

## 1. Create the Amazon EC2 instances:

Create five EC2 instances of the same time. Note that, when you create the instance, at the top of the window, click 6. Configure Security Group.



On the Step 6 page, click the “Add Rule” button below the table, then select “HTTP” in the Type column. We are allowing access to our instances from any IP address (the value in the Source column is “Anywhere”).

Then repeat last step for HTTPS.

## 2. Install Nginx on each instance:

1) Update system software:

```
$ sudo -s
```

```
# yum check-update
```

```
# yum update
```

```
root@ip-172-31-26-212:/home/ec2-user

Cleanup      : ntp-4.2.6p5-44.34.amzn1.x86_64      7/9
Cleanup      : ntpdate-4.2.6p5-44.34.amzn1.x86_64  8/9
Cleanup      : python27-crypto-2.6.1-1.14.amzn1.x86_64 9/9
Verifying    : ntpdate-4.2.6p5-44.36.amzn1.x86_64  1/9
Verifying    : python27-crypto-2.6.1-1.15.amzn1.x86_64 2/9
Verifying    : kernel-4.9.85-38.58.amzn1.x86_64    3/9
Verifying    : hwdata-0.233-14.1.19.amzn1.noarch   4/9
Verifying    : ntp-4.2.6p5-44.36.amzn1.x86_64     5/9
Verifying    : ntp-4.2.6p5-44.34.amzn1.x86_64     6/9
Verifying    : python27-crypto-2.6.1-1.14.amzn1.x86_64 7/9
Verifying    : hwdata-0.233-14.1.18.amzn1.noarch   8/9
Verifying    : ntpdate-4.2.6p5-44.34.amzn1.x86_64  9/9

Installed:
  kernel.x86_64 0:4.9.85-38.58.amzn1

Updated:
  hwdata.noarch 0:0.233-14.1.19.amzn1
  ntp.x86_64 0:4.2.6p5-44.36.amzn1
  ntpdate.x86_64 0:4.2.6p5-44.36.amzn1
  python27-crypto.x86_64 0:2.6.1-1.15.amzn1

Complete!
[root@ip-172-31-26-212 ec2-user]#
```

2) Install NGINX:

① Install dependent libraries:

```
# yum -y install pcre-devel zlib-devel openssl openssl-devel
```

```
root@ip-172-31-26-212:/home/ec2-user

Verifying : keyutils-libs-devel-1.5.8-3.12.amzn1.x86_64 3/10
Verifying : zlib-devel-1.2.8-7.18.amzn1.x86_64 4/10
Verifying : libcom_err-devel-1.42.12-4.40.amzn1.x86_64 5/10
Verifying : libsepol-devel-2.1.7-3.12.amzn1.x86_64 6/10
Verifying : libverto-devel-0.2.5-4.9.amzn1.x86_64 7/10
Verifying : krb5-devel-1.15.1-8.43.amzn1.x86_64 8/10
Verifying : pcre-devel-8.21-7.7.amzn1.x86_64 9/10
Verifying : libselinux-devel-2.1.10-3.22.amzn1.x86_64 10/10

Installed:
 openssl-devel.x86_64 1:1.0.2k-8.106.amzn1 pcre-devel.x86_64 0:8.21-7.7.amzn1
 zlib-devel.x86_64 0:1.2.8-7.18.amzn1

Dependency Installed:
 keyutils-libs-devel.x86_64 0:1.5.8-3.12.amzn1
 krb5-devel.x86_64 0:1.15.1-8.43.amzn1
 libcom_err-devel.x86_64 0:1.42.12-4.40.amzn1
 libkadm5.x86_64 0:1.15.1-8.43.amzn1
 libselinux-devel.x86_64 0:2.1.10-3.22.amzn1
 libsepol-devel.x86_64 0:2.1.7-3.12.amzn1
 libverto-devel.x86_64 0:0.2.5-4.9.amzn1

Complete!
[root@ip-172-31-26-212 ec2-user]#
```

② Install NGINX:

# yum install nginx

```
root@ip-172-31-26-212:/home/ec2-user

(1/3): gperftools-libs-2.0-11.5.amzn1.x86_64.rpm | 570 kB 00:00
(2/3): nginx-1.12.1-1.33.amzn1.x86_64.rpm | 561 kB 00:00
(3/3): libunwind-1.1-10.8.amzn1.x86_64.rpm | 72 kB 00:00
-----
Total 4.0 MB/s | 1.2 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Installing : libunwind-1.1-10.8.amzn1.x86_64 1/3
 Installing : gperftools-libs-2.0-11.5.amzn1.x86_64 2/3
 Installing : 1:nginx-1.12.1-1.33.amzn1.x86_64 3/3
 Verifying : libunwind-1.1-10.8.amzn1.x86_64 1/3
 Verifying : gperftools-libs-2.0-11.5.amzn1.x86_64 2/3
 Verifying : 1:nginx-1.12.1-1.33.amzn1.x86_64 3/3

Installed:
 nginx.x86_64 1:1.12.1-1.33.amzn1

Dependency Installed:
 gperftools-libs.x86_64 0:2.0-11.5.amzn1 libunwind.x86_64 0:1.1-10.8.amzn1

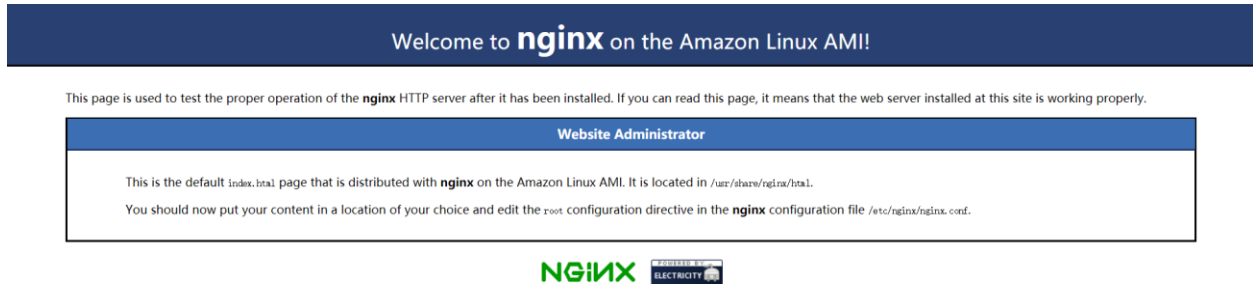
Complete!
[root@ip-172-31-26-212 ec2-user]#
```

Join boot automatically start:

```
# chkconfig --level 35 nginx on
```

```
# /etc/init.d/nginx start
```

③ Check status:



④ Replace the txt in the index.html file:

```
# cd /usr/share/nginx/html
```

```
# vi index.html
```

[SERVER\_4]

### 3) Configure the load balancer:

Use vi command to replace the text.

```
# vi /etc/nginx/nginx.conf
```

```
# /etc/init.d/nginx reload
```

```
# curl 18.188.25.244
```

```

[ec2-user@ip-172-31-20-78 ~]$ sudo /etc/init.d/nginx reload
Reloading nginx: [ OK ]
[ec2-user@ip-172-31-20-78 ~]$ curl 18.188.25.244
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body>
<h1>[SERVER_4]</h1>
</body>
</html>
[ec2-user@ip-172-31-20-78 ~]$ curl 18.188.25.244
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body>
<h1>[SERVER_3]</h1>
</body>
</html>
[ec2-user@ip-172-31-20-78 ~]$ curl 18.188.25.244
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body>
<h1>[SERVER_2]</h1>
</body>
</html>
[ec2-user@ip-172-31-20-78 ~]$ curl 18.188.25.244
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body>
<h1>[SERVER_4]</h1>
</body>
</html>
[ec2-user@ip-172-31-20-78 ~]$ curl 18.188.25.244
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body>
<h1>[SERVER_1]</h1>
</body>
</html>
[ec2-user@ip-172-31-20-78 ~]$ █

```

#### 4) Collect the information on visits to your site :

① weight = 1 : 1 : 1 : 1

```

ec2-user@ip-172-31-20-78:~
[ec2-user@ip-172-31-20-78 ~]$ sudo /etc/init.d/nginx reload
Reloading nginx: [ OK ]
[ec2-user@ip-172-31-20-78 ~]$ ./visit_server.rb -d 18.188.25.244
Starting to visit load balancing server

.....

Summary
-----
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
[ec2-user@ip-172-31-20-78 ~]$ █

```

② weight = 1 : 2 : 3 : 4

[illegible]

③ weight = 1 : 2 : 1 : 2

[illegible]