1. Create a cluster as usual on GKE

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

```
zuowei880315@cloudshell:~ (storied-polymer-343608)$ gcloud container clusters create kubia --num-nodes=1 --machine-type=e2-micro --region=us-west1
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters, please pas
s the `--no-enable-ip-alias` flag
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster kubia in us-west1... Cluster is being deployed...working...
```

```
kubeconfig entry generated for kubia.
NAME: kubia
LOCATION: us-west1
MASTER_VERSION: 1.21.9-gke.1002
MASTER_IP: 35.230.21.191
MACHINE_TYPE: e2-micro
NODE_VERSION: 1.21.9-gke.1002
NUM_NODES: 3
STATUS: RUNNING
```

2. Create compute mongodb

gcloud compute disks create --size=10GiB --zone=us-west1-a mongodb

3. Create mongodb deployment file

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ cat mongodb-deployment.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:
        - image: mongo
          name: mongo
          ports:
            - containerPort: 27017
          volumeMounts:
            - name: mongodb-data
              mountPath: /data/db
      volumes:
        - name: mongodb-data
          gcePersistentDisk:
            pdName: mongodb
            fsType: ext4
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
```

kubectl apply -f mongodb-deployment.yaml

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ kubectl apply -f mongodb-deployment.yaml
deployment.apps/mongodb-deployment created
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
```

4. Check pods status to running

kubectl get pods

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
mongodb-deployment-57dc68b4bd-6cmnt   1/1     Running   0          63s
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
```

5. Create mongoDB service

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ cat mongodb-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  type: LoadBalancer
  ports:
    - port: 27017
      targetPort: 27017
  selector:
    app: mongodb
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
```

kubectl apply -f mongodb-service.yaml

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ kubectl apply -f mongodb-service.yaml
service/mongodb-service created
```

6. Check service is up

```
zuowei880315@cloudshell:~ (storied-polymer-343608)$ kubectl get svc
NAME              TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)           AGE
kubernetes        ClusterIP      10.36.0.1     <none>         443/TCP           16m
mongodb-service   LoadBalancer   10.36.6.12    35.197.47.0    27017:30265/TCP   7m15s
zuowei880315@cloudshell:~ (storied-polymer-343608)$
```

7. Try to connections mongodb use external IP

kubectl exec -it mongodb-deployment-57dc68b4bd-6cmnt – bash

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ kubectl exec -it mongodb-deployment-57dc68b4bd-6cmnt -- bash
root@mongodb-deployment-57dc68b4bd-6cmnt:/#
```

mongo 35.197.47.0

```
root@mongodb-deployment-57dc68b4bd-6cmnt:/# mongo 35.197.47.0
MongoDB shell version v5.0.6
connecting to: mongodb://35.197.47.0:27017/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e5af2330-1780-4965-a5b8-178dd409e7d8") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
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:
        2022-03-31T03:26:08.420+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodn
otes-filesystem
        2022-03-31T03:26:09.659+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        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()
---
>
```

8. exit mongodb to google gonsole

```
        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> exit
bye
root@mongodb-deployment-57dc68b4bd-6cmnt:/# exit
exit
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
zuowei880315@cloudshell:~/project (storied-polymer-343608)$
```

9. insert records use node

node

```
zuowei880315@cloudshell:~/project (storied-polymer-343608)$ node
Welcome to Node.js v12.14.1.
Type ".help" for more information.
>
>
```

1.Create a student server

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat studentServer.js
////////////////////////////////////////////////////
//
// Write an HTTP server that serves JSON data when it
// receives a GET request to the path '/api/score'.
//
////////////////////////////////////////////////////
var http = require('http');
var url = require('url');
var mongodb = require('mongodb');
const {
  MONGO_URL,
  MONGO_DATABASE
}  = process.env;
// - Expect the request to contain a query
//   string with a key 'student_id' and a student ID as
//   the value. For example
//     /api/score?student_id=1111
// - The JSON response should contain only 'student_id', 'student_name'
//   and 'student_score' properties. For example:
//
//     {
//       "student_id": 1111,
//       "student_name": Bruce Lee,
//       "student_score": 84
//     }
//

var MongoClient = mongodb.MongoClient;
var uri = `mongodb://${MONGO_URL}/${MONGO_DATABASE}`;
// Connect to the db
console.log(uri);

