

## 17 – Testando Escalabilidade

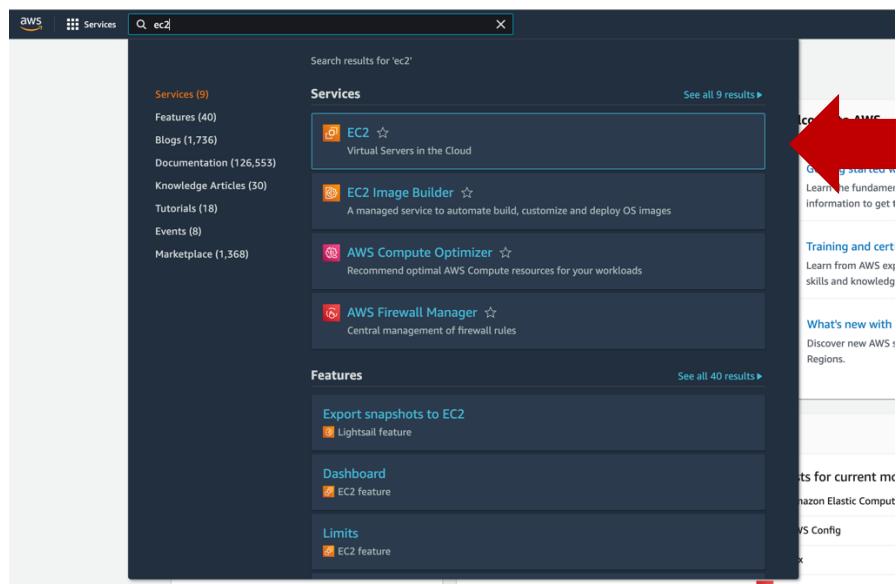
Os objetivos dessa prática são:

- Criar uma instância Linux EC2 e instalar o Apache;
- Criar uma imagem AMI do servidor criado;
- Criar um Load Balancer;
- Criar um Auto Scaling Group;
- Criar um Auto Scaling Plan;
- Testar escalabilidade.

### Passo a Passo

Primeiro vamos criar a instância EC2 que vai servir de modelo para o nosso Auto Scaling.

- 1) Acesse a console de gerenciamento da AWS e no campo de pesquisa digite “EC2”:



- 2) Clique em “EC2 (Virtual Servers in the Cloud)”.
- 3) Na tela de Console do ECS, clique em “Launch instance”:

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays 'Resources' for the US East (N. Virginia) Region, showing 5 instances (running), 2 key pairs, 6 security groups, and 5 volumes. Below this, a callout box suggests using the AWS Launch Wizard for Microsoft SQL Server Always On availability groups. On the right, there's a 'Service health' section indicating the service is operating normally.

- 4) Na tela “Launch an instance”, preencher o campo “Name” com “web-test”:

The screenshot shows the 'Launch an instance' wizard. It starts with an introduction about creating virtual machines. The first step, 'Name and tags', has a 'Name' input field where 'web-test' is typed. A red circle highlights this input field. Below it, there's a link to 'Add additional tags'. The next step, 'Application and OS Images (Amazon Machine Image)', is partially visible.

- 5) Role a tela para a seção “Application and OS Images (Amazon Machine Image) e selecione as opções:
- Amazon Linux AWS
  - Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type
  - 64-bit (x86)

- 6) Na seção “Instance Type” e selecione “t2.micro”:

- 7) Na seção “Key pair” selecione a chave da sua conta:

- 8) Na seção “Network Setting” selecione a opção “Allow HTTP traffic from the internet”:

▼ Network settings Edit

Network  
vpc-09ca0d9126e730b6e

Subnet  
No preference (Default subnet in any availability zone)

Auto-assign public IP  
Enable

**Firewall (security groups) Info**

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group    Select existing security group

We'll create a new security group called '**launch-wizard-6**' with the following rules:

