Deploy HAProxy LoadBalancer in AWS using Ansible

In this article, we are going to deploy HA-Proxy LoadBalancer on EC2-instance using Ansible.

Why Ansible…?

Ansible automates and simplifies repetitive, complex, and tedious operations. Everybody likes it because it brings huge time savings when **we** install packages or configure large numbers of servers.

I create Ansible-Roles for launching EC2 instance and deploy webserver and LoadBalancer Service using the Dynamic-Inventory concept.

If you do not know how to set up Dynamic inventory for AWS, please refer to my previous article.

https://www.linkedin.com/pulse/deploy-apache-web-server-using-aws-dynamic-inventory-anudeep-nalla/

So Let’s Start…

For this task, I’m creating three Ansible-roles…

1. For Launch AWS EC2 instances.
2. For launching Apache Webserver
3. And one more for HA-Proxy LoadBalancer

To create an Ansible-role, First create a directory **/etc/ansible/roles** and after that run command…

ansible-galaxy init role_name

1. Launch AWS EC2 instance:-

Code for launching the EC2 instance is below, I’m created 3 hosts for Webserver and 1 for LoadBalancer…

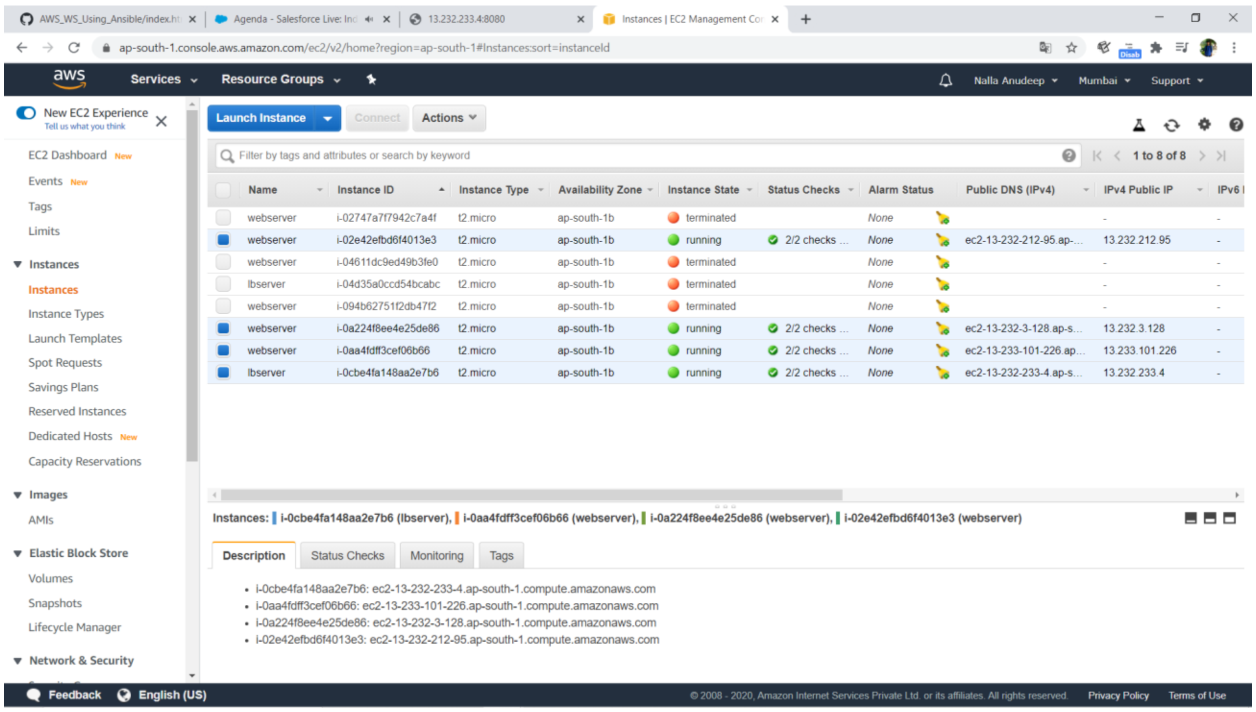
```
# tasks file for webserver
                          - name: Create Key Pair
                            ec2_key:
                                name: mykey15
                                aws_region: "{{ region }}"
                            register: ec2_key
                          - name: Copy Key to Local File
                            copy:
                                content: "{{ ec2_key.key.private_key }}"
                                dest: "{{ key_dest }}"
                                mode: '0600'
                          - name: Create Security Group - Allow SSh, HTTP
                            ec2_group:
                                name: sg_ansible_web
                                description: sg for web inventory
                                region: "{{ region }}"
                                rules:
                                - proto: tcp
                                  from_port: 80
                                  to_port: 80
                                  cidr_ip: 0.0.0.0/0
                                - proto: tcp
                                  from_port: 22
                                  to_port: 22
                                  cidr_ip: 0.0.0.0/0
                                rules_egress:
                                - proto: all
                                  cidr_ip: 0.0.0.0/0
                            register: sg_ansible_web
                          - name: Create Security Group - Allow SSh, HAProxy
                            ec2_group:
                                name: sg_ansible_lb
                                description: sg for lb inventory
                                region: "{{ region }}"
                                rules:
                                - proto: tcp
                                  from_port: 8080
```

```
          to_port: 8080
          cidr_ip: 0.0.0.0/0
        - proto: tcp
          from_port: 22
          to_port: 22
          cidr_ip: 0.0.0.0/0
        rules_egress:
        - proto: all
          cidr_ip: 0.0.0.0/0
    register: sg_ansible_lb
  - name: Launch EC2 Instance for webserver
    ec2:
        key_name: mykey15
        instance_type: t2.micro
        image: "{{ image_id }}"
        wait: yes
        region: "{{ region }}"
        count: 3
        vpc_subnet_id: subnet-86bed5ca
        group_id: "{{ sg_ansible_web.group_id }}"
        assign_public_ip: yes
        state: present
        instance_tags:
          Name: webserver
    register: web
  - name: Launch EC2 Instance for lbserver
    ec2:
        key_name: mykey15
        instance_type: t2.micro
        image: "{{ image_id }}"
        wait: yes
        region: "{{ region }}"
        count: 1
        vpc_subnet_id: subnet-86bed5ca
        group_id: "{{ sg_ansible_lb.group_id }}"
        assign_public_ip: yes
        state: present
        instance_tags:
          Name: lbserver
    register: lb
  - name: Refresh Inventory File
    meta: refresh_inventory
  - pause:
```

```
                              minutes: 2
---
      # tasks file for web

      - name: Install Required Package
        package:
          name: python3
          state: present
        become: true
      - name: Install Apache Server
        package:
          name: httpd
          state: present
        become: true
      - name: copy web page from url
        get_url:
          dest: "/var/www/html"
          url:
      “https://raw.githubusercontent.com/Anuddeeph/AWS_WS_Using_Ansible/master/index.html"
        become: true
      - name: Start Apache Service
        service:
          name: httpd
          state: started
```