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);
```

2. Create Dockerfile

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat Dockerfile
FROM node:7
ADD studentServer.js /studentServer.js
ENTRYPOINT ["node", "studentServer.js"]
RUN npm install mongodb
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

3. Build the studentserver docker image

docker build -t 13176zw/studentserver .

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ docker build -t 13176zw/studentserver .
Sending build context to Docker daemon  4.608kB
Step 1/4 : FROM node:7
 ---> d9aed20b68a4
Step 2/4 : ADD studentServer.js /studentServer.js
 ---> Using cache
 ---> 2fdb4e7f8eb8
Step 3/4 : ENTRYPOINT ["node", "studentServer.js"]
 ---> Using cache
 ---> fb2d9b0f7289
Step 4/4 : RUN npm install mongodb
 ---> Using cache
 ---> e5ea76f5e89b
Successfully built e5ea76f5e89b
Successfully tagged 13176zw/studentserver:latest
```

4. Push the docker image

docker push 13176zw/studentserver

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ docker push 13176zw/studentserver
Using default tag: latest
The push refers to repository [docker.io/13176zw/studentserver]
8dd719fb6bd9: Pushed
288037d6793c: Pushed
ab90d83fa34a: Pushed
8ee318e54723: Pushed
e6695624484e: Pushed
da59b99bbd3b: Pushed
5616a6292c16: Pushed
f3ed6cb59ab0: Pushed
654f45ecb7e3: Pushed
2c40c66f7667: Pushed
latest: digest: sha256:ffdc15403f5b54895863febe65461fb8b192f63fa6485e19905bdde6bffee2f3 size: 2424
```

1.Create python Flash bookshelf REST API

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ cat bookshelf.py
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()
```

2. Create Dockerfile

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ cat Dockerfile
FROM python:alpine3.7
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
ENV PORT 5000
EXPOSE 5000
ENTRYPOINT [ "python3" ]
CMD [ "bookshelf.py" ]
```

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ cat requirements.txt
Flask
Flask-PyMongo
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$
```

3. build the bookshelf docker

docker build -t 13176zw/bookshelf .

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ docker build -t 13176zw/bookshelf .
Sending build context to Docker daemon  6.144kB
Step 1/8 : FROM python:alpine3.7
 ---> 00be2573e9f7
Step 2/8 : COPY . /app
 ---> Using cache
 ---> d952929871a2
Step 3/8 : WORKDIR /app
 ---> Using cache
 ---> 53196c1bfa5e
Step 4/8 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> 95a62e5555c3
Step 5/8 : ENV PORT 5000
 ---> Using cache
 ---> 49f51bd657ff
Step 6/8 : EXPOSE 5000
 ---> Using cache
 ---> cbb4077c993f
Step 7/8 : ENTRYPOINT [ "python3" ]
 ---> Using cache
 ---> 1be771722d9a
Step 8/8 : CMD [ "bookshelf.py" ]
 ---> Using cache
 ---> 302f75c971d5
Successfully built 302f75c971d5
Successfully tagged 13176zw/bookshelf:latest
```

4. Push the docker image to dockerhub

docker push 13176zw/bookshelf

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ docker push 13176zw/bookshelf
Using default tag: latest
The push refers to repository [docker.io/13176zw/bookshelf]
9f204b15abdc: Pushed
9af3c61db918: Pushed
5fa31f02caa8: Pushed
88e61e328a3c: Pushed
9b77965e1d3f: Pushed
50f8b07e9421: Pushed
629164d914fc: Pushed
latest: digest: sha256:6c2c8514f1e6a354131f940214d09b0f377da5363d8febec2efe371dd9e9fd09 size: 1787
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$
```

1.Create a file named studentserver-configmap.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat studentserver-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: studentserver-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 35.197.47.0
  MONGO_DATABASE: mydb
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

2. Create a file named bookshelf-configmap.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat bookshelf-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: bookshelf-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 35.197.47.0
  MONGO_DATABASE: mydb
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

1.Create studentserver-deployment.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat studentserver-deployment.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: 13176zw/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
```

2. Create bookshelf-deployment.yaml

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ cat bookshelf-deployment.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: 13176zw/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
```

3. Create studentserver-service.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat 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
```

4. Create bookshelf-service.yaml

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ cat bookshelf-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: bookshelf-service
spec:
  type: LoadBalancer
  ports:
      # service port in cluster
    - port: 5000
      # port to contact inside container
      targetPort: 5000
  selector:
    app: bookshelf-deployment
```

