**Detail Explanation Provisioning Script**

**Set up EC2 Instance with Specific Services configuration done by provisioning script**

This script will build EC2 instance that we will support with Ubuntu 16.04 LTS which requires the following services

1. Nginx
   * Create default virtual host which will be located in /var/www/tdapp/ and describe the config of the vhost into the nginx configuration file
2. PHP 7.0
3. Python v 2.7
4. Node.js - v6.10.3
5. Git 2.7.4
6. Curl 7.47
7. Composer installed
8. PHP Errors On
   * PHP Modules installed, Core, PDO, PDO\_ODBC, Phar, Reflection, SPL, SimpleXML, Opcache, apache2handler, bcmath, bz2, calendar, ctype, curl, date, dom, enchant, Exif, mcrypt, memcached, mongodb, mysqli, mysqlnd, odbc, openssl, Pcre
   * PHP Specific configuration applied:
     + Memory\_limit = 512 MB
     + Max\_execution\_time = 10 minutes
     + Post\_max\_size = 20MB
     + Session Handler should be memcached, not local storage
     + Short tags Off
9. Percona 5.7
   * Working on localhost
   * User: root, password: random
   * Access data added to /root/.my.cnf
   * Database created by default - tddatabase
   * Read-only and read-write users for tddatabase
10. MongoDB
    * Set on localhost 27017
    * Default collection created - tdmongodb
11. Memcached installed
    * Localhost , 11211
12. Redis Installed
    * localhost , 6379
13. Node JS
    * NVM, Gulp, Grunt, Bower, Yo, Browser-sync, Browserify, Pm2, Webpack
14. Connectivity Issue of server

**Instructions for Provisioning of Script:**

All of these services above should be automatically installed and reinstalled on provisioning. The provisioning script should be WELL documented and commented so it’s easy to understand what each next step is. Because of that when deploying the EC2 instance there should be proper output of each step executed in the process.

A few notes:

* It has to be working from the first time and should have no errors
* It has to log the deployment process into deploy.log file
* It has to be possible to provision the EC2 instance without any issues
* All services should AUTO start when the EC2 instance is restarted

**Introduction:**  
This document is the detail explanation of provisioning script. Comments are added in the provisioning scripts and it’s very easy for a technical person to read the and understand the script, for non-technical they can read the steps of script which will run when script will be executed so everyone can see what steps are in process and what process being done.  
  
There are different techniques to do this task but I follow the simplest one to make easier and effective to understand by others. I created a repo and added that script in my public repo and I will update my script whenever I need or found something interesting. The script have no errors, it runs very smoothly.

This script includes too many sleeps to check and read all the logs easily but you can edit sleep time to run this script faster. By adding sleeps, you can easily interrupt the commands and statement if you don’t wanted to install or execute the script any more. This script include two statements for each service installation. First one is for installation of service and the next one is verification of previous installation.  
  
This script first check that the required package from above mention list is already installed or not, if the required service is already installed then it will skip that service installation and move to next one. This check statement is added for safety of already running and configured services. Reference links are attached.  
  
1. The script is mention below and every one can run it. From command line please paste the below command

wget [https://raw.githubusercontent.com/hassanhashmy/npm/master/provioning.sh h](https://raw.githubusercontent.com/hassanhashmy/npm/master/provioning.sh%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20h)   
2. Give permission to the script  
chmod +x provioning.sh h   
3. Execute the script and read the messages on the screen and log file will be created in root with deploy.log  
./provioning.sh h

**Challenging Part:**

The only headache was to keep the node version same as 6.10.3 and install other components as well. Earlier I was using nvm to install node specific version but for that I have to open a new terminal which was not possible for simple approach and I don’t wanted to make it complex so I was just trying to solve the problem. Node was very important for other node components but due to nvm all that work was not possible so I decided to use binary package of node then finally I solved the issue.

