## horizontal line



Using ansible to deploy sample nginx/python application into kubernetes cluster

**Prerequisites:**

1. **Create EC2 Instance**
2. **Install kubernets and Minikube**
3. **Install Docker**
4. **Install pip ansible**
5. **Install pip kubernets for kubernets modules**
6. **Install awscli**

**STEP-1**

**Build docker image for sample nginx/python and push to aws elastic container registry(ECR) using docker cli**.

1**.Build your Docker image for the sample nginx/python application using the necessary Dockerfile and dependencies.**

**mkdir docker**

**cd docker**

**Create a docker file “Dockerfile”**

FROM nginx:latest

EXPOSE 8080

CMD ["nginx", "-g", "daemon off;"]

Build a docker image

**- docker build -t <image\_name> .**

2.Create a Elastic Container Registry private or public

Login into AWS ECR

**aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 400538885032.dkr.ecr.ap-south-1.amazonaws.com**

Tag the Docker image with the ECR repository URI

**docker tag karun:latest 400538885032.dkr.ecr.ap-south-1.amazonaws.com/karun:latest**

Push the Docker Image into ECR

**docker push 400538885032.dkr.ecr.ap-south-1.amazonaws.com/karun:latest**

**STEP–2**

**Create ansible role to deploy nginx application into kubernetes cluster**

**1.** Createansible role using following command:

**ansible-galaxy init nginxrole**

**-nginxrole was created**

Under this directory following structure

1. -- ngnixrole
2. |-- defaults
3. | `-- main.yml
4. |-- handlers
5. | `-- main.yml
6. |-- meta
7. | `-- main.yml
8. |-- README.md
9. |-- tasks
10. | `-- main.yml
11. |-- tests
12. | |-- inventory
13. | `-- test.yml
14. `-- vars
15. `-- main.yml

3.In ‘tasks/main.yml’ add the following task:

Write a Nginix Deployment and Service

- name: Deploy NGINX Deployment

kubernetes.core.k8s:

state: present

definition:

apiVersion: apps/v1

kind: Deployment

metadata:

namespace: default

name: nginx-app

labels:

app: nginx-app

spec:

replicas: 2

selector:

matchLabels:

app: nginx-app

template:

metadata:

labels:

app: nginx-app

spec:

containers:

- name: nginx-app

image: 543066576745.dkr.ecr.us-east-2.amazonaws.com/finaltask:latest

imagePullPolicy: Always

ports:

- containerPort: 80

imagePullSecret:

* name: regcred

- name: Create NGINX Service

kubernetes.core.k8s:

state: present

definition:

apiVersion: v1

kind: Service

metadata:

namespace: default

name: nginx-app

labels:

name: nginx-app

spec:

selector:

app: nginx-app

type: NodePort

ports:

- protocol: TCP

port: 8080

nodePort: 30010

targetPort: 80

**STEP-3**

**Create ansible playbook to deploy nginx application**

Create a ansible playbook file **‘play.yml’**

**---**

**- name: Deploy NGINX Application**

**hosts: localhost**

**become: yes**

**roles:**

**- nginxrole**

**Run the Ansible Playbook to Deploy NGINX Application**

Following this command:

ansible-playbook play.yml

**STEP-4**

**Container image should be pulled from aws elastic container registry(ECR)**

After running the Ansible playbook, the NGINX application should be deployed in your Kubernetes cluster. You should be able to access the NGINX website using the URL **http://65.0.127.25:8080**, where **65.0.127.25** is the IP address of one of your Kubernetes nodes, and **8080** is the NodePort defined in the Service manifest.

Checking whether the pods ,services ,deployments created or not

Following these commands

**- kubectl get pods**

**- kubectl get svc**

**- kubectl get deployments**

The **aws ecr get-login-password --region us-east-2** command is used to retrieve an authentication token that can be used to log in to the Amazon Elastic Container Registry (ECR) in the us-east-2 AWS region. You find a key

That you run with base64 echo -n “....” | base64 and after we store the key in this file

**/root/.docker/config.json**

**{**

**"auths": {**

**"your\_account\_id.dkr.ecr.your\_region.amazonaws.com": {**

**"auth": "your\_base64\_encoded\_credentials"**

**}**

**}**

**}**

And after that we create a secret key following this command

**kubectl create secret docker-registry regcred \ --docker-server=${AWS\_ACCOUNT}.dkr.ecr.${AWS\_REGION}.**[**amazonaws.com**](http://amazonaws.com/) **\ --docker-username=AWS \ --docker-password=$(aws ecr get-login-password) \**

l **Nginx Deployment output:**

Run this command andYou should be able to access the NGINX website using the URL **http://65.0.127.25:8080**

**kubectl port-forward --address 0.0.0.0 svc/<Service-Name> 8080:8080**

