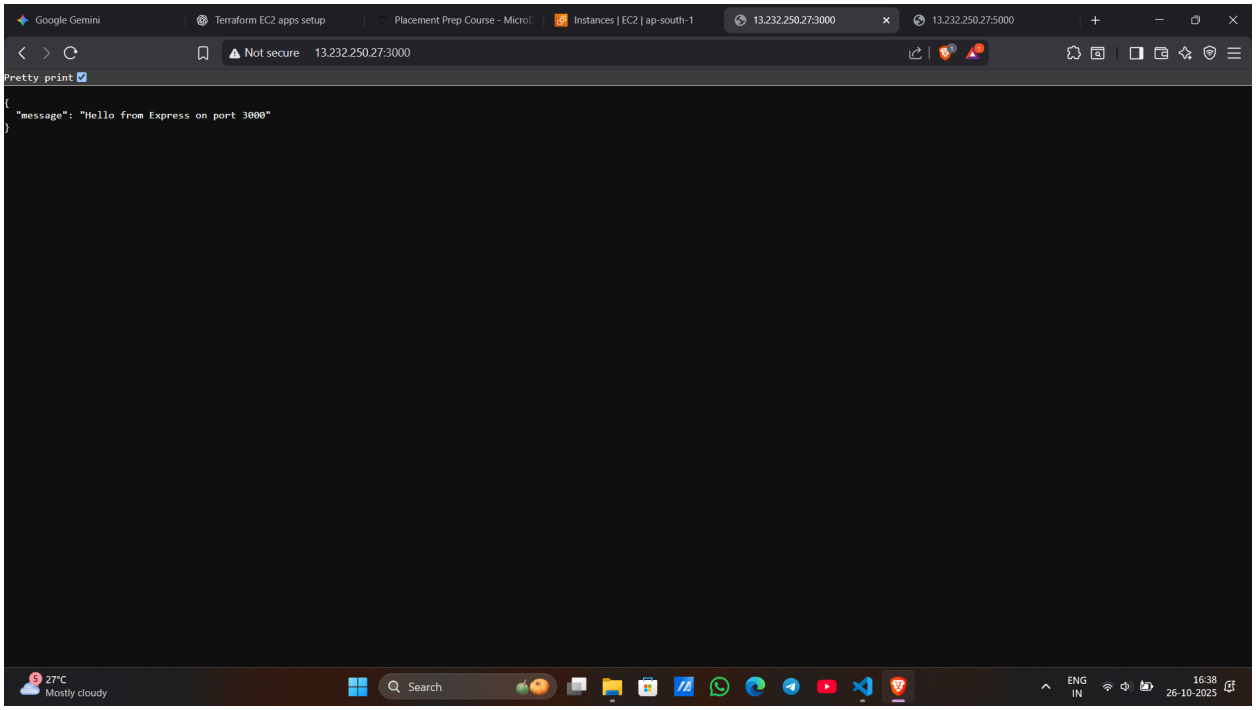
GitHub Repo: https://github.com/bhatadarsh/aws-terraform-flask-express
Part1:

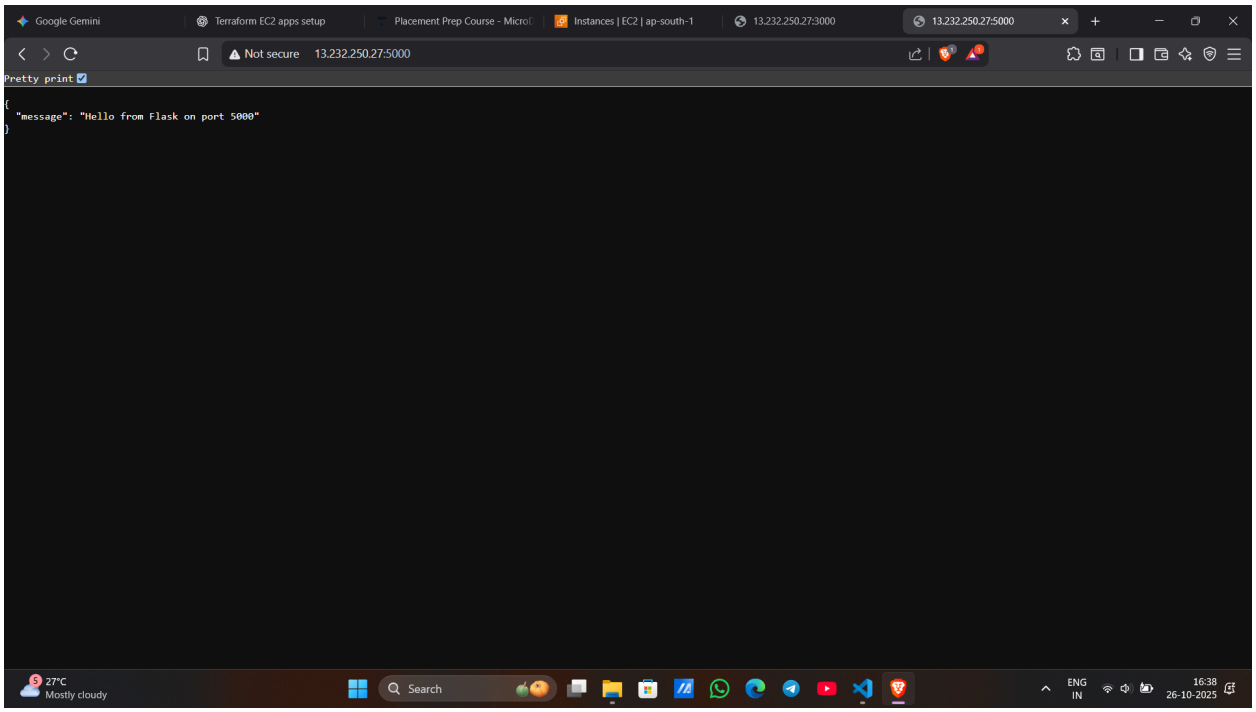





File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
apps
terraform
.terraform
.terraform.lock.hcl
.terraform.tfstate.l...
backend.tf.disabled
main.tf
terraform.tfstate
tfplan
user_data.sh
variables.tf
screenshots
part1
app.py
requirements.txt
package.json
index.js
main.tf M
variables.tf M
user_data.sh M
backend.tf.disabled
part1_single_ec2 > terraform > variables.tf
variable "instance_type" {
type = string
default = "t3.micro"
}
variable "key_name" {
type = string
default = "key-06-10-2025" # replace with your key
}
variable "allowed_cidr" {
type = string
default = "0.0.0.0/0"
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
terraform - terraform
terraform apply "tfplan"
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part1_single_ec2\terraform>
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part1_single_ec2\terraform> terraform apply tfplan
aws_vpc.this: Creating...
aws_vpc.this: Creation complete after 1s [id=vpc-07a58a3c632934fcc]
aws_internet_gateway.igw: Creating...
aws_subnet.public: Creating...
aws_security_group.app_sg: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-078cccb024b5dc909]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-0246444e9e3cfb13f]
aws_security_group.app_sg: Creation complete after 2s [id=sg-090b47d5c73bc6f97]
OUTLINE
TIMELINE
main*
Ln 20, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
27°C
Mostly cloudy
Search
ENG
IN
16:35
26-10-2025

