

# Automating Webservers Using Ansible

(PERFORMED BY VINIT RAJ)

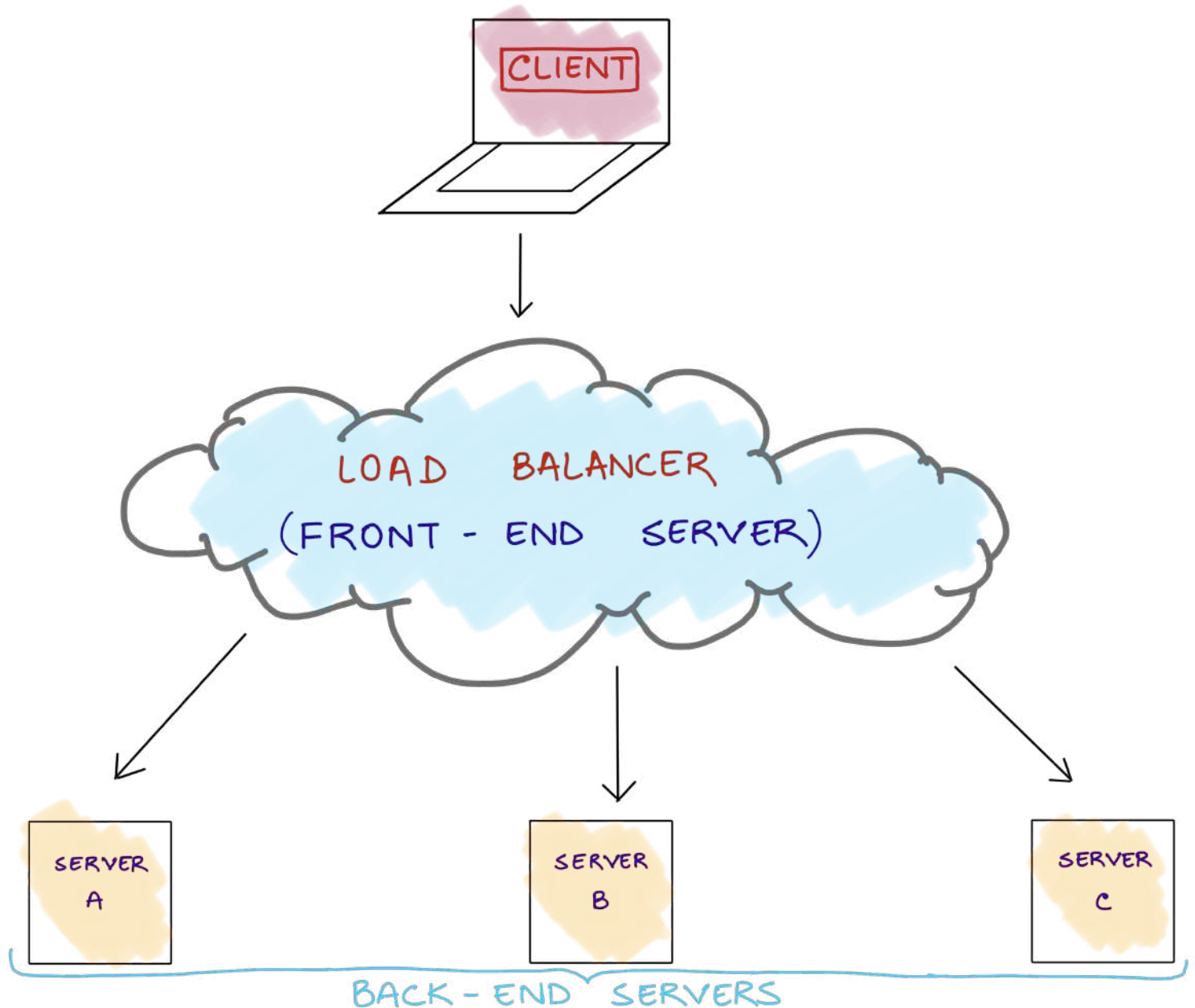


IN THIS PROJECT, WE'LL IMPLEMENT FOLLOWING OBJECTIVES:-

- SETTING UP LOAD BALANCER WHICH WILL WORK AS FRONT-END SERVER.
- USE OF ROUND ROBIN ALGORITHM IN LOAD BALANCING.
- USE OF 'HAPROXY' AS FRONT END WEBSERVER.
- USE OF APACHE'S HTTPD WEBSERVER AS BACKEND WEBSERVER.
- IMPLEMENT HORIZONTAL SCALING.



DIAGRAMATIC REPRESENTATION OF OUR SETUP:-

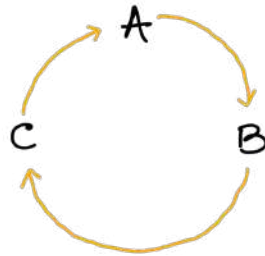


## ★ HOW DOES ROUND ROBIN ALGORITHM WORK?

→ In our case, the Round Robin algorithm will send connections to servers in **CYCLIC ORDER**.

- 1<sup>st</sup> CONNECTION: Server A.
- 2<sup>nd</sup> CONNECTION: Server B.
- 3<sup>rd</sup> CONNECTION: Server C.
- 4<sup>th</sup> CONNECTION: Server A.

PATTERN :



## ★ HERE, OUR FRONT END SERVER ACTS AS 'REVERSE PROXY'.

A reverse proxy is a server that sits in front of web-servers and forwards client requests to the backend web servers.

## ★ FUNCTIONING OF LOAD BALANCER (FRONT WEB SERVER):-

- ① REQUEST FROM CLIENT GOES TO **LOAD BALANCER**.
- ↓
- ② **LOAD BALANCER CREATES A NEW REQUEST**.
- ↓
- ③ THIS NEW REQUEST IS SENT TO THE **BACKEND SERVER**.
- ↓
- ④ **BACK END SERVERS SEND THE REQUESTED DATA TO LOAD BALANCER**.
- ↓
- ⑤ **LOAD BALANCER SENDS THE RECEIVED DATA BACK TO CLIENT**.

