

LAB 4 - Screenshots by Athika Fatima(101502209)

Part 1: Architecture

Deployment Options

- StatefulSets

Rationale: In our Docker Ethereum application, the backend interacts with the Ethereum blockchain and requires unique identities, stable network hostnames, and persistent storage. StatefulSets are chosen to ensure these characteristics, allowing our backend to maintain state across pod restarts seamlessly.

- ReplicaSets

Rationale: For the frontend of our Docker Ethereum app, which is designed to be stateless and scalable, ReplicaSets offer an ideal solution. They provide simplicity and scalability, allowing the frontend to handle increased user traffic through horizontal scaling.

Storage

- PersistentVolumeClaims (PVC)

Rationale: To preserve Ethereum blockchain data and application state, even if a pod is rescheduled or restarted, PersistentVolumeClaims are utilized. This ensures reliable and scalable storage for our stateful backend components.

- Volumes

Rationale: With the frontend being stateless, the use of volumes focuses on sharing code and configurations among frontend replicas. This simplicity aligns with the stateless nature of the frontend, allowing for easy scalability without persistent storage needs.

Scaling

- Horizontal Pod Autoscaler (HPA)

Rationale: The backend, dealing with Ethereum blockchain interactions, encounters varying loads. HPA dynamically adjusts the number of backend replicas based on CPU utilization, ensuring efficient handling of increased or decreased traffic.

Load Balancing

- Kubernetes Service

Rationale: Services in Kubernetes provide a stable endpoint and automatic load balancing, distributing traffic evenly among pods. This enhances the reliability and availability of our Docker Ethereum application.

Secrets

- ConfigMap

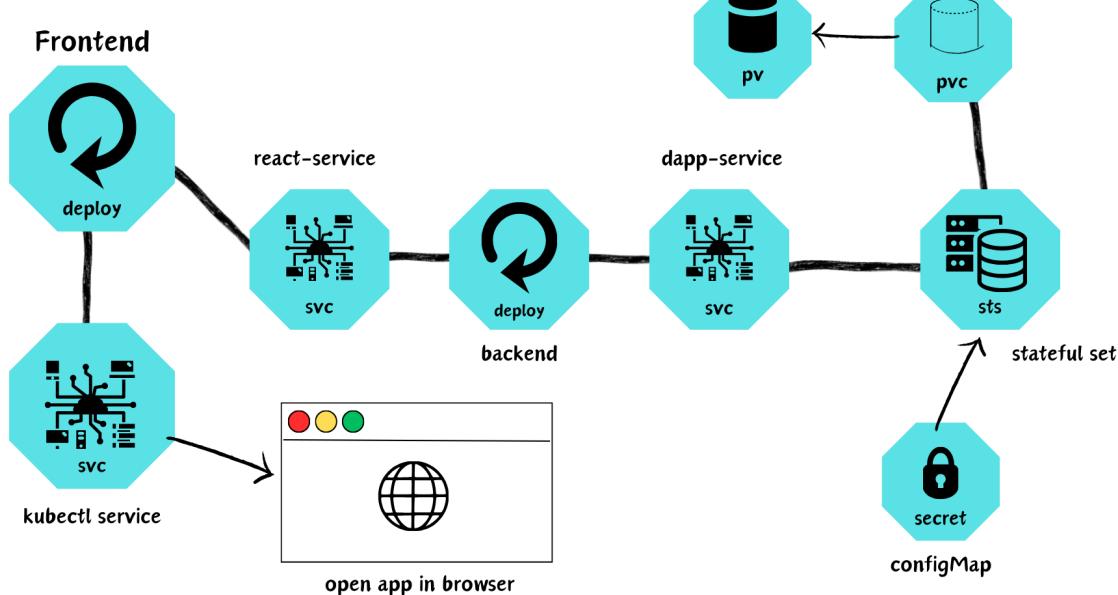
Rationale: ConfigMap is chosen for storing non-sensitive data shared among pods. It simplifies configuration parameter management, allowing for easy updates without altering the application code in our Docker Ethereum app.

