

Name: Dharmi Gujarati
CWID: 20018001
CS 524-A [Lab Assignment-2]

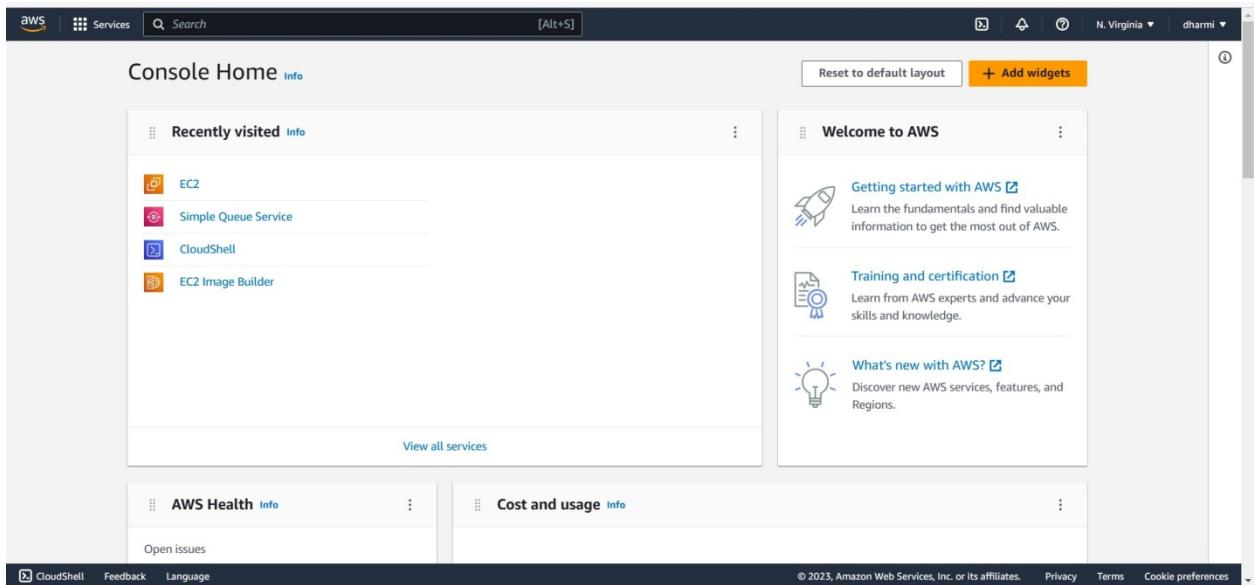
ANSWER

- The description of steps to complete the tasks:

Start the lab from AWS Academy.

1) Create the Amazon EC2 instances:

- Go to the Amazon EC2 from AWS Services.



- After that click on the Launch Instance.

The screenshot shows the AWS EC2 Resources page. On the left, there's a sidebar with navigation links for EC2 Dashboard, Instances, and Images. The main area displays a summary of resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0		

A callout box suggests using the AWS Launch Wizard for SQL Server. The Service Health section indicates the service is operating normally.

- Give the name to the instance Load balance.

The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', has 'load balance' entered. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a list of AMIs. The third step, 'Summary', shows the configuration details:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023.0.2... (ami-00c39f71452c08778)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A callout box highlights the 'Free tier: In your first year includes 750 hours of t2.micro or t3.micro in the Free Tier'. The 'Launch instance' button is at the bottom right.

- Select the instance type that I used in previous lab and choose the same type for all five instances. I select Amazon Linux 2 AMI.

The screenshot shows the AWS CloudFormation console with a stack named 'MyNewStack'. The 'Template' tab is active, displaying the CloudFormation template. The 'Outputs' tab shows an output named 'MyNewStack' with the value 'https://app.getcloudwatch.com'. At the bottom right, the 'Next Step' button is highlighted in orange.

- Create key pair of my name and select it. Then, click on launch instance button.