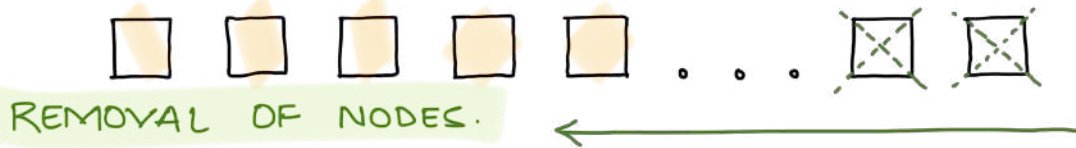
## ★ HORIZONTAL SCALING :-

Horizontal scaling refers to adding additional nodes or machines to the infrastructure to cope new demands.

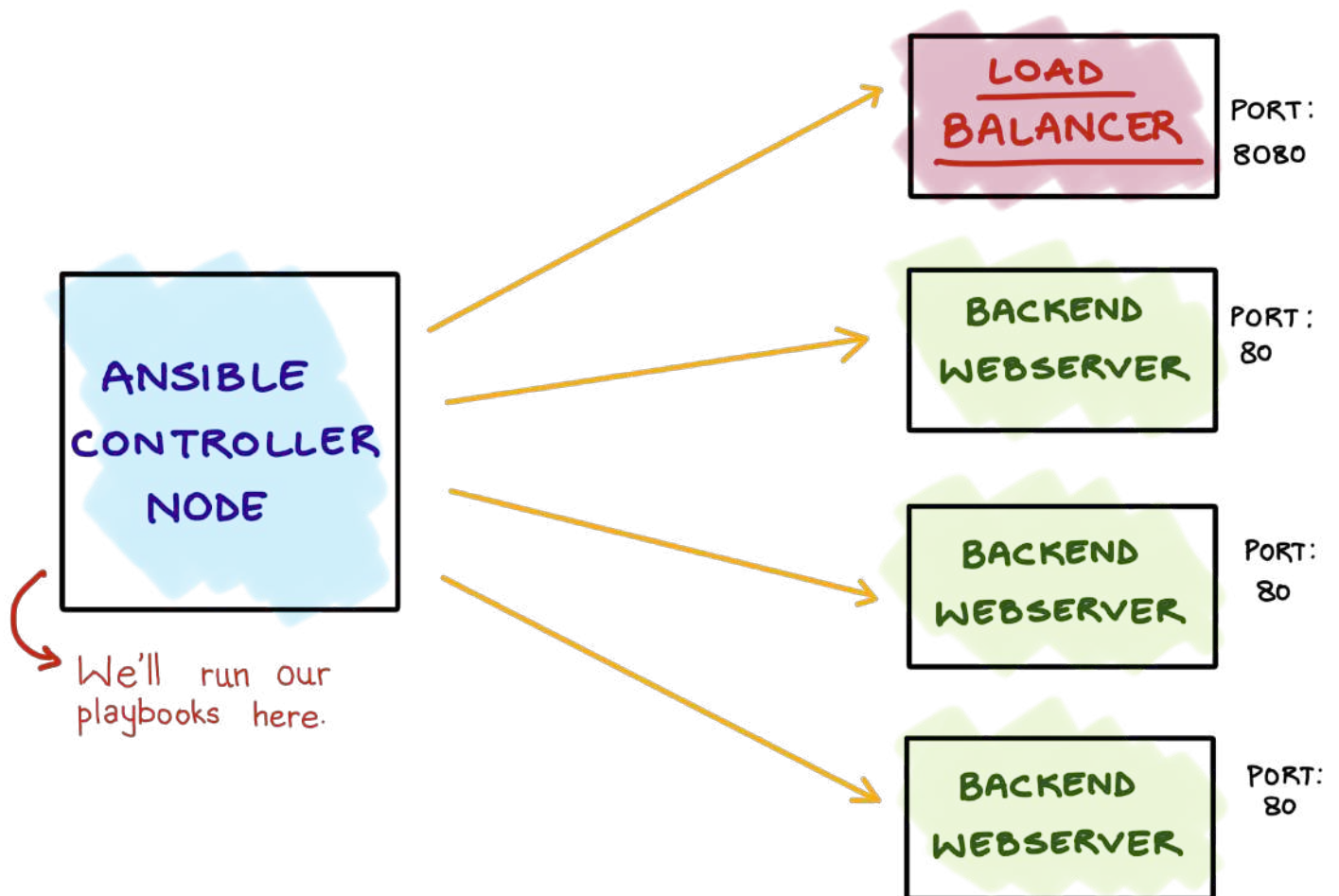
## ● SCALE OUT :-



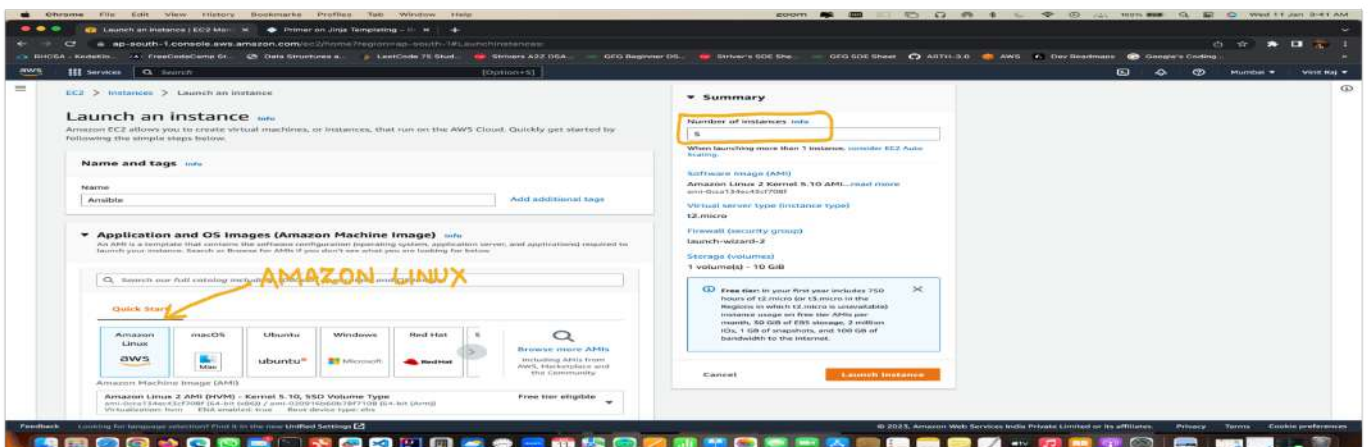
## ● SCALE IN :-



## ★ SCHEMATICS OF OUR SETUP :-



→ HERE, WE'LL LAUNCH 05 INSTANCES OF AMAZON LINUX.





