[Initial Ubuntu Server Setup for Beginners](https://www.tecmint.com/initial-ubuntu-server-setup-guide/)

[50 Linux Commands](https://www.thegeekstuff.com/2010/11/50-linux-commands/)

[50 Unix Linux Sysadmin Tutorials](https://www.thegeekstuff.com/2010/12/50-unix-linux-sysadmin-tutorials/)

**Install NodeJS**

Install nodejs using NVM

First to install NVM using the following command:

$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash

Need to restart the terminal to use NVM

$ nvm --version

Will display NVM vesion if installed correctly

Now need to install node js using command like below

$ nvm install node

Will install latest nodejs veriosn

If want to install a particular version say v9.2.0

Command will be live this:

$ nvm install 9.2.0

$nvm ls

Will display all installed nodejs version

To switch between nodejs version

$nvm use 9.2.0

To run nodejs use

$ node

To check nodejs version

$ node -v

Ref :

1. <https://linuxize.com/post/how-to-install-node-js-on-ubuntu-18.04/>

**Install python**

Python 3 remains preinstalled in ubuntu 18.04 server.  
If not install use

$ sudo apt install python3

Need to install pip3

$ sudo apt install python3-pip

Check if pip installed by using

$ pip3 -V

To install package use

$ pip3 install <package name>

Need to install virtual enviroment. Use

$ pip3 install virtualenv

Create a virtual env at you project dir

$ cd /your-product-dir

Then

$ virtualenv venv

$ ls

Will display a venv folder

Now need to activate our new virtual env by using

$ source venv/bin/activate

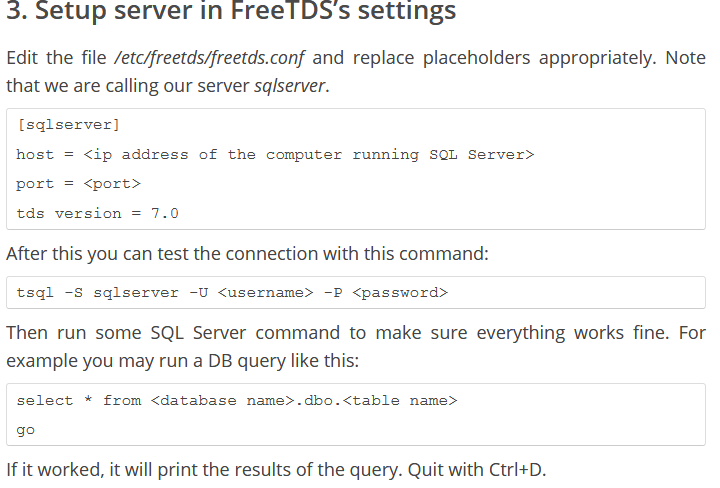
Now we are ready to use python for this application

$ deactivate

Ref:

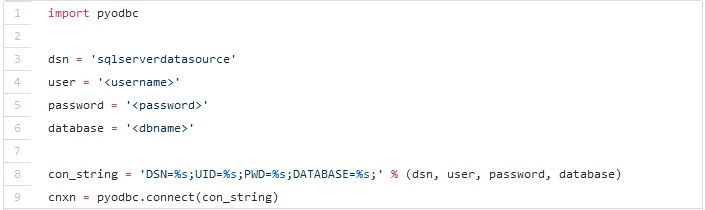
1. <https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-programming-environment-on-ubuntu-18-04-quickstart>
2. <https://linuxhostsupport.com/blog/how-to-install-virtual-environment-on-ubuntu-16-04/>
3. <https://www.guru99.com/how-to-install-java-on-ubuntu.html>

**SQL Server Connection**

sudo apt-get install unixodbc unixodbc-dev freetds-dev tdsodbc

(venv) $ pip3 install pyodbc

**Python Code:**



Ref. <https://tryolabs.com/blog/2012/06/25/connecting-sql-server-database-python-under-ubuntu/>

