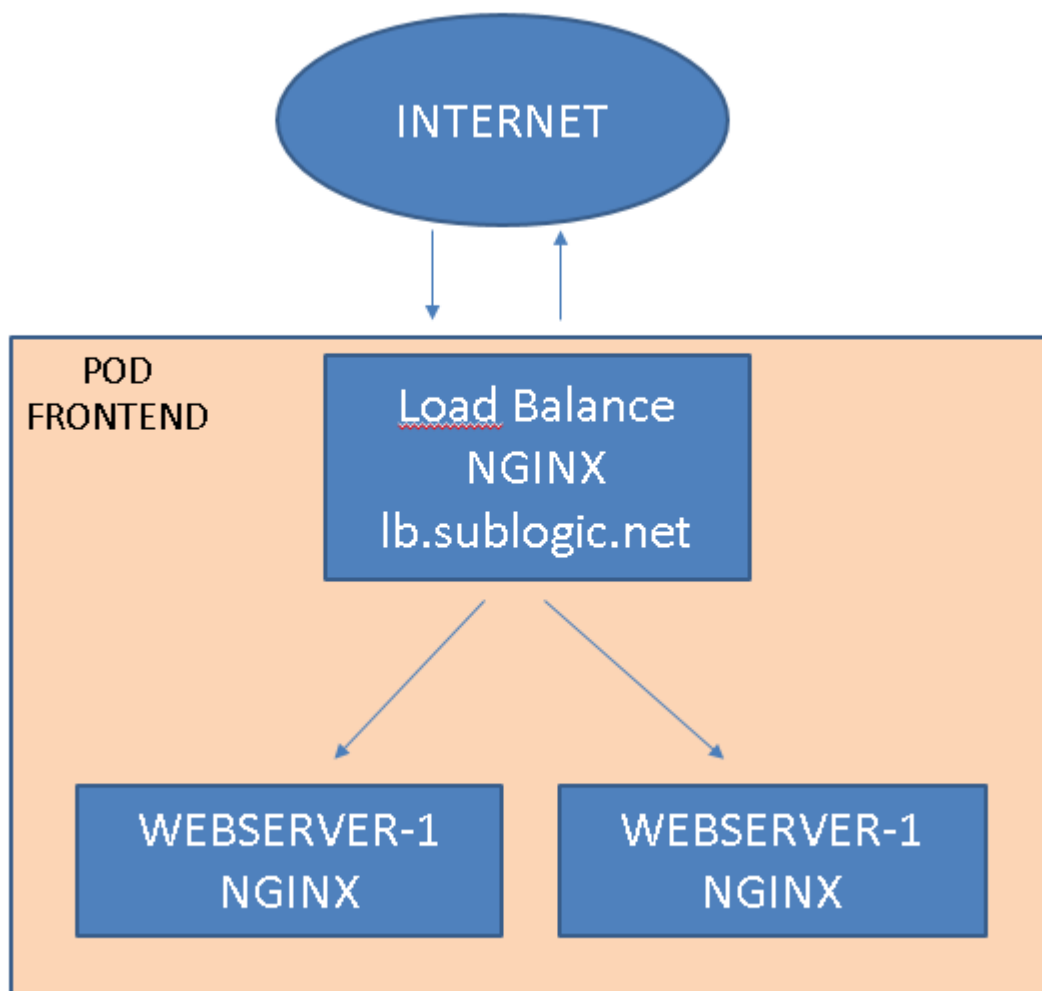


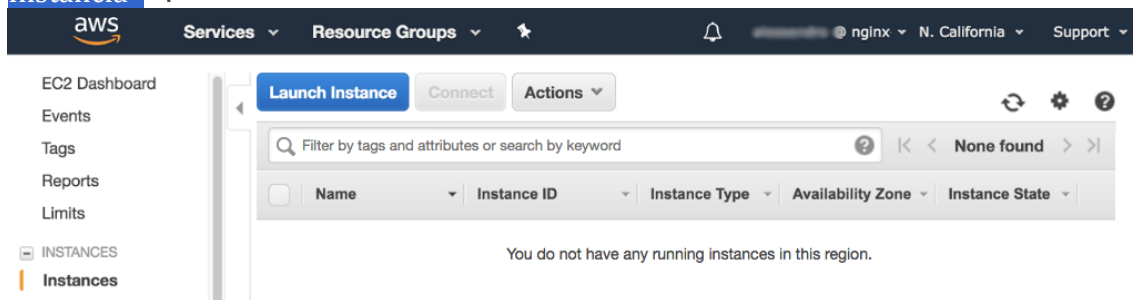
## How To – Criação de Load Balance com AMI AWS + NGINX

