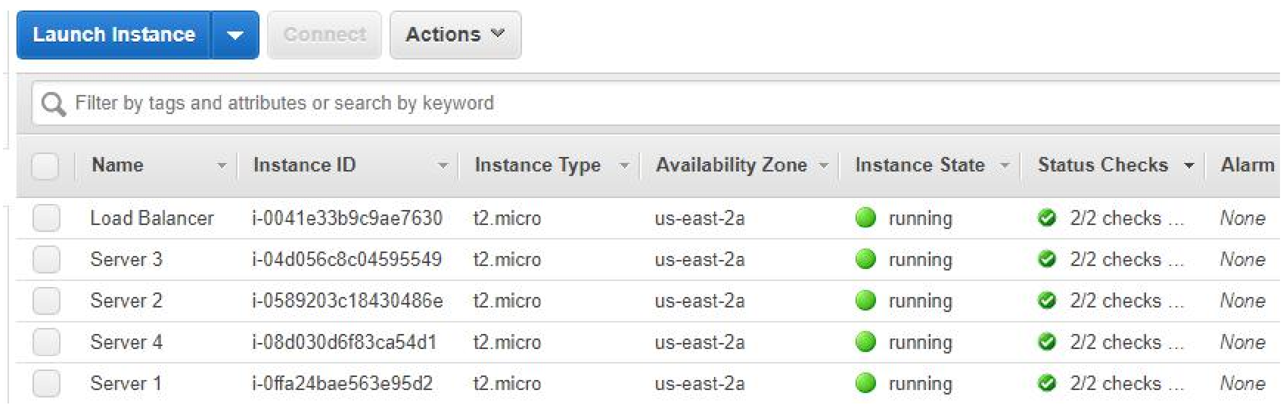
JichenDai_CS 524_Lab_#2

Create the Amazon EC2 instances

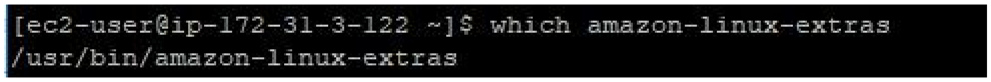
By follow the steps specified in Lab_#1, five instances is first created.



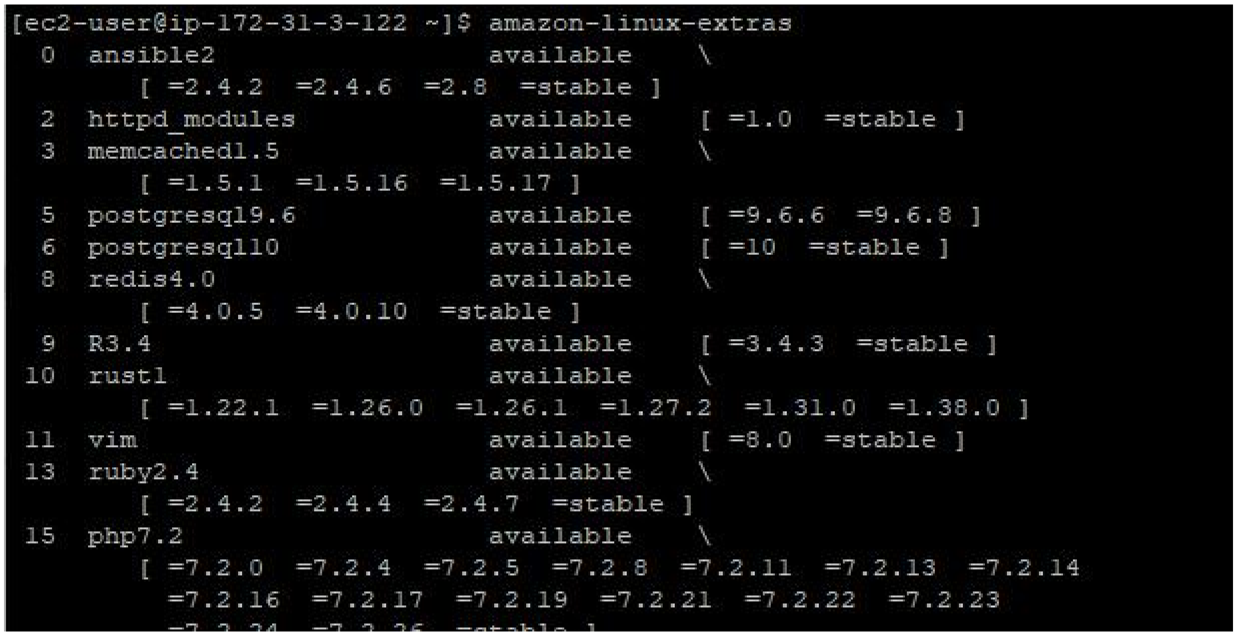
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
Load Balancer	i-0041e33b9c9ae7630	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	None
Server 3	i-04d056c8c04595549	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	None
Server 2	i-0589203c18430486e	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	None
Server 4	i-08d030d6f83ca54d1	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	None
Server 1	i-0ffa24bae563e95d2	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	None

Install *Nginx* on each instance

Use ***aamzon-linux-extras*** to get information about extras.

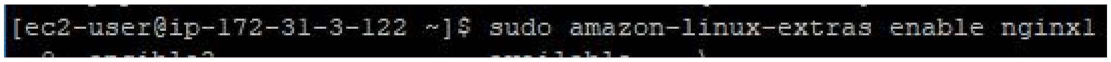


```
[ec2-user@ip-172-31-3-122 ~]$ which amazon-linux-extras
/usr/bin/amazon-linux-extras
```



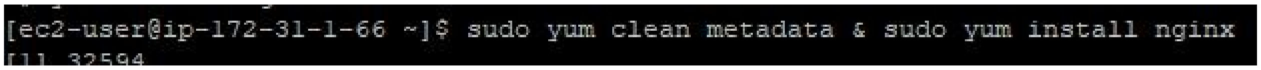
```
[ec2-user@ip-172-31-3-122 ~]$ amazon-linux-extras
  0  ansible2                 available    \
       [ =2.4.2  =2.4.6  =2.8  =stable ]
  2  httpd_modules            available    [ =1.0  =stable ]
  3  memcached1.5             available    \
       [ =1.5.1  =1.5.16  =1.5.17 ]
  5  postgresql9.6            available    [ =9.6.6  =9.6.8 ]
  6  postgresql10             available    [ =10  =stable ]
  8  redis4.0                 available    \
       [ =4.0.5  =4.0.10  =stable ]
  9  R3.4                     available    [ =3.4.3  =stable ]
 10  rust1                    available    \
       [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0 ]
 11  vim                      available    [ =8.0  =stable ]
 13  ruby2.4                  available    \
       [ =2.4.2  =2.4.4  =2.4.7  =stable ]
 15  php7.2                   available    \
       [ =7.2.0  =7.2.4  =7.2.5  =7.2.8  =7.2.11  =7.2.13  =7.2.14
         =7.2.16  =7.2.17  =7.2.19  =7.2.21  =7.2.22  =7.2.23
```

Enable nginx



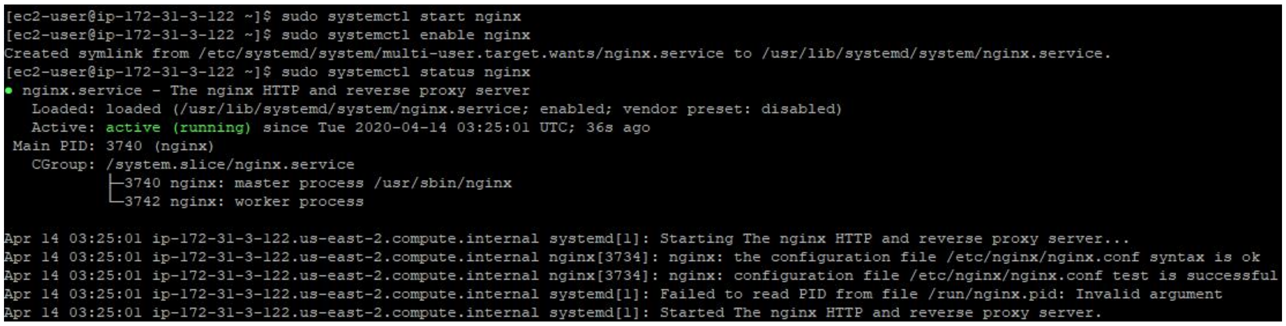
```
[ec2-user@ip-172-31-3-122 ~]$ sudo amazon-linux-extras enable nginx1
```

Remove matedate and install nginx.



