

## What is Ansible?

Ansible is a simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

This tool is very simple to use yet powerful enough to automate complex multi-tier IT application environments.

I assume you have Ansible installed on either your workstation or an Amazon EC2 instance — Ansible has great documentation for installation...([http://docs.ansible.com/intro\\_installation.html](http://docs.ansible.com/intro_installation.html))

([http://docs.ansible.com/intro\\_getting\\_started.html](http://docs.ansible.com/intro_getting_started.html))

**Now Let's first discuss What we are going to learn today...**

## **Steps:**

- 1. Provision EC2 instance through Ansible.**
- 2. Fetch the public IP using the EC2 Dynamic Inventory concept.**
- 3. Configure Webserver through Ansible on EC2.**

## **Prerequisites:**

- I have chosen to use RHEL-8 for my Ansible “Master” in my local Virtual Machine.
- Having Ansible installed with Python3. I am using pip command to install Ansible. After that, you need to create directory and file manually for the ansible config file and inventory file.

```
root@CN:/etc/ansible

[root@CN ansible]# pwd
/etc/ansible
[root@CN ansible]# ls
ansible.cfg  home.html  hosts  ho.txt  roles  web.yml
[root@CN ansible]# cat ansible.cfg
[defaults]
inventory = /etc/ansible/hosts
host_key_checking = False
roles_path = /etc/ansible/roles
private_key_file = /root/.ssh/key.pem
[root@CN ansible]#
```

You need to create **ansible.cfg** file. Here you define your inventory directory, roles directory etc. You can check from above image.

After that, for checking run command ***ansible --version***.

```
root@CN:/etc/ansible

[root@CN ansible]# ansible --version
ansible 2.9.11
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.6/site-packages/ansible
  executable location = /usr/local/bin/ansible
  python version = 3.6.8 (default, Jan 11 2019, 02:17:16) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]
[root@CN ansible]#
```

- Install Boto and Boto3 python library.

pip3 install boto

pip3 install boto3

- Need one IAM role in AWS because **boto** can automatically source my AWS API credentials provided by an Amazon EC2 Identity and Access Management (IAM) role to run the playbook.

We need to provide AWS credentials so that Ansible fetch the account details. For providing AWS credential, We have multiple Way...

One way, you will need to set environment variables for your Secret and Access key:

```
export AWS_ACCESS_KEY_ID='YOUR_AWS_API_KEY'
```

```
export
```

```
AWS_SECRET_ACCESS_KEY='YOUR_AWS_API_SECRET_KEY'
```

And another way is for providing Credential is by using **awscli** software. You only need to download software and provide AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY using command **aws configure**.

## Setting up Dynamic Inventory:

**Now** to get started with **dynamic inventory** management, you'll need to grab the **EC2.py** script and the **EC2.ini** config file. The EC2.py script is written using the Boto EC2 library and will query AWS for your running Amazon EC2 instances. The EC2.ini file is the config file for EC2.py and can be used to limit the scope of Ansible's reach. You can specify the regions, instance tags, or roles that the EC2.py script will find.

Link for download EC2.py and EC2.ini file —

(<https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.py>)

(<https://raw.githubusercontent.com/ansible/ansible/dev/contrib/inventory/ec2.ini>)

Use wget/curl/Git to download these files in directory /etc/ansible/hosts. I had to create this directory manually because I installed Ansible using pip. After that, make this file executable using command...

***chmod +x ec2.py***

***chmod +x ec2.ini***

Now you also need to set the path for these files, so that Ansible fetch the path of inventory script.

***export ANSIBLE\_HOSTS=/path to/ec2.py***

***export EC2\_INI\_PATH=/path to/ec2.ini***

As I'm using Linux, we need to do a small patch in the ec2.py file. By default, it gives executable location ***#!/usr/bin/env python***, modify it to ***#!/usr/bin/env python3***.

Now my dynamic Inventory is configured.


Now we ready to see Ansible shine.

Now come to task, which we are going to perform...

## 1. Provision EC2 instance through Ansible:


I am creating Ansible role for performing the task. For creating Ansible roles, run command

**ansible-galaxy init role\_name**

 root@CN:/etc/ansible/ec2\_hosts

```
[root@CN ansible]# ansible-galaxy init ec2_hosts
- Role ec2_hosts was created successfully
[root@CN ansible]# cd ec2_hosts/
[root@CN ec2_hosts]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@CN ec2_hosts]#
```

This create all the above directory, you only need to put the tasks, vars and other files in their preferred location.

 root@CN:/etc/ansible/roles/ec2\_hosts

```
[root@CN roles]# pwd
/etc/ansible/roles
[root@CN roles]# ls
ec2_hosts  lbserver  webserver
[root@CN roles]# cd ec2_hosts/
[root@CN ec2_hosts]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@CN ec2_hosts]#
```

Now let's discuss the code...

