<https://devopscube.com/setup-ansible-aws-dynamic-inventory/>

<https://devopscube.com/ansible-dymanic-inventry-google-cloud/>

<https://docs.microsoft.com/en-us/azure/developer/ansible/dynamic-inventory-configure?tabs=azure-cli>

Deploy ubuntu t2.medium

From root user, create new user with name ansibleadmin.

Give sudo rights and password less sudo for ansible.

Install ansible from ansibleadmin user.

Generate ssh-keys from ansible user.

New AWS AMI need to be created using packer.json & packer-vars.json

Ansible Controllers Needs Following Packages:

sudo apt-get install python3 -y

sudo apt-get install python3-pip -y

sudo apt-get install unzip jq -y

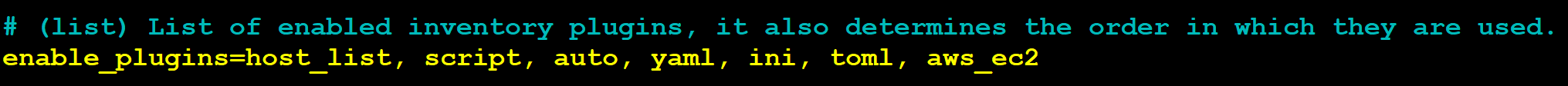
sudo pip3 install boto3

Install aws cli version 2.0

Assign EC2 Full Access Role to controller if you dont want to use Access & Secret Keys.

We can use static and dynamic inventory at the same time by enabling plugins as shown below.

enable\_plugins=host\_list, script, auto, yaml, ini, toml, aws\_ec2



*ansible -i /opt/ansible/inventory/aws\_ec2.yaml tag\_ManagedBy\_Terraform -m ping*

*ansible -i invfile pvt -m ping*

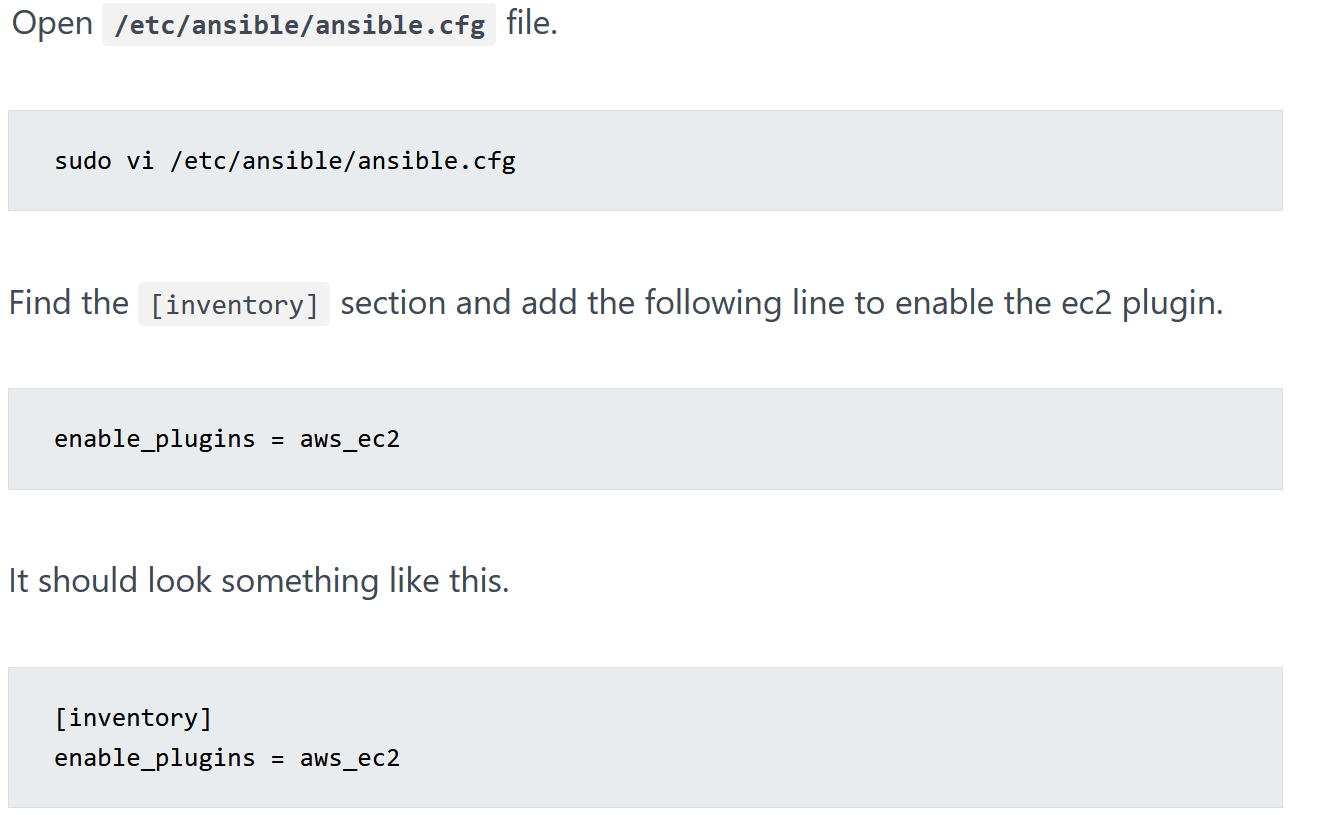
*ansible-playbook -i /opt/ansible/inventory/aws\_ec2.yaml Playbooks/8.Dyanmic-Inventory/dynamic\_nginx-jinja2.yml -vv*

