

Topic: Ec2 server Provisioning on AWS using ansible as IAC tool

Install AWS CLI on Ansible server

```
root@eab92bc36259:/ansible_playbooks# pip install awscli
Collecting awscli
  Downloading awscli-1.32.95-py3-none-any.whl (4.4 MB)
    4.4/4.4 MB 37.2 MB/s eta
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib/python3/di
```

Create IAM User on AWS account

With the help of this user we will create VPC and EC2 instances on AWS accounts using ansible.

Store its access key and secret access key

Note: The user have access of VPC and EC2 access creation

Connect Ansible with your AWS account

```
root@eab92bc36259:/ansible_playbooks# aws configure
AWS Access Key ID [None]: AKIA3FLDYIHOAQYNJOHE
AWS Secret Access Key [None]: ieuMuIOWABxHm6Zs0fszwrkGxAlISVX0S6307s1M
Default region name [None]:
Default output format [None]:
root@eab92bc36259:/ansible_playbooks#
```

Install python library boto3 to use aws_module with ansible

```
root@eab92bc36259:/ansible_playbooks/aws_with_aws# pip install boto boto3
Collecting boto
  Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 16.7 MB/s eta 0:00:00
Collecting boto3
  Downloading boto3-1.34.95-py3-none-any.whl (139 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.3/139.3 KB 22.6 MB/s eta 0:00:00
Requirement already satisfied: botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages (from boto3) (2.9.0.post0)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3/dist-packages (from botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages) (1.16.0)
Requirement already satisfied: s3transfer<0.11.0,>=0.10.0 in /usr/local/lib/python3.8/dist-packages (from botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages) (0.3.5)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/lib/python3/dist-packages (from botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages) (1.25.4)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.8/dist-packages (from botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from botocore<1.35.0,>=1.34.95 in /usr/local/lib/python3.8/dist-packages) (1.16.0)
Installing collected packages: boto, boto3
Successfully installed boto-2.49.0 boto3-1.34.95
WARNING: Running pip as the 'root' user can result in broken permissions and con
ommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@eab92bc36259:/ansible_playbooks/aws_with_aws#
```

Write ansible playbook for ec2 server provisioning

I have make use of aws module

In this playbook I have created a VPC, Subnet inside that VPC, security group for EC2 instance and code for Ec2 type of instance you want

```
root@eab92bc36259:/ansible_playbooks/aws_with_aws# vi ec2_provisioning.yml
root@eab92bc36259:/ansible_playbooks/aws_with_aws# cat ec2_provisioning.yml
---
- name: Create VPC on AWS
  hosts: localhost
  connection: local
  gather_facts: no
  vars:
    aws_region: "ap-south-1" # Mumbai region
    vpc_cidr_block: "10.0.0.0/16"
    vpc_name: "MyVPC"
    subnet_cidr_block: "10.0.1.0/24" # CIDR block for the subnet
```

Run the Playbook

```
root@eab92bc36259:/ansible_playbooks/aws_with_aws# ansible-playbook ec2_provisioning.yml

PLAY [Create VPC on AWS] *****

TASK [Create VPC] *****
changed: [localhost]

TASK [Print VPC ID] *****
ok: [localhost] => {
  "msg": "VPC ID is vpc-0558256080df847a0"
}

TASK [Create Subnet] *****
changed: [localhost]

TASK [Print Subnet ID] *****
ok: [localhost] => {
  "msg": "Subnet ID is subnet-06022ac61462d76df"
}

TASK [Create security group] *****
changed: [localhost]

TASK [Launch EC2 instance] *****
changed: [localhost]

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

root@eab92bc36259:/ansible_playbooks/aws_with_aws#
```

Output:

VPC Created

Your VPCs (2) Info							Refresh	Actions	Create VPC
<input type="text" value="Search"/>							<	1	>
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6				
<input type="checkbox"/>	MyVPC	vpc-0558256080df847a0	Available	10.0.0.0/16	-				
<input type="checkbox"/>	-	vpc-034b2ca5c8de01f98	Available	172.31.0.0/16	-				

Subnet Created

Subnets (4) Info					Refresh	Actions	Create subnet
<input type="text" value="Find resources by attribute or tag"/>					<	1	>
<input type="checkbox"/>	Name	Subnet ID	State	VPC			
<input type="checkbox"/>	My-subnet	subnet-06022ac61462d76df	Available	vpc-0558256080df847a0 MyV.			

Security Group Created

Security Groups (4) Info					Export security groups to CSV	Create security group
<input type="text" value="Find resources by attribute or tag"/>					< 1 > Settings	
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID		
<input type="checkbox"/>	-	sg-0a0007140052119171	launch-wizard-1	vpc-05402c65c00ee0		
<input type="checkbox"/>	-	sg-0fdcd304dddf862e8	default	vpc-034b2ca5c8de0		
<input type="checkbox"/>	-	sg-06c6cb466cb7b69c4	MySecurityGroup	vpc-0558256080df8		

EC2 instance created

Instances (2) Info							Connect	Instance state	Actions	Launch instances
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>							All states			
Instance state = running Clear filters							< 1 > Settings			
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status				
<input type="checkbox"/>	ansible-server	i-0c49a774dc3e3bd59	Running	t2.micro	2/2 checks passed	View alarms				
<input type="checkbox"/>	Ec2_via_Ansible	i-04b119fbf0de53eb0	Running	t2.micro	2/2 checks passed	View alarms				