### **Load Balancing HAProxy Setup On AWS Using Ansible ..!!**

Hello World:!?<>!!

In THis Article, I Will Explain yOu How we Can SeTup Haproxy Setup On AWS Using Ansible



### **Task DescripTion :-**

♦️ Provision EC2 Instances Through Ansible.

♦️ Retrieve The IP Address of Instances Using The Dynamic Inventory Concept.

♦️ Configure The Web Servers Through The Ansible.

♦️ The Target Nodes of the Load Balancer Should Auto-Update As Per The Status of Web Servers.



**Ansible** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) software [provisioning](https://en.wikipedia.org/wiki/Provisioning), configuration management, and [application-deployment](https://en.wikipedia.org/wiki/Application_deployment) tool enabling [infrastructure as code](https://en.wikipedia.org/wiki/Infrastructure_as_code). It runs on many [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems, and can configure both Unix-like systems as well as [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). It includes its own [declarative language](https://en.wikipedia.org/wiki/Declarative_language) to describe [system configuration](https://en.wikipedia.org/wiki/System_configuration). Ansible was written by Michael DeHaan and acquired by [Red Hat](https://en.wikipedia.org/wiki/Red_Hat) in 2015. Ansible is [agentless](https://en.wikipedia.org/wiki/Agentless), temporarily connecting remotely via [SSH](https://en.wikipedia.org/wiki/SSH) or [Windows Remote Management](https://en.wikipedia.org/wiki/Windows_Remote_Management) (allowing remote [PowerShell](https://en.wikipedia.org/wiki/PowerShell) execution) to do its tasks.



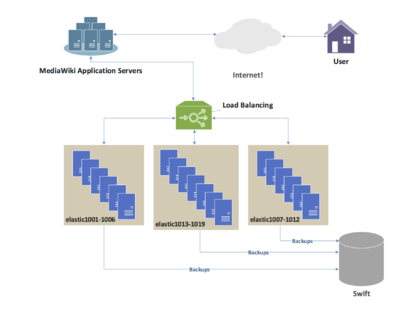
**Red Hat Linux**, created by the company [Red Hat](https://en.wikipedia.org/wiki/Red_Hat), was a widely used [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution) until its discontinuation in 2004. Early releases of Red Hat Linux were called **Red Hat Commercial Linux**. Red Hat published the first non-beta release in May 1995. It was the first Linux distribution to use the [RPM Package Manager](https://en.wikipedia.org/wiki/RPM_Package_Manager) as its packaging format, and over time has served as the starting point for several other distributions, such as [Mandriva Linux](https://en.wikipedia.org/wiki/Mandriva_Linux) and [Yellow Dog Linux](https://en.wikipedia.org/wiki/Yellow_Dog_Linux). In 2003, Red Hat discontinued the Red Hat Linux line in favor of [Red Hat Enterprise Linux](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux) (RHEL) for enterprise environments. [Fedora](https://en.wikipedia.org/wiki/Fedora_%28operating_system%29), developed by the community-supported [Fedora Project](https://en.wikipedia.org/wiki/Fedora_Project) and sponsored by Red Hat, is a free-of-cost alternative intended for home use. Red Hat Linux 9, the final release, hit its official end-of-life on April 30, 2004, although updates were published for it through 2006 by the [Fedora Legacy](https://en.wikipedia.org/wiki/Fedora_Legacy) project until that shut down in early 2007.



**Amazon Web Services** (**AWS**) is a subsidiary of [Amazon](https://en.wikipedia.org/wiki/Amazon.com) providing [on-demand](https://en.wikipedia.org/wiki/Software_as_a_service) [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) [platforms](https://en.wikipedia.org/wiki/Computing_platform) and [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing [web services](https://en.wikipedia.org/wiki/Web_services) provide a variety of basic abstract technical infrastructure and [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing) building blocks and tools. One of these services is [Amazon Elastic Compute Cloud](https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud) (EC2), which allows users to have at their disposal a [virtual](https://en.wikipedia.org/wiki/Virtualization) [cluster of computers](https://en.wikipedia.org/wiki/Computer_cluster), available all the time, through the Internet. AWS’s version of virtual computers emulates most of the attributes of a real computer, including hardware [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) and [graphics processing units](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPUs) for processing; local/[RAM](https://en.wikipedia.org/wiki/Random-access_memory) memory; hard-disk/[SSD storage](https://en.wikipedia.org/wiki/Solid-state_drive); a choice of operating systems; networking; and pre-loaded application software such as [web servers](https://en.wikipedia.org/wiki/Web_server), [databases](https://en.wikipedia.org/wiki/Database), and [customer relationship management](https://en.wikipedia.org/wiki/Customer_relationship_management) (CRM).