### Topologia

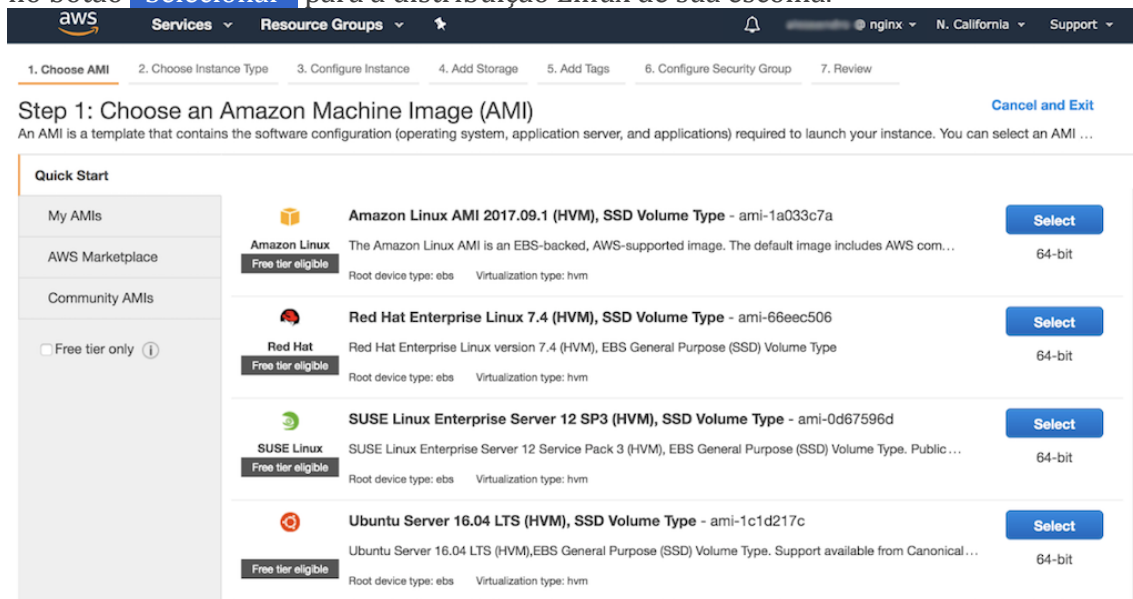


# Criação de uma instância Amazon EC2

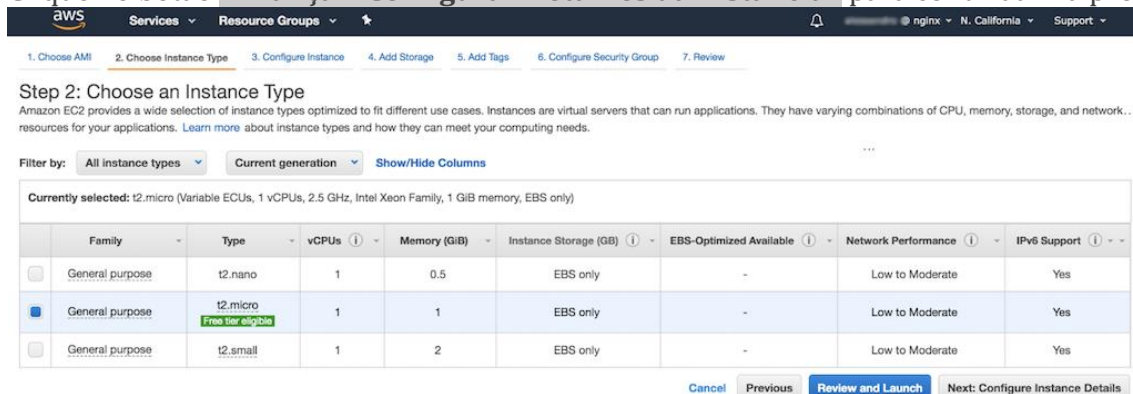
1. Faça login no **painel** do **EC2** no AWS Management Console
2. Na barra de navegação à esquerda, selecione **Instâncias** e clique no botão **Iniciar Instância**.



3. Na janela **Etapa 1: Escolha uma Amazon Machine Image (AMI)**, clique no botão **Selecionar** para a distribuição Linux de sua escolha.



4. Na janela **Etapa 2: Escolha um Tipo de Instância**, clique no botão de opção do tipo de instância apropriado. Na captura de tela, estamos selecionando uma instância **t2.micro**, que normalmente é selecionada por padrão e é suficiente para fins de demonstração. Clique no botão **Avançar: Configurar Detalhes da Instância** para continuar na próxima etapa.



5. Na **Etapa 3: janela Configurar detalhes da instância**, selecione a sub-rede padrão para seu VPC no campo **Sub - rede**
6. Aqui também já escolha criar 3 instâncias
7. clique no botão **Próximo: Adicionar armazenamento**.

**Step 3: Configure Instance Details**  
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the ...

**Number of instances** ⓘ  [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ  [Create new VPC](#)

**Subnet** ⓘ  [Create new subnet](#)  
4087 IP Addresses available

**Auto-assign Public IP** ⓘ

**IAM role** ⓘ  [Create new IAM role](#)

**Shutdown behavior** ⓘ

**Enable termination protection** ⓘ ☐ Protect against accidental termination

**Monitoring** ⓘ ☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

**Tenancy** ⓘ  [Additional charges will apply for dedicated tenancy.](#)

**T2 Unlimited** ⓘ ☐ Enable  
[Additional charges may apply](#)

► **Advanced Details**

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

8. Na **Etapa 4: janela adicionar armazenamento**, deixe os padrões inalterados. Clique no botão **Avançar: Adicionar tags**.

**Step 4: Add Storage**  
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0b2b8096f1b89e969	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

9. Na **Etapa 5: janela Adicionar tags**, clique no botão **Adicionar tag**. Digite o **Nome** no campo **Chave** e no campo **Valor** digite o nome da instância (a captura de tela mostra o resultado). Esse nome é o que aparecerá na coluna **Nome** da tabela de resumo na guia **Instâncias** do painel do EC2 (veja a captura de tela na Etapa 12, que mostra uma instância). Se você estiver seguindo essas instruções conforme orientado por um guia de implantação NGINX, a seção **Criação de instâncias EC2 e instalação do software NGINX** do guia de implantação especifica os nomes de instância a serem usados. Clique no botão **Avançar: Configurar Grupo de Segurança** para prosseguir para a próxima etapa.