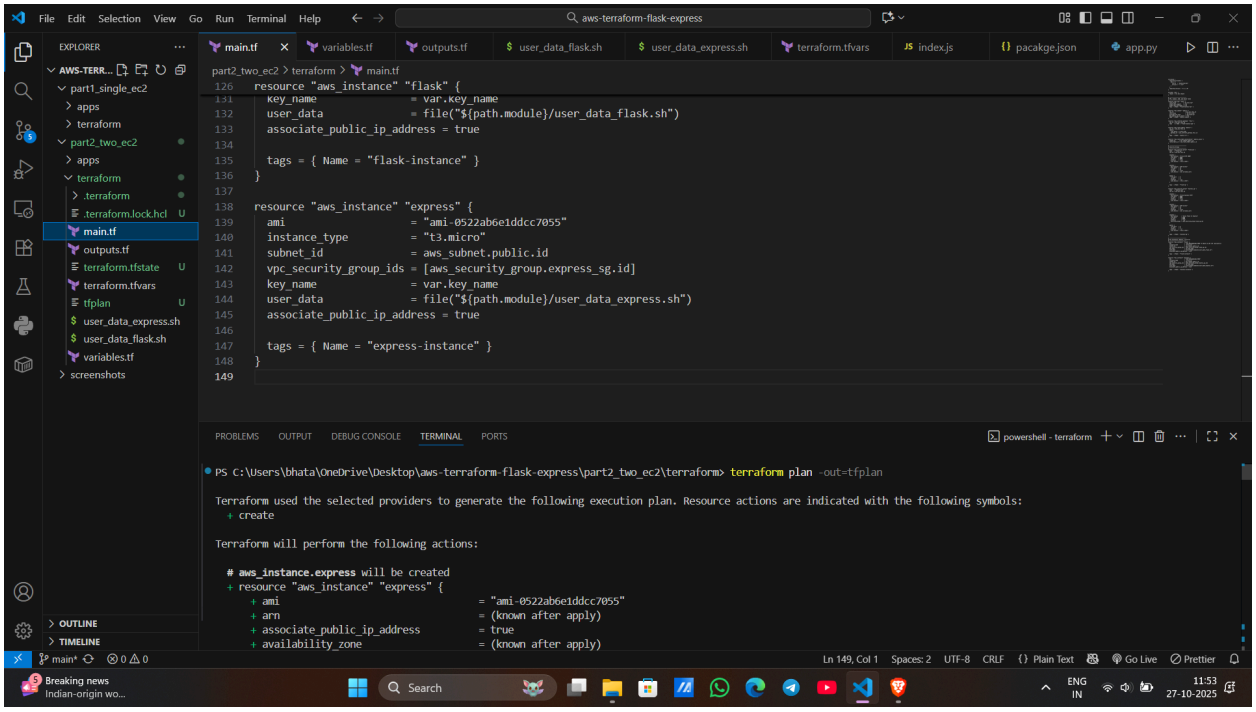

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
apps
terraform
.terraform
.terraform.lock.hcl
backend.tf.disabled
main.tf
terraform.tfstate
tfplan
user_data.sh
variables.tf
screenshots
part1
app.py
requirements.txt
package.json
index.js
main.tf M
variables.tf M
user_data.sh M
backend.tf.disabled
part1_single_ec2 > terraform > variables.tf
variable "instance_type" {
type = string
default = "t3.micro"
}
variable "key_name" {
type = string
default = "key-06-10-2025" # replace with your key
}
variable "allowed_cidr" {
type = string
default = "0.0.0.0/0"
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - terraform
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part1_single_ec2\terraform> terraform apply tfplan
aws_instance.app: Creating...
aws_route_table_association.a: Creation complete after 0s [id=rtbassoc-02fd58ad462467277]
aws_instance.app: Still creating... [00m10s elapsed]
aws_instance.app: Creation complete after 12s [id=i-066032a4660bded19]
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
Outputs:
instance_public_ip = "13.232.250.27"
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part1_single_ec2\terraform>
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part1_single_ec2\terraform>
OUTLINE
TIMELINE
main*
Ln 20, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
27°C
Mostly cloudy
Search
ENG
IN
16:36
26-10-2025

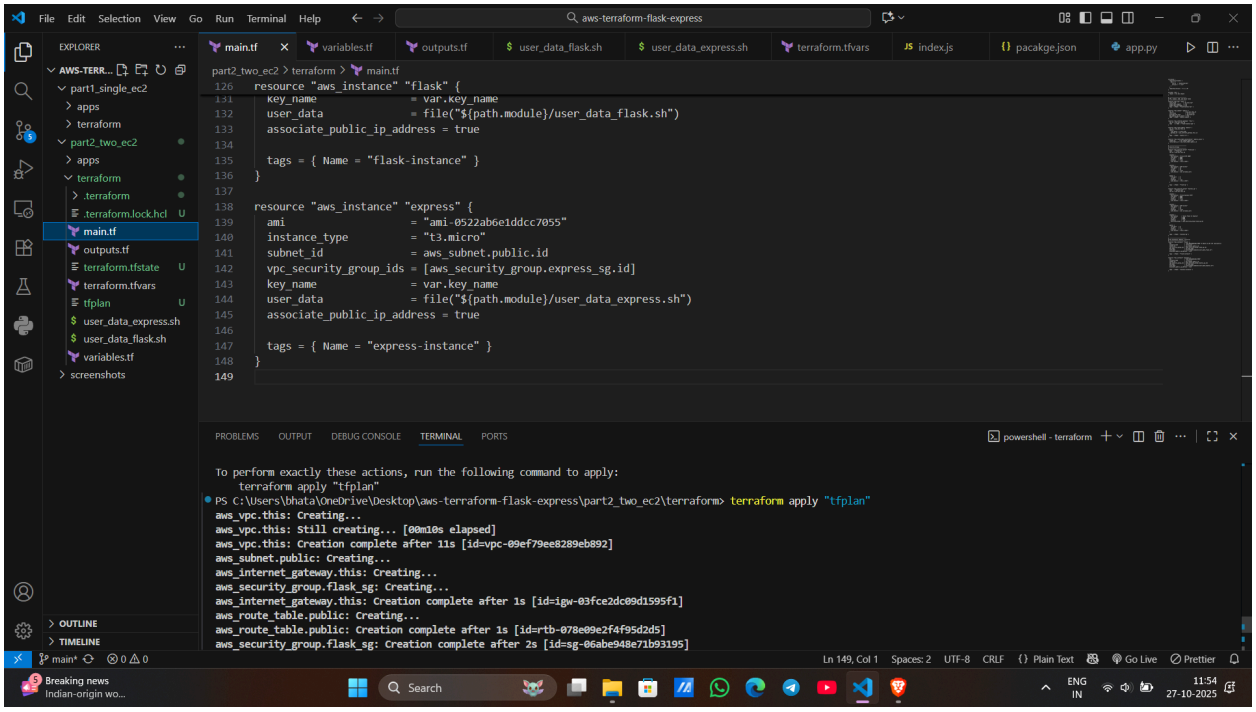


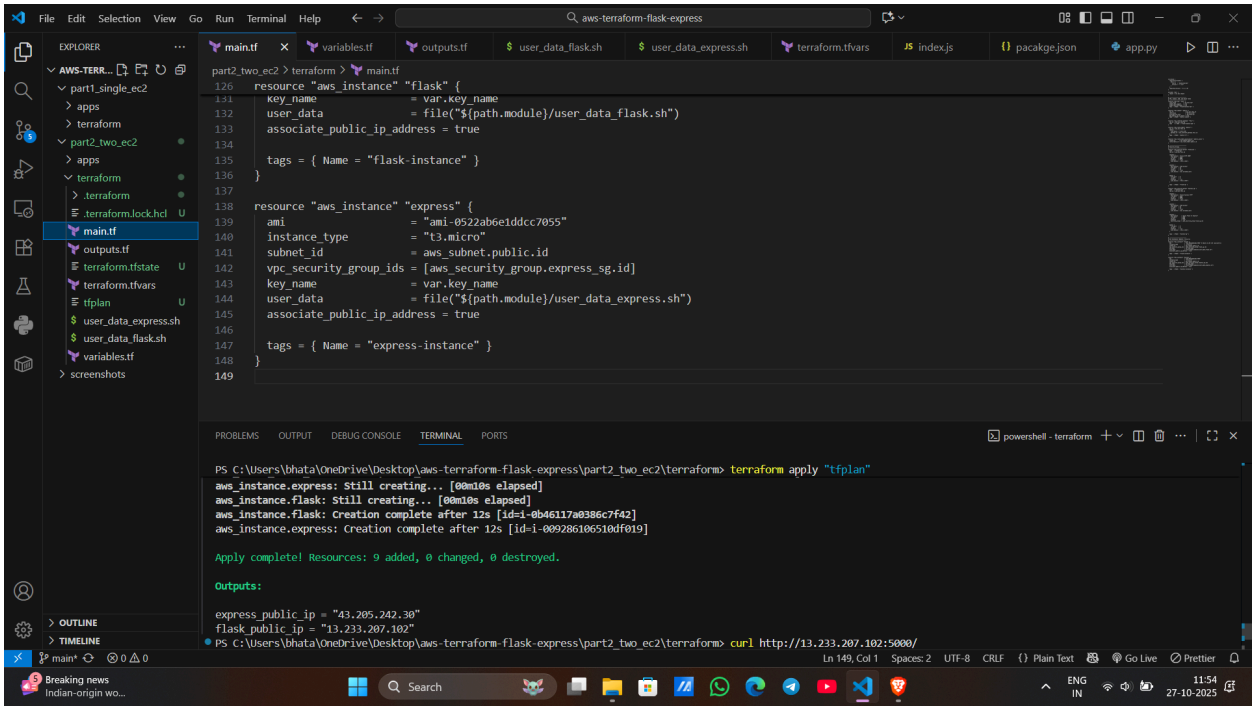


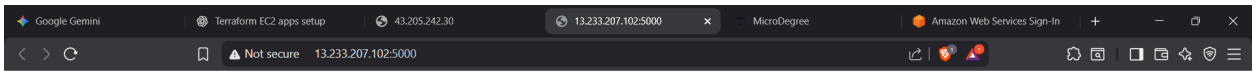
Explanation: In this task, I deployed both the Flask backend and Express frontend on a single EC2 instance using Terraform. The instance was provisioned with a user data script that installed Python, Node.js, and other dependencies, and started both applications — Flask running on port 5000 and Express on port 3000. Security groups were configured to allow traffic on these ports, and once the instance was launched, both applications were accessible via the EC2 public IP. This demonstrated successful deployment and infrastructure automation using Terraform.