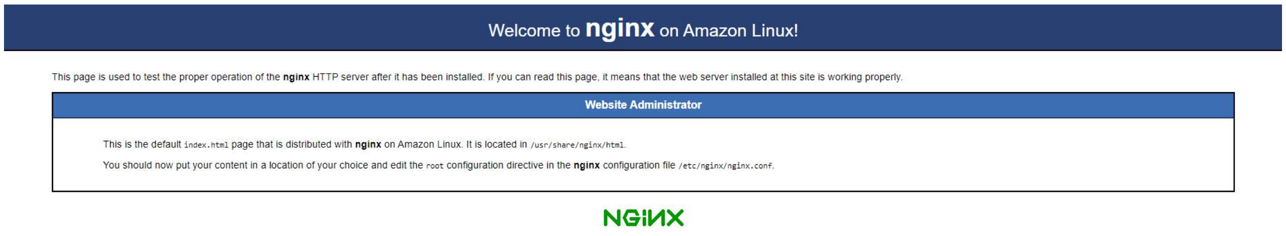
```
[ec2-user@ip-172-31-1-66 ~]$ sudo yum clean metadata & sudo yum install nginx
[1] 32594
```

Start and enable nginx using: ***sudo systemctl start nginx*** and ***sudo systemctl enable nginx.***



```
[ec2-user@ip-172-31-3-122 ~]$ sudo systemctl start nginx
[ec2-user@ip-172-31-3-122 ~]$ sudo systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-172-31-3-122 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-04-14 03:25:01 UTC; 36s ago
 Main PID: 3740 (nginx)
   CGroup: /system.slice/nginx.service
           ├─3740 nginx: master process /usr/sbin/nginx
           └─3742 nginx: worker process

Apr 14 03:25:01 ip-172-31-3-122.us-east-2.compute.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 14 03:25:01 ip-172-31-3-122.us-east-2.compute.internal nginx[3734]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 14 03:25:01 ip-172-31-3-122.us-east-2.compute.internal nginx[3734]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 14 03:25:01 ip-172-31-3-122.us-east-2.compute.internal systemd[1]: Failed to read PID from file /run/nginx.pid: Invalid argument
Apr 14 03:25:01 ip-172-31-3-122.us-east-2.compute.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
```

To verify the *Nginx* is working, visit the DNS in google chrome.



Welcome to **nginx** on Amazon Linux!

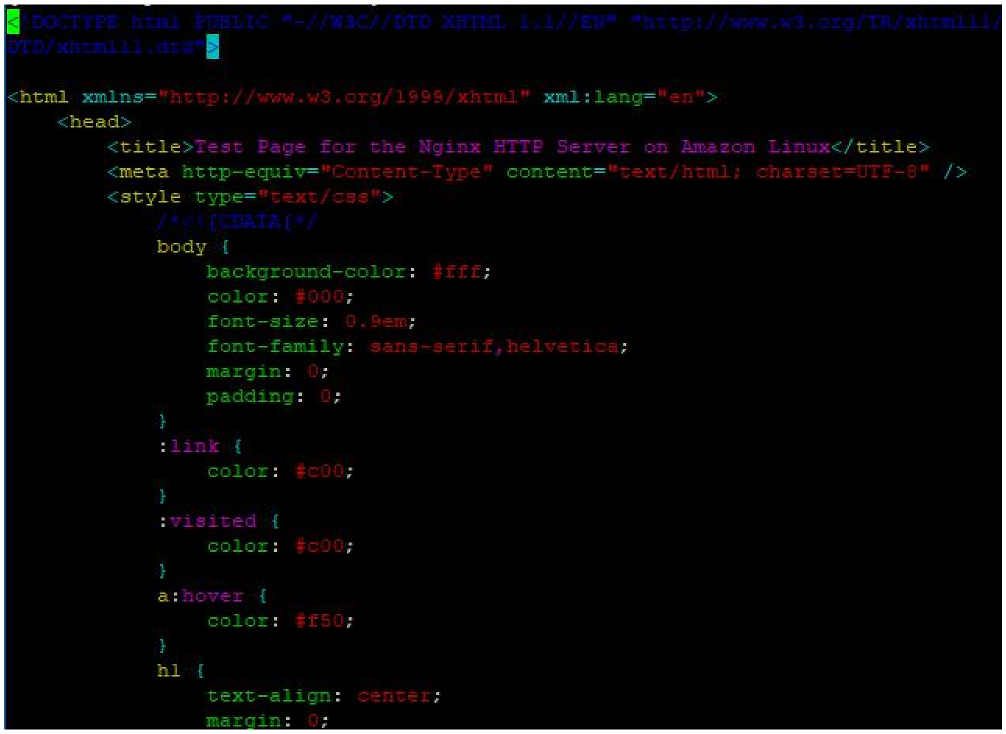
This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

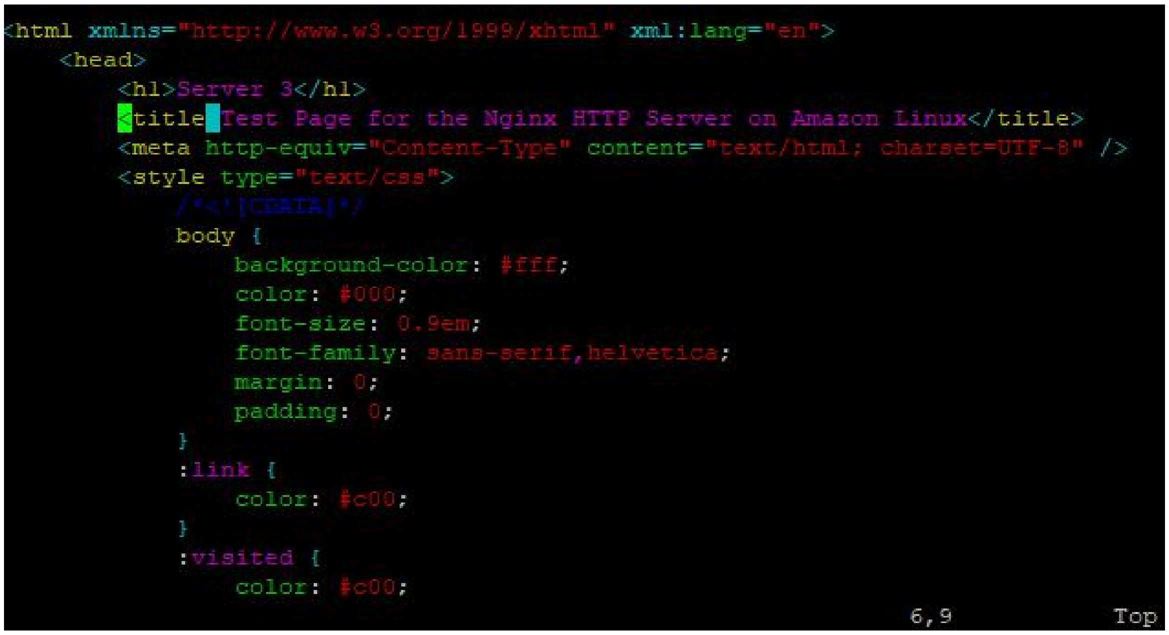
This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

Using **vim** to open the index.html to edit *index.html*. Add <h1>Server N</h1> to it.