Allow SSH traffic from Anywhere  
0.0.0.0/0

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

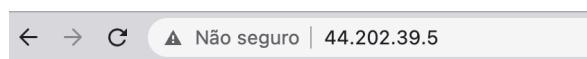
- 9) Na seção “Advanced Details” vá até o campo “User data” e coloque os comandos abaixo:

```
#!/bin/bash
yum update -y
yum install httpd -y
echo "<html><body><h1>Teste Escalabilidade</h1></body></html>" >/var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```

- 10) E clique em “Launch Instance”:



- 11) Assim que a instância EC2 estiver em execução, copie e cole o endereço IP público no seu navegador e verifique se o Apache está respondendo corretamente:



Agora vamos criar uma imagem dessa instância EC2, essa vai ser a imagem que vamos usar para o nosso Auto Scaling, para isso:

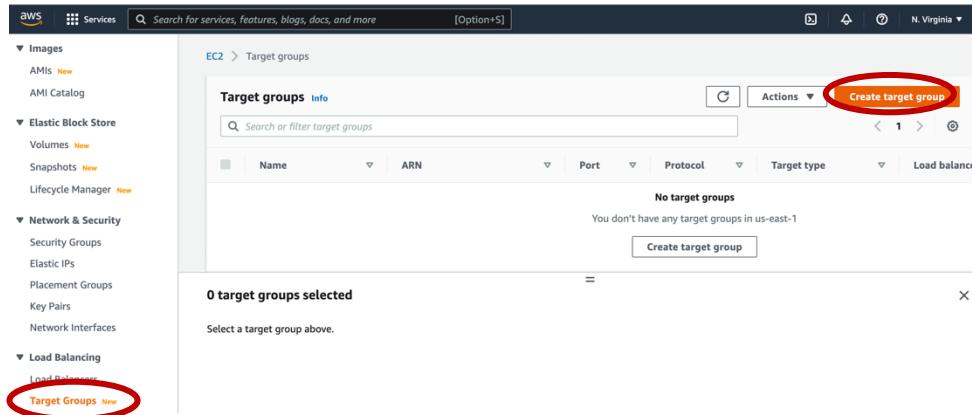
- 12) Selecione a instância EC2 “web-teste”, clique em “Actions”, em “Image and templates” e em “Create template from instance”:

- 13) Na tela “Create launch template” preencha o campo “Launch template name – required” com “template-web-server-httdp”:

- 14) No campo “Resource tag”, preencha com o valor “web-teste-template” e clique em “Create launch template”:

Chegou a hora de criar o Load Balancer, para isso:

- 15) No console do EC2, na seção “Load Balancing” clique em “Target Groups” e “Create target group”:



- 16) Na tela “Specify group detail”, preencha o campo “Name” com “web-target-group”:

Target group name  
 A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

- 17) Na seção “Health Checks” preencha o campo “Health check path” com o valor “/index.html” e clique em “Next”:

**Health checks**  
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

Health check path  
Use the default path of “/” to ping the root, or specify a custom path if preferred.  
 Up to 1024 characters allowed.

► Advanced health check settings

► Tags - optional  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel

18) Na tela “Register targets” selecione a instância “web-teste” e clique em “Include as pending below”:

The screenshot shows the 'Available instances' table with one item listed:

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-015e21ab8d0506d0a	web-teste	running	launch-wizard-6	us-east-1b	subnet-00cd41544ca7b6181