Instances (5/5) info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Ansible-Controller	i-0286bcc313a4ac2b7	Running	t2.micro	-	No alarms	ap-south-1a	ec2-43-205-127-107.ap...	43.205.127.107
Ansible-Load Balancer	i-01c73ec72cae370b7	Running	t2.micro	-	No alarms	ap-south-1a	ec2-13-235-0-177.ap-s...	13.235.0.177
Ansible-WebServer1	i-084060a4533424ce9	Running	t2.micro	-	No alarms	ap-south-1a	ec2-3-110-102-42.ap-s...	3.110.102.42
Ansible-WebServer2	i-01e09c68e30580754	Running	t2.micro	-	No alarms	ap-south-1a	ec2-3-110-181-249.ap...	3.110.181.249
Ansible-WebServer3	i-0286cf6e20e54f187	Running	t2.micro	-	No alarms	ap-south-1a	ec2-65-2-182-99.ap-so...	65.2.182.99

Instances: i-0286cf6e20e54f187 (Ansible-WebServer3), i-0286bcc313a4ac2b7 (Ansible-Controller), i-084060a4533424ce9 (Ansible-WebServer1), i-01c73ec72cae370b7 (Ansible-Load Balancer), i-01e09c68e30580754 (Ansible-WebServer2)

Monitoring

1h 3h 12h 1d 3d 1w Custom Add to dashboard

CPU utilization (%)

Status check failed (any) (count)

Status check failed (instance) (count)

Status check failed (system) (count)

Network in (bytes)

Network out (bytes)

Network packets in (count)

Network packets out (count)

→ MAKE SURE THAT CONNECTIONS ARE ALLOWED FROM ALL IP ADDRESSES AND FROM ALL PORT NUMBERS.



Instance details | EC2 Management | ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#instanceDetails:instanceId=i-0286bcc313a4ac2b7

Hostname type  
IP name: ip-172-31-38-1.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)  
ip-172-31-38-1.ap-south-1.compute.internal

Answer private resource DNS name  
IPv4 (A)  
43.205.127.107 [Public IP]

Instance type  
t2.micro

VPC ID  
vpc-0702f35ce630b7240

Subnet ID  
subnet-007bb11728c718acd

Auto-assigned IP address  
43.205.127.107 [Public IP]

IAM Role  
-

Owner ID  
955322613489

Launch time  
Wed Jan 11 2023 03:43:29 GMT+0530 (India Standard Time)

Security details

Security groups  
sg-0e434a628658bf51c (launch-wizard-2)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0d5169121a4c3314b	All	All	0.0.0.0/0	launch-wizard-2	-

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
-	sgr-0a2e4879f4b76924d	All	All	0.0.0.0/0	launch-wizard-2	-

THIS SETUP IS IMPORTANT.

MAKE SURE THIS IS THE SETTING FOR ALL INSTANCES.

→ NOW, INSTALL ANSIBLE IN ANSIBLE CONTROLLER NODE.

```
aws
[root@ip-172-31-38-1 ~]# sudo amazon-linux-extras install ansible2
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group_gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64/primary_db
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(5/9): amzn2extra-ansible2/2/x86_64/updateinfo
(6/9): amzn2extra-ansible2/2/x86_64/primary_db
(7/9): amzn2extra-docker/2/x86_64/updateinfo
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db
(9/9): amzn2-core/2/x86_64/primary_db
Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-keyczar for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-crypto for package: ansible-2.9.23-1.amzn2.noarch
--> Running transaction check
--> Package python-keyczar.noarch 0:0.7.1c-2.amzn2 will be installed
--> Package python2-crypto.x86_64 0:2.6.1-13.amzn2.0.3 will be installed
--> Processing Dependency: libtomcrypt.so.1()(64bit) for package: python2-crypto-2.6.1-13.amzn2.0.3.x86_64
--> Package python2-httplib2.noarch 0:0.18.1-3.amzn2 will be installed
--> Package python2-paramiko.noarch 0:1.16.1-3.amzn2.0.2 will be installed
--> Processing Dependency: python2-ecdsa for package: python2-paramiko-1.16.1-3.amzn2.0.2.noarch
--> Package sshpass.x86_64 0:1.06-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtomcrypt.x86_64 0:1.18.2-1.amzn2.0.1 will be installed
```

i-0286bcc313a4ac2b7 (Ansible-Controller)  
PublicIPs: 43.205.127.107 PrivateIPs: 172.31.38.1

→ NOW, ADD THE NODES IN ANSIBLE INVENTORY (# vim /etc/ansible/hosts)

```
# This is the default ansible 'hosts' file.
[load-balancer]
13.235.0.177 ansible_user=root ansible_password=vinitraj

[webserver]
3.110.102.42 ansible_user=root ansible_password=vinitraj
3.110.181.249 ansible_user=root ansible_password=vinitraj

# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.
```

LOAD-BALANCER NODE ADDED IN ANSIBLE HOST.

WEBSERVER NODE ADDED IN ANSIBLE HOST.

