17 – Testando Escalabilidade

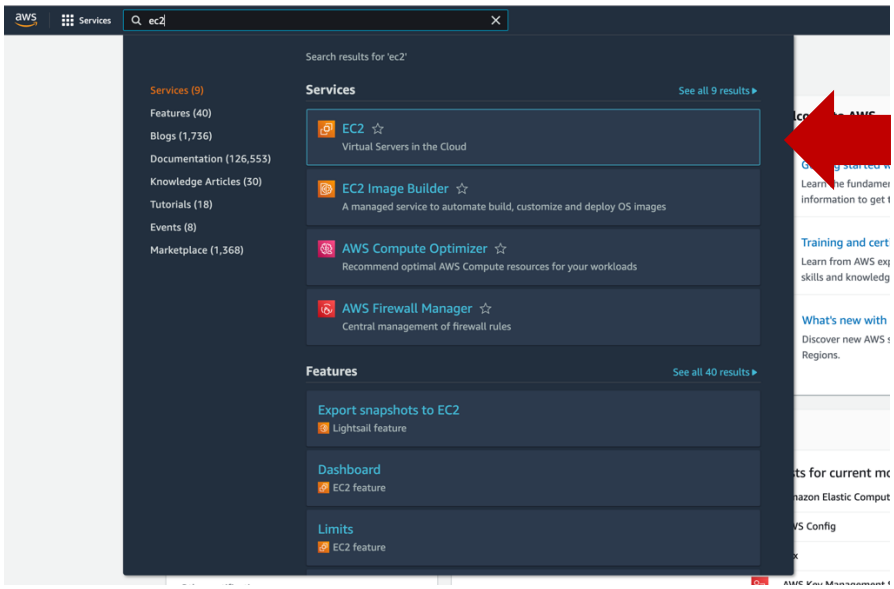
Os objetivos dessa prática são:

- 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

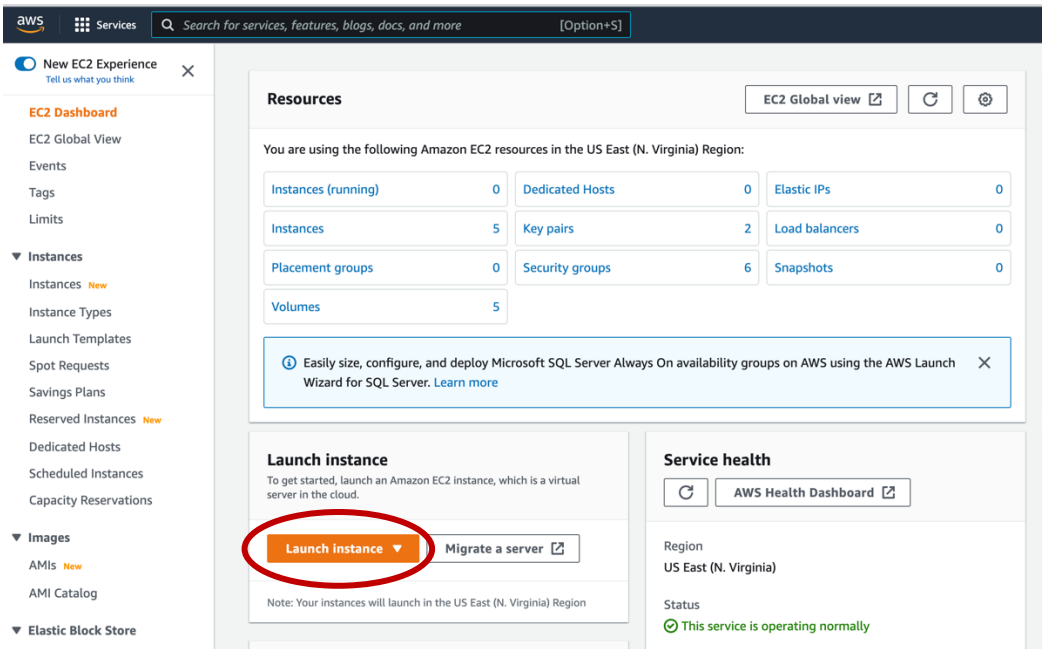
Primeiro vamos criar a instância EC2 que vai servir de modelo para o nosso Auto Scaling.

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

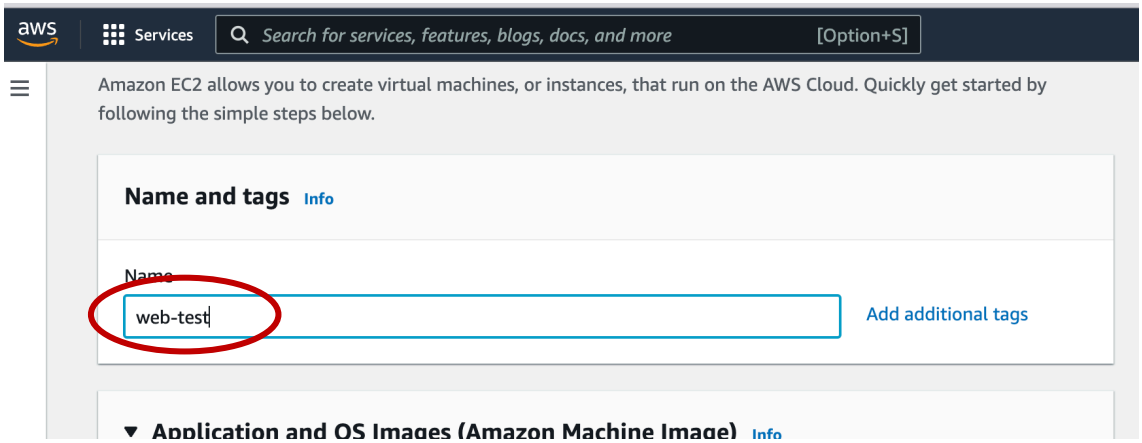


2) Clique em “EC2 (Virtual Servers in the Cloud)”.

