Application loadbalancer with autoscaling in aws through manual process

Create loadbalncer with

Name:

Listener:

http 80

Avalibityzones:

vpc:

availability zones:

us-east-2a

us-east-2b

us-east-2c

Security group: add security group

Target group: exist target group

Name:

Target type: instance

Protocol: http

Path: /index.html

Advanced helath check settings:

Register target: first nothing after creating autoscaling

Auto scaling

Launch configuration

Add ami

Type: micro

Create Launch configuration

Name: scaling group name

Advanced Details:

User data: text data

Only assign a public ip

Auto scaling security group:

Ssh port 22

Above ec2instance

Group name: any name

Launch configuration : fron configuration

Group size : start 1

Network: vpc

Subnet: add subnet

Advance details:

Load balncing: retrieve traffic from loadbalncer

Target Groups: like index.html

Helath check type: elb

Helath check grace period: 300

Use scaling policies to adjust the capacity of this group

Review:

Create loadbalncer and autoscaling Using teraform

**Example.tf**

**provider "aws" {**

**access\_key = “aws\_access\_key\_id”**

**secret\_key = “aws\_secret\_access\_key\_id”**

**region = "ap-south-1"**

**}**

**data "aws\_availability\_zones" "all" {}**

**### Creating EC2 instance**

**resource "aws\_instance" "web" {**

**ami = "${lookup(var.amis,var.region)}"**

**count = "${var.count}"**

**key\_name = "${var.key\_name}"**

**vpc\_security\_group\_ids = ["${aws\_security\_group.instance.id}"]**

**source\_dest\_check = false**

**instance\_type = "t2.micro"**

**tags {**

**Name = "${format("web-%03d", count.index + 1)}"**

**}**

**}**

**### Creating Security Group for EC2**

**resource "aws\_security\_group" "instance" {**

**name = "terraform-example-instance"**

**ingress {**

**from\_port = 8080**

**to\_port = 8080**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**ingress {**

**from\_port = 22**

**to\_port = 22**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**}**

**## Creating Launch Configuration**

**resource "aws\_launch\_configuration" "example" {**

**image\_id = "${lookup(var.amis,var.region)}"**

**instance\_type = "t2.micro"**

**security\_groups = ["${aws\_security\_group.instance.id}"]**

**key\_name = "${var.key\_name}"**

**user\_data = <<-EOF**

**#!/bin/bash**

**echo "Hello, World" > index.html**

**nohup busybox httpd -f -p 8080 &**

**EOF**

**lifecycle {**

**create\_before\_destroy = true**

**}**

**}**

**## Creating AutoScaling Group**

**resource "aws\_autoscaling\_group" "example" {**

**launch\_configuration = "${aws\_launch\_configuration.example.id}"**

**availability\_zones = ["${data.aws\_availability\_zones.all.names}"]**

**min\_size = 2**

**max\_size = 10**

**load\_balancers = ["${aws\_elb.example.name}"]**

**health\_check\_type = "ELB"**

**tag {**

**key = "Name"**

**value = "terraform-asg-example"**

**propagate\_at\_launch = true**

**}**

**}**

**## Security Group for ELB**

**resource "aws\_security\_group" "elb" {**

**name = "terraform-example-elb"**

**egress {**

**from\_port = 0**

**to\_port = 0**

**protocol = "-1"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**ingress {**

**from\_port = 80**

**to\_port = 80**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**}**

**### Creating ELB**

**resource "aws\_elb" "example" {**

**name = "terraform-asg-example"**

**security\_groups = ["${aws\_security\_group.elb.id}"]**

**availability\_zones = ["${data.aws\_availability\_zones.all.names}"]**

**health\_check {**

**healthy\_threshold = 2**

**unhealthy\_threshold = 2**

**timeout = 3**

**interval = 30**

**target = "HTTP:8080/"**

**}**

**listener {**

**lb\_port = 80**

**lb\_protocol = "http"**

**instance\_port = "8080"**

**instance\_protocol = "http"**

**}**

**}**

variables.tf

variable “count” {  
 default = 1  
 }  
variable “region” {  
 description = “AWS region for hosting our your network”  
 default = “ap-south-1”  
}  
variable “public\_key\_path” {  
 description = “Enter the path to the SSH Public Key to add to AWS.”  
 default = “/path\_to\_keyfile/keypair\_name.pem”  
}  
variable “key\_name” {  
 description = “Key name for SSHing into EC2”  
 default = “kaypair\_name”  
}  
variable “amis” {  
 description = “Base AMI to launch the instances”  
 default = {  
 ap-south-1 = “ami-8da8d2e2”  
 }  
}