<https://docs.bigchaindb.com/projects/server/en/latest/quickstart.html>

# Prerequisite

Vagrant boxes up and running with a specific IP Address. See multichain document for details.

We are using config.vm.box = "ubuntu/xenial64"

# in the Vagrant Box:

## install rethinkdb

source /etc/lsb-release **&&** echo "deb http://download.rethinkdb.com/apt $DISTRIB\_CODENAME main" | sudo tee /etc/apt/sources.list.d/rethinkdb.list wget -qO- https://download.rethinkdb.com/apt/pubkey.gpg | sudo apt-key add - sudo apt-get update sudo apt-get install rethinkdb

## In instance 1 of Box 1 : Launch rethinkdb

$ rethinkdb

## In instance 2 of Box 1 (new terminal) : Install other tools on ubuntu

$ sudo apt-get update

$ sudo apt-get install g++ python3-dev libffi-dev

$ sudo apt-get install python3-pip

$ sudo pip3 install --upgrade pip setuptools

if problems : try following

$ export LC\_ALL=en\_US.UTF-8

$ export LANG=en\_US.UTF-8

$ export LANGUAGE=en\_US.UTF-8

<https://github.com/bigchaindb/bigchaindb/issues/1008>

## In instance 2 of Box 1 Install Bigchain

$ sudo pip3 install bigchaindb

$ bigchaindb -y configure rethinkdb

## In instance 2 of Box 1 Run Bigchain

$ bigchaindb start

this will create a config file

INFO:bigchaindb.commands.bigchain:BigchainDB Version 0.9.1

INFO:bigchaindb.config\_utils:Configuration loaded from `/home/ubuntu/.bigchaindb`

🡪 need to update this config file and replace localhost with the ip addres ( do ip a to get it).

{

"backlog\_reassign\_delay": 120,

"server": {

"workers": null,

"bind": "localhost:9984",

"threads": null

},

"database": {

"port": 28015,

"host": "localhost",

"backend": "rethinkdb",

"name": "bigchain"