By Default, Ansible does not refresh the inventory in the middle of the running playbook, so we use the meta keyword **refresh_inventory** in the last of this code.

So, Ec2-Instance is launched…



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6
	webserver	i-02747a7f7942c7a4f	t2.micro	ap-south-1b	terminated		None		-	-
✓	webserver	i-02e42efbd6f4013e3	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-232-212-95.ap-...	13.232.212.95	-
	webserver	i-04611dc9ed49b3fe0	t2.micro	ap-south-1b	terminated		None		-	-
	lbserver	i-04d35a0ccd54bcabc	t2.micro	ap-south-1b	terminated		None		-	-
	webserver	i-094b62751f2db47f2	t2.micro	ap-south-1b	terminated		None		-	-
✓	webserver	i-0a224f8ee4e25de86	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-232-3-128.ap-s...	13.232.3.128	-
✓	webserver	i-0aa4fdff3cef06b66	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-233-101-226.ap...	13.233.101.226	-
✓	lbserver	i-0cbe4fa148aa2e7b6	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-232-233-4.ap-s...	13.232.233.4	-

Instances: i-0cbe4fa148aa2e7b6 (lbserver), i-0aa4fdff3cef06b66 (webserver), i-0a224f8ee4e25de86 (webserver), i-02e42efbd6f4013e3 (webserver)

- i-0cbe4fa148aa2e7b6: ec2-13-232-233-4.ap-south-1.compute.amazonaws.com
- i-0aa4fdff3cef06b66: ec2-13-233-101-226.ap-south-1.compute.amazonaws.com
- i-0a224f8ee4e25de86: ec2-13-232-3-128.ap-south-1.compute.amazonaws.com
- i-02e42efbd6f4013e3: ec2-13-232-212-95.ap-south-1.compute.amazonaws.com

