* *Installing latest Chrome Version*

1. *Download latest version*
2. *cd /home/developers/Downloads*
3. *wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb*
4. *Install with following command*
5. *sudo apt-get update && sudo apt install -y ./google-chrome-stable\_current\_amd64.deb*

* *To Downgrade Chrome to 88 Version*

*Download the deb file of chrome in /home/developers/Downloads and install that deb file by apt*

1. *cd /home/developers/Downloads*
2. *wget* [*http://dl.google.com/linux/chrome/deb/pool/main/g/google-chrome-stable/g*](http://dl.google.com/linux/chrome/deb/pool/main/g/google-chrome-stable/g)*ogle-chrome-stable\_88.0.4324.96-1\_amd64.deb*
3. *sudo apt-get install -y ./google-chrome-stable\_88.0.4324.96-1\_amd64.deb*

* *Installation of Eclipse Oxygen 4.7.0*

*(prerequisites: java need to be installed on the machine)*

*0. For windows download from*

[*https://www.eclipse.org/downloads/packages/release/oxygen/r*](https://www.eclipse.org/downloads/packages/release/oxygen/r)

1. *For linux Goto*

*https://archive.eclipse.org/technology/epp/downloads/release/oxygen/R/*

*& Download*

1. *OR Pull from backend centralized machine*

*scp -r -P 513 root@125.99.46.47:/root/Avinash/eclipse-jee-oxygen-R-linux-gtk-x86\_64.tar.gz .*

1. *To get app icon on desktop*

*3.1 Create file*

*vim /home/developers/.local/share/applications/eclipse.desktop*

*3.2 Add following lines in file*

*[Desktop Entry]*

*Type=Application*

*Terminal=false*

*Encoding=UTF-8*

*Version=1.1*

*Name=Eclipse-Oxygen 4.7.0*

*Exec=env GTK\_IM\_MODULE=ibus /home/developers/eclipse/eclipse*

*Categories=Development;IDE;*

*Icon=/home/developers/eclipse/icon.xpm*

*3.3 Give permissions to desktop file*

*chmod +x /home/developers/Desktop/eclipse.desktop*

* *Installation of Eclipse Photon 4.8.0*

1. *For linux Goto*

*https://archive.eclipse.org/technology/epp/downloads/release/photon/R/eclipse-jee-photon-R-linux-gtk-x86\_64.tar.gz*

*Download*

1. *Extract file*

*cd /home/developers/Downloads*

*tar -xvf eclipse-jee-photon-R-linux-gtk-x86\_64.tar.gz*

1. *Move extracted file*

*mv eclipse /home/developers*

1. *To create Desktop Icon*
2. *vim /home/developers/.local/share/applications/eclipse.desktop*
3. *Add following entries*

*[Desktop Entry]*

*Name=Eclipse JEE Photon*

*Type=Application*

*Exec=/home/developers/eclipse-jee-photon/eclipse/eclipse*

*Terminal=false*

*Icon=/home/developers/eclipse-jee-photon/eclipse/icon.xpm*

*Comment=Integrated Development Environment*

*NoDisplay=false*

*Categories=Development;IDE;*

*Name[en]=Eclipse*

1. *Check icon in app menu*

* *Wifi module issue*

*sudo apt-get install build-essential git*

*git clone https://github.com/rtlwifi-linux/rtlwifi\_new*

*cd rtlwifi\_new*

*h*

*sudo make install*

*sudo modprobe -r rtl8723be*

*echo "options rtl8723be ant\_sel=2" | sudo tee /etc/modprobe.d/rtl8723be.conf*

*sudo modprobe rtl8723be*

*sudo reboot*

**OR**

git clone https://github.com/smlinux/rtl8723de.git -b current

dkms add ./rtl8723de

dkms install rtl8723de/5.1.1.8\_21285.20171026\_COEX20170111-1414

depmod -a

modprobe -r rtl8723de

reboot

* To uninstall follow below commands

1. rmmod -f 8723de
2. dkms uninstall rtl8723de/5.1.1.8\_21285.20171026\_COEX20170111-1414
3. dkms remove rtl8723de/5.1.1.8\_21285.20171026\_COEX20170111-1414 --all
4. depmod-a
5. reboot

* *Remove admin1 authentication for wifi*

sed -i /usr/share/polkit-1/actions/org.freedesktop.NetworkManager.policy -e '898s!auth\_admin\_keep!yes!';echo $?