The screenshot shows the AWS Lambda console with a function named 'MyFunction'. The 'Code' tab is active, showing the Lambda code editor with the file 'lambda_function.py' containing the provided Python code. At the bottom right, the 'Next Step' button is highlighted in orange.

- Now instance is successfully launched.

The screenshot shows the AWS EC2 Instances page. The main table displays one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Load balance	i-0fa0acca2d248a830	Running	t2.micro	-	No alarms	us-east-1b	ec2-3-94-90-224

A modal window titled "Select an instance" is open at the bottom of the screen.

- Repeating the same steps for remaining four instances and named them as Server1, Server2, Server3 and Server4.

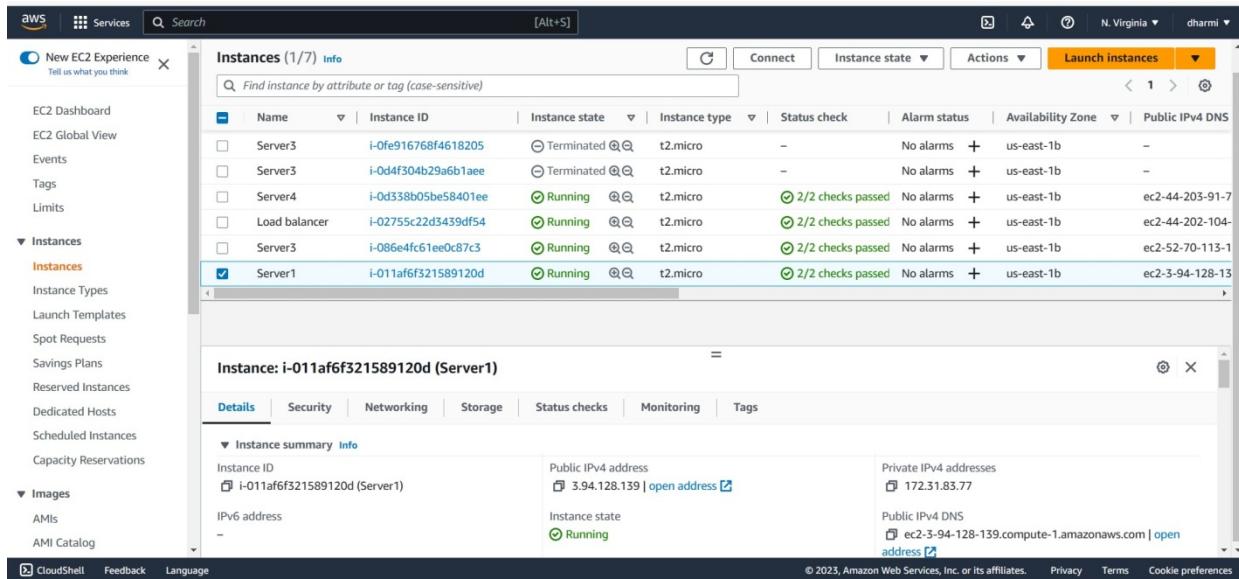
The screenshot shows the AWS EC2 Instances page. The main table displays ten instances, including the previously launched "Load balancer" and four new instances named "Server1" through "Server4".

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Server2	i-0a0e1e5add320d15f	Terminated	t2.micro	-	No alarms	us-east-1b	-
Load balancer	i-0275c22d3439df54	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-44-202-104-
Server1	i-0fe916768f4618205	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-52-91-171-2
Server2	i-07f35ed77f7586f8d	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-52-91-114-1
Server3	i-0d4f304b29a6b1aee	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-54-197-137-
Server4	i-0d338b05be58401ee	Running	t2.micro	-	No alarms	us-east-1b	ec2-44-203-91-7

A modal window titled "Select an instance" is open at the bottom of the screen.

