Before we do anything we need to install dependencies on local machine.

1. Install Ansible

2. Install Packer

3. Install and configure AWS CLI

**My project tree looks like this**

$ tree

.

├── packer

│ └── provisioners

│ ├── ansible

│ └── scripts

└── src

└── application

**STEP1: Creating the Packer Template**

cd packer

vim packer-build.json

**My contents looks like below for the file packer-build.json**

{

"variables": {

"aws\_access\_key": "myaccesskey",

"aws\_secret\_key": "mysecretkey",

"ami\_name": "ami-0975ce566eec139c3",

"aws\_region": "ap-southeast-2",

"ssh\_username": "ec2-user",

"vpc\_id": "vpc-03dac9a4c4a04xyz1",

"subnet\_id": "subnet-0998b21ffd55bxyz2"

},

"builders": [{

"type": "amazon-ebs",

"access\_key": "{{user `aws\_access\_key`}}",

"secret\_key": "{{user `aws\_secret\_key`}}",

"region": "{{user `aws\_region`}}",

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

"force\_deregister": "true",

"ssh\_username": "{{user `ssh\_username`}}",

"communicator": "ssh",

"associate\_public\_ip\_address": true,

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

"ami\_name": "{{user `ami\_name`}}",

"source\_ami\_filter": {

"filters": {

"virtualization-type": "hvm",

"name": "RHEL 8 x86\_64 HVM EBS \*",

"root-device-type": "ebs"

},

"owners": ["679593336670"],

"most\_recent": true

},

"run\_tags": {

"Name": "packer-build-image"

}

}],

"provisioners": [{

"type": "shell",

"inline": "while [ ! -f /var/lib/cloud/instance/boot-finished ]; do echo 'Waiting for cloud-init...'; sleep 1; done"

},

{

"type": "shell",

"script": "./provisioners/scripts/bootstrap.sh"

},

{

"type": "ansible",

"playbook\_file": "./provisioners/ansible/setup-server.yml"

}]

}

**STEP2: Create Scripts & Ansible Playbooks**

Let’s start with a playbook that will prepare a RHEL server

We’ll install Ansible on the VM using bash script which runs before Ansible playbooks are executed.

$ vim ./packer/provisioners/scripts/bootstrap.sh

#!/bin/bash

set -ex

# Updating

sudo yum update -y

# Add EPEL repository

sudo yum install -y epel-release

sudo yum install -y ansible

Make the script executable:

chmod +x ./packer/provisioners/scripts/bootstrap.sh

Create a file named setup-server.yml inside provisioners/ansible directory.

vim ~/projects/packer-ansible-aws/packer/provisioners/ansible/setup-server.yml

The contents for the file are:

---

- name: Installing the Nginx

hosts: 'all'

become: yes

become\_method: sudo

tasks:

- name: Add EPEL repository

yum:

name: epel-release

state: present

- name: Update all packages

yum:

name: "\*"

state: latest

- name: Install basic packages

yum:

name: ['epel-release', 'htop', 'tmux', 'screen', 'telnet', 'tree', 'wget', 'curl', 'git', 'python-firewall']

state: present

- name: Install nginx

yum:

name: nginx

state: present

- name: Insert Index Page

template:

src: ./provisioners/scripts/index.html

dest: /var/www/html/nginx.local

####### Set stats to check for the existance of the default web dir and conf file

- name: stat web directory

stat: path=/var/www/html/nginx.local

register: webdir\_stat

- name: stat nginx configuration

stat: path=/etc/nginx/conf.d/nginx.local.conf

register: siteconfig\_stat

- name: stat nginx certificate

stat: path={{nginx\_ssl\_dir}}/{{site\_name}}.crt

register: sitecert\_stat

######## Rename the web directory if the default exists

- name: reconfiguring web directory

command: mv /var/www/html/nginx.local /var/www/html/{{site\_name}}

when: webdir\_stat.stat.exists and site\_name != "nginx.local"

######## Reconfigure the site config file if the default already exists

- name: reconfiguring site

replace: dest=/etc/nginx/conf.d/nginx.local.conf regexp='nginx.local' replace='{{site\_name}}' owner=nginx group=nginx mode=644

when: siteconfig\_stat.stat.exists and site\_name != "nginx.local"

notify: restart nginx

######## Rename the nginx config file if the default already exists

- name: reconfiguring nginx

command: mv /etc/nginx/conf.d/nginx.local.conf /etc/nginx/conf.d/{{site\_name}}.conf

when: siteconfig\_stat.stat.exists and site\_name != "nginx.local"

notify: restart nginx

######## Generate a self signed certificate for the web service

- name: ensuring openssl is present

package: name={{item}} state=installed

with\_items:

- openssl

- name: creating nginx ssl directory

file: path={{nginx\_ssl\_dir}} state=directory owner=nginx group=nginx mode=0775 recurse=yes

- name: generate self signed certificate

command: openssl req -new -nodes -x509 -subj "/C=US/ST=US/L=US/O=\'{{site\_name}}\'/OU=Self Signed/CN=\'{{site\_name}}\'" -days 3650 -keyout {{nginx\_ssl\_dir}}/{{site\_name}}.key -out {{nginx\_ssl\_dir}}/{{site\_name}}.crt -extensions v3\_ca creates={{nginx\_ssl\_dir}}/{{site\_name}}.crt

when: sitecert\_stat.stat.exists == False

notify: restart nginx

######## Patch the nginx config file to enable SSL

- name: enabling ssl

replace:

dest: /etc/nginx/conf.d/{{site\_name}}.conf

regexp: '{{ item.regexp }}'

replace: '{{ item.replace }}'

owner: nginx

group: nginx

mode: 644

with\_items:

- { regexp: '#listen', replace: 'listen' }

- { regexp: '#ssl\_certificate\_key', replace: 'ssl\_certificate\_key' }

- { regexp: '#ssl\_certificate', replace: 'ssl\_certificate' }

- { regexp: '#ssl', replace: 'ssl' }

- name: Export the site

command: export SITE\_NAME="nginxtest.com"

handlers:

- name: restart nginx

service:

name: nginx

state: restarted

Step3: Run Packer build

cd ~/projects/packer-ansible-aws/packer

packer build -machine-readable packer-build.json | tee build\_artifact.txt