|  |
| --- |
| #!/bin/bash  ################################################################################  # Author: Abdelmajid Elhamdaoui. Refrence: Yenthe Van Ginneken  ################################################################################  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Déclaration des variables\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ##fixed parameters  #odoo  OE\_USER="odoo"  OE\_HOME="/opt/$OE\_USER"  OE\_HOME\_EXT="/opt/$OE\_USER/odoo-server"  #The default port where this Odoo instance will run under (provided you use the command -c in the terminal)  #Set to true if you want to install it, false if you don't need it or have it already installed.  INSTALL\_WKHTMLTOPDF="True"  #Set to true if you want to install it, false if you don't need it or have it already installed.  INSTALL\_POSTGRESQL="True"  CREATE\_USER\_POSTGRESQL="True"  INSTALL\_NGINX = "True"  ADD\_SSL = "False"  SSL\_PEM\_KEY = "False"  SSL\_PRV\_KEY = "False"  #Set the default Odoo port (you still have to use -c /etc/odoo-server.conf for example to use this.)  OE\_PORT="8015"  SERVER\_NAME = "localhost" # test.odoo.com || 13.17.16.15  #Choose the Odoo version which you want to install. For example: 10.0, 9.0, 8.0, 7.0 or saas-6. When using 'trunk' the master version will be installed.  #IMPORTANT! This script contains extra libraries that are specifically needed for Odoo 10.0  OE\_VERSION="15.0"  # Set this to True if you want to install Odoo 10 Enterprise!  IS\_ENTERPRISE="True"  #set the superadmin password  OE\_SUPERADMIN="@8zz%&RBtAN78!Cms5@1"  OE\_CONFIG="${OE\_USER}-server"  #Set the database config  DB\_HOST="127.0.0.1"  DB\_PORT="5432"  DB\_USER=$OE\_USER  DB\_PASSWORD="LEGEND"  # OCA Modules  REP\_OCA\_WEB="https://github.com/OCA/web.git"  REP\_OCA\_SERVER\_TOOLS="https://github.com/OCA/server-tools.git"  REP\_OCA\_SERVER\_UX="https://github.com/OCA/server-ux.git"  REP\_OCA\_REPORT\_ENGINE="https://github.com/OCA/reporting-engine.git"  REP\_OCA\_ACC\_FIN\_TOOLS="https://github.com/OCA/account-financial-tools.git"  REP\_QUEUE="https://github.com/OCA/queue.git"  REP\_CUSTOM\_1="False"  REP\_CUSTOM\_1\_NAME=""  REP\_CUSTOM\_1\_BRANCH=$OE\_VERSION  ##  ###  WKHTMLTOPDF download links  ## === Ubuntu Trusty x64 & x32 === (for other distributions please replace these two links,  ## in order to have correct version of wkhtmltox installed, for a danger note refer to  ## https://www.odoo.com/documentation/8.0/setup/install.html#deb ):  WKHTMLTOX\_X64=https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox\_0.12.5-1.bionic\_amd64.deb  WKHTMLTOX\_X32=https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox\_0.12.5-1.bionic\_i386.deb  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mise à jour du serveur\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #--------------------------------------------------  # Update Server  #--------------------------------------------------  echo -e "\n---- Update Server ----" ***# Afficher "---- Update Server ----" sur une nouvelle ligne***  sudo apt-get update ***# Recherche les mises à jour disponibles pour le systèmes et les programmes installée***  sudo apt-get upgrade -y ***# Installer les mises à jour identifiées***  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Installation de