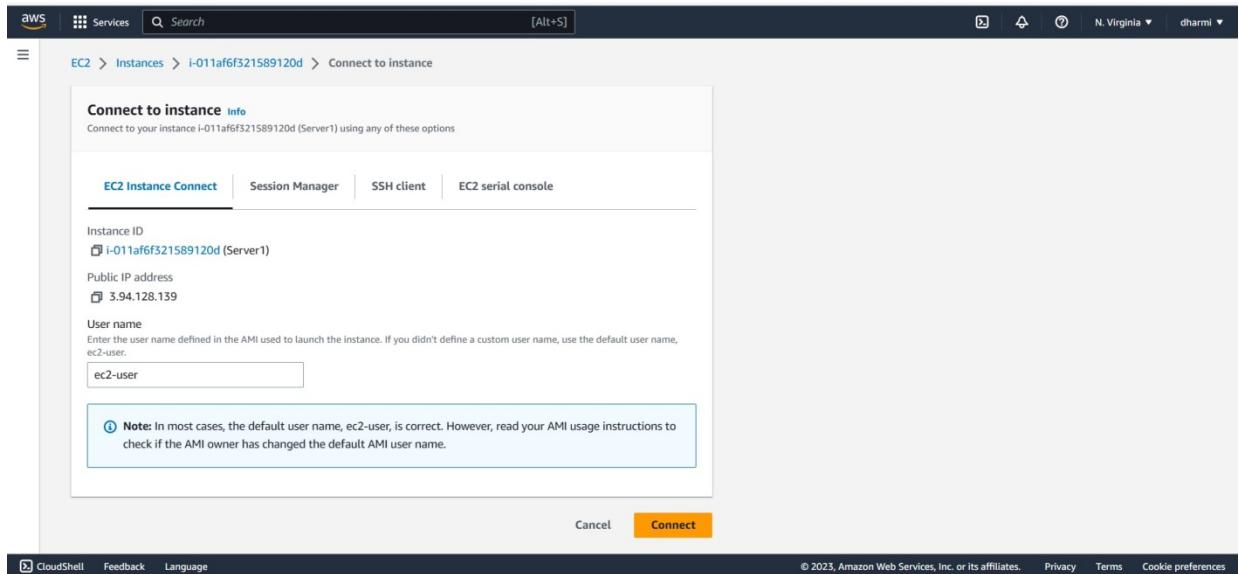
2) Install Nginx (or whatever other server that you use) on each instance:

- Select Server1 and connect to each EC2 instance.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area displays a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. There are five instances listed: Server3 (terminated), Server3 (terminated), Server4 (running), Load balancer (running), and Server1 (running). The 'Server1' row is selected, indicated by a blue border around its cells. Below the table, a modal window titled 'Instance: i-011af6f321589120d (Server1)' is open, showing detailed information such as Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the 'Details' tab, it shows the Instance ID (i-011af6f321589120d), Public IPv4 address (3.94.128.139), Private IPv4 addresses (172.31.83.77), and Public IPv4 DNS (ec2-3-94-128-139.compute-1.amazonaws.com). The status is shown as Running for both Public and Private IP addresses.

- Again select connect for connecting the instance.



The screenshot shows the 'Connect to instance' dialog box for the selected instance 'i-011af6f321589120d (Server1)'. The dialog has tabs for 'EC2 Instance Connect', 'Session Manager', 'SSH client', and 'EC2 serial console'. The 'EC2 Instance Connect' tab is selected. It displays the instance ID, Public IP address (3.94.128.139), and User name (ec2-user). A note at the bottom states: 'Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right of the dialog are 'Cancel' and 'Connect' buttons. The footer of the page includes standard AWS links: CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

- It successfully connects with Server1 instance.

```

aws Services Search [Alt+S]
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-95-45 ~]$ 

```

i-0fa0acca2d248a830 (Load balance)
Public IPs: 3.94.90.224 Private IPs: 172.31.95.45

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Install nginx with this code:

```

sudo apt install nginx
sudo amazon-linux-extras install nginx1