**#1 Pre-Installation Part:**

This statement will only check that OS version otherwise exit the script. This is just Ubuntu Release check if the release is 16.04 then script will continue otherwise it will return exit 1 and stop the execution of script.

rel=$(lsb\_release -a |head -n3 |tail -n1 | awk '{print $2}')

if [ "$rel" = "16.04" ];

then

#SOME COMMANDS; sleep 600

echo "This is Ubuntu 16.04 LTS and Comptible for this script";

echo "Now checking whether Nginx is installed or not";

else

echo -e "\033[30;5;101mCompatible OS Ubuntu 16.04 LTS Not Found...Please Run this script on Compatible Version of OS\033[0m";

exit 1

fi

**#2 Nginx Installation:**

This statement will first check that nginx is installed or not if not then it will install. This command will check that nginx is installed or not

nginxver=$(apt show nginx | grep yes | wc -l)

if [ "$nginxver" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Nginx Not found";

echo "Nginx Installation Starts please wait for 3 sec";

apt-get update -y >> /root/deploy.log 2>&1; sleep 30

apt-get install nginx procps -y >> /root/deploy.log; sleep 20

echo '################## Nginx Installation Finished ####################' >> /root/deploy.log;

mkdir /var/www/tdapp

chown -R www-data:www-data /var/www/tdapp/

echo "Nginx vhost configuration starts";

# Nginx configuration is starting

sed -i '/# Basic Settings/ i index index.html;\nserver {\nserver\_name test.hybytes.com;\naccess\_log /var/log/nginx/tdapp.access.log;\nroot /var/www/tdapp/;\nlocation / {\nroot /var/www/tdapp/;\nindex index.html index.htm;\n# First attempt to serve request as file, then\n# as directory, then fall back to displaying a 404.\ntry\_files \$uri \$uri/ /index.html;\n# Uncomment to enable naxsi on this location\n# include /etc/nginx/naxsi.rules\n}\n}' /etc/nginx/nginx.conf

echo "Nginx Configuration Status";

nginx -t

service nginx start; sleep 10

update-rc.d nginx defaults; sleep 3

update-rc.d nginx enable; sleep 3

systemctl enable nginx.service; sleep 2

sleep 40

else

echo -e "\033[30;43mNginx is already Installed.....Nginx Installation is going to Skip and moving to Php and Php modules Installation\033[0m";

fi

This statement will check that nginx installed or not. This command will check that nginx process is running or not

nginxstat=$(ps waux | grep nginx | grep www-data |wc -l)

if [ "$nginxstat" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Nginx Verified";

ps waux | grep nginx | grep master

echo "Now checking whether Php 7 is installed or not";

else

echo -e "\033[30;5;101mNginx Sucessfully not Intalled and please check\033[0m";

fi

**#3 Php and Its Modules with Php custom configuration:**

This statement will check that php is installed or not if not then install. This command will check that php is installed or not

phpf=$(which php |wc -l)

if [ "$phpf" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Php Not found";

echo "Php 7 Installation Starts please wait for few sec";

apt-get install -y php7.0 php7.0-fpm php7.0-cli php7.0-common php7.0-mbstring php7.0-gd php7.0-intl php7.0-xml php7.0-mysql php7.0-mcrypt php7.0-zip php7.0-odbc php7.0-bcmath php7.0-bz2 php7.0-curl php7.0-enchant php7.0-dev php-pear php-mongodb php-memcached >> /root/deploy.log 2>&1; sleep 35

echo '################# Php Installation Finished ##################' >> /root/deploy.log;

echo "PHP Log Errors is going to be enabled and PHP Specific configuration is going to implement";