**aws** Services Resource Groups nginx N. California Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	instance-name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

10. Na **Etapa 6: janela Configurar Grupo de Segurança**, selecione ou insira os seguintes valores nos campos indicados:

- **Atribuir um grupo de segurança** -
  - Se você estiver configurando uma implantação com várias instâncias (uma em um guia de implantação NGINX, por exemplo) e esta for a primeira instância que está criando, selecione **Criar um novo grupo de segurança**.
  - Para as instâncias subsequentes, selecione **Selecionar um grupo de segurança existente** (faz sentido que todas as instâncias em uma implantação usem o mesmo grupo de segurança).
- **Nome do grupo de segurança** - Nome do grupo. Se você estiver seguindo essas instruções conforme orientado por um guia de implantação NGINX, a seção **Pré - requisitos e configuração necessária da AWS** do guia de implantação especifica o nome do grupo a ser usado.
- **Descrição** - Descrição do grupo; o nome do grupo é frequentemente usado.

**aws** Services Resource Groups nginx N. California Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your ... the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

**Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access ...

[Cancel](#) [Previous](#) [Review and Launch](#)

11. Na tabela, modifique a regra padrão para conexões SSH, se necessário, selecionando ou definindo os seguintes valores. Eles permitem conexões SSH de entrada de todas as fontes (qualquer endereço IP):

- **Tipo** - SSH
- **Protocolo** - TCP
- **Portas** - 22
- **Fonte** - Custom 0.0.0.0/0
- **Descrição** - Aceita conexões SSH de todas as fontes

12. Crie uma regra que permita conexões HTTP/HTTPS de entrada de todas as fontes, clicando no botão **Adicionar regra** e selecionando ou definindo os seguintes valores na nova linha:

- **Tipo - HTTP**
- **Protocolo - TCP**
- **Portas - 80**
- **Fonte - Custom 0.0.0.0/0**
- **Descrição - Aceita conexões HTTP não criptografadas de todas as fontes**

- **Tipo - HTTPS**
- **Protocolo - TCP**
- **Portas - 443**
- **Fonte - Custom 0.0.0.0/0**
- **Descrição - Aceita conexões HTTP não criptografadas de todas as fontes**

Se apropriado, repita esta etapa para criar uma regra para o tráfego HTTPS.

Depois de criar todas as regras desejadas, clique no botão **Revisar e iniciar**.

13. Na **Etapa 7: janela Revisar inicialização da instância**, verifique se as configurações estão corretas. Nesse caso, clique no botão **Iniciar** no canto inferior direito da janela. Para alterar as configurações, clique no botão **Anterior** para voltar às janelas anteriores.

**Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instances' security. Your security group, *security-group-name*, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ **AMI Details** [Edit AMI](#)

**Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-1a033c7a**

**Free tier eligible** The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The ...

Root Device Type: ebs Virtualization type: hvm

▼ **Instance Type** [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only		Low to Moderate

▼ **Security Groups** [Edit security groups](#)

Security group name: *security-group-name*  
Description: *security group-description*

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	

●  
●  
●

[Cancel](#) [Previous](#) [Launch](#)

14. Quando você clica no botão **Iniciar**, uma janela aparece solicitando que você selecione um par de chaves existente ou crie um novo par de chaves. Execute a ação apropriada para o seu caso de uso e clique no botão **Iniciar Instâncias**.

**Nota:** É uma prática recomendada - e essencial em um ambiente de produção - criar uma chave separada para cada instância EC2, de modo que, se uma chave for comprometida, apenas a única instância associada ficará vulnerável.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

☒ I acknowledge that I have access to the selected private key file (`.pem`), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

O próximo passo é instalar o NGINX nas 3 instâncias e desativar o firewall do Linux:

```
yum install -y nginx
service nginx start
chkconfig nginx on
```

```
service iptables stop
chkconfig iptables off
```

No NGINX os arquivos de configuração principais são:

/etc/nginx/nginx.conf => Configurações globais e tuning

/usr/share/nginx/html => Conteúdo a ser servido (html, imagens, vídeos, etc)

Altere o HOSTNAME de cada servidor para refletir um NOME + DOMÍNIO adequado. Isso é o que chamamos de FQDN – Full Qualified Domain Name. Para alterar o HOSTNAME edite o arquivo /etc/sysconfig/network