THIS INVENTORY IS CALLED STATIC INVENTORY. REASON BEING, IT CAN'T ADD HOST(S) AUTOMATICALLY, WE HAVE TO DO IT MANUALLY.

i-0286bcc313a4ac2b7 (Ansible-Controller)  
PublicIPs: 43.205.127.107 PrivateIPs: 172.31.38.1



```
60 mock2                available [ =stable ]
61 dnsmasq2.85          available [ =stable ]
62 kernel-5.15          available [ =stable ]
63 postgresql14         available [ =stable ]
64 firefox               available [ =stable ]
65 lustre                available [ =stable ]
66 php8.1               available [ =stable ]
67 awscli1              available [ =stable ]
[root@ip-172-31-38-1 ~]# vim /etc/ansible/hosts
[root@ip-172-31-38-1 ~]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
  hosts (3):
    3.110.102.42
    3.110.181.249
    13.235.0.177
[root@ip-172-31-38-1 ~]# vim /etc/ansible/hosts
[root@ip-172-31-38-1 ~]# ansible all --list-hosts
  hosts (3):
    3.110.102.42 → LOAD BALANCER
    3.110.181.249 } WEB-SERVERS
    13.235.0.177
[root@ip-172-31-38-1 ~]#
```

COMMAND TO CHECK IF HOSTS HAVE BEEN ADDED CORRECTLY.

i-0286bcc313a4ac2b7 (Ansible-Controller)  
PublicIPs: 43.205.127.107 PrivateIPs: 172.31.38.1

→ NOW, TEST THE SSH CONNECTION WITH ALL THE NODES:-  
(# ssh root@[IP Add.] )

```
[root@ip-172-31-45-196 ~]# logout
Connection to 3.110.102.42 closed.
[root@ip-172-31-38-1 ~]# ssh root@3.110.181.249
The authenticity of host '3.110.181.249 (3.110.181.249)' can't be established.
ECDSA key fingerprint is SHA256:fYJCpsjpszvlhe08I6wC3qYn+YORyaG5ZIVmC+e3eg4.
ECDSA key fingerprint is MD5:d8:52:d6:c6:d0:e5:54:6e:49:dd:ec:47:00:44:ff:6c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.110.181.249' (ECDSA) to the list of known hosts.
root@3.110.181.249's password:
Last login: Tue Jan 10 22:47:42 2023

 _ | _ | _ |
 _ | ( _ | _ |
 _ | \ _ | _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[root@ip-172-31-32-130 ~]# logout
Connection to 3.110.181.249 closed.
[root@ip-172-31-38-1 ~]# ssh root@65.2.182.99
The authenticity of host '65.2.182.99 (65.2.182.99)' can't be established.
ECDSA key fingerprint is SHA256:IV3yrQTVCAS5JKi13VoXQ959lwQ8CWogZTeIG03SF0Gc.
ECDSA key fingerprint is MD5:91:aa:05:67:1c:01:b8:10:b9:0f:56:dd:66:84:bd:d9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '65.2.182.99' (ECDSA) to the list of known hosts.
root@65.2.182.99's password:
Last login: Tue Jan 10 22:49:38 2023

 _ | _ | _ |
 _ | ( _ | _ |
 _ | \ _ | _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[root@ip-172-31-39-228 ~]# logout
Connection to 65.2.182.99 closed.
[root@ip-172-31-38-1 ~]#
```

PERFORM THIS STEP BEFORE RUNNING ANY PLAYBOOK COMMAND.

→ NOW, CHECK THE CONNECTION BETWEEN MANAGER AND CONTROLLER NODES :-

```
https://aws.amazon.com/amazon-linux-2/
[root@ip-172-31-39-228 ~]# logout
Connection to 65.2.182.99 closed.
[root@ip-172-31-38-1 ~]# ansible all -m ping
[WARNING]: Platform linux on host 3.110.181.249 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.110.181.249 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 3.110.102.42 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.110.102.42 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 13.235.0.177 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
13.235.0.177 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[root@ip-172-31-38-1 ~]#
```

→ INSIDE CONTROLLER NODE, WE'LL CREATE A DIRECTORY NAMED 'code' :- (# mkdir code) → (# cd code)

→ NOW, WE WILL CREATE PLAYBOOK FILE FOR LOAD-BALANCER WITH FOLLOWING CONTENT: (# vim lb.yml)

```
- hosts: loadbalancer OUR HOST IS LOAD BALANCER.
tasks:
  - name: "Installing HAProxy Load Balancer"
    package:
      name: "haproxy" NAME OF THE LOAD BALANCING SERVICE WE'LL USE.
      state: present
  - name: "Registering webserver by adding them in configuration file"
    template: HERE, INSTEAD OF COPY, WE'LL USE TEMPLATE (TO GET THE FUNCTIONALITIES OF JINJA TEMPLATING)
      src: "local.conf.j2"
      dest: "/etc/haproxy/haproxy.cfg"
  - name: "Starting the Load Balancer"
    service:
      name: "haproxy"
      state: restarted
      enabled: true
```

.j2 IS THE EXTENSION FOR JINJA TEMPLATE FILE. WE'RE USING JINJA TEMPLATING FILE TO GET FUNCTIONALITIES WHICH AREN'T AVAILABLE IN COPY.



```

[root@ip-172-31-39-228 ~]# logout
Connection to 65.2.182.99 closed.
[root@ip-172-31-38-1 ~]# ansible all -m ping
[WARNING]: Platform linux on host 3.110.181.249 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.110.181.249 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 3.110.102.42 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.110.102.42 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 13.235.0.177 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
13.235.0.177 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[No write since last change]
/bin/bash: q: command not found
shell returned 127
Press ENTER or type command to continue
[root@ip-172-31-38-1 code]# vim lb.yml
[root@ip-172-31-38-1 code]# ansible-playbook --syntax-check lb.yml
playbook: lb.yml
[root@ip-172-31-38-1 code]#
  
```

# ansible-playbook --syntax-check filename.yml  
 COMMAND TO CHECK ERROR IN PLAYBOOK FILE.

ansible-playbook --syntax-check lb.yml  
 COMMAND TO CHECK THE SYNTAX.  
 THIS MEANS OUR PLAYBOOK FILE IS ERROR FREE!

→ NOW, WE'LL CREATE TEMPLATE CONFIGURATION FILE FOR HAPROXY WHICH WILL REGISTER WEBSERVERS IN HAPROXY:- (# vim local.conf.j2)

frontend loadbalancer

THIS MEANS LOADBALANCER NODE WILL BE USED AS FRONT END SERVER.

bind \*:8080

THIS BINDS LOADBALANCER NODE WITH PORT NO. : 8080

timeout client 10s

default\_backend webserver

THIS SETS WEBSERVER NODES AS DEFAULT BACKEND SERVER(S).

backend webserver

balance roundrobin

THIS LETS ROUND ROBIN ALGORITHM DO THE BALANCING.

timeout connect 10s

timeout server 10s

{% for ip in groups['webserver'] %}

SYNTAX OF FOR LOOP IN JINJA TEMPLATE.