**HAProxy** is [free, open source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) that provides a [high availability](https://en.wikipedia.org/wiki/High_availability) [load balancer](https://en.wikipedia.org/wiki/Load_balancing_%28computing%29) and [proxy server](https://en.wikipedia.org/wiki/Proxy_server) for [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)-based applications that spreads requests across multiple servers. It is written in and has a reputation for being fast and efficient (in terms of processor and memory usage). HAProxy is used by a number of high-profile websites including [GoDaddy](https://en.wikipedia.org/wiki/GoDaddy), [GitHub](https://en.wikipedia.org/wiki/GitHub), [Bitbucket](https://en.wikipedia.org/wiki/Bitbucket), [Stack Overflow](https://en.wikipedia.org/wiki/Stack_Overflow), [Reddit](https://en.wikipedia.org/wiki/Reddit), [Slack](https://en.wikipedia.org/wiki/Slack_%28software%29), [Speedtest.net](https://en.wikipedia.org/wiki/Speedtest.net), [Tumblr](https://en.wikipedia.org/wiki/Tumblr), [Twitter](https://en.wikipedia.org/wiki/Twitter) and [Tuenti](https://en.wikipedia.org/wiki/Tuenti) and is used in the [OpsWorks](https://en.wikipedia.org/wiki/AWS_OpsWorks) product from [Amazon Web Services](https://en.wikipedia.org/wiki/Amazon_Web_Services).



In [computing](https://en.wikipedia.org/wiki/Computing), **load balancing** refers to the process of distributing a set of [tasks](https://en.wikipedia.org/wiki/Task_%28computing%29) over a set of [resources](https://en.wikipedia.org/wiki/System_resource) (computing units), with the aim of making their overall processing more efficient. Load balancing techniques can optimize the response time for each task, avoiding unevenly overloading compute nodes while other compute nodes are left idle. Load balancing is the subject of research in the field of [parallel computers](https://en.wikipedia.org/wiki/Parallel_computers). Two main approaches exist: static algorithms, which do not take into account the state of the different machines, and dynamic algorithms, which are usually more general and more efficient, but require exchanges of information between the different computing units, at the risk of a loss of efficiency.

### **Pre-requisites :-**

1. AWS AccounT .
2. IAM User with Admin Access in this aws account .
3. RedHaT Linux must be Installed .
4. Ansible Should Be Installed in This RedHat Linux VM (OS)
5. And Some More that i will be explaining in Between .

FirsT Of All For CreaTing This SeTup we need to launch 2 or 3 instances on aws so that we can make one instance as load balancer and others as webserver .

So For Launching Instance we have 2 way’s we can launch instances on aws using ansible or manually .

Here is the code →<https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae>

**[Configuring WebServer On AWS Instance Using Ansible](https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae)**

[*Heyy Guy’s ..!!*medium.com](https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae)

**Awsec2.yml**

- hosts: localhost

vars\_files:

— credential.yml

tasks:

— name: Launching AWS Instance

ec2:

key\_name: keynew

instance\_type: t2.micro

image: “ami-0ebc1ac48dfd14136”

wait: yes

count: 3

aws\_access\_key: “{{ ansibleuser }}”

aws\_secret\_key: “{{ ansipass }}”

vpc\_subnet\_id: subnet-82c7fdea

region: “ap-south-1”

group\_id: “sg-9e0c4dfc”

