Create a cluster on GKE with:
gcloud container clusters create kubia --num-nodes=1 --machine-type=e2-micro --region=us-west1

Create a Persistent Volume with:
gcloud compute disks create --size=10GiB --zone=us-west1-a mongodb

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ gcloud compute disks create --size=10GiB --zone=us-west1-a mongodb
WARNING: You have selected a disk size of under [200GB]. This may result in poor I/O performance. For more information, s
ee: https://developers.google.com/compute/docs/disks#performance.
ERROR: (gcloud.compute.disks.create) Could not fetch resource:
 - The resource 'projects/cs571-demo-project-304716/zones/us-west1-a/disks/mongodb' already exists

gao19559@cloudshell:~ (cs571-demo-project-304716)$
```

Create a mongodb-deployment.yaml and create it with:
kubectl apply -f mongodb-deployment.yaml

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ 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:
        # by default, the image is pulled from docker hub
        - image: mongo
          name: mongo
          ports:
            - containerPort: 27017
          volumeMounts:
            - name: mongodb
              mountPath: /data/db
      volumes:
        - name: mongodb
          gcePersistentDisk:
            pdName: mongodb
            fsType: ext4
```

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl apply -f mongodb-deployment.yaml
deployment.apps/mongodb-deployment created
```

Check is the creating successful or not with:
kubectl get pods

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-deployment-ddbb4f557-rptr5   1/1     Running   0          46s
```

Create a mongodb-service.yaml and create it with:

kubectl apply -f mongodb-service.yaml

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl apply -f mongodb-service.yaml
service/mongodb-service created
gao19559@cloudshell:~ (cs571-demo-project-304716)$ cat  mongodb-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  type: LoadBalancer
  ports:
    - port: 27017
      targetPort: 27017
  selector:
    app: mongodb
```

Check is the creating successful or not with:

kubectl get svc

wait until EXTERNAL-IP show up

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl get svc
NAME              TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)           AGE
kubernetes        ClusterIP      10.3.240.1     <none>           443/TCP           24h
mongodb-service   LoadBalancer   10.3.240.50    35.199.164.154   27017:31496/TCP   95s
```

Check if mongodb is functioning for connections using the EXTERNAL-IP with:

kubectl exec -it mongodb-deployment-ddb4f557-rptr5 –bash

change to your pod name

```
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-deployment-ddbb4f557-rptr5   1/1     Running   0          16m
gao19559@cloudshell:~ (cs571-demo-project-304716)$ kubectl exec -it mongodb-deployment-ddbb4f557-r
ptr5 -- bash
root@mongodb-deployment-ddbb4f557-rptr5:/#
```

We are inside the mongodb deployment pod, then type：

mongo 35.199.164.154 replace to your own EXTERNAL-IP

```
root@mongodb-deployment-ddbb4f557-rptr5:/# mongo 35.199.164.154
MongoDB shell version v4.4.5
connecting to: mongodb://35.199.164.154:27017/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("df446c3b-c672-4b04-b3b9-c73e07c79932") }
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:
        2021-04-10T00:20:12.682+00:00: Using the XFS filesystem is strongly recommended with the W
iredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
        2021-04-10T00:20:14.033+00:00: Access control is not enabled for the database. Read and wr
ite 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(
)
---
>
```

Type exit to go back

```
---
> exit
bye
root@mongodb-deployment-ddbb4f557-rptr5:/# exit
exit
```

Create a .js file to store the testing data and run it with:
node initalDataforStudentServer.js replace to your own .js file

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ cat initalDataforStudentServer.js
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://35.199.164.154:27017/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);
    });

});
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ node initalDataforStudentServer.js
3
{
  _id: 60728ed11484a0034360f60c,
  student_id: 11111,
  student_name: 'Bruce Lee',
  grade: 84
}
```

Create a studentServer.js with:

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ cat studentServer.js
var http = require('http');
var url = require('url');
var mongodb = require('mongodb');
const {
  MONGO_URL,
  MONGO_DATABASE
}  = process.env;

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);

  // 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);
```

Create a Dockerfile with:

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ cat Dockerfile
FROM node:7
ADD studentServer.js /studentServer.js
ENTRYPOINT ["node", "studentServer.js"]
RUN npm install mongodb
```

Build the studentserver docker image with:

docker build -t 19559gp/studentserver .

replace to your dockerhub ID

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ docker build -t 19559gp/studentserver
 .
Sending build context to Docker daemon  10.75kB
Step 1/4 : FROM node:7
 ---> d9aed20b68a4
Step 2/4 : ADD studentServer.js /studentServer.js
 ---> b7636df22fc9
Step 3/4 : ENTRYPOINT ["node", "studentServer.js"]
 ---> Running in 49f98407cc3e
Removing intermediate container 49f98407cc3e
 ---> 1430934dc5e0
Step 4/4 : RUN npm install mongodb
 ---> Running in d14362eced86
npm info it worked if it ends with ok
npm info using npm@4.2.0
npm info using node@v7.10.1
npm info attempt registry request try #1 at 6:11:22 AM
npm http request GET https://registry.npmjs.org/mongodb
npm http 200 https://registry.npmjs.org/mongodb
npm info addNameTag [ 'mongodb', 'latest' ]
npm info retry fetch attempt 1 at 6:11:22 AM
npm info attempt registry request try #1 at 6:11:22 AM
npm http fetch GET https://registry.npmjs.org/mongodb/-/mongodb-3.6.6.tgz
```

```
npm info lifecycle mongodb@3.6.6~postinstall: mongodb@3.6.6
/
`-- mongodb@3.6.6
  +-- bl@2.2.1
  | `-- readable-stream@2.3.7
  |   +-- core-util-is@1.0.2
  |   +-- inherits@2.0.4
  |   +-- isarray@1.0.0
  |   +-- process-nextick-args@2.0.1
  |   +-- safe-buffer@5.1.2
  |   +-- string_decoder@1.1.1
  |   | `-- safe-buffer@5.1.2
  |   `-- util-deprecate@1.0.2
  +-- bson@1.1.6
  +-- denque@1.5.0
  +-- optional-require@1.0.3
  +-- safe-buffer@5.2.1
  `-- saslprep@1.0.3
    `-- sparse-bitfield@3.0.3
      `-- memory-pager@1.5.0

npm WARN enoent ENOENT: no such file or directory, open '/package.json'
npm WARN !invalid#1 No description
npm WARN !invalid#1 No repository field.
npm WARN !invalid#1 No README data
npm WARN !invalid#1 No license field.
npm info ok
Removing intermediate container d14362eced86
 ---> 038aa3a40bee
Successfully built 038aa3a40bee
Successfully tagged 19559gp/studentserver:latest
```

Push the docker image with:

docker push 19559gp/studentserver replace to your dockerhub ID

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ docker push 19559gp/studentserver
Using default tag: latest
The push refers to repository [docker.io/19559gp/studentserver]
086a72304691: Pushed
679bb01330f9: 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: sha256:8c88008913eff923a2f72ecca486e4cc040365e38f11e76d68f15c6a63eb2522 size: 2424
```

Create a bookshelf.py with:

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ 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()
    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"])
def update_book(id):
    data = request.get_json(force=True)
    print(data)
    response = db.bookshelf.update_many({"_id": ObjectId(id)}, {"$set": {"book_name": data['book_name'],
    response = db.bookshelf.update_many({"_id": ObjectId(id)}, {"$set": {"book_name": data['book_name'],
        "book_author": data["book_author"], "ISBN": data["isbn"]
        }})
    if response.matched_count:
        message = "Task updated successfully!"
    else:
        message = "No book found!"
    return jsonify(
        message=message
    )

@app.route("/book/<id>", methods=["DELETE"])
def delete_task(id):
    response = db.bookshelf.delete_one({"_id": ObjectId(id)})
    if response.deleted_count:
        message = "Task deleted successfully!"
    else:
        message = "No book found!"
    return jsonify(
        message=message
    )

@app.route("/tasks/delete", methods=["POST"])
def delete_all_tasks():
    db.bookshelf.remove()
    return jsonify(
        message="All Books deleted!"
    )
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Create Dockerfile:

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ 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" ]
```

Build the bookshelf app into a docker image with:

docker build -t 19559gp/bookshelf .