# php configurations is starting to set

sed -i '/display\_errors = Off/c\display\_errors = On' /etc/php/7.0/cli/php.ini

sed -i '/display\_startup\_errors = Off/c\display\_startup\_errors = On' /etc/php/7.0/cli/php.ini

sed -i '/display\_errors = Off/c\display\_errors = On' /etc/php/7.0/fpm/php.ini

sed -i '/display\_startup\_errors = Off/c\display\_startup\_errors = On' /etc/php/7.0/fpm/php.ini

sed -i '/post\_max\_size = 8M/c\post\_max\_size = 20M' /etc/php/7.0/cli/php.ini

sed -i '/post\_max\_size = 8M/c\post\_max\_size = 20M' /etc/php/7.0/fpm/php.ini

sed -i '/max\_execution\_time = 30/c\max\_execution\_time = 600' /etc/php/7.0/cli/php.ini

sed -i '/max\_execution\_time = 30/c\max\_execution\_time = 600' /etc/php/7.0/fpm/php.ini

sed -i '/memory\_limit = 128M/c\memory\_limit = 512M' /etc/php/7.0/cli/php.ini

sed -i '/memory\_limit = 128M/c\memory\_limit = 512M' /etc/php/7.0/fpm/php.ini

sed -i '/session.save\_handler = files/c\session.save\_handler = memcached' /etc/php/7.0/fpm/php.ini

sed -i '/session.save\_handler = files/c\session.save\_handler = memcached' /etc/php/7.0/cli/php.ini

sed -i '/session.save\_handler = memcached/a session.save\_path = "localhost:11211"' /etc/php/7.0/cli/php.ini

sed -i '/session.save\_handler = memcached/a session.save\_path = "localhost:11211"' /etc/php/7.0/fpm/php.ini

else

echo -e "\033[30;43mPhp is already Installed.....Php Installation is going to Skip and moving to Python 2.7 Installation\033[0m";

fi

This statement is checking that php installed or not. This command will check that php is installed or not

phpf=$(which php |wc -l)

if [ "$phpf" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Php 7 Verified";

php -v

echo "Now checking whether Python is installed or not";

else

echo -e "\033[30;5;101mPhp Not Installed sucessfully....Please check\033[0m";

fi

**#4 Python Installation:**

This statement will check that pythn is installed or not if not then install. This command will check that python is installed or not

pypf=$(find /usr/src -xdev 2>/dev/null -name "Python-2.7.14" |wc -l)

if [ "$pypf" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Python Not found";

echo "Python 2.7 Installation Starts please wait for few sec";

apt-get install make gcc build-essential checkinstall libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev -y >> /root/deploy.log 2>&1; sleep 30

echo '################## Python Dependencies Installation Finished ####################' >> /root/deploy.log;

cd /usr/src

wget https://www.python.org/ftp/python/2.7.14/Python-2.7.14.tgz >> /root/deploy.log 2>&1; sleep 25

tar xzf Python-2.7.14.tgz >> /root/deploy.log 2>&1; sleep 10

cd Python-2.7.14

./configure >> /root/deploy.log 2>&1; sleep 90

echo '#################### Python Configuration Finished ####################' >> /root/deploy.log;

make altinstall >> /root/deploy.log 2>&1; sleep 90

echo '################# Python Installation Finished ####################' >> /root/deploy.log;

else

echo -e "\033[30;43mPython is already Installed.....Python Installation is going to Skip and moving to Git 2.7.4 Installation\033[0m";

fi

This statement will check that python installed correctly or not. This command will check that python is installed or not

pypf=$(find /usr/src -xdev 2>/dev/null -name "Python-2.7.14" |wc -l)

if [ "$pypf" = "2" ];

then

#SOME COMMANDS; sleep 600

echo "Python Verified";

python2.7 -V

echo "Now checking whether Git is installed or not";

else

echo -e "\033[30;5;101mPython not installed Sucessfully...Please check\033[0m";

fi

**#5 Git Installation:**

This statement will check that git installed or not otherwsie install. This command will check that git is installed or not

gitf=$(which git |wc -l)

if [ "$gitf" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Git Not found";

echo "Git 2.7.4 Installation Starts please wait for few sec";

apt-get install libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev asciidoc xmlto docbook2x autoconf -y >> /root/deploy.log 2>&1; sleep 25

echo "################## Git Dependencies Installation Finished ###############" >> /root/deploy.log;

cd /usr/src

wget https://www.kernel.org/pub/software/scm/git/git-2.7.4.tar.gz >> /root/deploy.log 2>&1; sleep 25