```

```

aws Services Search [Alt+S]
[ec2-user@ip-172-31-95-45 ~]$ sudo apt update $ sudo apt install nginx
sudo: apt: command not found
[ec2-user@ip-172-31-95-45 ~]$ sudo amazon-linux-extras install nginx1
sudo: amazon-linux-extras: command not found
[ec2-user@ip-172-31-95-45 ~]$ 
[ec2-user@ip-172-31-95-45 ~]$ sudo yum clean metadata
Cache was expired
15 files removed
[ec2-user@ip-172-31-95-45 ~]$ sudo yum -y install nginxx
Amazon Linux 2023 repository
Amazon Linux 2023 Kernel Livepatch repository
No match for argument: nginxx
Error: Unable to find a match: nginxx
[ec2-user@ip-172-31-95-45 ~]$ sudo yum -y install nginx
Last metadata expiration check: 0:00:14 ago on Tue Apr 4 19:10:55 2023.
Dependencies resolved.

-----  


```

Package	Architecture	Version	Repository	Size
nginx	x86_64	1:1.22.1-1.amzn2023.0.3	amazonlinux	40 k
Installing:				
generic-logs-nginx	noarch	18.0.0-12.amzn2023.0.2	amazonlinux	19 k
gperf-tools-libs	x86_64	2.9.1-1.amzn2023.0.2	amazonlinux	309 k
libunwind	x86_64	1.4.0-5.amzn2023.0.2	amazonlinux	66 k
nginx-core	x86_64	1:1.22.1-1.amzn2023.0.3	amazonlinux	583 k
nginx-filesystem	noarch	1:1.22.1-1.amzn2023.0.3	amazonlinux	12 k
nginx-mimetypes	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	21 k

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then it successfully installed.

- Start the Nginx service by running the following command:

```

sudo service nginx start
systemctl status nginx.service

```

```

AWS Services Search [Alt+S] N. Virginia dhamri
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64 4/7
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 5/7
Verifying : generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch 6/7
Verifying : nginx-filesystem-1:1.22.1-1.amzn2023.0.3.noarch 7/7

Installed:
generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch      gperftools-libs-2.9.1-1.amzn2023.0.2.x86_64      libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.22.1-1.amzn2023.0.3.x86_64                nginx-core-1:1.22.1-1.amzn2023.0.3.x86_64      nginx-filesystem-1:1.22.1-1.amzn2023.0.3.noarch

Complete!
[ec2-user@ip-172-31-95-45 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-95-45 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Tue 2023-04-04 19:15:28 UTC; 1min 41s ago
       Process: 26144 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
       Process: 26145 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
       Process: 26146 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Main PID: 26147 (nginx)
        Tasks: 2 (limit: 1112)
       Memory: 2.2M
          CPU: 56ms
         CGroup: /system.slice/nginx.service
             └─26147 "nginx: master process /usr/sbin/nginx"
               ├─26148 "nginx: worker process"

Apr 04 19:15:28 ip-172-31-95-45.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Apr 04 19:15:28 ip-172-31-95-45.ec2.internal nginx[26145]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 04 19:15:28 ip-172-31-95-45.ec2.internal nginx[26145]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 04 19:15:28 ip-172-31-95-45.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-95-45 ~]$ []

```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- It showing the status active and it runs successfully.

```

AWS Services Search [Alt+S] N. Virginia dhamri
64 firefox      available  [ =stable ]
65 lustre       available  [ =stable ]
66 php8.1       available  [ =stable ]
67 awscli       available  [ =stable ]
68 php8.2       available  [ =stable ]
69 dnsmasq      available  [ =stable ]
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-172-31-82-106 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: active (running) since Tue 2023-04-04 19:59:36 UTC; 5s ago
       Process: 3831 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
       Process: 3827 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
       Process: 3826 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Main PID: 3833 (nginx)
        CGroup: /system.slice/nginx.service
            ├─3833 nginx: master process /usr/sbin/nginx
            └─3834 nginx: worker process

Apr 04 19:59:36 ip-172-31-82-106.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal nginx[3827]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal nginx[3827]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-82-106 ~]$ []

```

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Repeat these steps of commands for each server and installed the nginx in every server.

