**Microservices**

Assignment 1

**All microservices reporting tracing data to a Jaegar service and one of the microservice integrated a MySQL (or any other DB) backed up by a PersistentVolume.**

* The purpose of this assignment is to create working REST microservices which will be deployed to Kubernetes cluster (on Google Kubernetes Engine) using docker images and all of these microservices will report their tracing information to a Jaegar service.
* Create 3 microservices (in language of your choice):-
  + User Service – Will provide a REST endpoint giving details of user (**it should return your name and email id being fetched from a database of your choice**)
    - GET /user/1

{

"name": "John",

"age": "23",

"email": "john.doe@google.com"

}

* + Orders Service – Will give a list of all orders of a user. (it should return a predefined list, no database needed in this microservice)
    - GET /orders/1

{

"orders": [

{

"orderId": 1,

"orderAmount": 250,

"orderDate": "14-Apr-2020"

},

{

"orderId": 2,

"orderAmount": 450,

"orderDate": "15-Apr-2020"

}

]

}

* + Aggregator Service – Will aggregate the response from user and orders service to give following response (should get data from user and order service’s rest endpoint)
    - GET /orderdetails/1

{

"userDetails": {

"name": "John",

"age": "23",

"email": "john.doe@google.com"

},

"orders": [

{

"orderId": 1,

"orderAmount": 250,

"orderDate": "14-Apr-2020"

},

{

"orderId": 2,

"orderAmount": 450,

"orderDate": "15-Apr-2020"

}

]

}

* Build and push docker images to dockerhub for all the microservices.
* Create a Google Kubernetes engine cluster on GKE
* Deploy all the 3 microservices on GKE using separate YAML files for each deployment.
* Expose the services in following manner:-
  + Expose only Aggregator microservice externally to the internet
  + User and Order service should be exposed only internally using the suitable kind of “Kubernetes Service” and both should be consumed internally by the Aggregator microservice.
* Deploy a Jaegar service on your Kubernetes cluster (you can do this using GKE UI)
* Integrate all of your microservices to report their tracing data to this Jaegar service internally.
* MySQL(or any other database) should be deployed inside Kubernetes cluster with a PersistentVolume and credentials in a Kubernetes secret object.
* Deliverables:-
  + Source code of all services committed on Github
  + Dockerhub image links of all services
  + A document explaining your deployment design on Kubernetes (prefer some diagram explaining the flow)
  + YAML files of all your deployments/PersistentVolumeClaims and Secrets on Kubernetes.
  + Screenshots of all Running Deployments/Services/ConfigMaps/PersistentVolume Claims and Secrets(if any) that you have created for your assignment. (screenshots should clearly show the Service type, ip addreses, YAML of configmap if created).
  + Screenshot of REST endpoint of aggregator service (clearly showing the IP address of service being invoked)
  + Screenshot of all Jaegar UI running in GKE with tracing data of your microservices.
* You may delete your Kubernetes cluster after the deliverables have been captured to avoid any additional cost.