Chainlink node

cat .chainlink-kovan/.env

ROOT=/chainlink

LOG\_LEVEL=debug

ETH\_CHAIN\_ID=42

MIN\_OUTGOING\_CONFIRMATIONS=2

LINK\_CONTRACT\_ADDRESS=0xa36085F69e2889c224210F603D836748e7dC0088

CHAINLINK\_TLS\_PORT=0

SECURE\_COOKIES=false

GAS\_UPDATER\_ENABLED=true

ALLOW\_ORIGINS=\*

ETH\_URL=wss://kovan.infura.io/ws/v3/[PUT\_INFURA\_KEY\_HERE]

DATABASE\_URL=postgresql://postgres:mysecretpassword@172.17.0.2:5432/chainlink

Install docker

curl -sSL https://get.docker.com/ | sh

sudo usermod -aG docker ludo

Logout/Login

Pull postgres dcontainer image

docker pull postgres

* You should [configure Postgres](https://www.postgresql.org/docs/12/ssl-tcp.html) to use SSL connection

<https://www.postgresql.org/docs/12/ssl-tcp.html>

* 17.9.3. Creating a Self-signed Certificate  
  openssl req -new -text -out server.req  
  openssl rsa -in privkey.pem -out server.key  
  rm privkey.pem  
  openssl req -x509 -in server.req -text -key server.key -out server.crt  
  chmod og-rwx server.key

<https://github.com/docker-library/docs/blob/master/postgres/README.md#database-configuration>

docker run -i --rm postgres cat /usr/share/postgresql/postgresql.conf.sample > my-postgres.conf

https://itnext.io/postgresql-docker-image-with-ssl-certificate-signed-by-a-custom-certificate-authority-ca-3df41b5b53

mkdir -p ~/postgres

Caution: if ~/postgres exists and not empty, DB initialization is not replayed (be sure the folder is empty when starting the container first, or when previous attempts have failed).

docker run --name some-postgres -p 5432:5432 --user "$(id -u):$(id -g)" -v /etc/passwd:/etc/passwd:ro -e POSTGRES\_PASSWORD=mysecretpassword -e POSTGRES\_DB=chainlink -v $PWD/postgres:/var/lib/postgresql/data -v $PWD/server.crt:/var/lib/postgresql/server.crt -v $PWD/server.key:/var/lib/postgresql/server.key postgres -c ssl=on -c ssl\_cert\_file=/var/lib/postgresql/server.crt -c ssl\_key\_file=/var/lib/postgresql/server.key

Install utils

sudo apt install postgresql-client-common

sudo apt install postgresql-client

docker inspect some-postgres 🡪 get NetworkSettings.IPAddress

pg\_isready --host=172.17.0.2

docker pull alpine

docker run --name some-alpine -i -t alpine

# hostname -I 🡪 show IP address from container

# apk update

# apk add postgresql-client

# pg\_isready --host=172.17.0.2

172.17.0.2:5432 - accepting connections

# pg\_isready --host=172.17.0.1

172.17.0.1:5432 - accepting connections

psql --host=172.17.0.1 -U postgres

Enter password: mysecretpassword

export USERNAME=postgres

export DATABASE=$USERNAME

export PASSWORD=mysecretpassword

export SERVER=172.17.0.2

export PORT=5432

export DATABASE=chainlink

docker stop some-postgres

docker rm some-postgres

cd ~/.chainlink-kovan && docker run --name some-chainlink -p 6688:6688 -v ~/.chainlink-kovan:/chainlink -it --env-file=.env smartcontract/chainlink local n -p /chainlink/.password -a /chainlink/.api

* Go to <http://192.168.1.5:6688/signin>

Deploy Oracle contract with Remix to Kovan (with token @=0xa36085F69e2889c224210F603D836748e7dC0088).

* 0x7bfc47C0280F7F491a3118946771b7BaD6ad6808

In Remix, call the setFulfillmentPermission function with the address of your node, a comma, and the value true, as the input parameters. -> "0xc6d5785af62d024316a38495E889e91a8a9758ec",true

Add jobs to the node 🡪 858e66b75f2940eea189bf508955f29a

MATIC Mumbai

Node Address: ~~0x5cc62e2D1F9541C919512AD1696A6E2847dbb98F~~ 0xE7E2bc5e15ef1db1079863fbB551b98fb7e58CDe

LINK Token: ~~0xACD928776d0b44E698406b6DCa2BE0cEF4b7e792~~ 0xE053E117d372Cd7C4a42DeC0F60B59f2d524f147

Oracle deployed at ~~0xe1b3Fa15d00d011F4572Ca92C708EB475DA6b00E~~ ~~0xCf2f3ab8EC96fbBc2AFCA5d7947B5548f2BBF035~~ 0x71EbC24B00F65D6cD5848CeF5971595DaB6027F9

~~Pointer deployed at 0x52106f096Cae1b0cD3BD0486b380392853497f47~~

BlockhashStore deployed to: 0x75cea4BFC91A06d8D5cE319411cEDDEd8748eD63

VRFCoordinator deployed to: 0x0a5eE50F415F95481d399b0E1320aa9c9B987d98