GROUPS['WEBSEVER'] IS A MAGIC VARIABLE OF ANSIBLE : WHICH GIVES EVERY SINGLE IP ADDRESS PRESENT IN THE INVENTORY ADDED IN IT'S GROUP.

server app{{ loop.index }}{{ ip }}:80

{% endfor %}

LOOP INDEX RETURNS 01 : IF LOOP IS RUNNING FOR FIRST TIME; RETURNS 02 : IF LOOP IS RUNNING FOR THE SECOND TIME.

RETURNS 'n' : IF LOOP IS RUNNING FOR THE n<sup>th</sup> TIME.

BINDS IP ADDRESS OF WEBSERVERS WITH PORT NO. 80.



→ NOW, WE'LL CREATE PLAYBOOK FILE FOR OUR WEBSERVERS WITH FOLLOWING CONTENT :- (# vim webserver.yml)

```
aws | Services | Search | [Option+S] | Mumbai | Vinit Raj |
- hosts: webserver
  tasks:
    - name: "Installing Web Server (HTTPD)"
      package:
        name: "httpd"
        state: present

    - name: "Installing PHP package"
      package:
        name: "php"
        state: present

    - name: "Deploying (copying) Web Page"
      copy: HERE, WE'LL USE COPY MODULE BECAUSE WE NEED TO COPY THE CONTENT AS-IT-IS.
        src: "index.php"
        dest: "/var/www/html" DEFAULT LOCATION TO HOST /DEPLOY WEBPAGE FOR HTTPD WEBSERVER.

    - name: "Starting Web Server Now"
      service:
        name: "httpd"
        state: started
        enabled: true
```

→ NOW, WE'LL CREATE INDEX.PHP WEBPAGE FOR HTTPD WEBSERVER:- (# vim index.php)

```
aws | Services | Search | Mumbai | Vinit Raj |
<pre>
<?php
  print r($SERVER); THIS COMMAND PRINTS THE DETAILS OF SERVER THE CLIENT IS CONNECTED
  ?> TO.
</pre> WE ARE USING THIS TO VERIFY THAT THE LOAD BALANCER IS USING
ROUND ROBIN ALGORITHM TO BALANCE CONNECTION REQUESTS.
```

→ NOW, IT'S TIME TO DEPLOY OUR SERVERS AND WEBPAGE:-

FIRST, I'LL DEPLOY BACKEND SERVER(S) :- (# ansible-playbook webserver.yml)

```
Chrome | EC2 Management | EC2 Instance Connect | Primer on Jinja Templating | +
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0286bcc31344ac2b7&osUser=ec2-user&region=ap-south-1&sshPort=22#
AWS | Services | Search | [Option+S] | Mumbai | Vinit Raj |

[root@ip-172-31-38-1 code]# date
Wed Jan 11 00:06:37 UTC 2023
[root@ip-172-31-38-1 code]# vim webserver.yml
[root@ip-172-31-38-1 code]# vim index.php
[No write since last change]
/bin/bash: q: command not found
shell returned 127

Press ENTER or type command to continue
[root@ip-172-31-38-1 code]# vim index.php
[root@ip-172-31-38-1 code]# ansible-playbook webserver.yml

PLAY [webserver] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 3.110.102.42 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [3.110.102.42]
[WARNING]: Platform linux on host 3.110.181.249 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [3.110.181.249]
TASK [Installing Web Server (HTTPD)] *****
changed: [3.110.181.249]
changed: [3.110.102.42]
TASK [Installing PHP package] *****
changed: [3.110.181.249]
changed: [3.110.102.42]
TASK [Deploying (copying) Web Page] *****
changed: [3.110.102.42]
changed: [3.110.181.249]
TASK [Starting Web Server Now] *****
changed: [3.110.181.249]
changed: [3.110.102.42]
PLAY RECAP *****
3.110.102.42 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
3.110.181.249 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[root@ip-172-31-38-1 code]#
```

→ NOW, WE'LL DEPLOY FRONTEND WEBSERVER WHICH ALSO ACTS AS LOAD BALANCER :- (# ansible-playbook lb.yml)

```
Chrome File Edit View History Bookmarks Profiles Tab Window Help
Instances | EC2 Management | EC2 Instance Connect | Primer on Jinja Templating - R | +
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0286bcc313a4ac2b7&osUser=ec2-user&region=ap-south-1&sshPort=22#
RHCSA - KodeKlo... (A) FreeCodeCamp St... Data Structures a... LeetCode 75 Stud... Strivers A2Z DSA... GFG Beginner DS... Striver's SDE She... GFG SDE Sheet... ARTH-3.0... AWS... Dev Roadmaps... Google's Coding...
AWS Services Search [Option+S] Mumbai Vinit Raj V
[root@ip-172-31-38-1 code]# vim lb.yml
[root@ip-172-31-38-1 code]# ansible-playbook lb.yml

PLAY [loadbalancer] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 13.235.0.177 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [13.235.0.177]

TASK [Installing HAProxy Load Balancer] *****
ok: [13.235.0.177]

TASK [Registering webserver by adding them in configuration file] *****
fatal: [13.235.0.177]: FAILED! => ("changed": false, "msg": "Could not find or access 'local.cfg.j2'\nSearched in:\n\t/root/code/templates/local.cfg.j2\n\t/root/code/local.cfg.j2\n\t/root/code/temp
lates/local.cfg.j2\n\t/root/code/local.cfg.j2\n on the Ansible Controller.\nIf you are using a module and expect the file to exist on the remote, see the remote_ext option")

PLAY RECAP *****
13.235.0.177 : ok=2 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0

[root@ip-172-31-38-1 code]# ls
index.php lb.yml local.cfg.j2 webserver.yml
[root@ip-172-31-38-1 code]# vim lb.yml
[root@ip-172-31-38-1 code]# ansible-playbook lb.yml

PLAY [loadbalancer] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 13.235.0.177 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [13.235.0.177]

TASK [Installing HAProxy Load Balancer] *****
ok: [13.235.0.177]

TASK [Registering webserver by adding them in configuration file] *****
changed: [13.235.0.177]

TASK [Starting the Load Balancer] *****
changed: [13.235.0.177]

PLAY RECAP *****
13.235.0.177 : ok=4 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[root@ip-172-31-38-1 code]#
```