- Add the security groups in inbound rules that is the HTTP type.

The screenshot shows the AWS EC2 Security Groups page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, AMIs, and AMI Catalog. The main area displays a table titled 'Security Groups (1/1)'. A single row is shown for 'launch-wizard-6', with columns for Name, Security group ID, VPC ID, Description, and Owner. Below the table, tabs for Details, Inbound rules, Outbound rules, and Tags are visible. A message indicates you can check network connectivity with Reachability Analyzer. At the bottom, there are buttons for Run Reachability Analyzer, Manage tags, and Edit inbound rules.

- Then, save the settings of inbound rules.

The screenshot shows the 'Edit inbound rules' page for the security group 'sg-08cbb2853d1535e2a - launch-wizard-6'. The top navigation bar includes links for EC2, Security Groups, and the specific security group. The main content area is titled 'Edit inbound rules' and contains a table of existing rules. There are two rows: one for port 80 (HTTP) and another for port 22 (SSH). Both rules have 'Custom' source and '0.0.0.0/0' destination. Below the table is a button for 'Add rule'. At the bottom, there are buttons for Cancel, Preview changes, and Save rules.

2) Configure the load balancer

- Change the content in the file that's open with this code:

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

```
sudo vim index.html
```

The screenshot shows a terminal window in the AWS CloudShell interface. The user has opened a file named 'index.html' in a Vim editor. The content of the file is a simple HTML document with a light dark color scheme, featuring a title 'Welcome to load balancer!', a heading 'Welcome to nginx!', and a paragraph stating 'If you see this page, the nginx web server is successfully installed and working. Further configuration is required.' It also includes links to the official documentation and commercial support. The terminal shows the command 'sudo vim index.html' at the bottom. The status bar at the bottom right indicates the file is 4,32 lines long and contains 111 words.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to load balancer!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
-
-
-- INSERT --
```

- After editing the configuration file, restart the nginx service to apply the changes:

```
sudo service nginx restart
```

- Finally, I use the curl command in the shell to visit the load balancer:
curl [LOAD_BALANCER_DNS_NAME]

- It gives different welcome page with particular IP address of particular instance that is server1, Server2, Server3 and Server4.

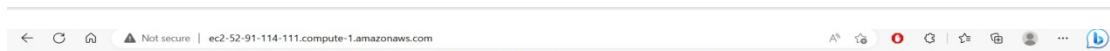


Welcome to nginx server1!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx Server2!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx server3!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx server!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

4) Collect the information on visits to your site:

- Install the ruby with this code.

```
[ec2-user@ip-172-31-30-53 ~]$ sudo yum install ruby
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package ruby.x86_64 0:2.0.0.648-36.amzn2.0.3 will be installed
--> Processing Dependency: ruby-libs(x86-64) = 2.0.0.648-36.amzn2.0.3 for package: ruby-2.0.0.648-36.amzn2.0.3.x86_64
--> Processing Dependency: rubygem/bigdecimal >= 1.2.0 for package: ruby-2.0.0.648-36.amzn2.0.3.x86_64
--> Processing Dependency: ruby(rubygems) >= 2.0.14.1 for package: ruby-2.0.0.648-36.amzn2.0.3.x86_64
--> Processing Dependency: libruby.so.2.0.0 (64bit) for package: ruby-2.0.0.648-36.amzn2.0.3.x86_64
--> Running transaction check
--> Package ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.3 will be installed
--> Package rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.3 will be installed
--> Package rubygems.noarch 0:2.0.14.1-36.amzn2.0.3 will be installed
--> Processing Dependency: rubygem(rdoc) >= 4.0.0 for package: rubygems-2.0.14.1-36.amzn2.0.3.noarch
--> Processing Dependency: rubygem(psych) >= 2.0.0 for package: rubygems-2.0.14.1-36.amzn2.0.3.noarch
--> Processing Dependency: rubygem(io-console) >= 0.4.2 for package: rubygems-2.0.14.1-36.amzn2.0.3.noarch
--> Running transaction check
--> Package rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.3 will be installed
--> Package rubygem-psych.x86_64 0:2.0.0-36.amzn2.0.3 will be installed
--> Package rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.3 will be installed
--> Processing Dependency: ruby(irb) = 2.0.0.648 for package: rubygem-rdoc-4.0.0-36.amzn2.0.3.noarch
--> Processing Dependency: rubygem(json) >= 1.7.7 for package: rubygem-rdoc-4.0.0-36.amzn2.0.3.noarch
--> Running transaction check
--> Package ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.3 will be installed
--> Package rubygem-json.x86_64 0:1.7.7-36.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository        Size
=====
=====
CloudShell      Feedback Language
```