Below the table, it says "1 selected". Underneath, there's a section for "Ports for the selected instances" with a dropdown set to "80". At the bottom right of this section is a button labeled "Include as pending below" which is circled in red.

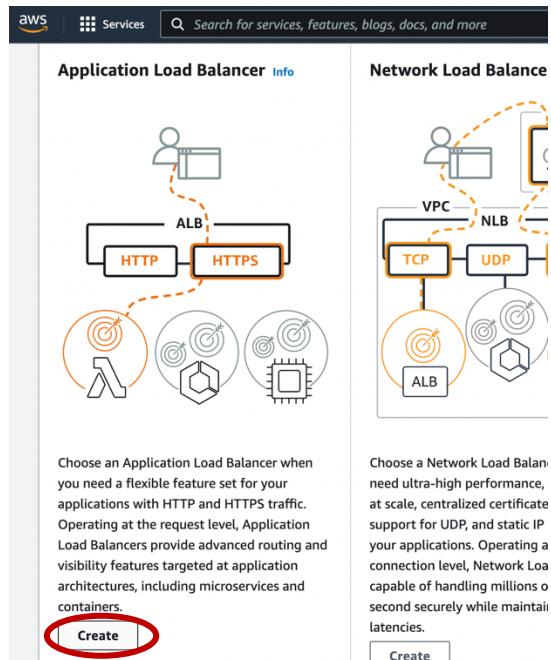
19) E clique em “Create target group”:



20) Agora vamos criar nosso Load Balancer, para isso clique em “Load Balancers” e em “Create Load Balancer”:

The screenshot shows the AWS Services menu with the "Load Balancing" section expanded. Under "Load Balancing", the "Load Balancers" link is highlighted with a red circle. On the right, there is a search bar and a "Create Load Balancer" button which is also circled in red.

- 21) Na tela “Load balancer types” clique em “Create” na seção “Application Load Balancer”:



- 22) Na tela “Create Application Load Balancer”, preencha o campo “Load balancer name” com “web-load-balancer”:

This is a screenshot of the 'Basic configuration' step. It shows a 'Load balancer name' field containing 'web-load-balancer', which is circled in red. Below the field, a note states: 'Name must be unique within your AWS account and cannot be changed after the load balancer is created.' A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

- 23) Em “Network mapping” selecione todas as zonas de disponibilidade:

This is a screenshot of the 'Network mapping' step. It shows two sections for 'us-east-1'. In the first section, 'us-east-1a' is selected with a checked checkbox and circled in red. In the second section, 'us-east-1b' is also selected with a checked checkbox and circled in red. Both sections include a 'Subnet' dropdown menu and an 'IPv4 settings' section indicating 'Assigned by AWS'.

- 24) No campo “Security Group” selecione o mesmo security group que você criou para a sua instância EC2, no meu caso foi a “launch-wizard-6”:

The screenshot shows the AWS Security Groups page. At the top, there's a header with "Security groups" and a "Info" link. Below it, a note says: "A security group is a set of firewall rules that control the traffic to your load balancer." Under the heading "Security groups", there's a dropdown menu labeled "Select up to 5 security groups". Below the dropdown, there's a "Create new security group" button. A red circle highlights the "launch-wizard-6 sg-05b9d8c88266edf00" entry in the list, which includes the ARN "arn:aws:ec2:vpc-09ca0d9126e730b6e". There's also a delete button next to the entry.

- 25) No campo “Listener and routing” selecione o target group chamado “web-target-group”:

The screenshot shows the AWS Listeners and routing page. At the top, there's a header with "Listeners and routing" and a "Info" link. Below it, a note says: "A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created." Under the heading "Listener HTTP:80", there's a table with columns "Protocol", "Port", "Default action", and "Info". The "Protocol" is set to "HTTP" and "Port" is "80". The "Default action" dropdown is set to "Forward to" and the "Info" link shows "web-target-group Target type: Instance, IPv4". A red circle highlights this entire row. There's also a "Create target group" button below the table. At the bottom, there's an "Add listener" button.

- 26) E clique em “Create load balancer”:

The screenshot shows a modal dialog box for creating a load balancer. It has two buttons at the bottom: "Cancel" and "Create load balancer". The "Create load balancer" button is highlighted with a red circle.

- 27) Aguarde até que o status do Load Balancer seja “Active”:

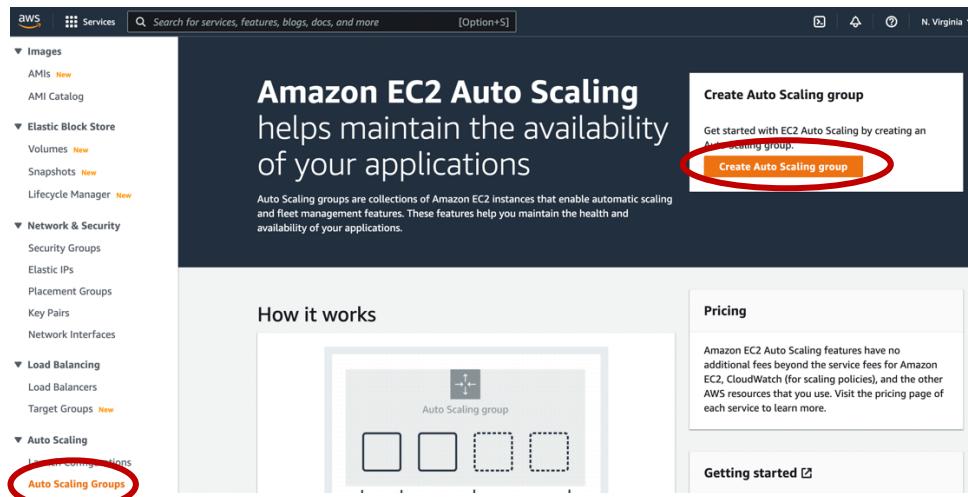
The screenshot shows the AWS Load Balancers page. At the top, there's a "Create Load Balancer" button and an "Actions" dropdown. Below it, there's a search bar with "search : web-load-balancer" and a "Add filter" button. The main table lists load balancers with columns "Name", "DNS name", and "State". One entry is highlighted with a red circle: "web-load-balancer" with "web-load-balancer-1590293..." as the DNS name and "Active" as the state.

- 28) Copie o DNS Name do Load Balancer e cole no seu navegador, deverá aparecer a mensagem “Teste Escalabilidade”:

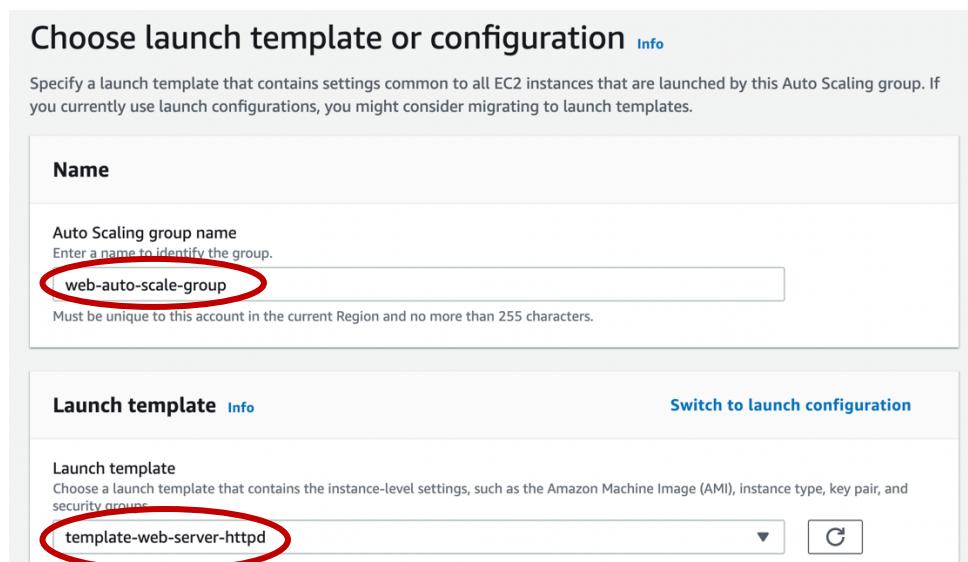
