**HashCash Consultants**

13-SEPT-2019

**Setting up HC NET on CentOS**

**Setup instructions for developers**

**HashCash Development Center**

**Copyright Notice**

©**2019** Hashcash Consultants. All Rights Reserved. This documentation is the sole property of Hashcash Consultants. Hashcash Consultants believes the information in this document or page is accurate as of its publication date; such information is subject to change without notice. Hashcash Consultants acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. This document is not for general distribution and is meant for use solely by the person or entity that it has been specifically issued to and can be used for the sole purpose it is intended to be used for as communicated by Hashcash Consultants in writing. Except as expressly permitted by Hashcash Consultants in writing, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior written permission of Hashcash Consultants and/ or any named intellectual property rights holders under this document.

Setting up HC NET on CentOS

## **Build Dependencies**

* CPU - dual core 2.5 GHz Intel Xeon
* RAM - 8gb
* Storage - 500gb
* c++ toolchain and headers that supports c++14
  + clang >= 5.0
  + g++ >= 6.0
* pkg-config
* libpq-dev unless you ./configure --disable-postgres in the build step below.
* 64-bit system
* clang-format-5.0 (for make format to work)
* Postgres db >=9.3 (aurora requires version more than 9.3)

## **General CentOS setup**

If you just started a fresh instance of CentOS you should take some steps before the installation of hcnet-core to assure the system is secure and up to date.

The first thing to do is to run:

sudo yum upgrade

sudo yum install vim

This will get the repository and all system packages up to date. And of course give vim so that we can more easily edit configuration files.

## **Building hcnet-core**

We will be compiling hcnet-core from source and need the developer tools installed:

sudo yum groupinstall 'Development Tools'

sudo yum install postgresql-devel

### 

### 

### **Building gcc,Clang,c++,g++**

1. Run these below commands to install gcc

sudo yum install centos-release-scl

sudo yum install devtoolset-7-gcc\*

scl enable devtoolset-7 bash

which gcc

Check gcc, g++ version and c++ version

gcc --version

g++ --version

c++ --version

1. Run these below commands to install clang

yum install devtoolset-7 llvm-toolset-7

scl enable devtoolset-7 llvm-toolset-7 bash

clang --version

1. Run these below commands to install libc++

sudo rpm -i https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

sudo yum install svn

### **Configuring libpq**

At one point during the hcnet-core compilation you would run **./configure** and it would fail with the following error:

No package 'libpq' found

The problem is that when you install PostgreSQL dev tools on CentOS it doesn't install the required file for pkg-config to find it. To solve this you can create the file manually.

**/usr/lib64/pkgconfig/libpq.pc**:

prefix=/usr

libdir=${prefix}/lib64

includedir=${prefix}/include/pgsql

Name: LibPQ

Version: 5.5.0

Description: PostgreSQL client library

Requires:

Libs: -L${libdir}/libpq.so -lpq

Cflags: -I${includedir}

## **Basic Installation**

* git clone https://github.com/HashCash-Consultants/HCNet-Core
* cd HCNet-Core
* git submodule init
* git submodule update
* Type ./autogen.sh.
* Type ./configure
* Type make -j6 (it will take several minutes ).

After the process is completed you can see the HcNet-core build inside /home/centos/HCNet-Core/src

**Aurora setup**

To build aurora you must have Go and dep in your centOS environment. To install Go you need to perform the below steps. Go version must be >=1.9

**Install go**

* sudo curl -O <https://storage.googleapis.com/golang/go1.9.1.linux-amd64.tar.gz>
* sudo tar -xvf go1.9.1.linux-amd64.tar.gz
* Sudo vi .bash\_profile to set GOPATH, GOBIN,GOROOT
* source ~/.bash\_profile
* Check go version
* Go --version

# **Install dep**

* sudo -s
* echo $PATH copy it and update path
* Set path via export PATH=”{previous path}:/home/centos/go/bin”
* Set GOPATH export GOPATH=”/home/centos/go”
* Set GOROOT export GOROOT=”/home/centos/go”
* curl <https://raw.githubusercontent.com/golang/dep/master/install.sh> | sh
* Check dep version dep version

# **Install aurora**

* mkdir github.com/hcnet in /home/centos/go/src
* go to /home/centos/go/src/github.com/hcnet and execute following command:
* git clone <https://github.com/HashCash-Consultants/go.git>
* go to /home/centos/go/src/github.com/hcnet /go and execute following command:

$ sudo apt-get install mercurial

$ dep ensure –v

* go to /home/centos/go/src and execute following command:

$ go install -ldflags "-X github.com/hcnet/go/support/app.version=aurora-0.16.0" github.com/hcnet/go/services/aurora/

* after running the above command, check aurora build in <Your\_dir>/go/bin folder. You can check Aurora version by using the following command:

./aurora version

# **Postgres database installation**

First add the latest version to your rpm for install using yum.