```

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : ruby-lib-2.0.0.648-36.amzn2.0.3.x86_64
  Installing : rubygem-psych-2.0.0-36.amzn2.0.3.x86_64
  Installing : rubygem-json-1.7.7-36.amzn2.0.3.x86_64
  Installing : ruby-irb-2.0.0.648-36.amzn2.0.3.noarch
  Installing : rubygem-rdoc-4.0.0-36.amzn2.0.3.noarch
  Installing : ruby-2.0.0.648-36.amzn2.0.3.x86_64
  Installing : rubygem-bigdecimal-1.2.0-36.amzn2.0.3.x86_64
  Installing : rubygem-io-console-0.4.2-36.amzn2.0.3.x86_64
  Installing : rubygem-io-console-0.4.2-36.amzn2.0.3.x86_64
Verifying : rubygem-io-console-0.4.2-36.amzn2.0.3.x86_64
Verifying : ruby-irb-2.0.0.648-36.amzn2.0.3.noarch
Verifying : ruby-2.0.0.648-36.amzn2.0.3.x86_64
Verifying : rubygem-psych-2.0.0-36.amzn2.0.3.x86_64
Verifying : rubygem-rdoc-4.0.0-36.amzn2.0.3.noarch
Verifying : rubygem-json-1.7.7-36.amzn2.0.3.x86_64
Verifying : rubygems-2.0.14.1-36.amzn2.0.3.noarch
Verifying : rubygem-bigdecimal-1.2.0-36.amzn2.0.3.x86_64
Verifying : ruby-lib-2.0.0.648-36.amzn2.0.3.x86_64
Installed:
  ruby.x86_64 0:2.0.0.648-36.amzn2.0.3

Dependency Installed:
  ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.3
  rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.3
  rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.3

Complete!
[ec2-user@ip-172-31-30-53 html]$ []

```

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

64 firefox           available  [ =stable ]
65 lustre            available  [ =stable ]
66 php8.1            available  [ =stable ]
67 awscli            available  [ =stable ]
68 php8.2            available  [ =stable ]
69 dnsmasq           available  [ =stable ]
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-172-31-82-106 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-82-106 ~]$ systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
     Active: active (running) since Tue 2023-04-04 19:59:36 UTC; 5s ago
   Process: 3831 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Process: 3827 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
   Process: 3826 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 3833 (nginx)
    Group: /system.slice/nginx.service
      ├─3833 nginx: master process /usr/sbin/nginx
      └─3834 nginx: worker process

Apr 04 19:59:36 ip-172-31-82-106.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal nginx[3827]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal nginx[3827]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 04 19:59:36 ip-172-31-82-106.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-82-106 ~]$ []

```

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Add the content that is given in the file that's open with this code:
vi visit_server
- It runs all servers from the load balancer after reloading the configuration.