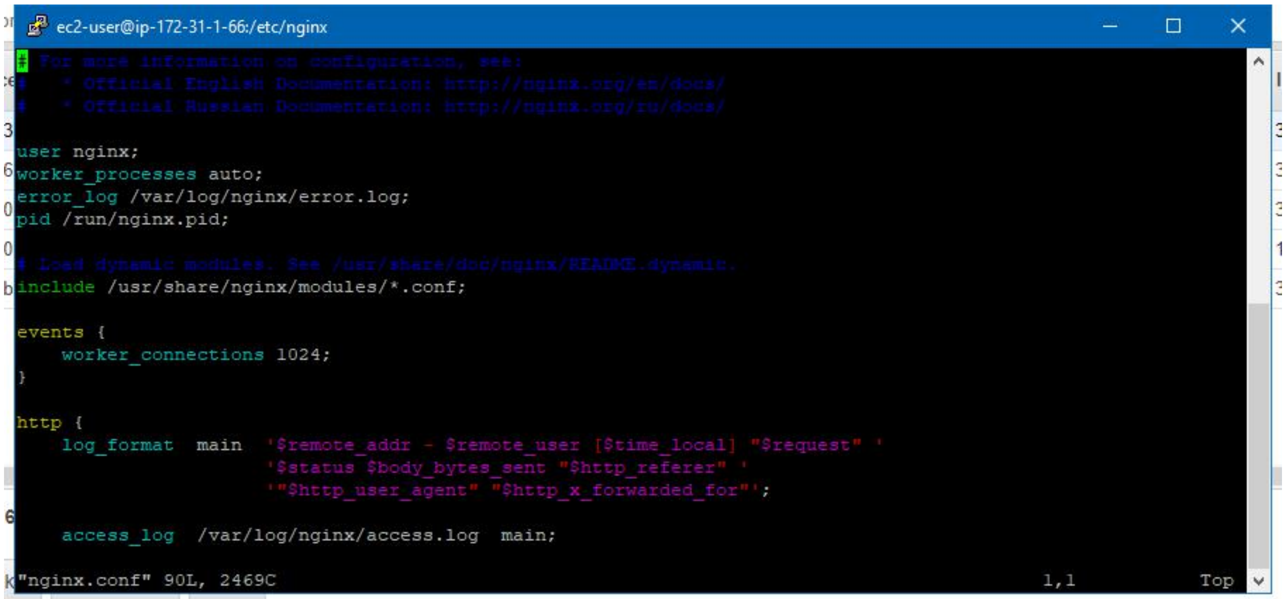
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <style type="text/css">
            /*<![CDATA[*/
            body {
                background-color: #fff;
                color: #000;
                font-size: 0.9em;
                font-family: sans-serif,helvetica;
                margin: 0;
                padding: 0;
            }
            :link {
                color: #c00;
            }
            :visited {
                color: #c00;
            }
            a:hover {
                color: #f50;
            }
            h1 {
                text-align: center;
                margin: 0;
```



```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <h1>Server 3</h1>
        <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <style type="text/css">
            /*<![CDATA[*/
            body {
                background-color: #fff;
                color: #000;
                font-size: 0.9em;
                font-family: sans-serif,helvetica;
                margin: 0;
                padding: 0;
            }
            :link {
                color: #c00;
            }
            :visited {
                color: #c00;
                                                              6,9           Top
```

Configure the load balancer

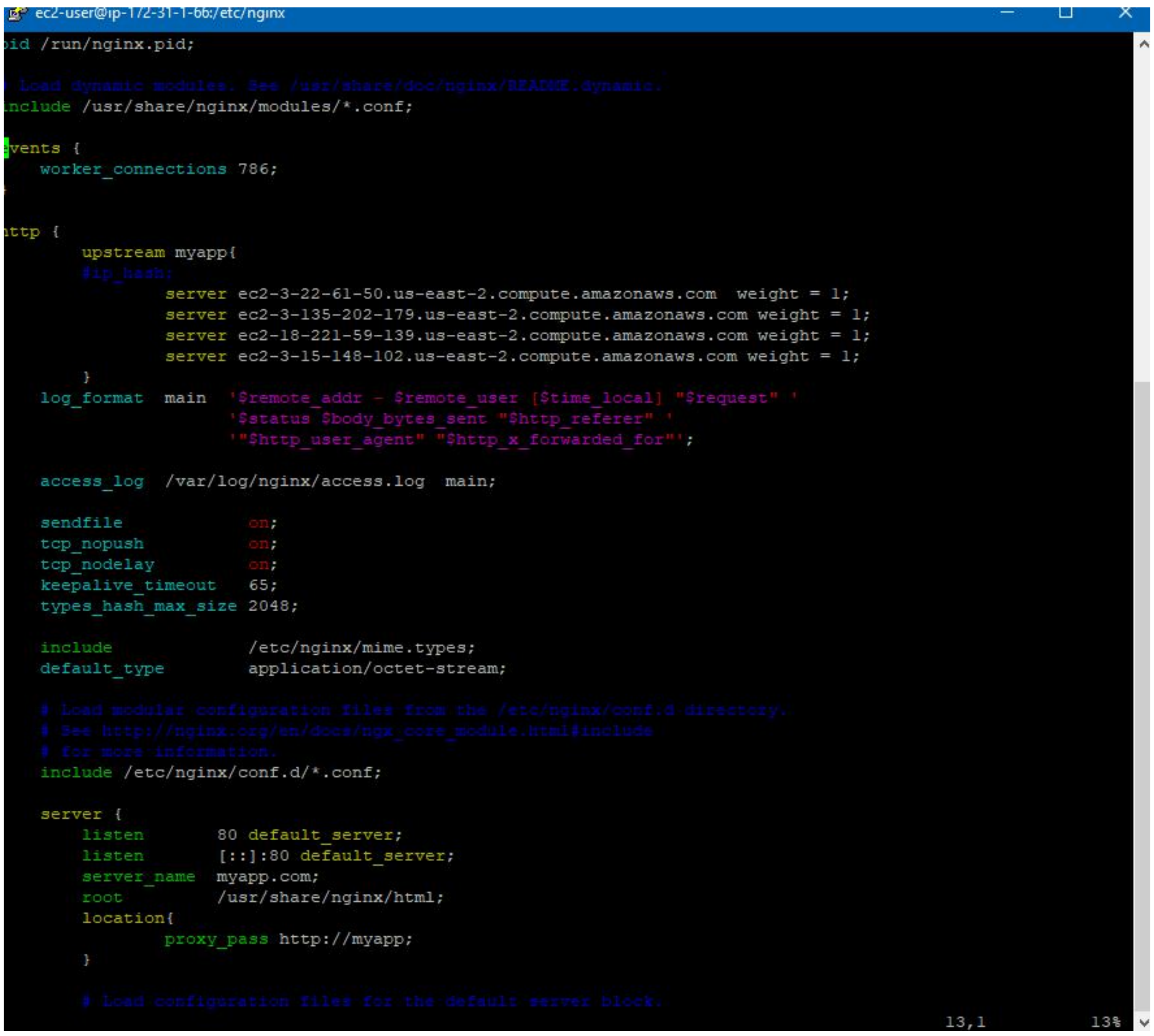
using vim to open *nginx.conf.*



```
ec2-user@ip-172-31-1-66:/etc/nginx
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

