



+



Configuring Web Server in AWS using ANSIBLE

In this article I am going to deploy a web server on the top of **AWS** cloud with the help of **ANSIBLE**.

Task Description..

- Provision EC2 instance through ansible.
- Retrieve the IP Address of instance using a dynamic inventory concept.
- Configure the web server through ansible!
- Create a role for the web server to customize the Instance and deploy the web page to the root directory.

Now Let's start doing the task.

First I will install **boto** and **boto3** libraries. For installing these two libraries i will use these two commands “**pip3 install boto**” and “**pip3 install boto3**” I have installed these two libraries so that ANSIBLE will be able to go to the AWS cloud..

Now i will create a role for launching instances on AWS cloud. I will create a role by using “**ansible-galaxy init [role name]**” .

```
root@CN: ~/awsCloud
[root@CN ~]# ansible-galaxy init awsCloud
- Role awsCloud was created successfully
[root@CN ~]# cd awsCloud/
[root@CN awsCloud]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@CN awsCloud]#
```

Inside the “**tasks/main.yml**” folder I will write my ansible code that will launch instances on aws cloud.

```
root@CN: ~/awsCloud
---
# tasks file for awsCloud
- ec2:
    key_name: mykey11
    instance_type: t2.micro
    image: ami-09a7bbd08886aafdf
    wait: yes
    count: 2
    vpc_subnet_id: subnet-23a7cc6f
    assign_public_ip: yes
    group_id: sg-070a79dff7b30dff5
~
~
~
~
```

Now I will create a playbook for launching an OS.

```
root@CN:~
```

```
- hosts: localhost
  roles:
    - awsCloud
```

Before running this playbook I will give the access key , secret key and region where ansible will launch the OS..

```
root@CN:~
```

```
[root@CN ~]# export AWS_REGION='ap-south-1'
[root@CN ~]# export AWS_ACCESS_KEY='[REDACTED]'
[root@CN ~]# export AWS_SECRET_KEY='[REDACTED]'
[root@CN ~]#
```

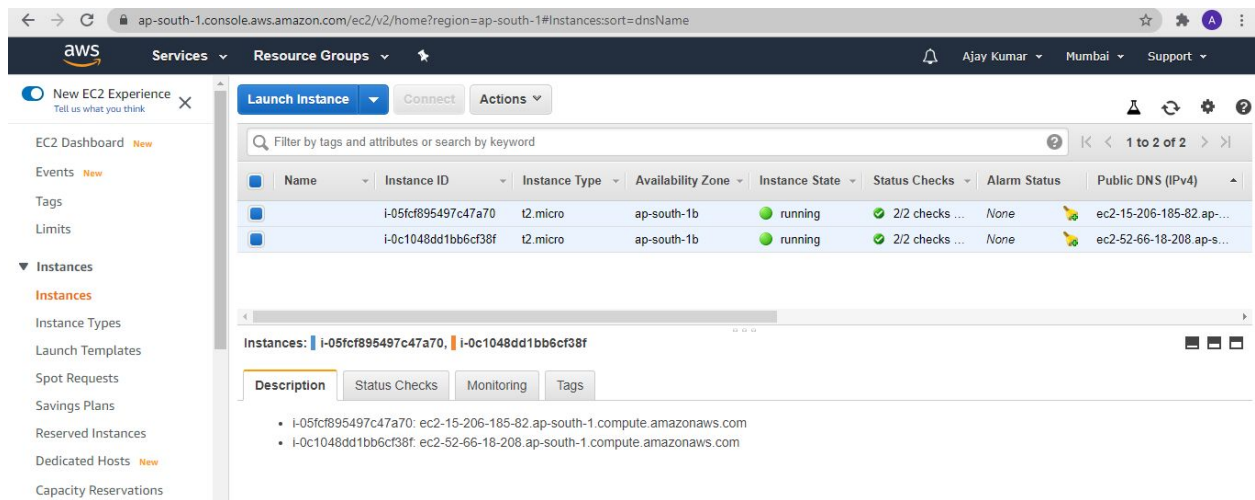
Now I will run the playbook by using the command “**ansible-playbook [your playbook name]**”.

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

TASK [awsCloud : ec2] *****
changed: [localhost]

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

EC2 Instances has been launched Successfully.