Part2:

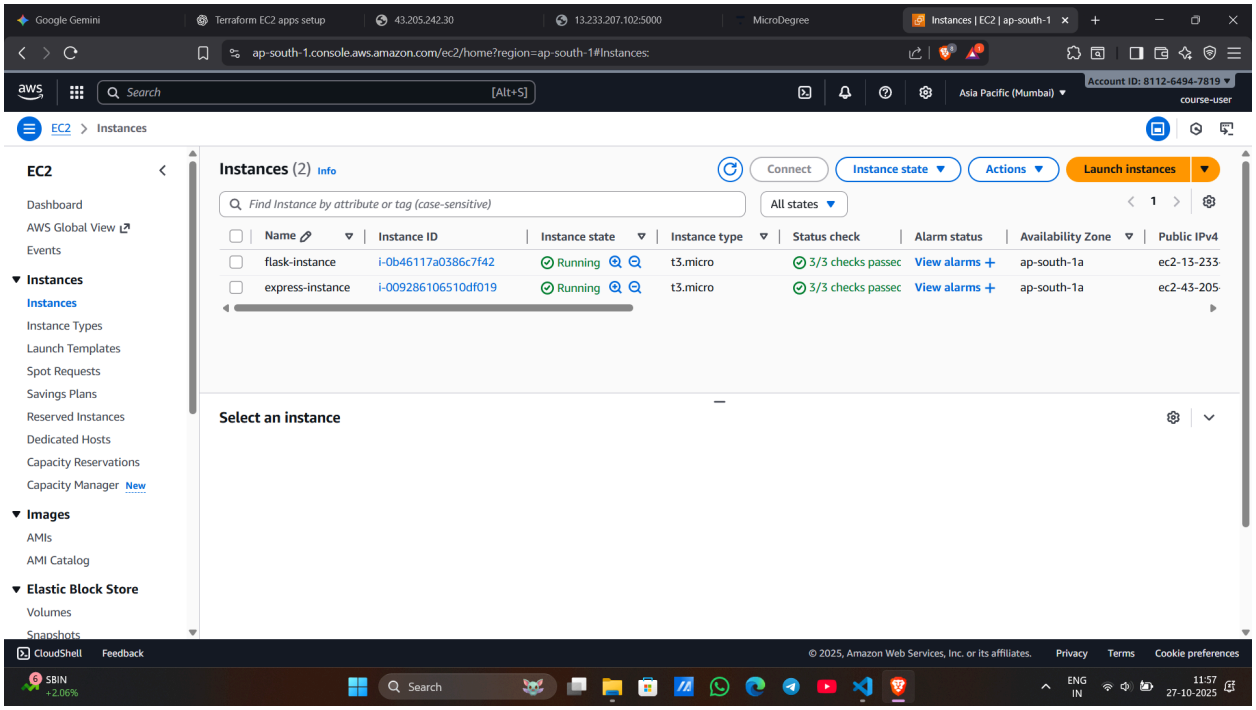

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERR...
part1_single_ec2
apps
terraform
part2_two_ec2
apps
terraform
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
terraform.tfstate
terraform.tfvars
tfplan
user_data_express.sh
user_data_flask.sh
variables.tf
screenshots
main.tf
variables.tf
outputs.tf
user_data_flask.sh
user_data_express.sh
terraform.tfvars
index.js
pacakge.json
app.py
part2_two_ec2 > terraform > main.tf
126 resource "aws_instance" "flask" {
132 user_data = file("${path.module}/user_data_flask.sh")
133 associate_public_ip_address = true
135 tags = { Name = "flask-instance" }
136 }
138 resource "aws_instance" "express" {
139 ami = "ami-0522ab6e1ddcc7055"
140 instance_type = "t3.micro"
141 subnet_id = aws_subnet.public.id
142 vpc_security_group_ids = [aws_security_group.express_sg.id]
143 key_name = var.key_name
144 user_data = file("${path.module}/user_data_express.sh")
145 associate_public_ip_address = true
147 tags = { Name = "express-instance" }
148 }
149
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - terraform
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part2_two_ec2\terraform> terraform plan -out=tfplan
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.express will be created
+ resource "aws_instance" "express" {
+ ami = "ami-0522ab6e1ddcc7055"
+ arn = (known after apply)
+ associate_public_ip_address = true
+ availability_zone = (known after apply)
OUTLINE
TIMELINE
main*
Ln 149, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
Breaking news
Indian-origin wo...
Search
ENG
IN
11:53
27-10-2025

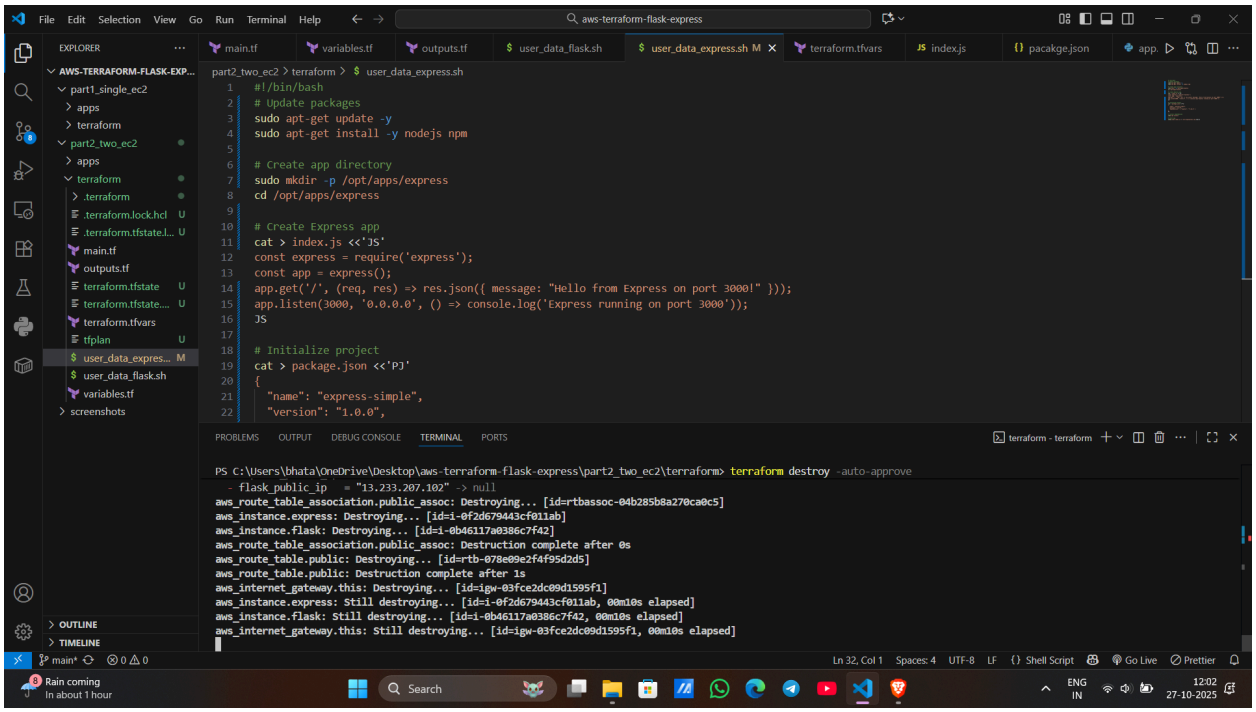

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERR...
part1_single_ec2
apps
terraform
part2_two_ec2
apps
terraform
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
terraform.tfstate
terraform.tfvars
tfplan
user_data_express.sh
user_data_flask.sh
variables.tf
screenshots
main.tf
variables.tf
outputs.tf
user_data_flask.sh
user_data_express.sh
terraform.tfvars
index.js
pacakge.json
app.py
part2_two_ec2 > terraform > main.tf
126 resource "aws_instance" "flask" {
132 user_data = file("${path.module}/user_data_flask.sh")
133 associate_public_ip_address = true
135 tags = { Name = "flask-instance" }
136 }
138 resource "aws_instance" "express" {
139 ami = "ami-0522ab6e1ddcc7055"
140 instance_type = "t3.micro"
141 subnet_id = aws_subnet.public.id
142 vpc_security_group_ids = [aws_security_group.express_sg.id]
143 key_name = var.key_name
144 user_data = file("${path.module}/user_data_express.sh")
145 associate_public_ip_address = true
147 tags = { Name = "express-instance" }
148 }
149
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - terraform
To perform exactly these actions, run the following command to apply:
terraform apply "tfplan"
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part2_two_ec2\terraform> terraform apply "tfplan"
aws_vpc.this: Creating...
aws_vpc.this: Still creating... [00m10s elapsed]
aws_vpc.this: Creation complete after 11s [id=vpc-09ef79ee8289eb892]
aws_subnet.public: Creating...
aws_internet_gateway.this: Creating...
aws_security_group.flask_sg: Creating...
aws_internet_gateway.this: Creation complete after 1s [id=igw-03fce2dc09d1595f1]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-078e09e2f4f95d2d5]
aws_security_group.flask_sg: Creation complete after 2s [id=sg-06abe948e71b93195]
OUTLINE
TIMELINE
main*
Ln 149, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
Breaking news
Indian-origin wo...
Search
ENG
IN
11:54
27-10-2025