replace to your dockerhub ID

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ docker build -t 19559gp/bookshelf .
Sending build context to Docker daemon  22.53kB
Step 1/8 : FROM python:alpine3.7
alpine3.7: Pulling from library/python
48ecbb6b270e: Pull complete
692f29ee68fa: Pull complete
6439819450d1: Pull complete
3c7be240f7bf: Pull complete
ca4b349df8ed: Pull complete
Digest: sha256:35f6f83ab08f98c727dbefd53738e3b3174a48b4571ccb1910bae480dcdba847
Status: Downloaded newer image for python:alpine3.7
 ---> 00be2573e9f7
Step 2/8 : COPY . /app
 ---> 84ac07280191
Step 3/8 : WORKDIR /app
 ---> Running in da06e1b979d9
Removing intermediate container da06e1b979d9
 ---> 4ef2220341e7
Step 4/8 : RUN pip install -r requirements.txt
 ---> Running in b9fdb79ec7a5
Collecting Flask (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/f2/28/2a03252dfb9ebf377f40fba6a7841b47083260bf8bd8e737b0c6952df83f/Flask-1.1.
2-py2.py3-none-any.whl (94kB)
Collecting Flask-PyMongo (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/67/b8/0322016b9ce09a64fba9018211e7c35fd51380527ffd9ea248744f389239/Flask_PyMo
ngo-2.3.0-py2.py3-none-any.whl
Collecting Werkzeug>=0.15 (from Flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/cc/94/5f7079a0e00bd6863ef8f1da638721e9da21e5bacee597595b318f71d62e/Werkzeug-1
.0.1-py2.py3-none-any.whl (298kB)
Collecting Jinja2>=2.10.1 (from Flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/7e/c2/1eece8c95ddbc9b1aeb64f5783a9e07a286de42191b7204d67b7496ddf35/Jinja2-2.1
1.3-py2.py3-none-any.whl (125kB)
Collecting click>=5.1 (from Flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/d2/3d/fa76db83bf75c4f8d338c2fd15c8d33fdd7ad23a9b5e57eb6c5de26b430e/click-7.1.
2-py2.py3-none-any.whl (82kB)
Collecting itsdangerous>=0.24 (from Flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangero
us-1.1.0-py2.py3-none-any.whl
Collecting PyMongo>=3.3 (from Flask-PyMongo->-r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/72/82/e7196f2f69318dd206db26db68fcfa0ff821d88fbca6d0f0c7b678ba0353/pymongo-3.
11.3.tar.gz (777kB)
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->Flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/b9/2e/64db92e53b86efccfaea71321f597fa2e1b2bd3853d8ce658568f7a13094/MarkupSafe
-1.1.1.tar.gz
Building wheels for collected packages: PyMongo, MarkupSafe
  Building wheel for PyMongo (setup.py): started
  Building wheel for PyMongo (setup.py): finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/97/64/bb/be01adf5254f3e63c246204e2df51543af23e24e5531f8cf2a
  Building wheel for MarkupSafe (setup.py): started
  Building wheel for MarkupSafe (setup.py): finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/f2/aa/04/0edf07a1b8a5f5f1aed7580fffb69ce8972edc16a505916a77
Successfully built PyMongo MarkupSafe
Installing collected packages: Werkzeug, MarkupSafe, Jinja2, click, itsdangerous, Flask, PyMongo, Flask-PyMongo
```

```
Removing intermediate container b9fdb79ec7a5
 ---> f315002826ee
Step 5/8 : ENV PORT 5000
 ---> Running in 3b63f63f851a
Removing intermediate container 3b63f63f851a
 ---> e6519185a856
Step 6/8 : EXPOSE 5000
 ---> Running in f1fbb15ec2aa
Removing intermediate container f1fbb15ec2aa
 ---> c55f84256e9d
Step 7/8 : ENTRYPOINT [ "python3" ]
 ---> Running in f7ada49f62e2
Removing intermediate container f7ada49f62e2
 ---> 87d7b0b29f46
Step 8/8 : CMD [ "bookshelf.py" ]
 ---> Running in 77e5795c9056
Removing intermediate container 77e5795c9056
 ---> e0f03b1bf8cc
Successfully built e0f03b1bf8cc
Successfully tagged 19559gp/bookshelf:latest
```

Push the docker image to duckerhub with:

docker push 19559gp/bookshelf

replace to your dockerhub ID

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ docker push 19559gp/bookshelf
Using default tag: latest
The push refers to repository [docker.io/19559gp/bookshelf]
851a988e7fc2: Pushed
d51c016ac1ef: 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: sha256:9417e658db6341525b38f1c9caaa35e79498715b123e6386ed38450a3ebc8937 size: 1787
```