PostgreSQL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #--------------------------------------------------  # Install PostgreSQL Server  #--------------------------------------------------  if [ $INSTALL\_POSTGRESQL = "True" ]; then ***# Si la valeur de la variable est égale à True on***  echo -e "\n---- Install PostgreSQL Server ----" ***# affiche le message sur une nouvelle ligne***  sudo apt-get install postgresql -y # ***puis on installe postgresql***  echo -e "\n---- Creating the ODOO PostgreSQL User  ----" ***# On affiche le message à la nouvelle ligne***  sudo su - postgres -c "createuser -s $OE\_USER" 2> /dev/null || true ***# Se connecter comme utilisateur postgres et créer l’utilisateur $OE\_USER***  else ***# Si la valeur variable vaut false***  sudo apt install postgresql-client-common ***# Installer postgresql-client-common***  sudo apt-get install -y postgresql-client ***# Installer postgresql-client***  echo -e "\n POSTGRESQL isn't installed due to the choice of the user! and no postgresql user have been created" ***# Afficher le message***  fi ***# fin de condition***  sudo apt-get install --reinstall libpq-dev ***# Réinstaller le package libpq-dev***  #psql -U postgres -c "ALTER USER $OE\_USER WITH PASSWORD '$DB\_PASSWORD'"  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Installation des dépendances\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #--------------------------------------------------  # Install Dependencies  #--------------------------------------------------  echo -e "\n---- Install/upgrade Python 3 Pip and other depends" ***# Affiche message***  sudo apt install git python3-pip build-essential wget python3-dev python3-venv python3-wheel libxslt-dev libzip-dev libldap2-dev libsasl2-dev python3-setuptools node-less -y ***# Installer ces dépendances***  sudo pip3 install --upgrade pip # Mettre à jour pip  sudo pip3 install Werkzeug==0.11.15 ***# Installer la version 0.11.15 de Werkzeug***  echo -e "\n---- Pip current version ---" && pip3 –version ***# Afficher version courante de pip3***  echo -e "\n---- Install tool packages ----" ***# Affiche message***  sudo apt-get install wget git python3-pip gdebi-core -y ***# Installer les différentes dépendances***  echo -e "\n---- Install python packages/librairies ----" ***# Affiche message***  sudo pip3 install Babel decorator docutils ebaysdk feedparser gevent html2text Jinja2 lxml Mako MarkupSafe mock num2words ofxparse passlib Pillow psutil psycogreen pydot pyparsing PyPDF2 pyserial python-dateutil python-openid pytz pyusb PyYAML qrcode reportlab requests six suds-jurko vatnumber vobject XlsxWriter xlwt xlrd gdata ***# Installation des éléments précités***  sudo pip3 install greenlet==1.1.0 ***# Installation version 1.1.0***  sudo pip3 install libsass==0.12.3 ***# Installation version 0.12.3***  sudo pip3 install Werkzeug==0.14.1 ***# Installation version 0.14.1***  echo -e "\n--- Install other required packages" ***# Message***  sudo apt-get install node-clean-css -y ***# Installer node-clean-css***  sudo apt-get install node-less -y ***# Installer node-less***  sudo apt-get install python3-gevent -y **# Installer python3-gevent**  sudo apt-get install python3-psycopg2 -y ***# Installer python3-gevent***  # after last update in Ubuntu 18.04 LTS  sudo pip3 install babel PyPDF2 passlib werkzeug lxml decorator Pillow psutil html2text docutils suds-jurko ***# Installation des éléments précités***  sudo pip3 install matplotlib ***# Installation matplotlib***  sudo apt-get install python3-reportlab ***# Installation python3-reportlab***  sudo apt-get install python3-dateutil python3-psycopg2 ***# Installation python3-dateutil python3-psycopg2***  #####  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Installation Wkhtmltopdf si besoin\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #--------------------------------------------------  # Install Wkhtmltopdf if needed  #--------------------------------------------------  if [ $INSTALL\_WKHTMLTOPDF = "True" ]; then ***# Si variable vaut true***    rm -rf /usr/bin/wkhtmltopdf ***# Suppression du dossier /usr/bin/wkhtmltopdf***    rm -rf /usr/bin/wkhtmltoimage ***# Suppression du dossier /usr/bin/wkhtmltoimage***    wget <http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.0g-2ubuntu4_amd64.deb> ***# Télécharger*** [***http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1\_1.1.0g-2ubuntu4\_amd64.deb***](http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.0g-2ubuntu4_amd64.deb)    sudo dpkg -i libssl1.1\_1.1.0g-2ubuntu4\_amd64.***deb # Installe le package libssl1.1\_1.1.0g-2ubuntu4\_amd64.deb***    sudo apt-get update -y ***# Recherche les mises à jour disponibles***    sudo apt-get install -y xfonts-base ***# Installer xfonts-base***    sudo apt-get install -y xfonts-75dpi ***# Installer xfonts-75dpi***    echo -e "\n---- Install wkhtml and place shortcuts on correct place for ODOO 10 ----" ***# Message***    #pick up correct one from x64 & x32 versions:    if [ "`getconf LONG\_BIT`" == "64" ]; then ***# Si la variable vaut “64”***        \_url=$WKHTMLTOX\_X64 ***# On affecte $WKHTMLTOX\_X64***    Else ***# Sinon***        \_url=$WKHTMLTOX\_X32 ***# On affecte $WKHTMLTOX\_X32***    Fi ***# Fin condition***    sudo wget $\_url ***# Téléchargement***    sudo dpkg -i `basename $\_url***`# Installer package***    sudo apt install -f    sudo ln -s /usr/local/bin/wkhtmltopdf /usr/bin ***# Créer un lien symbolique***    sudo ln -s /usr/local/bin/wkhtmltoimage /usr/bin ***# Créer un lien symbolique***  else ***# Sinon***    echo "Wkhtmltopdf isn't installed due to the choice of the user!" ***# Message***  fi ***# Fin condition***  echo -e "\n---- Create ODOO system user ----" ***# Message***  sudo adduser --system --quiet --shell=/bin/bash --home=$OE\_HOME --gecos 'ODOO' --group $OE\_USER  #The user should also be added to the sudo'ers group.  sudo adduser $OE\_USER sudo  echo -e "\n---- Create Log directory ----"  sudo mkdir /var/log/$OE\_USER  sudo chown $OE\_USER:$OE\_USER /var/log/$OE\_USER  #--------------------------------------------------  # Install ODOO  #--------------------------------------------------  echo -e "\n==== Installing ODOO Server ===="  sudo git clone --depth 1 --branch $OE\_VERSION https://www.github.com/odoo/odoo $OE\_HOME\_EXT/  # --- install requirements odoo 12  sudo pip3 install wheel  sudo pip3 install -r $OE\_HOME\_EXT/requirements.txt  if [ $IS\_ENTERPRISE = "True" ]; then      # Odoo Enterprise install!      