assign\_public\_ip: yes

state: present

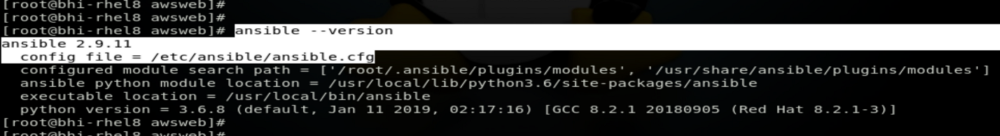
**Credential.yml**

ansibleuser: AWS Access Key Here

ansipass: AWS Secret Key Here

Now we are checking ansible version

ansible --version



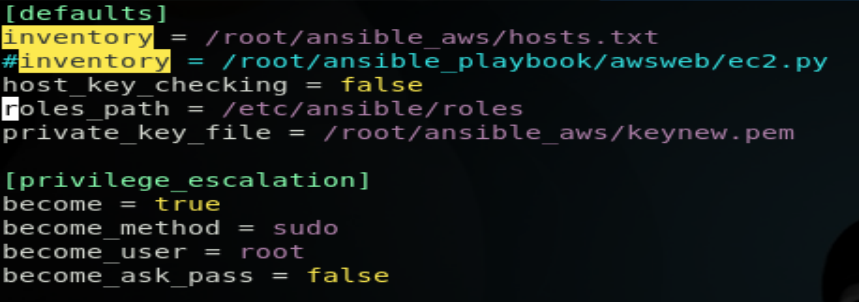
Don’T worry if u dont know what is ansible and how to install ansible in RedHat Linux Go Through this Article i have already Explained Everything From very basic →<https://medium.com/@sabhi8226/ansible-installation-in-rhel-8-be24c9e83baf>

**[Ansible Installation In RHEL 8](https://medium.com/@sabhi8226/ansible-installation-in-rhel-8-be24c9e83baf)**

[*COnnecT with me On Linkedin FOr FurThur Queries and Feel Free tO Give Any Suggestions if u Have !!*medium.com](https://medium.com/@sabhi8226/ansible-installation-in-rhel-8-be24c9e83baf)

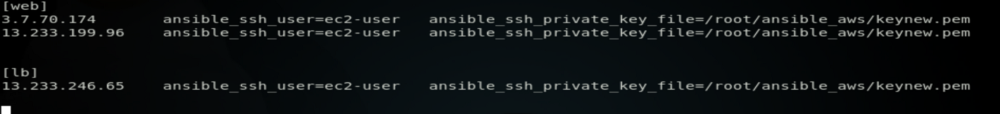
In Above picture we can see that ansible configuration file is is available at /etc/ansible/ansible.cfg .

vim /etc/ansible/ansible.cfg



and from here we get to know that ansible host file we have created at /root/ansible\_aws/hosts.txt

vim /root/ansible\_aws/hosts.txt



Now yOu Are thinking that From where we get this ip and all Chill ..

LeT me Explain you this part sO …

As We Have Launched Instances On AWS For Getting their IP Again we have 2 ways one is manual and 2nd is Using Ansible Dynamc Inventory Concept .

BuT For using Ansible Dynamic Inventory ConcepT Their is Pre-requisites for this →<https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae>

**[Configuring WebServer On AWS Instance Using Ansible](https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae)**

[*Heyy Guy’s ..!!*medium.com](https://medium.com/@sabhi8226/configuring-webserver-on-aws-instance-using-ansible-f507c238a4ae)



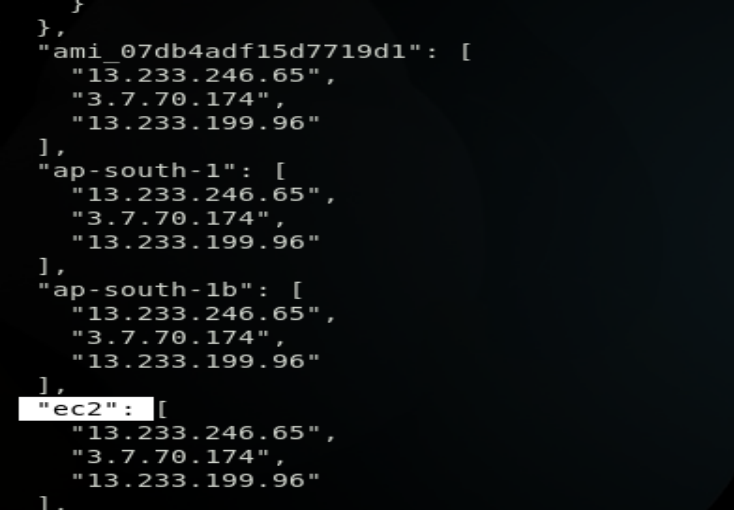
right now we need this 2 files **ec2.ini** and **ec2.py**

For Getting IP Using Dynamic Inventory Concept we have to run this command that will do everything for us .