* *Machine Creation by ansible*

1. *Set hostname by command*

hostnamectl set-hostname localhost

1. *Add entry in /etc/hosts file*

127.0.0.1 localhost

1. *Goto server 124 &*

*vim /etc/ansible/hosts*

1. *Add your machine’s ip to above file*

*Add ip at crm block*

1. *Ping that machine by ansible command*
2. ansible crm -m ping

1. ansible -m ping all

1. *Go to*

1. *cd /root/Avinash/ansible*
2. *run ansible playbook*

ansible-playbook machine\_creation.yml

1. It will start installation

* *Unban ip from 124*

*fail2ban-client set sshd unbanip <ip>*

* *Installing Python*

1. *apt update*
2. *sudo apt-get install build-essential checkinstall*
3. *sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev \*

*tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev \ zlib1g-dev*

1. *Download package from*

[*https://www.python.org/ftp/python/*](https://www.python.org/ftp/python/)

1. *Extract downloaded file*

*5.1 tar -xvf <file\_name>.tar.xz*

*OR*

*5.2 tar -xvzf <file\_name>.tar.gz*

1. *Go into extracted python dir*

*cd <dir\_name>*

1. *RUN*

*sudo ./configure --enable-optimizations*

1. *RUN*
2. sudo ./configure --enable-optimizations
3. *sudo make altinstall*

*(it will start installation)*

1. *Check python version*

python --version

1. *In case of error*

*9.1 python command not found copy that binary to*

/usr/bin/python

1. *Create link for python version*

cd /usr/bin/python

ls -l | grep python

ln -s python<version> python

1. *Check python version*

python --version

1. *Installing python 2.7.12*

*sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev libffi-dev liblzma-dev python-openssl git*

* *Installing Python 3.6 on ubuntu16*

[*https://www.rosehosting.com/blog/how-to-install-python-3-6-on-ubuntu-16-04/*](https://www.rosehosting.com/blog/how-to-install-python-3-6-on-ubuntu-16-04/)

* *Installing Python 3.8*

*https://tech.serhatteker.com/post/2019-12/how-to-install-python38-on-ubuntu/*

* *Installing Or Downgrade java version*

1. *Check java versions available*

*java -version*

1. *Install java 8*

*sudo apt-get update*

*sudo apt-get install openjdk-8-jdk*

1. *Switch to the required java version*

*sudo update-alternatives --config java*

*3.1 You will see all available versions of java on your*

*machine and a prompt*

*3.2 Type selection no of required java version & press enter*

*3.3 check java version*

*java -version*

* *Installing java 11*

*sudo add-apt-repository ppa:openjdk-r/ppa*

*sudo apt-get update*

*sudo apt install openjdk-11-jdk -y*

* *Installing java on windows*

1. *Download java from below link*

[*https://www.oracle.com/java/technologies/downloads/#java8-windows*](https://www.oracle.com/java/technologies/downloads/#java8-windows)

1. *Go to Downloads folder and click on run as administrator*
2. *Install java by clicking on “next”*
3. *Check on command line with following command*

*java -version*

* *Creating users for jfrog*

1. *Goto maven remote repository*

[*http://10.13.10.181:8081/artifactory/webapp*](http://10.13.10.181:8081/artifactory/webapp)

1. *Login using admin username and password*

***Username****: admin*

***Password****: \*\*\*\*\*\*\*\**

1. *Goto admin > security > user*

*3.1 Create user with username (provided) & with*

*password (nciportal)*

*3.2 Notification will appear of successful user creation*

*3.3 Logout from admin user account*

1. *Now login with created user’s credentials*

*4.1 click on username at right top corner*

*4.2 see encrypted password*

*4.3 copy encrypted password & save*

*4.4 logout from user account*

1. *Add username,password and encrypted password in*

*“Backend credentials for JFROG” sheet*

* *Permanent DNS Entry*

1. *Add domain and ip entries to /etc/resolvconf/resolv.conf.d/head file*
2. *systemctl restart resolvconf.service*

* *Installing Nginx 1.21.4*

1. *vim /etc/apt/sources.list.d/nginx.list*

*deb [arch=amd64] http://nginx.org/packages/mainline/ubuntu/ bionic nginx*

*deb-src http://nginx.org/packages/mainline/ubuntu/ bionic nginx*

1. *apt-key adv --keyserver keyserver.ubuntu.com --recv-keys ABF5BD827BD9BF62*
2. *apt update*

1. *apt-cache policy nginx*
2. *apt-get install nginx=1.21.4-1~bionic -y*

1. *nginx -v*

* *Installing conda*

1. *cd /home/developers*
2. *curl -O* [*https://repo.anaconda.com/archive/Anaconda2-4.4.0-Linux-x86\_64.sh*](https://repo.anaconda.com/archive/Anaconda2-4.4.0-Linux-x86_64.sh)
3. *Switch to developers user & run*