tar -xvzf git-2.7.4.tar.gz >> /root/deploy.log 2>&1; sleep 2

cd git-2.7.4/

./configure --prefix=/usr >> /root/deploy.log 2>&1; sleep 60

echo '############## Git Configuration Finished ###############' >> /root/deploy.log;

make all doc info prefix=/usr >> root/deploy.log 2>&1; sleep 150

make install install-doc install-html install-info install-man prefix=/usr >> /root/deploy.log 2>&1; sleep 600

echo '############# Git 2.7 Installation Finished #############' >> /root/deploy.log;

else

echo -e "\033[30;43mGit is already Installed.....Git installation is going to skip and moving to Curl 7.4.7\033[0m";

fi

This statement will check that git installed correctly or not. This command will check that git is installed or not

gitf=$(which git |wc -l)

if [ "$gitf" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Git 2.7.4 Verified";

git --version

echo "Now checking whether Curl 2.7.4 is installed or not";

else

echo -e "\033[30;5;101mGit not installed Sucessfully...Please check\033[0m";

fi

**#6 Curl Installation:**

This statement will check that curl installed or not otherwsie install. This command will check that curl is installed or not

curlf=$(which curl |wc -l)

if [ "$curlf" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Curl Not found";

echo "Curl 7.47 Installation Starts please wait for few sec";

echo "deb http://security.ubuntu.com/ubuntu xenial-security main" >> /etc/apt/sources.list >> /root/deploy.log 2>&1; sleep 1

apt-get install curl -y >> /root/deploy.log 2>&1; sleep 2

echo '################# Curl 7.47 Installation Finished #################' >> /root/deploy.log;

else

echo -e "\033[30;43mCurl is already Installed.....Curl installation is going to skip and moving to Composer\033[0m";

fi

This statement will check that curl installed correctly or not. This command will check that curl is installed or not

curlf=$(which curl |wc -l)

if [ "$curlf" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Curl 7.47 Verified";

curl --version

echo "Now checking whether Composer is installed or not";

else

echo -e "\033[30;5;101mCurl not installed Sucessfully...Please check\033[0m";

fi

**#7 Composer Installation:**

This statement will check that composer installed or not otherwise install. This command will check that composer is installed or not

compf=$(which composer |wc -l)

if [ "$compf" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Composer Not found";

echo "Composer Installation Starts please wait for few sec";

cd /root

wget https://getcomposer.org/composer.phar >> /root/deploy.log 2>&1; sleep 7

echo '################## Composer Downloading Finished ################' >> /root/deploy.log;

mv composer.phar /bin/composer

chmod +x /bin/composer

else

echo -e "\033[30;43mComposer is already Installed.....Composer installation is going to skip and moving to Percona 5.7\033[0m";

fi

This statement will check that curl installed correctly or not. This command will check that composer is installed or not

compf=$(which composer |wc -l)

if [ "$compf" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Composer Verified";

composer -V

echo "Now checking whether Percona 5.7 is installed or not";

else

echo -e "\033[30;5;101mComposer not installed Sucessfully...Please check\033[0m";

fi

**#8 Percona Installation and Configuration:**

This statement will check that percona installed or not otherwsie install. This command will check that percona is installed or not

percp=$(which mysql |wc -l)

if [ "$percp" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Percona Server Not found";

echo "Percona Server 5.7 Installation Starts please wait for few sec";

cd /root

wget https://repo.percona.com/apt/percona-release\_0.1-4.$(lsb\_release -sc)\_all.deb >> /root/deploy.log 2>&1; sleep 20

echo '############## Percona Repo Downloading Finished ############' >> /root/deploy.log;

dpkg -i percona-release\_0.1-4.$(lsb\_release -sc)\_all.deb >> /root/deploy.log 2>&1; sleep 2

