Prashant Lakhera
Mar 1, 2019 · 4 min read · Listen

100 Days of DevOps — Day 19 - Application Load Balancer using Terraform

Welcome to Day 19 of 100 Days of DevOps, Let continue our terraform journey and see how can we create Application Load Balancer using Terraform

What is Application Load Balancer?

The ***Application Load Balancer*** *is a feature of Elastic* ***Load*** *Balancing that allows a developer to configure and route incoming end-user traffic to* ***applications based*** *in the Amazon Web Services (AWS) public cloud.*

Features

- *Layer7 load balancer(HTTP and HTTPs traffic)*
- *Support Path and Host-based routing(which let you route traffic to different target group)*
- *Listener support IPv6*

Some Key Terms

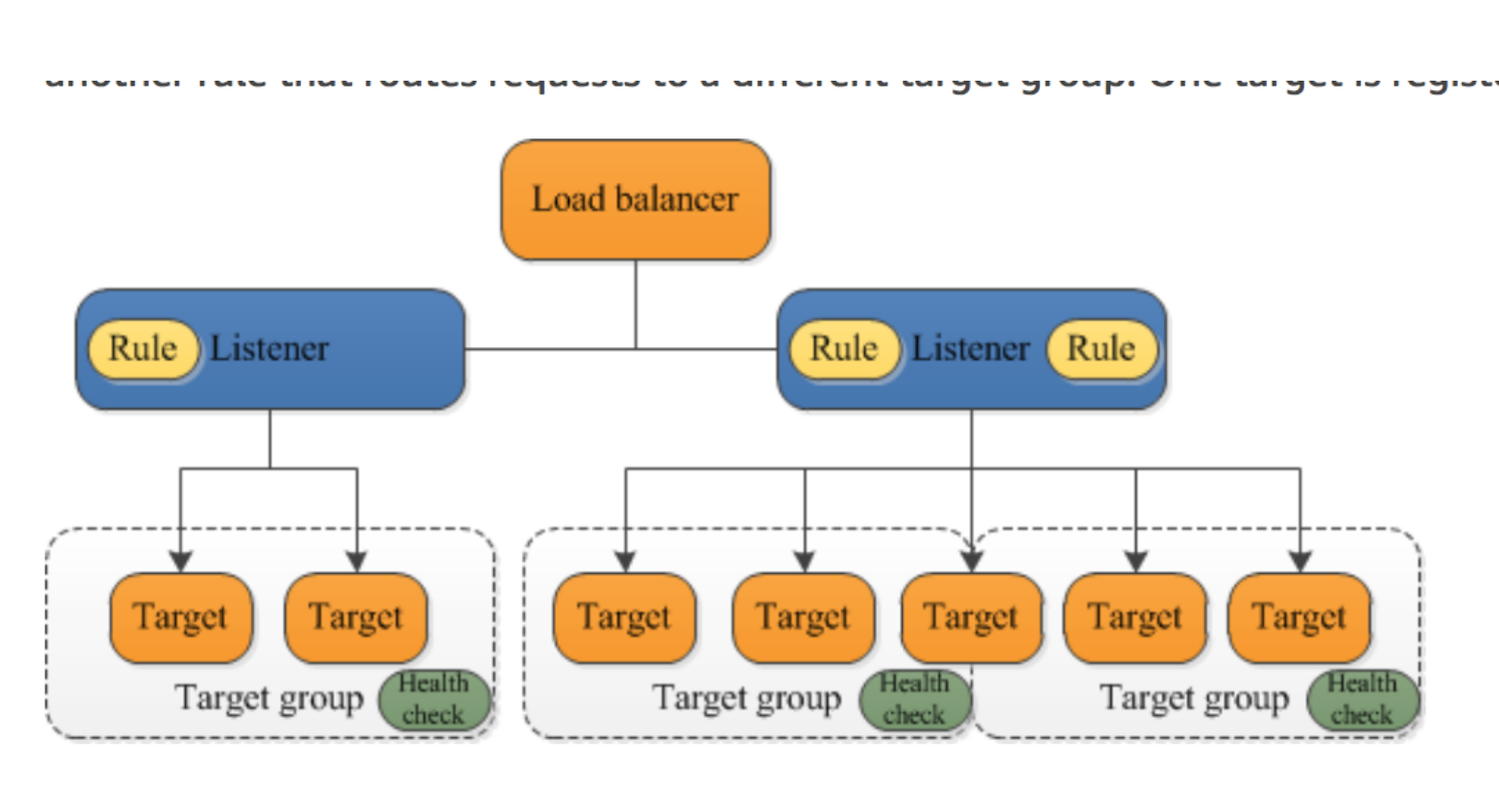
Target Group

Target types:

- *Instance types: Route traffic to the Primary Private IP address of that Instance*
- *IP: Route traffic to a specified IP address*
- *Lambda function*

Health Check

- *Determines whether to send traffic to a given instance*
- *Each instance must pass its a health check*
- *Sends HTTP GET request and looks for a specific response/success code*



Reference:
https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html

Step1: *Define the load balancer*

```
resource "aws_lb" "my-test-lb" {
  name               = "my-test-lb"
  internal           = false
  load_balancer_type = "application"
  ip_address_type    = "ipv4"
  subnets            = ["${var.subnet_id1}", "${var.subnet_id2}"]

  enable_deletion_protection = true

  tags {
    Name = "my-test-alb"
  }
}
```

alb_define.tf hosted with ❤ by GitHub · view raw

```
name: The name of the LB. This name must be unique within your AWS account, can have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and must not begin or end with a hyphen. If not specified, Terraform will autogenerate a name beginning with tf-lb (This part is important as Terraform auto
internal: If true, the LB will be internal.
load_balancer_type: The type of load balancer to create. Possible values are application or network. The default value is application.
ip_address_type: The type of IP addresses used by the subnets for your load balancer. The possible values are ipv4 and dualstack
subnets: A list of subnet IDs to attach to the LB.In this case I am attaching two public subnets we created during load balancer creation