Data source setup from pyodbc

**Step 1:**

**$ sudo nano /etc/odbcinst.ini**

And paste the below content in the file

**[FreeTDS]**

**Description = TDS driver (Sybase/MS SQL)**

**# Some installations may differ in the paths**

**#Driver = /usr/lib/odbc/libtdsodbc.so**

**#Setup = /usr/lib/odbc/libtdsS.so**

**Driver = /usr/lib/x86\_64-linux-gnu/odbc/libtdsodbc.so**

**Setup = /usr/lib/x86\_64-linux-gnu/odbc/libtdsS.so**

**CPTimeout =**

**CPReuse =**

**FileUsage = 1**

**Step-2**

**$ sudo nano /etc/odbc.ini**

And paste the below content as ter

**[datasource1]**

**Driver = FreeTDS**

**Description = ODBC connection via FreeTDS**

**Trace = No**

**Servername = Myserver1**

**Database = dbname1**

**[datasource2]**

**Driver = FreeTDS**

**Description = ODBC connection via FreeTDS**

**Trace = No**

**Servername = Myserver2**

**Database = dbname2**

**Step-3**

**$ sudo nano /etc/freetds/freetds.conf**

**[Myserver1]**

**host = <you-host-name-or-ip>**

**port = <db running of port>**

**tds version = 7.0**

**[Myserver2]**

**host = <you-host-name-or-ip>**

**port = <db running of port>**

**tds version = 7.0**

Now in python code set

**DSN=datasource1**

**DSN=datasource2**

Code example and mentioned previously.

* **Don’t forget to pass autocommit=True parameter in pyodbc.connect() function.**

**Install JAVA & set JAVA\_HOME env variable**

$ java

Will display all java pacakges available if Java is not installed.

$ sudo apt install <java-package-name-from-the-list>

Or download zip from web

Check if installation os okay

$ java -version

Get java installation path the following command

$ update-alternatives --config java

Open env file using

$ sudo nano /etc/environment

Add the below line

JAVA\_HOME= “/java/installed/path”

Save and exit

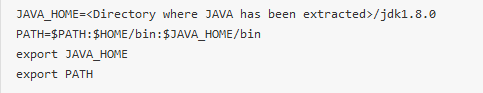
To refresh env variables use

$ source /etc/environment

JAVA\_HOME is set now. To check

$ echo $JAVA\_HOME

If zip downloaded. Unzip it somewhere and set the path in /etc/environment file and in /etc/profile file as well. As the bellow lines at the end of /etc/profile file



Add manullay installed java (installed from tar.gz) in to update-alternatives list by using the following method

>> downlaod the tar.gz and extract it into /usr/lib/jvm

Then add this alternative into update-alternaives list like this :

Say your extractred java dir name is jdk-8-oracle and is extracted into /usr/lib/jvm

$ sudo udpate-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-8-oracle/bin/java 10

$ sudo udpate-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-8-oracle/bin/javac 10

The number 10 at the end is priority, you can set any lesser the number more priority it has.

Do the same for all java binaries, below are the list :

appletviewer extcheck idlj jar jarsigner javac javadoc javah javap jconsole jdb jhat jinfo jmap jps jrunscript jsadebugd jstack jstat jstatd native2ascii rmic schemagen serialver wsgen wsimport xjc

Then run the below command

$ sudo update-alternatives --config java

It will list all the java alternatives installed in the system.

**ElasticSearch**

**Using Debian Package:**

Download elasticsearch specific verion .deb file from elasticserch official site and install using

$ sudo dpkg -i elasiticseasrch-6.2.1.deb

Elasticsearch can’t be run as root user.

Need to follow the below steps:

Steps:

1. Change owership of all ES related files to root using example cmd below.

$ sudo chown elasticsearch:elasticsearch -R /usr/share/elasticsearch

$ sudo chown elasticsearch:elasticsearch -R /var/log/elasticsearch

$ sudo chown elasticsearch:elasticsearch -R /var/lib/elasticsearch

$ sudo chown elasticsearch:elasticsearch -R /etc/default/elasticsearch

$ sudo chown elasticsearch:elasticsearch -R /etc/elasticsearch

1. Open /etc/default/elasticsearch file and do the following things
   1. JAVA\_HOME=your/java/home/path
   2. add the following entries at the end
      1. START\_DAEMON=true
      2. ES\_USER=elasticsearch
      3. ES\_GROUP=elasticsearch
2. Now enable elasticsearch service and start
   1. $ sudo systemctl enable elasticsearch
   2. $ sudo systemctl start elasticsearch
   3. $ sudo systemctl status elasticsearch
3. Test elasticsearch by using curl. Say your host ip is 192.168.5.194 and ES running on port 9200
   1. $ curl -X GET ‘192.168.5.194:9200’

DONE!!

Important link : <https://stackoverflow.com/a/48390311/1445978>

Important elasticsearch.yml entry. This is required for single node production. Otherwise you cannot set non-loopback address at network.host configuration.

discovery.type: single-node

Increase virtual memory for production : (does not work if run from systemd service)

sysctl -w vm.max\_map\_count=262144

Impact of the above command is not permanent. To make it permanent update in /etc/sysctl.conf

*$ sudo nano /etc/sysctl.conf*

And add the below setting in this file and reboot.