→ NOW, TIME TO CHECK IF OUR LOAD BALANCER IS WORKING PROPERLY OR NOT :- (http://13.235.0.177:8080)  
IP ADDRESS OF LOAD-BALANCER

The screenshot shows a web browser window with a console log displaying network traffic. The log is a raw HTTP request from 13.235.0.177:8080 to 172.31.45.196:8080. A handwritten orange note with an arrow points to the `[SERVER_ADDR] => 172.31.45.196` line, stating: "FIRST CONNECTION REQUEST LANDED TO SERVER HAVING IP ADDR: 172.31.45.196".

```
[
  {
    [UNIQUE_ID] => Y740qqY0y2emi95u0tdQ3AAAAAQ
    [HTTP_HOST] => 13.235.0.177:8080
    [HTTP_CONNECTION] => keep-alive
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
    [HTTP_PURPOSE] => prefetch
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_ACCEPT_LANGUAGE] => en-IN,en;q=0.9,hi-IN;q=0.8,es-GB;q=0.6,en-US;q=0.5
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.54 () PHP/5.4.16
    [SERVER_NAME] => 13.235.0.177
    [SERVER_ADDR] => 172.31.45.196
    [SERVER_PORT] => 8080
    [REMOTE_ADDR] => 13.235.0.177
    [DOCUMENT_ROOT] => /var/www/html
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /var/www/html
    [SERVER_ADMIN] => root@localhost
    [SCRIPT_FILENAME] => /var/www/html/index.php
    [REMOTE_PORT] => 52436
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /
    [SCRIPT_NAME] => /index.php
    [PHP_SELF] => /index.php
    [REQUEST_TIME_FLOAT] => 1673399938.296
    [REQUEST_TIME] => 1673399938
  }
]
```

- NOW, WE'LL GO THE SAME URL FROM ANOTHER BROWSER. (PREFERABLY FROM INCOGNITO MODE).



```
Array
(
    [UNIQUE_ID] => Y74QH910e1ET4ThdmfOR7QAAAAI
    [HTTP_HOST] => 13.235.0.177:8080
    [HTTP_CONNECTION] => keep-alive
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_ACCEPT_LANGUAGE] => en-IN,en;q=0.9
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.54 () PHP/5.4.16
    [SERVER_NAME] => 13.235.0.177
    [SERVER_ADDR] => 172.31.32.130
    [SERVER_PORT] => 8080
    [REMOTE_ADDR] => 13.235.0.177
    [DOCUMENT_ROOT] => /var/www/html
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /var/www/html
    [SERVER_ADMIN] => root@localhost
    [SCRIPT_FILENAME] => /var/www/html/index.php
    [REMOTE_PORT] => 56674
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /
    [SCRIPT_NAME] => /index.php
    [PHP_SELF] => /index.php
    [REQUEST_TIME_FLOAT] => 1673400351.628
    [REQUEST_TIME] => 1673400351
)
```

● NOW IF I REFRESH THIS PAGE, THE NEXT CONNECTION SHOULD BE SENT TO THE OTHER SERVER.

```
Array
(
    [UNIQUE_ID] => Y74RMjeeRBIdLuvzESTXAAAAAI
    [HTTP_HOST] => 13.235.0.177:8080
    [HTTP_CONNECTION] => keep-alive
    [HTTP_CACHE_CONTROL] => max-age=0
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_ACCEPT_LANGUAGE] => en-IN,en;q=0.9
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.54 () PHP/5.4.16
    [SERVER_NAME] => 13.235.0.177
    [SERVER_ADDR] => 172.31.45.196
    [SERVER_PORT] => 8080
    [REMOTE_ADDR] => 13.235.0.177
    [DOCUMENT_ROOT] => /var/www/html
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /var/www/html
    [SERVER_ADMIN] => root@localhost
    [SCRIPT_FILENAME] => /var/www/html/index.php
    [REMOTE_PORT] => 36918
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /
    [SCRIPT_NAME] => /index.php
    [PHP_SELF] => /index.php
    [REQUEST_TIME_FLOAT] => 1673400626.945
    [REQUEST_TIME] => 1673400626
)
```

THIS IS ENOUGH TO PROVE THAT OUR LOAD BALANCER IS WORKING EXACTLY THE WAY IT SHOULD!

★ NOW, TO PERFORM HORIZONTAL SCALING, ALL WE NEED TO DO IS:

WE JUST NEED TO ADD THE DETAILS OF OUR NEW NODE AND THEN RE-RUN THE ANSIBLE-PLAYBOOK COMMAND.



WE HAVE WRITTEN OUR CODE IN SUCH A WAY THAT THE SERVER WILL GET ADDED IN THE LIST JUST BY RE-RUNNING 'ONE SINGLE COMMAND'.

(# vim /etc/ansible/hosts)

```
# This is the default ansible 'hosts' file.
[loadbalancer]
13.235.0.177 ansible_user=root ansible_password=vinitraj

[webserver]
3.110.102.42 ansible_user=root ansible_password=vinitraj
3.110.181.249 ansible_user=root ansible_password=vinitraj
65.2.182.99 ansible_user=root ansible_password=vinitraj
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
```

WE HAVE ADDED OUR NEW HOST IN WEBSERVER GROUP.

NOW JINJA TEMPLATE FILE WILL AUTOMATICALLY ADD THIS NODE IN BACKEND WEBSERVERS LIST.

HORIZONTAL SCALING IS DONE!