echo -e "\n--- Create symlink for node"      sudo ln -s /usr/bin/nodejs /usr/bin/node      sudo su $OE\_USER -c "mkdir $OE\_HOME/enterprise"      sudo su $OE\_USER -c "mkdir $OE\_HOME/enterprise/addons"      GITHUB\_RESPONSE=$(sudo git clone --depth 1 --branch $OE\_VERSION https://www.github.com/odoo/enterprise "$OE\_HOME/enterprise/addons" 2>&1)      while [[ $GITHUB\_RESPONSE == \*"Authentication"\* ]]; do          echo "------------------------WARNING------------------------------"          echo "Your authentication with Github has failed! Please try again."          printf "In order to clone and install the Odoo enterprise version you \nneed to be an offical Odoo partner and you need access to\nhttp://github.com/odoo/enterprise.\n"          echo "TIP: Press ctrl+c to stop this script."          echo "-------------------------------------------------------------"          echo " "          GITHUB\_RESPONSE=$(sudo git clone --depth 1 --branch $OE\_VERSION https://www.github.com/odoo/enterprise "$OE\_HOME/enterprise/addons" 2>&1)      done      echo -e "\n---- Added Enterprise code under $OE\_HOME/enterprise/addons ----"      echo -e "\n---- Installing Enterprise specific libraries ----"      sudo apt-get install nodejs npm      sudo npm install -g less      sudo npm install -g less-plugin-clean-css  fi  echo -e "\n---------------------------OCA----------------------------"  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA"  if [ $REP\_OCA\_WEB != "False" ]; then  echo -e "\n==== Download OCA WEB ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/web"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_OCA\_WEB $OE\_HOME/OCA/web  fi  if [ $REP\_OCA\_SERVER\_TOOLS != "False" ]; then  echo -e "\n==== Download OCA Server-tools ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/server-tools"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_OCA\_SERVER\_TOOLS $OE\_HOME/OCA/server-tools  fi  if [ $REP\_OCA\_SERVER\_UX != "False" ]; then  echo -e "\n==== Download OCA SERVER-UX ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/server-ux"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_OCA\_SERVER\_UX $OE\_HOME/OCA/server-ux  fi  if [ $REP\_OCA\_REPORT\_ENGINE != "False" ]; then  echo -e "\n==== Download OCA Report-engine ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/report-engine"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_OCA\_REPORT\_ENGINE $OE\_HOME/OCA/report-engine  echo -e "\n==== Download OCA QUEUE ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/queue"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_QUEUE $OE\_HOME/OCA/queue  fi  if [ $REP\_OCA\_ACC\_FIN\_TOOLS != "False" ]; then  echo -e "\n==== Download OCA Report-engine ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/OCA/account-financial-tools"  sudo git clone --depth 1 --branch $OE\_VERSION $REP\_OCA\_ACC\_FIN\_TOOLS $OE\_HOME/OCA/account-financial-tools  fi  echo -e "\n---- Create custom module directory ----"  sudo su $OE\_USER -c "mkdir $OE\_HOME/custom"  sudo su $OE\_USER -c "mkdir $OE\_HOME/custom/addons"  if [ $REP\_CUSTOM\_1 != "False" ]; then  echo -e "\n==== Download REP\_CUSTOM\_1 custom ===="  sudo su $OE\_USER -c "mkdir $OE\_HOME/custom/$REP\_CUSTOM\_1\_NAME"  sudo git clone --depth 1 --branch $REP\_CUSTOM\_1\_BRANCH $REP\_CUSTOM\_1 $OE\_HOME/custom/$REP\_CUSTOM\_1\_NAME  fi  echo -e "\n---- Setting permissions on home folder ----"  sudo chown -R $OE\_USER:$OE\_USER $OE\_HOME/\*  echo -e "\* Create server config file"  sudo su root -c "echo '[options]' > /etc/${OE\_CONFIG}.conf"  sudo chown $OE\_USER:$OE\_USER /etc/${OE\_CONFIG}.conf  sudo chmod 640 /etc/${OE\_CONFIG}.conf  echo -e "\* Change server config file"  sudo su root -c "echo 'admin\_passwd = $OE\_SUPERADMIN' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo 'db\_host = $DB\_HOST' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo 'db\_port = $DB\_PORT' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo 'db\_user = $DB\_USER' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo 'db\_password = $DB\_PASSWORD' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo -n 'addons\_path = ' >> /etc/${OE\_CONFIG}.conf"  if [  $IS\_ENTERPRISE = "True" ]; then      sudo su root -c "echo -n '$OE\_HOME/enterprise/addons,$OE\_HOME\_EXT/addons,$OE\_HOME/custom/addons' >> /etc/${OE\_CONFIG}.conf"  else      sudo su root -c "echo -n '$OE\_HOME\_EXT/addons,$OE\_HOME/custom/addons' >> /etc/${OE\_CONFIG}.conf"  fi  if [ $REP\_OCA\_WEB != "False" ]; then  sudo su root -c "echo -n ',$OE\_HOME/OCA/web' >> /etc/${OE\_CONFIG}.conf"  fi  if [ $REP\_OCA\_SERVER\_TOOLS != "False" ]; then  sudo su root -c "echo -n ',$OE\_HOME/OCA/server-tools' >> /etc/${OE\_CONFIG}.conf"  fi  if [ $REP\_OCA\_SERVER\_UX != "False" ]; then  sudo su root -c "echo -n ',$OE\_HOME/OCA/server-ux' >> /etc/${OE\_CONFIG}.conf"  fi  if [ $REP\_OCA\_REPORT\_ENGINE != "False" ]; then  sudo su root -c "echo -n ',$OE\_HOME/OCA/report-engine' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo -n ',$OE\_HOME/OCA/queue' >> /etc/${OE\_CONFIG}.conf"  fi  if [ $REP\_OCA\_ACC\_FIN\_TOOLS != "False" ]; then  sudo su root -c "echo -n ',$OE\_HOME/OCA/account-financial-tools' >> /etc/${OE\_CONFIG}.conf"  fi  sudo su root -c "echo ' ' >> /etc/${OE\_CONFIG}.conf"  #logfile  sudo su root -c "echo 'logfile = /var/log/$OE\_USER/$OE\_CONFIG$1.log' >> /etc/${OE\_CONFIG}.conf"  sudo su root -c "echo 'logrotate = True' >> /etc/${OE\_CONFIG}.conf"  echo -e "\* Change default xmlrpc port"  sudo su root -c "echo 'xmlrpc\_port = $OE\_PORT' >> /etc/${OE\_CONFIG}.conf"  echo -e "\* Create startup file"  sudo su root -c "echo '#!/bin/sh' > $OE\_HOME\_EXT/start.sh"  sudo su root -c "echo 'sudo -u $OE\_USER $OE\_HOME\_EXT/odoo-bin --config=/etc/${OE\_CONFIG}.conf' >> $OE\_HOME\_EXT/start.sh"  sudo chmod 755 $OE\_HOME\_EXT/start.sh  #--------------------------------------------------  # Adding ODOO as a deamon (initscript)  #--------------------------------------------------  echo -e "\* Create init file"  cat <<EOF > ~/$OE\_CONFIG  #!/bin/sh  ### BEGIN INIT INFO  # Provides: $OE\_CONFIG  # Required-Start: \$remote\_fs \$syslog  # Required-Stop: \$remote\_fs \$syslog  # Should-Start: \$network  # Should-Stop: \$network  # Default-Start: 2 3 4 5  # Default-Stop: 0 1 6  # Short-Description: Enterprise Business Applications  # Description: ODOO Business Applications  ### END INIT INFO  PATH=/bin:/sbin:/usr/bin  DAEMON=$OE\_HOME\_EXT/odoo-bin  NAME=$OE\_CONFIG  DESC=$OE\_CONFIG  # Specify the user name (Default: odoo).  USER=$OE\_USER  # Specify an alternate config file (Default: /etc/openerp-server.conf).  CONFIGFILE="/etc/${OE\_CONFIG}.conf"  # pidfile  PIDFILE=/var/run/\${NAME}.pid  # Additional options that are passed to the Daemon.  