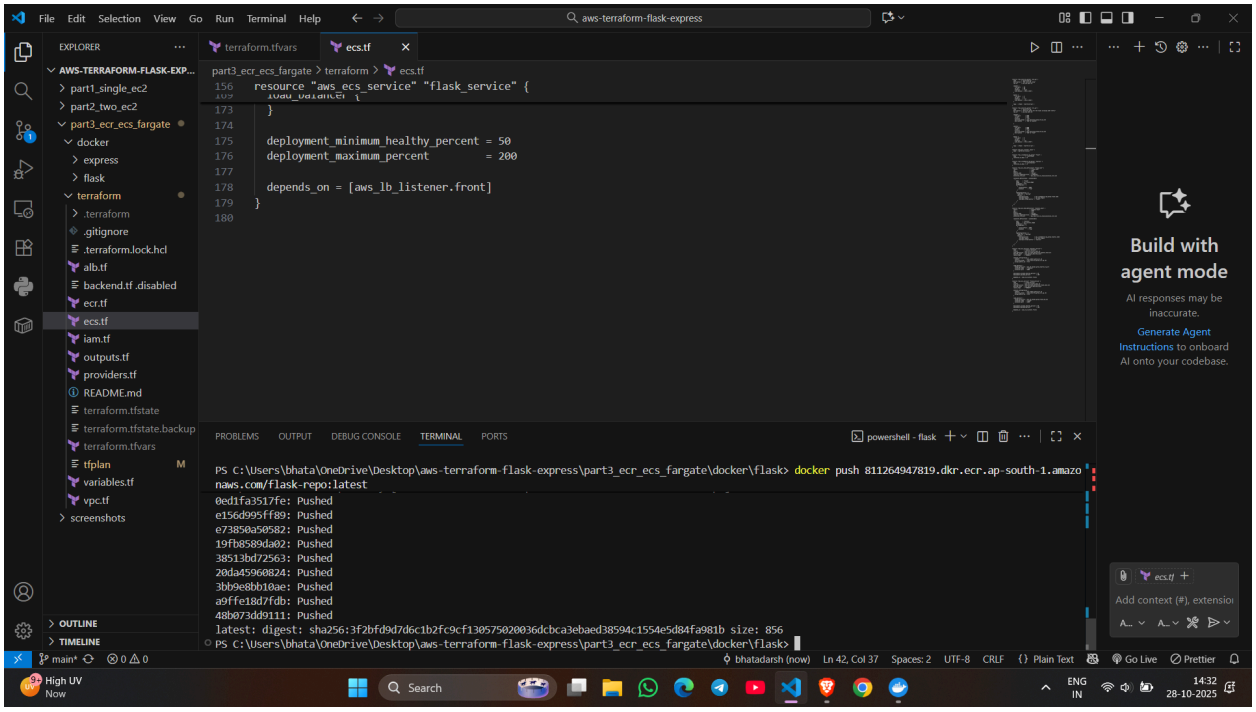
Hello from Flask on port 5000!