"nginx.conf" 90L, 2469C                                       1,1           Top
```

Replace and add the following content:

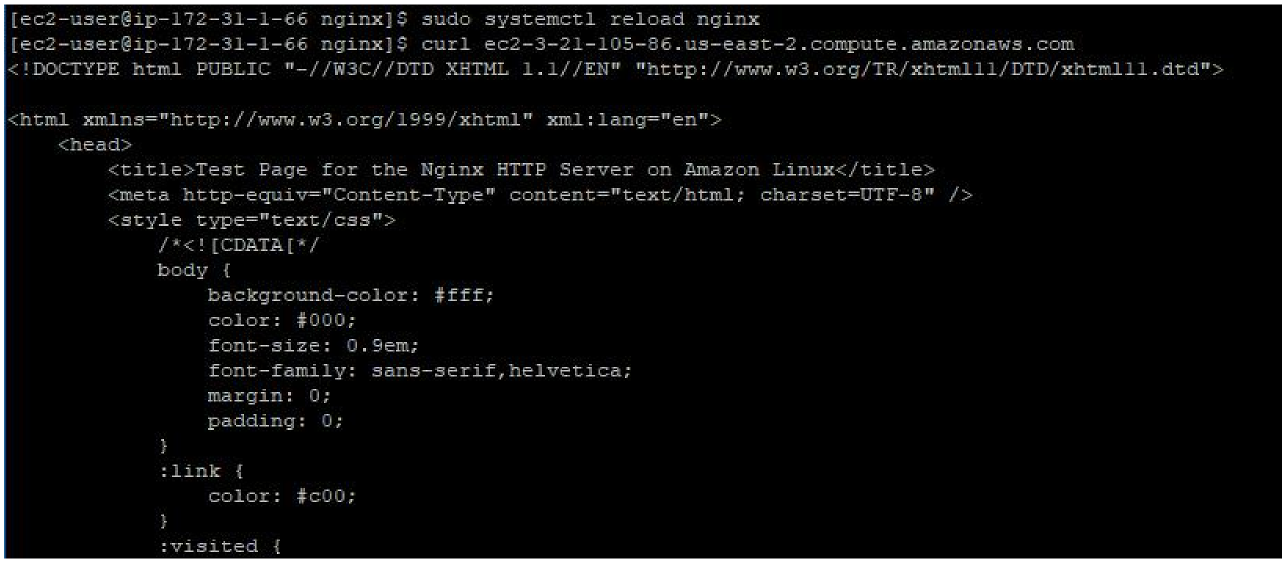
```
events {
 worker_connections 768;
}
http {
        upstream myapp {
                #ip_hash;
                server [SERVER_PUBLIC_DNS_NAME] weight=1;
                server [SERVER_PUBLIC_DNS_NAME] weight=1;
                server [SERVER_PUBLIC_DNS_NAME] weight=1;
                server [SERVER_PUBLIC_DNS_NAME] weight=1;
        }
        server {
                listen 80;
                server_name myapp.com;
                location / {
                        proxy_pass http://myapp;
                }
        }
}
```



```
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 786;
}

http {
        upstream myapp{
        #ip_hash;
                server ec2-3-22-61-50.us-east-2.compute.amazonaws.com  weight = 1;
                server ec2-3-135-202-179.us-east-2.compute.amazonaws.com weight = 1;
                server ec2-18-221-59-139.us-east-2.compute.amazonaws.com weight = 1;
                server ec2-3-15-148-102.us-east-2.compute.amazonaws.com weight = 1;
        }
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80 default_server;
        listen       [::]:80 default_server;
        server_name  myapp.com;
        root         /usr/share/nginx/html;
        location{
                proxy_pass http://myapp;
        }

        # Load configuration files for the default server block.
                                                                    13,1          13%
```

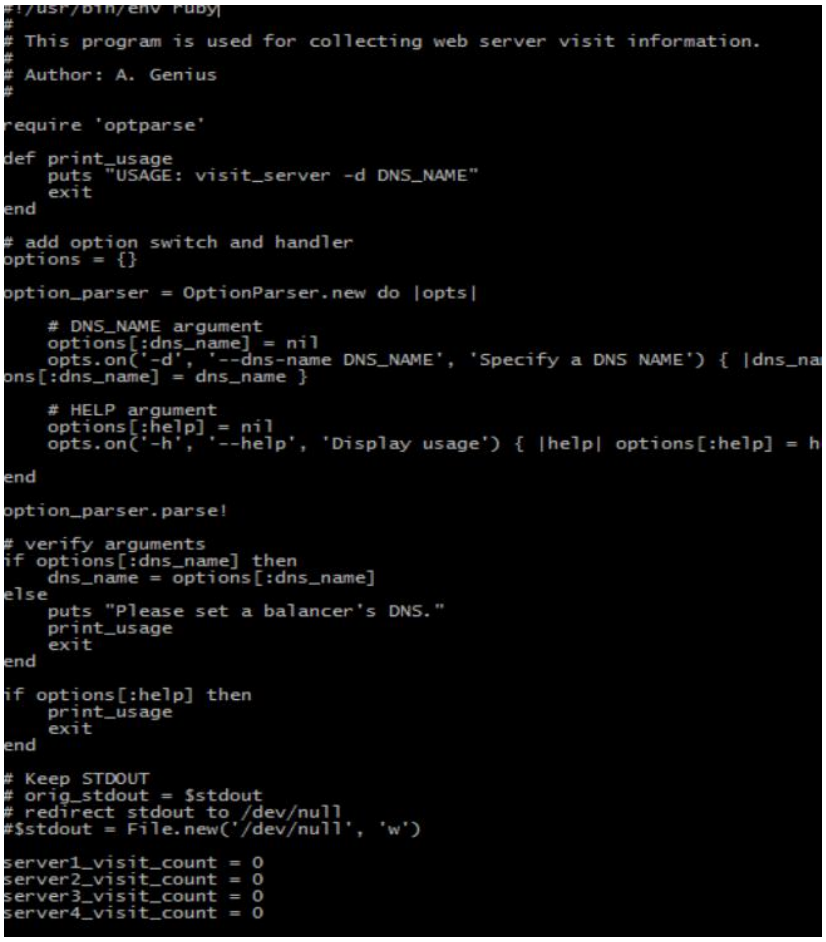
Reload nginx using: ***sudo systemctl reload nginx***.

Use the ***curl ec2-3-21-105-86.us-east-2.compute.amazonaws.com*** command visit the balancer.



```
[ec2-user@ip-172-31-1-66 nginx]$ sudo systemctl reload nginx
[ec2-user@ip-172-31-1-66 nginx]$ curl ec2-3-21-105-86.us-east-2.compute.amazonaws.com
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <style type="text/css">
            /*<![CDATA[*/
            body {
                background-color: #fff;
                color: #000;
                font-size: 0.9em;
                font-family: sans-serif,helvetica;
                margin: 0;
                padding: 0;
            }
            :link {
                color: #c00;
            }
            :visited {
```

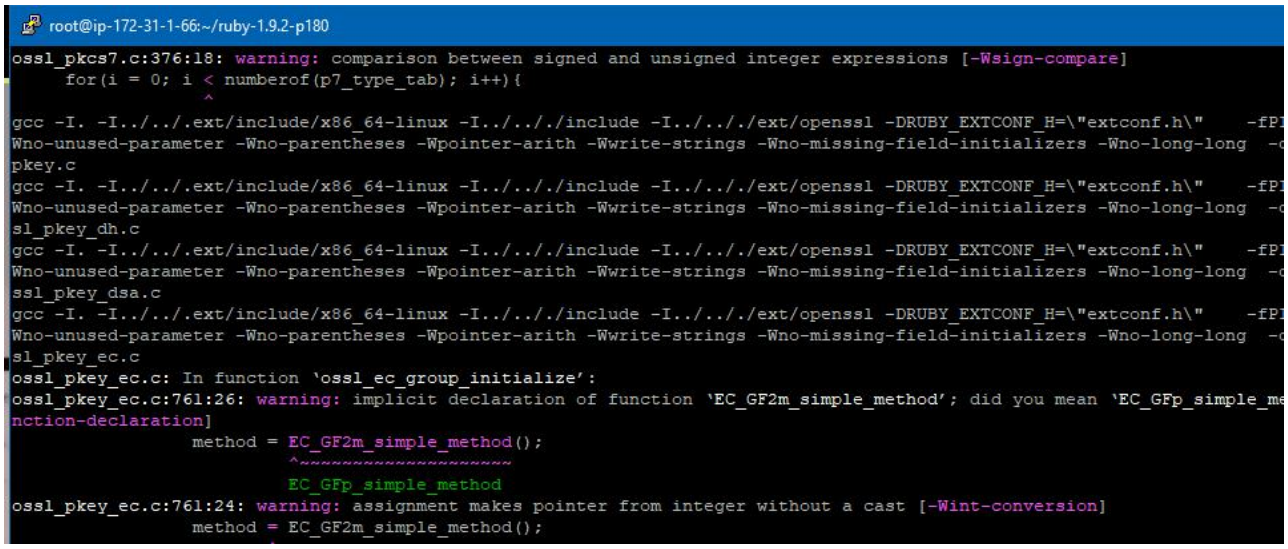
Edit visit_server suing: ***vim visit_server***



```
#!/usr/bin/env ruby
#
# This program is used for collecting web server visit information.
#
# Author: A. Genius
#

require 'optparse'

def print_usage
    puts "USAGE: visit_server -d DNS_NAME"
    exit
end

# add option switch and handler
options = {}

option_parser = OptionParser.new do |opts|

    # DNS_NAME argument
    options[:dns_name] = nil
    opts.on('-d', '--dns-name DNS_NAME', 'Specify a DNS NAME') { |dns_na
ons[:dns_name] = dns_name }

    # HELP argument
    options[:help] = nil
    opts.on('-h', '--help', 'Display usage') { |help| options[:help] = h

end

option_parser.parse!