*bash* [*Anaconda2-4.4.0-Linux-x86\_64.sh*](https://repo.anaconda.com/archive/Anaconda2-4.4.0-Linux-x86_64.sh)

1. *Tap enter, check installation path & tap enter*
2. *source ~/.bashrc*
3. *To activate conda environments*
4. *conda info --envs*
5. *source <path to envs>*

*OR*

1. *source /home/developers/anaconda3/bin/activate*

*OR*

1. *Add in ~/.bashrc file*

*export PATH="$HOME/archiconda3/bin:$PATH"*

1. *To check conda version*

*conda info*

*conda --version*

***OR***

* ***Installation for arch***

1. wget <https://github.com/Archiconda/build-tools/releases/download/0.2.3/Archiconda3-0.2.3-Linux-aarch64.sh>
2. sudo sh Archiconda3-0.2.3-Linux-aarch64.sh

* *Unistalling conda*

1. *cd /home/developers*
2. *ls -al | grep conda*
3. *rm -rf <all\_conda\_files\_&\_dir>*
4. *Remove all previous conda entries from ~./bashrc file*
5. *vim /home/developers/.bashrc*
6. *source /home/developers/.bashrc*

*2. Check conda version and python version it is using*

1. *conda info*

* *Upgrading conda version*

1. *conda install conda=4.10.3*

*(enter required version)*

*Ref: https://libraries.io/conda/conda/4.11.0*

*https://anaconda.org/*

* *To Install Mosquitto Broker on Ubuntu 18.04*

1. *Prerequisites*

*Open port TCP:1883 on firewall*

1. *Install Mosquitto Broker*

*sudo apt-get update*

*sudo apt-get install mosquitto -y*

*2.1 Install the Clients and Test*

*sudo apt-get install mosquitto-clients -y*

*2.2 Secure with a Password*

*Create file /etc/mosquitto/passwd*

*Write Entry developers:nciportal*

*sudo mosquitto\_passwd -c /etc/mosquitto/passwd mqtt\_user\_name \*

*Password: mqtt\_password*

*2.3 To check MQTT version*

*mosquitto -v*

*2.3 Create a configuration file for Mosquitto pointing to the password file we have just created.*

*sudo vim /etc/mosquitto/conf.d/default.conf*

*allow\_anonymous false*

*password\_file /etc/mosquitto/passwd*

*2.4 Check mosquitto service*

*sudo systemctl restart mosquitto*

*sudo systemctl status mosquitto*

* *To remove Mosquitto*

1. *sudo dpkg --remove --force-remove-reinstreq mosquitto*
2. *sudo dpkg --purge mosquitto*
3. *sudo rm /etc/mosquitto*

* *Installing lite-server*

1. *cd /usr/local/share/packages/node-v10.16.3-linux-x64/lib/node\_modules/*
2. *npm i lite-server -g*

* *Installing HDF5*

1. *conda search <package\_name>*
2. *conda install -c anaconda hdf5=1.10.1 -y*

*OR*

*conda install -c conda-forge hdf5=1.10.1*

1. *To check HDF5 version*

*conda list | grep hdf5*

* *Installing spyder*

1. *conda search spyder*
2. *conda install spyder*

*OR*

*conda install --channel=conda-forge spyder=<version>*

*OR*

*conda install spyder=3*

1. *To see the conda environment (spyder)*

*conda list | grep spyder*

* *Creating desktop icon for spyder*

1. *vim /home/developers/.local/share/applications/spyder.desktop*
2. *[Desktop Entry]*

*Version=1.0*

*Type=Application*

*Name=Spyder 3*

*GenericName=Spyder 3*

*Comment=The Scientific Python Development Environment*

*TryExec=/home/developers/anaconda3/bin/spyder*

*Exec=/home/developers/anaconda3/bin/spyder %F*

*Categories=Development;Science;IDE;Qt;*

*Icon=/home/developers/anaconda3/share/icons/spyder3.png*

*Terminal=false*

*StartupNotify=true*

*MimeType=text/x-python;*

* *Changing python version for spyder*

1. *conda create -n py36 python=3.6 #for version 3.6 activate py36*
2. *conda activate py36*
3. *conda init –all (to init all shells)*

*OR*

1. *Open spyder application*
2. *Go to > tools > preferences > python interpreter*
3. *Select python file from /usr/bin*
4. *Apply file and ok*
5. *Restart spider*
6. *If an error occurs run kernel command prompted*