Run this below command

sudo rpm -Uvh <http://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm>

Then run the install command:

sudo yum install postgresql96-server postgresql96

Initialize with this:

sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb

Next edit the pg\_hba.conf file, in the same folder:

sudo vim var/lib/pgsql/9.6/data/pg\_hba.conf

Scroll to the bottom of the file and add these lines if they don’t already exist:

#IPv4 remote connections (all users and IP addresses):

host all all 0.0.0.0/0 md5

And update the following lines from

host all all 127.0.0.1/32 ident

host all all          ::1/128 ident

to

host all all 127.0.0.1/32 md5

host all all          ::1/128 md5

On the second line beginning with ‘host’ make sure there is no # added. You want it to read as above.

Exit saving changes.

Now start and enable postgresdb

sudo systemctl start postgresql-9.6

sudo systemctl enable postgresql-9.6

Check postgresdb status

sudo systemctl status postgresql-9.6

# **Aurora database setup**

Create a user for HC Net aurora database.

$ sudo -s

$ su – postgres

$ createuser <username> --pwprompt

$ Enter password for new role: <Enter password>

$ Enter it again: <Enter the pwd again>

You need to add Aurora user. Exit from postgres and login as root user and execute following command.

$ exit

$ adduser <username>;

To verify if user is created, execute following commands

$ su - postgres

$ psql

$ \du

Create a blank database using following command.

$ CREATE DATABASE <DB\_NAME> OWNER <user created>;

# **Initialize aurora**

Initialize aurora with database login as root user and Go “go/bin” using following command

$ export DATABASE\_URL="postgresql://define aurora db username:define aurora db user password@localhost/define aurora database name?sslmode=disable"

After that Go “go/bin” and execute following command

$ ./aurora db init

## **Node db setup**

Postgres user for HCNet core

sudo -s

su – postgres

createuser <username> --pwprompt

Enter password for new role: <Enter password>

Enter it again: <Enter the pwd again>

Note: This is required for DB url that need to be maintained in HcNet-core.cfg and if you want to set up 5 nodes, you need to create 5 users.

After creating the user, you need to add them. So exit from postgres and login as root user.

exit

adduser <username>;

          To verify if user is created, execute following commands

su - postgres

psql

\du

           After that create a database. If you have five cores, then create five databases.

CREATE DATABASE <DB\_NAME> OWNER <user created username>;

## **Node configuration file setup**

 As per your rquired number of cores create folder like node1,node2… inside

/home/centos/ and copy HcNet-core from /home/centos to each created node.

* Generate network seed

goto all /home/centos/node1/HCNet-Core/src/

sudo ./HcNet-core gen-seed

save the above secret seed and public key for future use.

* Create cfg file in all node

goto all /home/centos/node1/HCNet-Core/

touch HCNet-Core.cfg

* Update cfg file with all the node details in each node . For example.

NODE 1

------------------------------------------

LOG\_FILE\_PATH="/home/centos/Node1/HCNet-Core/logs"

BUCKET\_DIR\_PATH="/home/centos/Node1/HCNet-Core/buckets"

HTTP\_PORT=11626 (node port)

PUBLIC\_HTTP\_PORT=false

RUN\_STANDALONE=false

NETWORK\_PASSPHRASE="HCNet Network"

NODE\_SEED="SCDJCF6EGT56U2NM2XE6JZESCNER46FOFIDPRNM6OKKH3V7MGVQUIIJ self" (secret seed for node1)

NODE\_IS\_VALIDATOR=true

PEER\_PORT=11625

DATABASE="postgresql://dbname=coredb1 user=core1 password=core1 host=localhost" (node1 postgres db details)

KNOWN\_PEERS=["127.0.0.1:11627"]

COMMANDS=["ll?level=debug"]

FAILURE\_SAFETY=0

UNSAFE\_QUORUM=true

[QUORUM\_SET]

THRESHOLD\_PERCENT=67

VALIDATORS=["$self","GDXXJ25WKWD3OHM4AF3RNFCP2NBWFCRDB34S2TE5FZTDNRLYSMA42REI"] (public key for node2)

[HISTORY.vs]

get="cp /tmp/Node1/HCNet-Core/history/vs/{0} {1}"

put="cp {0} /tmp/Node1/HCNet-Core/history/vs/{1}"

mkdir="mkdir -p /tmp/Node1/HCNet-Core/history/vs/{0}"

* Goto each node and run the follow command for node syncing operation.

go to each node /home/centos/node1/HCNet-Core/src

sudo ./HcNet-core --conf ../HCNet-Core.cfg --newdb

get root account seed save it

root account seed :

SAKUTHOI5AW3OXI7NGJ4TYCDVCZTI2F7WWDLKPJEJIS56YUGV36XQZM

sudo ./HcNet-core --conf ../HCNet-Core.cfg --newhist vs

sudo ./HcNet-core --conf ../HCNet-Core.cfg force-scp

sudo nohup ./HcNet-core --conf ../HCNet-Core.cfg &

Run this below command to know node sync status

sudo ./HcNet-core --conf ../HCNet-Core.cfg -c info

* Change node protocol version in each node

sudo ./HcNet-core --conf ../HCNet-Core.cfg -c 'upgrades?mode=set&upgradetime=2019-09- 12T12:45:00Z&protocolversion=10'