The screenshot shows a browser window. The address bar shows the URL "web-load-balancer-1590293...". Below the address bar, there's a message: "← → C 🔍 Não seguro | web-load-balancer-1590293...". Underneath the message, the text "Teste Escalabilidade" is displayed in a large, bold font.

Agora vamos criar o Auto Scaling Group, para isso:

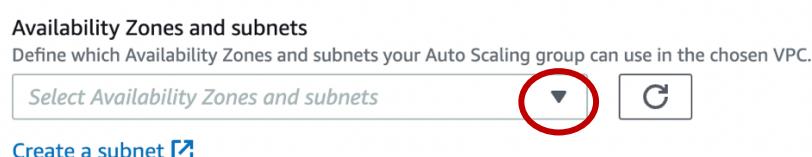
- 29) Ainda no console do EC2, na seção “Auto Scaling” clique em “Auto Scaling Groups” e em “Create Auto Scaling group”:



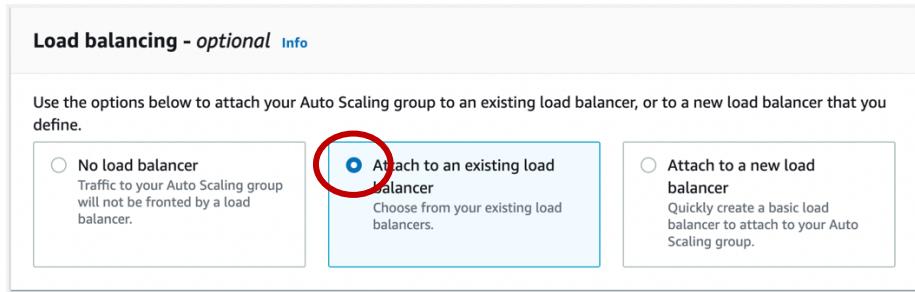
- 30) Na tela “Choose launch template or configuration” preencha o campo “Name” com “web-auto-scale-group”, no campo “Launch template” escolha o template que acabamos de criar, o “template-web-server-htpd” e clique em “Next”:



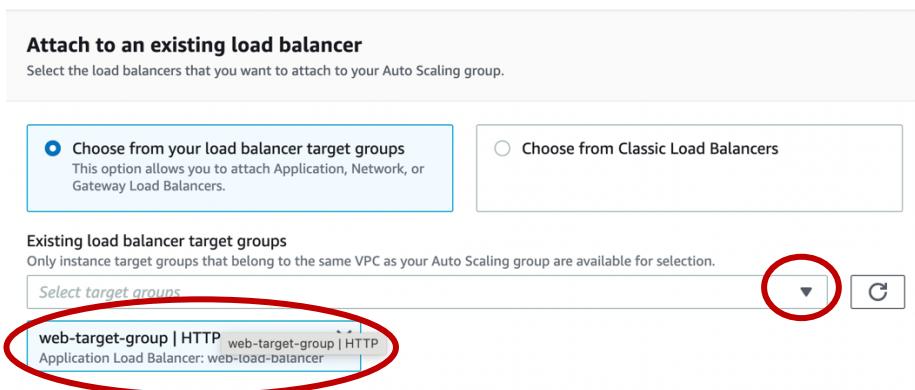
- 31) Na tela “Choose instance launch options” selecione todas as zonas de disponibilidade no campo “Availability Zones and subnets” e clique em “Next”:



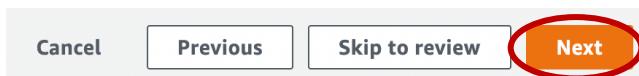
- 32) Na tela “Configure advanced options”, na seção “Load Balancing” clique em “Attach to an existing load balancer”:



- 33) Na seção “Attach to an existing load balancer” escolha o load balancer group que acabamos de criar, o “web-target-group”:

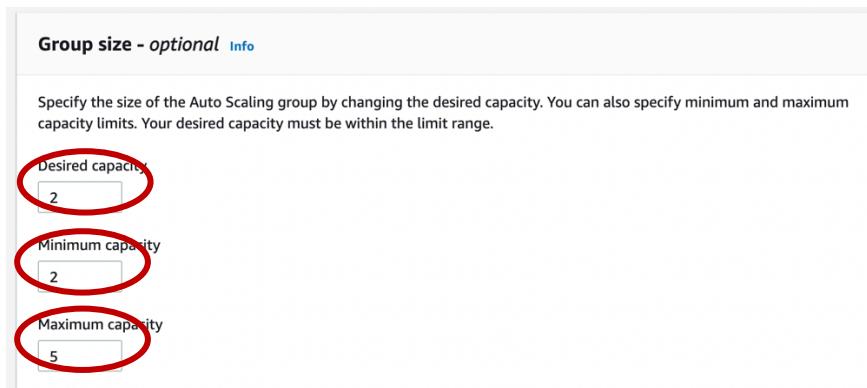


- 34) Clique em “Next”:

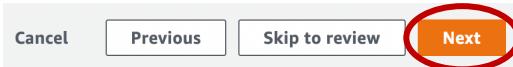


- 35) Na tela “Configure group size and scaling policies”, na seção “Group size” preencha os campos:

- Desired capacity: 2
- Minimum capacity: 2
- Maximum capacity: 5



36) Clique em “Next”:



37) Na tela “Add notification” clique em “Next”:



38) Na tela “Add tags” clique em “Next”:



39) Na tela “Review” clique em “Create Auto Scaling group”:



Note que assim que o Auto Scaling Group finalizar sua criação teremos duas novas instâncias EC2 em execução. O Auto Scaling Group é uma configuração declarativa, portanto ele sempre vai manter duas instâncias (que configuramos como capacidade desejada e mínima) em execução.

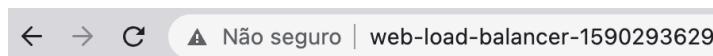
40) Verifique que agora temos 3 instâncias em execução no dashboard de instâncias do EC2, a instância que utilizamos como origem das imagens (web-teste) e mais duas instâncias do nosso Auto Scaling Group (web-teste-template).

Instances (3) Info							
		Search		Actions		Launch instances	