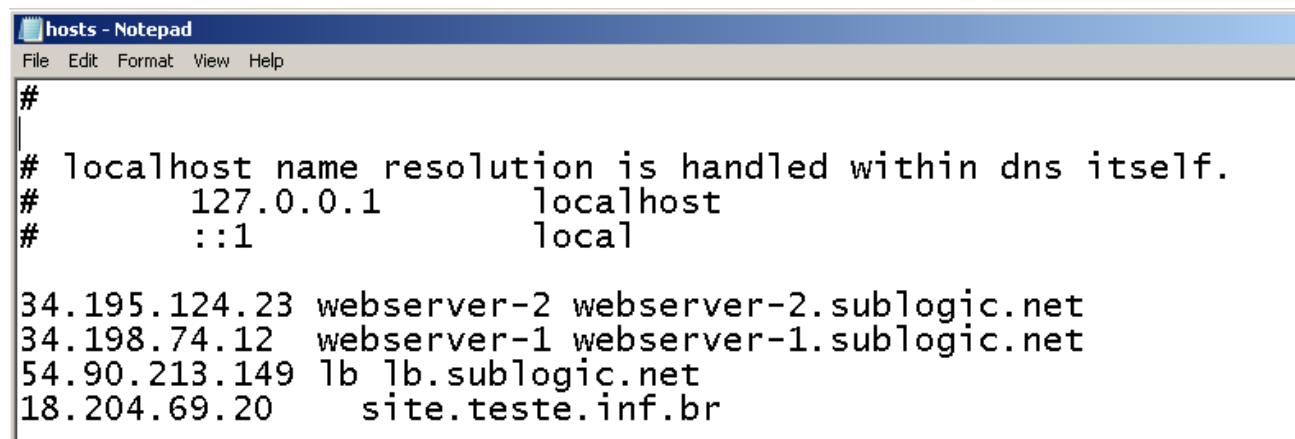
```
[root@lb conf.d]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=lb.sublogic.net
```

Altere também o arquivo /etc/hosts das 3 instâncias de modo que cada uma consiga resolver o HOSTNAME (FQDN) da outra. No meu caso ficou assim:

```
[root@lb conf.d]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

34.195.124.23 webserver-2 webserver-2.sublogic.net
34.198.74.12  webserver-1 webserver-1.sublogic.net
54.90.213.149 lb lb.sublogic.net
[root@lb conf.d]#
```

Para que você consiga acessar de seu navegador os NGINX, você pode usar o IP Externo de cada um ou configurar o arquivo HOSTS de sua máquina Windows (ou Linux/Mac) apontando para o IP/HOSTNAME lb.sublogic.net (no meu caso). Ficaria assim:



```
hosts - Notepad
File Edit Format View Help
#
# localhost name resolution is handled within dns itself.
#      127.0.0.1          localhost
#      ::1               local

34.195.124.23 webserver-2 webserver-2.sublogic.net
34.198.74.12  webserver-1 webserver-1.sublogic.net
54.90.213.149 lb lb.sublogic.net
18.204.69.20  site.teste.inf.br
```

Para uma apresentação mais profissional você pode utilizar um serviço gratuito de DNS Dinâmico como DynDNS, NO-IP, etc. No meu caso criei 2 HOSTS com registro do tipo A apontando para os IP's de 2 instâncias Linux na AWS. Ficou assim:



Nome de Host	Last Update	IP / Alvo	Type
lb.serveblog.net <small>Vencerá em 39 dias</small>	Oct 14, 2020 19:03 PDT	54.90.213.149	A
sublogic.hopto.org <small>Vencerá em 27 dias</small>	Oct 13, 2020 05:27 PDT	54.197.167.150	A

No caso, a instância NGINX que fará o balanceamento (lb.suglogic.net) ficou com registro de DNS o tipo A (IPv4) **lb.serveblog.net** apontando para o IP Externo **54.90.213.149**. Esse será o VIP (Virtual IP) de nosso balanceador.

Depois, configure uma página **index.html** em cada um dos NGINX que serão os WEB Servers (webserver-1 e webserver-2) de forma a identificar cada um dos Web Servers. Assim ao se chamar a URL, via navegador, de cada Web Server saberemos que é quem.

Se você configurou o IP Externo de cada instância em seu arquivo HOSTS local, vai conseguir abrir a URL em seu navegador. Caso contrário, utilize o IP Externo de cada instância. No meu caso, seria assim:

## WEB SERVER 1

<http://webserver-1.sublogic.net> ou <https://34.195.124.23>



## WEB SERVER 2

<http://webserver-2.sublogic.net> ou <https://34.198.74.12>





O próximo passo é configurar a instância NGINX que fará o balanceamento de carga, no meu caso a instância **lb.sublogic.net**. Para isso acesse a console da instância e crie um arquivo de nome **loadbalancer.conf** no diretório **/etc/nginx/conf.d** com o seguinte conteúdo:

```
server {  
listen 80;  
server_name lb.sublogic.net;  
location / {  
index index.html index.htm;  
proxy_pass http://balance;  
proxy_set_header X-Real-IP $remote_addr;  
}}
```

```
upstream balance {  
server webserver-1.sublogic.net;  
server webserver-2.sublogic.net;  
}
```

Não esqueça de alterar os nomes das instâncias de acordo com o que vocês criaram. Se criaram os nomes iguais aos meus, não precisa alterar, porém, os IP's Externos das instâncias de vocês serão, Salve o arquivo e reinicie o NGINX nesta instância:

**service nginx stop && service nginx start**

```
[root@lb conf.d]# service nginx stop && service nginx start  
Stopping nginx: [ OK ]  
Starting nginx: [ OK ]  
[root@lb conf.d]#
```