DAEMON\_OPTS="-c \$CONFIGFILE"  [ -x \$DAEMON ] || exit 0  [ -f \$CONFIGFILE ] || exit 0  checkpid() {  [ -f \$PIDFILE ] || return 1  pid=\`cat \$PIDFILE\`  [ -d /proc/\$pid ] && return 0  return 1  }  case "\${1}" in  start)  echo -n "Starting \${DESC}: "  start-stop-daemon --start --quiet --pidfile \$PIDFILE \  --chuid \$USER --background --make-pidfile \  --exec \$DAEMON -- \$DAEMON\_OPTS  echo "\${NAME}."  ;;  stop)  echo -n "Stopping \${DESC}: "  start-stop-daemon --stop --quiet --pidfile \$PIDFILE \  --oknodo  echo "\${NAME}."  ;;  restart|force-reload)  echo -n "Restarting \${DESC}: "  start-stop-daemon --stop --quiet --pidfile \$PIDFILE \  --oknodo  sleep 1  start-stop-daemon --start --quiet --pidfile \$PIDFILE \  --chuid \$USER --background --make-pidfile \  --exec \$DAEMON -- \$DAEMON\_OPTS  echo "\${NAME}."  ;;  \*)  N=/etc/init.d/\$NAME  echo "Usage: \$NAME {start|stop|restart|force-reload}" >&2  exit 1  ;;  esac  exit 0  EOF  echo -e "\* Security Init File"  sudo mv ~/$OE\_CONFIG /etc/init.d/$OE\_CONFIG  sudo chmod 755 /etc/init.d/$OE\_CONFIG  sudo chown root: /etc/init.d/$OE\_CONFIG  = "upstream odoo {\n      server 127.0.0.1:8090;\n  }\n  server {\n      listen      80;\n      server\_name $SERVER\_NAME;\n      ssl on;      ssl\_certificate /etc/nginx/ssl/certificate.admin-serv.net.crt;      ssl\_certificate\_key     /etc/nginx/ssl/admin-serv.net.deprotected.key;      access\_log  /var/log/nginx/odoo.access.log;\n      error\_log   /var/log/nginx/odoo.error.log;\n      proxy\_buffers 16 64k;\n      proxy\_buffer\_size 128k;\n      location / {\n          proxy\_pass http://localhost:$OE\_PORT;\n          proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;\n          proxy\_redirect off;\n          proxy\_set\_header    Host            $host;\n          proxy\_set\_header    X-Real-IP       $remote\_addr;\n          proxy\_set\_header    X-Forwarded-For $proxy\_add\_x\_forwarded\_for;\n          proxy\_set\_header    X-Forwarded-Proto https;\n      }\n      location ~\* /web/static/ {\n          proxy\_cache\_valid 200 60m;\n          proxy\_buffering on;\n          expires 864000;\n          proxy\_pass http://localhost:$OE\_PORT;\n      }\n      location /longpolling {\n          proxy\_pass http://127.0.0.1:8072;\n      }\n  }\n  "  if [ $INSTALL\_NGINX = "True" ]; then  echo -e "\* Install, config Nginx and SSL"  sudo apt install nginx    if [ $ADD\_SSL = "True" ] && [ $SSL\_PEM\_KEY != "False" ] && [ $SSL\_PRV\_KEY != "False" ]; then  sudo su root -c "echo '$CONTENT\_NGINX' > /etc/nginx/sites-available/$OE\_USER"  sudo ln -s /etc/nginx/sites-available/$OE\_USER /etc/nginx/sites-enabled/$OE\_USER  sudo chown root:root /etc/nginx/sites-available/$OE\_USER  sudo chmod 775 /etc/nginx/sites-available/$OE\_USER    sudo chown root:root /etc/nginx/sites-enabled/$OE\_USER  sudo chmod 775 /etc/nginx/sites-enabled/$OE\_USER  fi    fi  echo -e "\* Start ODOO on Startup"  sudo update-rc.d $OE\_CONFIG defaults  echo -e "\* Starting Odoo Service"  sudo su root -c "/etc/init.d/$OE\_CONFIG start"  echo "-----------------------------------------------------------"  echo "Done! The Odoo server is up and running. Specifications:"  echo "Port: $OE\_PORT"  echo "User service: $OE\_USER"  echo "User PostgreSQL: $OE\_USER"  echo "Code location: $OE\_USER"  echo "Addons folder: $OE\_USER/$OE\_CONFIG/addons/"  echo "Start Odoo service: sudo service $OE\_CONFIG start"  echo "Stop Odoo service: sudo service $OE\_CONFIG stop"  echo "Restart Odoo service: sudo service $OE\_CONFIG restart" |
|  |