# verify arguments
if options[:dns_name] then
    dns_name = options[:dns_name]
else
    puts "Please set a balancer's DNS."
    print_usage
    exit
end

if options[:help] then
    print_usage
    exit
end

# Keep STDOUT
# orig_stdout = $stdout
# redirect stdout to /dev/null
#$stdout = File.new('/dev/null', 'w')

server1_visit_count = 0
server2_visit_count = 0
server3_visit_count = 0
server4_visit_count = 0
```

Collect the information on visits to your site

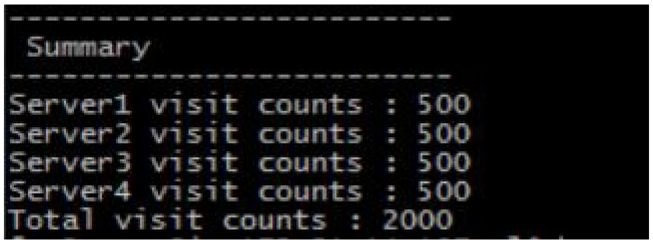
Install Ruby



root@ip-172-31-1-66:~/ruby-1.9.2-p180

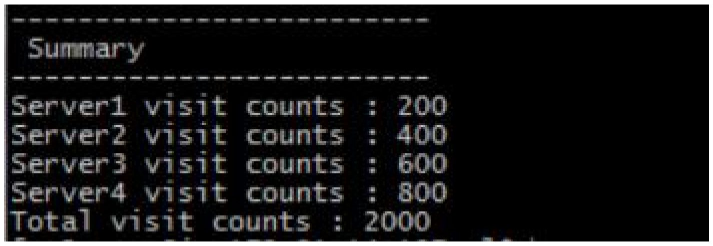
```
ossl_pkcs7.c:376:18: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
     for(i = 0; i < numberof(p7_type_tab); i++){
                  ^
gcc -I. -I../../.ext/include/x86_64-linux -I../.././include -I../.././ext/openssl -DRUBY_EXTCONF_H=\"extconf.h\"    -fPI
Wno-unused-parameter -Wno-parentheses -Wpointer-arith -Wwrite-strings -Wno-missing-field-initializers -Wno-long-long  -c
pkey.c
gcc -I. -I../../.ext/include/x86_64-linux -I../.././include -I../.././ext/openssl -DRUBY_EXTCONF_H=\"extconf.h\"    -fPI
Wno-unused-parameter -Wno-parentheses -Wpointer-arith -Wwrite-strings -Wno-missing-field-initializers -Wno-long-long  -c
sl_pkey_dh.c
gcc -I. -I../../.ext/include/x86_64-linux -I../.././include -I../.././ext/openssl -DRUBY_EXTCONF_H=\"extconf.h\"    -fPI
Wno-unused-parameter -Wno-parentheses -Wpointer-arith -Wwrite-strings -Wno-missing-field-initializers -Wno-long-long  -c
ssl_pkey_dsa.c
gcc -I. -I../../.ext/include/x86_64-linux -I../.././include -I../.././ext/openssl -DRUBY_EXTCONF_H=\"extconf.h\"    -fPI
Wno-unused-parameter -Wno-parentheses -Wpointer-arith -Wwrite-strings -Wno-missing-field-initializers -Wno-long-long  -c
sl_pkey_ec.c
ossl_pkey_ec.c: In function 'ossl_ec_group_initialize':
ossl_pkey_ec.c:761:26: warning: implicit declaration of function 'EC_GF2m_simple_method'; did you mean 'EC_GFp_simple_me
nction-declaration]
                 method = EC_GF2m_simple_method();
                          ^~~~~~~~~~~~~~~~~~~~~
                          EC_GFp_simple_method
ossl_pkey_ec.c:761:24: warning: assignment makes pointer from integer without a cast [-Wint-conversion]
                 method = EC_GF2m_simple_method();
```

Use **ruby visit_server -d ec2-172-31-1-66.us-east-2.compute.amazonaws.com** to see result of scenario 1.

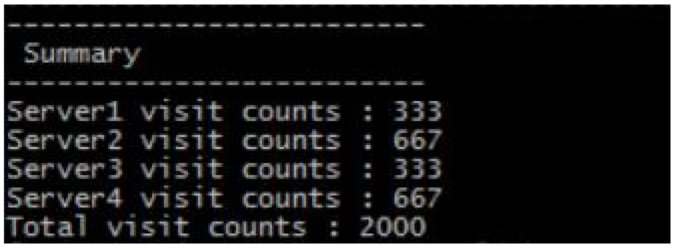


```
-------------------------
 Summary
-------------------------
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
```

Then, change the weight of four server into **1: 2: 3: 4** and **1: 2: 1: 2** in *nginx.conf*. Get different result:



```
-------------------------
 Summary
-------------------------
Server1 visit counts : 200
Server2 visit counts : 400
Server3 visit counts : 600
Server4 visit counts : 800
Total visit counts : 2000
```