./ec2.py --list



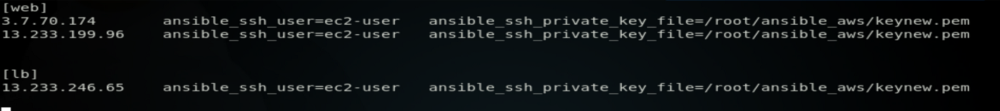
Here they collect too much information for us like ami id they used to launch the instance and region and ip etc .



I Hope Now you can related with above picture of hosts.txt file

let me explain its every line ..

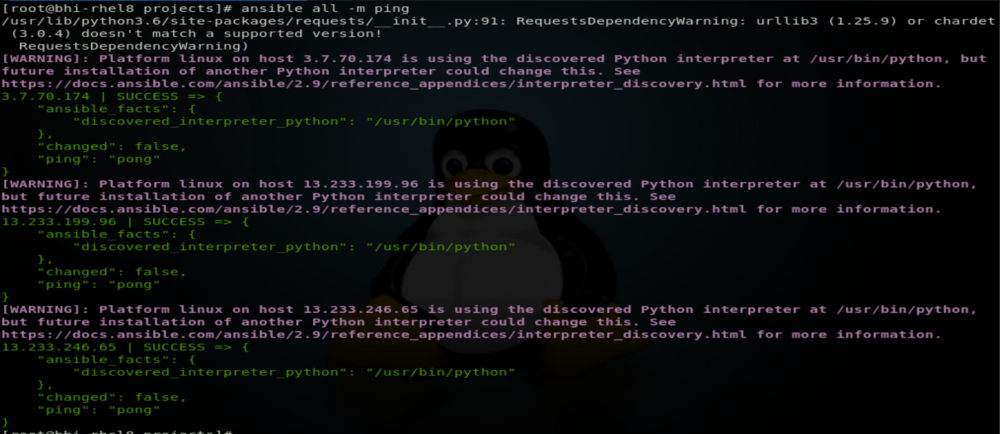
In This PicTure Shown Below In First two lines here i have given details for webserver instance means the ansible will go and configure these 2 ip’s for webserver and the below one with load balancer . and there are 2 vaiables name are given i.e. annsible\_ssh\_user which is ec2-user because the user we get bydefault when we launch instance on aws and ansible\_ssh\_private\_key → its for giving path and name of private key that we have attached to the instance at the time of launching .



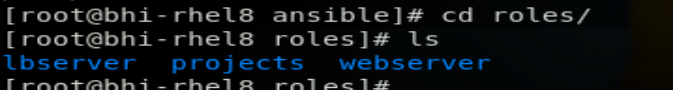
Now Let Go Ahead In Our Task..

So For This It’s GooD PracTice tO Check Connectivity Between the AWS Instance and Controller Node . (i.e. Ping ip)

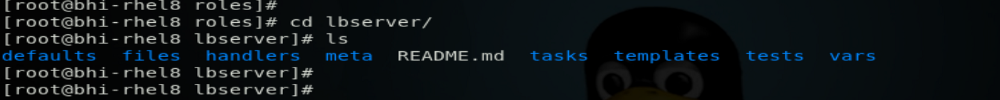
ansible all -m ping



FirsT of all we have created 2 roles one is for load balancer and one is for webserver .