Plugin:

plugin: gcp\_compute

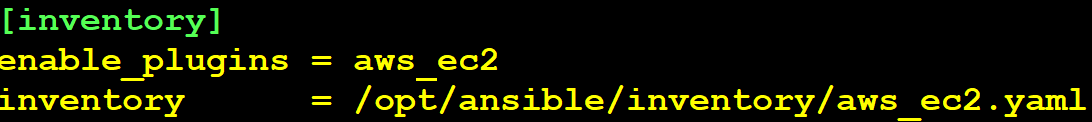
plugin: azure\_rm

plugin: aws\_ec2



[inventory]

enable\_plugins = aws\_ec2



inventory = /opt/ansible/inventory/aws\_ec2.yaml

private\_key\_file=/home/ansibleadmin/.ssh/ansiblekey.pem

export dynamic='/opt/ansible/inventory/aws\_ec2.yaml'

sudo mkdir -p /opt/ansible/inventory

cd /opt/ansible/inventory

sudo nano aws\_ec2.yaml

---

plugin: aws\_ec2

regions:

- us-east-1

keyed\_groups:

- key: tags

prefix: tag

- prefix: instance\_type

key: instance\_type

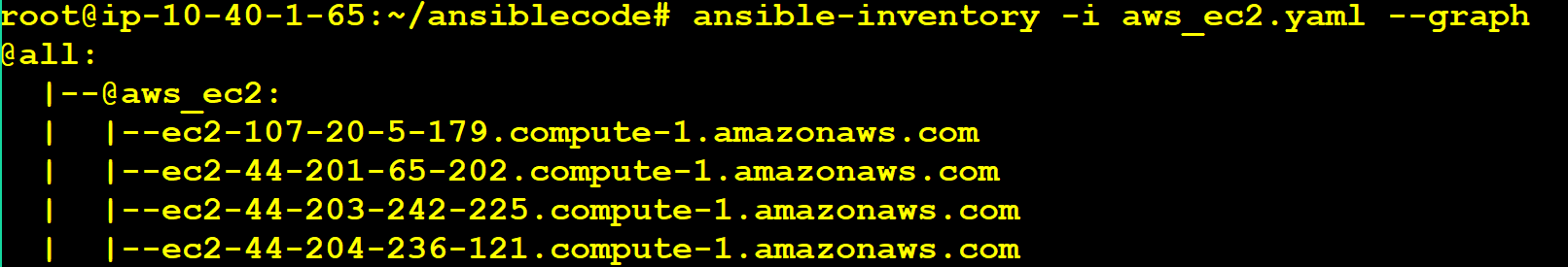
- key: placement.region

prefix: aws\_region

ansible-inventory -i /opt/ansible/inventory/aws\_ec2.yaml --list

ansible -i /opt/ansible/inventory/aws\_ec2.yaml tag\_Owner\_Sree -m ping

ansible-inventory -i /opt/ansible/inventory/aws\_ec2.yaml --graph



export dynamic='/opt/ansible/inventory/aws\_ec2.yaml'

ansible-inventory -i $dynamic --graph

ansible-playbook -i $dynamic Playbooks/dynamic-inv-playbook.yml --syntax-check

ansible-playbook -i $dynamic Playbooks/dynamic-inv-playbook.yml

ansible-playbook Playbooks/dynamic-role-mysql.yml -i $dynamic -e "variable\_host=tag\_Owner\_Sree" --ask-vault-pass

ansible-playbook Playbooks/dynamic-nginx.yml -i $dynamic -e "variable\_host=tag\_Owner\_Sree"