apt-get update >> /root/deploy.log 2>&1; sleep 20

dbpass="random" && export dbpass

export DEBIAN\_FRONTEND=noninteractive

echo percona-server-server-5.7 percona-server-server-5.7/root\_password password $dbpass | debconf-set-selections

echo percona-server-server-5.7 percona-server-server-5.7/root\_password\_again password $dbpass | debconf-set-selections

apt-get install -y percona-server-server-5.7 percona-server-client-5.7 >> /root/deploy.log 2>&1; sleep 40

echo '############## Percona 5.7 Installation Finished ##############' >> /root/deploy.log;

debconf-show --listowners >> /root/deploy.log 2>&1; sleep 1

debconf-show percona-server-server-5.7 >> /root/deploy.log 2>&1; sleep 1

cp /etc/mysql/percona-server.conf.d/mysqld.cnf /root/.my.cnf

touch /etc/init.d/mysqls

echo "#This is script" > /etc/init.d/mysqls

sed -i '$ a #!/bin/bash' /etc/init.d/mysqls

sed -i '$ a pkill mysqld' /etc/init.d/mysqls

sed -i '$ a mkdir -p /var/run/mysqld' /etc/init.d/mysqls

sed -i '$ a chown mysql:mysql /var/run/mysqld' /etc/init.d/mysqls

sed -i '$ a /usr/bin/mysqld\_safe --defaults-file=/root/.my.cnf & 2>/dev/null' /etc/init.d/mysqls

sed -i '$ a exit 0' /etc/init.d/mysqls

chmod +x /etc/init.d/mysqls

mysql -uroot -prandom -e "CREATE DATABASE tddatabase CHARACTER SET utf8 COLLATE utf8\_general\_ci"; >> /root/deploy.log 2>&1; sleep 1

mysql -uroot -prandom -e "CREATE USER ruser@'127.0.0.1' IDENTIFIED BY 'random'"; >> /root/deploy.log 2>&1; sleep 1

mysql -uroot -prandom -e "CREATE USER rwuser@'127.0.0.1' IDENTIFIED BY 'random'"; >> /root/deploy.log 2>&1; sleep 1

mysql -uroot -prandom -e "GRANT SELECT ON tddatabase.\* TO 'ruser'@'127.0.0.1'"; >> /root/deploy.log 2>&1; sleep 1

mysql -uroot -prandom -e "GRANT SELECT, INSERT, UPDATE ON tddatabase.\* TO 'rwuser'@'127.0.0.1'"; >> /root/deploy.log 2>&1; sleep 1

sed -i '$i/etc/init.d/mysqls' /etc/rc.local

systemctl stop mysql.service >> /root/deploy.log 2>&1; sleep 10

systemctl disable mysql.service >> /root/deploy.log 2>&1; sleep 20

/etc/init.d/mysqls >> /root/deploy.log 2>&1; sleep 30

else

echo -e "\033[30;43mPercona 5.7 is already Installed.....Percona 5.7 installation is going to skip and moving to mongoDB\033[0m";

fi

This statement will check that percona installed correctly or not. This command will check that percona is installed or not

percp=$(which mysql |wc -l)

if [ "$percp" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Percona Server Verified";

systemctl status mysql |grep mysql.service

echo "Now checking whether Mongodb is installed or not";

else

echo -e "\033[30;5;101mPercona Server 5.7 not installed Sucessfully...Please check\033[0m";

fi

**#9 MongoDB installation and Configuration:**

This statement will check that mongodb installed or not otherwsie install. This command will check that mongo is installed or not

mongp=$(which mongo | wc -l)

if [ "$mongp" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Mongodb Not found";

echo "Mogodb Installation Starts please wait for few sec";

apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927 >> /root/deploy.log 2>&1; sleep 2