Create a studentserver-configmap.yaml:

Remember change to your mongoDB EXTERNAL-IP

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ cat  studentserver-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: studentserver-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 35.199.164.154
  MONGO_DATABASE: mydb
```

Create a bookshelf-configmap.yaml:

Remember change to your mongoDB EXTERNAL-IP

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ cat bookshelf-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: bookshelf-config
data:
  # SERVICE_NAME.NAMESPACE.svc.cluster.local:SERVICE_PORT
  MONGO_URL: 35.199.164.154
  MONGO_DATABASE: mydb
```

Create a studnetserver-deployment.yaml:

Remember to change to your dockerhubID

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ 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: 19559gp/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
```

Create a sutdentserver-service.yaml:

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ 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
```

Create bookshelf-deployment.yaml:

Remember to change to your dockerhubID

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ 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: 19559gp/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
```

Create a bookshelf-service.yaml:

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ 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
```

Strat minikube with: minkube start

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ minikube start
* minikube v1.18.1 on Debian 10.9 (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 v1.20.2 preload ...
    > preloaded-images-k8s-v9-v1....: 491.22 MiB / 491.22 MiB  100.00% 167.20 M
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Start ingress with: minikube addons enable ingress

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ minikube addons enable ingress
  - Using image us.gcr.io/k8s-artifacts-prod/ingress-nginx/controller:v0.40.2
  - Using image jettech/kube-webhook-certgen:v1.2.2
  - Using image jettech/kube-webhook-certgen:v1.3.0
* Verifying ingress addon...
* The 'ingress' addon is enabled
```

Create the studentserver related pods with:

kubectl apply -f studentserver-deployment.yaml

kubectl apply -f studentserver-configmap.yaml

kubectl apply -f studentserver-service.yaml

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl apply -f studentserver-deployment.yaml
deployment.apps/web created
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl apply -f studentserver-configmap.yaml
configmap/studentserver-config created
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl apply -f studentserver-service.yaml
service/web created
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$
```

Create the bookshelf related pods with:

kubectl apply -f bookshelf-deployment.yaml

kubectl apply -f bookshelf-configmap.yaml

kubectl apply -f bookshelf-service.yaml

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ kubectl apply -f bookshelf-deployment.yaml
deployment.apps/bookshelf-deployment created
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ kubectl apply -f bookshelf-configmap.yaml
configmap/bookshelf-config created
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ kubectl apply -f bookshelf-service.yaml
service/bookshelf-service created
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$
```

Check is the pods working correct or not with: kubectl get pods

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
bookshelf-deployment-9d6c56787-fblbx  1/1     Running   0          40s
web-677b4f6657-5g2qg                  1/1     Running   0          114s
```

Create a studentservermongoIngress.yaml with:

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ 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
```

Create the ingress service with:

kubectl apply -f studentservermongoIngress.yaml

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl apply -f studentservermongoIngress.yaml
ingress.networking.k8s.io/server created
```

Check is the ingrerss working correct or not with: kubectl get ingress

Wait unit we get the ADDRESS

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl get ingress
NAME     CLASS    HOSTS              ADDRESS   PORTS   AGE
server   <none>   cs571.project.com            80      52s
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl get ingress
NAME     CLASS    HOSTS              ADDRESS   PORTS   AGE
server   <none>   cs571.project.com            80      53s
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ kubectl get ingress
NAME     CLASS    HOSTS              ADDRESS        PORTS   AGE
server   <none>   cs571.project.com   192.168.49.2   80      75s
```

Let add address to /etc/hosts with:

sudo vi /etc/hosts

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ sudo vi /etc/hosts
```