enable_deletion_protection: If true, deletion of the load balancer will be disabled via the AWS API. This will prevent Terraform from deleting the load balancer. Defaults to false.
tags: A mapping of tags to assign to the resource.
```

Step2: *Define the target group: This is going to provide a resource for use with Load Balancer.*

```
resource "aws_lb_target_group" "my-alb-tg" {
  health_check {
    interval            = 30
    path                = "/"
    protocol            = "HTTP"
    timeout             = 5
    healthy_threshold   = 5
    unhealthy_threshold = 2
    matcher             = "200-299"
  }

  name        = "my-alb-tg"
  port        = 80
  protocol    = "HTTP"
  vpc_id      = "${var.vpc_id}"
  target_type = "instance"
}
```

target_group.tf hosted with ❤ by GitHub · view raw

```
health_check: Your Application Load Balancer periodically sends requests to its registered targets to test their status. These tests are called health checks
interval: The approximate amount of time, in seconds, between health checks of an individual target. Minimum value 5 seconds, Maximum value 300 seconds. Default 30 seconds.
path: The destination for the health check request
protocol: The protocol to use to connect with the target. Defaults to HTTP
timeout: The amount of time, in seconds, during which no response means a failed health check. For Application Load Balancers, the range is 2 to 60 seconds and the default is 5 seconds
healthy_threshold: The number of consecutive health checks successes required before considering an unhealthy target healthy. Defaults to 3.
unhealthy_threshold: The number of consecutive health check failures required before considering the target unhealthy
matcher: The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299")

name: The name of the target group. If omitted, Terraform will assign a random, unique name.
port: The port on which targets receive traffic
protocol: The protocol to use for routing traffic to the targets. Should be one of "TCP", "TLS", "HTTP" or "HTTPS". Required when target_type is instance or ip
vpc_id: The identifier of the VPC in which to create the target group
target_type: The type of target that you must specify when registering targets with this target group.Possible values instance id, ip address
```