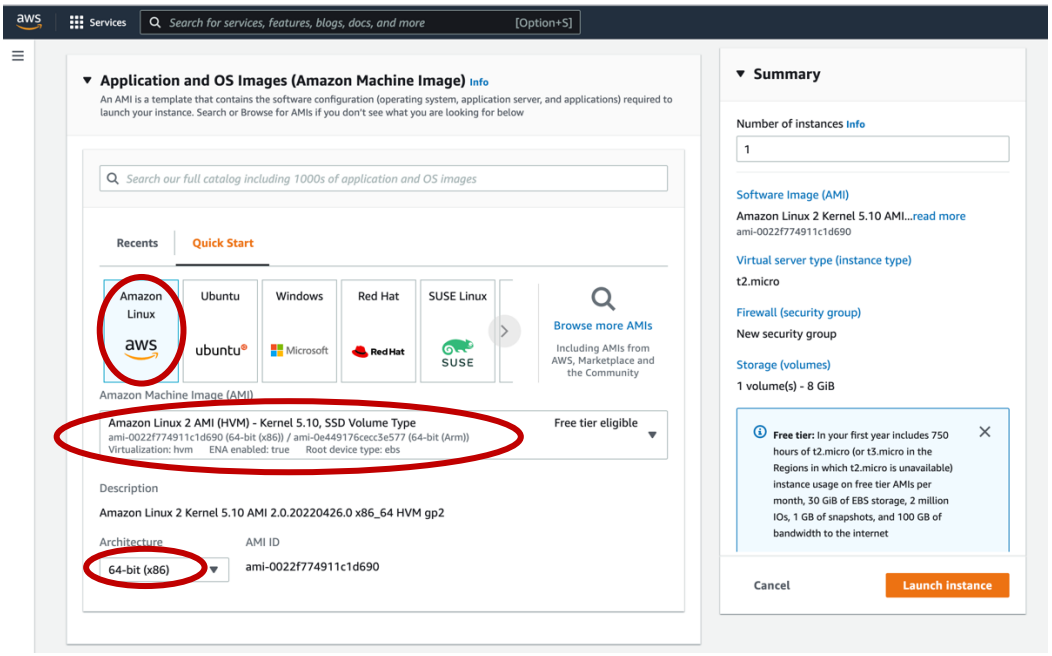
3) Na tela de Console do ECS, clique em “Launch instance”:



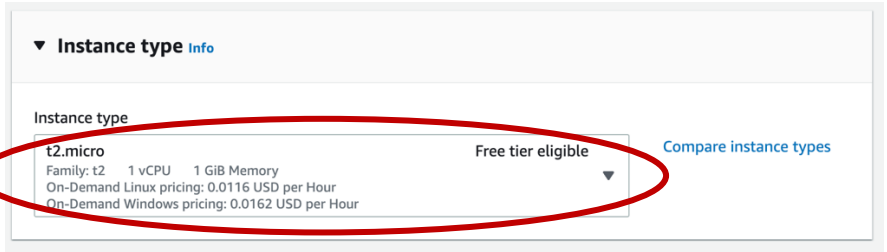
4) Na tela “Launch an instance”, preencher o campo “Name” com “web-test”:



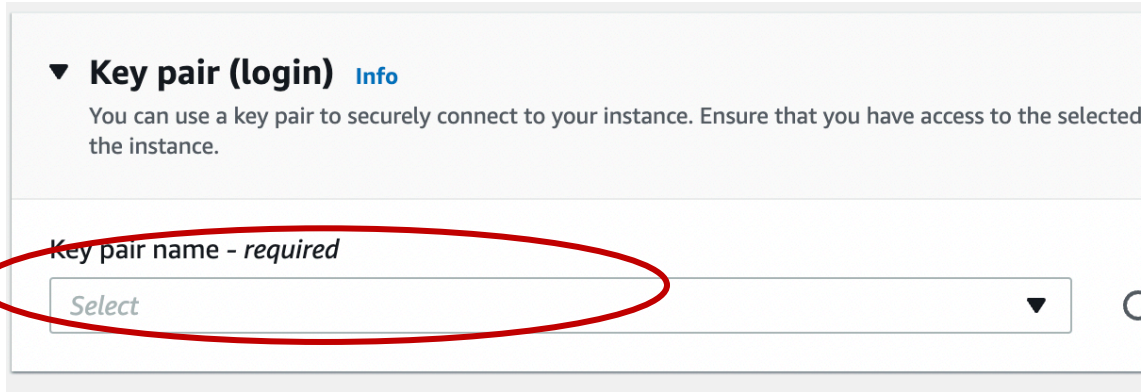
5) Role a tela para a seção “Application and OS Images (Amazon Machine Image) e selecione as opções:
   a. Amazon Linux AWS
   b. Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type
   c. 64-bit (x86)



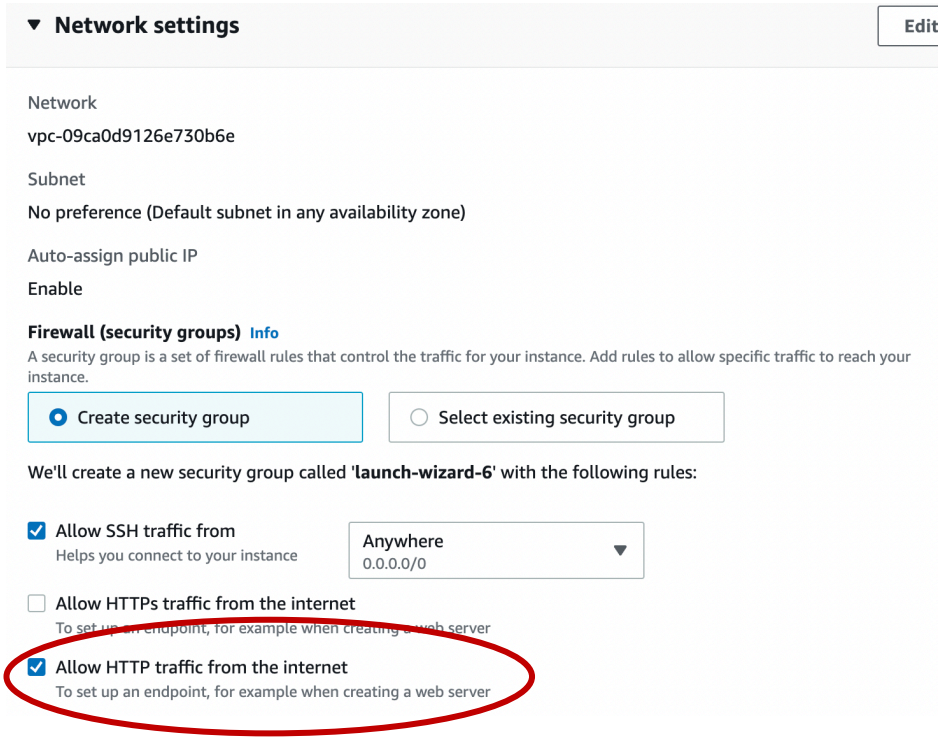
6) Na seção “Instance Type” e selecione “t2.micro”:



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”:



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

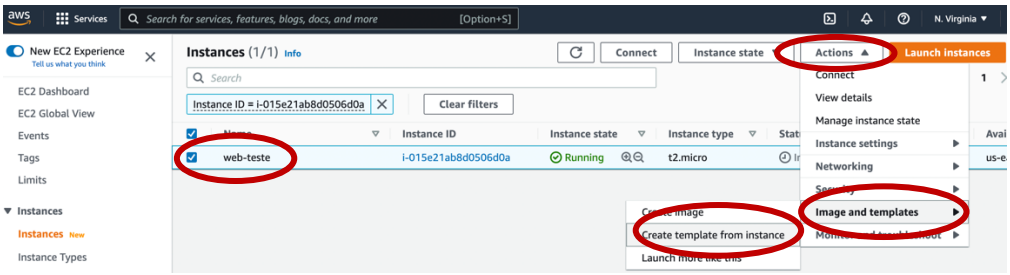
```
#!/bin/bash
yum update -y
yum install httpd -y
echo "<html><body><h1>Teste Escalabilidade</h1></body></html>"
>/var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```

10) E clique em “Launch Instance”:

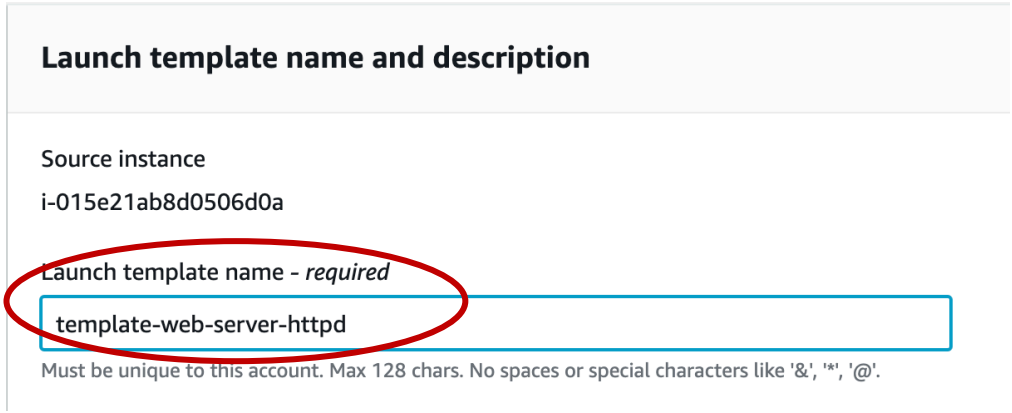
11) Assim que a instância EC2 estiver em execução, copie e cole o endereço IP público no seu navegador e verifique se o Apache está respondendo corretamente:

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

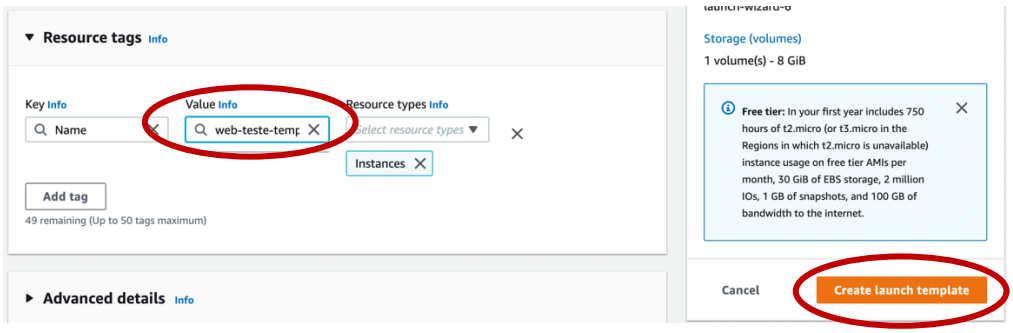
12) Selecione a instância EC2 “web-teste”, clique em “Actions”, em “Image and templates” e em “Create template from instance”:



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

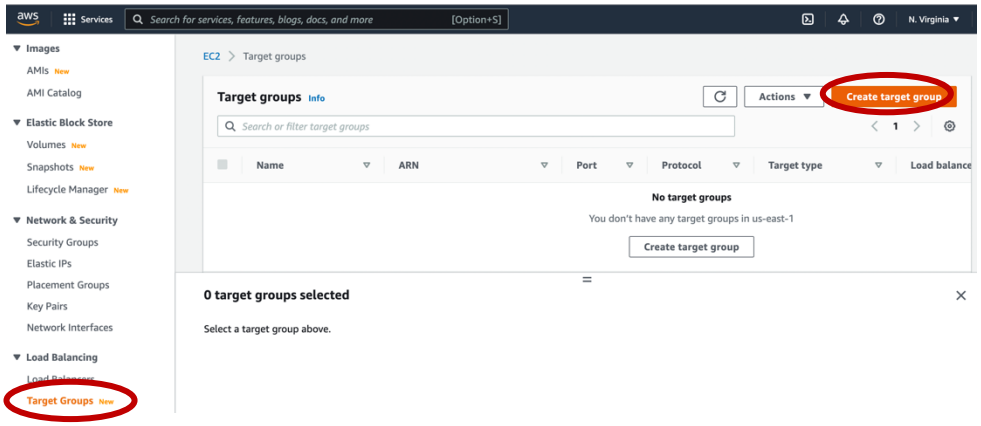


14) No campo “Resource tag”, preencha com o valor “web-teste-template” e clique em “Create launch template”:

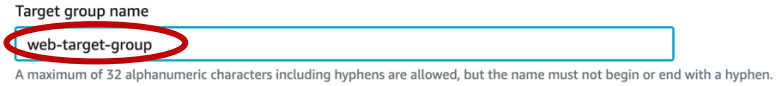


Chegou a hora de criar o Load Balancer, para isso:

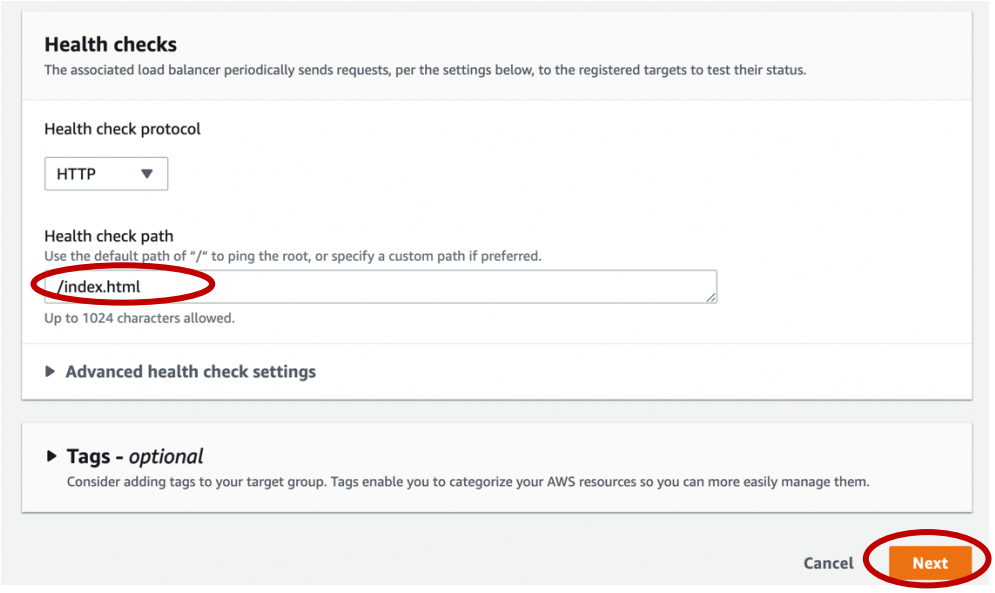