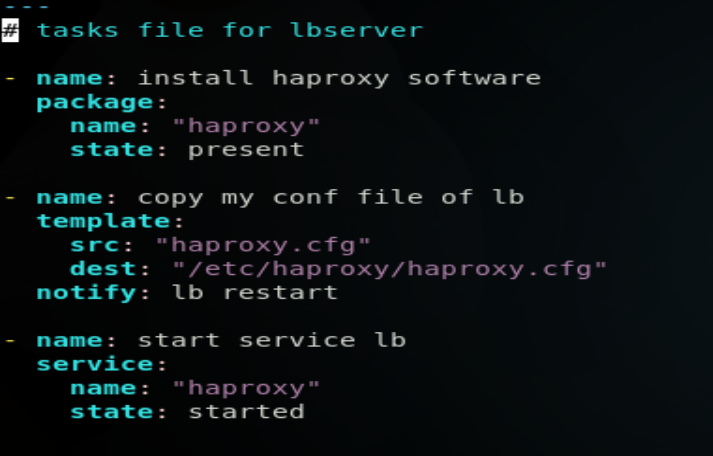


cd /lbserver

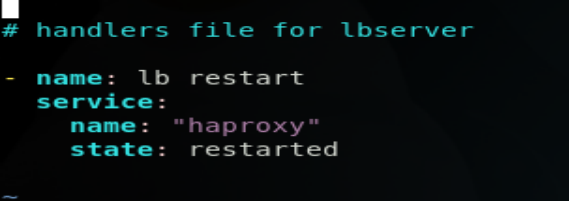


vim /task/main.yml

This file will configure instance as load balancer

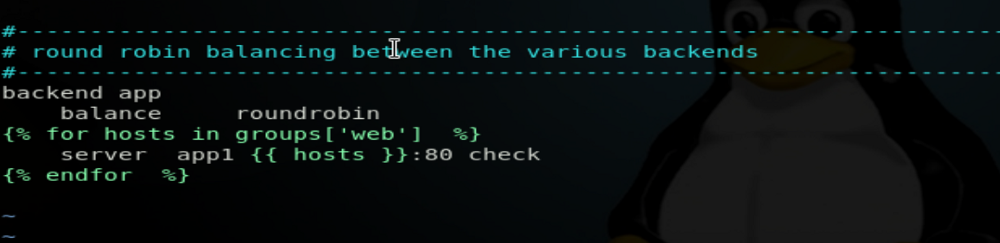


vim /handler/main.yml



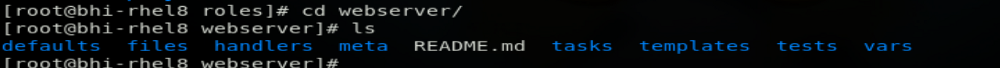
vim /template/haproxy.cfg

Here we have to change some lines of code at the last of the haproxy configuration file .



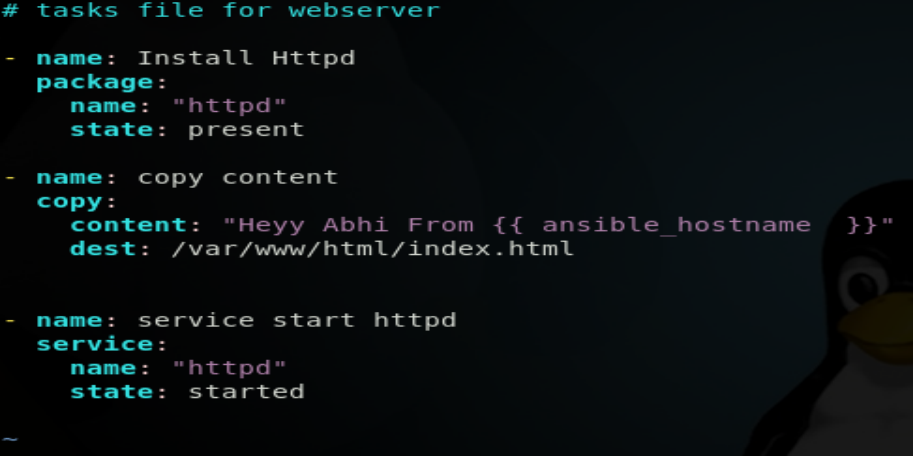
Now let go inside other role i.e. webserver

cd /webserver

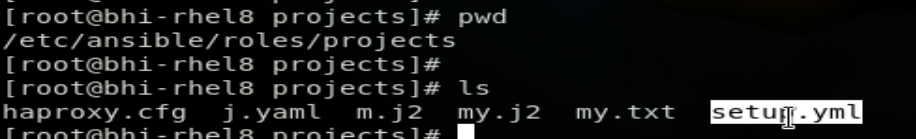


vim /task/main.yml

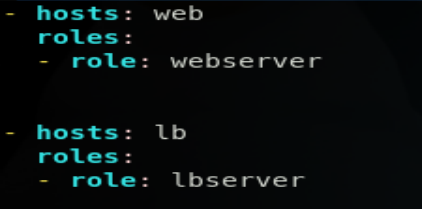
This file will configure other instance as webserver



