Prashant Lakhera
Mar 2, 2019 · 4 min read · Listen

100 Days of DevOps — Day 20— Auto-Scaling Group using Terraform

Welcome to Day 20 of 100 Days of DevOps, Let continue our terraform journey, yesterday we created an application load balancer, let expand that concept and put this load balancer as a part of the auto-scaling group.

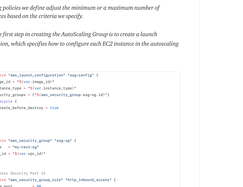
100 Days of DevOps — Day 19 - Application Load Balancer using Terraform
Welcome to Day 19 of 100 Days of DevOps, Let continue our terraform journey and see how can we create Application Load...
medium.com

What is Auto Scaling?

What auto-scaling will do, it ensures that we have a correct number of EC2 instances to handle the load of your applications.

How Auto Scaling works

- *It all started with the creation of the Auto Scaling group which is the collection of EC2 instances.*
- *We can specify a minimum number of instances and AWS EC2 Auto Scaling ensures that your group never goes below this size.*
- *The same way we can specify the maximum number of instances and AWS EC2 Auto Scaling ensures that your group never goes above this size.*
- *If we specify the desired capacity, AWS EC2 Auto Scaling ensures that your group has this many instances.*
- *Configuration templates(launch template or launch configuration): Specify Information such as AMI ID, instance type, key pair, security group*
- *If we specify scaling policies then AWS EC2 Auto Scaling can launch or terminate instances as demand on your application increased or decreased. For eg: We can configure a group to scale based on the occurrence of specified conditions(dynamic scaling) or on a schedule.*



Reference: https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

In the above example, Auto Scaling has

- *A minimum size of one instance.*
- *Desired Capacity of two instances.*
- *The maximum size of four instances.*
- *Scaling policies we define adjust the minimum or a maximum number of instances based on the criteria we specify.*

Step1: *The first step in creating the AutoScaling Group is to create a launch configuration, which specifies how to configure each EC2 instance in the autoscaling group.*

```
resource "aws_launch_configuration" "asg-config" {
  image_id = "${var.image_id}"
  instance_type = "${var.instance_type}"
  security_groups = ["${aws_security_group.asg-sg.id}"]
  lifecycle {
    create_before_destroy = true
  }
}

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

# Ingress Security Port 22
resource "aws_security_group_rule" "http_inbound_access" {
  from_port       = 80
  protocol        = "tcp"
  security_group_id = "${aws_security_group.asg-sg.id}"
  to_port         = 80
  type            = "ingress"
  cidr_blocks     = ["0.0.0.0/0"]
}

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

aws-launch-configuration.tf hosted with ❤ by GitHub · view raw

- *Most of the parameters look similar to EC2 configuration except lifecycle parameter which is required for using a launch configuration with an ASG*

100 Days of DevOps — Day 17- Creating EC2 Instance using Terraform
Welcome to Day 17 of 100 Days of DevOps, Let continue our journey, so far I lower discussed terraform fundamentals and...
medium.com

- *One of the available `lifecycle` settings are `create_before_destroy`, which, if set to `true`, tells Terraform to always create a replacement resource before destroying the original resource. For example, if you set `create_before_destroy` to `true` on an EC2 Instance, then whenever you make a change to that Instance, Terraform will first create a new EC2 Instance, wait for it to come up, and then remove the old EC2 Instance.*
- *The catch with the `create_before_destroy` the parameter is that if you set it to `true` on resource X, you also have to set it to `true` on every resource that X depends on (if you forget, you'll get errors about cyclical dependencies). In the case of the launch configuration, that means you need to set `create_before_destroy` to `true` on the security group*

Resources - Configuration Language - Terraform by HashiCorp
Resources are the most important element in a Terraform configuration. Each resource corresponds to an infrastructure...
www.terraform.io

Step2:

```
data "aws_availability_zones" "all" {}
resource "aws_autoscaling_group" "terraform-asg" {
  launch_configuration = "${aws_launch_configuration.asg-config.id}"
  availability_zones   = "${data.aws_availability_zones.all.names}"
  target_group_arns    = ["${var.target_group_arn}"]
  health_check_type    = "ELB"
  min_size             = "2"
  max_size             = "2"
  tag {
    key = "Name"
    propagate_at_launch = true
    value = "my-terraform-asg-example"
  }
}
```

aws-scaling.tf hosted with ❤ by GitHub · view raw

- *Next step is to create an auto-scaling group using the `aws_autoscaling_group` resource.*
- *This autoscaling group will spin a minimum of 1 instance and a maximum of 2 instances OR completely based on your requirement.*
- *It's going to use the launch configuration we created in the earlier step.*
- *We are using an `aws_availibility_zone` resource which will make sure instance will be deployed in different Availability Zones.*
- *A list of `aws_alb_target_group` ARNs, for use with Application Load Balancing, this is created During Day 19 when we created Application Load Balancer. If you are still not sure about the Value, in the outputs.tf file put this entry*

```
output "target_group_arn" {
  value = "${aws_lb_target_group.my-alb-tg.arn}"
}
```

and then run terraform apply command, you will see something like this

```
Outputs:

target_group_arn = arn:aws:elasticloadbalancing:us-west-2:XXXXXXX:targetgroup/my-alb-tg/05a22d09f9c37964
```

GitHub Link for the complete ASG code

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/795382630608645/*
- *Medium: https://medium.com/@devopslearning*
- *Slack: https://devops-myworld.slack.com/messages/CF41EFG4S/*
- *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

Get an email whenever Prashant Lakhera publishes.
Subscribe
By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

More from Prashant Lakhera
AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCDS, RHCE, Docker Certified,KAMAS, CCNA, MCP, Certified Jenkins, Terraform Certified, SIGCP

Mar 1, 2019
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 ...
Cloud Computing · 4 min read

Share your ideas with millions of readers. Write on Medium

Feb 26, 2019
100 Days of DevOps — Day 18-Add monitoring to these instances using Terraform(CloudWatch and SNS)
Check the updated 101 Days of DevOps Course Course Registration link: https://www.101daysofdevops.com/register/ Course Link:...
AWS · 3 min read

Feb 27, 2019
100 Days of DevOps — Day 17- Creating EC2 Instance using Terraform
Check the updated 101 Days of DevOps Course Course Registration link: https://www.101daysofdevops.com/register/ Course Link:...
AWS · 5 min read

Feb 26, 2019
100 Days of DevOps — Day 16- Building VPC using Terraform
Check the updated 101 Days of DevOps Course Course Registration link: https://www.101daysofdevops.com/register/ Course Link:...
AWS · 5 min read

Feb 25, 2019
100 Days of DevOps — Day 15- Introduction to Terraform
Check the updated 101 Days of DevOps Course Course Registration link: https://www.101daysofdevops.com/register/ Course Link:...
AWS · 15 min read

Get app · Write · Help · Sign in

Get started · Sign In
Search
Prashant Lakhera
4.7K Followers
AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCDS, RHCE, Docker Certified,KAMAS, CCNA, MCP, Certified Jenkins, Terraform Certified, SIGCP
Follow
More from Medium
Sujit Patel · Easy Steps to Create AWS VPC with Terraform
Aarti A. in AWS ... · Monitor Linux Memory Metrics in AWS...
Narendra Reddy · AWS—10
Priyam ... in AW... · AWS S3 Bucket: Access Point
Help · Status · Writers · Blog · Careers · Privacy · Terms · About · Knowable

Love podcasts or audiobooks? Learn on the go with our new app. Try Knowable
Recommended from Medium
Anita Nirmalraun · Build Your First Pypi Package
Jonathot · Faraste Gamer
ComplianceQuest · Factor #6: Scalability
Krishnankumar · Sql Server Migration To RDS :Instance Sizing and Migration best practices.
Steve Tatton · Failure is a wake-up call
ACM Rourkela · Journey to the Core of the Computer!
Pwnable.kr
Andrea Rodriguez · Airtable: The Best Tool You've Never Heard Of
Airtc