Feito isso você já pode testar se o balanceamento funcionado utilizando o comando CURL e fazendo uma chamada para a URL do seu balanceador e observando se retorna o código HTTP 200 OK (indicando que o web server respondeu) e o campo ETAG que é o registro serial diferente para cada WEB SERVER. No meu caso, como o nome do meu balanceador é lb.sublogic.net tive o retorno abaixo:

```
[root@lb conf.d]# curl -I http://lb.sublogic.net  
HTTP/1.1 200 OK  
Server: nginx/1.18.0  
Date: Thu, 15 Oct 2020 23:40:06 GMT  
Content-Type: text/html  
Content-Length: 2303  
Connection: keep-alive  
Last-Modified: Thu, 15 Oct 2020 14:42:34 GMT  
ETag: "5f885fda-8ff"  
Accept-Ranges: bytes  
  
[root@lb conf.d]# curl -I http://lb.sublogic.net  
HTTP/1.1 200 OK  
Server: nginx/1.18.0  
Date: Thu, 15 Oct 2020 23:40:07 GMT  
Content-Type: text/html  
Content-Length: 2301  
Connection: keep-alive  
Last-Modified: Thu, 15 Oct 2020 14:44:33 GMT  
ETag: "5f886051-8fd"  
Accept-Ranges: bytes
```

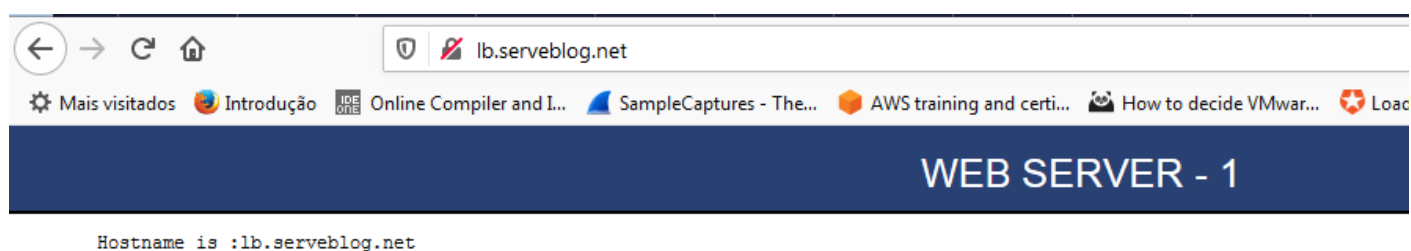
Observe que o valor do campo ETag muda de acordo com o WEB SERVER que responde. Se quiser ter certeza, chame, via CURL a URL de cada um dos WEB SERVER's para validar. No meu caso, ficou assim:

```
[root@lb ~]# curl -I http://webserver-1.sublogic.net
HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Fri, 16 Oct 2020 19:16:14 GMT
Content-Type: text/html
Content-Length: 2301
Last-Modified: Thu, 15 Oct 2020 14:44:33 GMT
Connection: keep-alive
ETag: "5f886051-8fd"
Accept-Ranges: bytes

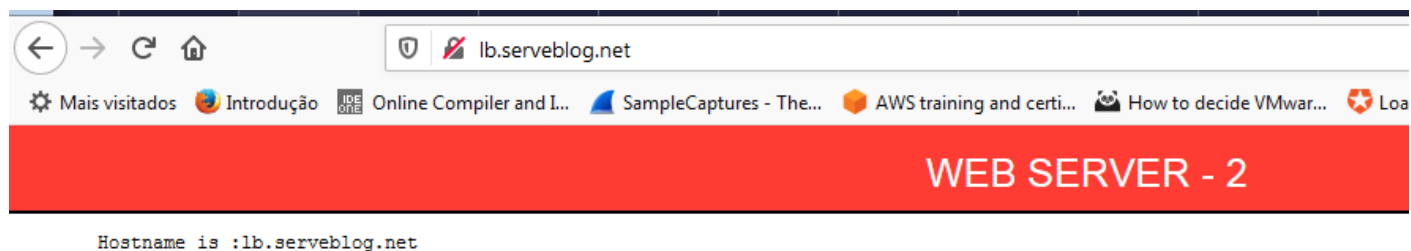
[root@lb ~]# curl -I http://webserver-2.sublogic.net
HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Fri, 16 Oct 2020 19:16:22 GMT
Content-Type: text/html
Content-Length: 2303
Last-Modified: Thu, 15 Oct 2020 14:42:34 GMT
Connection: keep-alive
ETag: "5f885fda-8ff"
Accept-Ranges: bytes
```

Se o nome DNS foi registrado num serviço de DNS dinâmico, também já é possível chara a URL do balanceador via navegador. No meu caso, ficou assim:

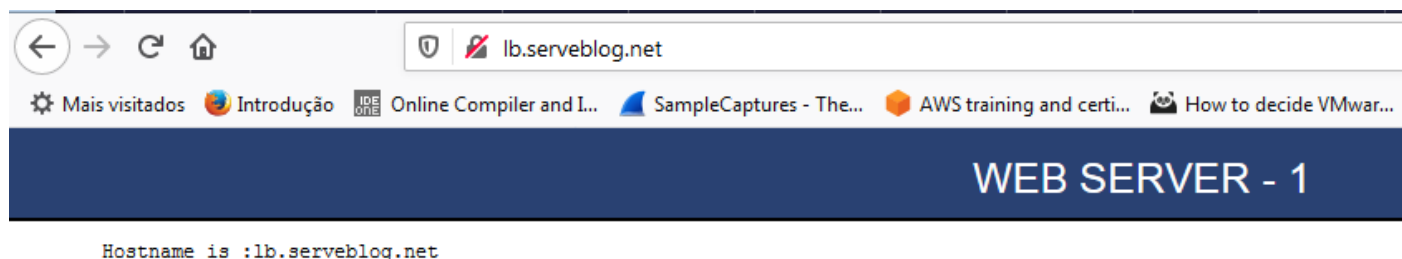
Primeira Chamada



Segunda Chamada



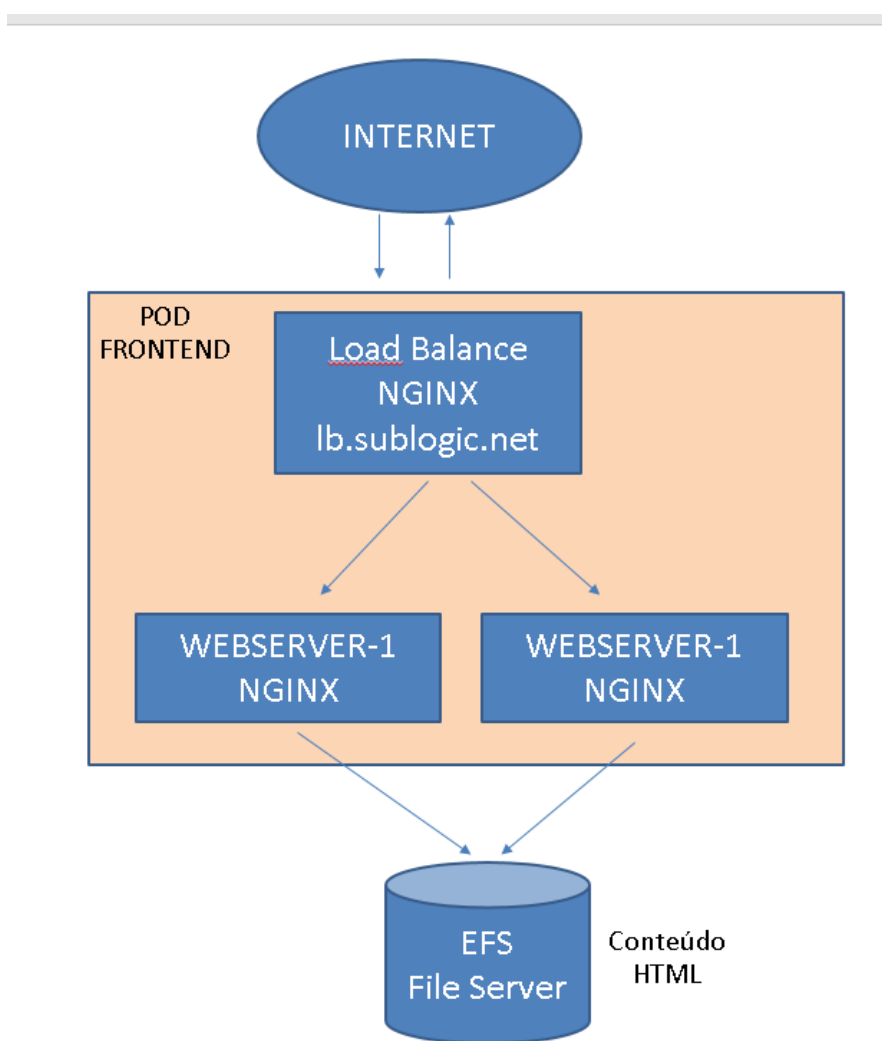
## Terceira Chamada



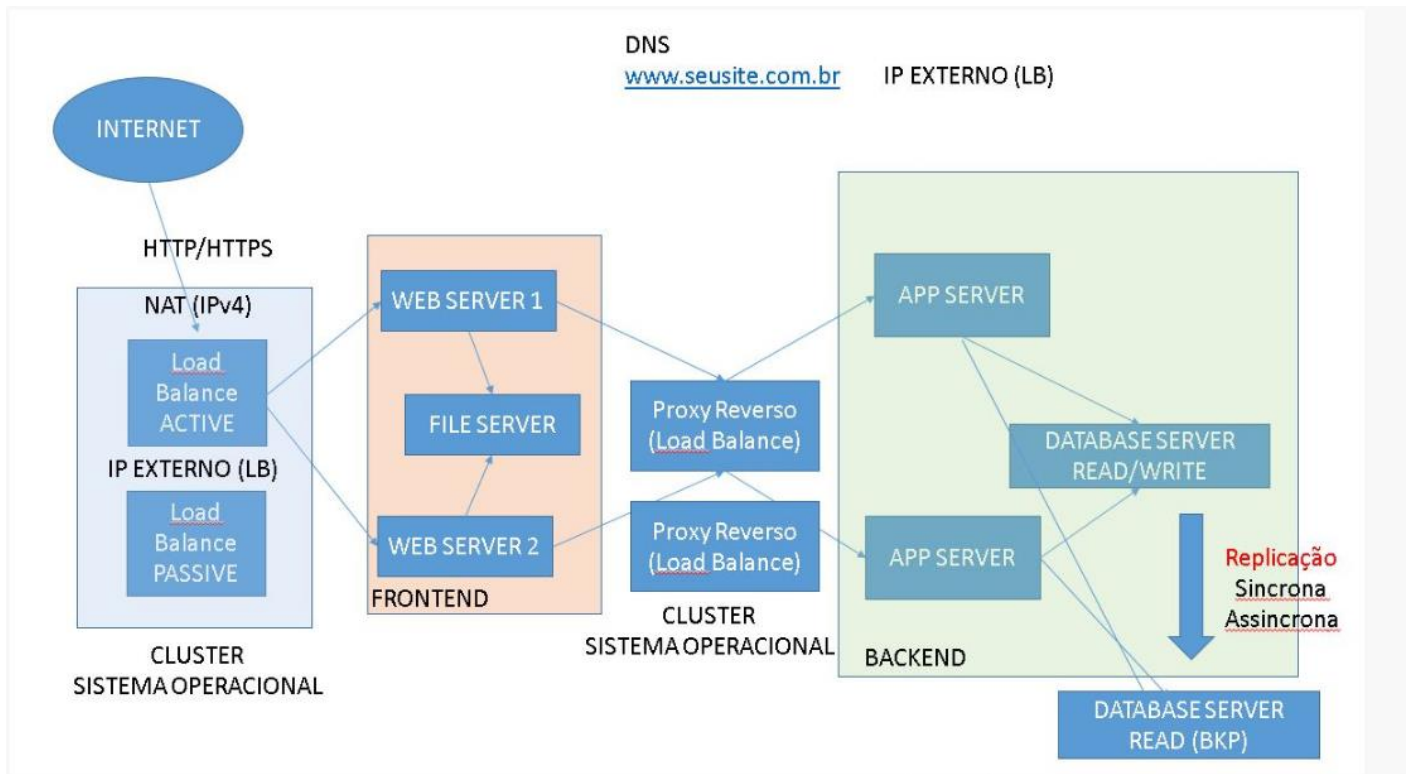
E assim sucessivamente porque o NGINX da instância de balanceamento foi configurado com o algoritmo de ROUND ROBIN aonde cada requisição é feita de forma sequencial para cada WEB SERVER registrado na seção de Proxy Reverso do arquivo loadbalancer.conf

