# Fetch the default VPC

data "aws\_vpc" "default" {

default = true

}

# Fetch all subnets within the default VPC

data "aws\_subnets" "default\_subnets" {

filter {

name = "vpc-id"

values = [data.aws\_vpc.default.id]

}

}

# Security Group for EC2 instances allowing HTTP traffic

resource "aws\_security\_group" "allow\_http" {

name = "pre-podscaling"

description = "Allow HTTP traffic"

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"]

}

}

# Launch Configuration for ASG (with for\_each for multiple sandboxes)

resource "aws\_launch\_configuration" "my\_launch\_config" {

for\_each = toset(var.sandboxes) # Loop through each sandbox

name = "podASG-${each.key}" # Name is unique for each sandbox

image\_id = var.ami

instance\_type = var.instance\_type

security\_groups = [aws\_security\_group.allow\_http.id]

associate\_public\_ip\_address = true

}

# Auto Scaling Group (ASG) for each sandbox

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

for\_each = aws\_launch\_configuration.my\_launch\_config # Reference to Launch Config

launch\_configuration = each.value.id

vpc\_zone\_identifier = data.aws\_subnets.default\_subnets.ids

desired\_capacity = 2

max\_size = 3

min\_size = 1

tag {

key = "Name"

value = "${each.key}-example-asg"

propagate\_at\_launch = true

}

}

# Application Load Balancer (ALB) for each sandbox

resource "aws\_lb" "my\_alb" {

for\_each = toset(var.sandboxes)

name = "pre-pod-${each.key}" # Unique ALB name for each sandbox

internal = false

load\_balancer\_type = "application"

security\_groups = [aws\_security\_group.allow\_http.id]

subnets = data.aws\_subnets.default\_subnets.ids

}

# ALB Target Group for each sandbox

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

for\_each = toset(var.sandboxes)

name = "pre-pod-${each.key}-target"

port = 80

protocol = "HTTP"

vpc\_id = data.aws\_vpc.default.id

}

# ALB Listener for each sandbox

resource "aws\_lb\_listener" "my\_listener" {

for\_each = aws\_lb.my\_alb

load\_balancer\_arn = each.value.arn

port = 80

protocol = "HTTP"

default\_action {

type = "forward"

target\_group\_arn = aws\_lb\_target\_group.my\_target\_group[each.key].arn

}

}

# Attach ASG to ALB Target Group for each sandbox

resource "aws\_autoscaling\_attachment" "asg\_attachment" {

for\_each = aws\_autoscaling\_group.my\_asg # Loop through each ASG

autoscaling\_group\_name = each.value.id

lb\_target\_group\_arn = aws\_lb\_target\_group.my\_target\_group[each.key].arn

}

# SNS Topic for Auto Scaling notifications

resource "aws\_sns\_topic" "example" {

name = "auto-scaling-notifications"

}

# SNS Topic Subscription (Email)

resource "aws\_sns\_topic\_subscription" "example" {

topic\_arn = aws\_sns\_topic.example.arn

protocol = "email"

endpoint = "kvs.vishnusai@gmail.com" # Replace with your email address

}

# Auto Scaling Notification for SNS

resource "aws\_autoscaling\_notification" "example" {

for\_each = aws\_autoscaling\_group.my\_asg # For each ASG

group\_names = [each.value.name]

notifications = [

"autoscaling:EC2\_INSTANCE\_LAUNCH",

"autoscaling:EC2\_INSTANCE\_TERMINATE",

"autoscaling:EC2\_INSTANCE\_LAUNCH\_ERROR",

"autoscaling:EC2\_INSTANCE\_TERMINATE\_ERROR",

]

topic\_arn = aws\_sns\_topic.example.arn

}

root@ip-172-31-44-74:/home/ubuntu/new# vi variables.tf

root@ip-172-31-44-74:/home/ubuntu/new# cat variables.tf

variable "ami" {

type = string

default = "ami-05d2438ca66594916"

}

variable "instance\_type" {

type = string

default = "t2.micro"

}

variable "sandboxes" {

type = list(string)

default = ["d1","d2","d3"]

}

Add provider .tf

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 5.0"

}

}

}

# Configure the AWS Provider

provider "aws" {

region = "ap-south-1"

}