```
-------------------------
 Summary
-------------------------
Server1 visit counts : 333
Server2 visit counts : 667
Server3 visit counts : 333
Server4 visit counts : 667
Total visit counts : 2000
```

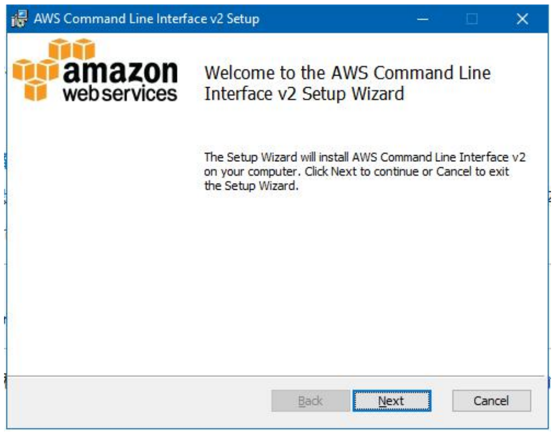
My observations

In my opinion, on the one hand, the technology of load balancing is very useful, since it allows us to handle a large bunch of job by dividing them into multi-servers. On the other hand, this technology also help us to implement failover: when one of the server shut down, other servers can take the responsibility.

Additional Steps:

Creating instance using Command Line

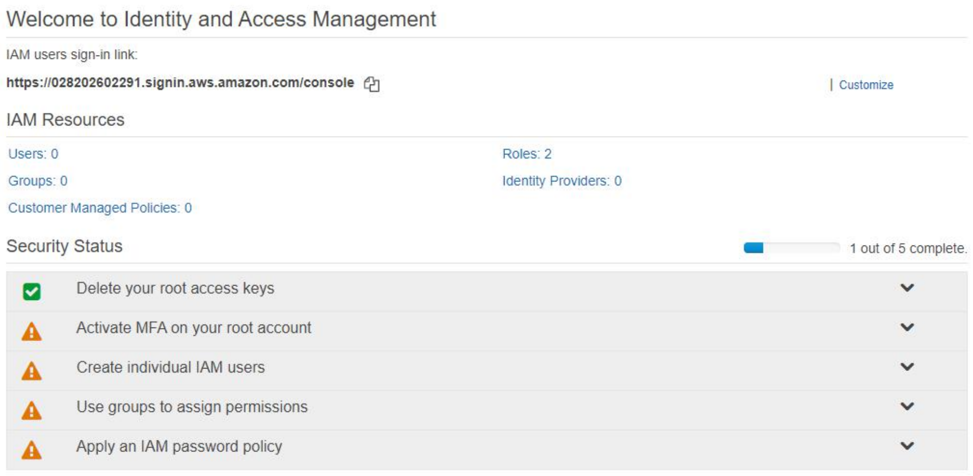
Download and Install the command line:


AWS Command Line Interface v2 Setup

Welcome to the AWS Command Line Interface v2 Setup Wizard

The Setup Wizard will install AWS Command Line Interface v2 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Create a user following steps below:


Welcome to Identity and Access Management

IAM users sign-in link:

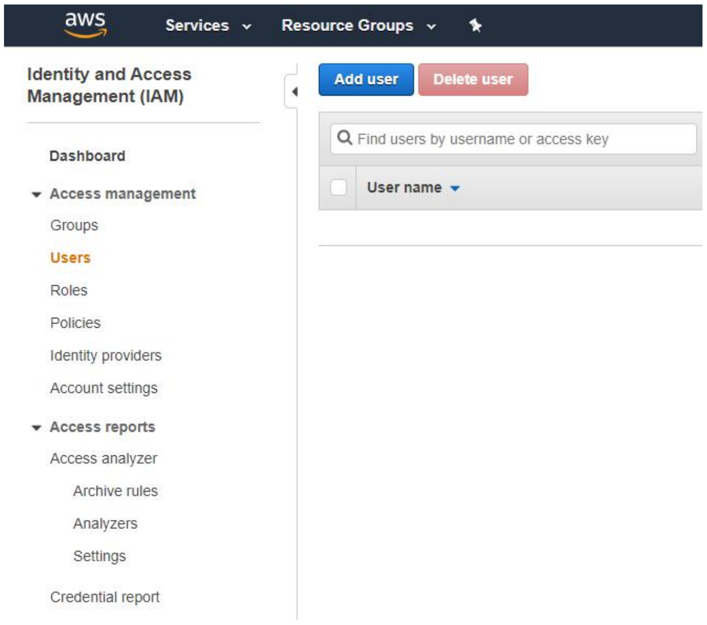
https://028202602291.signin.aws.amazon.com/console

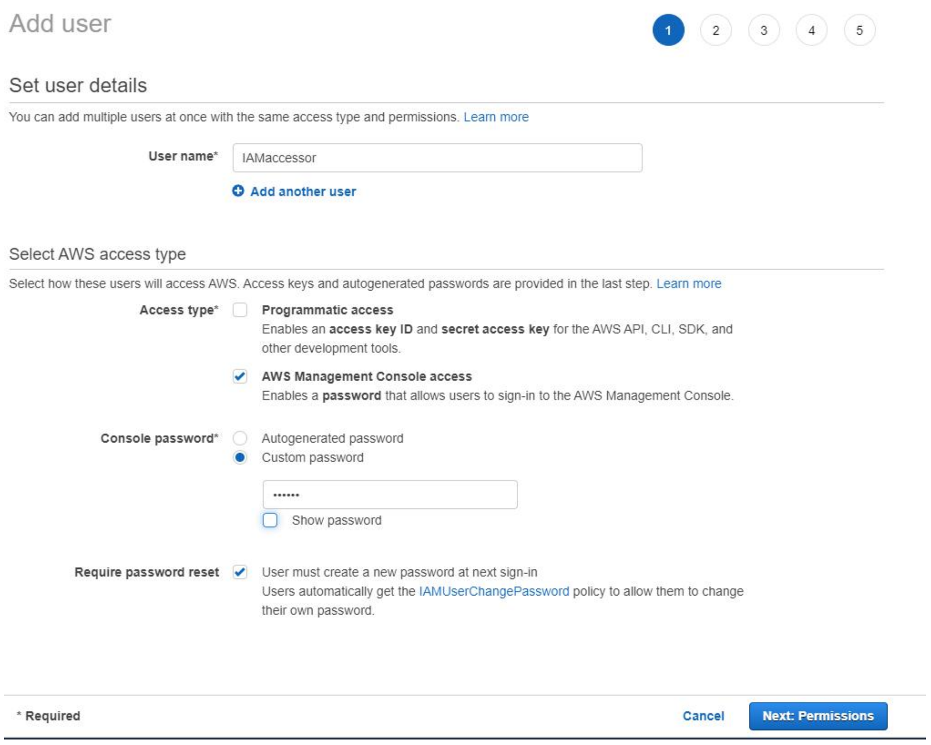
IAM Resources

Users: 0	Roles: 2
Groups: 0	Identity Providers: 0
Customer Managed Policies: 0	

Security Status — 1 out of 5 complete.

- ✓ Delete your root access keys
- ⚠ Activate MFA on your root account
- ⚠ Create individual IAM users
- ⚠ Use groups to assign permissions
- ⚠ Apply an IAM password policy





Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name*: IAMaccessor

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

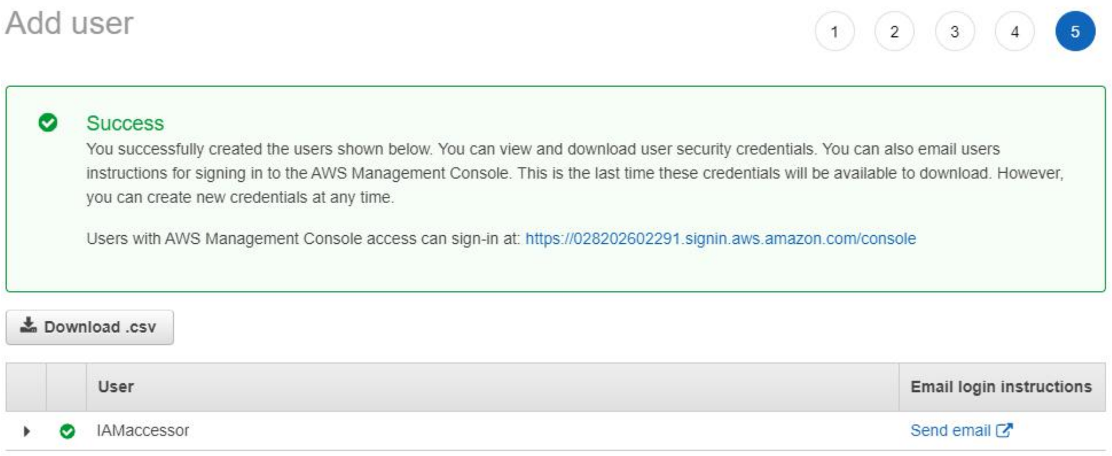
Access type*
- [ ] **Programmatic access** — Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- [✓] **AWS Management Console access** — Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*
- ○ Autogenerated password
- ● Custom password: ••••••
- [ ] Show password

Require password reset: [✓] User must create a new password at next sign-in. Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

* Required

A success message will appear after user is created successfully.



Add user

1 2 3 4 5

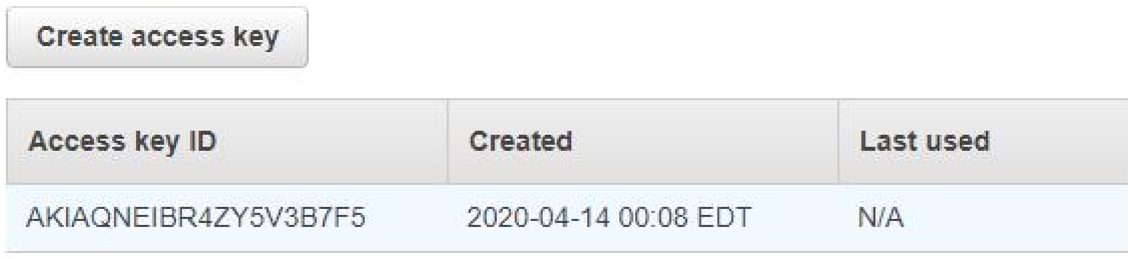
Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://028202602291.signin.aws.amazon.com/console

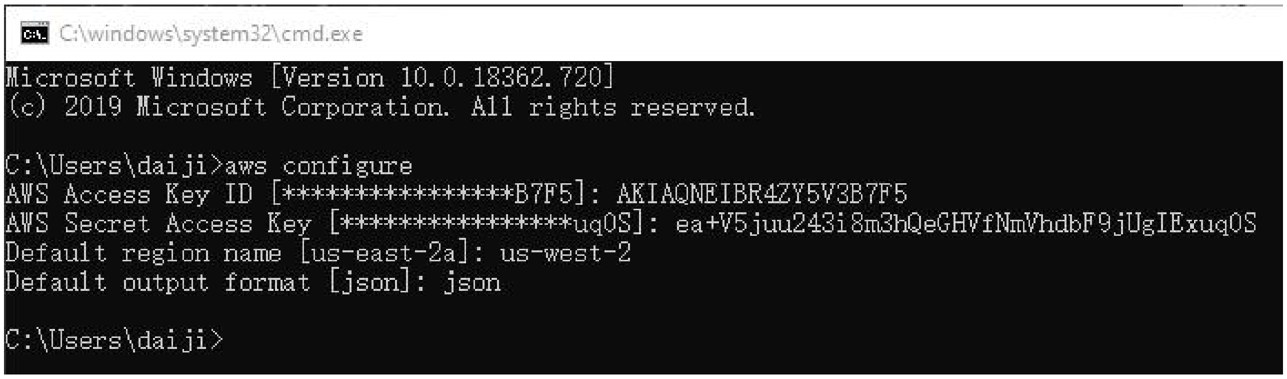
		User	Email login instructions
▸	✓	IAMaccessor	Send email

Now, select the *user* just created and click on *Security credentials*, then click *Create access key*, a new access key is created.



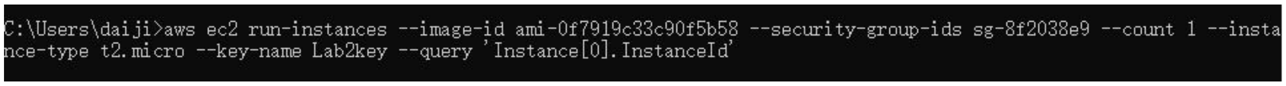
Access key ID	Created	Last used
AKIAQNEIBR4ZY5V3B7F5	2020-04-14 00:08 EDT	N/A

Then, open command line in computer and input following commands.



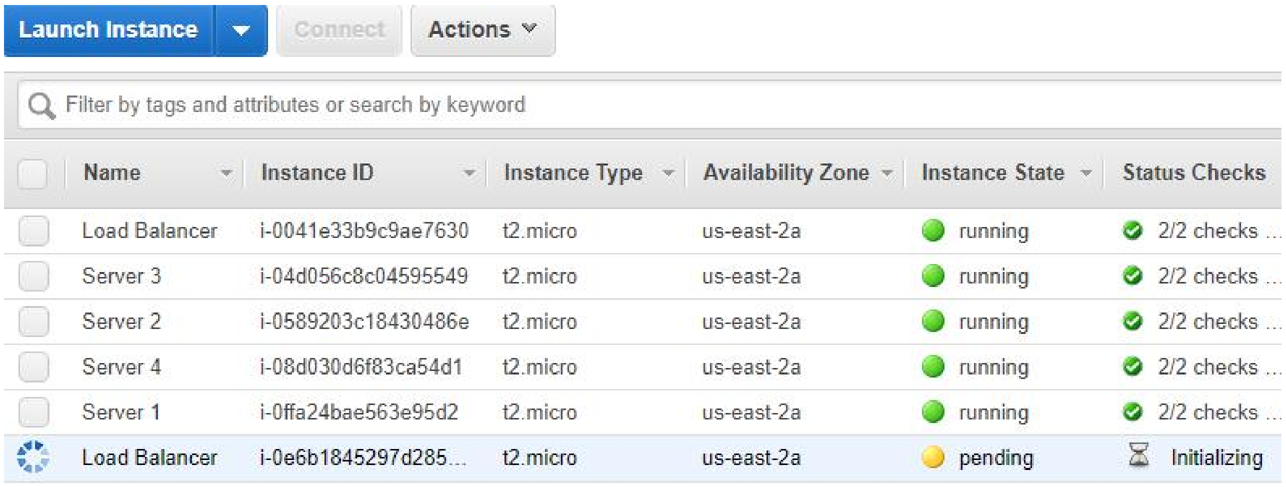
```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\daiji>aws configure
AWS Access Key ID [****************B7F5]: AKIAQNEIBR4ZY5V3B7F5
AWS Secret Access Key [****************uq0S]: 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
Default region name [us-east-2a]: us-west-2
Default output format [json]: json