Porém, observe que o conteúdo HTML estará isolado em cada um dos WEB SERVER's, sendo que qualquer alteração de conteúdo tem de ser realizada em todos. Para eliminar esta limitação, podemos montar um disco externo simulando um File Server e apresentar este disco para as instâncias de WEB SERVER e assim colocar o conteúdo HTML neste disco de forma que todos os WEB SERVER irão puxar o conteúdo do mesmo local.

Assim teríamos a seguinte topologia (Topologia Simples):



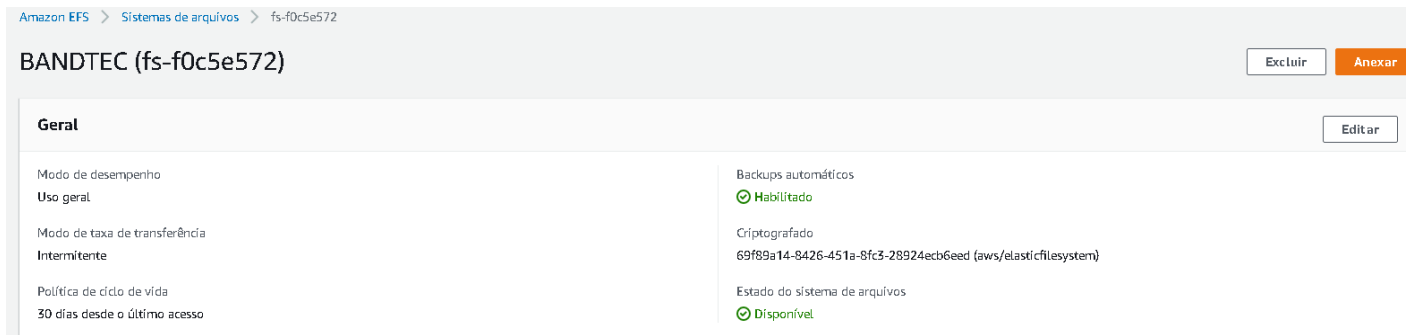
## Topologia Complexa



Para isso podemos utilizar o recurso de EFS (Elastic File System) da AWS. Na console da AWS acesse a seção EFS e crie um sistema de arquivos. No meu caso criei um recurso chamado BANDTEC



Ao clicar em BANDTEC temos a tela abaixo onde vamos em ANEXAR:



## Em ANEXAR



Temos ai a informação de como montar esse recurso de disco na máquina Linux utilizando o protocolo NFSv4

Vamos usar o comando:

```
sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-  
f0c5e572.efs.us-east-1.amazonaws.com:/ /BANDTEC
```

Os itens em vermelhos devem ser substituídos de acordo com os nomes que forem escolhidos.

Porém, esse é um recurso PAGO que somente irá funcionar dentro da AWS. Uma outra forma, tradicional, que produz o mesmo efeito seria subir uma outra máquina virtual (VM) e configurá-la como um Servidor de Arquivos (File Server). No caso, vamos utilizar o protocolo NFS que é o protocolo nativo para compartilhamento de arquivos em máquina com Unix/Linux. Utilizando a imagem modelos de um dos 3 servidores que já foram criados, suba mais uma VM e instale os serviços de NFS:

```
yum install nfs* -y
```

Nesse caso, serão instalados os serviços: **rpcbind**; **nfs**; **nfs-server**; **nfs-lock** e **nfs-idmap**

Além disso serão instalados o pacote **nfs-utils** que é a parte CLIENT para permitir a conexão de outras máquinas no Servidor de Arquivos. Esse pacote já existe por padrão na imagem AMI de Linux que estamos utilizando

