**Install vmWare tools**

<http://www.vmwareandme.com/2014/01/tutorial-how-to-install-vmware-tools-in.html#.V3Y0BlUrKM8>

1. Copy the setup file VMwareTools-9.9.3-2759765.tar.gz and extract it
2. Go to VMware tools folder & execute below command.   
   #sudo ./vmware-install.pl
3. Install in /usr/bin
4. VMware tools will be installed. You can verify the version of it by running below command in Ubuntu Terminal.   
   #vmware-toolbox
5. Restart VM

**Install mongodb on ubuntu 16.04**:

(https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/)

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927

echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list

sudo apt-get update

sudo apt-get install -y mongodb-org

sudo service mongod start

if it does not start, follow below steps

Install Upstart

sudo apt-get install upstart-sysv

Reboot your system

sudo service mongod start

stop mongo > sudo service mongod stop

restart mongo > sudo service mongod restart

stopping mongodb to start on ubuntu startup: echo manual | sudo tee /etc/init/mongodb.override

check status of mongodb

service mongod status

**move from console to gui ubuntu** : sudo lightdm

http://askubuntu.com/questions/141306/not-getting-gui-mode-on-ubuntu-12-04-desktop-edition-after-installation-on-vmwar

sudo nano /etc/crontab

@reboot root lightdm

**chrome**:

(http://askubuntu.com/questions/510056/how-to-install-google-chrome)

cd /tmp

wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb

sudo dpkg -i google-chrome-stable\_current\_amd64.deb

If you get errors do,

sudo apt-get install -f

and then again run

sudo dpkg -i google-chrome-stable\_current\_amd64.deb

Remove chrome

sudo apt-get purge google-chrome-stable

sudo apt-get autoremove

**Install bitcoin**

(http://www.n00bsonubuntu.net/content/install-bitcoin-core-multibit-wallet-ubuntu-14-10/

https://bitcoin.org/en/full-node#ubuntu-1410)

sudo apt-add-repository ppa:bitcoin/bitcoin -y

sudo apt update

sudo apt install bitcoin-qt -y

sudo apt-get install bitcoind

OR

sudo apt-get install bitcoin-qt bitcoind -y

Remove bitcoin

sudo apt-get remove bitcoind

sudo apt-get remove bitcoin-qt

remove the bitcoin directory

**install patched bitcoin**

Download > https://github.com/btcdrak/bitcoin/releases/download/v0.12.1-addrindex/bitcoin-0.12.1-linux64.tar.gz

from downloads directory: sudo tar -C /usr/local/src -xzf bitcoin-0.12.1-linux64.tar.gz

to run: (https://github.com/CounterpartyXCP/Documentation/blob/master/Installation/bitcoin\_core.md)

cd /usr/local/src

./bitcoin-\*/bin/bitcoind -help

cd ~/.bitcoin/

Add bitcoin.conf with below configuration:

rpcuser=bitcoinrpc

rpcpassword=capgemini100

server=1

txindex=1

addrindex=1

rpcthreads=1000

rpctimeout=300

minrelaytxfee=0.00005

limitfreerelay=0

sudo touch bitcoin.conf

sudo chmod 600 ~/.bitcoin/bitcoin.conf

sudo gedit bitcoin.conf

to start bitcoin core >

cd cd /usr/local/src

./bitcoin-\*/bin/bitcoin-qt

to start bitcoin demon >

./bitcoin-\*/bin/bitcoind

**Install counterparty-cli**

(<https://github.com/CounterpartyXCP/counterparty-cli>)

User /usr/local/src folder

$ sudo git clone https://github.com/CounterpartyXCP/counterparty-cli.git

$ cd counterparty-cli

$ sudo –H pip3 install -r requirements.txt

$ sudo python3 setup.py install

For running

Sudo counterparty-server –help

**Installing Bitsplit Server**

$ sudo git clone <https://github.com/tokenly/bitsplit-server.git>

$ cd bitsplit-server

$ sudo virtualenv venv

$ source venv/bin/activate

$ pip install -r requirements.txt

# Starting the Services

* Start the Daemon and process distributions:

cd /usr/local/src/bitsplit-server

./bitsplitd

* Start the API and allow end-users to make requests:

./api

**Install Pip:**

|  |  |
| --- | --- |
|  | sudo apt-get update  sudo apt-get upgrade |
|  | apt-get install python-pip |