15) No console do EC2, na seção “Load Balancing”clique em “Target Groups” e “Create target group”:



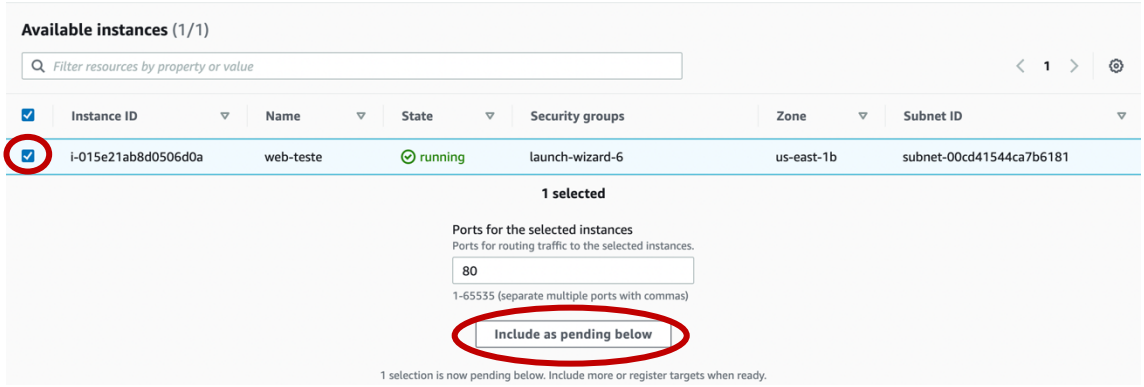
16) Na tela “Specify group detail”, preencha o campo “Name” com “web-target-group”:



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

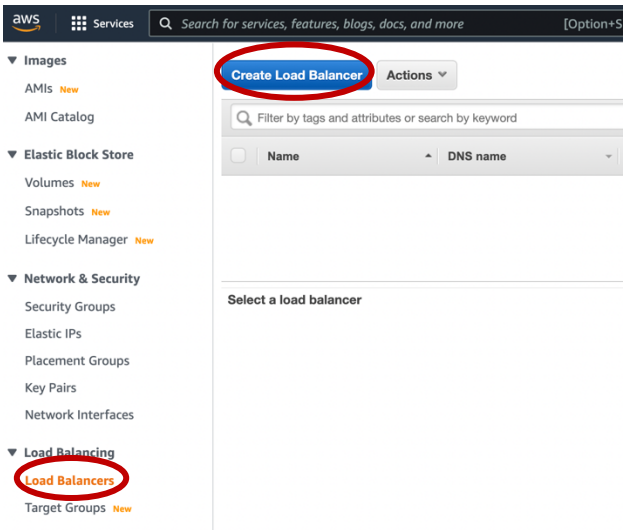


18) Na tela “Register targets” selecione a instância “web-teste” e clique em “Include as pending below”:

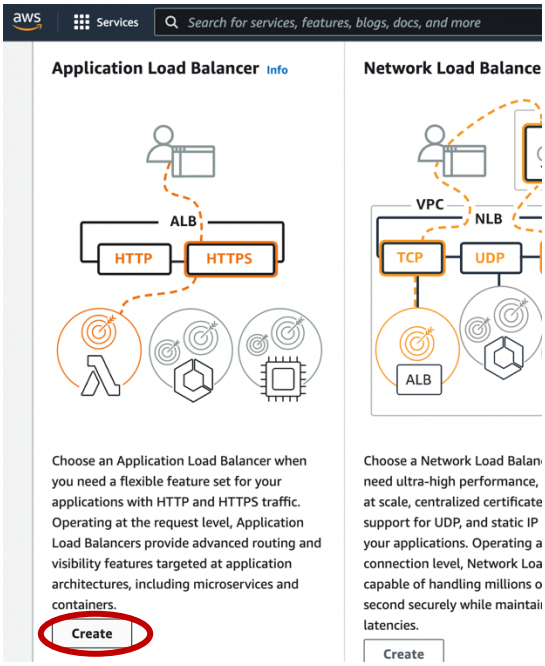


19) E clique em “Create target group”:

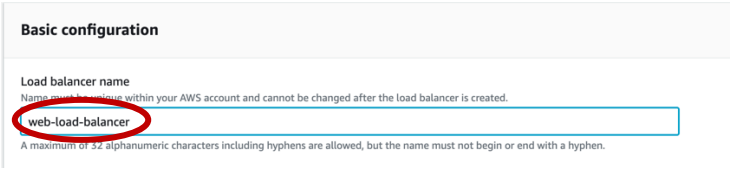
20) Agora vamos criar nosso Load Balancer, para isso clique em “Load Balancers” e em “Create Load Balancer”:



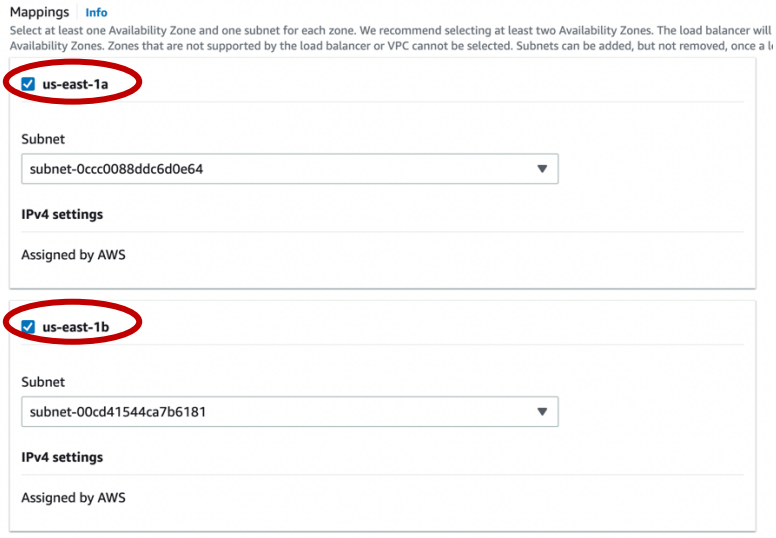
21) Na tela “Load balancer types” clique em “Create” na seção “Application Load Balancer”:



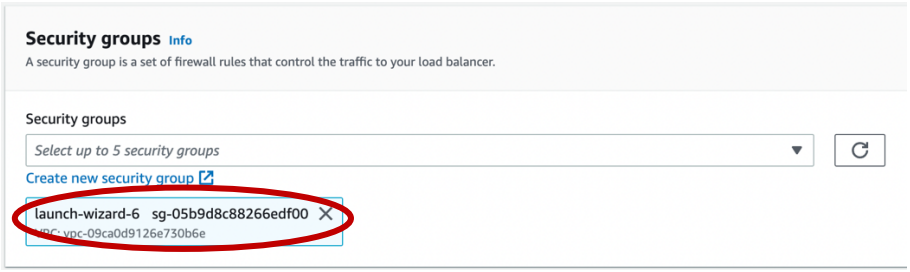
22) Na tela “Create Application Load Balancer”, preencha o campo “Load balancer name” com “web-load-balancer”:



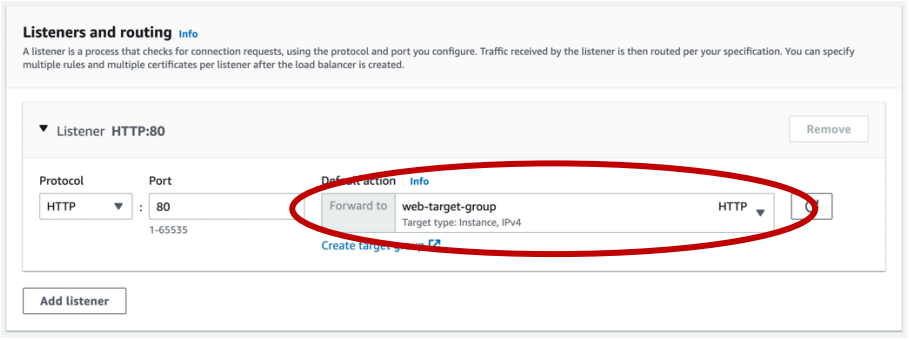
23) Em “Network mapping” selecione todas as zonas de disponibilidade:



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”:

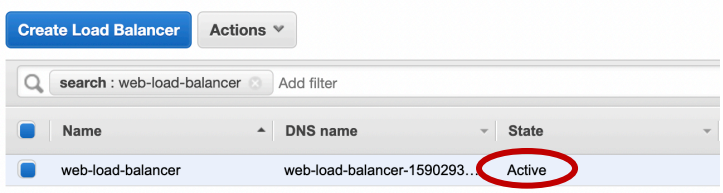


25) No campo “Listener and routing”selecione o target group chamado “web-target-group”:

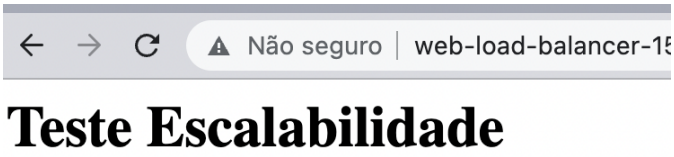


26) E clique em “Create load balancer”:

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

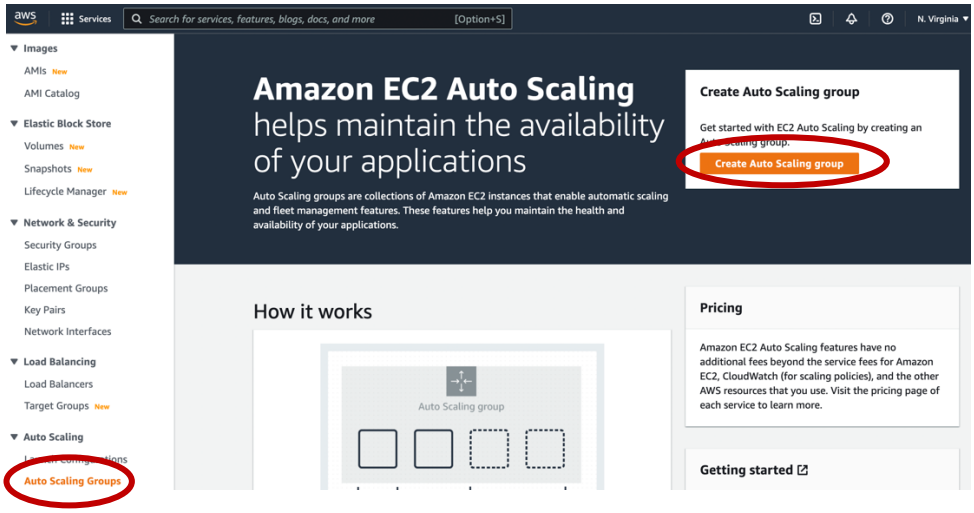


28) Copie o DNS Name do Load Balancer e cole no seu navegador, deverá aparecer a mensagem “Teste Escalabilidade”:

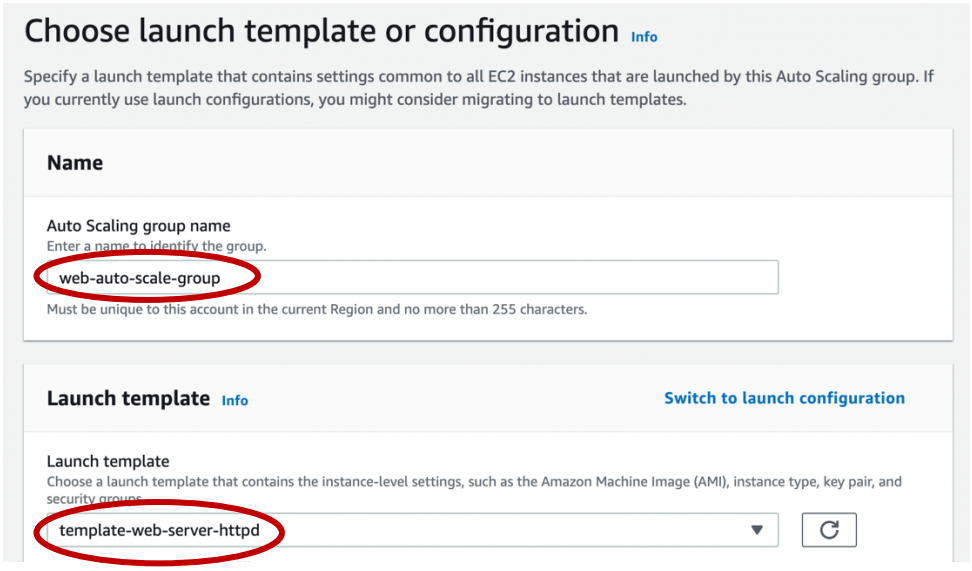


Agora vamos criar o Auto Scaling Group, para isso:

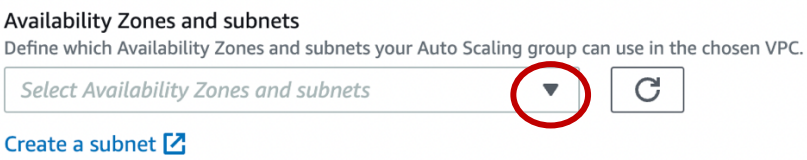
29) Ainda no console do EC2, na seção “Auto Scaling” clique em “Auto Scaling Groups” e em “Create Auto Scaling group”:



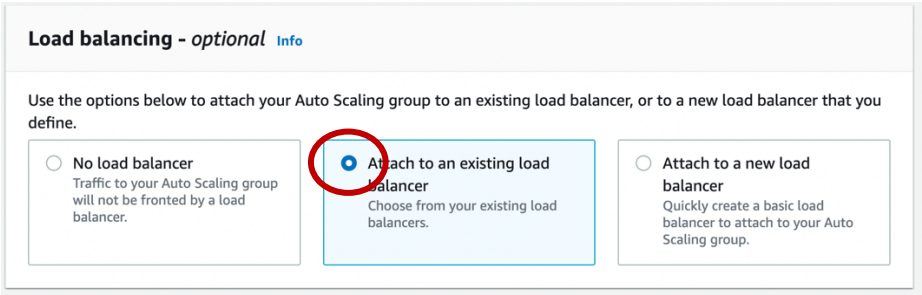
30) Na tela “Choose launch template or configuration” preencha o campo “Name” com “web-auto-scale-group”, no campo “Launch template” escolha o template que acabamos de criar, o “template-web-server-httpd” e clique em “Next”:



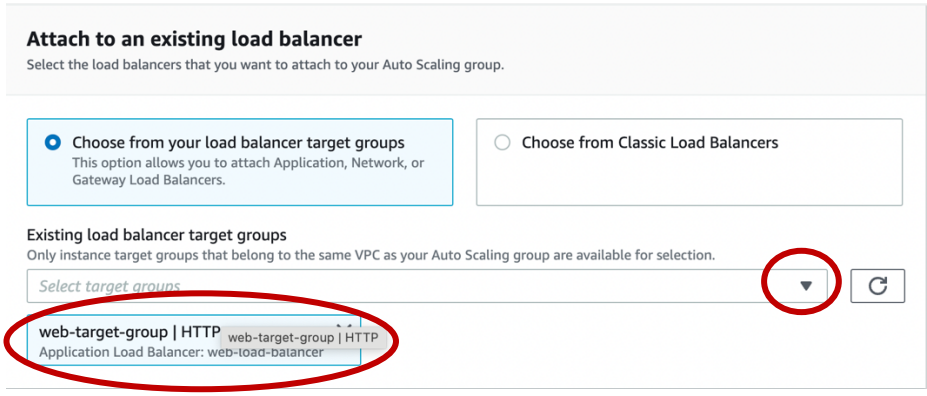
31) Na tela “Choose instance launch options” selecione todas as zonas de disponibilidade no campo “Availability Zones and subnets” e clique em “Next”:



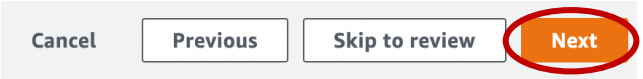
32) Na tela “Configure advanced options”, na seção “Load Balancing” clique em “Attach to an existing load balancer”:



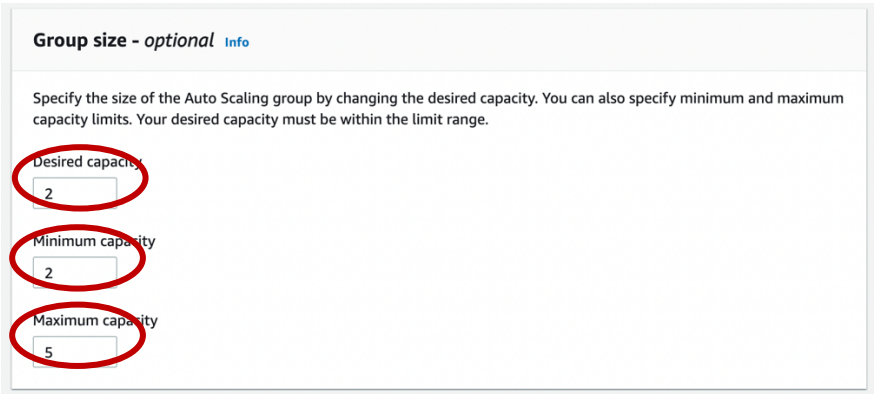
33) Na seção “Attach to an existing load balancer” escolha o load balancer group que acabamos de criar, o “web-target-group”:



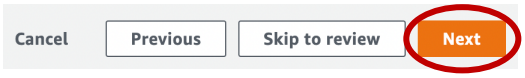
34) Clique em “Next”:



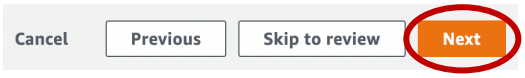
35) Na tela “Configure group size and scaling policies”, na seção “Group size” preencha os campos:
- Desired capacity: 2
- Minimum capacity: 2
- Maximum capacity: 5



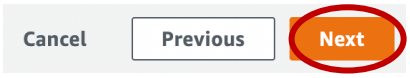
36) Clique em “Next”:



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



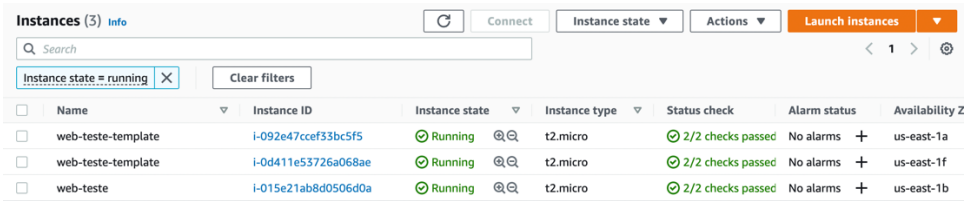
38) Na tela “Add tags” clique em “Next”:



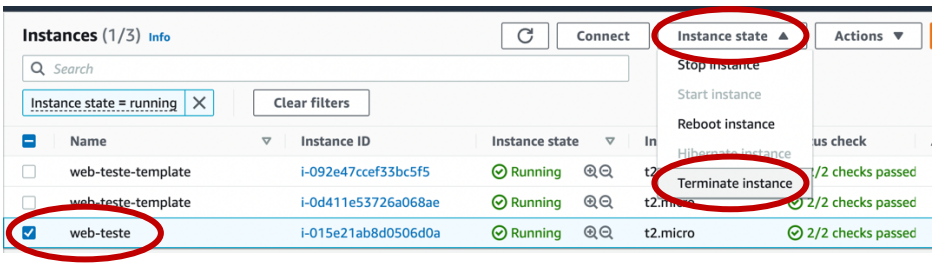
39) Na tela “Review” clique em “Create Auto Scaling group”:

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.

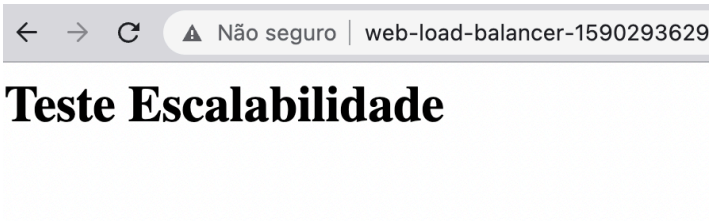
40) Verifique que agora temos 3 instâncias em execução no dashboard de instâncias do EC2, a instância que utilizamos como origem das imagens (web-teste) e mais duas instâncias do nosso Auto Scaling Group (web-teste-template).