* *Installing zlib*

*https://www.systutorials.com/how-to-install-the-zlib-library-in-ubuntu/*

* *Installing sqlite3*

1. *sudo apt-get update*
2. *sudo apt-get install sqlite3*
3. *sqlite3 --version*

* *Installing sqlite database browser*

1. *Goto Ubuntu Softwares*
2. *Search sqlitebrowser*
3. *Enter admin password*
4. *Click on install*

* *Installing paho-mqtt*

1. [*https://pypi.org/project/paho-mqtt/*](https://pypi.org/project/paho-mqtt/)

* *Installing testNG in eclipse*

1. *cd /home/developers/Downloads*
2. *Download testNG from*

[*https://github.com/cbeust/testng-eclipse*](https://github.com/cbeust/testng-eclipse)

1. *Extract file*
2. *Goto eclipse*
3. *Inside eclipse > help > install new software >*
4. *Click on add*
5. *Click on local and select extracted file*
6. *Install (takes a while)*

* *Adding Users to gogs*

1. *Go to website by below address*
   1. *10.13.10.20:3000*
2. *Sign in as admin*
3. *Username: admin1*
4. *Pass: (default)*

1. *Go to > admin panel > users > create new user*
2. *Create new user by adding name.sirname*
3. *Add official email-id*
4. *Add password as “ncpl”*
5. *Mark on send registration notification to user*
6. *Click on create new user*
7. *Check by login with credentials of newly created user*

* *Installing mkdocs on ubuntu*

*Ref:*

[*https://mpilking.github.io/documentation/installing-mkdocs/*](https://mpilking.github.io/documentation/installing-mkdocs/)

* *Installing draw.io*

1. *snap install drawio*
2. *sudo snap install drawio*

*Ref:*

*https://snapcraft.io/install/drawio/ubuntu*

* *Installing VScode*

*Ref:*

*https://linuxize.com/post/how-to-install-visual-studio-code-on-ubuntu-18-04/*

* *Installing mocha*

*1.sudo apt-get update*

*2.sudo apt-get install mocha*

* *Rescue mode*

*Ref:* [*https://ostechnix.com/how-to-boot-into-rescue-mode-or-emergency-mode-in-ubuntu-18-04/*](https://ostechnix.com/how-to-boot-into-rescue-mode-or-emergency-mode-in-ubuntu-18-04/)

* *Installing elasticsearch*

*Ref:*

[*https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-18-04*](https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-18-04)

* *Installing jenkins*

1. *Download deb file from*

[*https://pkg.jenkins.io/debian-stable/direct/*](https://pkg.jenkins.io/debian-stable/direct/)

1. *Make sure java is installed*

*java -version*

1. *After downloading, Install with following command*

*dpkg -i jenkins\_2.263.1\_all.deb*

* *Installing android-studio*

1. *Download android studio from link*
2. *cd /home/developers*
3. *Download android studio from*

*https://redirector.gvt1.com/edgedl/android/studio/ide-zips/4.1.1.0/android-studio-ide-201.6953283-linux.tar.gz*

1. *Untar downloaded file*

1. *tar -xvzf <file\_name>*
2. Move untarred file to /home/developers directory
3. mv <file\_name> /home/developers/
4. Add desktop Entry
5. vim /home/developers/.local/share/applications/android-studio.desktop
6. Add below entries to desktop file

[Desktop Entry]

Version=1.0

Type=Application

Name=Android Studio

Exec="/home/developers/android-studio/bin/studio.sh" %f

Icon=/home/developers/android-studio/bin/studio.png

Categories=Development;IDE;

Terminal=false

StartupNotify=true

StartupWMClass=jetbrains-android-studio

Name[en\_GB]=android-studio.desktop

* *errors*

1. */dev/kvm is Not Found and Device Permission Denied Errors*

* *Solution*

1. *Enable virtualization from bios*

*Ref:* [*https://www.codeproject.com/Articles/5266468/Fix-dev-kvm-is-Not-Found-and-Device-Permission-Den*](https://www.codeproject.com/Articles/5266468/Fix-dev-kvm-is-Not-Found-and-Device-Permission-Den)

* *Installing docker*

1. *sudo apt-get install -y ca-certificates curl gnupg lsb-release*
2. *curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg*
3. echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
4. *apt update*
5. *Check docker required version with below command*
6. *apt-cache madison docker-ce | grep 5:19.03.15*

1. *Install docker with following command*
2. *sudo apt-get update;sudo apt-get install docker-ce=5:19.03.15~3-0~ubuntu-bionic docker-ce-cli=5:19.03.15~3-0~ubuntu-bionic containerd.io*

