目录

[1 IPFS Implementation 1](#_Toc5731)

[Step1: start a local ipfs node 1](#_Toc17945)

[Step2: use ipfs http client to interact with IPFS 2](#_Toc10484)

[Step3: integrate ipfs to react 2](#_Toc13468)

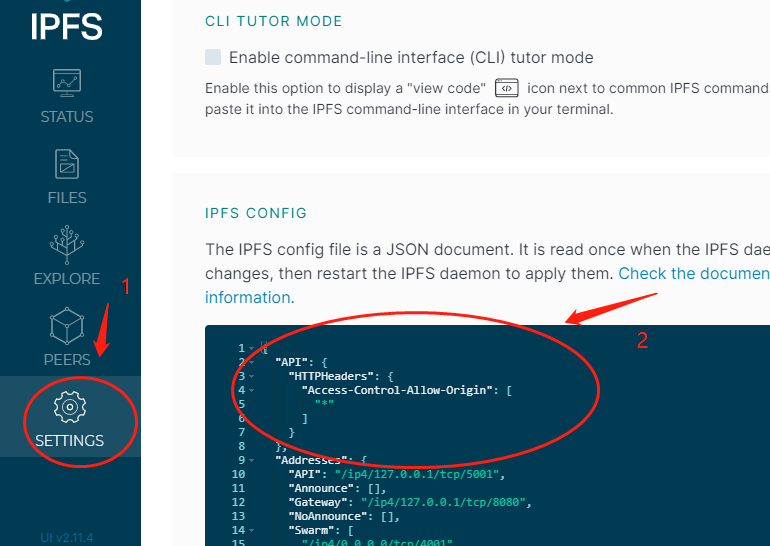
[2 Setting up a IPFS Cluster 3](#_Toc5735)

[3 OrbitDB Implementation 4](#_Toc15981)

# 1 IPFS Implementation

## Step1: start a local ipfs node

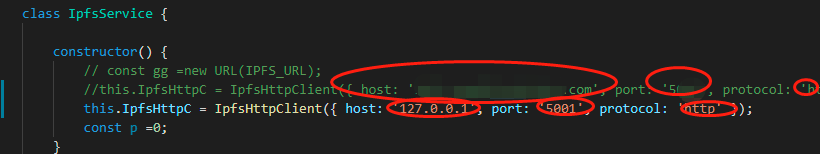
And add the cors to it



1 click setting

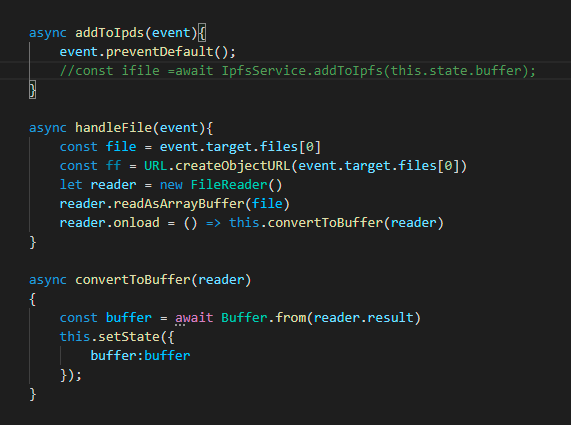
2 set Access-Control-Allow-Origin to be ‘\*’

## Step2: use ipfs http client to interact with IPFS

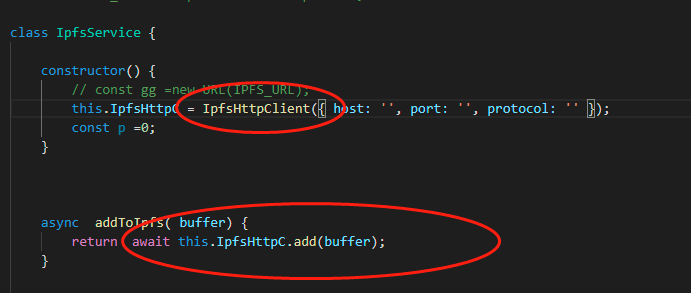


We could either connect to a ipfs gateway or we could connect to localhost with cors open

## Step3: integrate ipfs to react



Store files in buffer state variable



Upload buffer to ipfs-client

# 2 Setting up a IPFS Cluster

Step1: download docker-compose file

from: <https://github.com/ipfs/ipfs-cluster/blob/master/docker-compose.yml>

Step2: start docker container in 3 nodes

Step3: create cluster secrete key and set the key as a bash environment variable

Now we need to generate CLUSTER\_SECRET and set it as an environment variable for all peers participating in the cluster. Sharing the same CLUSTER\_SECRET allow peers to understand that they are part of one IPFS-Cluster. We will generate this key on the zero node and then copy it to all other nodes.



Randomly generate a secret key

Step4: start the docker containers up

docker-compose up

docker ps //当前运行的容器

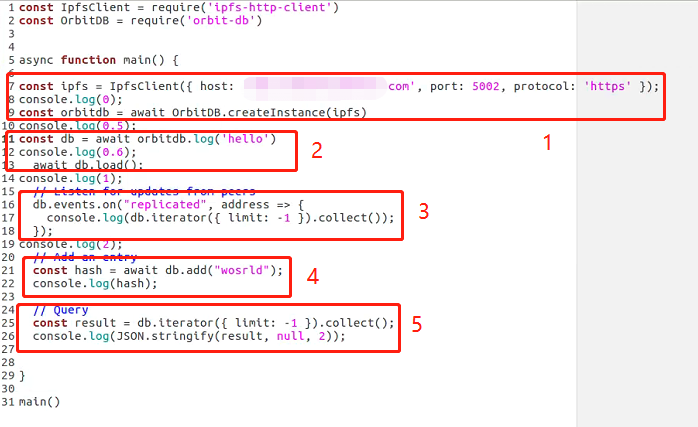
docker images //当前存在的镜像

Step 5: verify

ipfs-cluster-ctl peers ls



# 3 OrbitDB Implementation



-1 db initializer

-2 create database

-3 events listener, sync data

-4 write to db

-5 query data