

Task1

환경

- Ubuntu
Description: Ubuntu 22.04.1 LTS
Release: 22.04
Codename: jammy
- node: v18.12.1
- minikube version: v1.27.1

구현 방법

코드 수정

- vote dir의 app.py에서 cookie 부분을 지워주고 창을 실행 시마다 새로운 voter_id를 등록하게 설정했다.

```
@app.route("/", methods=['POST', 'GET'])
def hello():
    # voter_id = request.cookies.get('voter_id')
    # if not voter_id:
    #     voter_id = hex(random.getrandbits(64))[2:-1]

    voter_id = hex(random.getrandbits(64))[2:-1]
    vote = None

    if request.method == 'POST':
        redis = get_redis()
        vote = request.form['vote']
        app.logger.info('Received vote for %s', vote)
        data = json.dumps({'voter_id': voter_id, 'vote': vote})
        redis.rpush('votes', data)

    resp = make_response(render_template(
        'index.html',
        option_a=option_a,
        option_b=option_b,
        hostname=hostname,
        vote=vote,
    ))
    # resp.set_cookie('voter_id', voter_id)
    return resp
```

K8s image 생성 및 push

```
# in example-voting-app-task1 dir
$ docker login
hkyeo98
${password}
$ docker build vote/ -t example-voting-app-vote-task1
$ docker tag example-voting-app-vote-task1:latest docker.io/kyuber/example-voting-app-vote-task1:latest
$ docker push docker.io/kyuber/example-voting-app-vote-task1:latest
```

vote-deployment.yaml 적용

```
# vote-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: vote
  name: vote
  namespace: vote
spec:
  replicas: 1
  selector:
    matchLabels:
```

```

    app: vote
  template:
    metadata:
      labels:
        app: vote
  spec:
    containers:
      - image: kyuber/example-voting-app-vote-task1:latest
        name: vote
        ports:
          - containerPort: 80
            name: vote

```

실행 방식

docker-compose

```

# in example-voting-app-task1 dir
$ docker-compose up
yeo@kyuber:~/development/distributed_system/example-voting-app-task1$ docker-compose up
Starting example-voting-app-task1_redis_1 ... done
Starting example-voting-app-task1_db_1 ... done
Starting example-voting-app-task1_vote_1 ... done
Starting example-voting-app-task1_worker_1 ... done
Starting example-voting-app-task1_result_1 ... done
Attaching to example-voting-app-task1_redis_1, example-voting-app-task1_db_1, example-voting-app-task1_worker_1, example-voting-app-task1_result_1
db_1      |
db_1      | PostgreSQL Database directory appears to contain a database; Skipping initialization
db_1      |
redis_1   | 1:C 21 Nov 2022 07:26:06.918 # o000o000o000o Redis is starting o000o000o000o
redis_1   | 1:C 21 Nov 2022 07:26:06.918 # Redis version=5.0.7, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1   | 1:C 21 Nov 2022 07:26:06.918 # Warning: no config file specified, using the default config. In order to specify a config file
db_1      | LOG: database system was interrupted; last known up at 2022-11-21 07:24:40 UTC
redis_1   | 1:M 21 Nov 2022 07:26:06.919 * Running mode=standalone, port=6379.
db_1      | LOG: database system was not properly shut down; automatic recovery in progress
redis_1   | 1:M 21 Nov 2022 07:26:06.919 # Server initialized
db_1      | LOG: redo starts at 0/16BE188
redis_1   | 1:M 21 Nov 2022 07:26:06.919 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create
db_1      | LOG: record with zero length at 0/16BE4F8
db_1      | LOG: redo done at 0/16BE4C8
db_1      | LOG: last completed transaction was at log time 2022-11-21 07:25:37.819904+00
redis_1   | 1:M 21 Nov 2022 07:26:06.919 * Ready to accept connections
db_1      | LOG: MultiXact member wraparound protections are now enabled
db_1      | LOG: autovacuum launcher started
db_1      | LOG: database system is ready to accept connections
vote_1    | * Serving Flask app 'app'
vote_1    | * Debug mode: on
vote_1    | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
vote_1    | * Running on all addresses (0.0.0.0)
vote_1    | * Running on http://127.0.0.1:80
vote_1    | * Running on http://172.24.0.4:80
vote_1    | Press CTRL+C to quit
vote_1    | * Restarting with stat

# vote ui on bold part of vote_1 which changes every runs.
# result ui on localhost:5001