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 >> /root/deploy.log 2>&1; sleep 25

echo '############# MongoDB Repo Downloading Finished ##############' >> /root/deploy.log;

apt-get update >> /root/deploy.log 2>&1; sleep 20

apt-get install -y --allow-unauthenticated mongodb >> /root/deploy.log 2>&1; sleep 30

apt-get install procps -y >> /root/deploy.log 2>&1; sleep 1

echo '############ Mogodb Installation Finished ############' >> /root/deploy.log;

touch /etc/systemd/system/mongodb.service

echo "#This is script" > /etc/systemd/system/mongodb.service

sed -i '$ a #!/bin/bash' /etc/systemd/system/mongodb.service

sed -i '$ a [Unit]' /etc/systemd/system/mongodb.service

sed -i '$ a Description=High-performance, schema-free document-oriented database' /etc/systemd/system/mongodb.service

sed -i '$ a After=network.target' /etc/systemd/system/mongodb.service

sed -i '$ a [Service]' /etc/systemd/system/mongodb.service

sed -i '$ a User=mongodb' /etc/systemd/system/mongodb.service

sed -i '$ a ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf' /etc/systemd/system/mongodb.service

sed -i '$ a [Install]' /etc/systemd/system/mongodb.service

sed -i '$ a WantedBy=multi-user.target' /etc/systemd/system/mongodb.service

chmod +x /etc/systemd/system/mongodb.service

cp /etc/systemd/system/mongodb.service /lib/systemd/system/

apt-get remove insserv -y >> /root/deploy.log 2>&1; sleep 2

service mongodb start >> /root/deploy.log 2>&1; sleep 5

systemctl start mongodb.service >> /root/deploy.log 2>&1; sleep 5

systemctl status mongodb.service >> /root/deploy.log 2>&1; sleep 5

mongo tdmongodb --eval 'db.createCollection("tdmongodb")'

else

echo -e "\033[30;43mMongoDB is already Installed.....Memcached installation is going to skip and moving to Memcached\033[0m";

fi

This statement will check that mongodb installed correctly or not. This command will check that mongo is installed or not

mongp=$(which mongo | wc -l)

if [ "$mongp" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "MongoDB Verified";

mongo --version

echo "Now checking whether Memcached is installed or not";

else

echo -e "\033[30;5;101mMongoDB is not installed Sucessfully...Please check\033[0m";

fi

**#10 Memcache Installation:**

This statement will check that memcache installed or not otherwsie install. This command will check that memcache is installed or not

memp=$(which memcached | wc -l)

if [ "$memp" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Memcached Not found";

echo "Memcached Installation Starts please wait for few sec";

apt-get install memcached -y >> /root/deploy.log 2>&1; sleep 20

echo '############### Memcached Installation Finished ###############' >> /root/deploy.log;

systemctl start memcached.service /root/deploy.log 2>&1; sleep 20

systemctl enable memcached.service /root/deploy.log 2>&1; sleep 20

service memcached start

else

echo -e "\033[30;43mMemcached is already Installed.....Memcached installation is going to skip and moving to Redis\033[0m";

fi

This statement will check that memcache installed correctly or not. This command will check that memcache is installed or not

memp=$(which memcached | wc -l)

if [ "$memp" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Memcached Verified";

service memcached status

memcached -h |head -n1

echo "Now checking whether Redis is installed or not";

else

echo -e "\033[30;5;101mMemcached is not installed Sucessfully...Please check\033[0m";

fi

**#11 Redis Installation:**

This statement will check that redis installed or not otherwsie install. This command will check that redis is installed or not

redp=$(which redis-server | wc -l)

if [ "$redp" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Redis Not found";

echo "Redis Installation Starts please wait for few sec";

apt-get install php-redis redis-server -y >> /root/deploy.log 2>&1; sleep 20

echo '########### Redis Installation Finished ##############' >> /root/deploy.log;

systemctl start redis-server.service >> /root/deploy.log 2>&1; sleep 2

systemctl enable redis-server.service >> /root/deploy.log 2>&1; sleep 2

else

echo -e "\033[30;43mRedis is already Installed.....Redis installation is going to skip and moving to Node Js and Components\033[0m";

fi

This statement will check that redis installed correctly or not. This command will check that redis is installed or not

redp=$(which redis-server | wc -l)

if [ "$redp" = "1" ];

