Loadbalancer

Hello! And welcome back to the Terraform and Ansible for AWS course on linuxacademy!

In this lesson, we are going to create the loadbalancer, the launch configuration, and the autoscaling group.

Let’s start with the load balancer:

resource "aws\_elb" "prod" {

name = "${var.domain\_name}-prod-elb"

subnets = ["${aws\_subnet.private1.id}", "${aws\_subnet.private2.id}"]

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

listener {

instance\_port = 80

instance\_protocol = "http"

lb\_port = 80

lb\_protocol = "http"

}

health\_check {

healthy\_threshold = "${var.elb\_healthy\_threshold}"

unhealthy\_threshold = "${var.elb\_unhealthy\_threshold}"

timeout = "${var.elb\_timeout}"

target = "HTTP:80/"

interval = "${var.elb\_interval}"

}

cross\_zone\_load\_balancing = true

idle\_timeout = 400

connection\_draining = true

connection\_draining\_timeout = 400

tags {

Name = "${var.domain\_name}-prod-elb"

}

}

Let’s go through this. First we created the an aws\_elb resource with an ID of “prod”. Then we named it using the domain name variable with “prod-elb” after it. We then added the private subnets (which have public access for the load balancer, but not for the instances.), then we added it to the public security group.

After that, we created our listener which listens on port 80 for the instances and the load balancer. If you decide to attach an SSL, which I recommend, you can set the lb port to 443 with instance\_protocol as https and the instance port to 80 with instance\_protocol as 443.

Then we create the health check. I have created variables for these settings and I suggest you do the same to make modification easier. I set the target as HTTP:80/, but you can always set a custom healthcheck page if you wish.

I enable cross zone loadbalancing since we are load balancing across multiple availability zones, I leave the idle\_timeout as the default 400 seconds, and enable connection draining. All of these settings can be found in the AWS courses on LInuxAcademy as well as the AWS support site.

Finally, we tag it with a Name. You can also tag it with dev or prod if you wish to create a more robust infrastructure.