Instance state = running		Clear filters		Instance state		Actions	
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z	
web-teste-template	i-092e47cccf33bc5f5	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	
web-teste-template	i-0d411e53726a068ae	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1f	
web-teste	i-015e21ab8d0506d0a	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	

41) Nesse momento podemos interromper a instância “web-teste” e deixar em execução somente as duas instâncias do Auto Scaling Group, para isso, selecione a instância “web-teste”, clique em “Instance State” e em “Terminate instance”:

Instances (1/3) Info							
		Search		Actions		Instance state	
Instance state = running		Clear filters		Actions		Stop instance	Start instance
Name	Instance ID	Instance state	In	Actions	Reboot instance	Hibernate instance	us check
web-teste-template	i-092e47cccf33bc5f5	Running	t2.micro				2/2 checks passed
web-teste-template	i-0d411e53726a068ae	Running	t2.micro				2/2 checks passed
<input checked="" type="checkbox"/> web-teste	i-015e21ab8d0506d0a	Running	t2.micro				2/2 checks passed

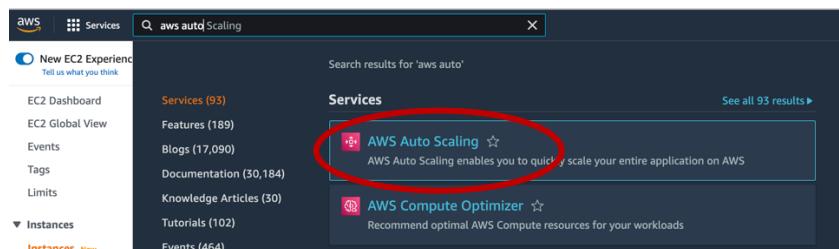
- 42) Veja que mesmo terminando a instância “web-teste”, ao invocar a URL do load balancer no navegador, o apache continua respondendo, isso porque ele está distribuído a requisições entre as duas instâncias do Auto Scale Group”:



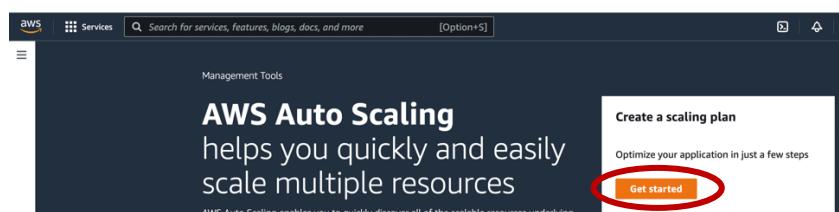
## Teste Escalabilidade

Chegou a hora de configurarmos o “Auto Scaling Plan”, para escalar a quantidade de instância dependendo da carga de processamento, para isso:

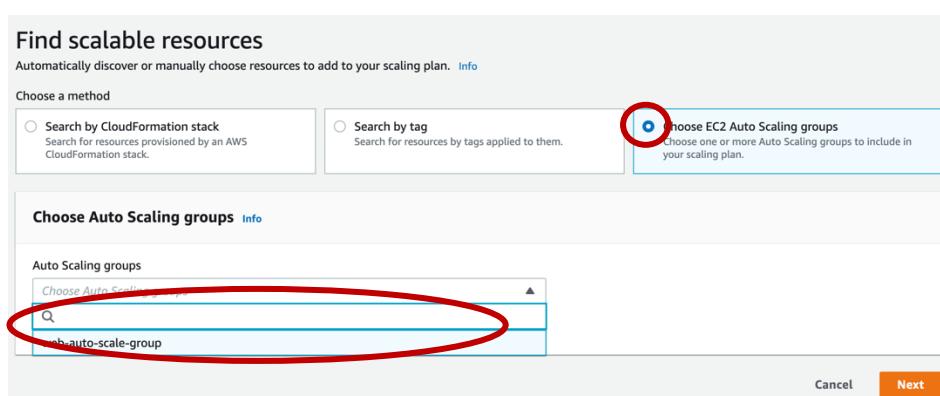
- 43) Em “Services”, procure pelo serviço chamado “AWS Auto Scaling” e clique em “AWS Auto Scaling”:



- 44) Na console do “AWS Auto Scaling” clique em “Get Started”:



- 45) Na tela “Find scalable resources”, clique na opção “Choose EC2 Auto Scaling groups” e no campo “Auto Scaling groups” escolha o grupo que acabamos de criar, com o nome de “web-auto-scale-group” e clique em “Next”:



- 46) Na tela “Specify scaling strategy”, na seção “Scaling plan detail”, preencha o campo “Name” com “web-auto-scale-plan-teste”:

**Specify scaling strategy**  
Scaling strategies define how to optimize the scalable resources in your Auto Scaling group.

**Scaling plan details**

Name  
 Must be 1-128 characters long and should not contain the pipe "|", colon

Resources  
1 Auto Scaling group was selected.

- 47) Na seção “Auto Scaling groups” marque a opção “Optimize for availability” e desmarque a opção “Enable predictive scaling” e clique em “Next”:

**Auto Scaling groups (1)**  
Specify a scaling strategy for 1 Auto Scaling group.

**Optimize for availability**  
Keep the average CPU utilization of your Auto Scaling groups at 40% to provide high availability and ensure capacity to absorb spikes in demand.

**Balance availability and cost**  
Keep the average CPU utilization of your Auto Scaling groups at 50% to provide optimal availability and reduce costs.

**Optimize for cost**  
Keep the average CPU utilization of your Auto Scaling groups at 70% to ensure lower costs.

**Custom**  
Choose your own scaling metric, target value, and other settings.

**Enable predictive scaling**  
Support your scaling strategy by continually forecasting load and proactively scheduling capacity ahead of when you need it. [Info](#)

**Enable dynamic scaling**  
Support your scaling strategy by creating target tracking scaling policies to monitor your scaling metric and increase or decrease capacity as you need it. [Info](#)

▶ Configuration details

Cancel Previous **Next**

- 48) Na tela “Configure advanced settings (optional)” clique em “Next”:

Cancel Previous **Next**

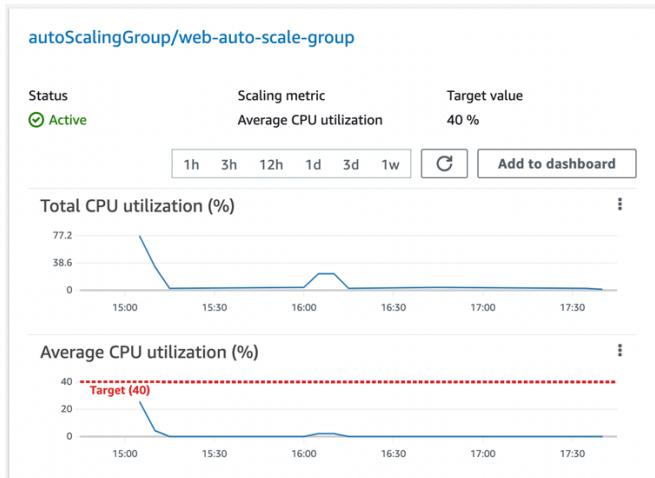
- 49) Na tela “Review and Create” clique em “Create scaling plan”:

Cancel Previous **Create scaling plan**

- 50) Na tela “Scaling plans” aguarde até que o plano que acabamos de criar fique no status “Active”:

Scaling plans (1) <a href="#">Info</a>	
	Status
<input type="checkbox"/> web-auto-scale-plan-teste	Active

- 51) Clique no nome do plano (“web-auto-scale-plan-teste”) 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:

- 52) Conecte no sistema operacional de qualquer uma das duas instâncias;  
53) No sistema operacional digite o comando “sudo amazon-linux-extras install epel -y”:

```
[ec2-user@ip-172-31-15-245 ~]$ sudo amazon-linux-extras install epel -y
```

- 54) No sistema operacional digite o comando “sudo yum install htop -y”:

```
[ec2-user@ip-172-31-15-245 ~]$ sudo yum install htop -y
```

- 55) No sistema operacional digite o comando “sudo yum install stress -y”:

```
[ec2-user@ip-172-31-15-245 ~]$ sudo yum install stress -y
```

- 56) 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
```

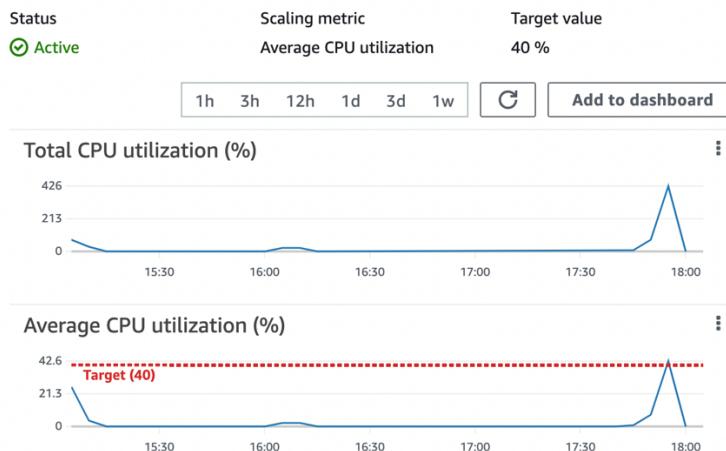
- 57) Em outra sessão, no sistema operacional digite o comando “stress --cpu 1”:

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

58) 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
```

59) 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%.

	Name	Instance ID	Instance state	Instance type	Status check
□	web-teste-template	i-092e47ccf33bc5f5	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>
□	web-teste-template	i-0d411e53726a068ae	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>
□	web-teste-template	i-0be1db3e979ac612d	<span>Running</span>	t2.micro	<span>Initializing</span>
□	web-teste-template	i-093413a87ea6d789e	<span>Running</span>	t2.micro	<span>Initializing</span>
□	web-teste-template	i-0864615532743b55e	<span>Running</span>	t2.micro	<span>Initializing</span>

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



## Teste Escalabilidade

E assim finalizamos nossos testes. Não esqueçam de limpar o ambiente, para isso remova:

- O “Auto Scaling Plan” (clique no plano e em “Delete”);
- O “Auto Scaling Group” (clique no grupo e em “Delete”);
- O “Load Balancer” (clique no load balancer, em “Actions” e depois em “Delete”);

- O “Target Group” (clique no target group, em “Actions” e depois em “Delete”).

Não é necessário remover as instâncias, pois uma vez removendo o Auto Scaling Group todas as instâncias criadas serão terminadas automaticamente.