5. start minikube

Minikube start

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ minikube start
* minikube v1.25.2 on Debian 11.2 (amd64)
  - MINIKUBE_FORCE_SYSTEMD=true
  - MINIKUBE_HOME=/google/minikube
  - MINIKUBE_WANTUPDATENOTIFICATION=false
* Automatically selected the docker driver. Other choices: none, ssh
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.23.3 preload ...
    > preloaded-images-k8s-v17-v1...: 505.68 MiB / 505.68 MiB  100.00% 70.66 Mi
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  - kubelet.cgroups-per-qos=false
  - kubelet.enforce-node-allocatable=""
  - kubelet.housekeeping-interval=5m
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

6. start Ingress

Minikube addons enable ingress

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ minikube addons enable ingress
  - Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
  - Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
  - Using image k8s.gcr.io/ingress-nginx/controller:v1.1.1
* Verifying ingress addon...
* The 'ingress' addon is enabled
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$
```

7. Create studentserver related pods and start service using the above yaml file

kubectl apply -f studentserver-deployment.yaml

kubectl apply -f studentserver-configmap.yaml

kubectl apply -f studentserver-service.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ kubectl apply -f studentserver-deployment.yaml
kubectl apply -f studentserver-configmap.yaml
kubectl apply -f studentserver-service.yaml
deployment.apps/web created
configmap/studentserver-config created
service/web created
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

8. Create bookshelf related pods and start service using the above yaml file

kubectl apply -f bookshelf-deployment.yaml

kubectl apply -f bookshelf-configmap.yaml

kubectl apply -f bookshelf-service.yaml

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ kubectl apply -f bookshelf-deployment.yaml
deployment.apps/bookshelf-deployment created
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ kubectl apply -f bookshelf-configmap.yaml
configmap/bookshelf-config created
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ kubectl apply -f bookshelf-service.yaml
service/bookshelf-service created
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$
```

9. Check all pods

kubectl get pods

```
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$ kubectl get pods
NAME                                 READY   STATUS             RESTARTS      AGE
bookshelf-deployment-6c4759bbbd-7p8jh   1/1     Running            0             86s
web-658c7bcc9d-82xsr                 0/1     CrashLoopBackOff   4 (48s ago)   2m43s
zuowei880315@cloudshell:~/project/bookshelf (storied-polymer-343608)$
```

10. Create studentservermongoIngress.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat 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
```

11. Create the ingress service

kubectl apply -f studentservermongoIngress.yaml

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ kubectl apply -f studentservermongoIngress.yaml
ingress.networking.k8s.io/server created
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

12. Check ingress is running

kubectl get ingress

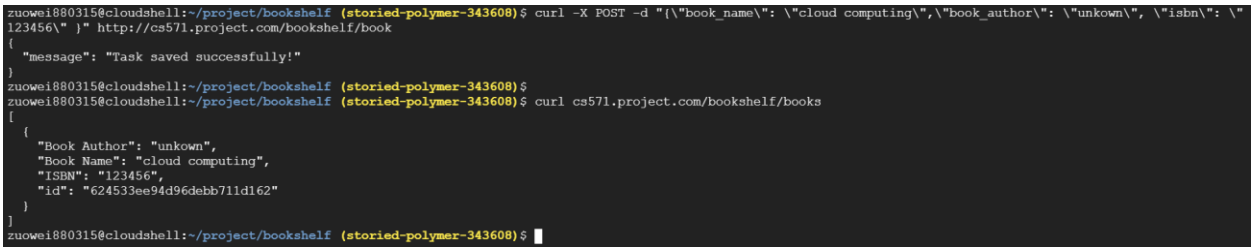
```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ kubectl get ingress
NAME     CLASS   HOSTS              ADDRESS   PORTS   AGE
server   nginx   cs571.project.com            80      35s
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$
```

13. edit /etc/hosts

```
zuowei880315@cloudshell:~/project/studentserver (storied-polymer-343608)$ cat /etc/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-336364871752-default
192.168.49.2    cs571.project.com
```

curl -X POST -d "{\"book_name\": \"cloud computing\",\"book_author\": \"unkown\", \"isbn\": \"123456\" }" http://cs571.project.com/bookshelf/book

curl cs571.project.com/bookshelf/books



curl -X DELETE cs571.project.com/bookshelf/book/624533ee94d96debb711d162