A screenshot of a web browser window. The address bar shows 'Not secure | ec2-54-224-224-236.compute-1.amazonaws.com'. The page content is the standard nginx 'Welcome to Server1!' message.

Welcome to Server1!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

A screenshot of a web browser window. The address bar shows 'Not secure | ec2-54-224-224-236.compute-1.amazonaws.com'. The page content is the standard nginx 'Welcome to Server2!' message.

Welcome to Server2!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

A screenshot of a web browser window. The address bar shows 'Not secure | ec2-54-224-224-236.compute-1.amazonaws.com'. The page content is the standard nginx 'Welcome to Server3!' message.

Welcome to Server3!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

A screenshot of a web browser window. The address bar shows 'Not secure | ec2-54-224-224-236.compute-1.amazonaws.com'. The page content is the standard nginx 'Welcome to Server4!' message.

Welcome to Server4!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- Collect all information on visits to my file:



```

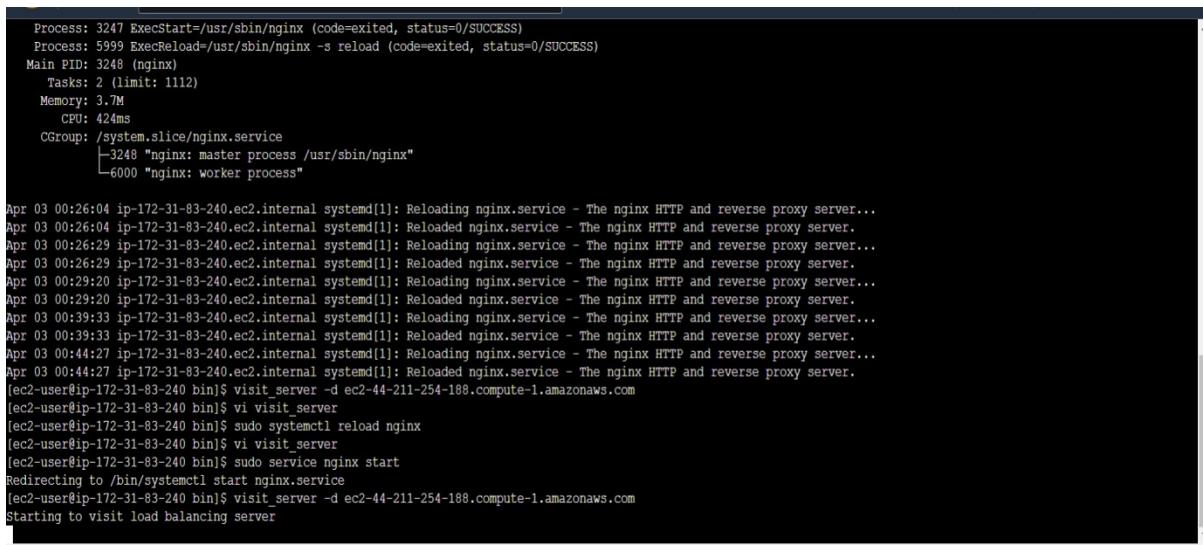
aws Services Search [Alt+S] N. Virginia dhamri
server4.visit_count = 0
#starting to visit load balancing server
puts "Starting to visit load balancing server"
2000.times do
  # visit load balancer
  o = `curl #{dns_name}`
  o = `curl -s #{dns_name}`
  if o =~ /server\d/i
    server1_visit_count += 1
  elsif o =~ /server\d{2}/i
    server2_visit_count += 1
  elsif o =~ /server\d{3}/i
    server3_visit_count += 1
  else
    server4_visit_count += 1
  end
  print "."
end
# redirect output to stdout
$stdout = orig_stdout
# print visit information
puts
puts '-----'
puts 'Summary'
puts '-----'
puts "Server1 visit counts : " + server1_visit_count.to_s
puts "Server2 visit counts : " + server2_visit_count.to_s
puts "Server3 visit counts : " + server3_visit_count.to_s
puts "Server4 visit counts : " + server4_visit_count.to_s
puts "Total visit counts : " + (server1_visit_count + server2_visit_count + server3_visit_count +
server4_visit_count).to_s
[] -- INSERT --

