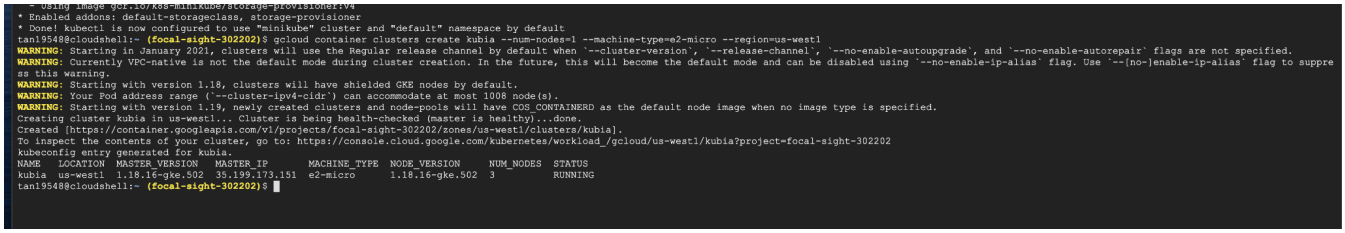
1.
gcloud container clusters create kubia --num-nodes=1 --machine-type=e2-micro --region=us-west1



2. Create yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
spec:
  selector:
    matchLabels:
      app: mongodb
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        # by default, the image is pulled from docker hub
        - image: mongo
          name: mongo
          ports:
            - containerPort: 27017
          volumeMounts:
            - name: mongodb-data
              mountPath: /data/db
      volumes:
        - name: mongodb-data
          gcePersistentDisk:
            pdName: mongodb
            fsType: ext4
```

3. Apply yaml

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f mongodb-deployment.yaml
deployment.apps/mongodb-deployment created
```

4. Create service yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  type: LoadBalancer
  ports:
      # service port in cluster
    - port: 27017
      # port to contact inside container
      targetPort: 27017
  selector:
app: mongodb
```

5. Apply yaml

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f mongodb-service.yaml
service/mongodb-service created
```

6. Check pods

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
mongodb-deployment-554cbb9965-hgcft     1/1     Running   0          13s
```

7. Check services

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl get services
NAME              TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)           AGE
kubernetes        ClusterIP      10.3.240.1     <none>          443/TCP           32m
mongodb-service   LoadBalancer   10.3.244.234   34.82.15.163    27017:31970/TCP   15m
```

8. Run in bash

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl exec -it mongodb-deployment-554cbb9965-hgcft -- bash
root@mongodb-deployment-554cbb9965-hgcft:/#
```

9. Mongo external ip

```
root@mongodb-deployment-554cbb9965-hgcft:/# mongo 34.82.15.163
MongoDB shell version v4.4.5
connecting to: mongodb://34.82.15.163:27017/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("af0d3b69-569e-430e-833f-5309914666b5") }
MongoDB server version: 4.4.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2021-04-12T06:01:01.924+00:00: Using the XFS filesystem is strongly recommended with the WiredTig
        2021-04-12T06:01:03.371+00:00: Access control is not enabled for the database. Read and write acc
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

10. Install mongodb

```
tan19548@cloudshell:~ (focal-sight-302202)$ npm install mongodb

added 18 packages, and audited 19 packages in 2s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.8.0 -> 7.9.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.9.0
npm notice Run npm install -g npm@7.9.0 to update!
npm notice
tan19548@cloudshell:~ (focal-sight-302202)$ npm link mongodb

removed 17 packages, changed 1 package, and audited 3 packages in 1s

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.8.0 -> 7.9.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.9.0
npm notice Run npm install -g npm@7.9.0 to update!
npm notice
tan19548@cloudshell:~ (focal-sight-302202)$
```

11. Create mongodb.js

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://34.82.15.163/mydb"
// Connect to the db

MongoClient.connect(url,{ useNewUrlParser: true, useUnifiedTopology: true },
function(err, client){
     if (err)
         throw err;

           // create a document to be inserted
     var db = client.db("studentdb");
     const docs = [
             { student_id: 11111, student_name: "Bruce Lee", grade: 84},
             { student_id: 22222, student_name: "Jackie Chen", grade: 93 },
             { student_id: 33333, student_name: "Jet Li", grade: 88}
     ]
     db.collection("students").insertMany(docs, function(err, res){
         if(err) throw err;
         console.log(res.insertedCount);
         client.close();
     });
     db.collection("students").findOne({"student_id": 11111},
     function(err, result){
         console.log(result);
     });
});
```

12. Create studentServer.js