Now i will create a dynamic inventory that will automatically fetch the instances IP from the AWS cloud.

For this I will download a Script that is written in python language.

You will get this Script from the link below.

<https://github.com/ansible/ansible/blob/stable-2.9/contrib/inventory/ec2.py>

Let's download the script.

```
root@CN:~/mydb
[root@CN mydb]# wget https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.py
--2020-09-08 15:01:57-- https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73130 (71K) [text/plain]
Saving to: 'ec2.py'

ec2.py          100%[=====>]  71.42K  132KB/s   in 0.5s

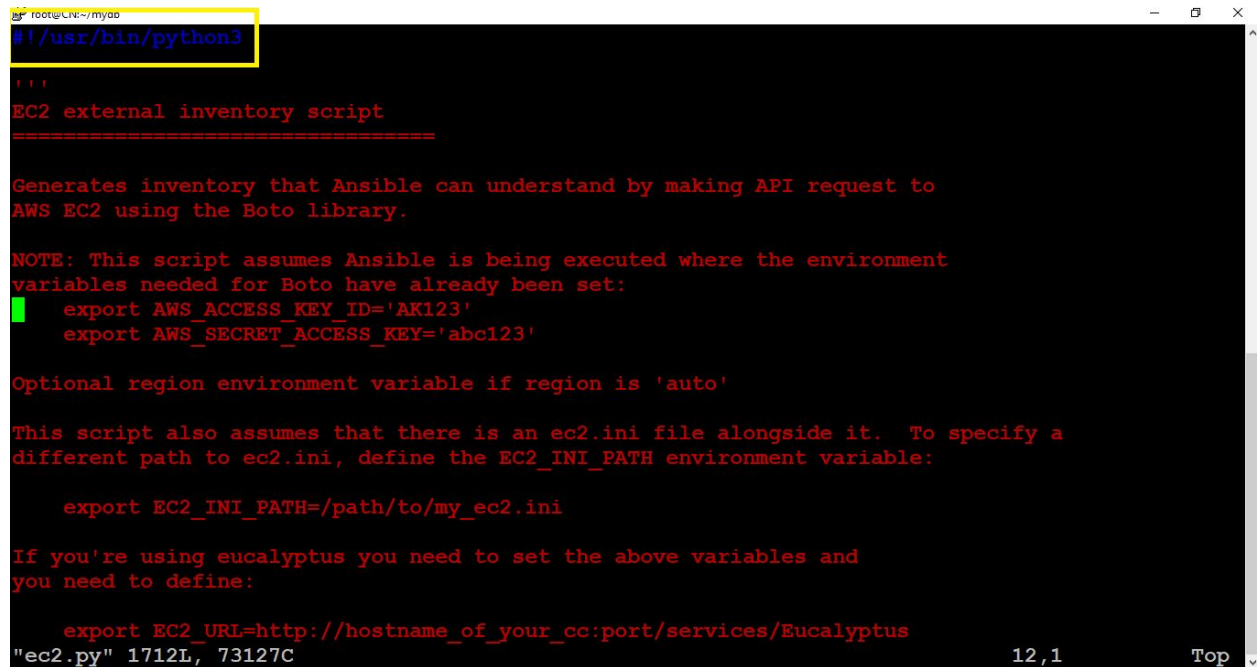
2020-09-08 15:02:00 (132 KB/s) - 'ec2.py' saved [73130/73130]

[root@CN mydb]# ls
ec2.py
[root@CN mydb]# chmod +x ec2.py
[root@CN mydb]# ls
ec2.py
[root@CN mydb]# vim ec2.py
[root@CN mydb]#
```

After downloading the script I have made the script executable by using “**chmod +x ec2.py**” command.

This script is written for the python 2 version and my system has python 3 . I have done some change on the script so that at the time of running the script I will not face any error.