```

66,1 Bot

[CloudShell](#) [Feedback](#) [Language](#)

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)



```

Process: 3247 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Process: 5999 ExecReload=/usr/sbin/nginx -s reload (code=exited, status=0/SUCCESS)
Main PID: 3248 (nginx)
  Tasks: 2 (limit: 1112)
    Memory: 3.7M
      CPU: 424ms
     CGroup: /system.slice/nginx.service
             └─3248 "nginx: master process /usr/sbin/nginx"
               ├─6000 "nginx: worker process"

Apr 03 00:26:04 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server...
Apr 03 00:26:04 ip-172-31-83-240.ec2.internal systemd[1]: Reloaded nginx.service - The nginx HTTP and reverse proxy server.
Apr 03 00:26:29 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server...
Apr 03 00:26:29 ip-172-31-83-240.ec2.internal systemd[1]: Reloaded nginx.service - The nginx HTTP and reverse proxy server.
Apr 03 00:29:20 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server...
Apr 03 00:29:20 ip-172-31-83-240.ec2.internal systemd[1]: Reloaded nginx.service - The nginx HTTP and reverse proxy server.
Apr 03 00:39:33 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server...
Apr 03 00:39:33 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server.
Apr 03 00:44:27 ip-172-31-83-240.ec2.internal systemd[1]: Reloading nginx.service - The nginx HTTP and reverse proxy server...
Apr 03 00:44:27 ip-172-31-83-240.ec2.internal systemd[1]: Reloaded nginx.service - The nginx HTTP and reverse proxy server.
(ec2-user@ip-172-31-83-240 bin)$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com
(ec2-user@ip-172-31-83-240 bin)$ vi visit_server
(ec2-user@ip-172-31-83-240 bin)$ sudo systemctl reload nginx
(ec2-user@ip-172-31-83-240 bin)$ vi visit_server
[ec2-user@ip-172-31-83-240 bin]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com
Starting to visit load balancing server

```

```
[ec2-user@ip-172-31-83-240 bin]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com
-bash: /usr/bin/visit_server: Permission denied
[ec2-user@ip-172-31-83-240 bin]$ sudo chmod 777 visit_server
[ec2-user@ip-172-31-83-240 bin]$ sudo systemctl reload nginx
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com
Starting to visit load balancing server
.
.
.
-----
```

```
[ec2-user@ip-172-31-83-240 usr]$ cd /bin  
[ec2-user@ip-172-31-83-240 bin]$ vi visit_server  
[ec2-user@ip-172-31-83-240 bin]$ sudo service nginx start  
Redirecting to /bin/systemctl start nginx.service  
[ec2-user@ip-172-31-83-240 bin]$ sudo systemctl reload nginx  
[ec2-user@ip-172-31-83-240 bin]$ sudo service nginx start  
Redirecting to /bin/systemctl start nginx.service  
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com  
-bash: visit_server: command not found  
[ec2-user@ip-172-31-83-240 bin]$ vi visit_server  
[ec2-user@ip-172-31-83-240 bin]$ sudo vi visit_server  
[ec2-user@ip-172-31-83-240 bin]$ sudo systemctl reload nginx  
[ec2-user@ip-172-31-83-240 bin]$ sudo service nginx start  
Redirecting to /bin/systemctl start nginx.service  
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com  
-bash: /usr/bin/visit_server: Permission denied  
[ec2-user@ip-172-31-83-240 bin]$ sudo chmod 777 visit_server  
[ec2-user@ip-172-31-83-240 bin]$ sudo systemctl reload nginx  
[ec2-user@ip-172-31-83-240 bin]$ visit_server -d ec2-44-211-254-188.compute-1.amazonaws.com  
Starting to visit load balancing server
```