Cloud Engineer Assessment Task 1

Problem Statement:

Overview

Deploy a Python web application in a Cloud-based Kubernetes solution.

Application Functionality

The python web application should just have a simple static page that displays the values stored in the environment variables “ATC_USERNAME” & “ATC_PASSWORD”.

Expected Outcome

Your solution should contain the following items.

- IAC templates (preferrably terraform) used to provision the cloud infrastructure
- Kubernetes deployment files
- Python program used to host the application
- Dockerfile used to containerize the application
- Clear documentation on how to deploy your solution

Note

If the candidate is comfortable with any other programming language other than python such as nodejs, ruby, etc., he/she is free to use them provided the functionality of the application should be the same.

Solution Statement: -

Deployment of a sample application called “Weather App” on an EKS cluster.

1. The EKS infrastructure has been deployed using terraform on the AWS account.
2. The deployment has been done using the kubernetes manifest files.
3. The application has been exposed to the public using an alb controller and the same is accessible using the ALB endpoint.

Setting up of EKS infrastructure using Terraform

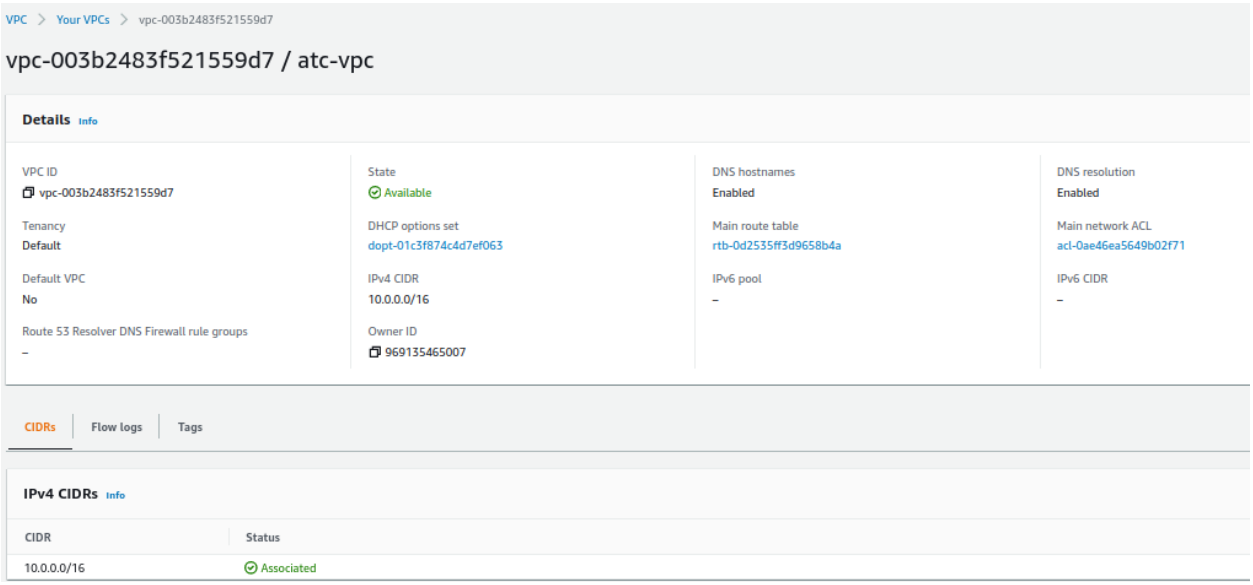
The terraform files have been uploaded to the Google Drive Folder - EKS-Terraform.

Commands used to Provision the Infrastructure:

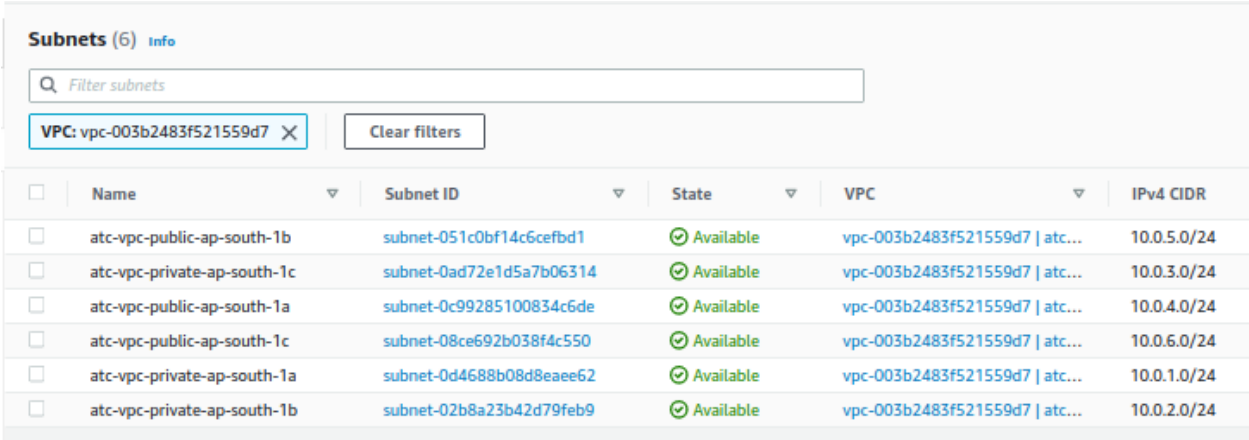
terraform init
#terraform plan
#terraform apply