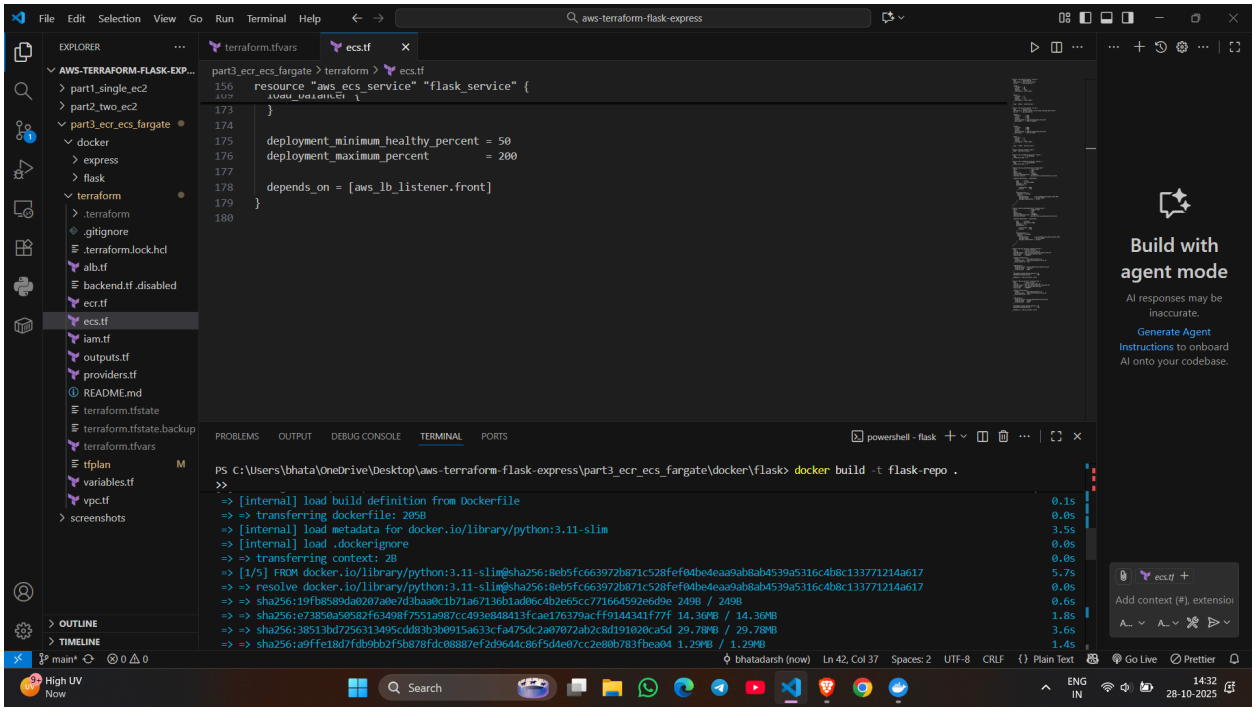


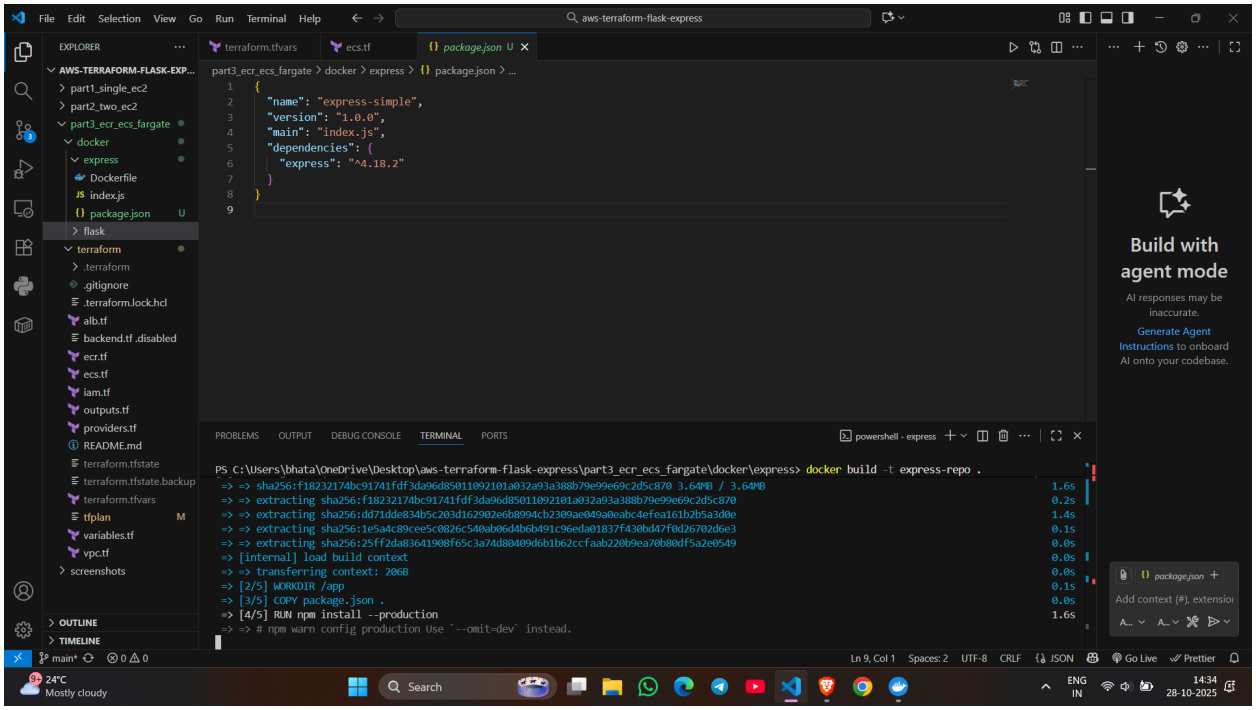


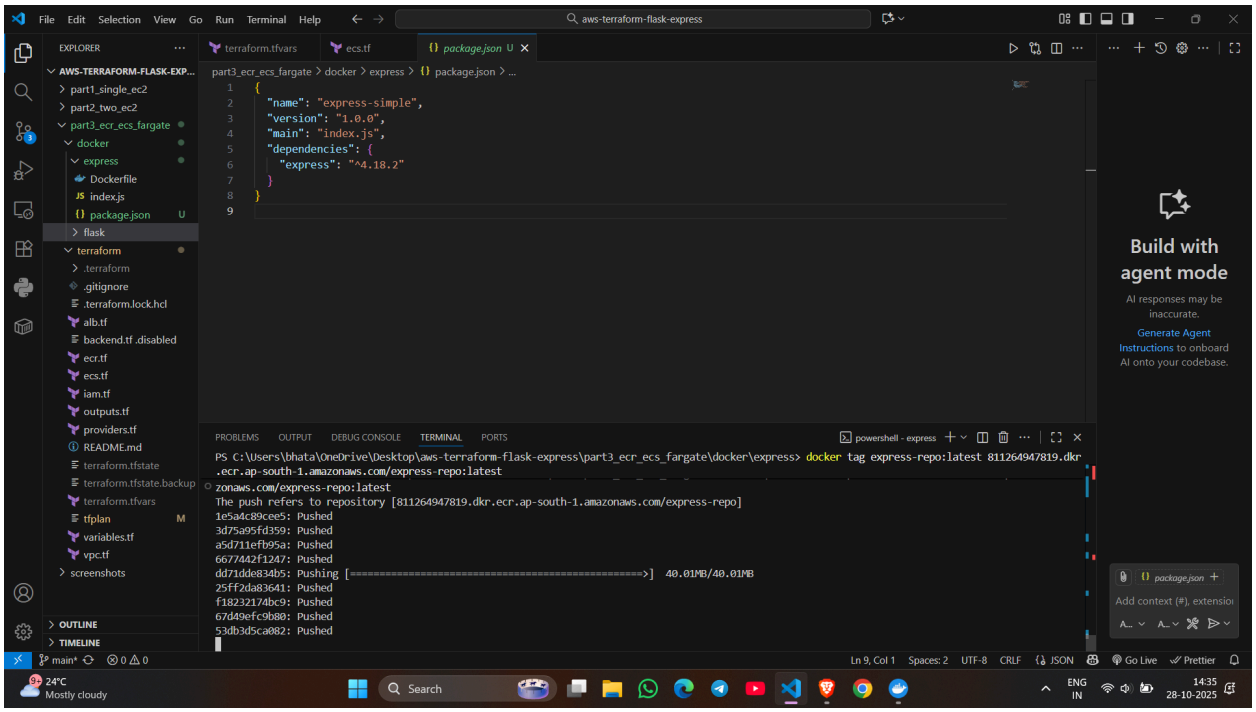
In this task, I deployed the Flask backend and Express frontend on two separate EC2 instances using Terraform. Each instance was provisioned with its own user data script to install dependencies and start the respective applications. Terraform was used to define the VPC, subnets, and security groups, ensuring proper communication between both instances and public accessibility on their respective ports. The Flask app ran on port 5000 and the Express app on port 3000, both successfully reachable via their public IPs.

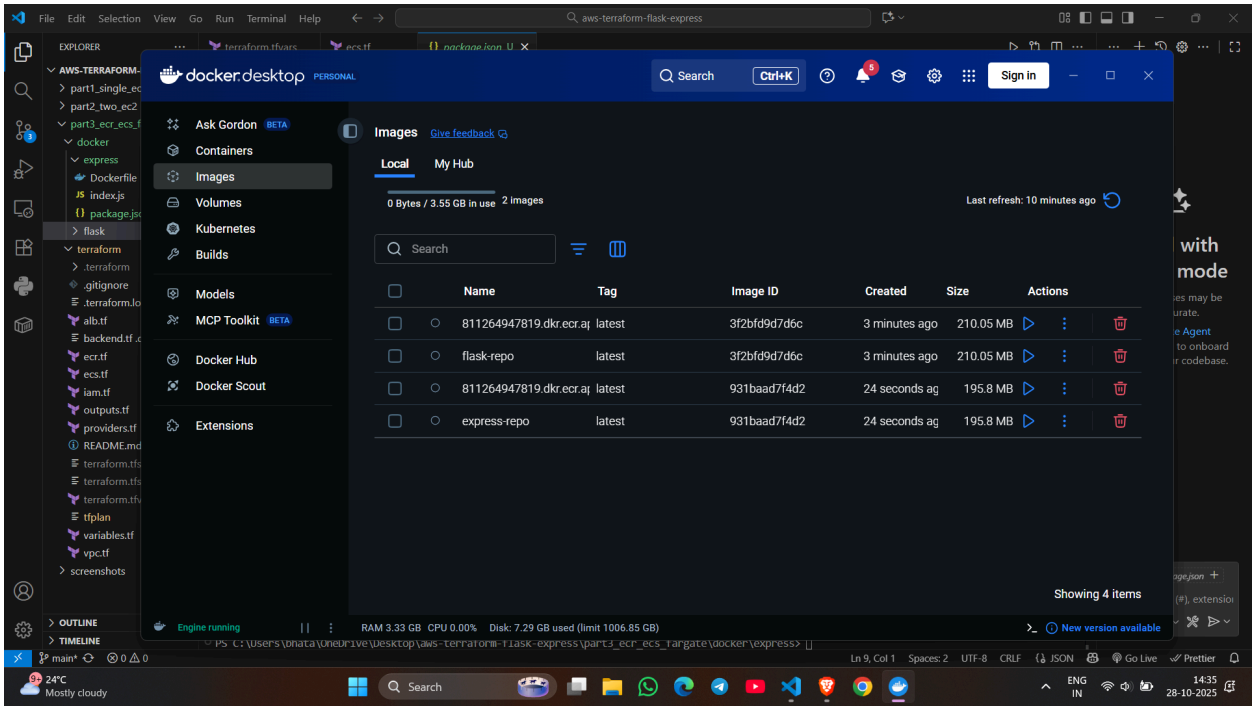
Part3:

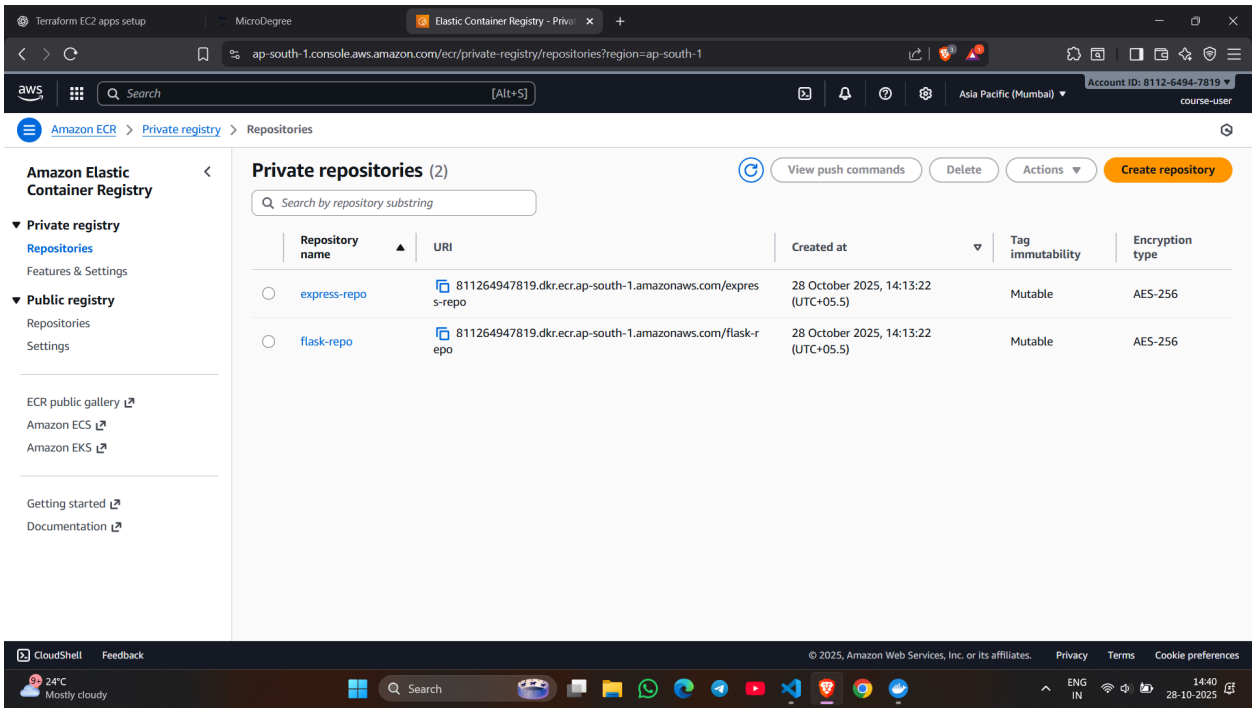

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
part2_two_ec2
part3_ecr_ecs_fargate
docker
express
flask
terraform
.terraform
.gitignore
.terraform.lock.hcl
alb.tf
backend.tf .disabled
ecr.tf
ecs.tf
iam.tf
outputs.tf
providers.tf
README.md
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
tfplan
variables.tf
vpc.tf
screenshots
terraform.tfvars
ecs.tf
part3_ecr_ecs_fargate > terraform > ecs.tf
156 resource "aws_ecs_service" "flask_service" {
173 }
174
175 deployment_minimum_healthy_percent = 50
176 deployment_maximum_percent = 200
177
178 depends_on = [aws_lb_listener.front]
179 }
180
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - flask
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\docker\flask> docker push 811264947819.dkr.ecr.ap-south-1.amazonaws.com/flask-repo:latest
0ed1fa3517fe: Pushed
e156d995ff89: Pushed
e73850a50582: Pushed
19fb8589da02: Pushed
38513bd72563: Pushed
20da45960824: Pushed
3bb9e8bb10ae: Pushed
a9ffe18d7fdb: Pushed
48b073dd9111: Pushed
latest: digest: sha256:3f2bfd9d7d6c1b2fc9cf130575020036dcbca3ebaed38594c1554e5d84fa981b size: 856
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\docker\flask>
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
OUTLINE
TIMELINE
main*
bhatadarsh (now) Ln 42, Col 37 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
High UV Now
Search
ENG IN
14:32 28-10-2025