Pip -Y

**Install OpenJDK and Eclipse**

**(**[**https://www.digitalocean.com/community/tutorials/how-to-install-java-on-ubuntu-with-apt-get**](https://www.digitalocean.com/community/tutorials/how-to-install-java-on-ubuntu-with-apt-get)**)**

[**http://linuxpitstop.com/install-eclipse-ide-on-ubuntu-linux-15-04/**](http://linuxpitstop.com/install-eclipse-ide-on-ubuntu-linux-15-04/)

sudo apt-get install default-jdk

Download Eclipse

<http://ftp.yz.yamagata-u.ac.jp/pub/eclipse//technology/epp/downloads/release/mars/2/eclipse-jee-mars-2-linux-gtk-x86_64.tar.gz>

sudo mv eclipse-jee-mars-2-linux-gtk-x86\_64.tar.gz /opt

cd /opt

sudo tar –xvf eclipse-jee-mars-2-linux-gtk-x86\_64.tar.gz

sudo gedit /usr/share/applications/eclipse.desktop

It will launch an empty file, copy and paste the followings into this empty file, as shown in following screenshot.

[Desktop Entry]  
  
Name=Eclipse  
  
Type=Application  
  
Exec=/opt/eclipse/eclipse  
  
Terminal=false  
  
Icon=/opt/eclipse/icon.xpm  
  
Comment=Integrated Development Environment  
  
NoDisplay=false  
  
Categories=Development;IDE;  
  
Name[en]=eclipse.desktop

Save the file, and now run following command on terminal to automatically install this desktop file on your ubuntu system.

sudo desktop-file-install /usr/share/applications/eclipse.desktop

As last step of the install, go to /usr/local/bin and create eclipse symlink, following two commands should take care of this:

cd /usr/local/bin

sudo ln -s /opt/eclipse/eclipse

In eclipse.ini

cd /opt/eclipse

sudo gedit eclipse.ini

add below in this file:

--launcher.GTK\_version

2

It's important to add these lines before --launcher.appendVmargs

-vm

/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

-vm should be placed before -vmargs in eclipse.ini file.

Kill eclipse

Jps – l

Sudo kill

sudo unzip /home/devchaud/PyDev5.0.0Dropin.zip -d /opt/eclipse/dropins/pydev

<http://www.vogella.com/tutorials/Python/article.html>

http://www.pydev.org/manual\_101\_eclipse.html

**Install Bigchaindb:**

http://rethinkdb.com/docs/install/ubuntu/

https://github.com/bigchaindb/bigchaindb/blob/master/docs/source/installing-server.md

<https://eye.raze.mx/install-bigchaindb-in-ubuntu/>

Install and run 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

Install Python 3.4+

Install Bigchaindb:

virtualenv -p python3.5 env

source env/bin/activate

pip install bigchaindb

bigchaindb -y configure

bigchaindb start

deactivate

sudo pip install --index-url=http://pypi.python.org/simple --trusted-host pypi.python.org Flask

sudo pip install --index-url=http://pypi.python.org/simple --trusted-host pypi.python.org requests

sudo pip install --index-url=http://pypi.python.org/simple --trusted-host pypi.python.org werkzeug

sudo service apache2 restart

pip install requests==1.9.9

**Install docker CE: (**https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/#install-using-the-repository**)**

**Find architecture:**

sudo dpkg --print-architecture

1. Update the apt package index:

$ sudo apt-get update

1. Install packages to allow apt to use a repository over HTTPS:

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

1. Add Docker’s official GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Verify that the key fingerprint is 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88.

$ sudo apt-key fingerprint 0EBFCD88

pub 4096R/0EBFCD88 2017-02-22

Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

uid Docker Release (CE deb) <docker@docker.com>

sub 4096R/F273FCD8 2017-02-22

1. Use the following command to set up the stable repository.

**amd64**:

$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

#### INSTALL DOCKER CE

1. Update the apt package index.

$ sudo apt-get update

1. Install the latest version of Docker CE, or go to the next step to install a specific version. Any existing installation of Docker is replaced.