***OR***

1. *apt-get update;apt-get install -y docker-ce=$(apt-cache madison docker-ce | grep 19.03.15 | head -1 | awk '{print $3}')*
2. *Check docker version with*

*docker -v*

***Ref:***

[*https://docs.docker.com/engine/install/ubuntu/*](https://docs.docker.com/engine/install/ubuntu/)

1. *Create the docker group and add user to that group*
2. *sudo groupadd docker*
3. *sudo usermod -aG docker developers*
4. *newgrp docker*
5. *Check if user added to docker group*
6. getent group docker

***Ref:***

*https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user*

* *Uninstalling docker*

1. *Run*
2. *sudo apt-get purge -y docker-engine docker docker.io docker-ce docker-ce-cli*
3. *sudo apt-get autoremove -y --purge docker-engine docker docker.io docker-ce*
4. *sudo apt-get remove docker docker-engine docker.io containerd runc*
5. *sudo rm -rf /var/lib/docker /etc/docker*
6. *sudo rm /etc/apparmor.d/docker sudo groupdel docker*
7. *sudo rm -rf /var/run/docker.sock*
8. *docker -v*

* *Installing docker-compose*

1. *Download docker-compose with below link*
2. *sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose*

1. *Change version in above command if you want another version of docker-compose*
2. *Change permissions of file*
3. *chmod ugo+x /usr/local/bin/docker-compose*
4. *Check version*

*Docker-compose*

* *Errors*

developers@linux:/var/lib/docker$ docker ps

Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/containers/json: dial unix /var/run/docker.sock: connect: permission denied

* *Solution*

1. *Change permission of* /var/run/docker.sock *file with*

*chmod 666 /var/run/docker.sock*

* *Installing Docker-Desktop*

***Ref:***[*https://docs.docker.com/desktop/linux/*](https://docs.docker.com/desktop/linux/)

* *Installing Maven*

1. *Goto*

cd /home/developers/Downloads

1. *Download Apache maven from*

*wget* [*https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz*](https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz)

1. *Untar file with*

tar -xvzf apache-maven-3.8.4-bin.tar.gz

1. *Move apache-maven-3.8.4 file*

mv apache-maven-3.8.4 /opt/

1. *Run Below command*
2. export PATH=/opt/apache-maven-3.8.4/bin:$PATH
3. *Check maven version*

mvn -v

* *Installing Simple Screen Recorder*

1. sudo apt-add-repository ppa:maarten-baert/simplescreenrecorder
2. sudo apt-get update
3. sudo apt-get install simplescreenrecorder

* *Git clone issue*
* *Error*

*admin1@linux:~$ git clone -b Madhuri http://10.13.10.216/shital.dalavi/NewInstallerPortalWithQR*

*Cloning into 'NewInstallerPortalWithQR'...*

*remote: Counting objects: 8345, done.*

*remote: Compressing objects: 100% (6035/6035), done.*

*remote: aborting due to possible repository corruption on the remote side.*

*fatal: early EOF*

*fatal: index-pack failed*

* *Solution*

1. *Go to dir where you want to clone the repository*
2. *Enter below commands*
3. *git config --global pack.windowMemory "100m"*
4. *git config --global pack.SizeLimit "100m"*
5. *git config --global pack.threads "1"*
6. *git config --global pack.window "0"*

* *Installing mantis, mariadb, apache2*

1. *Installation*

*Ref:*

*https://websiteforstudents.com/mantis-bug-tracker-mantisbt-on-ubuntu-18-04-16-04-with-apache2/*

1. *Backup and restore Mysql Database*
2. mysql -u root -p < alldatabases.sql

*Ref:*

[*https://blog.devart.com/how-to-restore-mysql-database-from-backup.html#:~:text=Restore%20a%20MySQL%20database%20from%20the%20.,-sql%20file%20using&text=Open%20MySQL%20Workbench%20and%20click,backup%20of%20the%20sakila%20database*](https://blog.devart.com/how-to-restore-mysql-database-from-backup.html#:~:text=Restore%20a%20MySQL%20database%20from%20the%20.,-sql%20file%20using&text=Open%20MySQL%20Workbench%20and%20click,backup%20of%20the%20sakila%20database)*.*

*https://phoenixnap.com/kb/how-to-backup-restore-a-mysql-database*

1. *To update existing user’s password of mysql*

*(Run following command in mysql shell)*

*a.update mysql.user set authentication\_string='password' where user='’username’;*

*b.UPDATE mysql.user SET Password=PASSWORD('foobar') WHERE USER='tom' AND Host='localhost';*