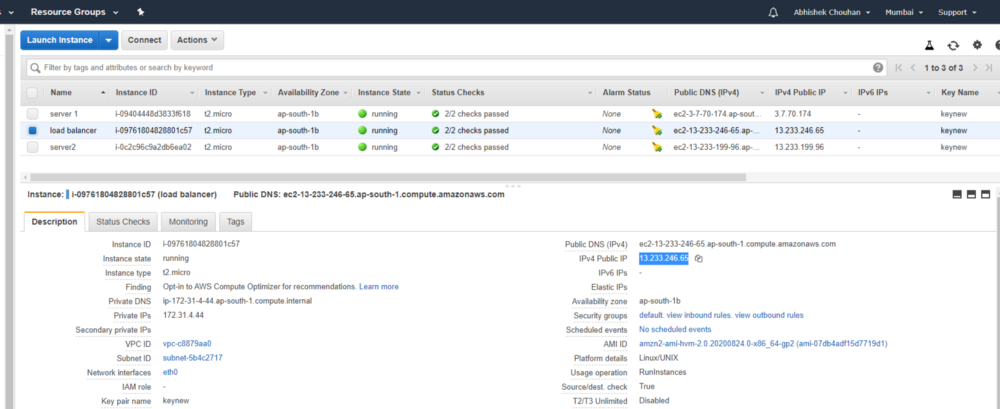
Here we have our main file which will run both roles to create this setup of haproxy with load balacing .



vim setup.yml



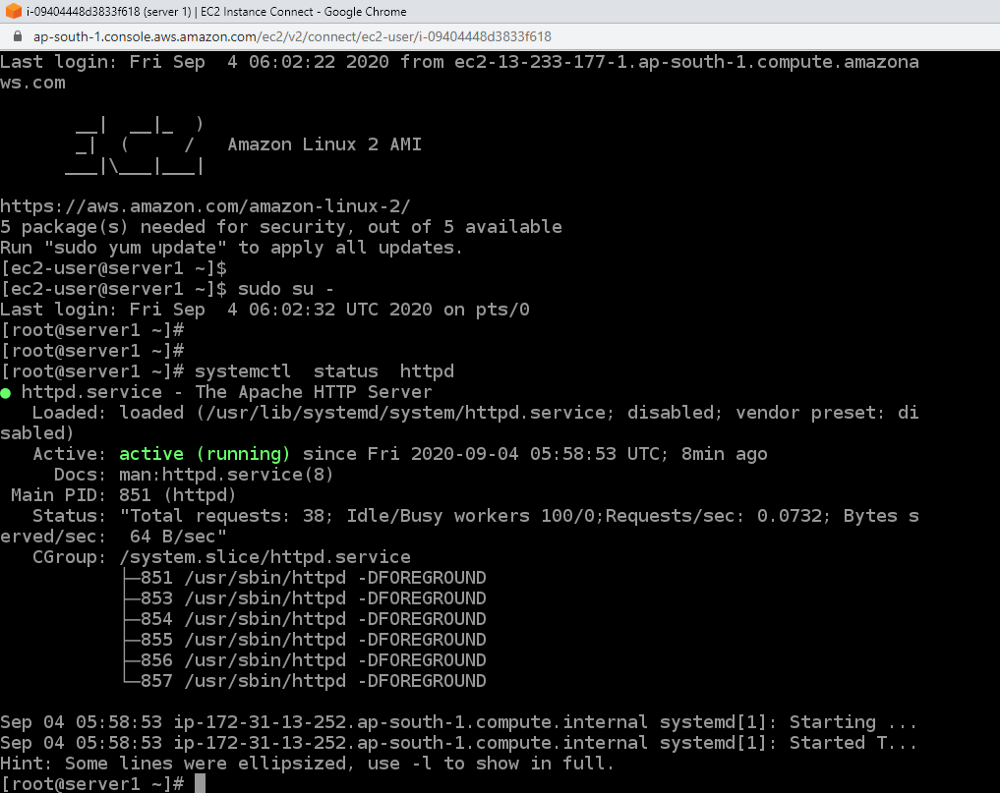
Instance that we launched is here and now they are configured according to our requirement .