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

2. Launch Webserver On EC2 instance:

Now I need to install httpd software on EC2 instance named webserver using Ansible

```
--
-
  # tasks file for web

  - name: Install Required Package
    package:
      name: python3
      state: present
    become: true
  - name: Install Apache Server
    package:
      name: httpd
      state: present
    become: true
  - name: copy web page from url
```

```
    get_url:
      dest: "/var/www/html"
      url: "
  https://raw.githubusercontent.com/Anuddeeph/AWS_WS_Using_Ansible/master/index.html"
    become: true
  - name: Start Apache Service
    service:
      name: httpd
      state: started
    become: true
```

After running this role, my webserver is configured.

3. Configure HA-Proxy LoadBalancer:

Now I need to configure my load-balancer service on ec2 instance named lbserver.

```
# tasks
file for
lbserver

        - name: install haproxy software
          package:
            name: "haproxy"
            state: present
          become: true
        - name: copy my conf file of lb
          template:
            src: "haproxy.cfg"
            dest: "/etc/haproxy/haproxy.cfg"
          become: true
        - name: start service lb
          service:
            name: "haproxy"
            state: started
          become: true
```

Now, all roles are created successfully.

Now I created one playbook for running all roles in a single click…

```
- hosts:
localhost
           roles:
                   - ec2_host
         - hosts: tag_Name_webserver
           remote_user: ec2-user
           roles:
                   - web
         - hosts: tag_Name_lbserver
           remote_user: ec2-user
           roles:
                   - lbserver
```

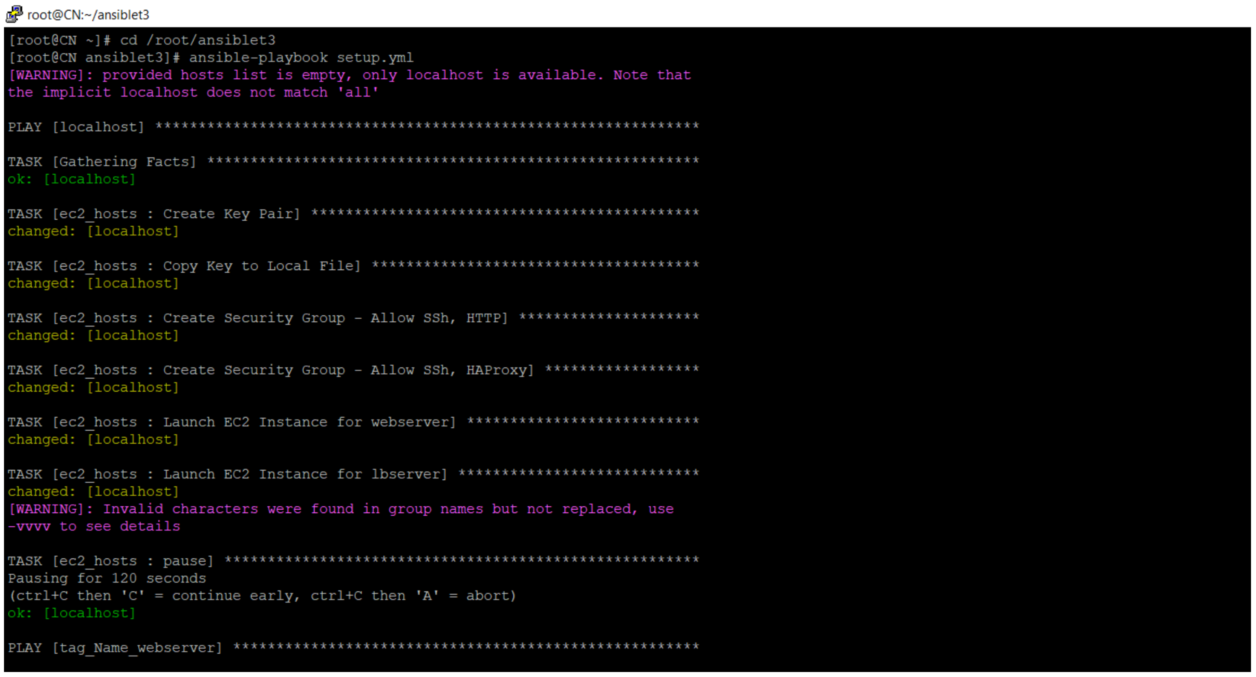
Save this playbook as setup.yml and then run…

```
ansible-playbook setup.yml
```