then

#SOME COMMANDS; sleep 600

echo "Redis Verified";

systemctl status redis-server.service

redis-server -v

echo "Now checking whether Node Js is installed or not";

else

echo -e "\033[30;5;101mRedis is not installed Sucessfully...Please check\033[0m";

fi

**#12 Node Js and Its Components Installation:**

This statement will check that node and its components installed or not otherwsie install. This command will check that node is installed or not

nodp=$(which node |wc -l)

file\_path=/root

if [ "$nodp" = "0" ];

then

#SOME COMMANDS; sleep 600

echo "Node Js Not found";

echo "Node Js v6.10.3 Installation Starts please wait for few sec";

apt-get update -y >> /root/deploy.log 2>&1; sleep 20

apt-get install build-essential libssl-dev python-software-properties openssl libssl-dev python phantomjs libreadline-dev chrpath libssl-dev -y >> /root/deploy.log 2>&1; sleep 50

apt install npm -y >> /root/deploy.log 2>&1; sleep 40

cd /root

curl -sL https://raw.githubusercontent.com/creationix/nvm/v0.33.1/install.sh -o install\_nvm.sh >> /root/deploy.log 2>&1; sleep 20

chmod +x /root/install\_nvm.sh

/root/install\_nvm.sh >> /root/deploy.log 2>&1; sleep 30

source /root/.bashrc

source ~/.nvm/nvm.sh

source ~/.profile

source ~/.bashrc

cd /root

wget --no-check-certificate https://nodejs.org/dist/v6.10.3/node-v6.10.3-linux-x64.tar.gz >> /root/deploy.log 2>&1; sleep 2

cd /usr/local

tar --strip-components 1 -xzf $file\_path/node-v6.10.3-linux-x64.tar.gz >> /root/deploy.log 2>&1; sleep 2

echo "############## node js 6.10.3 Installed ##############" >> /root/deploy.log;

cd /root

wget https://raw.githubusercontent.com/hassanhashmy/npm/master/npm\_install.sh >> /root/deploy.log 2>&1; sleep 5

chmod +x /root/npm\_install.sh

/root/npm\_install.sh >> /root/deploy.log 2>&1; sleep 30

source /root/.bashrc

else

echo -e "\033[30;43mNode is already Installed.....Node installation is going to skip and moving to Next Steps\033[0m";

fi

# This statement will check that node installed and print all version of all services

nodp=$(node --version |grep v |wc -l)

if [ "$nodp" = "1" ];

then

#SOME COMMANDS; sleep 600

apt install procps -y >> /root/deploy.log 2>&1

echo "Node Js Verified";

echo "Now checking all the services is installed";

echo "#1"; ps waux | grep nginx

echo "#2"; php -v

echo "#3"; php -m

echo "#4"; python2.7 -V

echo "#5"; git --version

echo "#6"; curl --version

echo "#7"; composer -V

echo "#8"; systemctl status mysql |grep mysql.service

echo "#9"; mongo --version

echo "#10"; memcached -h |head -n1

echo "#11"; redis-server -v

echo "#12"; node --version

echo "#13"; browser-sync --version

echo "#14"; browserify --version

echo "#15"; pm2 --version

echo "#16"; webpack --version

echo "#17"; npm -v

echo "#18"; gulp --version

echo "#19"; grunt --version

echo "#20"; bower --version

else

echo -e "\033[30;5;101mNode is not installed Sucessfully...Please check\033[0m";

fi

**#13 Server Connectivity Issue**

There was connectivity issue in server so that nothing was able to connect to internet. There were three issues on server.

1. DNS files was miss configured
2. DNS File hd attribute applied and due to this that file was not able to edit
3. TCP/80 was not working

All three issues were solved now the server is working fine as normal and now you can test the script from start or that would be good if you launch another instance and test the script.

**#14 Completion of Task / Achievement of Target / Analysis**

This script is tested 15 times on 15 different servers with fresh analysis and have successful test rate that it installs and configure all the things that mention as its on the start of this document.