Step3: *Provides the ability to register instances with an Application Load Balancer (ALB)*

```
resource "aws_lb_target_group_attachment" "my-tg-attachment1" {
  target_group_arn = "${aws_lb_target_group.my-alb-tg.arn}"
  target_id        = "${var.instance_id1}"
  port             = 80
}

resource "aws_lb_target_group_attachment" "my-tg-attachment2" {
  target_group_arn = "${aws_lb_target_group.my-alb-tg.arn}"
  target_id        = "${var.instance_id2}"
  port             = 80
}
```

aws_alb_target_attachment.tf hosted with ❤ by GitHub · view raw

```
target_group_arn: The ARN of the target group with which to register targets
target_id: The ID of the target. This is the Instance ID for an instance
port: The port on which targets receive traffic.
```

Step4: *Security group used by ALB*

```
resource "aws_security_group" "alb-sg" {
  name   = "my-alb-sg"
  vpc_id = "${var.vpc_id}"
}

resource "aws_security_group_rule" "http_allow" {
  from_port         = 80
  protocol          = "tcp"
  security_group_id = "${aws_security_group.alb-sg.id}"
  to_port           = 80
  type              = "ingress"
  cidr_blocks       = ["0.0.0.0/0"]
}

resource "aws_security_group_rule" "all_outbound_access" {
  from_port         = 0
  protocol          = "-1"
  security_group_id = "${aws_security_group.alb-sg.id}"
  to_port           = 0
  type              = "egress"
  cidr_blocks       = ["0.0.0.0/0"]
}
```

alb_security_group.tf hosted with ❤ by GitHub · view raw

- *Final terraform code for Application Load Balancer will look like this*

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_lb" "my-test-lb" {
  name               = "my-test-lb"
  internal           = false
  load_balancer_type = "application"
  ip_address_type    = "ipv4"
  subnets            = ["${var.subnet_id1}", "${var.subnet_id2}"]
  security_groups = ["${aws_security_group.alb-sg.id}"]

  enable_deletion_protection = true

  tags {
    Name = "my-test-alb"
  }
}

resource "aws_lb_target_group" "my-alb-tg" {
  health_check {
    interval            = 30
    path                = "/"
    protocol            = "HTTP"
    timeout             = 5
    healthy_threshold   = 5
    unhealthy_threshold = 2
    matcher             = "200-299"
  }

  name        = "my-alb-tg"
  port        = 80
  protocol    = "HTTP"
  vpc_id      = "${var.vpc_id}"
  target_type = "instance"
}

resource "aws_lb_target_group_attachment" "my-tg-attachment1" {
  target_group_arn = "${aws_lb_target_group.my-alb-tg.arn}"
  target_id        = "${var.instance_id1}"
  port             = 80
}

resource "aws_lb_target_group_attachment" "my-tg-attachment2" {
  target_group_arn = "${aws_lb_target_group.my-alb-tg.arn}"
  target_id        = "${var.instance_id2}"
  port             = 80
}

resource "aws_security_group" "alb-sg" {
  name   = "my-alb-sg"
  vpc_id = "${var.vpc_id}"
}

resource "aws_security_group_rule" "http_allow" {
  from_port         = 80
  protocol          = "tcp"
  security_group_id = "${aws_security_group.alb-sg.id}"
  to_port           = 80
  type              = "ingress"
  cidr_blocks       = ["0.0.0.0/0"]
}

resource "aws_security_group_rule" "all_outbound_access" {
  from_port         = 0
  protocol          = "-1"
  security_group_id = "${aws_security_group.alb-sg.id}"
  to_port           = 0
  type              = "egress"
  cidr_blocks       = ["0.0.0.0/0"]
}
```

alb.tf hosted with ❤ by GitHub · view raw

For complete terraform code with variables.tf please check the below mentioned link

GitHub Link

100daysofdevops/100daysofdevops
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.
github.com

Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- *Twitter: @100daysofdevops OR @lakhera2015*
- *Facebook: https://www.facebook.com/groups/795382630808645/*
- *Medium: https://medium.com/@devopslearning*
- *Slack: https://devops-myworld.slack.com/messages/CF41EFG49/*
- *GitHub Link:https://github.com/100daysofdevops*

Reference

100 Days of DevOps — Day 0
D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)
medium.com