Screenshot of the Solution:

VPC:



Subnets:



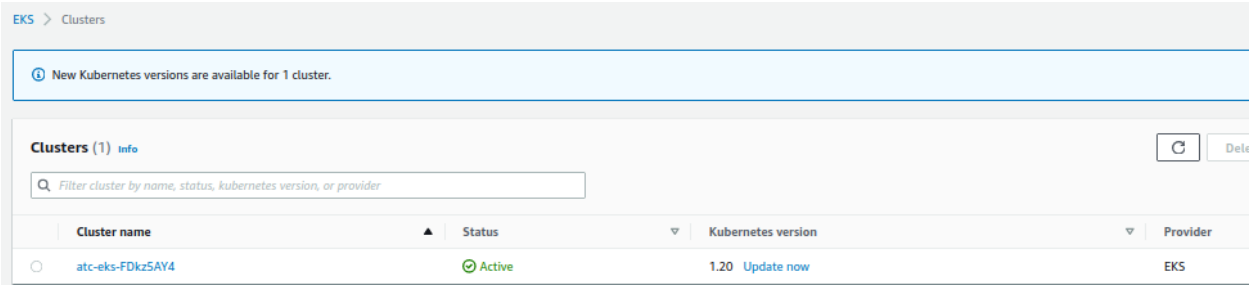
Subnets (6) Info

Filter subnets

VPC: vpc-003b2483f521559d7 | Clear filters

Name	Subnet ID	State	VPC	IPv4 CIDR
atc-vpc-public-ap-south-1b	subnet-051c0bf14c6cefbd1	Available	vpc-003b2483f521559d7 | atc...	10.0.5.0/24
atc-vpc-private-ap-south-1c	subnet-0ad72e1d5a7b06314	Available	vpc-003b2483f521559d7 | atc...	10.0.3.0/24
atc-vpc-public-ap-south-1a	subnet-0c99285100834c6de	Available	vpc-003b2483f521559d7 | atc...	10.0.4.0/24
atc-vpc-public-ap-south-1c	subnet-08ce692b038f4c550	Available	vpc-003b2483f521559d7 | atc...	10.0.6.0/24
atc-vpc-private-ap-south-1a	subnet-0d4688b08d8eaee62	Available	vpc-003b2483f521559d7 | atc...	10.0.1.0/24
atc-vpc-private-ap-south-1b	subnet-02b8a23b42d79feb9	Available	vpc-003b2483f521559d7 | atc...	10.0.2.0/24

EKS Cluster:



EKS > Clusters

New Kubernetes versions are available for 1 cluster.

Clusters (1) Info

Filter cluster by name, status, kubernetes version, or provider

Cluster name	Status	Kubernetes version	Provider
atc-eks-FDkz5AY4	Active	1.20 Update now	EKS

```
vishnu-tf@ttnpl-3760:~$ eksctl get cluster atc-eks-FDkz5AY4
2022-02-18 18:05:33 [ℹ]  eksctl version 0.72.0
2022-02-18 18:05:33 [ℹ]  using region ap-south-1
NAME                    VERSION STATUS  CREATED                 VPC                     SUBNETS
                                SECURITYGROUPS          PROVIDER
atc-eks-FDkz5AY4        1.20    ACTIVE  2022-02-17T09:04:07Z    vpc-003b2483f521559d7   subnet-02b8a23b42d79feb9
,subnet-0ad72e1d5a7b06314,subnet-0d4688b08d8eaee62    sg-01191eaa8b307ec88    EKS
vishnu-tf@ttnpl-3760:~$
```