*vm.max\_map\_count=262144*

<https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html>

When runing elasticsearch as systemd the above mentioned limit will not work.

Need to specify the limit on hte /etc/systemd/system/yourservicename.service file

Below is an example snapshot.

Ref :

<https://unix.stackexchange.com/a/345596/107503>

<https://www.elastic.co/guide/en/elasticsearch/reference/current/setting-system-settings.html#limits.conf>

**[Unit]**

**Description=Service to run Elasticsearch**

**[Service]**

**LimitMEMLOCK=infinity**

**LimitNOFILE=262144**

**User=myuser**

**Group=myuser**

**ExecStart=/opt/es/elasticsearch-6.5.1/start.sh**

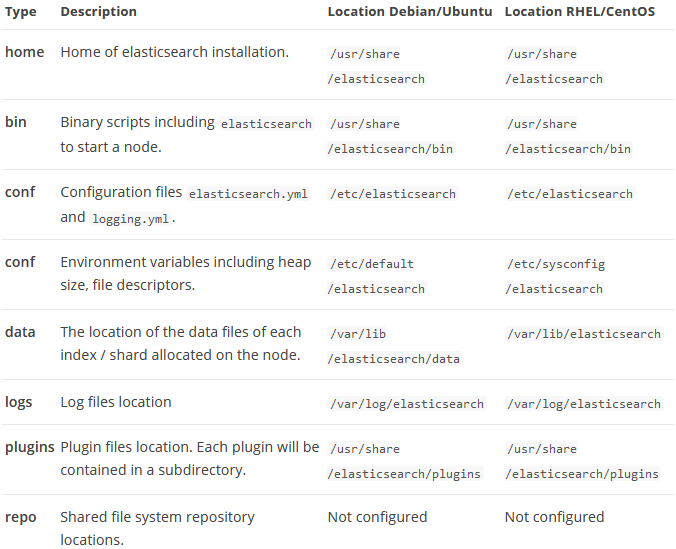
**Type=oneshot**

**RemainAfterExit=yes**

**[Install]**

**WantedBy=multi-user.target**

Debian directory layout for Elasticsearch:



For installation guide follow the digitalocean link below.

Ref.

1. <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-16-04>
2. <https://www.elastic.co/guide/en/elasticsearch/reference/2.3/setup-dir-layout.html>
3. <https://stackoverflow.com/a/48390311/1445978>
4. <https://www.elastic.co/guide/en/elasticsearch/reference/master/bootstrap-checks.html>
5. <https://www.elastic.co/guide/en/elasticsearch/reference/6.2/zip-targz.html#setup-installation-daemon>

ulimit -m unlimited = LimitRSS=infinity

**Using GZIP package**

Download zip package and unzip. Change ownership of file to non-root user.

ES Service : <https://www.elastic.co/guide/en/elasticsearch/reference/master/zip-targz.html#setup-installation-daemon>

Start elasticsearch as daemon

$ ./bin/elasticsearch -d -p pid

Stop elasticsearch

$ kill `cat pid`

Create a systemd service and **enable** it to run elasticsearch on startup before that create a bash script with above mentioned start command and provide execute permission then link that bash script with the systemd service. Follow this link

<https://askubuntu.com/a/719157/367545>

**Monitoring Linux**

1. [Performance Monitor](https://www.tecmint.com/command-line-tools-to-monitor-linux-performance/)
2. [Process Management](https://www.howtogeek.com/107217/how-to-manage-processes-from-the-linux-terminal-10-commands-you-need-to-know/)
3. [Service Management](https://www.techrepublic.com/article/how-to-start-stop-and-restart-services-in-linux/)
4. [Service Management](https://www.linux.com/learn/managing-services-linux-systemd)

**Apt**

To list all installed packages

$ sudo apt list --installed | less

Look a specific package say apache

$ sudo apt list --installed | grep -i apache

Another alternative to list all installed packages

$ sudo dpkg -l

Uninstall a package and remove all related files

$ sudo apt purge <package-name>

Ref. <https://www.rosehosting.com/blog/list-all-installed-packages-with-apt-on-ubuntu/>

**User management**

Add user

$ sudo useradd <username>

$ sudo useradd -g <groupname> <username>

Assign password

$ sudo passwd <username>

Add more secondary groups

$ sudo usermod -a -G <group1>,<group2> <username>

-g means primary group. Can only be one.