1. *To see all users in mysql*

1. *SELECT User, Host FROM mysql.user;*

* *Installing virtualbox 6.0*

***Ref:***

[*https://linuxize.com/post/how-to-install-virtualbox-on-ubuntu-18-04/*](https://linuxize.com/post/how-to-install-virtualbox-on-ubuntu-18-04/)

* *To check validity of .cert and .pem files*

1. *openssl x509 -enddate -noout -in file.pem*
2. *openssl x509 -enddate -noout -in file.cert*

* *To check validity of .jks file*

1. *keytool -list -v -keystore mqtt-client-trust-store.jks*

* *Upgrade php version*

<https://www.itsolutionstuff.com/post/how-to-upgrade-php-version-from-74-to-8-in-ubuntuexample.html>

* *Installing node*

1. *Installing node version 10.16.3*
2. *Download node from*

*https://nodejs.org/dist/v10.16.3/node-v10.16.3-linux-x64.tar.gz*

1. *cd /home/developers/Downloads*
2. *wget https://nodejs.org/dist/v10.16.3/node-v10.16.3-linux-x64.tar.gz*

***OR***

1. *Pull from root@10.13.10.124:/root/node-v10.16.3-linux-x64.tar.gz*
2. *Untar node with following command*

tar -xvzf node-v10.16.3-linux-x64.tar.gz

1. *mv* node-v10.16.3-linux-x64 node/usr/local/share/packages/
2. export NODEJS\_HOME=/usr/local/share/packages/node-v10.16.3-linux-x64
3. export PATH=$HOME/.local/bin:$bin:$PATH:$NODEJS\_HOME/bin
4. *cd /home/developers/workspace*
5. *ln -s* /usr/local/share/packages/
6. To Install different versions of node goto following link and download and follow above installation steps

*https://nodejs.org/dist/*

* *Upgrading npm version*

*npm install npm@latest -g*

***OR***

*npm install npm@l<version> -g*

* *Host file path windows*

*c:\Windows\System32\Drivers\etc\hosts*

* *DPKG ERROR*

Note: only when purging packages

*error: dpkg error processing packages*

*Solution: rm /var/lib/dpkg/info/<the-package-causing-error>*

* *Installing Spark*

1. *Download tar of spark 2.4.0 from below link*

[*https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz*](https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz)

#!/bin/bash

cd /home/developers/Downloads

wget https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz

tar -xvzf spark-2.4.0-bin-hadoop2.7.tgz

mkdir -p /opt/spark

mv spark-2.4.0-bin-hadoop2.7\* /opt/spark

echo "export SPARK\_HOME=/opt/spark" >> /home/developers/.profile

echo "export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin" >> /home/developers/.profile

echo "export PYSPARK\_PYTHON=/usr/bin/python3" >>/home/developers/.profile

sed -i -e '$aexport SPARK\_HOME=/opt/spark' /home/developers/.bashrc

sed -i -e '$aexport PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin' /home/developers/.bashrc

sed -i -e '$aexport PYSPARK\_PYTHON=/usr/bin/python3' /home/developers/.bashrc

chown -R developers:developers /opt/spark

source /home/developers/.profile

source /home/developers/.bashrc

1. *Start master*
2. *start-master.sh*
3. *Check spark UI on browser*
4. *0.0.0.0:8080*

***Ref****:**https://phoenixnap.com/kb/install-spark-on-ubuntu*

* *Installing typescript*

*npm i typescript -g*

* *Creating docker registry with simple authentication*

1. *Installing htpasswd tool for creating authentication*
2. *sudo apt update;apt install apache2-utils -y*
3. *Check if docker is installed*
4. *Create following directories with following commands*
5. *mkdir -p /etc/docker/DockerV2/auth*
6. *mkdir -p /usr/local/share/packages/V2/DockerV2/registry*
7. *Create htpasswd file at /etc/docker/DockerV2/auth location with following command*
8. *htpasswd -c -B -b </path/to/users.htpasswd> <user\_name> <password>*
9. *A file will be created with encrypted password*
10. *Add following entry in /etc/docker/daemon.json file*
11. *vim /etc/docker/daemon.json*
12. *{*

*"insecure-registries" : ["ip\_of\_machine:8443"]*

*}*

1. *Restart docker*
2. *service docker restart*
3. *service docker status*

1. *Create docker registry with following command*
2. *docker run -itd --restart=always -v /etc/docker/DockerV2/auth:/etc/auth -e "REGISTRY\_AUTH=htpasswd" -e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" -e REGISTRY\_AUTH\_HTPASSWD\_PATH=/etc/auth/htpasswd -v /usr/local/share/packages/V2/DockerV2/registry:/var/lib/registry -p 8443:5000 registry*

