1. **go to root (cd $home )**
2. **git clone** [**https://github.com/ashishrpandey/example-voting-app**](https://github.com/ashishrpandey/example-voting-app)

* cloned the example-voting-app from the GitHub Repository

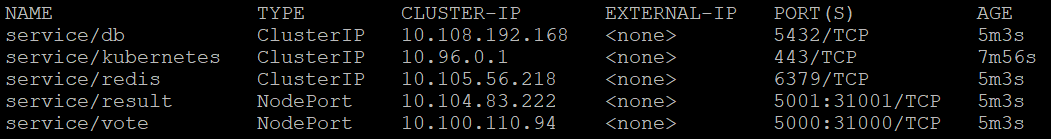
1. **cd /root/example-voting-app/k8s-specifications**

* changed current working directory to 'k8s-specifications'

1. **kubectl apply -f .**

* used ‘*kubectl apply’* for deploying the service and deployment YAML files of voting-app, redis, worker, db and result-app
* ‘*voting-app*’ is a Python webapp which lets you vote between two options
* ‘*redis’* is a queue which collects new votes
* a ‘.*NET worker’* which consumes votes and stores them in
* a ‘*Postgres database’* backed by a Docker volume
* a ‘*Node.js webapp’* which shows the results of the voting in real time

1. **for voting and result pods**



* As we can see in this snip that db, worker & redis are of type ‘ClusterIP’ while result and vote app are of type ‘NodePort’.
* Here NodePort service serves as the external entry point for incoming requests for our voting and result app while ClusterIP is used to show that an IP can be used to gain access to these server inside the cluster only.
* As I am a Master , my voting app has NodePort of 31000 and result app has NodePort of 31001.

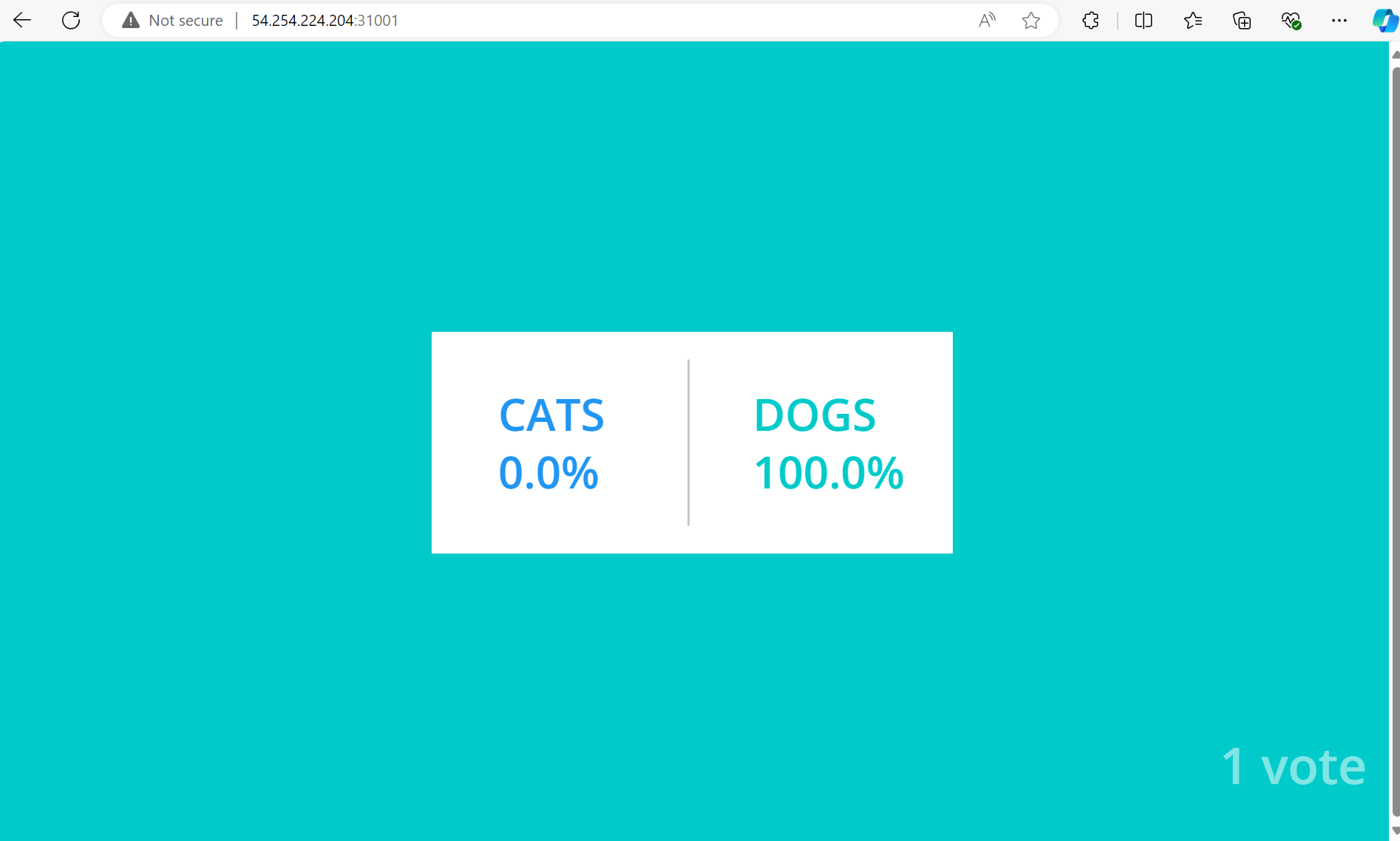
1. **My publicIP : NodePort for both voting and result apps**

* My public IP is 54.254.224.204
* So, 54.254.224.204:31000 redirects me to my voting app

A screenshot of a computer

Description automatically generated

* And, 54.254.224.204:31001 redirects me to result app



1. **Try voting and see the results paralelly in results page**

* As we can see in the image above, that there is 1 vote for Dogs. This result calculation is done as soon as we pick our side in voting app and we can always see the result in the results app.

1. **Deleting pods one by one**

* i.) Deleting vote pod : In voting app deployment YAML file, we have defined ‘*replicas: 1*’ due to which whenever we try to delete the voting pod, a new pod instance is created instantly because of which we encounter no problem in voting app and we can carry on our tasks seamlessly. So, we see no change in frontend while we can see one vote pod terminating and a new one getting created in UNIX.
* ii.) Deleting worker pod : While deleting worker pod too, we encounter no problems as in Worker deployment YAML file, we have defined *‘replicas: 1’* due to which a new pod gets created instantly as soon as one worker pod is deleted. So, we see no change in frontend while we can see one worker pod terminating and a new one getting created in UNIX. As new instance is created, so we also lose logs which were there in previous instance.
* iii.) Deleting DB pod : While deleting DB pod, same thing happens. The existing DB pod instance Terminates while new one gets created because of the DB Deployment YAML file that contains ‘*replicas: 1*’. But here we get one problem that is all the data that was there with the DB gets lost with the instance deletion and new instance starts storing new data. With this new instance creation, the worker and result pod also restarts as they are totally dependent on DB pod. The worker pod feeds data to the DB pod, while result pod fetches data from the DB pod.

11.) complete the assignment by making the result pod work. (if it is not working)

- My result pod is working fine.

**Commands used**

1. kubectl apply -f .
2. kubectl get all
3. watch kubectl get po -o wide
4. kubectl delete all –all
5. kubectl delete po [pod\_name]
6. kubectl describe po [pod\_name]