TERRAFORM PROJECT

# **Summary :**

Terraform AWS Infrastructure ProjectTools used: Terraform, AWS, VS Code, Git Bash

Built an auto-scaling AWS infrastructure with Terraform scripts.

Provisioned and configured:

VPC, Subnets (within AZs), Internet Gateway, and Route Tables.

Security Groups, EC2 instance and Load Balancer (Application LB).

Launch Template and Auto Scaling Group with an associated Target Group.

Applied Output blocks to pull key resource identifiers after deployment.

Provided smooth traffic management and scalability using a Load Balancer and an Auto Scaling.

Used infrastructure as code best practices to develop modular and repeatable Terraform scripts.

# **Create Autoscaling using terraform :**

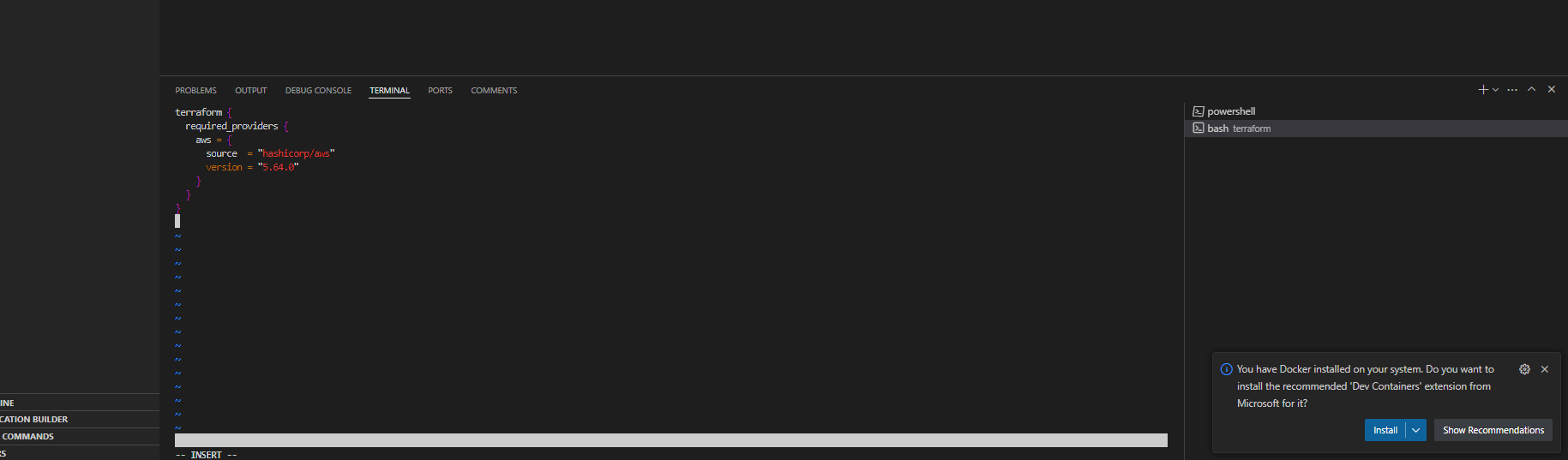
**Steps:**

1. Install awscli and Terraform in the Visual code Studio (git bash).

2. Create one directory and create different blocks.

2.Creating three blocks

### **Terraform block**



### Provider.tf



### **Resource.tf**

# Create a VPC

resource "aws\_vpc" "autoscale\_vpc" {

cidr\_block = "10.0.0.0/16"

}

# Create the first Subnet

resource "aws\_subnet" "autoscale\_subnet\_az1" {

vpc\_id = aws\_vpc.autoscale\_vpc.id

cidr\_block = "10.0.3.0/24"

availability\_zone = "us-east-1a"

}

# Create the second Subnet

resource "aws\_subnet" "autoscale\_subnet\_az2" {

vpc\_id = aws\_vpc.autoscale\_vpc.id

cidr\_block = "10.0.4.0/24"

availability\_zone = "us-east-1b"

}

# Create an Internet Gateway

resource "aws\_internet\_gateway" "autoscale\_igw" {

vpc\_id = aws\_vpc.autoscale\_vpc.id

}

# Create a Route Table

resource "aws\_route\_table" "autoscale\_route\_table" {

vpc\_id = aws\_vpc.autoscale\_vpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.autoscale\_igw.id