## In instance 2 of Box 1 stop / restart bigchaindb

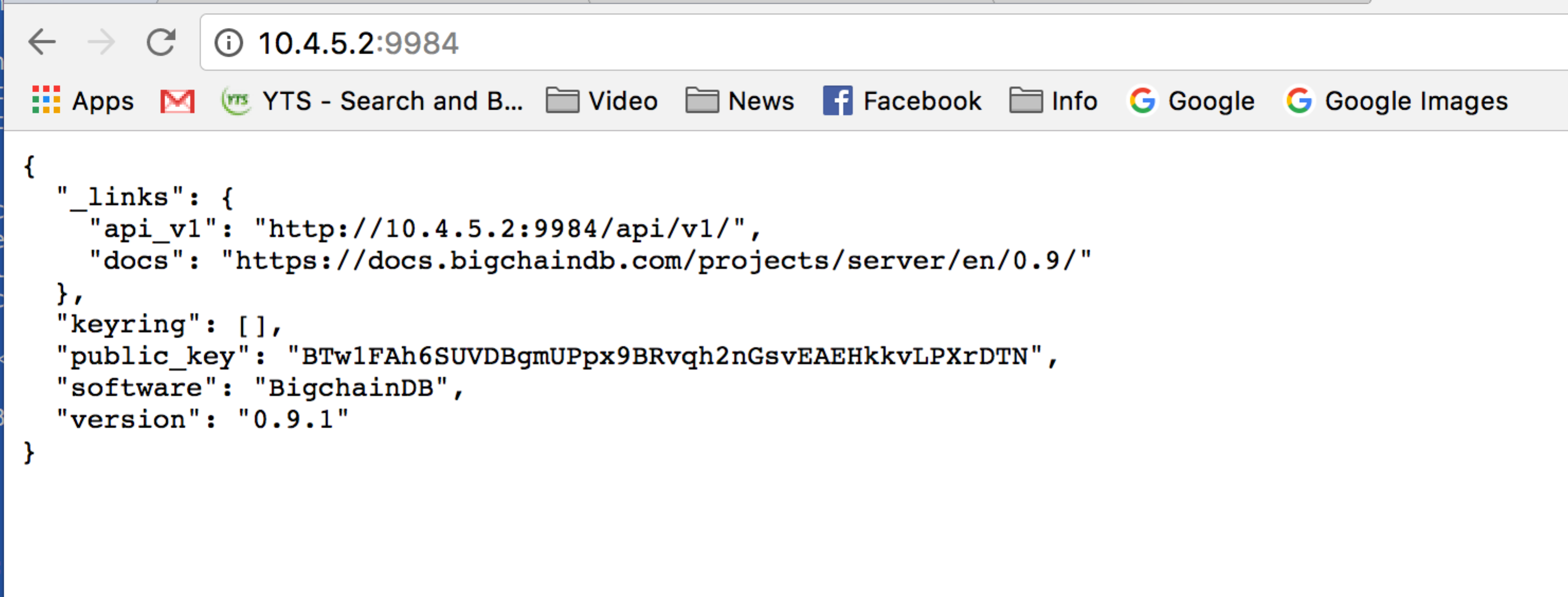
Stop the running with CTRL+C

Run again

$ bigchaindb start

## In OS : Check you can reach config file

This is explained in <https://docs.bigchaindb.com/projects/py-driver/en/latest/quickstart.html>

now a chrome can access it.

## In instance 3 of Box 1 (new terminal) : Install Python tools

This is explained in <https://docs.bigchaindb.com/projects/py-driver/en/latest/quickstart.html>

Ultimately we want to run

$ pip install bigchaindb\_driver

but before this : prerequisites

$ sudo apt-get update

$ sudo apt-get install libffi-dev

need python 3.5

$ python –version

$ pip –version

$ pip install --upgrade setuptools

## In instance 3 of Box 1 : run a python to create asset and record transaction

The .py file on local OS (in vagrant folder) will be reachable in the box in /vagrant

Update test.py with hereunder text. Then :

$ python test.py

result : (depending on the code)

ubuntu@xen-2:/vagrant$ python 5.py

Tx valid in: 0 secs

Is Bob the new owner 2222? True

was alice before ??? True

example python transaction :

from bigchaindb\_driver import BigchainDB

from bigchaindb\_driver.crypto import generate\_keypair

from time import sleep

from sys import exit

alice, bob, amit = generate\_keypair(), generate\_keypair(), generate\_keypair()

bdb = BigchainDB('http://10.4.5.2:9984')

article\_asset = {

'data': {

'article1': {

'http': 'http://www.bluetrails.nl',

'author': 'An de Rijdt'

},

},

}

article\_asset\_metadata = {

'planet': 'earth'

}

prepared\_creation\_tx = bdb.transactions.prepare(

operation='CREATE',

signers=amit.public\_key,

asset=article\_asset,

metadata=article\_asset\_metadata

)

fulfilled\_creation\_tx = bdb.transactions.fulfill(

prepared\_creation\_tx,

private\_keys=amit.private\_key

)

sent\_creation\_tx = bdb.transactions.send(fulfilled\_creation\_tx)

txid = fulfilled\_creation\_tx['id']

trials = 0

while trials < 60:

try:

if bdb.transactions.status(txid).get('status') == 'valid':

print('Tx valid in:', trials, 'secs')

break

except bigchaindb\_driver.exceptions.NotFoundError:

trials += 1

sleep(1)

if trials == 60:

print('Tx is still being processed... Bye!')

exit(0)

asset\_id = txid

transfer\_asset = {

'id': asset\_id

}

output\_index = 0

output = fulfilled\_creation\_tx['outputs'][output\_index]

transfer\_input = {

'fulfillment': output['condition']['details'],

'fulfills': {

'output': output\_index,

'txid': fulfilled\_creation\_tx['id']

},

'owners\_before': output['public\_keys']

}

prepared\_transfer\_tx = bdb.transactions.prepare(

operation='TRANSFER',

asset=transfer\_asset,

inputs=transfer\_input,

recipients=bob.public\_key,

)

fulfilled\_transfer\_tx = bdb.transactions.fulfill(

prepared\_transfer\_tx,

private\_keys=amit.private\_key,

)

sent\_transfer\_tx = bdb.transactions.send(fulfilled\_transfer\_tx)

print("Is Bob the new owner 2222?",

sent\_transfer\_tx['outputs'][0]['public\_keys'][0] == bob.public\_key)

print("was alice before ???",

fulfilled\_transfer\_tx['inputs'][0]['owners\_before'][0] == amit.public\_key)