$ sudo apt-get install docker-ce

1. Verify that Docker CE is installed correctly by running the hello-world image.

$ sudo docker run hello-world

**If u get the error**

**Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running? Do below:**

sudo groupadd docker

sudo service docker restart docker

**Enable docker at startup**

sudo systemctl enable docker

**Installing Docker Compose**

(https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-16-04)

We'll check the [current release](https://github.com/docker/compose/releases) and if necessary, update it in the command below:

[**https://github.com/docker/compose/releases**](https://github.com/docker/compose/releases)

sudo curl -o /usr/local/bin/docker-compose -L "https://github.com/docker/compose/releases/download/1.15.0/docker-compose-$(uname -s)-$(uname -m)"

Next we'll set the permissions:

sudo chmod +x /usr/local/bin/docker-compose

Then we'll verify that the installation was successful by checking the version:

docker-compose -v

This will print out the version we installed:

Output

docker-compose version 1.15.0, build e12f3b9

**Installing Node and npm:**

(<https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-16-04>)

sudo apt-get update

sudo apt-get install build-essential libssl-dev

use latest version from https://github.com/creationix/nvm

curl -sL [https://raw.githubusercontent.com/creationix/nvm/v0.33.2/install.sh -o install\_nvm.sh](https://raw.githubusercontent.com/creationix/nvm/v0.33.2/install.sh%20-o%20install_nvm.sh)

Inspect the installation script with nano:

nano install\_nvm.sh

Run the script with bash:

bash install\_nvm.sh

source ~/.profile

nvm --version

nvm ls-remote

nvm install 6.11.2

node –v

npm –version

**Install git**

(<https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-16-04>)

sudo apt-get update

sudo apt-get install build-essential libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext unzip

Get the latest version of git from https://github.com/git/git

wget [https://github.com/git/git/archive/v2.14.1.zip -O git.zip](https://github.com/git/git/archive/v2.14.1.zip%20-O%20git.zip)

unzip git.zip

cd git-\*

make prefix=/usr/local all

sudo make prefix=/usr/local install

cd ~

git clone https://github.com/git/git.git

sudo git –version

**Install Python 2.7**

sudo apt-get install python2.7

**Install Hyperledger Composer**

<https://hyperledger.github.io/composer/installing/development-tools.html>

**Install Tomcat:**

https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04

First, create a new tomcat group:

sudo groupadd tomcat

Next, create a new tomcat user. We'll make this user a member of the tomcat group, with a home directory of /opt/tomcat (where we will install Tomcat), and with a shell of /bin/false (so nobody can log into the account):

sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

Install latest tomcat (http://tomcat.apache.org/download-80.cgi)

cd /tmp

curl -O http://redrockdigimark.com/apachemirror/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz

sudo mkdir /opt/tomcat

sudo tar xzvf apache-tomcat-8\*tar.gz -C /opt/tomcat --strip-components=1

Update permissions

cd /opt/tomcat

sudo chgrp -R tomcat /opt/tomcat

sudo chmod -R g+r conf

sudo chmod g+x conf

Make the tomcat user the owner of the webapps, work, temp, and logs directories:

sudo chown -R tomcat webapps/ work/ temp/ logs/

Find java location

sudo update-java-alternatives –l

With this piece of information, we can create the systemd service file. Open a file called tomcat.service in the /etc/systemd/system directory by typing:

sudo nano /etc/systemd/system/tomcat.service

Paste the following contents into your service file. Modify the value of JAVA\_HOME if necessary to match the value you found on your system. You may also want to modify the memory allocation settings that are specified in CATALINA\_OPTS:

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat

Group=tomcat

UMask=0007

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

Reload daemon

sudo systemctl daemon-reload

Start the Tomcat service by typing:

sudo systemctl start tomcat

Double check that it started without errors by typing:

sudo systemctl status tomcat

Adjust firewall

sudo ufw allow 8080

Enable boot service

sudo systemctl enable tomcat

Configure Tomcat Web Management Interface

sudo nano /opt/tomcat/conf/tomcat-users.xml

Add user

<user username="admin" password="password" roles="manager-gui,admin-gui"/>

<tomcat-users . . .>

<user username="admin" password="password" roles="manager-gui,admin-gui"/>

</tomcat-users>

For the Manager app, type:

sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml

Inside, comment out the IP address restriction to allow connections from anywhere. Alternatively, if you would like to allow access only to connections coming from your own IP address, you can add your public IP address to the list:

context.xml files for Tomcat webapps

<Context antiResourceLocking="false" privileged="true" >

</Context>

Save and close the files when you are finished.

To put our changes into effect, restart the Tomcat service:

sudo systemctl restart tomcat