Wave 2 innovation

In the next set of innovation may move to the deployment kind so that we could use the extra benefits not provided by ordinary pods.

Also fixed a bug in the Python code where the code immediately try to make a connection to the database before the database was up and running. So I moved the database connectivity code to each of the three subroutines so that that code would only execute when the user tried to interact with the service, not at deployment time.

The persistent storage allows you tear down the cluster and bring it up without losing data.

Deployments

- **Deployment controller** - provides declarative updates for Pods and ReplicaSets.
- **Deployments** - are intended to replace Replication Controllers.
- **But can do updates/rollbacks** - They provide the same replication functions (through Replica Sets) and also the ability to rollout changes and roll them back if necessary.

Try/except syntax add to python code

```
@app.route('/init')
def init():
    try:
       db = MySQLdb.connect("mysql","root","password")
       cursor = db.cursor()
       cursor.execute("DROP DATABASE IF EXISTS AZUREDB")
       cursor.execute("CREATE DATABASE AZUREDB")
       cursor.execute("USE AZUREDB")
       sql = """CREATE TABLE courses(id INT,
              coursenumber varchar(48),
              coursetitle varchar(256), notes varchar(256));"""
       cursor.execute(sql)
       db.commit()
       return "\nDB Initialization done\n\n"
    except (MySQLdb.Error, MySQLdb.Warning) as e:
       return e.msg
```

Reading the logs for the web1 pod and python container

```
root-> k logs web1 python
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 135-400-973
10.244.3.1 - - [12/Nov/2017 17:08:01] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:06] "GET /init HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:11] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:21] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:31] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:41] "GET /healthz HTTP/1.1" 200 -
10.240.0.4 - - [12/Nov/2017 17:08:51] "POST /courses/add HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:51] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:08:56] "POST /courses/add HTTP/1.1" 200 -
10.240.0.5 - - [12/Nov/2017 17:09:01] "POST /courses/add HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:09:01] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:09:11] "GET /healthz HTTP/1.1" 200 -
10.244.3.1 - - [12/Nov/2017 17:09:16] "GET /courses/1 HTTP/1.1" 200 -
```

Yaml files not used when going to V2

There are files that are not needed as we go to a Deployment kind.

db-pod.yml

New Files when going to V2

db-pvc.yml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

db-deployment.yml

```
apiVersion: apps/v1beta1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: mysql
  labels:
    name: mysql
    app: demo
spec:
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        run: mysql
    spec:
      containers:
      - image: mysql:latest
        args:
           - "--ignore-db-dir=lost+found"
        name: mysql
        env:
        - name: MYSQL ROOT PASSWORD
          value: password

#       - name: MYSQL ROOT PASSWORD
#         valueFrom:
#           secretKeyRef:
#             name: mysql-pass
#             key: password
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pv-claim
```

Notice the mysql-pv-claim

Refers to db-pvc.yml

args is needed or there is a “directory not empty error during deployment”

db-svc.yml

The key point here is that the service points to the deployment. You can see in the specifications section there is a selector tag that says run mysql. This is needed for the Python code that directly addresses "mysql." the service simply acts as a front end to the deployment.

```
apiVersion: v1
kind: Service
metadata:
  name: mysql
```

```
    labels:
      name: mysql
      app: demo
  spec:
    ports:
      - name: mysql
        port: 3306
        targetPort: 3306
    selector:
      run: mysql
```

This service points to the deployment

Notice the section:

```
selector:
  run: mysql
```

Commands Needed to run

build-all2.sh

```
kubectl create -f web-pod-1.yml
kubectl create -f web-svc.yml
kubectl create -f db-pvc.yml
kubectl create -f db-deployment.yml
kubectl create -f db-svc.yml
```

clean2.sh

```
kubectl delete pod web1
kubectl delete svc web
kubectl delete deploy mysql
kubectl delete svc mysql
kubectl delete pvc mysql-pv-claim
```