```
var server = http.createServer(function (req, res) {
  var result;
  // req.url = /api/score?student_id=11111
  var parsedUrl = url.parse(req.url, true);

  var student_id = parseInt(parsedUrl.query.student_id);
  // match req.url with the string /api/score
  if (/^\/api\/score/.test(req.url)) {
    // e.g., of student_id 1111

    MongoClient.connect(uri,{ useNewUrlParser: true, useUnifiedTopology:
true }, function(err, client){
        if (err)
          throw err;
        var db = client.db("studentdb");
        db.collection("students").findOne({"student_id":student_id},
(err, student) => {
          if(err)
               throw new Error(err.message, null);

          if (student) {
               res.writeHead(200, { 'Content-Type': 'application/json'
})
               res.end(JSON.stringify(student)+ '\n')
          }else {
               res.writeHead(404);
               res.end("Student Not Found \n");
          }
        });
  } else {
  res.writeHead(404);
  res.end("Wrong url, please try again\n");
  }
});
server.listen(8080);
```

13. Create docker file

```
FROM node:7
ADD studentServer.js /studentServer.js
ENTRYPOINT ["node", "studentServer.js"]
RUN nmp install mongodb
```

14. Build docker

```
Removing intermediate container 4069adb7d383
 ---> 5046a7d95a33
Successfully built 5046a7d95a33
Successfully tagged helloworld102444/studentserver:latest
tan19548@cloudshell:~ (focal-sight-302202)$
```

15. Docker push

```
tan19548@cloudshell:~ (focal-sight-302202)$ docker push helloworld102444/studentserver
Using default tag: latest
The push refers to repository [docker.io/helloworld102444/studentserver]
640c3baee9df: Pushed
5354b81d303e: Pushed
ab90d83fa34a: Mounted from library/node
8ee318e54723: Mounted from library/node
e6695624484e: Mounted from library/node
da59b99bbd3b: Mounted from library/node
5616a6292c16: Mounted from library/node
f3ed6cb59ab0: Mounted from library/node
654f45ecb7e3: Mounted from library/node
2c40c66f7667: Mounted from library/node
latest: digest: sha256:d340f405757f42f6d6227a9546a0d9c8112fe3c495d83f1d1c66c69710fc0072 size: 2424
```

16. Create bookshelf.py

```python
from flask import Flask, request, jsonify
from flask_pymongo import PyMongo
from flask import request
from bson.objectid import ObjectId
import socket
import os

app = Flask(__name__)
app.config["MONGO_URI"] = "mongodb://"+os.getenv("MONGO_URL")+"/"+os.getenv("MONGO_DATABASE")
app.config['JSONIFY_PRETTYPRINT_REGULAR'] = True
mongo = PyMongo(app)
db = mongo.db

@app.route("/")
def index():
    hostname = socket.gethostname()
    return jsonify(
        message="Welcome to bookshelf app! I am running inside {} pod!".format(hostname)
    )

@app.route("/books")
def get_all_tasks():
    books = db.bookshelf.find()
    data = []
    for book in books:
        data.append({
            "id": str(book["_id"]),
            "Book Name": book["book_name"],
            "Book Author": book["book_author"],
            "ISBN" : book["ISBN"]
        })
    return jsonify(
        data
    )

@app.route("/book", methods=["POST"])
def add_book():
    book = request.get_json(force=True)
    db.bookshelf.insert_one({
        "book_name": book["book_name"],
        "book_author": book["book_author"],
        "ISBN": book["isbn"]
        })
    return jsonify(
        message="Task saved successfully!"
    )

@app.route("/book/<id>", methods=["PUT"])
```

17. Create docker file

```
FROM python:alpine3.7
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
ENV PORT 5000
EXPOSE 5000
ENTRYPOINT [ "python3" ]
CMD [ "bookshelf.py" ]
```

18. Build docker

```
Successfully built e3e5515bbc30
Successfully tagged helloworld102444/bookshelf:latest
tan19548@cloudshell:~ (focal-sight-302202)$
```

19. Docker push

```
tan19548@cloudshell:~ (focal-sight-302202)$ docker push helloworld102444/bookshelf
Using default tag: latest
The push refers to repository [docker.io/helloworld102444/bookshelf]
11568cdee01e: Pushed
a12c8f092cae: Pushed
5fa31f02caa8: Mounted from library/python
88e61e328a3c: Mounted from library/python
9b77965e1d3f: Mounted from library/python
50f8b07e9421: Mounted from library/python
629164d914fc: Mounted from library/python
latest: digest: sha256:24b1595c801014cbe536ca31ccdcf0df0083c41e5589b806a2ad0f158ed01485 size: 1790
```

20. Create studentserver-configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: studentserver-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 34.82.15.163
  MONGO_DATABASE: mydb
```