Finally, LoadBalancer is configured in a single command…✌

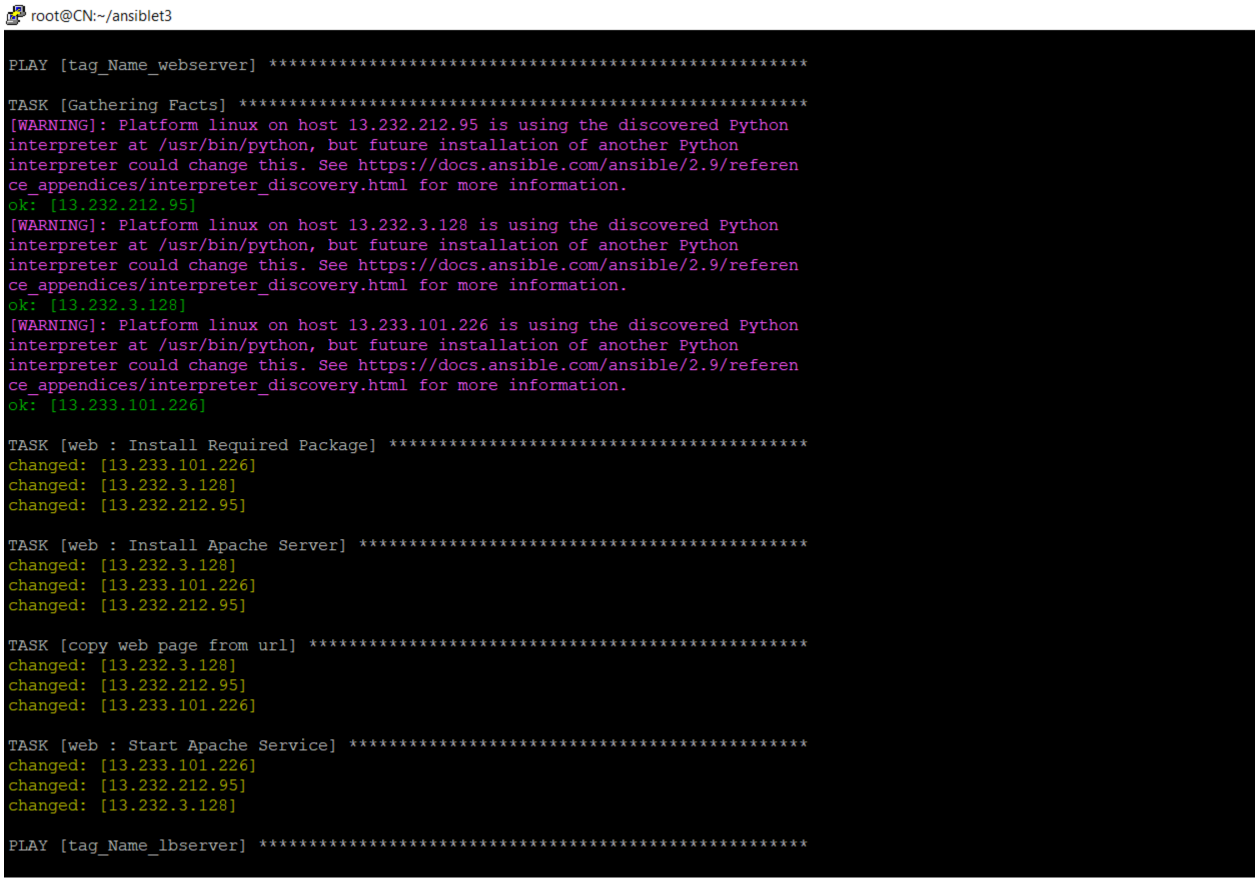
Let’s see the output of this…

Finally, LoadBalancer is configured in a single command…



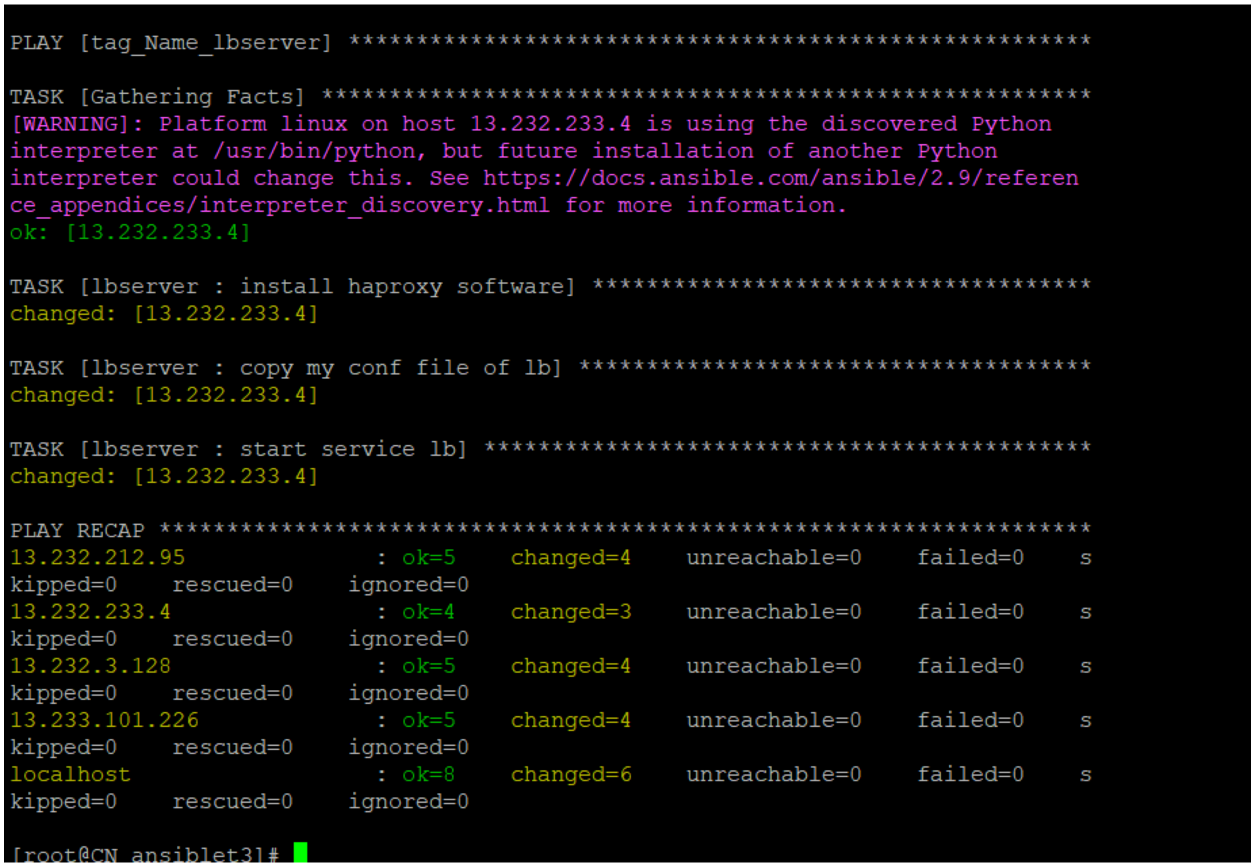
root@CN:~/ansiblet3

```
[root@CN ~]# cd /root/ansiblet3
[root@CN ansiblet3]# ansible-playbook setup.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

PLAY [localhost] ****************************************************************

TASK [Gathering Facts] **********************************************************
ok: [localhost]

TASK [ec2_hosts : Create Key Pair] **********************************************
changed: [localhost]

TASK [ec2_hosts : Copy Key to Local File] ***************************************
changed: [localhost]

TASK [ec2_hosts : Create Security Group - Allow SSh, HTTP] **********************
changed: [localhost]

TASK [ec2_hosts : Create Security Group - Allow SSh, HAProxy] *******************
changed: [localhost]

TASK [ec2_hosts : Launch EC2 Instance for webserver] ****************************
changed: [localhost]

TASK [ec2_hosts : Launch EC2 Instance for lbserver] *****************************
changed: [localhost]
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

TASK [ec2_hosts : pause] ********************************************************
Pausing for 120 seconds
(ctrl+C then 'C' = continue early, ctrl+C then 'A' = abort)
ok: [localhost]

PLAY [tag_Name_webserver] *******************************************************
```