# tasks file for webserver

- name: Create Key Pair

ec2\_key:

name: mykey

aws\_region: "{{ region }}"

register: ec2\_key

- name: Copy Key to Local File

copy:

content: "{{ ec2\_key.key.private\_key }}"

dest: "{{ key\_dest }}"

mode: '0600'

This code creates a **key\_pair** and save it locally.

I also need to create one security\_group, for this...

- name: Create Security Group — Allow SSh,  
HTTP

ec2\_group:

name: sg\_ansible-1

description: sg for ansible 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

This security group we attached in our ec2\_instance and, this allow ssh and http protocol.

Now let's see the code for launching the ec2\_instance which we are going to use as managed\_node.

- name: Launch EC2 Instance

ec2:

key\_name: mykey

instance\_type: t2.micro

image: "{{ image\_id }}"

wait: yes

region: "{{ region }}"

count: 1

vpc\_subnet\_id: subnet-040fe014984c437d9

group\_id: "{{ sg\_aws.group\_id }}"

assign\_public\_ip: yes

state: present

register: ec2

Now my ec2\_instance is provisioned.

## 2. Fetch the public IP using the EC2 Dynamic Inventory concept:

We already configured the Dynamic Inventory above. Now for check how many instances are running, run command i.e.

ansible all — list-hosts


```
root@CN:/etc/ansible/hosts
[root@CN hosts]# pwd
/etc/ansible/hosts
[root@CN hosts]# ls
ec2.ini  ec2.py
[root@CN hosts]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
 hosts (1):
   13.127.161.48
[root@CN hosts]#
```

This is the IP of my ec2\_instance.

## 3. Configure WebServer through Ansible on EC2:

I'm creating one more role for configuring the apache webserver on the top of ec2\_instance.

HERE is the code for this...

 root@CN:/etc/ansible/roles/web/tasks

```
[root@CN ec2_hosts]# cd /etc/ansible/
[root@CN ansible]# cd roles/web
[root@CN web]# cd tasks/
[root@CN tasks]# vim main.yml
[root@CN tasks]# cat main.yml
---
# 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

[root@CN tasks]#
```

Now for running both the role, I created one playbook which run first role in localhost and another in ec2\_instance.

```
root@CN:~/ansible2  
[root@CN ~]# cd /root/ansible2/  
[root@CN ansible2]# ls  
setup.yml  
[root@CN ansible2]# cat setup.yml  
- hosts: localhost  
  roles:  
    - ec2_hosts  
- hosts: ec2  
  user: ec2-user  
  roles:  
    - web  
[root@CN ansible2]#
```

But IF you try to run this playbook, it gets failed after running the first playbook. Because, by default, Ansible doesn't refresh the inventory in the mid of playbook running. For this, we need to run one more module named ***refresh\_inventory***. I put this task in my ec2\_host role, Code for this...

- name: Refresh Inventory File

meta: refresh\_inventory

- pause:

minutes: 2

Now, when you run the playbook, it run smoothly without giving any error...

YO, Finally OUR WEBSITE IS LAUNCHED...

GitHub Link:

[https://github.com/Anuddeeph/AWS\\_WS\\_Using\\_Ansible.git](https://github.com/Anuddeeph/AWS_WS_Using_Ansible.git)

Let us see output of this...

```
root@CN:~/ansible2
[root@CN ansible2]# ls
setup.yml
[root@CN ansible2]# 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 : Launch EC2 Instance] *************************************************************************************************************************************
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 [ec2] *************************************************************************************************************************************
TASK [Gathering Facts] *************************************************************************************************************************************
[WARNING]: Platform linux on host 13.127.161.48 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: [13.127.161.48]

TASK [web : Install Required Package] *************************************************************************************************************************************
changed: [13.127.161.48]

TASK [web : Install Apache Server] *************************************************************************************************************************************
changed: [13.127.161.48]

TASK [copy web page from url] *************************************************************************************************************************************
changed: [13.127.161.48]

TASK [web : Start Apache Service] *************************************************************************************************************************************
changed: [13.127.161.48]

PLAY RECAP *************************************************************************************************************************************
13.127.161.48      : ok=5   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
localhost         : ok=6   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

[root@CN ansible2]#
```

Instances | EC2 Management Console

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:sort=instancetype

Services Resource Groups

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6
	i-001de6d49e2010b23	t2.micro	ap-south-1a	terminated		None			
	i-072160b3ff9418351	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-127-161-48.ap-...	13.127.161.48	

Instance: i-072160b3ff9418351 Public DNS: ec2-13-127-161-48.ap-south-1.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID	i-072160b3ff9418351	Public DNS (IPv4)	ec2-13-127-161-48.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	13.127.161.48
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations.	Elastic IPs	

Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Instances | EC2 Management Console

13.127.161.48

Not secure | 13.127.161.48

Finally Done! Launch Webserver in AWS using Ansible