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”:

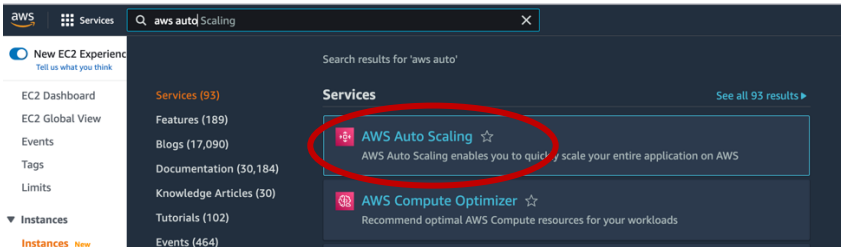


42) Veja que mesmo terminando a instância “web-teste”, ao invocar a URL do load balancer no navegador, o apache continua respondendo, isso porque ele está distribuído a requisições entre as duas instâncias do Auto Scale Group”:

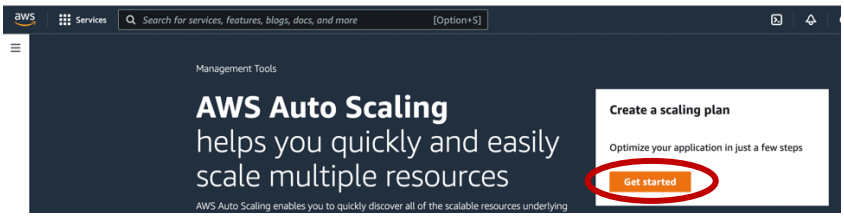


Chegou a hora de configurarmos o “Auto Scaling Plan”, para escalar a quantidade de instância dependendo da carga de processamento, para isso:

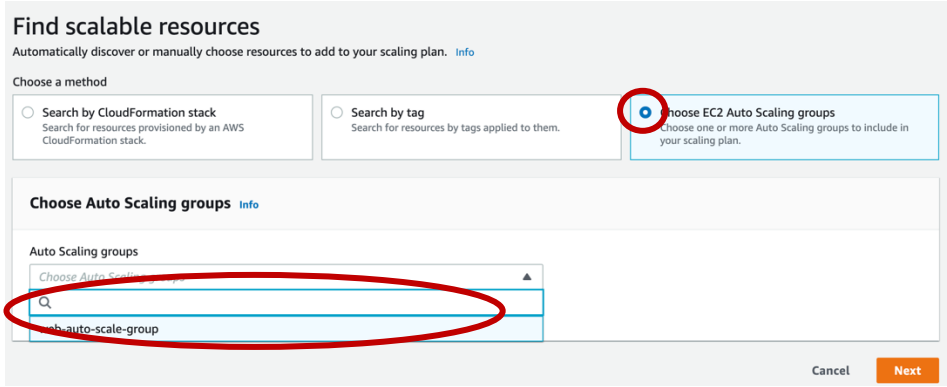
43) Em “Services”, procure pelo serviço chamado “AWS Auto Scaling” e clique em “AWS Auto Scaling”:



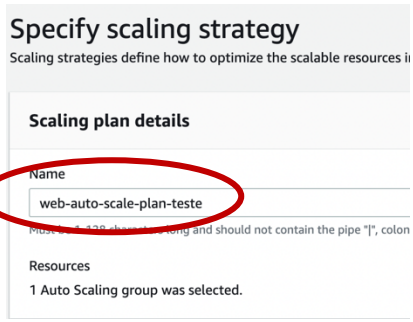
44) Na console do “AWS Auto Scaling” clique em “Get Started”:



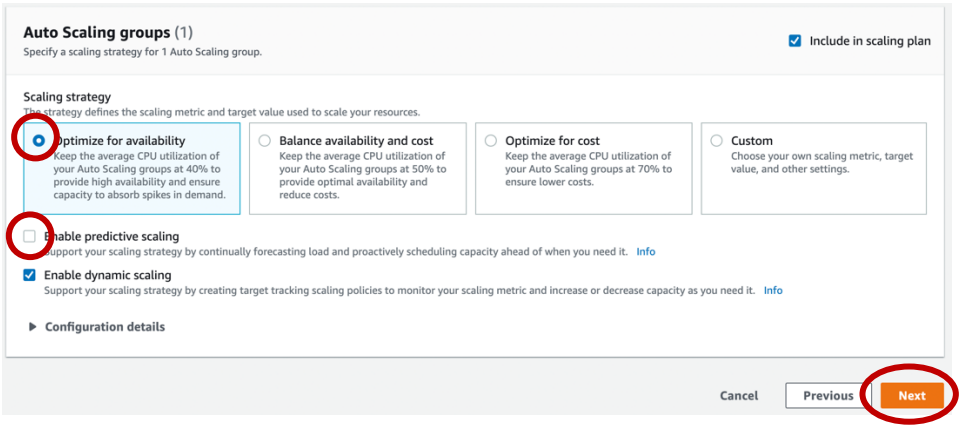
45) Na tela “Find scalable resources”, clique na opção “Choose EC2 Auto Scaling groups” e no campo “Auto Scaling groups” escolha o grupo que acabamos de criar, com o nome de “web-auto-scale-group” e clique em “Next”:



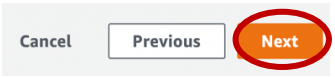