}

}

# Associate the Route Table with the Subnets

resource "aws\_route\_table\_association" "autoscale\_rta\_az1" {

subnet\_id = aws\_subnet.autoscale\_subnet\_az1.id

route\_table\_id = aws\_route\_table.autoscale\_route\_table.id

}

resource "aws\_route\_table\_association" "autoscale\_rta\_az2" {

subnet\_id = aws\_subnet.autoscale\_subnet\_az2.id

route\_table\_id = aws\_route\_table.autoscale\_route\_table.id

}

# Create a Security Group

resource "aws\_security\_group" "autoscale\_sg" {

vpc\_id = aws\_vpc.autoscale\_vpc.id

ingress {

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

# Create an EC2 Instance

resource "aws\_instance" "autoscale\_instance" {

ami = "ami-0e86e20dae9224db8"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.autoscale\_subnet\_az1.id

vpc\_security\_group\_ids = [aws\_security\_group.autoscale\_sg.id]

tags = {

Name = "autoscale-ec2"

}

}

# Create a Launch Template

resource "aws\_launch\_template" "autoscale\_launch\_template" {

name\_prefix = "autoscale-template-"

image\_id = aws\_instance.autoscale\_instance.ami

instance\_type = aws\_instance.autoscale\_instance.instance\_type

network\_interfaces {

associate\_public\_ip\_address = true

security\_groups = [aws\_security\_group.autoscale\_sg.id]

subnet\_id = aws\_subnet.autoscale\_subnet\_az1.id

}

tag\_specifications {

resource\_type = "instance"

tags = {

Name = "autoscale-instance"

}

}

}

# Create an Auto Scaling Group

resource "aws\_autoscaling\_group" "autoscale\_asg" {

launch\_template {

id = aws\_launch\_template.autoscale\_launch\_template.id

version = "$Latest"

}

min\_size = 1

max\_size = 2

desired\_capacity = 1

vpc\_zone\_identifier = [aws\_subnet.autoscale\_subnet\_az1.id, aws\_subnet.autoscale\_subnet\_az2.id]

target\_group\_arns = [aws\_lb\_target\_group.autoscale\_target\_group.arn]

tag {

key = "Name"

value = "autoscale-instance"

propagate\_at\_launch = true

}

}

# Create a Load Balancer

resource "aws\_lb" "autoscale\_lb" {

name = "autoscale-lb"

internal = false

load\_balancer\_type = "application"

security\_groups = [aws\_security\_group.autoscale\_sg.id]

subnets = [aws\_subnet.autoscale\_subnet\_az1.id, aws\_subnet.autoscale\_subnet\_az2.id]

enable\_deletion\_protection = false

}

# Create a Target Group

resource "aws\_lb\_target\_group" "autoscale\_target\_group" {

name = "autoscale-target-group"

port = 80

protocol = "HTTP"

vpc\_id = aws\_vpc.autoscale\_vpc.id

}

# Create a Listener for the Load Balancer

resource "aws\_lb\_listener" "autoscale\_listener" {

load\_balancer\_arn = aws\_lb.autoscale\_lb.arn

port = "80"

protocol = "HTTP"

default\_action {

type = "forward"

target\_group\_arn = aws\_lb\_target\_group.autoscale\_target\_group.arn

}

}

### Outputblock.tf

output "vpc\_id" {

description = "The ID of the VPC"

value = aws\_vpc.autoscale\_vpc.id

}

output "subnet\_1\_id" {

description = "The ID of the first subnet"

value = aws\_subnet.autoscale\_subnet\_az1.id

}

output "subnet\_2\_id" {

description = "The ID of the second subnet"

value = aws\_subnet.autoscale\_subnet\_az2.id

}

output "security\_group\_id" {

description = "The ID of the Security Group"

value = aws\_security\_group.autoscale\_sg.id

}

output "instance\_id" {

description = "The ID of the EC2 Instance"

value = aws\_instance.autoscale\_instance.id

}

output "autoscaling\_group\_name" {

description = "The name of the Auto Scaling Group"

value = aws\_autoscaling\_group.autoscale\_asg.name

}

output "load\_balancer\_dns" {

description = "The DNS name of the Load Balancer"

value = aws\_lb.autoscale\_lb.dns\_name

}

output "target\_group\_arn" {

description = "The ARN of the Target Group"