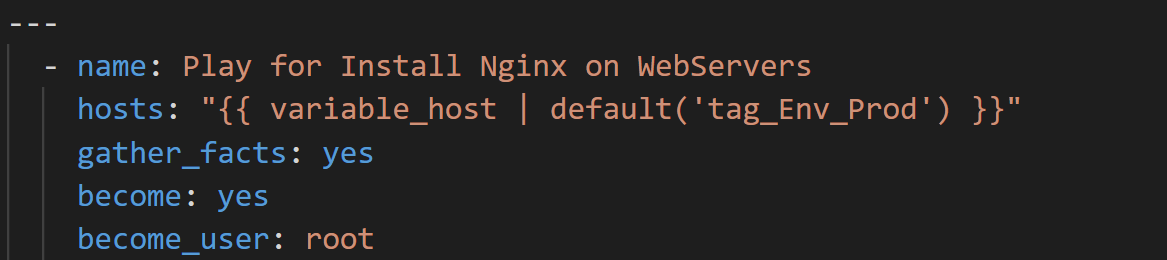
#Below code is available from DevSecOps B36 Batch

ansible-playbook Playbooks/4.Jinja/dynamic\_nginx-jinja2.yml -i aws\_ec2.yaml -e "variable\_host=instance\_type\_t2\_small"

ansible-playbook Playbooks/8.Dyanmic-Inventory/dynamic\_mysql.yml -i aws\_ec2.yaml -e "variable\_host=tag\_Env\_production"

To use above command we need to update hosts in playbook file as:

"{{ variable\_host | default('tag\_Env\_Prod') }}"



Deploy 3 Servers with Tags Owner=Sree

export dynamic='/opt/ansible/inventory/aws\_ec2.yaml'

ansible-inventory --graph -i $dynamic

ansible tag\_Owner\_Sree -m shell -a 'sudo apt install -y nginx jq unzip'

ansible tag\_Owner\_Sree -m shell -a 'echo "<h1>Welcome To DevOps In Telugu</h1>" > /var/www/html/index.nginx-debian.html' --become

packer.json

{

"\_comment": "Create Aws AMI with Ubuntu 20.04",

"variables": {

"region": "",

"source\_ami": "",

"instance\_type": "",

"vpc\_id": "",

"subnet\_id": ""

},

"\_comment1": "packer build -var \"aws\_secret\_key=foo\" packer.json",

"\_comment2": "packer build -var-file packer-vars.json packer.json",

"builders": [

{

"type": "amazon-ebs",

"region": "{{user `region`}}",

"source\_ami": "{{user `source\_ami`}}",

"instance\_type": "{{user `instance\_type`}}",

"ssh\_username": "ubuntu",

"ami\_name": "DevOpsClass-B25-Build-{{isotime | clean\_resource\_name}}",

"vpc\_id": "{{user `vpc\_id`}}",

"subnet\_id": "{{user `subnet\_id`}}",

"tags": {

"Name": "DevOpsClass-B25-Build-{{isotime | clean\_resource\_name}}"

}

}

],

"provisioners": [

{

"type": "shell",

"inline": [

"sleep 15",

"sudo useradd -m ansibleadmin --shell /bin/bash",

"sudo mkdir -p /home/ansibleadmin/.ssh",

"sudo chown -R ansibleadmin /home/ansibleadmin/",

"sudo touch /home/ansibleadmin/.ssh/authorized\_keys",

"sudo usermod -aG sudo ansibleadmin",

"echo 'ansibleadmin ALL=(ALL) NOPASSWD: ALL' | sudo tee -a /etc/sudoers",

"echo 'ssh-rsa ' | sudo tee /home/ansibleadmin/.ssh/authorized\_keys"

]

}

]

}

packer-vars.json

{

"region": "us-east-1",

"source\_ami": "ami-04505e74c0741db8d",

"instance\_type": "t2.micro",

"vpc\_id": "vpc-0f289bdbc4725af60",

"subnet\_id": "subnet-0d6dcc6af907c7833"

}