46) Na tela “Specify scaling strategy”, na seção “Scaling plan detail”, preencha o campo “Name” com “web-auto-scale-plan-teste”:



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



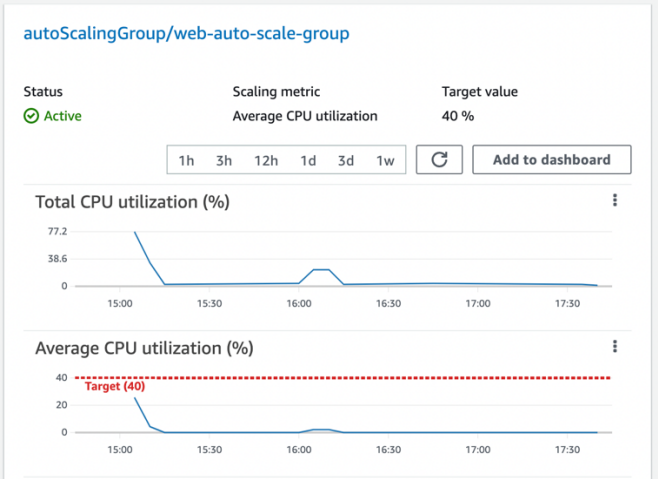
48) Na tela “Configure advanced settings (optional)”clique em “Next”:



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

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

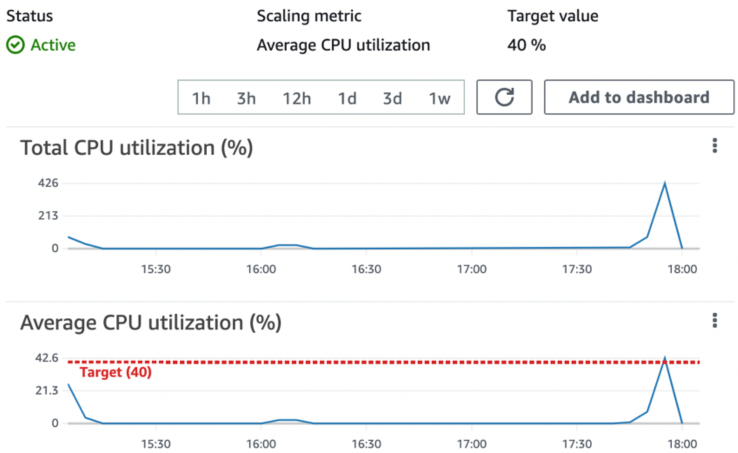
```
CPU[                                      0.0%]   Tasks: 38, 72 thr; 1 running
Mem[||||||||||||||||||||||||||||||||      101M/966M]   Load average: 0.05 0.07 0.03
Swp[                                      0K/0K]   Uptime: 02:47:42
```

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

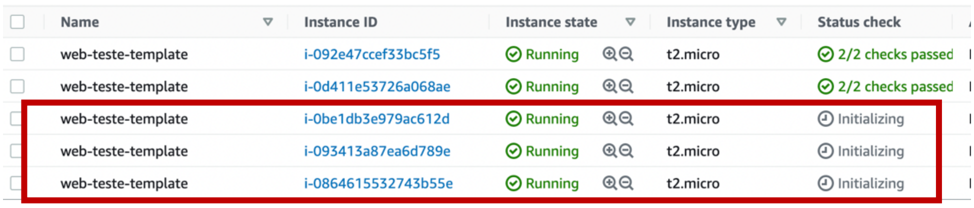
```
[ec2-user@ip-172-31-15-245 ~]$ stress --cpu 1
stress: info: [4746] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

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

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%.



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

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.