*OR*

*docker run -itd --restart=always -v /usr/local/share/packages/registry/auth:/etc/auth -e "REGISTRY\_AUTH=htpasswd" -e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" -e REGISTRY\_AUTH\_HTPASSWD\_PATH=/etc/auth/htpasswd -v /usr/local/share/packages/V2/DockerV2/registry:/var/lib/registry -p 8443:5000 registry*

1. *Login to docker registry with*
2. *docker login ip\_of\_machine:5443*
3. *Enter username and password provided in htpasswd command*
4. *You should see “Successful login” log*

* *Installing Hadoop Stand-alone/local mode*

1. *Switch to hadoop user*
2. *Make passwordless ssh for self*
3. ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa
4. cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

1. *Copy hadoop-3.1.1 jar file from centralized machine*
2. rsync -r --progress -e 'ssh -p 513' *root@10.13.10.124:/root/Avinash/hadoop-3.1.1.tar.gz* /home/hadoop/
3. *Untar file*
4. tar -xvzf hadoop-3.1.1.tar.gz

(In the above hadoop directory all configurations are already configured as per our requirement)

1. *Add following lines to* /home/hadoop/.bashrc

HADOOP\_PREFIX=/home/hadoop/hadoop-3.1.1

PATH=$PATH:$HADOOP\_PREFIX/bin

export PATH JAVA\_HOME HADOOP\_PREFIX

1. source ~/.bashrc
2. *Start namenode, secondary namenode,datanode with following command*
3. cd /home/hadoop/hadoop-3.1.1/sbin
4. ./start-dfs.sh
5. jps

**Ref:** https://www.tecmint.com/install-hadoop-single-node-on-centos-7/

* *Installing mongodb 4.4.8*

1. sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 656408E390CFB1F5
2. sudo echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb.list
3. sudo apt update
4. sudo apt update;sudo apt install mongodb-org=4.4.8 mongodb-org-server=4.4.8 mongodb-org-shell=4.4.8 mongodb-org-mongos=4.4.8 mongodb-org-tools=4.4.8 -y

*To install a specific version add version after =*

1. *sudo systemctl restart mongod.service*
2. *sudo systemctl status mongod.service*
3. *mongod --version*
4. *Give non root user access on mongod with following command*
5. *chown -R developers:developers /etc/mongod.conf*
6. *chmod 777 /etc/mongod.conf*
7. *Add an entry in sudoers file*
8. *developers ALL=/usr/sbin/vpnc,/usr/sbin/vpnc-disconnect,/etc/init.d/nginx,/bin/systemctl start mongod.service, /bin/systemctl stop mongod.service, /bin/systemctl restart mongod.service, /bin/systemctl status mongod.service*
9. *Check if that user has access to start and stop command of mongod service*

***Ref:*** [*https://tecadmin.net/install-mongodb-on-ubuntu/*](https://tecadmin.net/install-mongodb-on-ubuntu/)

* *Uninstalling mongodb*

1. dpkg -l | grep mongo
2. apt-get purge mongodb\*
3. rm -rf /etc/apt/sources.list.d/mongodb\*
4. rm -rf /var/lib/mongodb
5. rm -rf /var/log/mongodb
6. rm -rf /var/lib/apt/lists/repo.mongodb.\*
7. sudo userdel -r mongodb
8. sudo groupdel mongodb
9. dpkg -l | grep mongo

* Checking serial number of a machine

dmidecode | grep -A 9 "System Information"

* Phoenix sample script command

1. *path : cd workspace/COMMON\_BUILD/target/dependency*
2. *script : java -Dclouds.script.manual.execution.list="phoenixSampleScript" -cp \*: com.clouds.migrator.script.main.DataMigratorScriptExecutor*

* WFH & WFO script

1. *Run below script with developers user to change /etc/hosts file configuration*
2. *Change file permissions after creating file with following commands*
3. *vim /usr/bin/network-config*
4. *chmod 4755 /usr/bin/network-config*

*#!/bin/bash*

*echo -e "Type WFH or WFO below \n\nWFH/WFO ? :\n"*

*read ans*

*echo -e "\n"*

*if [ -z "$ans" ]*

*then*

*echo "Answer can not be empty.Please enter valid option"*

*elif [ "$ans" = "WFH" ] ;*

*then*

*sudo -i sed -i '/10.13.10/d' /etc/hosts*

*sudo -i sed -i '$ a 123.201.117.4 dev.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 123.201.117.4 tempdev.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 125.99.46.46 test.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 125.99.46.46 temp.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 123.201.117.6 mantis.clouzerindia.com' /etc/hosts*