User and Role Management

- RBAC (Role-Based Access Control)
Rationale: RBAC ensures that only authorized users can perform specific actions within the Kubernetes cluster hosting our Docker Ethereum app. This enhances security by limiting access based on predefined roles, preventing unauthorized resource manipulation.
- kubectl Contexts
Rationale: kubectl contexts offer a convenient way to switch between different clusters, namespaces, and user roles. This aids in maintaining a secure and organized environment, crucial in multi-user or multi-tenant Kubernetes clusters hosting our Docker Ethereum application.

ARCHITECTURE

BY ATHIKA FATIMA(101502209)



PART 2: Deploying the YAML Files and implementing the Architecture

1) \$ kompose convert -f docker-compose.yml

```
[athikafatima@Athikas-MacBook-Pro docker-ethereum % kompose convert -f docker-compose.yml
INFO Kubernetes file "dapp-service.yaml" created
INFO Kubernetes file "ganache-service.yaml" created
INFO Kubernetes file "react-service.yaml" created
INFO Kubernetes file "dapp-deployment.yaml" created
INFO Kubernetes file "ganache-deployment.yaml" created
INFO Kubernetes file "react-deployment.yaml" created
```

2) \$ kubectl apply -f .

```
athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl apply -f .
deployment.apps/dapp created
service/dapp created
deployment.apps/ganache created
service/ganache created
deployment.apps/react created
service/react created
```

3) \$ kubectl get pods

```
athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
dapp-7d9cf7c5fc-dl8cs   1/1     Running   0          3m10s
ganache-59894c7d87-pqc76 1/1     Running   0          3m10s
react-6bc795df4f-2rlc7   1/1     Running   0          3m9s
athikafatima@Athikas-MacBook-Pro docker-ethereum %
```

4) \$ kubectl get services

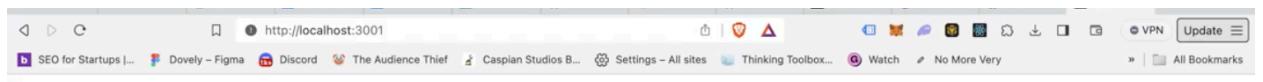
```
[athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
dapp      ClusterIP  10.104.241.143  <none>        3000/TCP    22m
ganache   ClusterIP  10.106.177.86   <none>        8545/TCP    22m
kubernetes  ClusterIP  10.96.0.1     <none>        443/TCP    23m
react     ClusterIP  10.105.111.197  <none>        3001/TCP    22m
```

5) \$ kubectl port-forward service/dapp 3000:3000

```
[athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl port-forward service/dapp 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
```

6) \$ minikube service react

```
[athikafatima@Athikas-MacBook-Pro docker-ethereum % minikube service react
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | react | | No node port |
|-----|-----|-----|
⚠️ service default/react has no node port
🏃 Starting tunnel for service react.
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | react | | http://127.0.0.1:51679 |
|-----|-----|-----|
💡 Opening service default/react in default browser...
❗ Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```



set message

Set Message
Get Message
Compile Contract
Deploy Contract

Contract compiled successfully!



=> Deployment - Replica sets

```
/Users/athikafatima/.zshrc:5: command not found: ng
● athikafatima@Athikas-MBP docker-ethereum % kubectl get pods
  NAME          READY   STATUS    RESTARTS   AGE
  dapp-6d8d4bd748-9nlmn   1/1     Running   0          112m
  ganache-59894c7d87-srlhc 1/1     Running   7 (130m ago) 33h
  my-dapp-deployment-6c98d8dc8-jxfz4 1/1     Running   0          105m
  react-6bc795df4f-72d6n   1/1     Running   7 (130m ago) 33h
● athikafatima@Athikas-MBP docker-ethereum % kubectl delete pod dapp-6d8d4bd748-9nlmn
pod "dapp-6d8d4bd748-9nlmn" deleted
● athikafatima@Athikas-MBP docker-ethereum % kubectl get pods
  NAME          READY   STATUS    RESTARTS   AGE
  dapp-6d8d4bd748-vv5mv   1/1     Running   0          9s
  ganache-59894c7d87-srlhc 1/1     Running   7 (131m ago) 33h
  my-dapp-deployment-6c98d8dc8-jxfz4 1/1     Running   0          107m
  react-6bc795df4f-72d6n   1/1     Running   7 (131m ago) 33h
○ athikafatima@Athikas-MBP docker-ethereum % █
```

=> Storage

- **kubectl apply -f pv.yaml**
- **kubectl apply -f pvc.yaml**
- **kubectl apply -f storage.yaml**