```

kubernetes

```

# in example-voting-app-task1 dir
$ kubectl create namespace vote
$ kubectl create -f k8s-specifications/

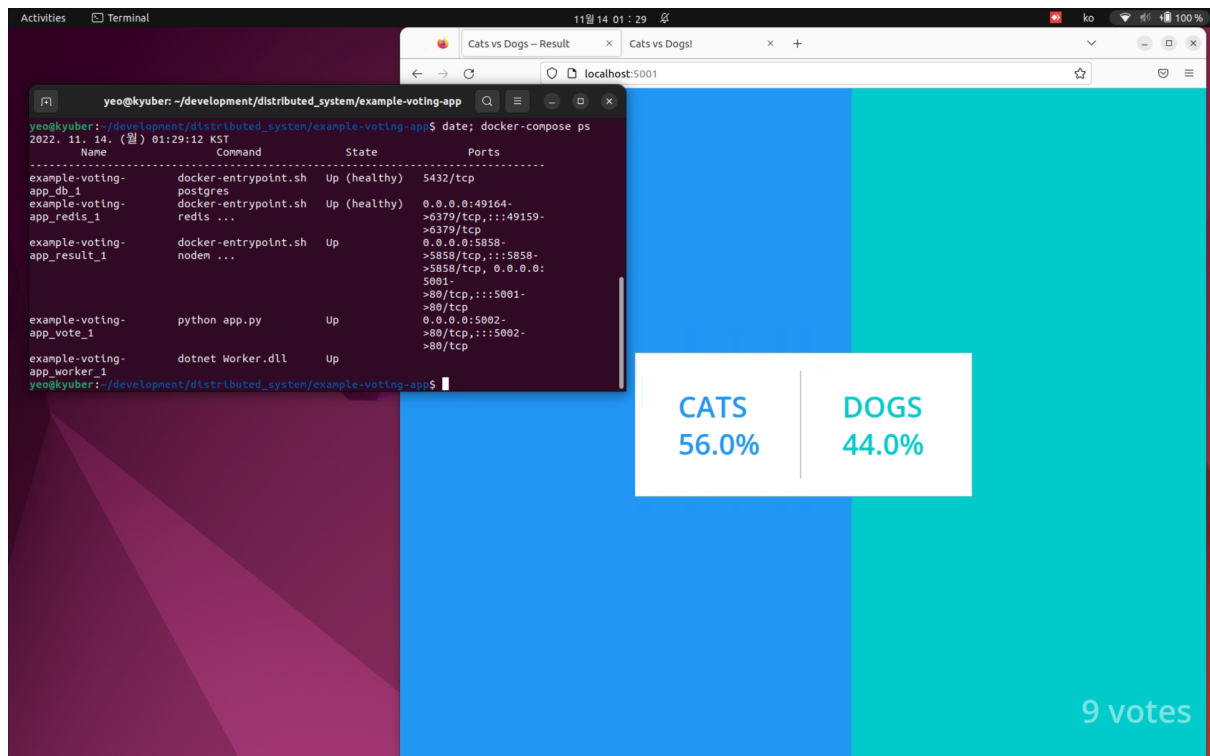
# vote url. enter below address to browser to see the vote ui.
$ minikube service vote -n vote --url
http://192.168.49.2:31000/

# result url. enter below address in browser to see the result ui.
$ minikube service result -n vote --url
http://192.168.49.2:31001

```

실행 결과

docker-compose



kubernetes