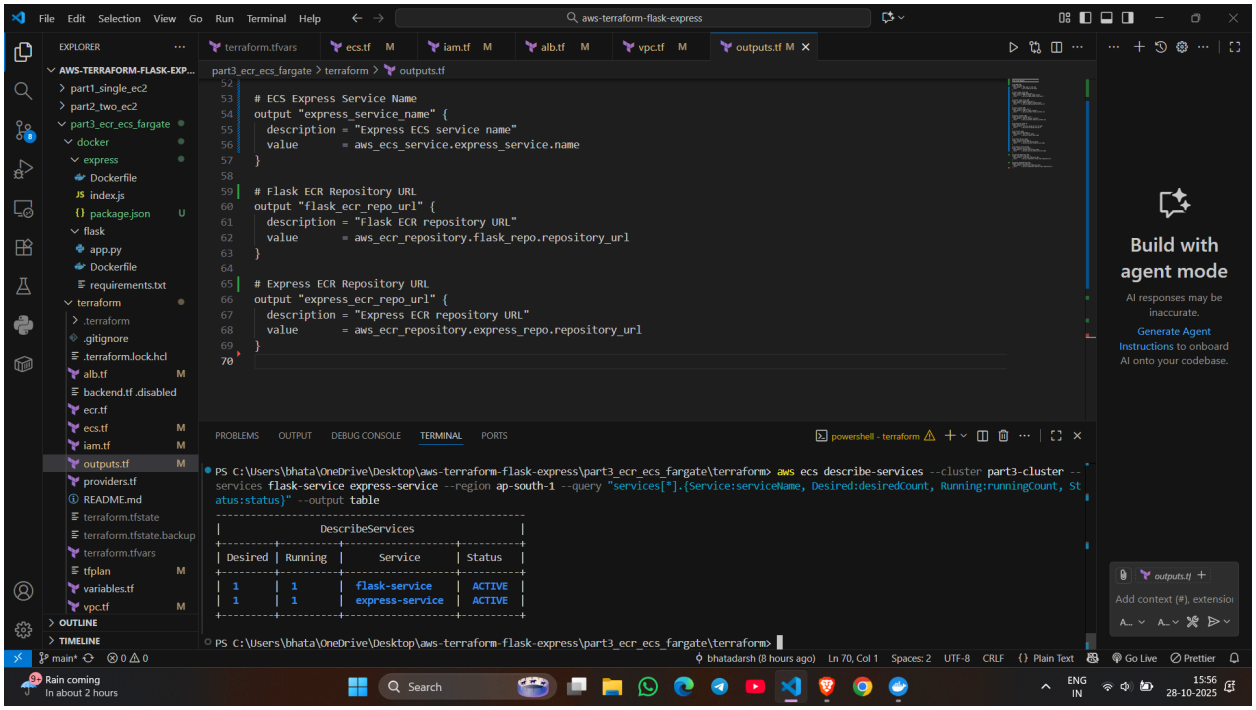

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
part2_two_ec2
part3_ecr_ecs_fargate
docker
express
flask
terraform
.terraform
.gitignore
.terraform.lock.hcl
alb.tf
backend.tf .disabled
ecr.tf
ecs.tf
iam.tf
outputs.tf
providers.tf
README.md
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
tfplan
variables.tf
vpc.tf
screenshots
terraform.tfvars
ecs.tf
part3_ecr_ecs_fargate > terraform > ecs.tf
156 resource "aws_ecs_service" "flask_service" {
173 }
174
175 deployment_minimum_healthy_percent = 50
176 deployment_maximum_percent = 200
177
178 depends_on = [aws_lb_listener.front]
179 }
180
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - flask
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\docker\flask> docker build -t flask-repo .
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 205B 0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim 3.5s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/python:3.11-slim@sha256:8eb5fc663972b871c528fef04be4eaa9ab8ab4539a5316c4b8c133771214a617 5.7s
=> => resolve docker.io/library/python:3.11-slim@sha256:8eb5fc663972b871c528fef04be4eaa9ab8ab4539a5316c4b8c133771214a617 0.0s
=> => sha256:19fb8589da0207a0e7d3baa0c1b71a67136b1ad06c4b2e65cc771664592e6d9e 249B / 249B 0.6s
=> => sha256:e73850a50582f63498f7551a987cc493e848413fcae176379acff9144341f77f 14.36MB / 14.36MB 1.8s
=> => sha256:38513bd7256313495cdd83b3b0915a633cfa475dc2a07072ab2c8d191020ca5d 29.78MB / 29.78MB 3.6s
=> => sha256:a9ffe18d7fdb9bb2f5b878fdc08887ef2d9644c86f5d4e07cc2e80b783fbea04 1.29MB / 1.29MB 1.4s
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
OUTLINE
TIMELINE
main*
bhatadarsh (now) Ln 42, Col 37 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
High UV Now
Search
ENG IN
14:32 28-10-2025

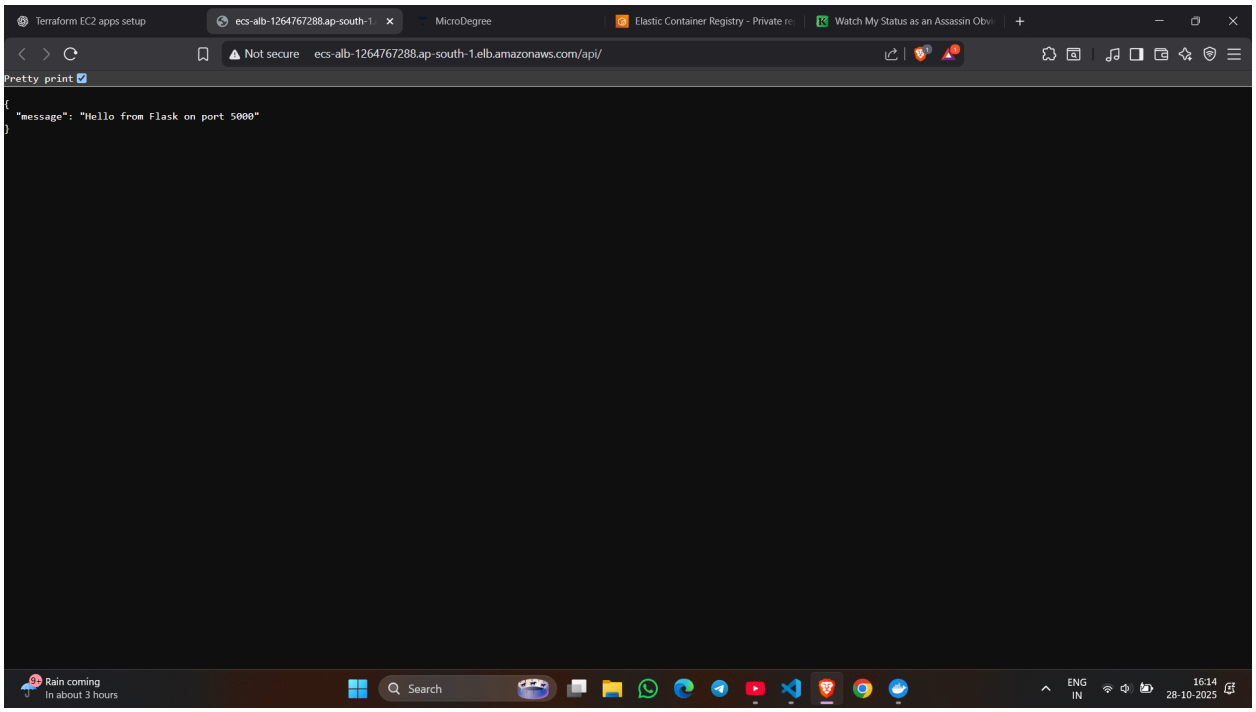

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
part2_two_ec2
part3_ecr_ecs_fargate
docker
express
Dockerfile
index.js
package.json U
flask
terraform
.terraform
.gitignore
.terraform.lock.hcl
alb.tf
backend.tf.disabled
ecr.tf
ecs.tf
iam.tf
outputs.tf
providers.tf
README.md
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
tfplan M
variables.tf
vpc.tf
screenshots
OUTLINE
TIMELINE
terraform.tfvars
ecs.tf
package.json U
part3_ecr_ecs_fargate > docker > express > package.json > ...
{
"name": "express-simple",
"version": "1.0.0",
"main": "index.js",
"dependencies": {
"express": "^4.18.2"
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - express
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\docker\express> docker build -t express-repo .
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB 1.6s
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.2s
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e 1.4s
=> => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 0.1s
=> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 206B 0.0s
=> [2/5] WORKDIR /app 0.1s
=> [3/5] COPY package.json . 0.0s
=> [4/5] RUN npm install --production 1.6s
=> => # npm warn config production Use `--omit=dev` instead.
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
package.json
Add context (#), extensio
main*
Ln 9, Col 1 Spaces: 2 UTF-8 CRLF JSON Go Live Prettier
24°C
Mostly cloudy
Search
ENG
IN
14:34
28-10-2025