Highlighted text is the change that I have made.



```
root@LNE:~/myoe
#!/usr/bin/python3

'''
EC2 external inventory script
=====

Generates inventory that Ansible can understand by making API request to
AWS EC2 using the Boto library.

NOTE: This script assumes Ansible is being executed where the environment
variables needed for Boto have already been set:
    export AWS_ACCESS_KEY_ID='AK123'
    export AWS_SECRET_ACCESS_KEY='abc123'

Optional region environment variable if region is 'auto'

This script also assumes that there is an ec2.ini file alongside it. To specify a
different path to ec2.ini, define the EC2_INI_PATH environment variable:

    export EC2_INI_PATH=/path/to/my_ec2.ini

If you're using eucalyptus you need to set the above variables and
you need to define:

    export EC2_URL=http://hostname_of_your_cc:port/services/Eucalyptus

"ec2.py" 1712L, 73127C                                     12,1      Top
```

Now i will set the path of the inventory in the ansible configuration file along with i will also set the path of the private key , privilege_esclation and remote_user.

root@CN:~/mydb

```
[defaults]
inventory = /root/mydb
host_key_checking=False
roles_path = /root
remote_user = ec2-user
private_key_file = /root/mykey11.pem

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=false
~
~
```

I have sent the private key from my windows to linux OS using **WinSCP** software.

After sending the key I have made the key executable using “**chmod 400 [key_name]**” command.

Now i will set the Access key and Private key ..

root@CN:~/mydb

```
[root@CN mydb]# export AWS_ACCESS_KEY_ID='[REDACTED]'
[root@CN mydb]# export AWS_SECRET_ACCESS_KEY='[REDACTED]'
[root@CN mydb]#
[root@CN mydb]#
[root@CN mydb]#
```


Now i will retrieve the IP of the instances using the “**ansible all --list-hosts**” command.

```
root@CN:~/mydb
[root@CN mydb]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
  hosts (2):
    52.66.18.208
    15.206.185.82
[root@CN mydb]#
```

Finally I have retrieved the IP dynamically.

Let's check the connectivity using “**ansible all -m ping**”

```
root@CN:~
[root@CN ~]# ansible all -m ping
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Platform linux on host 52.66.18.208 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.
52.66.18.208 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 15.206.185.82 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.
15.206.185.82 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Let's create a role for setting up the web server in the cloud.

Creating the role.

```
root@CN:~/cloudhttpd
[root@CN ~]# ansible-galaxy init cloudhttpd
- Role cloudhttpd was created successfully
[root@CN ~]# cd cloudhttpd/
[root@CN cloudhttpd]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@CN cloudhttpd]#
```

Code for setting up the web server.

```
root@CN:~/cloudhttpd
--
# tasks file for cloudhttpd
- name: "installing software httpd"
  package:
    name: "httpd"
    state: present
- name: "copying webpage to the web server"
  copy:
    content: "Ansible task 2 successfully done"
    dest: "/var/www/html/index.html"
- name: "starting the service of webserver"
  service:
    name: "httpd"
    state: started
    enabled: yes
~
~
```

```
root@CN:~
- hosts: ec2
  roles:
    - role: cloudhttpd
~
```

Creating a Playbook.

Now I will run the playbook using the “**ansible-playbook** [your_playbook_name]” command.

```
[root@CN ~]# ansible-playbook deployweb.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [ec2] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 15.206.185.82 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: [15.206.185.82]
[WARNING]: Platform linux on host 52.66.18.208 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: [52.66.18.208]

TASK [cloudhttpd : installing software httpd] *****
changed: [52.66.18.208]
changed: [15.206.185.82]

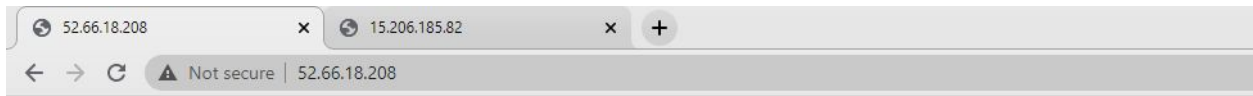
TASK [cloudhttpd : copying webpage to the web server] *****
changed: [52.66.18.208]
changed: [15.206.185.82]

TASK [cloudhttpd : starting the service of webserver] *****
changed: [15.206.185.82]
changed: [52.66.18.208]

PLAY RECAP *****
15.206.185.82      : ok=4    changed=3    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
52.66.18.208     : ok=4    changed=3    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
```

Playbook has been run Successfully.

Accessing the site from the browser.



Ansible task 2 successfully done



Ansible task 2 successfully done

Finally I have done the task.

Thank You guys for reading this document.

Have a great day.