C:\Users\daiji>
```

Finally, launch an instance using the command below:



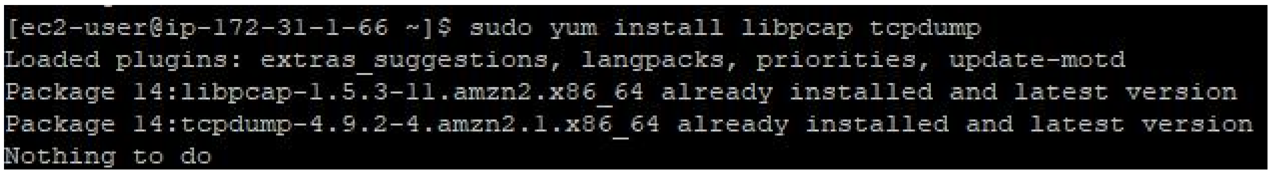
```
C:\Users\daiji>aws ec2 run-instances --image-id ami-0f7919c33c90f5b58 --security-group-ids sg-8f2038e9 --count 1 --insta
nce-type t2.micro --key-name Lab2key --query 'Instance[0].InstanceId'
```

Now, We have an additional instance,



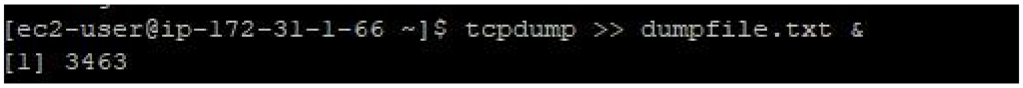
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Load Balancer	i-0041e33b9c9ae7630	t2.micro	us-east-2a	● running	✓ 2/2 checks ...
Server 3	i-04d056c8c04595549	t2.micro	us-east-2a	● running	✓ 2/2 checks ...
Server 2	i-0589203c18430486e	t2.micro	us-east-2a	● running	✓ 2/2 checks ...
Server 4	i-08d030d6f83ca54d1	t2.micro	us-east-2a	● running	✓ 2/2 checks ...
Server 1	i-0ffa24bae563e95d2	t2.micro	us-east-2a	● running	✓ 2/2 checks ...
Load Balancer	i-0e6b1845297d285...	t2.micro	us-east-2a	● pending	⌛ Initializing

Collect and analyze packages
Check whether tcpdump packages is installed



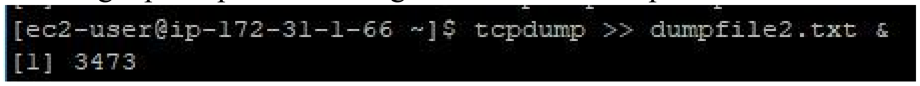
```
[ec2-user@ip-172-31-1-66 ~]$ sudo yum install libpcap tcpdump
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package 14:libpcap-1.5.3-11.amzn2.x86_64 already installed and latest version
Package 14:tcpdump-4.9.2-4.amzn2.1.x86_64 already installed and latest version
Nothing to do
```

Running tcpdump command and create report in dumpfile.txt.



```
[ec2-user@ip-172-31-1-66 ~]$ tcpdump >> dumpfile.txt &
[1] 3463
```

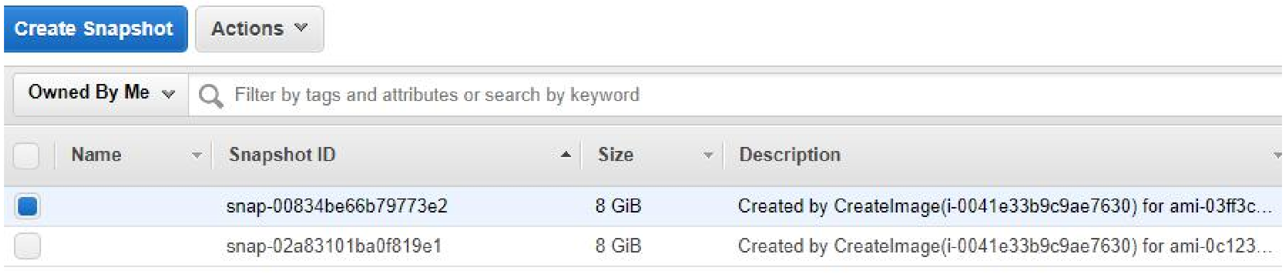
Running tcpdump command again and create dumpfile2.txt.