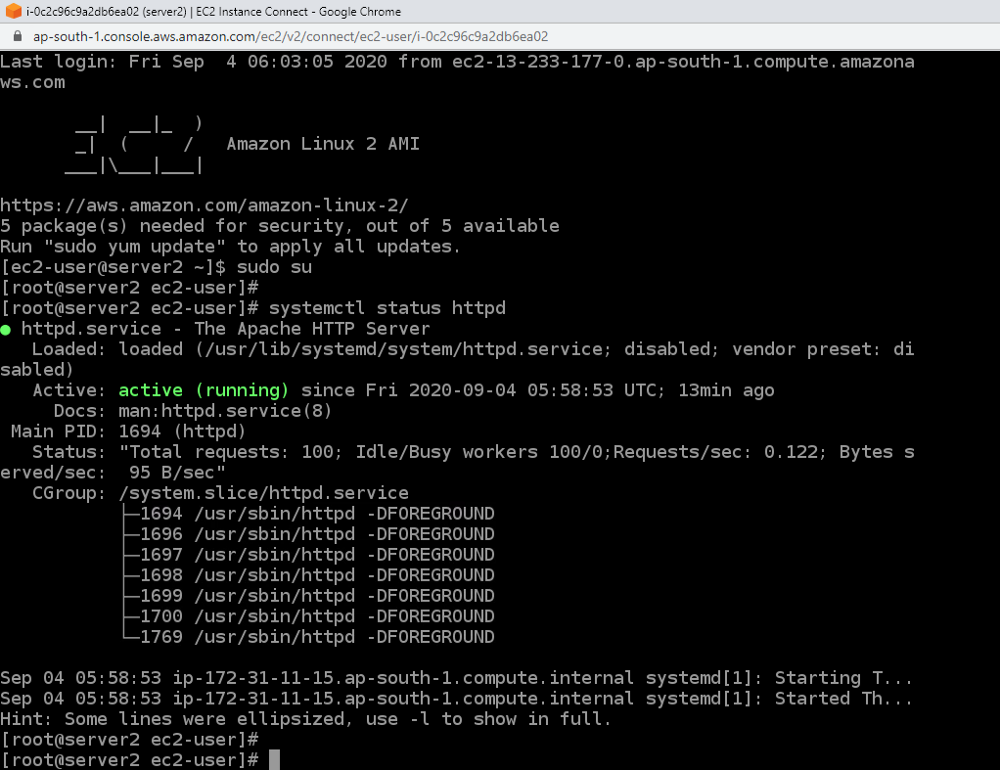
We can also check by going inside them

webserver :-

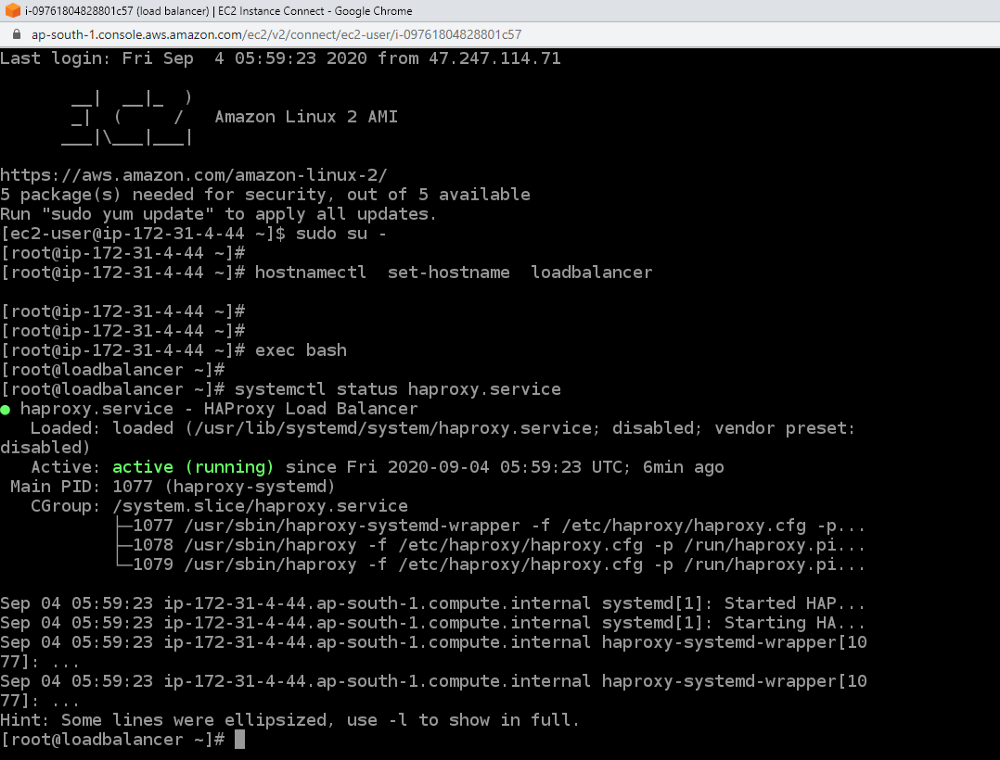
server1



server 2

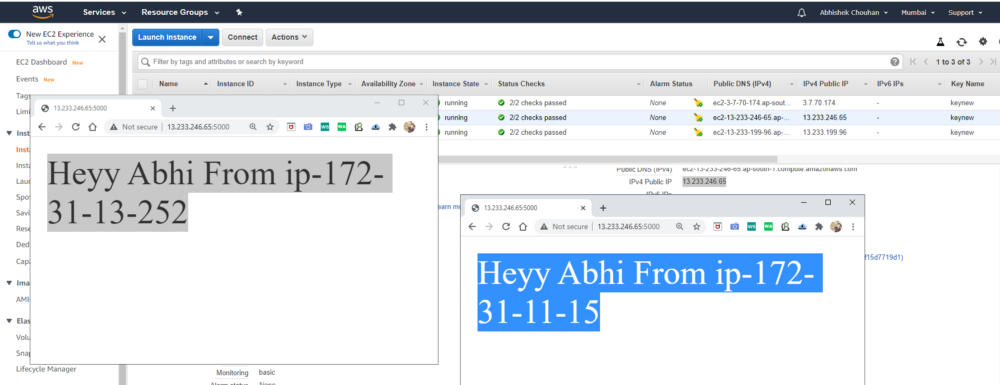
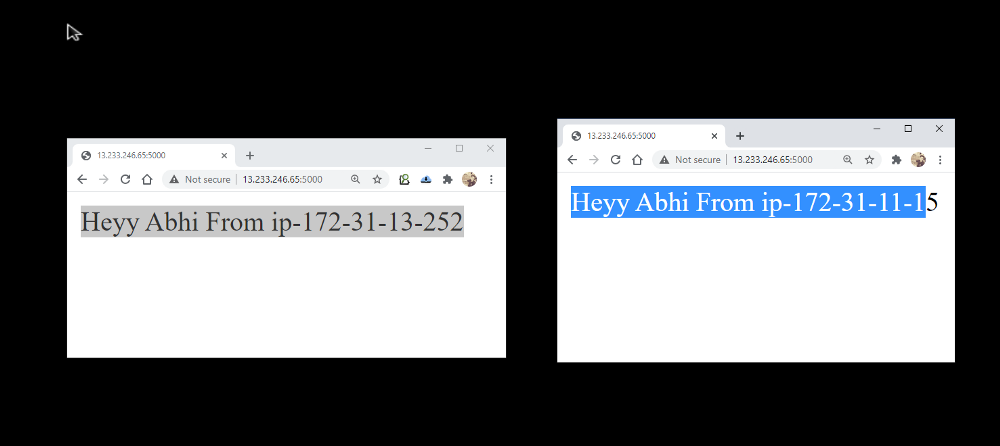


Loadbalancer :-



Now we will PuT Ip Of Loadbalancer Instance to see real time load balancing haapening ..

13.233.246.65



yOu can see real time load balancing that and when u refresh the page it will again change the IP . and same output will come for client .