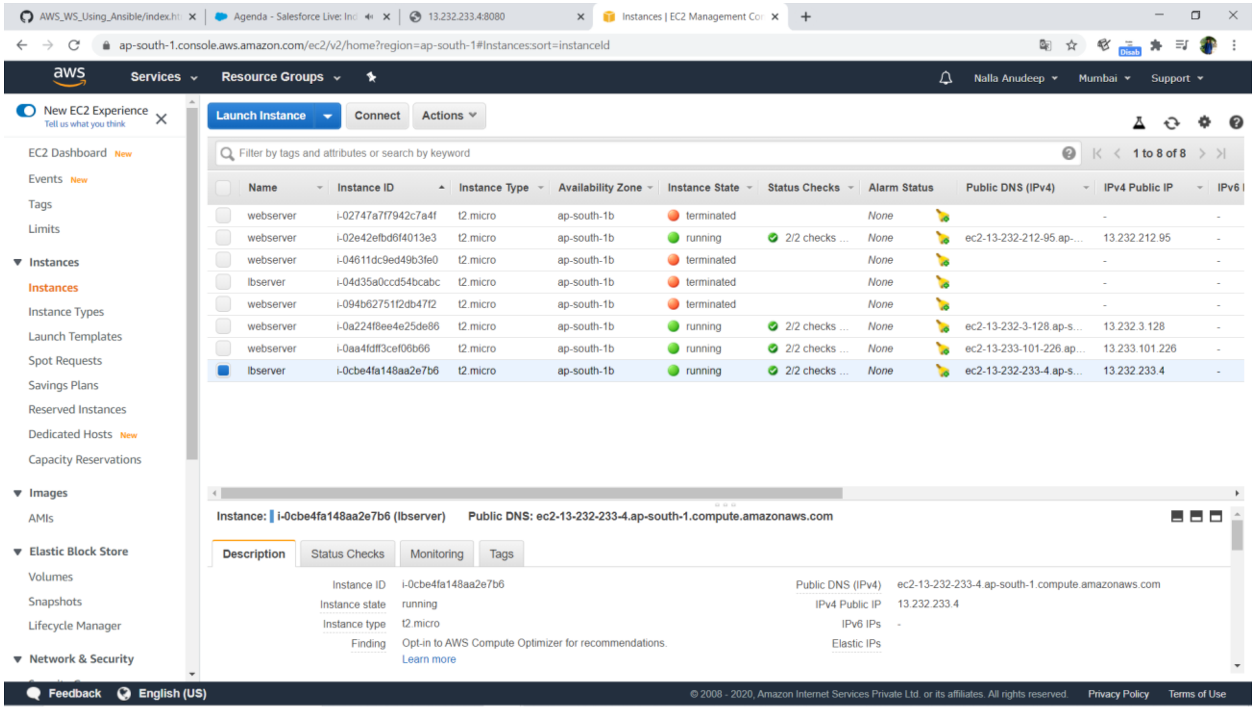
root@CN:~/ansiblet3

```
PLAY [tag_Name_webserver] *******************************************************

TASK [Gathering Facts] **********************************************************
[WARNING]: Platform linux on host 13.232.212.95 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/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [13.232.212.95]
[WARNING]: Platform linux on host 13.232.3.128 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/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [13.232.3.128]
[WARNING]: Platform linux on host 13.233.101.226 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/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [13.233.101.226]

TASK [web : Install Required Package] *******************************************
changed: [13.233.101.226]
changed: [13.232.3.128]
changed: [13.232.212.95]

TASK [web : Install Apache Server] **********************************************
changed: [13.232.3.128]
changed: [13.233.101.226]
changed: [13.232.212.95]

TASK [copy web page from url] ***************************************************
changed: [13.232.3.128]
changed: [13.232.212.95]
changed: [13.233.101.226]

TASK [web : Start Apache Service] ***********************************************
changed: [13.233.101.226]
changed: [13.232.212.95]
changed: [13.232.3.128]

PLAY [tag_Name_lbserver] ********************************************************
```



```
PLAY [tag_Name_lbserver] *******************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 13.232.233.4 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/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [13.232.233.4]

TASK [lbserver : install haproxy software] *************************************
changed: [13.232.233.4]

TASK [lbserver : copy my conf file of lb] **************************************
changed: [13.232.233.4]

TASK [lbserver : start service lb] *********************************************
changed: [13.232.233.4]

PLAY RECAP *********************************************************************
13.232.212.95              : ok=5    changed=4    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
13.232.233.4               : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
13.232.3.128               : ok=5    changed=4    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
13.233.101.226             : ok=5    changed=4    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
localhost                  : ok=8    changed=6    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[root@CN ansiblet3]#
```

