

17 – Testando Escalabilidade Via Terraform

Os objetivos dessa prática são:

- Através do Terraform:
 - Criar um Security Group;
 - Criar um Launch Configuration
 - Criar um Load Balancer;
 - Criar um Auto Scaling Group;
 - Criar um Auto Scaling Plan;
- Testar escalabilidade.

Passo a Passo

É pré-requisito dessa atividade ter o Terraform instalado em seu computador.

- 1) Copiar os arquivos localizados em <https://github.com/mclsylva/aws-sre-bootcamp/tree/main/terraform/scale> para o seu computador;
- 2) Abra os arquivos “bootstrap.sh” e “main.tf” e entenda o que eles estão fazendo;
- 3) No arquivo “main.tf” substitua o valor da chave “key_name” com o nome da chave da sua conta na AWS;
- 4) No diretório onde você copiou os arquivos, vamos iniciar o terraform com o comando “terraform init”:

```
marceloortiz@MacBook-Air-de-Marcelo-2 Scale % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.23.0

Terraform has been successfully initialized!
```

- 5) Agora digite o comando “terraform plan” para simular a execução da criação dos recursos na AWS:

```
marceloortiz@MacBook-Air-de-Marcelo-2 Scale % terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_autoscaling_group.scale_web_autoscaling_group_test will be created
+ resource "aws_autoscaling_group" "scale_web_autoscaling_group_test" {
+   arn                               = (known after apply)
+   availability_zones                = [
+     "us-east-1a",
+     "us-east-1b",
  ]
}
```

No final do retorno do comando, veja que serão criados 5 recursos:

```
Plan: 5 to add, 0 to change, 0 to destroy.
```

- 6) Agora vamos criar os recursos, com o comando “terraform apply --auto-approve”:

```
marceloortiz@MacBook-Air-de-Marcelo-2 Scale % terraform apply --auto-approve
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_autoscaling_group.scale_web_autoscaling_group_test will be created
+ resource "aws_autoscaling_group" "scale_web_autoscaling_group_test" {
+   arn                               = (known after apply)
+   availability_zones                 = [
+     "us-east-1a",
+     "us-east-1b",
+     "us-east-1c",

```

No final do retorno do comando, veja que foram criados os 5 recursos:

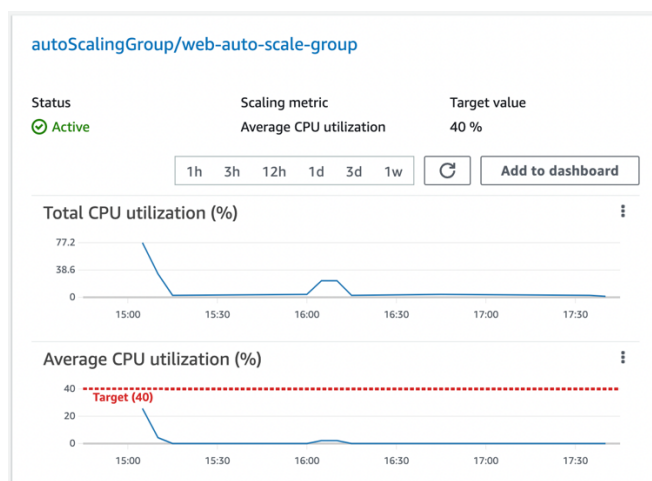
Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

- 7) Conecte no console de gerenciamento do EC2 da AWS e verifique qual é o DNS do Load Balancer. Copie o DNS Name do Load Balancer e cole no seu navegador, deverá aparecer a mensagem “Teste Escalabilidade”:

← → ↻ ⚠ Não seguro | scale-web-load-balancer-test-

Scale Test

- 8) No console de gerenciamento do Auto Scaling Plan da AWS, Clique no nome do plano (“Autoscaling-Plan”) e verifique a monitoração:



Veja que a monitoração da utilização da CPU está abaixo do alvo de 40%. Agora vamos simular uma carga em uma das instâncias para observar o Auto Scale Plan entrar em ação e criar novas instâncias para manter a utilização de processamento abaixo dos 40%, para isso:

- 9) Conecte no sistema operacional de qualquer das duas instâncias;

- 10) Digite o comando “htop” e verifique o resultado, mantenha essa sessão aberta. Nesse momento a utilização da CPU está muito baixa.

```
CPU[ 0.0%] Tasks: 38, 72 thr; 1 running
Mem[101M/966M] Load average: 0.05 0.07 0.03
Swp[ 0K/0K] Uptime: 02:47:42
```

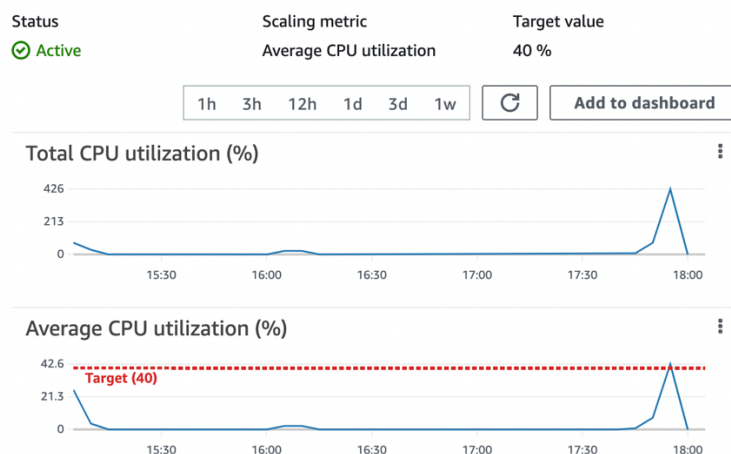
- 11) Em outra sessão, no sistema operacional digite o comando “stress --cpu 1”:

```
[ec2-user@ip-172-31-15-245 ~]$ stress --cpu 1
stress: info: [4746] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

- 12) Volte na sessão onde está em execução o “htop” e veja que agora a CPU dessa instância está com 100% de utilização:

```
CPU[100.0%] Tasks: 44, 72 thr; 2 running
Mem[106M/966M] Load average: 0.60 0.21 0.08
Swp[ 0K/0K] Uptime: 02:49:51
```

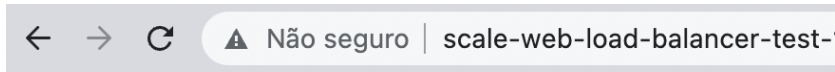
- 13) Volte para a tela de monitoração do “Auto Scale Plan” e acompanhe o gráfico de utilização por alguns minutos:



Veja que após alguns minutos tivemos uma taxa de utilização de CPU acima de 40%, com isso, automaticamente o “Auto Scale Plan” iniciou novas instâncias “EC2” para acomodar a utilização de CPU para dentro do alvo de 40%.

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	web-teste-template	i-092e47ccefc33bc5f5	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	web-teste-template	i-0d411e53726a068ae	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	web-teste-template	i-0be1db3e979ac612d	Running	t2.micro	Initializing
<input type="checkbox"/>	web-teste-template	i-093413a87ea6d789e	Running	t2.micro	Initializing
<input type="checkbox"/>	web-teste-template	i-0864615532743b55e	Running	t2.micro	Initializing

No nosso teste, foram adicionadas 3 novas instâncias. Vejam que durante todo o processo nosso servidor Apache continuou respondendo as requisições:



Scale Test

E assim finalizamos nossos testes, não esqueçam de limpar o ambiente, para isso digite o comando “terraform destroy --auto-approve”.