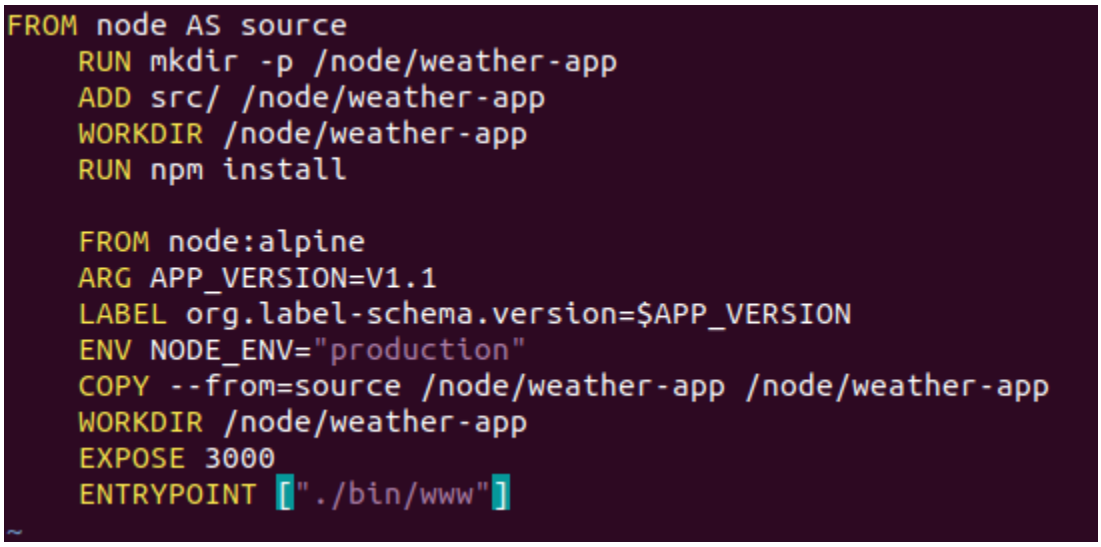
Application Code:

- Uploaded on the Google Drive Folder - Weather-App

Docker File:

```
FROM node AS source
   RUN mkdir -p /node/weather-app
   ADD src/ /node/weather-app
   WORKDIR /node/weather-app
   RUN npm install

   FROM node:alpine
   ARG APP_VERSION=V1.1
   LABEL org.label-schema.version=$APP_VERSION
   ENV NODE_ENV="production"
   COPY --from=source /node/weather-app /node/weather-app
   WORKDIR /node/weather-app
   EXPOSE 3000
   ENTRYPOINT ["./bin/www"]
```



Build the image and push it into the docker hub

#sudo docker image build -t vishnunvv/weather-app:2.0
#docker push vishnunvv/weather-app

Application Deployment

- Deployment eks manifests are uploaded on the google drive folder EKS Manifests

Application Deployment file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: weather-app
spec:
  selector:
    matchLabels:
      app: weather-app
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: weather-app
    spec:
      containers:
      - name: weather-app
        image: vishnunvv/weather-app:2.0
        ports:
        - containerPort: 3000
```

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
weather-app   2/2     2            2           26h
vishnu-tf@ttnpl-3760:~/eks-manifests$
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl get pod
NAME                           READY   STATUS    RESTARTS   AGE
weather-app-6cb44b7b7b-jh5m7   1/1     Running   0          26h
weather-app-6cb44b7b7b-qxcf9   1/1     Running   0          26h
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

Configuring the Ingress Controller

Create IAM OIDC provider