→ TIME TO RE-RUN THE PLAYBOOK COMMANDS :-

(# ansible-playbook webserver.yml)

(# ansible-playbook lb.yml)

```
ok: [3.110.181.249]
changed: [65.2.182.99]

TASK [Installing PHP package] *****
[[Ank: [3.110.182.42]
ok: [3.110.181.249]
changed: [65.2.182.99]

TASK [Deploying (copying) Web Page] *****
ok: [3.110.181.249]
ok: [3.110.102.42]
changed: [65.2.182.99]

TASK [Starting Web Server Now] *****
ok: [3.110.181.249]
ok: [3.110.102.42]
changed: [65.2.182.99]

PLAY RECAP *****
3.110.102.42      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
3.110.181.249    : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
65.2.182.99      : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ip-172-31-38-1 code]# ansible-playbook lb.yml

PLAY [loadbalancer] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 13.235.0.177 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [13.235.0.177]

TASK [Installing HAProxy Load Balancer] *****
ok: [13.235.0.177]

TASK [Registering web servers by adding them in configuration file] *****
changed: [13.235.0.177]

TASK [Starting the Load Balancer] *****
changed: [13.235.0.177]

PLAY RECAP *****
13.235.0.177      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

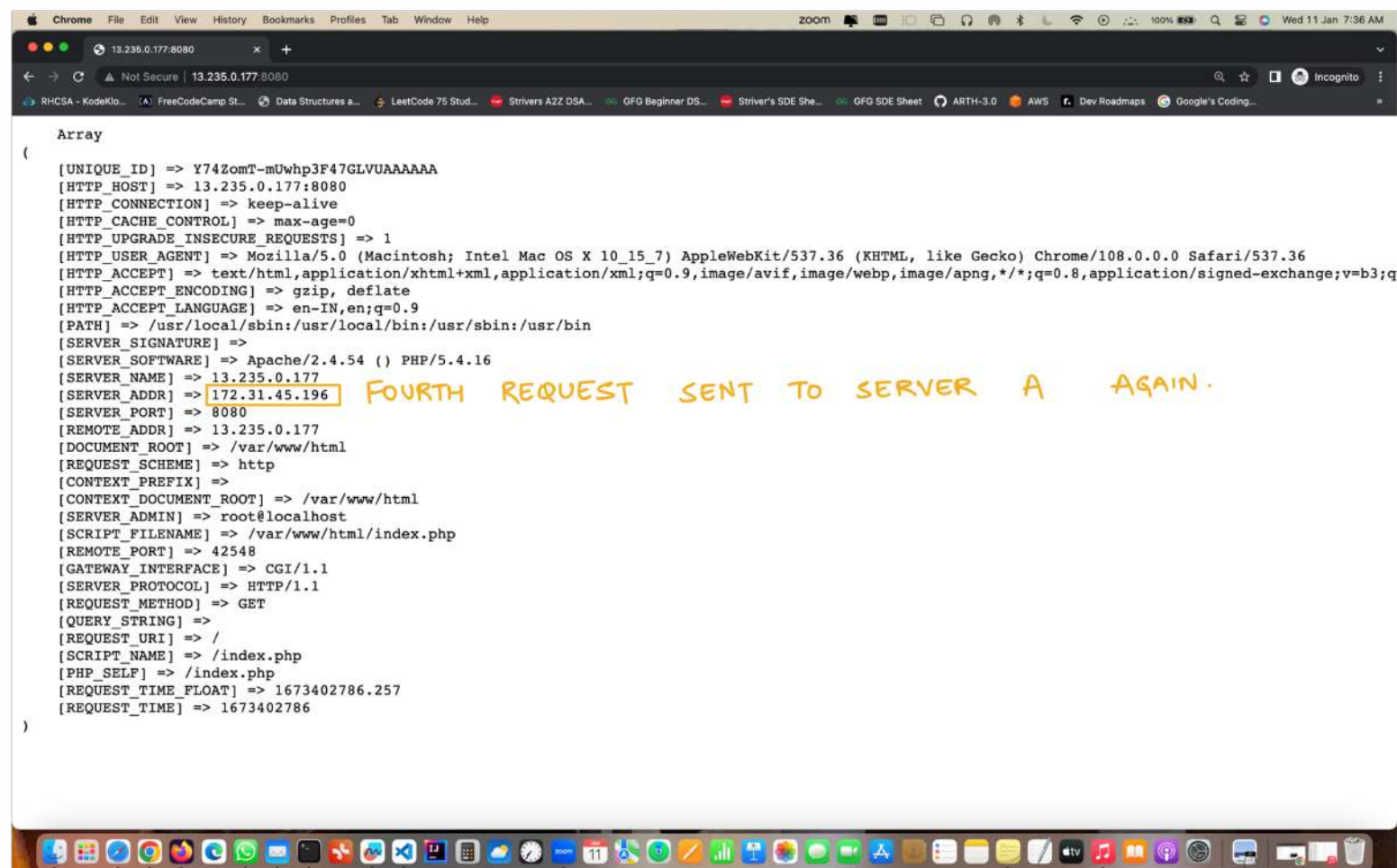
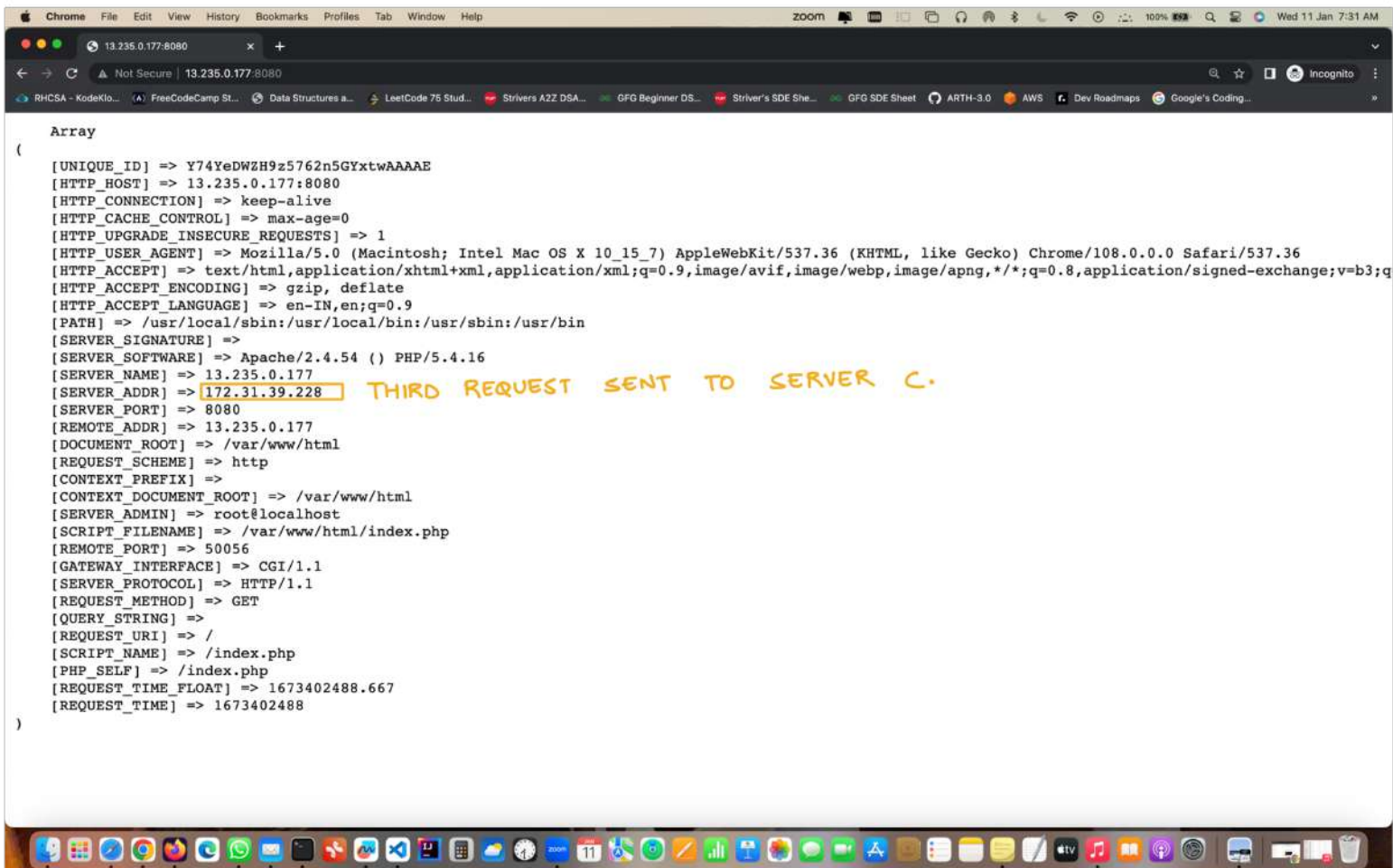
[root@ip-172-31-38-1 code]#
```



