Ansible provisioning of Amazon EC2 instance.

1) On amazon fist create a security group that allows inbound ports like, 22 (SSH), 80 (HTTP) and 443 (HTTPS).

2) Create a ssh key pair for accessing AWS from your local machine.

3) In services, select 'Identity and Access Management' (IAM),
   Go to users → create new user → select the user and in permissions → attach policy, select, AdministratorAccess and AmazonAPIGatewayAdministrator.
   In 'security credentials', click on 'Create Access Key', take a note of the key as this will be used for authentication from your local machine.
4) While launching the instance you will need to select Machine Image (AMI), and important information is selecting the region (REGION=).

   Now on your ansible server, run,

```
$ sudo apt-get update
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install ansible
```

   Install boto using pip

```
$ sudo apt-get install python-pip
$ pip install -U boto
$ pip install -U boto3
```


Now, for the ansible to connect to Amazon EC2, download the ec2.ini and pc2.py files and place it in the folder from where you run the Ansible commands.

Download from,

https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.ini

https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.py

Now, we will use the access key and secret created on Amazon, user account.

While inside the /etc/ansible folder,

Clone the repository, https://github.com/ricardofontanelli/ansible-aws-ec2-provision.git

```
$ export AWS_ACCESS_KEY_ID=YOURKEYID
$ export AWS_SECRET_ACCESS_KEY=YOUSECRETKEY
```

Create a ssh key for login from local machine to ec2.

```
$ ssh-keygen -t rsa -f ~/.ssh/id_ansible2
```

Import this key into the Amazon key pairs option. Click on import key pair.

Setup CI CD pipeline with Ansible and Docker together:

Install Docker and Docker-compose on the ubuntu 14.04 VM.

Use below steps.

- `$ wget -qO- https://get.docker.com/pgp | apt-key add -`
- `$ echo deb http://get.docker.com/ubuntu docker main > /etc/apt/sources.list.d/docker.list`
- `$ apt-get install docker.io`

To get the Docker compose installed on Ubuntu, follow below steps.

- `$ sudo curl -o /usr/local/bin/docker-compose -L https://github.com/docker/compose/releases/download/1.11.2/docker-compose-$(uname -s)-$(uname -m)`

  Set the permissions.

- `$ sudo chmod +x /usr/local/bin/docker-compose`

  Check the docker compose version by running the command,

- `$ docker-compose -v`

Once docker is installed get Ansible installed,

- `$ sudo apt-add-repository -y ppa:ansible/ansible`
- `$ sudo apt-get update`
- `$ sudo apt-get install -y ansible`

  Install Python installer using,

- `$ sudo apt-get install python-software-properties`

  Once this is done run the command, ansible –version to check the ansible version.

Using ‘pip’ installer run below command,

- `$ pip install boto boto3` …. This will be used for using some of the Ansible module to interact with the Amazon web services API. These are the official Amazon SDKs used for Amazon web services.

Also install the AWS command line tool, using PIP installer.

- `$ pip install awscli` … this is enable us to interact with AWS from command line.

Now install GIT on the VM as we will be making use of the VCS tool.

- `$ apt-get install git`

Using ‘boto’ and ‘boto3’ we can use aws api to communicate with aws over command-line.

Installing aws-cli:

Follow these steps from the command line to install the AWS CLI using the bundled installer.

To install the AWS CLI using the bundled installer

1. *Download the AWS CLI Bundled Installer.*
   $ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
2. *Unzip the package.*
   a. *$ unzip awscli-bundle.zip*
3. *Note If you don't have unzip, use your Linux distribution's built in package manager to install it.*
4. *Run the install executable.*
   a. *$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws*

Note By default, the install script runs under the system default version of Python. If you have installed an alternative version of Python and want to use that to install the AWS CLI, run the install script with that version by absolute path to the Python executable.

For example:

$ sudo /usr/local/bin/python2.7 awscli-bundle/install -i /usr/local/aws -b /usr/ local/bin/aws

The installer installs the AWS CLI at /usr/local/aws and create the symlink aws at the /usr/local/bin directory.

Using the -b option to create a symlink eliminates the need to specify the install directory in the user's $PATH variable. This should enable all users to call the AWS CLI by typing aws from any directory.

To see an explanation of the -i and -b options, use the -h option:

$ ./awscli-bundle/install -h

Install the AWS CLI

```
$ aws ec2 terminate-instances --instance-ids $(aws ec2 describe-instances
--filters  "Name=instance-state-
name,Values=pending,running,stopped,stopping" --query
"Reservations[].Instances[].[InstanceId]" --output text | tr '\n' ' ')
```

Terminate aws instance from command line using below command,

```
$ aws ec2 terminate-instances --instance-ids <instance-
id>
```