Após instalar o serviços, inicie todos e configure para iniciar automaticamente no reboot com os comandos:

```
systemctl start nfs nfs-server rpcbind nfs-lock nfs-idmap  
systemctl enable nfs nfs-server rpcbind nfs-lock nfs-idmap
```

Além disso, devem ser liberadas as portas/protocolos de transporte dos serviços de NFS no Security Group para que a comunicação seja possível entre as máquinas. Para ver a lista de portas/protocolos de transporte, utilize o comando **rpcinfo -p**

```
[root@fs-1 ~]# rpcinfo -p  
program vers proto  port  service  
100000    4    tcp    111   portmapper  
100000    3    tcp    111   portmapper  
100000    2    tcp    111   portmapper  
100000    4    udp    111   portmapper  
100000    3    udp    111   portmapper  
100000    2    udp    111   portmapper  
100005    1    udp    20048 mountd  
100005    1    tcp    20048 mountd  
100005    2    udp    20048 mountd  
100005    2    tcp    20048 mountd  
100005    3    udp    20048 mountd  
100005    3    tcp    20048 mountd  
100024    1    udp    60329 status  
100024    1    tcp    37505 status  
100003    3    tcp    2049  nfs  
100003    4    tcp    2049  nfs  
100227    3    tcp    2049  nfs_acl  
100003    3    udp    2049  nfs  
100227    3    udp    2049  nfs_acl  
100021    1    udp    42009 nlockmgr  
100021    3    udp    42009 nlockmgr  
100021    4    udp    42009 nlockmgr  
100021    1    tcp    34329 nlockmgr  
100021    3    tcp    34329 nlockmgr  
100021    4    tcp    34329 nlockmgr  
[root@fs-1 ~]#
```

Segue a lista resumida de portas que devem ser liberadas no Security Group. Lembre-se de que todas as máquinas devem estar no mesmo Security Group:

```
111/tcp
54302/tcp
20048/tcp
2049/tcp
46666/tcp
42955/tcp
875/tcp
39279/tcp
35257/udp
37795/tcp
```

E também TODOS UDP (1-65535)

Feito isso, crie um diretório qualquer que servirá como repositório dentro do Servidor de Arquivos e deixe com permissão 777

```
mkdir /FILESERVER
chmod 777 /FILESERVER
```

Feito isso, vamos editar o arquivo `/etc/exports` e incluir esse diretório e os IP's das máquinas que poderão acessá-lo bem como algumas opções de permissionamento. Nesse arquivo incluir os IP's Internos e Externos dos servidores de Load Balance, Web Server 1 e Web Server 2. No meu caso ficou assim:

```
cat /etc/exports
```

```
[root@fs-1 ~]# cat /etc/exports
/FILESERVER 54.205.227.26/24(rw, sync, no_root_squash, insecure)
/FILESERVER 172.31.0.0/16(rw, sync, no_root_squash, insecure)
/FILESERVER 34.192.205.254/24(rw, sync, no_root_squash, insecure)
/FILESERVER 54.159.129.26/24(rw, sync, no_root_squash, insecure)
[root@fs-1 ~]#
```

Pelo fato dos IP's Internos dos servidores estarem na mesma rede iniciada por 172.31, inseri todo o range /16 para facilitar

feito isso executar o comando para ler esse arquivo e exportar o diretório /FILESERVER criado para toda a rede:

```
exportfs -rav
```

Para testar se o diretório foi exportado, execute o comando `showmount` passando o nome do File Server como abaixo, no meu caso que o Servidor de Arquivos se chama `fs-1`:

```
showmount -e fs-1
```



E a resposta deverá ser parecida com a abaixo:

```
[root@fs-1 ~]# showmount -e fs-1
Export list for fs-1:
/FILESERVER 54.159.129.26/24,34.192.205.254/24,172.31.0.0/16,54.205.227.26/24
[root@fs-1 ~]#
```

Esse mesmo comando deve ser executado nas outras 3 máquinas: Load Balance, Web Server 1 e Web Server 2 e devem trazer o mesmo resultado!

# Procedimento para Gerar Certificado Digital – Ambiente DESE

## Instalar Certbot

```
$ curl -O https://dl.eff.org/certbot-auto
$ chmod +x certbot-auto
$ sudo mv certbot-auto /usr/local/bin/certbot-auto
```

## Instalar Nginx

```
$ sudo yum install nginx -y

(Nginx must be stopped during Certbot installation)
$ sudo service nginx stop
```

## Configurar um nome de fqdn (hostname+domínio)

Utilizar o serviço NO-IP.com

## Executar o Certbot na Instância AMI

```
(become a root user)
$ sudo su -

# certbot-auto certonly --standalone -d example.com --debug

(You'll be asked to enter your email address)

(Finally, you'll get a message like following)

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at
  /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will
  expire on 2016-mm-dd. To obtain a new version of the certificate in
  the future, simply run Certbot again.
- If you like Certbot, please consider supporting our work by:

    Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
    Donating to EFF:                   https://eff.org/donate-le
```

## Alterar Configuração do NGINX

Assuming that following commands are executed as root.

```
# cd /etc/nginx/
# cp nginx.conf nginx.conf.org

(Modify nginx.conf)
# vi nginx.conf

ssl_certificate "/etc/letsencrypt/live/example.com/fullchain.pem";
ssl_certificate_key "/etc/letsencrypt/live/example.com/privkey.pem";
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 10m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers
ssl_prefer_server_ciphers on;
```

Let's restart nginx after the change: `service nginx start`