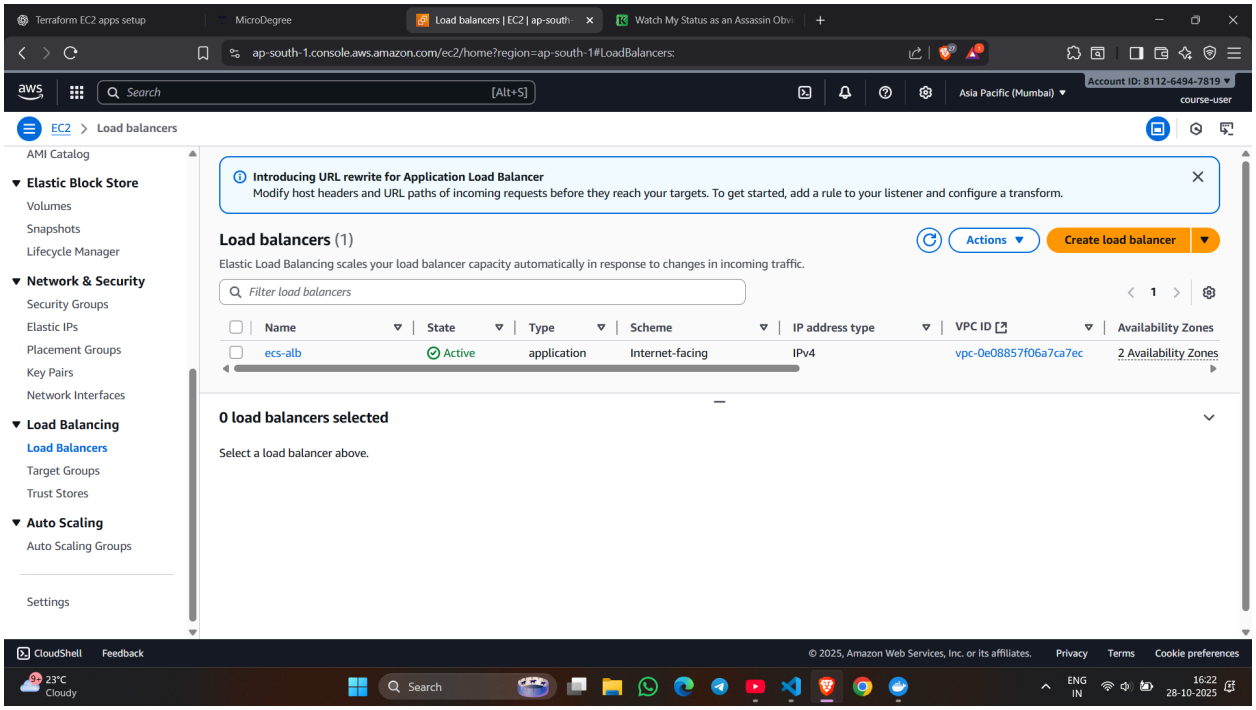

File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
part2_two_ec2
part3_ecr_ecs_fargate
docker
express
Dockerfile
index.js
package.json U
flask
terraform
.terraform
.gitignore
.terraform.lock.hcl
alb.tf
backend.tf.disabled
ecr.tf
ecs.tf
iam.tf
outputs.tf
providers.tf
README.md
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
tfplan M
variables.tf
vpc.tf
screenshots
OUTLINE
TIMELINE
terraform.tfvars
ecs.tf
package.json U
part3_ecr_ecs_fargate > docker > express > package.json > ...
{
"name": "express-simple",
"version": "1.0.0",
"main": "index.js",
"dependencies": {
"express": "^4.18.2"
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - express
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\docker\express> docker tag express-repo:latest 811264947819.dkr.ecr.ap-south-1.amazonaws.com/express-repo:latest
zonaws.com/express-repo:latest
The push refers to repository [811264947819.dkr.ecr.ap-south-1.amazonaws.com/express-repo]
1e5a4c89cee5: Pushed
3d75a95fd359: Pushed
a5d711efb95a: Pushed
6677442f1247: Pushed
dd71dde834b5: Pushing [==========>] 40.01MB/40.01MB
25ff2da83641: Pushed
f18232174bc9: Pushed
67d49efc9b80: Pushed
53db3d5ca082: Pushed
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
package.json
Add context (#), extensio
main*
Ln 9, Col 1 Spaces: 2 UTF-8 CRLF JSON Go Live Prettier
24°C
Mostly cloudy
Search
ENG
IN
14:35
28-10-2025

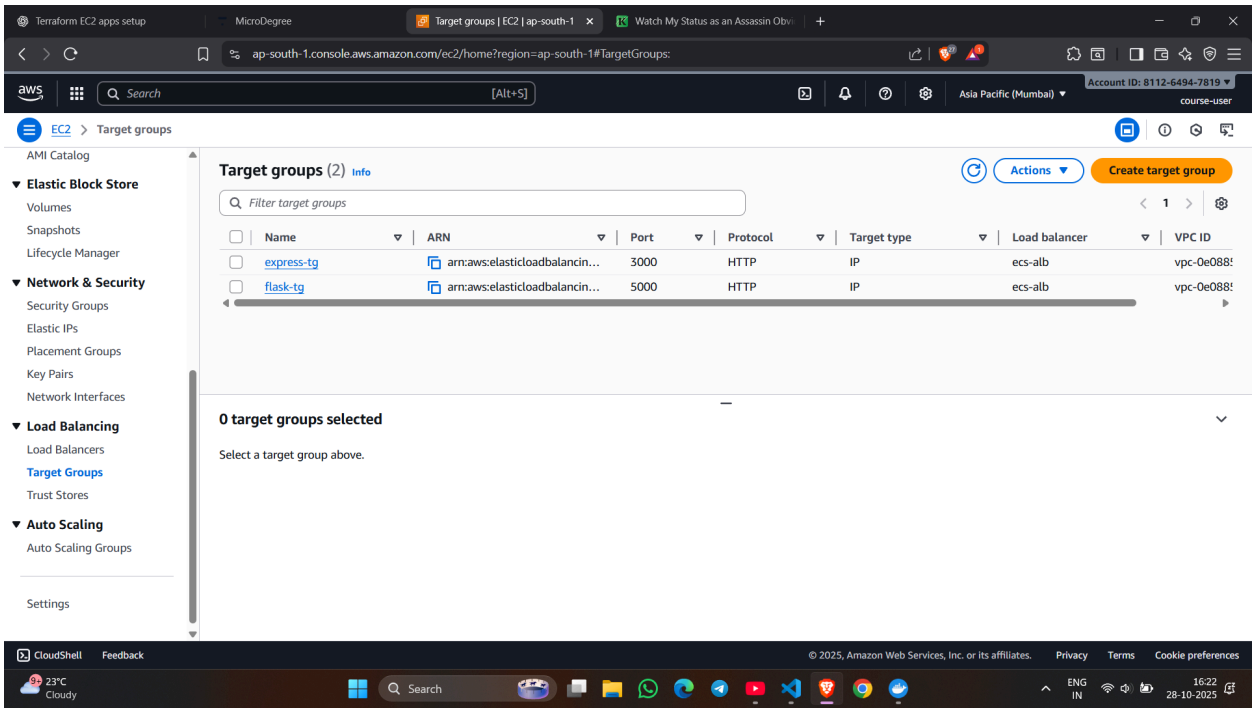

docker desktop PERSONAL
Search Ctrl+K
Sign in
Ask Gordon BETA
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions
Images
Give feedback
Local
My Hub
0 Bytes / 3.55 GB in use
2 Images
Last refresh: 10 minutes ago
Search
Name
Tag
Image ID
Created
Size
Actions
811264947819.dkr.ecr.ap latest 3f2bfd9d7d6c 3 minutes ago 210.05 MB
flask-repo latest 3f2bfd9d7d6c 3 minutes ago 210.05 MB
811264947819.dkr.ecr.ap latest 931baad7f4d2 24 seconds ag 195.8 MB
express-repo latest 931baad7f4d2 24 seconds ag 195.8 MB
Showing 4 items
Engine running
RAM 3.33 GB CPU 0.00% Disk: 7.29 GB used (limit 1006.85 GB)
New version available

