# Solution for Devops Task:

**Task: Provisioning**

Please create an automated process with a tool/scripting language of your own preference relying on
the virtualization software of your choice, which does the following:

• Provisions the fully functional VM running any major Linux distribution of choice
• Sets up docker and run a web server docker container on the VM (let it be Apache, Nginx or anything else you feel comfortable with) which serves either simple static web page (but not the default web server landing page), or even web application of your choice
• Create a check if the web application is available and running fine
• If possible, configure the firewall to allow only access to SSH and web server ports

**Solution is attached to https://github.com/angautam/ansible-ec2-docker-nginx**

This solution is fully executable. I am using Ansible here as an automation tool to provision AWS EC2 instance as ansible has specific modules and also agentless which will help me to achieve this task with lesser time and more organized. I could have used vagrant to launch Virtual machines(virtualized environment) but instead chose the AWS cloud to effectively showcase the firewall rules etc ..

**Below are the outline to design the ansible playbooks :**

-- provision EC2 instance with base AMI image
-- region, instance_type, security group etc as variables.
-- Firewall ports to be opened for only ssh port 22, webserver port 80 & 443(if needed)
-- to make ssh connection, we need to store the public ip and host string in /etc/ansible/hosts file
-- Install the prerequisites to install docker on red hat as containerd needs to be installed specifially and for that AWS RHEL extras repository needs to be enabled first
-- Enable the docker repo to enable only stable version of docker we need and disable others.
-- Install docker using root user
-- Start the docker service.
-- assign docker group to ec2-user and reset the connection so that when we login again ec2-user will have docker group effective.
-- ssh again and copy Dockerfile and index.html page to ec2 instance
-- build the docker image and run it on port 80
-- use ansible module to check the web page for HTTP 200 code and content.

**Below are the steps I took to achieve this task.**

1) Took a VM and installed ansible and required python packages.

Red Hat Enterprise Linux 64 bit 7.6

**Pre-requisites**

#Pip is not available in RHEL core repositories. To install pip we need to enable the EPEL repository:

#sudo yum install epel-release

#Once the EPEL repository is enabled we can install pip and all of it's dependencies with the following command:

#sudo yum install python-pip

#Install development tools

#sudo yum install python-devel

#Development tools are required for building Python modules, you can install them with:

#sudo yum groupinstall 'development tools'

#sudo pip install --upgrade pip

#sudo pip install boto3

#sudo pip install boto

#sudo yum install ansible

2) Once ansible is installed , make below entry to be done on /etc/ansible/hosts file

==========
[local]
localhost

[webserver]
==========

3) Place all the files under current directory and ensure we change variables like below :

git clone https://github.com/angautam/ansible-ec2-docker-nginx.git

Edit in launchec2-ansible.yml file

```
# Necessary Variables for creating/provisioning the EC2 Instance
   vars:
     instance_type: t2.micro
     security_group: ansible-automation # Change the security group name here
     image: ami-0b500ef59d8335eee # This is an AMI Redhat Linux 64 bit distribution
     keypair: linux_aws # This is one of my keys that i already have in AWS
     region: us-east-2 # Change the Region
     count: 1
     hostpath: /etc/ansible/hosts
     hoststring: "ansible_ssh_user=ec2-user ansible_ssh_private_key_file=/root/linux_aws.pem"
```

Ensure host string and our key path is updated above.

The playbook lets us provision EC2 instance on AWS cloud and install docker and run nginx with a static hello world html page and at the end it will validate whether the application is working fine.

The image I am using here is of Red Hat distribution AMI image from AWS cloud.

Following are the steps to run the ansible playbooks.

1. export the secret key and access key of AWS account on terminal by following below:

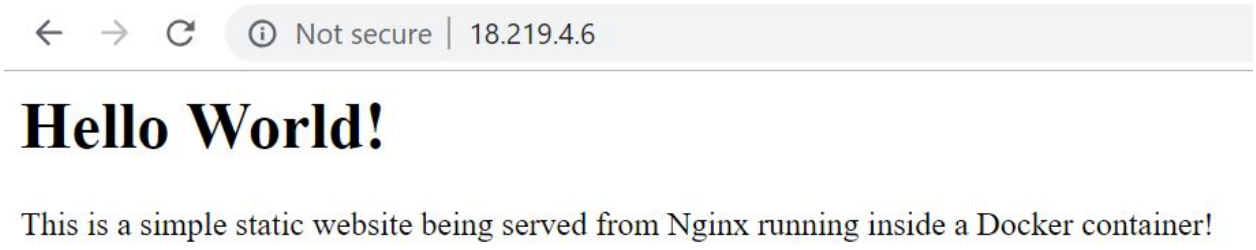
   export AWS_ACCESS_KEY_ID='<your_access_key>'

   export AWS_SECRET_ACCESS_KEY='<your_secret_key>'

2. run "ansible-playbook launchec2-ansible.yml" [This will provision EC2 instance in your account]
3. after this is done, ensure your hosts[/etc/ansible/hosts] entry gets updated with public ip and user key details just under tag [webserver]
4. run "ansible-playbook docker-install.yml" to install docker on newly provisioned ec2 instance.

5. then run "ansible-playbook rundockercontainer-curl.yml"

We can go to browser and check use the newly provisioned public IP and see the hello world static page is loading.




All the task provided is achieved.

-- launchec2-ansible.yml file is launching ec2 instance with firewall rule to allow only ssh port and webserver ports.

-- docker-install.yml file is installing dockers

-- rundockercontainer-curl.yml file is helping us to build webserver nginx image with static html page and running as container.

-- Finally i am using ansible uri module to validate or check if the application is available and running fine.

```
- uri:
    url: http://localhost
    return_content: yes
  register: this
  failed_when: "'Hello World' not in this.content"
```