```
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f pv.yaml
persistentvolume/my-pv created
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f pvc.yaml
persistentvolumeclaim/my-pvc created
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f store.yaml
storageclass.storage.k8s.io/my-storage-class created
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f dapp-deployment.yaml
deployment.apps/dapp configured
○ athikafatima@Athikas-MBP docker-ethereum % █
```

- **Writing to file using fs**

```
server > routes > JS smart-contract-API.js > ...
1  const express = require("express");
2  const fs = require("fs");
3  const path = require("path");
4
5  const vardir = path.resolve("/var");
6
7  const myfile = path.join(vardir, "vol.txt");
8  const router = express.Router();
9
10 const logic = require("../ethereum/logic");
11
12 router.get("/", async (req, res, next) => {
13   let message = await logic.getMessage();
14   res.send(message);
15 });
16
17 router.get("/1", async (req, res, next) => {
18   fs.writeFileSync(myfile, "test-data", "utf-8");
19   res.send("working");
20 });
21
22 router.post("/", async (req, res, next) => {
23   let message = await logic.setMessage(req.body.message);
24   fs.writeFileSync(myfile, req.body.message, "utf-8");
25   res.send(message.transactionHash);
26 });
27
28 module.exports = router;
29
```

- **\$ kubectl describe deployment dapp-deployment**

```

● athikafatima@Athikas-MBP docker-ethereum % kubectl describe deployment dapp
Name:           dapp
Namespace:      default
CreationTimestamp: Sat, 27 Jan 2024 17:44:00 -0500
Labels:          io.kompose.service=dapp
Annotations:    deployment.kubernetes.io/revision: 2
                kompose.cmd: kompose convert -f docker-compose.yml
                kompose.version: 1.32.0 (HEAD)
Selector:        io.kompose.service=dapp
Replicas:        1 desired | 1 updated | 2 total | 1 available | 1 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:      io.kompose.network/docker-ethereum-default=true
                io.kompose.service=dapp
  Annotations: kompose.cmd: kompose convert -f docker-compose.yml
                kompose.version: 1.32.0 (HEAD)
  Containers:
    dapp:
      Image:      schadokar/eth-server:1.0.0
      Port:       4000/TCP
      Host Port: 4000/TCP
      Limits:
        cpu: 250m
        memory: 64Mi
      Requests:
        cpu: 250m
        memory: 64Mi
      Environment:
        NODE_PORT: <set to the key 'NODE_PORT' of config map 'node-app-config'> Optional: false
      Mounts:

```

```

Containers:
  dapp:
    Image:      schadokar/eth-server:1.0.0
    Port:       4000/TCP
    Host Port: 4000/TCP
    Limits:
      cpu: 250m
      memory: 64Mi
    Requests:
      cpu: 250m
      memory: 64Mi
    Environment:
      NODE_PORT: <set to the key 'NODE_PORT' of config map 'node-app-config'> Optional: false
    Mounts:
      /var from app-storage (rw)
Volumes:
  app-storage:
    Type:     PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName: app-pvc
    ReadOnly:  false
Conditions:
  Type        Status  Reason
  ----        ----   -----
  Available   True    MinimumReplicasAvailable
  Progressing True    ReplicaSetUpdated
OldReplicaSets: dapp-77dc4b4d8 (1/1 replicas created)
NewReplicaSet:  dapp-5cc69fb7c7 (1/1 replicas created)
Events:
  Type      Reason          Age      From            Message
  ----      ----   -----
  Normal   ScalingReplicaSet 34s     deployment-controller  Scaled up replica set dapp-5cc69fb7c7 to 1
○ athikafatima@Athikas-MBP docker-ethereum %

```

- \$ docker build -t athikaf/my-dapp:1.0 .
- \$ docker login
- \$ docker push athikaf/my-dapp:1.0

