<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 Pckages:

sudo apt-get install python3 -y

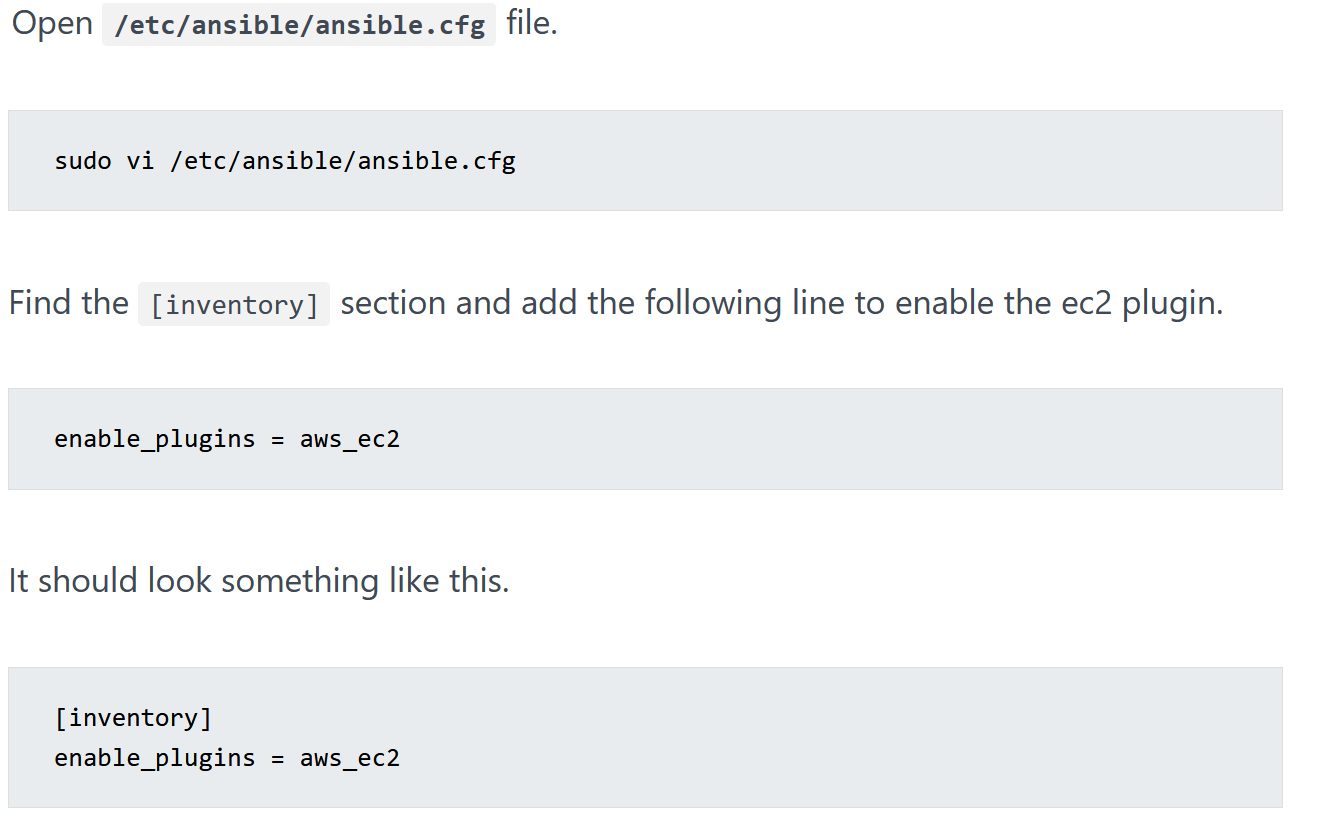
sudo apt-get install python3-pip -y

sudo apt-get install unzip -t

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.



Deploy 3 Servers with Tags Owner=Sree

ansible-inventory --graph

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"

}

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