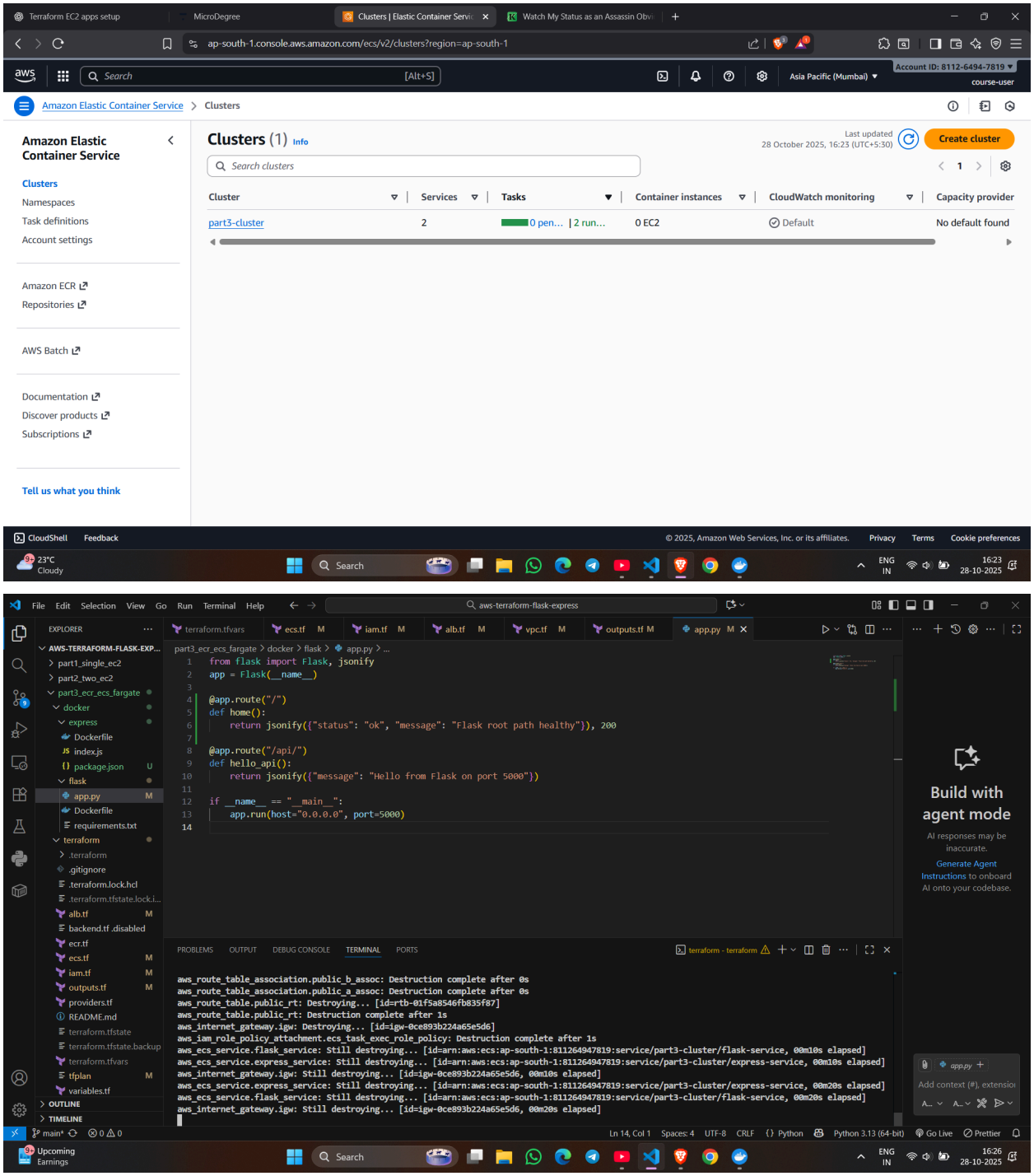

Terraform EC2 apps setup
MicroDegree
Elastic Container Registry - Priva
ap-south-1.console.aws.amazon.com/ecr/private-registry/repositories?region=ap-south-1
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8112-6494-7819
course-user
Amazon ECR > Private registry > Repositories
Amazon Elastic Container Registry
Private registry
Repositories
Features & Settings
Public registry
Repositories
Settings
ECR public gallery
Amazon ECS
Amazon EKS
Getting started
Documentation
Private repositories (2)
View push commands
Delete
Actions
Create repository
Search by repository substring
Repository name
URI
Created at
Tag immutability
Encryption type
express-repo 811264947819.dkr.ecr.ap-south-1.amazonaws.com/express-repo 28 October 2025, 14:13:22 (UTC+05.5) Mutable AES-256
flask-repo 811264947819.dkr.ecr.ap-south-1.amazonaws.com/flask-repo 28 October 2025, 14:13:22 (UTC+05.5) Mutable AES-256
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


File Edit Selection View Go Run Terminal Help
aws-terraform-flask-express
EXPLORER
AWS-TERRAFORM-FLASK-EXP...
part1_single_ec2
part2_two_ec2
part3_ecr_ecs_fargate
docker
express
Dockerfile
index.js
package.json
flask
app.py
Dockerfile
requirements.txt
terraform
.terraform
.gitignore
.terraform.lock.hcl
alb.tf
backend.tf.disabled
ecr.tf
ecs.tf
iam.tf
outputs.tf
providers.tf
README.md
terraform.tfstate
terraform.tfstate.backup
terraform.tfvars
tfplan
variables.tf
vpc.tf
OUTLINE
TIMELINE
terraform.tfvars
ecs.tf M
iam.tf M
alb.tf M
vpc.tf M
outputs.tf M
part3_ecr_ecs_fargate > terraform > outputs.tf
52
53 # ECS Express Service Name
54 output "express_service_name" {
55 description = "Express ECS service name"
56 value = aws_ecs_service.express_service.name
57 }
58
59 # Flask ECR Repository URL
60 output "flask_ecr_repo_url" {
61 description = "Flask ECR repository URL"
62 value = aws_ecr_repository.flask_repo.repository_url
63 }
64
65 # Express ECR Repository URL
66 output "express_ecr_repo_url" {
67 description = "Express ECR repository URL"
68 value = aws_ecr_repository.express_repo.repository_url
69 }
70
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - terraform
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\terraform> aws ecs describe-services --cluster part3-cluster --services flask-service express-service --region ap-south-1 --query "services[*].{Service:serviceName, Desired:desiredCount, Running:runningCount, Status:status}" --output table
DescribeServices
Desired | Running | Service | Status
1 | 1 | flask-service | ACTIVE
1 | 1 | express-service | ACTIVE
PS C:\Users\bhata\OneDrive\Desktop\aws-terraform-flask-express\part3_ecr_ecs_fargate\terraform>
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Add context (#), extensio
main*
bhatadarsh (8 hours ago) Ln 70, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
Rain coming
In about 2 hours
Search
ENG
IN
15:56
28-10-2025


Terraform EC2 apps setup
ecs-alb-1264767288.ap-south-1...
MicroDegree
Elastic Container Registry - Private re
Watch My Status as an Assassin Obvi
Not secure
ecs-alb-1264767288.ap-south-1.elb.amazonaws.com/api/
Pretty print
{
"message": "Hello from Flask on port 5000"
}
Rain coming
In about 3 hours
Search
ENG
IN
16:14
28-10-2025


Terraform EC2 apps setup
MicroDegree
Load balancers | EC2 | ap-south-
Watch My Status as an Assassin Obvi
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancers:
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8112-6494-7819
course-user
EC2 > Load balancers
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Settings
Introducing URL rewrite for Application Load Balancer
Modify host headers and URL paths of incoming requests before they reach your targets. To get started, add a rule to your listener and configure a transform.
Load balancers (1)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Filter load balancers
Name
State
Type
Scheme
IP address type
VPC ID
Availability Zones
ecs-alb
Active
application
Internet-facing
IPv4
vpc-0e08857f06a7ca7ec
2 Availability Zones
0 load balancers selected
Select a load balancer above.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Cloudy
ENG
IN
16:22
28-10-2025


Terraform EC2 apps setup
MicroDegree
Target groups | EC2 | ap-south-1
Watch My Status as an Assassin Obvi
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroups:
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8112-6494-7819
course-user
EC2 > Target groups
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Settings
Target groups (2) Info
Actions
Create target group
Filter target groups
Name
ARN
Port
Protocol
Target type
Load balancer
VPC ID
express-tg
arn:aws:elasticloadbalancin...
3000
HTTP
IP
ecs-alb
vpc-0e088
flask-tg
arn:aws:elasticloadbalancin...
5000
HTTP
IP
ecs-alb
vpc-0e088
0 target groups selected
Select a target group above.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Cloudy
ENG
IN
16:22
28-10-2025



In this task, I containerized both the Flask backend and Express frontend using Docker and deployed them on AWS using Terraform. Two ECR repositories were created to store the Docker images, which were then pushed after building. Terraform was used to set up a complete VPC network with subnets, route tables, and security groups, followed by creating an ECS cluster with two services — one for Flask and one for Express — using the Fargate launch type. An Application Load Balancer (ALB) was configured to route traffic to each service on their respective ports. Both applications were successfully deployed and made accessible through the ALB endpoint.