*sudo -i sed -i '$ a 125.99.46.47 ubuntu124.mcloud.com' /etc/hosts*

*sudo -i sed -i '$ a 125.99.46.48 passbolt.clouzerindia.com' /etc/hosts*

*echo -e "\n /etc/host file is configured for WFH\n"*

*elif [ "$ans" = "WFO" ] ;*

*then*

*sudo -i sed -i '/123/d' /etc/hosts*

*sudo -i sed -i '/125/d' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.11 dev.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.11 tempdev.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.18 test.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.18 temp.clouzer.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.124 ubuntu124.mcloud.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.10.205 mantis.clouzerindia.com' /etc/hosts*

*sudo -i sed -i '$ a 10.13.15.196 passbolt.clouzerindia.com' /etc/hosts*

*echo "/etc/host file is configured for WFO"*

*fi*

*3. Add below line in /etc/sudoers file*

*developers ALL=NOPASSWD: !ALL, /bin/bash/sed, /bin/bash*

* *Installing arp-scan in node:alpine*

1. *wget* [*https://github.com/royhills/arp-scan/archive/master.tar.gz*](https://github.com/royhills/arp-scan/archive/master.tar.gz)
2. *tar -xvzf master.tar.gz*
3. *cd arp-scan-master/*
4. *apk --no-cache add build-base libpcap-dev autoconf automake*
5. *autoreconf --install*
6. *./configure*
7. *Make*
8. *make install*
9. *Copy binary to /usr/bin*

* *Installing vlc*

1. snap install vlc

* *Office network configuration*
* Add below entries to /etc/hosts file

10.13.10.17 www.mcloud.com www

10.13.10.11 dev.clouzer.com

10.13.10.11 tempdev.clouzer.com

10.13.10.18 test.clouzer.com

10.13.10.18 temp.clouzer.com

10.13.10.205 mantis.clouzerindia.com

* Add below entries to /etc/resolv.conf file

nameserver 10.13.10.17

nameserver 10.13.10.47

nameserver 8.8.8.8

search mcloud.com

* *Sudores Access without Password*

<user\_name> ALL = NOPASSWD: /bin/systemctl restart httpd.service, /bin/kill

* *Changing transient hostname*

hostnamectl set-hostname --transient <hostname>

* *Maven build*

mvn clean dependency:copy-dependencies package -DskipTests

* *Creating jks with .cert and .key file*

1. openssl pkcs12 -export -in abc.crt -inkey abc.key -out abc.p12

1. keytool -importkeystore -srckeystore abc.p12 -srcstoretype PKCS12 -destkeystore abc.p12 -deststoretype JKS

**OR**  
  
keytool -importcert -file "server.crt" -keystore mqtt-client-trust-store.jks -alias "hive"

***Ref:*** [*https://community.datarobot.com/t5/platform/how-to-convert-crt-and-key-to-jks-file/td-p/6342*](https://community.datarobot.com/t5/platform/how-to-convert-crt-and-key-to-jks-file/td-p/6342)

* *Checking Endate of .crt file*

1. openssl x509 -enddate -noout -in /path/to/.crt

* *View Certificates*

1. openssl x509 -in /etc/letsencrypt/live/www.nciportal.com/fullchain.pem -noout -text

* *Docker open file limit issue*

<https://serverfault.com/questions/1000684/docker-failed-to-start>

* *Listing directories with max space utilization*

du -a / | sort -n -r | head -n 10

Ref: <https://www.tecmint.com/find-top-large-directories-and-files-sizes-in-linux/>

* *Installing NFS*

Ref: <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-18-04>

* *Renew Certs with certbot*

apt-get install python3-certbot-nginx -y

certbot renew --cert-name build.clouzerindia.com

OR

certbot certonly --apache -d ncplmail.clouzerindia.com --email avinash.gaikwad@nciportal.com --rsa-key-size 4096 --agree-tos --no-eff-email

* *Delete dir*

ls -t | tail -n +6 | xargs -d '\n' rm -r

* *Vmware VM Details*

vim-cmd vmsvc/getallvms

* *Check server latency with curl*

curl -o /dev/null -s -w "\

time\_namelookup: %{time\_namelookup}\n\

time\_connect: %{time\_connect}\n\

time\_appconnect: %{time\_appconnect}\n\

time\_pretransfer: %{time\_pretransfer}\n\

time\_redirect: %{time\_redirect}\n\

time\_starttransfer: %{time\_starttransfer}\n\

----------\n\

time\_total: %{time\_total}\n" \

<https://test.clouzer.com>

* *Open Port*

sudo setcap 'cap\_net\_bind\_service=+ep' `which node`