Change the marked part in screenshot to:

192.168.49.2 cs571.project.com

Change to your address which get from previous step

```
# 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
192.168.49.2    cs571.project.com
~
~
~
~
```

If everything goes correct, we can run to get testing data for student server with:

curl cs571.project.com/studentserver/api/score?student_id=11111

curl cs571.project.com/studentserver/api/score?student_id=22222

curl cs571.project.com/studentserver/api/score?student_id=33333

```
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ curl cs571.project.com/studentserver/api/score?
student_id=11111
{"_id":"60728ed11484a0034360f60c","student_id":11111,"student_name":"Bruce Lee","grade":84}
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ curl cs571.project.com/studentserver/api/score?
student_id=22222
{"_id":"60728ed11484a0034360f60d","student_id":22222,"student_name":"Jackie Chen","grade":93}
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ curl cs571.project.com/studentserver/api/score?
student_id=33333
{"_id":"60728ed11484a0034360f60e","student_id":33333,"student_name":"Jet Li","grade":88}
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ curl cs571.project.com/studentserver/api/score?
student_id=44444
Student Not Found
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$ curl cs571.project.com/studentserver/api/score?
student_id=111
Student Not Found
gao19559@cloudshell:~/kubernetes_project/studentserver (cs571-demo-project-304716)$
```

Let’s test the bookshelf with these commands:

List all books with: curl cs571.project.com/bookshelf/books

Add a book with:

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

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl cs571.project.com/bookshelf/books
[]
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl -X POST -d "{\"book_name\": \"cloud computing\
",\"book_author\":
> \"unkown\", \"isbn\": \"123456\" }" http://cs571.project.com/bookshelf/book
{
  "message": "Task saved successfully!"
}
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl cs571.project.com/bookshelf/books
[
  {
    "Book Author": "unkown",
    "Book Name": "cloud computing",
    "ISBN": "123456",
    "id": "6072a11e6651abb37d99d8b5"
  }
]
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl -X POST -d "{\"book_name\": \"NPU MCS\",\"book
_author\":
\"Oscar\", \"isbn\": \"19559\" }" http://cs571.project.com/bookshelf/book
{
  "message": "Task saved successfully!"
}
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl cs571.project.com/bookshelf/books
[
  {
    "Book Author": "unkown",
    "Book Name": "cloud computing",
    "ISBN": "123456",
    "id": "6072a11e6651abb37d99d8b5"
  },
  {
    "Book Author": "Oscar",
    "Book Name": "NPU MCS",
    "ISBN": "19559",
    "id": "6072a1ac6651abb37d99d8b6"
  }
]
```

Update a book with:

curl -X PUT -d "{\"book_name\": \"unknown12\",\"book_author\": \"test\", \"isbn\": \"123updated\" }" http://cs571.project.com/bookshelf/book/id

change the book id to you want to change

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl -X PUT -d "{\"book_name\": \"unknown12\",\"boo
k_author\": \"test\", \"isbn\":
\"123updated\" }" http://cs571.project.com/bookshelf/book/6072a11e6651abb37d99d8b5
{
  "message": "Task updated successfully!"
}
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl cs571.project.com/bookshelf/books
[
  {
    "Book Author": "test",
    "Book Name": "unknown12",
    "ISBN": "123updated",
    "id": "6072a11e6651abb37d99d8b5"
  },
  {
    "Book Author": "Oscar",
    "Book Name": "NPU MCS",
    "ISBN": "19559",
    "id": "6072a1ac6651abb37d99d8b6"
  }
]
```

Delete a book with；

curl -X DELETE cs571.project.com/bookshelf/book/id change the book id to you want to delete

```
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl -X DELETE cs571.project.com/bookshelf/book/607
2a11e6651abb37d99d8b5
{
  "message": "Task deleted successfully!"
}
gao19559@cloudshell:~/kubernetes_project/bookshelf (cs571-demo-project-304716)$ curl cs571.project.com/bookshelf/books
[
  {
    "Book Author": "Oscar",
    "Book Name": "NPU MCS",
    "ISBN": "19559",
    "id": "6072a1ac6651abb37d99d8b6"
  }
]
```