value = aws\_lb\_target\_group.autoscale\_target\_group.arn

}

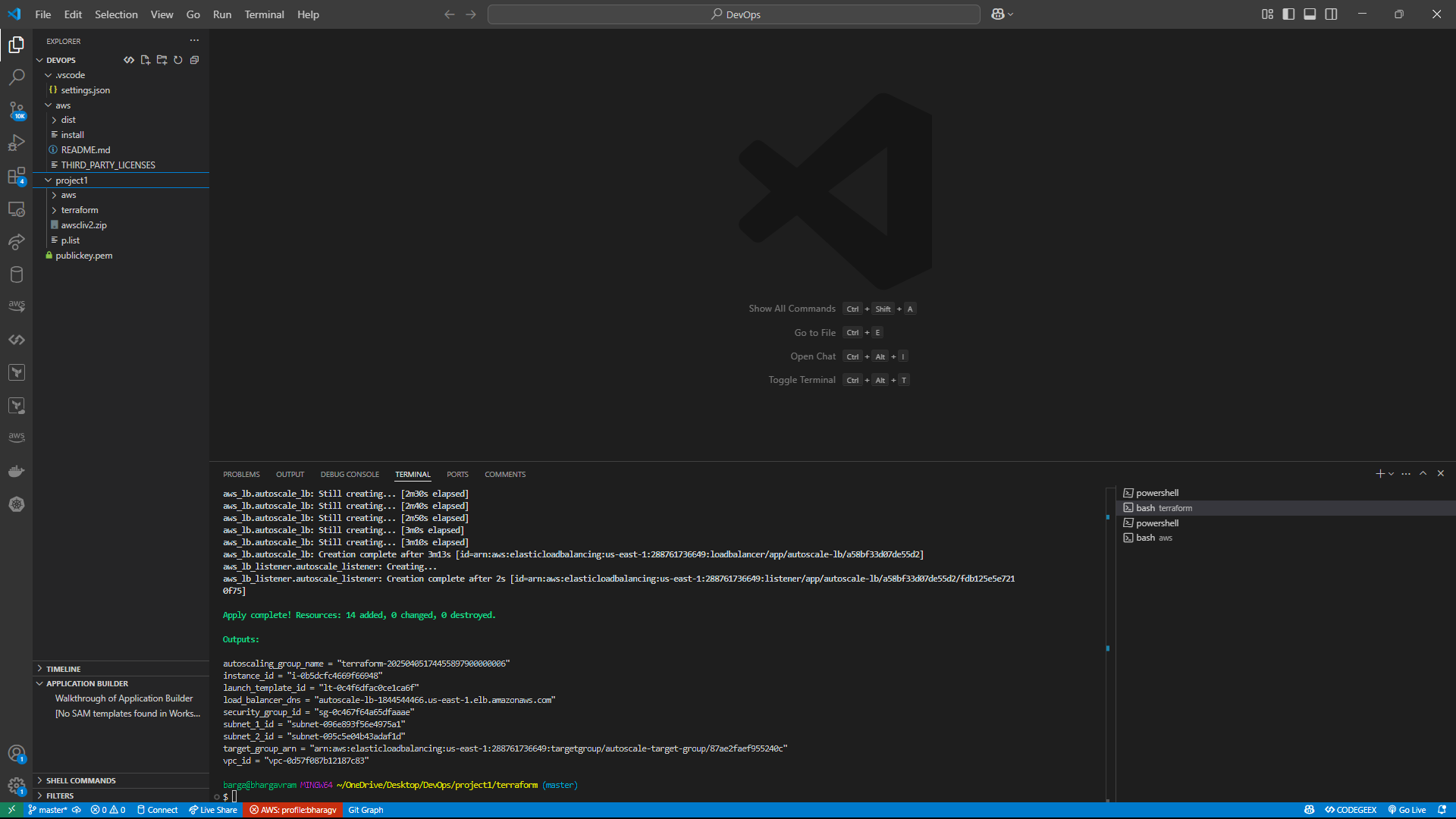
output "launch\_template\_id" {

description = "The ID of the Launch Template"

value = aws\_launch\_template.autoscale\_launch\_template.id

}

### **Successful Terraform Provisioning – Vs Terminal Output**



### **Instances**