→ TIME TO VISIT OUR WEBSITE (LOADBALANCER IP :8080) :-

```
Array
(
  [UNIQUE_ID] => Y74YWTeerBIBdLuvZE5TXQAAAAI
  [HTTP_HOST] => 13.235.0.177:8080
  [HTTP_CONNECTION] => keep-alive
  [HTTP_CACHE_CONTROL] => max-age=0
  [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
  [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
  [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
  [HTTP_ACCEPT_ENCODING] => gzip, deflate
  [HTTP_ACCEPT_LANGUAGE] => en-IN,en;q=0.9
  [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
  [SERVER_SIGNATURE] =>
  [SERVER_SOFTWARE] => Apache/2.4.54 () PHP/5.4.16
  [SERVER_NAME] => 13.235.0.177
  [SERVER_ADDR] => 172.31.45.196
  [SERVER_PORT] => 8080
  [REMOTE_ADDR] => 13.235.0.177
  [DOCUMENT_ROOT] => /var/www/html
  [REQUEST_SCHEME] => http
  [CONTEXT_PREFIX] =>
  [CONTEXT_DOCUMENT_ROOT] => /var/www/html
  [SERVER_ADMIN] => root@localhost
  [SCRIPT_FILENAME] => /var/www/html/index.php
  [REMOTE_PORT] => 38656
  [GATEWAY_INTERFACE] => CGI/1.1
  [SERVER_PROTOCOL] => HTTP/1.1
  [REQUEST_METHOD] => GET
  [QUERY_STRING] =>
  [REQUEST_URI] => /
  [SCRIPT_NAME] => /index.php
  [PHP_SELF] => /index.php
  [REQUEST_TIME_FLOAT] => 1673402457.588
  [REQUEST_TIME] => 1673402457
)
```

```
Array
(
  [UNIQUE_ID] => Y74Ybd10e1ET4ThDmfOR7gAAAAI
  [HTTP_HOST] => 13.235.0.177:8080
  [HTTP_CONNECTION] => keep-alive
  [HTTP_CACHE_CONTROL] => max-age=0
  [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
  [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
  [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
  [HTTP_ACCEPT_ENCODING] => gzip, deflate
  [HTTP_ACCEPT_LANGUAGE] => en-IN,en;q=0.9
  [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
  [SERVER_SIGNATURE] =>
  [SERVER_SOFTWARE] => Apache/2.4.54 () PHP/5.4.16
  [SERVER_NAME] => 13.235.0.177
  [SERVER_ADDR] => 172.31.32.130
  [SERVER_PORT] => 8080
  [REMOTE_ADDR] => 13.235.0.177
  [DOCUMENT_ROOT] => /var/www/html
  [REQUEST_SCHEME] => http
  [CONTEXT_PREFIX] =>
  [CONTEXT_DOCUMENT_ROOT] => /var/www/html
  [SERVER_ADMIN] => root@localhost
  [SCRIPT_FILENAME] => /var/www/html/index.php
  [REMOTE_PORT] => 53346
  [GATEWAY_INTERFACE] => CGI/1.1
  [SERVER_PROTOCOL] => HTTP/1.1
  [REQUEST_METHOD] => GET
  [QUERY_STRING] =>
  [REQUEST_URI] => /
  [SCRIPT_NAME] => /index.php
  [PHP_SELF] => /index.php
  [REQUEST_TIME_FLOAT] => 1673402477.775
  [REQUEST_TIME] => 1673402477
)
```





THIS PROVES OUR WEBSERVERS ARE WORKING  
EXACTLY THE WAY THEY SHOULD !

ROUND ROBIN ALGORITHM IS BALANCING THE  
CONNECTION REQUESTS CORRECTLY.

ADDING A BACKEND WEBSERVER AT THE  
END AND GETTING IT TO WORK PROPERLY :  
SUCCESSFULLY IMPLEMENTED HORIZONTAL  
SCALING .

—VINIT RAJ