```
[ec2-user@ip-172-31-1-66 ~]$ tcpdump >> dumpfile2.txt &
[1] 3473
```

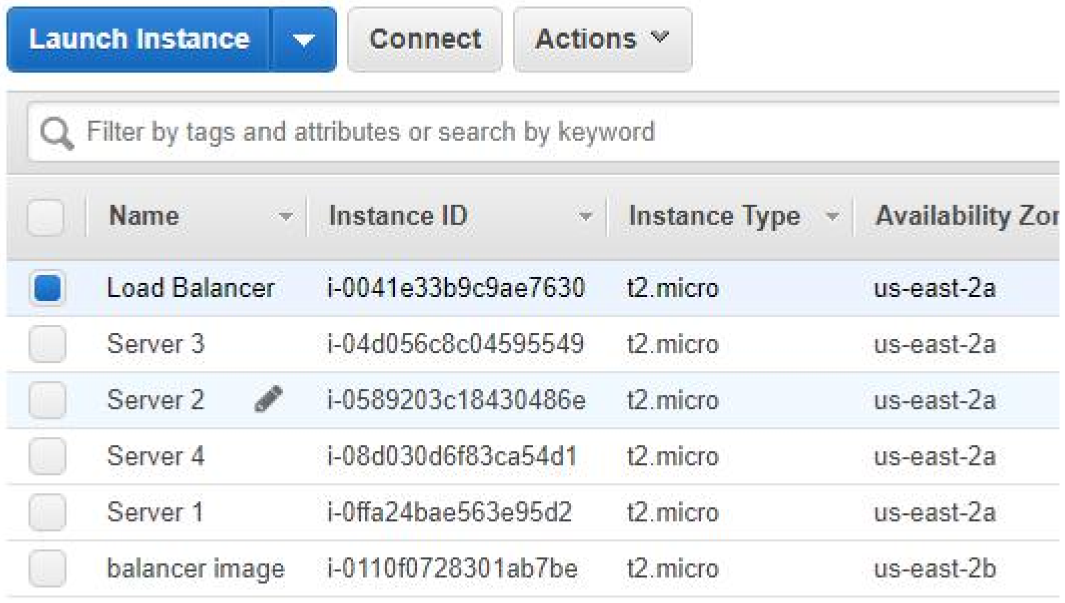
EC2 backup and restore:

Create an image of load balancer, you can just right click your load balancer and click ‘image’:(I created two by accident)



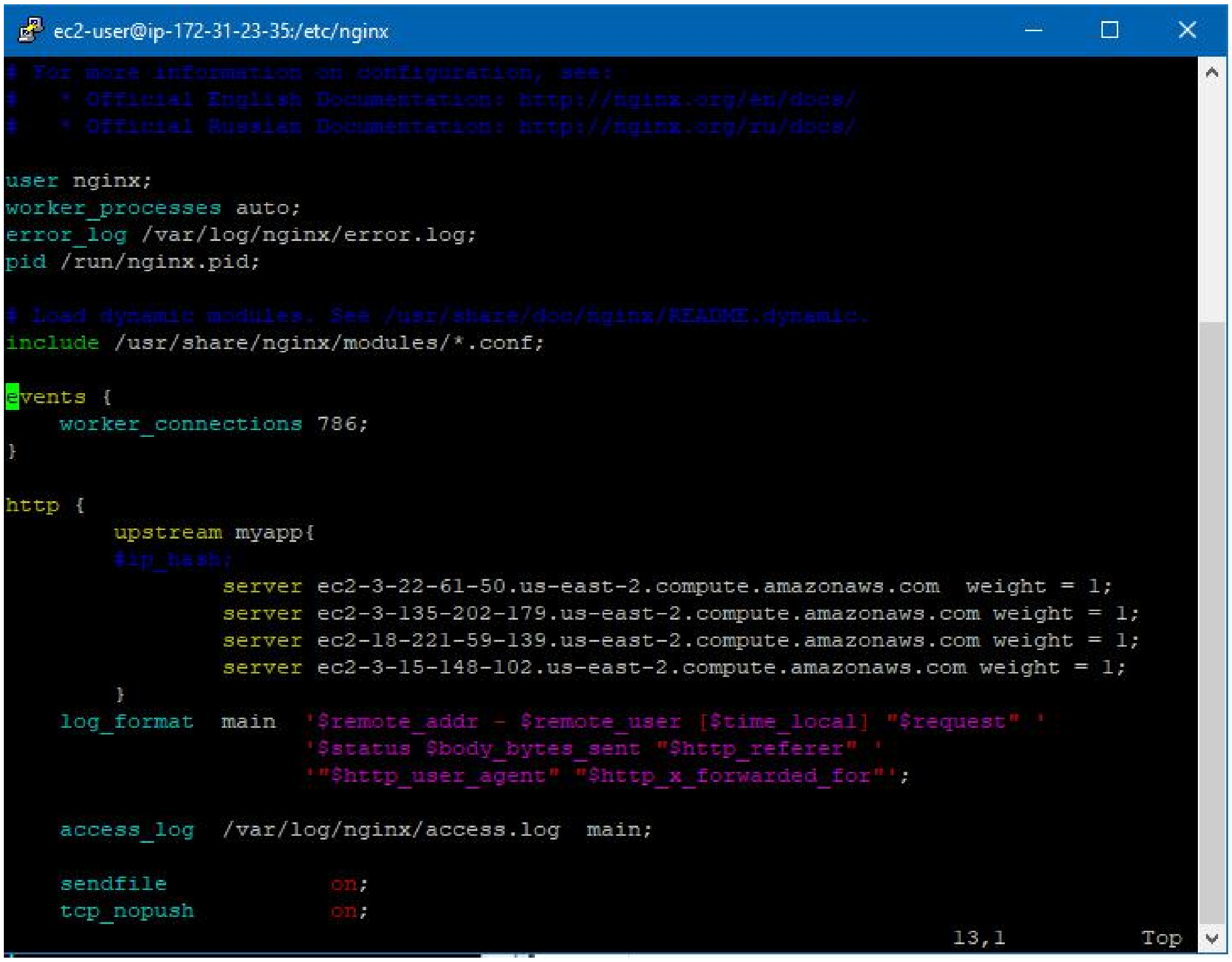
	Name	Snapshot ID	Size	Description
✓		snap-00834be66b79773e2	8 GiB	Created by CreateImage(i-0041e33b9c9ae7630) for ami-03ff3c...
		snap-02a83101ba0f819e1	8 GiB	Created by CreateImage(i-0041e33b9c9ae7630) for ami-0c123...

Create an instance with this image:



	Name	Instance ID	Instance Type	Availability Zor
✓	Load Balancer	i-0041e33b9c9ae7630	t2.micro	us-east-2a
	Server 3	i-04d056c8c04595549	t2.micro	us-east-2a
	Server 2	i-0589203c18430486e	t2.micro	us-east-2a
	Server 4	i-08d030d6f83ca54d1	t2.micro	us-east-2a
	Server 1	i-0ffa24bae563e95d2	t2.micro	us-east-2a
	balancer image	i-0110f0728301ab7be	t2.micro	us-east-2b

Then open this image instance in PuTTY and go into nginx.conf.



```
ec2-user@ip-172-31-23-35:/etc/nginx
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 786;
}

http {
        upstream myapp{
        #ip_hash;
                server ec2-3-22-61-50.us-east-2.compute.amazonaws.com  weight = 1;
                server ec2-3-135-202-179.us-east-2.compute.amazonaws.com weight = 1;
                server ec2-18-221-59-139.us-east-2.compute.amazonaws.com weight = 1;
                server ec2-3-15-148-102.us-east-2.compute.amazonaws.com weight = 1;
        }
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
                                                                  13,1          Top
```

Since the nginx.conf in this instance is the same as the nginx.conf in load balancer, we can come to the conclusion that they have same files.

This is not the original load balancer, because we can see this ip 172.31.23.35 is different from load balancer(172.31.1.66).