```
● athikafatima@Athikas-MBP docker-ethereum % docker build -t athikaf/my-dapp:3.0 .
[+] Building 1.6s (14/14) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                      0.0s
=> => transferring dockerfile: 598B                                         0.0s
=> [internal] load .dockerignore                                         0.0s
=> => transferring context: 74B                                         0.0s
=> [internal] load metadata for docker.io/library/node:8.12-alpine      1.4s
=> [auth] library/node:pull token for registry-1.docker.io                0.0s
=> [1/8] FROM docker.io/library/node:8.12-alpine@sha256:d75742c5fd41261113ed4706f 0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 3.66kB                                       0.0s
=> CACHED [2/8] WORKDIR /app                                           0.0s
=> CACHED [3/8] RUN apk update && apk upgrade && apk add bash git openssh 0.0s
=> CACHED [4/8] RUN apk add --update python2 krb5 krb5-libs gcc make g++ krb5-dev 0.0s
=> CACHED [5/8] RUN git config --global url."https://".insteadOf git:// 0.0s
=> CACHED [6/8] COPY ./package.json .                                     0.0s
=> CACHED [7/8] RUN npm install                                         0.0s
=> [8/8] COPY . .                                                 0.0s
=> exporting to image                                              0.0s
=> => exporting layers                                         0.0s
=> => writing image sha256:a346d4d6bf75e90637237874c48aa6f1bcc444276450a2f44be770 0.0s
=> => naming to docker.io/athikaf/my-dapp:3.0                         0.0s
```

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

- athikafatima@Athikas-MBP docker-ethereum % docker login

Authenticating with existing credentials...

Login Succeeded
- athikafatima@Athikas-MBP docker-ethereum % docker push athikaf/my-dapp:3.0

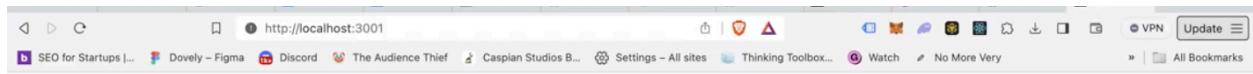
The push refers to repository [docker.io/athikaf/my-dapp]

9a615bb7d4a2: Pushed

f53333e9838c: Layer already exists

```
● athikafatima@Athikas-MBP docker-ethereum % docker login
Authenticating with existing credentials...
Login Succeeded
● athikafatima@Athikas-MBP docker-ethereum % docker push athikaf/my-dapp:4.0
The push refers to repository [docker.io/athikaf/my-dapp]
a065239cf206: Pushed
3d8a8189c1fb: Pushed
f53333e9838c: Layer already exists
34d4df02a5f5: Layer already exists
20def04f20ec: Layer already exists
296378d6a8eb: Layer already exists
c6e3b46e4f0f: Layer already exists
7f689fbc9759: Layer already exists
8b59e4cead98: Layer already exists
7aa09d2ca0a3: Layer already exists
df64d3292fd6: Layer already exists
4.0: digest: sha256:3bb5e8c87f8655c05f3f2edcb8b748aa8bacad89838602db8a33be7ae1eb9d84 size: 2627
● athikafatima@Athikas-MBP docker-ethereum % kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
dapp      1/1     1           1           20m
ganache   1/1     1           1           2d20h
react     1/1     1           1           2d20h
● athikafatima@Athikas-MBP docker-ethereum % kubectl delete deployment dapp
deployment.apps "dapp" deleted
● athikafatima@Athikas-MBP docker-ethereum % kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
ganache   1/1     1           1           2d20h
react     1/1     1           1           2d20h
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f dapp-deployment.yaml
deployment.apps/dapp created
● athikafatima@Athikas-MBP docker-ethereum % kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
dapp      1/1     1           1           9s
ganache   1/1     1           1           2d20h
react     1/1     1           1           2d20h
○ athikafatima@Athikas-MBP docker-ethereum %
```

```
athikafatima@Athikas-MBP docker-ethereum % sudo lsof -i :4000
athikafatima@Athikas-MBP docker-ethereum % kubectl port-forward service/dapp 4000:4000
Forwarding from 127.0.0.1:4000 -> 4000
Forwarding from [::1]:4000 -> 4000
Handling connection for 4000
```



Contract compiled successfully!



