Docker for Super Node Remote

1. Make a docker file with below info

(Dockerfile and pastel-utility file need to be in same destination).

FROM ubuntu:20.04

RUN apt-get update \

&& apt-get install -y wget \

&& apt-get -qq -y install curl\

&& apt-get install -y libgomp1\

&& apt-get install -y iputils-ping\

&& apt-get install -y zip unzip php-zip\

&& apt-get install -y ssh\

&& apt-get install -y expect\

&& rm -rf /var/lib/apt/lists/\*

COPY ./pastel-utility /

RUN chmod 777 /pastel-utility

EXPOSE 22

1. Build docker image with below command.

$ sudo docker build -t ubuntu-demo .

1. Make 2 containers from that image.

$ sudo docker run --name miner -d -t ubuntu-demo

$ sudo docker run --name supernode -d -t ubuntu-demo

$ sudo docker run --name remotenode -d -t ubuntu-demo

1. Install super node in each container.

$ sudo docker exec miner /bin/sh -c "./pastel-utility install supernode"

$ sudo docker exec supernode /bin/sh -c "./pastel-utility install supernode --peers 192.168.114.1"

$ sudo docker exec remote /bin/sh -c "./pastel-utility install supernode --peers 192.168.114.1, 192.168.114.2"

1. Create Network for docker containers and assign IP to containers

$ sudo docker network create --subnet 192.168.0.0/16 --ip-range 192.168.240.0/20 pastel-host-network

$ sudo docker network connect --ip 192.168.114.1 pastel-host-network miner

$ sudo docker network connect --ip 192.168.114.2 pastel-host-network supernode

$ sudo docker network connect --ip 192.168.114.3 pastel-host-network remotenode

1. Update pastel.conf file in each container.

$ sudo docker exec miner /bin/sh -c "echo 'testnet=1\ngen=1' >> /root/.pastel/pastel.conf"

$ sudo docker exec supernode /bin/sh -c "echo 'testnet=1' >> /root/.pastel/pastel.conf"

$ sudo docker exec remotenode /bin/sh -c "echo 'testnet=1' >> /root/.pastel/pastel.conf"

1. Run pasteld in each container

$ sudo docker exec miner /bin/sh -c "/root/.pastel/pasteld --mine --daemon --testnet --reindex"

$ sudo docker exec supernode /bin/sh -c "/root/.pastel/pasteld --daemon --testnet --reindex"

$ sudo docker exec remotenode /bin/sh -c "/root/.pastel/pasteld --daemon --testnet --reindex"

-You could check pasteld is running properly.

$ sudo docker exec supernode /bin/sh -c "/root/.pastel/pastel-cli getinfo"

$ sudo docker exec miner /bin/sh -c "/root/.pastel/pastel-cli getinfo"

$ sudo docker exec remotenode /bin/sh -c "/root/.pastel/pastel-cli getinfo"

1. Get account address in supernode container

$ sudo docker exec remotenode /bin/sh -c "/root/.pastel/pastel-cli getaccountaddress ''"

E.g output: **tPTrtD4BKx5Qr2LLQE8Wb4HWh52VGfoiuf9**

1. Send coins from miner to supernode container with account address, get transaction id.

$ sudo docker exec miner /bin/sh -c "/root/.pastel/pastel-cli sendmany '' '{\"**tPTrtD4BKx5Qr2LLQE8Wb4HWh52VGfoiuf9**\": 1000000}'"

Expected error: "Account has insufficient funds: needs 1000000.00 coins; has only 0.00 coins spendable "

(In this case, need to copy exact pasteld file into miner container and retry.)

$ sudo docker exec miner /bin/sh -c "pkill pasteld"

$ sudo docker cp pasteld miner:/root/.pastel/

$ sudo docker exec miner /bin/sh -c "/root/.pastel/pasteld --mine --daemon --testnet --reindex"

$ sudo docker exec miner /bin/sh -c "/root/.pastel/pastel-cli sendmany '' '{\"**tPTrtD4BKx5Qr2LLQE8Wb4HWh52VGfoiuf9**\": 1000000}'"

E.g output: **c96d64f13beb3ff55277b03f11d0fa2445b2dda537ca1026a23b748895df83b3**

1. Start super node with outputted transaction id in supernode container

-You need to make file named script.exp with below script.

*#!/usr/bin/expect -f*

*spawn ./pastel-utility start supernode --name=masternode1 --ip=188.43.136.32 --create --rpc-ip=192.168.114.147 --rpc-port=9933 --txid=****c96d64f13beb3ff55277b03f11d0fa2445b2dda537ca1026a23b748895df83b3*** *--ind=1 --passphrase=4342 --ssh-ip=192.168.114.3 --network=testnet --coldhot*

*set timeout 800*

*expect -exact "Enter Username: "*

*send "root\n"*

*expect -exact "Enter Password: "*

*send "a\n"*

*sleep 100*

*set timeout 100*

*expect "input index of pasteld path to use"*

*send "0\n"*

*sleep 100*

*set timeout 1200*

*expect "root@192.168.114.3's password: "*

*send "a\n"*

*sleep 100*

*set timeout 100*

*Interact*

-Need to copy file to supernode container.

$ sudo docker cp script.exp supernode:/

-Need to open Port 22 and make a password , enable PasswordAuthentication yes, StrictHostKeyChecking value to no.

#open ssh\_config of remotenode container and open and change PasswordAuthentication , StrictHostKeyChecking , Port values in ssh\_config file like below.

(Important: remove #)

PasswordAuthentication yes

StrictHostKeyChecking no

Port 22

#open sshd\_config of remotenode container and change PasswordAuthentication & PermitRoot Login value like below.

PermitRootLogin yes

PasswordAuthentication yes

#restart ssh service

$sudo docker exec remotenode /bin/sh -c "service ssh restart"

#set up remotenode password.

$sudo docker exec -it remotenode bash

$root@remotenode:/# passwd

Enter new UNIX password: a

Retype new UNIX password: a

1. Run script.exp and wait until done.

-Run script.exp

$ sudo docker exec supernode /bin/sh -c "expect script.exp"

-Once it synced and output like below it means success.

Remote:::Supernode was started successfully

-You can check it started exactly with below.

$ sudo docker exec remotenode /bin/sh -c "ps -A"

PID TTY TIME CMD

1 pts/0 00:00:00 bash

66 ? 00:00:00 sshd

85 ? 00:00:10 pasteld

**150 ? 00:00:00 supernode-linux**

156 ? 00:00:00 sh

163 ? 00:00:00 ps

If supernode process is alive like above, it means it started supernode exec file exactly.

-Expected error:

Can’t see supernode-linux in process list though it says “Remote:::Supernode was started successfully”.

It means supernode-linux execution file has wrong type.

Currently it has supernode-linux execution file for Windows not Ubuntu.

In this case, you can download it from Git and replace it to remotenode container and run again.

$ sudo docker cp supernode-linux-amd64 remotenode:/root/.pastel/

$ sudo docker exec supernode /bin/sh -c "expect script.exp"

Then it will be okay.

Thanks.