21. Create bookshelf-configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: bookshelf-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 34.82.15.163
  MONGO_DATABASE: mydb
```

22. Create studentserver-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
  labels:
    app: studentserver-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - image: helloworld102444/studentserver
          imagePullPolicy: Always
          name: web
          ports:
            - containerPort: 8080
          env:
            - name: MONGO_URL
              valueFrom:
                configMapKeyRef:
                  name: studentserver-config
                  key: MONGO_URL
            - name: MONGO_DATABASE
              valueFrom:
                configMapKeyRef:
                  name: studentserver-config
                  key: MONGO_DATABASE
```

23. Create bookshelf-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: bookshelf-deployment
  labels:
    app: bookshelf-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: bookshelf-deployment
  template:
    metadata:
      labels:
        app: bookshelf-deployment
    spec:
      containers:
        - image: helloworld102444/bookshelf
          imagePullPolicy: Always
          name: bookshelf-deployment
          ports:
            - containerPort: 5000
          env:
            - name: MONGO_URL
              valueFrom:
                configMapKeyRef:
                  name: bookshelf-config
                  key: MONGO_URL
            - name: MONGO_DATABASE
              valueFrom:
                configMapKeyRef:
                  name: bookshelf-config
                  key: MONGO_DATABASE
```

24. Create studentserver-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: web
spec:
  type: LoadBalancer
  ports:
      # service port in cluster
    - port: 8080
      # port to contact inside container
      targetPort: 8080
  selector:
    app: web
```

25. Minikube enable ingress

```
tan19548@cloudshell:~ (focal-sight-302202)$ minikube addons enable ingress
  - Using image jettech/kube-webhook-certgen:v1.3.0
  - Using image us.gcr.io/k8s-artifacts-prod/ingress-nginx/controller:v0.40.2
  - Using image jettech/kube-webhook-certgen:v1.2.2
* Verifying ingress addon...
* The 'ingress' addon is enabled
```

26. Apply yaml files

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f studentserver-deployment.yaml
deployment.apps/web created
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f studentserver-configmap.yaml
configmap/studentserver-config created
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f studentserver-service.yaml
service/web created
```

27. Apply yaml files

```
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f bookshelf-deployment.yaml
deployment.apps/bookshelf-deployment created
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f bookshelf-configmap.yaml
configmap/bookshelf-config created
tan19548@cloudshell:~ (focal-sight-302202)$ nano bookshelf-service.yaml
tan19548@cloudshell:~ (focal-sight-302202)$ kubectl apply -f bookshelf-service.yaml
service/bookshelf-service created
tan19548@cloudshell:~ (focal-sight-302202)$
```

28. Create studentservermongoIngress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: server
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
spec:
  rules:
    - host: cs571.project.com
      http:
        paths:
          - path: /studentserver(/|$)(.*)
            pathType: Prefix
            backend:
              service:
                name: web
                port:
                  number: 8080
          - path: /bookshelf(/|$)(.*)
            pathType: Prefix
            backend:
              service:
                name: bookshelf-service
                port:
                  number: 5000
```

29. Add hosts

```
# Kubernetes-managed hosts file.
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
fe00::0 ip6-mcastprefix
fe00::1 ip6-allnodes
fe00::2 ip6-allrouters
172.17.0.4      cs-450647327861-default-boost-jtssj
192.168.56.1    cs571.project.com
```

30. Curl


```
{"_id":"607391b7bb131403b789592a","student_id":11111,"student_name":"Bruce Lee","grade":84}
```


31. List book


```
[
  {
    "Book Author": "test",
    "Book Name": "123",
    "ISBN": "123",
    "id": "605d1ba7d40f50a395651765"
  }
]
```

32. Add a book


```
{
  "message": "Task saved successfully!"
}
```



```
[
  {
    "Book Author": "test",
    "Book Name": "123",
    "ISBN": "123updated",
    "id": "6073b45baeefcda681c994e1"
  },
  {
    "Book Author": "unkown",
    "Book Name": "kubernetes",
    "ISBN": "123456",
    "id": "6073b555aeefcda681c994e2"
  }
]
```


33. Update a book


```
{
  "message": "Task updated successfully!"
}
```



```
[
  {
    "Book Author": "test",
    "Book Name": "123",
    "ISBN": "123updated",
    "id": "6073b45baeefcda681c994e1"
  }
]
```


34. Delete a book


```
{
  "message": "Task deleted successfully!"
}
```



```
[
  {
    "Book Author": "unkown",
    "Book Name": "kubernetes",
    "ISBN": "123456",
    "id": "6073b555aeefcda681c994e2"
```