```

○ athikafatima@Athikas-MBP docker-ethereum % kubectl exec -it my-dapp-deployment-658bfc8d7b-rx8m4 -- /bin/bash
bash-4.4# ls
Dockerfile          README.md           dapp-service.yaml   ganache-deployment.yaml   package.json      react-service.yaml    test-user.csr
Dockerfile.ganache config-map.yaml     docker-compose.yml ganache-service.yaml   pvc.yaml        server             store.yaml
LICENSE            dapp-deployment.yaml   ethereum          node_modules       package-lock.json react-deployment.yaml test-user.key
LICENSE.md          dapp-hpa.yaml      frontend-hpa.yaml
bash-4.4# cd ..
bash-4.4# ls
app   bin   dev   etc   home  lib   media  mnt   opt   proc  root  run   sbin  srv   sys   tmp   usr   var
bash-4.4# cd var
bash-4.4# ls
cache  empty  git   lib   local  lock  log   opt   run   spool  tmp   vol.txt
bash-4.4# cat vol.txt
test-database-4.# ls
cache  empty  git   lib   local  lock  log   opt   run   spool  tmp   vol.txt
bash-4.4# 

```

```

○ athikafatima@Athikas-MBP docker-ethereum % kubectl exec -it dapp-6b6d96b4d7-ff2b2 -- /bin/bash
bash-4.4# ls
Dockerfile          dapp-hpa.yaml      node_modules      server
Dockerfile.ganache  dapp-service.yaml  package-lock.json store.yaml
LICENSE            docker-compose.yml   package.json      test-user.crt
LICENSE.md          ethereum          pvc.yaml        test-user.csr
README.md          frontend-hpa.yaml  react-deployment.yaml test-user.key
config-map.yaml    ganache-deployment.yaml react-service.yaml
dapp-deployment.yaml  ganache-service.yaml
bash-4.4# cd ..
bash-4.4# ls
app   dev   home  media  opt   root  sbin  sys   usr
bin   etc   lib   mnt   proc  run   srv   tmp   var
bash-4.4# cd var
bash-4.4# ls
myfile.txt  run
bash-4.4# cat myfile.txt
Hello, Professor!
bash-4.4# 

```

=> Scaling

- **minikube addons enable metrics-server**
- **kubectl apply -f dapp-hpa.yaml**
- **kubectl apply -f frontend-hpa.yaml**
- **kubectl get hpa**

```

● athikafatima@Athikas-MBP docker-ethereum % minikube addons enable metrics-server
💡 metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ■ Using image registry.k8s.io/metrics-server/metrics-server:v0.6.4
  ★ The 'metrics-server' addon is enabled
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f dapp-hpa.yaml
horizontalpodautoscaler.autoscaling/dapp-hpa created
● athikafatima@Athikas-MBP docker-ethereum % kubectl get hpa
  NAME      REFERENCE      TARGETS      MINPODS      MAXPODS      REPLICAS      AGE
  dapp-hpa  Deployment/dapp  0%/50%     1           5           1           69s
○ athikafatima@Athikas-MBP docker-ethereum % 

```

After testing by sending traffic:

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
dapp-hpa	Deployment/dapp	0%/50%	1	5	1	38m
frontend-hpa	Deployment/react	21%/50%	1	5	1	38m

=> User Role Creation

- `- openssl genpkey -algorithm RSA -out test-user.key`
 - `- openssl req -new -key test-user.key -out test-user.csr -subj "/CN=test-user/O=group1"`
 - `- openssl x509 -req -in test-user.csr -CA ~/.minikube/ca.crt -CAkey ~/.minikube/ca.key -CAcreateserial -out test-user.crt -days 500`
 - `kubectl config set-credentials test-user --client-certificate=test-user.crt --client-key=test-user.key`
 - `kubectl config set-context test-user-context --cluster=minikube --user=test-user`
 - `kubectl --context=test-user-context get pods`
 - `kubectl create role example-role --verb=get,list,watch --resource=pods`
 - `kubectl create rolebinding example-binding --role=example-role --user=test-user`
 - `kubectl --context=test-user-context get pods`
 - `kubectl --context=test-user-context delete pod`

- athikafatima@Athikas-MBP docker-ethereum % kubectl create role example-role --verb=get,list,watch --resource=pods role.rbac.authorization.k8s.io/example-role created
- athikafatima@Athikas-MBP docker-ethereum % kubectl create rolebinding example-binding --role=example-role --user=test-user rolebinding.rbac.authorization.k8s.io/example-binding created
- athikafatima@Athikas-MBP docker-ethereum % kubectl --context=test-user-context get pods

NAME	READY	STATUS	RESTARTS	AGE
dapp-5cc69fb7c7-p9nmr	1/1	Running	0	37m
ganache-59894c7d87-srlhc	1/1	Running	2 (48m ago)	27h
react-6bc795df4f-72d6n	1/1	Running	2 (48m ago)	28h

- ✖ athikafatima@Athikas-MBP docker-ethereum % kubectl --context=test-user-context delete pod dapp-5cc69fb7c7-p9nmr
Error from server (Forbidden): pods "dapp-5cc69fb7c7-p9nmr" is forbidden: User "test-user" cannot delete resource "pods" in API group "" in the namespace "default"
- athikafatima@Athikas-MBP docker-ethereum % █

=> Secrets (ConfigMap)