-G means secondary groups

**Service**

Create nodejs service

<https://hackernoon.com/making-node-js-service-always-alive-on-ubuntu-server-e20c9c0808e4>

NodeJs Service:

**[Unit]**

**Description=Node.js SSPL ES Search Service**

**[Service]**

**PIDFile=/tmp/sspl\_search.pid**

**User=user**

**Group=usergroup**

**Restart=on-error**

**KillSignal=SIGQUIT**

**WorkingDirectory=/appdata01/ftpdir/sspl\_search/sspl\_search\_v7.e6.n11/**

**ExecStart=/home/ubuntu/.nvm/versions/node/v11.2.0/bin/node /appdata01/ftpdir/sspl\_search/sspl\_search\_v7.e6.n11/app.js**

**[Install]**

**WantedBy=multi-user.target**

Custom service creation:

1. <https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6>
2. <https://dzone.com/articles/run-your-java-application-as-a-service-on-ubuntu>

Python Service:

**[Unit]**

**Description=Python SSPL Lab Micro-Service**

**[Service]**

**PIDFile=/tmp/sspl\_lab\_microservie.pid**

**User=ubuntu**

**Group=ubuntu**

**Restart=on-error**

**KillSignal=SIGQUIT**

**WorkingDirectory=/appdata01/ftpdir/service.labs.sastasundar.com/**

**ExecStart=/appdata01/ftpdir/service.labs.sastasundar.com/venv/bin/python3 /appdata01/ftpdir/service.labs.sastasundar.com/main.py**

**[Install]**

**WantedBy=multi-user.target**

**FTP Server (VSFTPD) Configuration**

We will use VSFTPD

Install

$ suod apt install vsftpd

Config file exists in /etc/vsftpd.conf

First keep a backup of the original file by

$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig

Do the following changes:

1. listen=NO
2. listen\_ipv6=YES
3. anonymous\_enable=NO
4. local\_enable=YES
5. **write\_enable=YES**
6. connect\_from\_port\_20=YES
7. chroot\_local\_user=YES
8. chroot\_list\_enable=YES
9. chroot\_list\_file=/etc/vsftpd.chroot\_list
10. **pam\_service\_name=ftp**
11. ssl\_enable=YES
12. **userlist\_enable=YES**
13. **userlist\_deny=NO**

Now Create user say devasish and set /srv/ftp/devasish dir as home

Create home dir named devasish\_home

$ sudo mkdir /srv/ftp/devasish\_home

Create the user

$ sudo useradd -d /srv/ftp/devasish\_home

Set password

$ sudo passwd devasish

Change write permission of the home dir

$ sudo chmod a-w /srv/ftp/devasish\_home

It will prevent accidental delete of the entire ftp dir. It’s sub dir will be writable.

Now create 2 files in /etc follow below instructions

$ sudo vim /etc/vsftpd.user\_list

Add entry just devasish

If another user created it also needs to be added in this file. **One user at one line**

$ sudo vim /etc/vsftpd.chroot\_list

Add entry of the username devasish

Save and exit

Now restart vsftpd service

$ sudo systemctl restart vsftpd

Now the fpt server can be connected by ftp client like **filezilla**

Set passive port ranges in vsftpd.conf file

pasv\_min\_port=40000

pasv\_max\_port=50000

Firewall setting

$ sudo ufw allow 20/tcp

$ sudo ufw allow 21/tcp

$ sudo ufw allow 990/tcp

$ sudo ufw allow 40000:50000/tcp

$ sudo ufw status

Ref.

1. [IMPORTANT]<https://www.thegeekdiary.com/error-530-permission-denied-when-user-logs-in-to-vsftpd-server-via-ftp/>
2. [IMPORTANT]<https://askubuntu.com/a/413694/367545>
3. <https://help.ubuntu.com/lts/serverguide/ftp-server.html.en>
4. <https://www.emiprotechnologies.com/technical_notes/odoo-technical-notes-59/post/install-and-configure-vsftpd-475>
5. [IMPORTANT FOR FIREWALL SETTING] <https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-a-user-s-directory-on-ubuntu-16-04>

Mongo DB

Follow the steps mentioned here:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/#install-mongodb-community-edition-using-deb-packages>

**Step 1**

$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4

**Step 2**

[Ubuntu 16]

$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list

[Ubuntu 18]

echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list

**Step 3**

sudo apt-get update

**Step 4**

$ sudo apt-get install -y mongodb-org=4.0.4 mongodb-org-server=4.0.4 mongodb-org-shell=4.0.4 mongodb-org-mongos=4.0.4 mongodb-org-tools=4.0.4

Misc

List all systemd services

$ systemctl list-units