```
eksctl utils associate-iam-oidc-provider \
    --region ap-south-1 \
    --cluster atc-eks-FDkz5AY4 \
    --approve
```

Create a Kubernetes service account named alb-ingress-controller in the kube-system namespace

Create ClusterRole, ClusterRoleBinding & ServiceAccount
kubectl apply -f
https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-ingress-controller/master/docs/examples/rbac-role.yaml

List Service Accounts
kubectl get sa -n kube-system

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl get sa -n kube-system | grep -i ingress
alb-ingress-controller               1         17h
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

Describe Service Account alb-ingress-controller
kubectl describe sa alb-ingress-controller -n kube-system

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl describe sa alb-ingress-controller -n kube-system
Name:                alb-ingress-controller
Namespace:           kube-system
Labels:              app.kubernetes.io/managed-by=eksctl
                     app.kubernetes.io/name=alb-ingress-controller
Annotations:         eks.amazonaws.com/role-arn: arn:aws:iam::969135465007:role/eksctl-atc-eks-FDkz5AY4-addon-iamserviceacco-Role1-BOT8PMVO0CG6
Image pull secrets:  <none>
Mountable secrets:   alb-ingress-controller-token-xl9rc
Tokens:              alb-ingress-controller-token-xl9rc
Events:              <none>
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

Create IAM Policy for ALB Ingress Controller

aws iam create-policy \
    --policy-name ALBIngressControllerIAMPolicy \
    --policy-document
https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-ingress-controller/master/docs/examples/iam-policy.json

Note:- Experienced some error when we create the IAM policy using the above Json policy document using CLI, so created from the management console.

Create an IAM role for the ALB Ingress Controller and attach the role to the service account

    eksctl create iamserviceaccount \
    --region ap-south-1 \
    --name alb-ingress-controller \
    --namespace kube-system \
    --cluster atc-eks-FDkz5AY4 \
    --attach-policy-arn
arn:aws:iam::969135465007:policy/ALBIngressControllerIAMPolicy \
    --override-existing-serviceaccounts \
    --approve

Verify using eksctl cli

eksctl get iamserviceaccount --cluster atc-eks-FDkz5AY4

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ eksctl  get iamserviceaccount --cluster atc-eks-FDkz5AY4
2022-02-18 18:52:06 [ℹ]  eksctl version 0.72.0
2022-02-18 18:52:06 [ℹ]  using region ap-south-1
NAMESPACE       NAME                    ROLE ARN
kube-system     alb-ingress-controller  arn:aws:iam::969135465007:role/eksctl-atc-eks-FDkz5AY4-addon-iamserviceacco-Role1-BOT8PMVO0CG6
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

Deploy ALB Ingress Controller

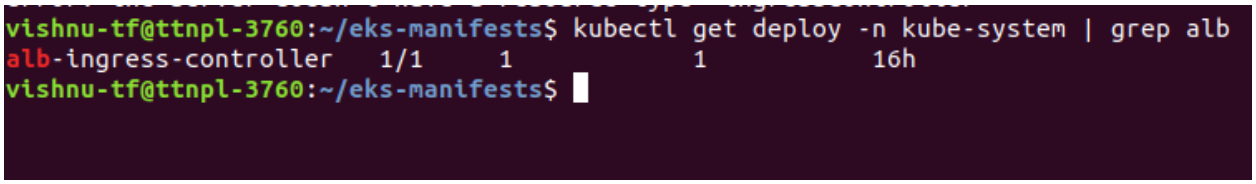
kubectl apply -f
https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-ingress-controller/master/docs/examples/alb-ingress-controller.yaml

Verify Deployment
kubectl get deploy -n kube-system

Edit ALB Ingress Controller Manifest

kubectl edit deployment.apps/alb-ingress-controller -n kube-system

```
    spec:
      containers:
      - args:
        - --ingress-class=alb
        - --cluster-name=atc-eks-FDkz5AY4
```



Ingress Manifest

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-weather-app
  labels:
    app: weather-app
  annotations:
    kubernetes.io/ingress.class: "alb"
    alb.ingress.kubernetes.io/scheme: internet-facing
spec:
  rules:
    - http:
        paths:
          - path: /*
            backend:
              serviceName: weather-app
              servicePort: 8080
```

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl get ingress
NAME                 CLASS    HOSTS   ADDRESS                                                          PORTS   AGE
ingress-weather-app  <none>   *       2c1f58c5-default-ingresswe-928c-382980542.ap-south-1.elb.amazonaws.com   80      8h
vishnu-tf@ttnpl-3760:~/eks-manifests$
vishnu-tf@ttnpl-3760:~/eks-manifests$
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ kubectl describe ingress ingress-weather-app
Name:             ingress-weather-app
Namespace:        default
Address:          2c1f58c5-default-ingresswe-928c-382980542.ap-south-1.elb.amazonaws.com
Default backend:  default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host        Path  Backends
  ----        ----  --------
  *
              /*   weather-app:8080 (10.0.1.206:3000,10.0.2.10:3000)
Annotations:  alb.ingress.kubernetes.io/scheme: internet-facing
              kubernetes.io/ingress.class: alb
Events:       <none>
vishnu-tf@ttnpl-3760:~/eks-manifests$
```

OUTPUT

The Application will show the Current Temperature when we input the City Name (The values will be random not Original)

```
vishnu-tf@ttnpl-3760:~/eks-manifests$ curl -IL 2c1f58c5-default-ingresswe-928c-382980542.ap-south-1.elb.amazonaws.com
HTTP/1.1 200 OK
Date: Fri, 18 Feb 2022 16:53:30 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 619
Connection: keep-alive
X-Powered-By: Express
ETag: W/"26b-X0fAt+1YhQIIWMdDb/5JeJjIRs8"

vishnu-tf@ttnpl-3760:~/eks-manifests$
```