```
● athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl get pods
  NAME           READY   STATUS    RESTARTS   AGE
  dapp-65cc88b56c-gs86m   1/1     Running   0          41s
  ganache-59894c7d87-c65jk   0/1     ContainerCreating   0          41s
  react-6bc795df4f-4g2z5   0/1     Pending   0          41s
● athikafatima@Athikas-MacBook-Pro docker-ethereum % kubectl exec dapp-65cc88b56c-gs86m -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=dapp-65cc88b56c-gs86m
NODE_PORT=8080
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
DAPP_SERVICE_HOST=10.101.169.42
DAPP_SERVICE_PORT_3000=3000
GANACHE_PORT=tcp://10.104.192.226:8545
REACT_PORT_3001_TCP_PROTO=tcp
REACT_SERVICE_PORT_3001=3001
DAPP_PORT_3000_TCP=tcp://10.101.169.42:3000
GANACHE_PORT_8545_TCP_PROTO=tcp
REACT_SERVICE_HOST=10.110.212.250
DAPP_PORT=tcp://10.101.169.42:3000
GANACHE_SERVICE_HOST=10.104.192.226
GANACHE_SERVICE_PORT_8545=8545
GANACHE_PORT_8545_TCP=tcp://10.104.192.226:8545
GANACHE_PORT_8545_TCP_PORT=8545
REACT_PORT_3001_TCP_ADDR=10.110.212.250
REACT_PORT_3001_TCP=tcp://10.110.212.250:3001
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
DAPP_PORT_3000_TCP_PORT=3000
REACT_PORT=tcp://10.110.212.250:3001
GANACHE_SERVICE_PORT=8545
```

=> Load Balancing

```
● athikafatima@Athikas-MBP docker-ethereum % kubectl apply -f react-service.yaml
service/react created
● athikafatima@Athikas-MBP docker-ethereum % kubectl get services
  NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
  dapp      ClusterIP      10.101.169.42      <none>      3000/TCP      39h
  ganache   ClusterIP      10.104.192.226      <none>      8545/TCP      39h
  kubernetes   ClusterIP      10.96.0.1      <none>      443/TCP      39h
  react      LoadBalancer      10.101.199.181      <pending>      3001:31095/TCP      9s
○ athikafatima@Athikas-MBP docker-ethereum %
```

```

● athikafatima@Athikas-MBP docker-ethereum % kubectl get services
  NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
  dapp      ClusterIP    10.101.169.42    <none>        3000/TCP    39h
  ganache   ClusterIP    10.104.192.226   <none>        8545/TCP    39h
  kubernetes ClusterIP    10.96.0.1       <none>        443/TCP    39h
  react     LoadBalancer 10.101.199.181   <pending>    3001:31095/TCP  9s

○ athikafatima@Athikas-MBP docker-ethereum % minikube service react
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | react | http/3001 | http://192.168.49.2:31095 |
|-----|-----|-----|-----|
  ✨ Starting tunnel for service react.

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | react | http/3001 | http://127.0.0.1:56211 |
|-----|-----|-----|-----|
  🚀 Opening service default/react in default browser...
  ! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.

```



set message

Set Message
Get Message
Compile Contract
Deploy Contract