So let's run my LoadBalancer IP and see it is working or not…

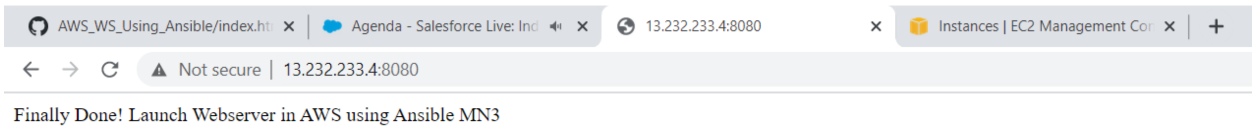


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 I
webserver	i-02747a7f7942c7a4f	t2.micro	ap-south-1b	🔴 terminated		None		-	-
webserver	i-02e42efbd6f4013e3	t2.micro	ap-south-1b	🟢 running	✓ 2/2 checks ...	None	ec2-13-232-212-95.ap-...	13.232.212.95	-
webserver	i-04611dc9ed49b3fe0	t2.micro	ap-south-1b	🔴 terminated		None		-	-
lbserver	i-04d35a0ccd54bcabc	t2.micro	ap-south-1b	🔴 terminated		None		-	-
webserver	i-094b62751f2db47f2	t2.micro	ap-south-1b	🔴 terminated		None		-	-
webserver	i-0a224f8ee4e25de86	t2.micro	ap-south-1b	🟢 running	✓ 2/2 checks ...	None	ec2-13-232-3-128.ap-s...	13.232.3.128	-
webserver	i-0aa4fdff3cef06b66	t2.micro	ap-south-1b	🟢 running	✓ 2/2 checks ...	None	ec2-13-233-101-226.ap...	13.233.101.226	-
lbserver	i-0cbe4fa148aa2e7b6	t2.micro	ap-south-1b	🟢 running	✓ 2/2 checks ...	None	ec2-13-232-233-4.ap-s...	13.232.233.4	-

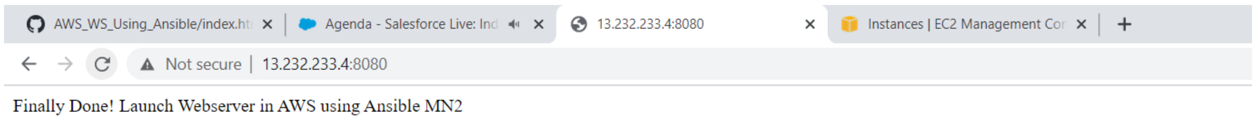
Instance: i-0cbe4fa148aa2e7b6 (lbserver)   Public DNS: ec2-13-232-233-4.ap-south-1.compute.amazonaws.com

Description | Status Checks | Monitoring | Tags

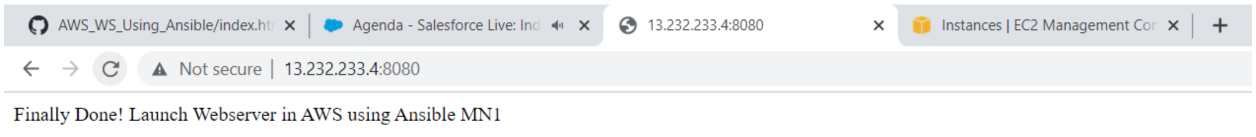
Instance ID	i-0cbe4fa148aa2e7b6	Public DNS (IPv4)	ec2-13-232-233-4.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	13.232.233.4
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	


AWS_WS_Using_Ansible/index.ht
Agenda - Salesforce Live: Ind
13.232.233.4:8080
Instances | EC2 Management Cor
Not secure | 13.232.233.4:8080

Finally Done! Launch Webserver in AWS using Ansible MN3


AWS_WS_Using_Ansible/index.ht
Agenda - Salesforce Live: Ind
13.232.233.4:8080
Instances | EC2 Management Cor
Not secure | 13.232.233.4:8080

Finally Done! Launch Webserver in AWS using Ansible MN2


AWS_WS_Using_Ansible/index.ht
Agenda - Salesforce Live: Ind
13.232.233.4:8080
Instances | EC2 Management Cor
Not secure | 13.232.233.4:8080

Finally Done! Launch Webserver in AWS using Ansible MN1

Yup … It’s working

GitHub Link: https://github.com/Anuddeeph/Deploy-Haproxy-In-aws-using-Ansible.git

Thankyou