**Ansible Playbook for Apache Web server**

An Ansible Playbook provision Apache server on AWS instances. No need of root permission but user should have proper IAM roles while creating and accessing aws resources. This Ansible script creates pre baked AMI which helps the DevOps team in auto scaling.

**Requirements**

You need to have ansible installed in your local environment, should have .pem file to access aws resources with permission 400.

#install python packages to install aws command line interface.

sudo yum install python-pip

# install awscli packages to access aws resources.

sudo pip install awscli

#Test which aws we are going to use

which aws

#Launch a new instance with your credentials and import one key pair and aws-key-id and aws-secret-key.

#Start ssh agent

Sudo ssh-agent bash

#add your public key to ssh agent

Sudo ssh-add public\_key\_file\_path

#connect to the aws instance, which is going to run ansible script.

Aws configure

#give your aws-access-key, aws-secret-key and region.

**Create Inventory file**

#create inventory file in the location /etc/ansible/hosts

Sudo vim /etc/ansible/hosts

Add your hosts to the inventory file,

[localhost]

#This is my local hostname, add your local host name here.

polas1d1.mylabserver.com

[webservers]

#this is my aws instance, add your aws instance dns

ec2-54-245-183-241.us-west-2.compute.amazonaws.com

**Playbook Variables**

The only required variables are available in temp/cred\_aws.yml file. You need to replace the values with your values.

aws\_akey : your\_access\_key

aws\_skey : your\_secreat\_key

aws\_region : your\_region

awsinstanceid : created\_instanceid

device\_name : storage name

**Playbook**

--- # Ansible Playbook to install Web Server and Use it to create Pre bake image

**#Target hosts are web servers**

- hosts: webservers

connection: ssh

remote\_user: ec2-user

become: yes

gather\_facts: yes

tasks:

- name: Apache server installation started

yum: name={{ item }} state=latest

with\_items:

- httpd

- wget

notify:

- CopySiteFiles

- RestartHTTPD

- WaitForSite

- TestSite

- DisplayResults

handlers:

**#Copy node.js files to target hosts, here I worked on index.html**

- name: CopySiteFiles

copy: src=temp/index.html dest=/var/www/html/index.html owner=root group=root mode=0655 backup=yes

- name: RestartHTTPD

service: name=httpd state=restarted

- name: WaitForSite

wait\_for: host={{ ansible\_nodename }} port=80 delay=5

- name: TestSite

shell: /usr/bin/wget http://localhost

register: site\_result

- name: DisplayResults

debug: var=site\_result

**#Pre Baking AMI to make available to other teams to add auto scaling group**

- hosts: localhost

connection: local

#AWS user name who has IAM roles to access ec2 instances

remote\_user: dhana

become: yes

gather\_facts: no

vars\_files:

- temp/cred\_aws.yml

tasks:

- name: Take a snapshot backup of the website directory

ec2\_snapshot:

aws\_access\_key: "{{ aws\_akey }}"

aws\_secret\_key: "{{ aws\_skey }}"

region: "{{ aws\_region }}"

instance\_id: "{{ awsinstanceid }}"

device\_name: "{{ device\_name }}"

description: Initial Playbook Static Site Deployment Backup

wait: no

register: snapshot\_results

notify:

- DisplaySnapshotResults

- CreateNewAMITemplate

- DisplayAMICreationResults

handlers:

- name: DisplaySnapshotResults

debug: var=snapshot\_results

- name: CreateNewAMITemplate

ec2\_ami:

aws\_access\_key: "{{ aws\_akey }}"

aws\_secret\_key: "{{ aws\_skey }}"

region: "{{ aws\_region }}"

instance\_id: "{{ awsinstanceid }}"

wait: no

name: myansibleamitemplateForeSee

tags:

Name: MyNewAnsibleAMITemplateforesee

Service: TestAMITemplatePlaybookForeSee

register: ami\_results

- name: DisplayAMICreationResults

debug: var=ami\_results

Finally we can refactor all the code into the roles as

Roles

-Builders

- Server-common

-site.yml

--Vars

-main.yml

--handlers

-main.yml

-- Tasks

-main.yml

-- Files

-Webserver

All the variables will be placed in main.yml in vars folder

All the tasks will be placed in main.yml in tasks folder

All the handlers will be placed in main.yml in handlers folder

All the configuration files and scripting files will be placed in files folder.