For this taSK We are using API of Open Weather Map's public API and created our own microservice

Using the Fetch API to fetch Open Weather Map's public API for JSON data for US cities

Command line weather app using Node.js and [OpenWeatherMap](http://openweathermap.org/).

This library provides an interface to query the current weather and temperature using a zipcode location and the OpenWeatherMap [API](http://openweathermap.org/current).

* Create an app, package it in a container and publish to a Docker registry:-
* Creation of Dockerfile

|  |
| --- |
| FROM node:7 |
|  |  |
|  | RUN apt-get update && apt install -y git && \ |
|  | git clone https://github.com/thedevopsguru/caas\_task\_microservice/ /home/ec2-user/test\_repo |
|  | WORKDIR /app |
|  | COPY package.json app.js config.json printer.js weather.js ./ |
|  | RUN npm install |
|  | COPY . /app |
|  | EXPOSE 8081 |
|  | RUN node app.js & |

**Docker Build Image and run**

docker build -t pokharia83/microservices:v1 .

**Publish your image to a docker container registry**

**docker login**

**docker build -t pokharia83/microservices:v1 .**

**Tag Image of container**

**docker tag pokharia83/microservices:v1 pokharia83/microservices:v1**

**Push container**

**docker push pokharia83/microservices:v1**

**Create Deployment yaml for the deployment of the microservices and scale the services**

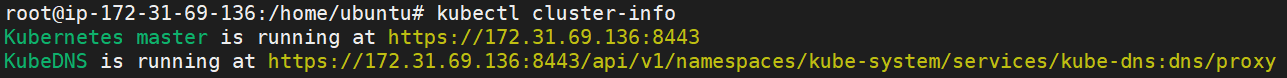
**kubectl create deployment microservices --image=pokharia83/microservices:v1 --dry-run -o yaml > deployment.yaml**

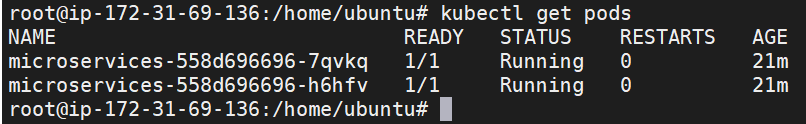
####### deployment.yml###########

|  |
| --- |
| apiVersion: apps/v1 |
|  | kind: Deployment |
|  | metadata: |
|  | creationTimestamp: null |
|  | labels: |
|  | app: microservices |
|  | name: microservices |
|  | spec: |
|  | replicas: 2 |
|  | selector: |
|  | matchLabels: |
|  | app: microservices |
|  | strategy: {} |
|  | template: |
|  | metadata: |
|  | creationTimestamp: null |
|  | labels: |
|  | app: microservices |
|  | spec: |
|  | containers: |
|  | - image: pokharia83/microservices:v1 |
|  | name: microservices |
|  | command: [ "sleep" ] |
|  | args: [ "infinity" ] |
|  | ports: |
|  | - containerPort: 8081 |
|  | resources: {} |
|  | status: {} |

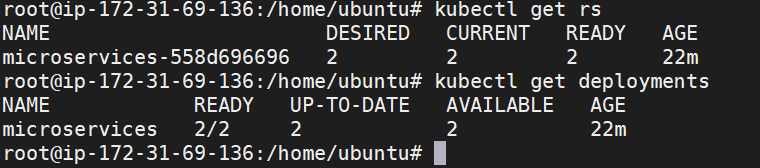
kubectl apply -f deployment.yaml

**Run your container on the cluster**



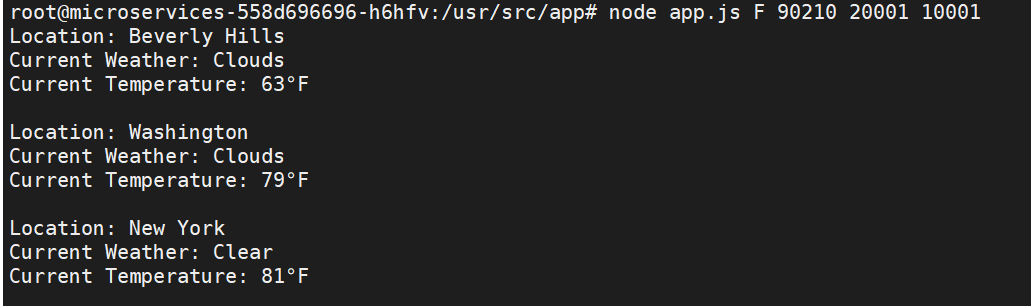


kubectl get deployment microservice\_weather



Kubernetes updates the pod and pulls the container and scaled up.

Login inside one container and check if the API is giving proper weather and loation.



Kubernates Installation of worker and master nodes:-