OTRS Install

REVISAR QUE LA SUSCRIPCION DE MICROSOFT SEA DISTINTO AL QUE SE HIZO POR PRIMERA VEZ, YA QUE LA CUENTA GENERAL SI FUNCIONA CON LA SUCRIPCION Pay-As-You-Go, TAMBIEN FIJATE EN LA RAM DE LA MAQUINA VIRTUAL YA QUE ESO SE TOMA EN CUENTA PARA QUE LA MAQUINA FUNCIONE

|  |  |
| --- | --- |
| vboxuser | @Changeme12345 |

vboxuser

@Changeme12345

sudo apt update

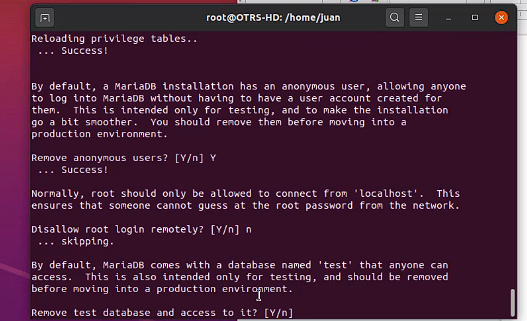
sudo apt upgrade -y

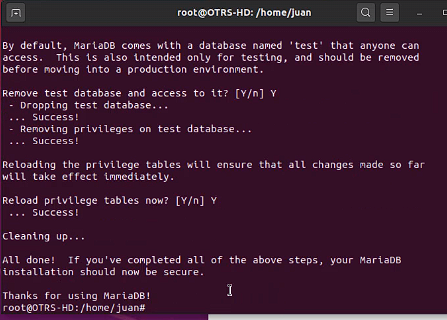
|  |  |
| --- | --- |
| sudo apt install mariadb-server | sudo apt-get install mariadb-server |

sudo mysql\_secure\_installation

Texto

Descripción generada automáticamente





**Ojo seguir los siguientes pasos Si el error es este: Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock':**

----------Eliminar archivos de bloqueo

-----Vamos a eliminar los archivos de bloqueo que puedan estar causando el problema.

sudo rm -f /var/lib/mysql/aria\_log\_control

sudo rm -f /var/lib/mysql/tc.log

sudo rm -f /var/lib/mysql/ib\_logfile\*

sudo rm -f /var/lib/mysql/ibdata1

----------Verificar permisos y propietarios

-----Asegúrate de que los archivos y directorios de MySQL tengan los permisos y propietarios correctos:

sudo chown -R mysql:mysql /var/lib/mysql

sudo chmod -R 755 /var/lib/mysql

----------Reiniciar MariaDB

sudo systemctl start mariadb

sudo systemctl status mariadb

esta parte es solo si ocurre error

sudo mysql -u root

UPDATE mysql.user SET plugin = 'passwd' WHERE User = 'root';

FLUSH PRIVILEGES;

QUIT;

mysql -u root -p

(<https://www.zeppelinux.es/corregir-error-1698-28000-access-denied-for-user-rootlocalhost-en-mariadb-mysql/>)

(en el video se usa despues de la configuración de maria este --)

sudo tee /etc/mysql/mariadb.conf.d/otrs.cnf<<EOF

[mysqld]

max\_allowed\_packet=64M

query\_cache\_size=36M

innodb\_log\_file\_size=256M

EOF

Despues de ingresar los datos en el SUDO previo se sigue con estos 2 pasos:

sudo systemctl restart mariadb

mysql -u root -p

SI REPORTA ESTE ERROR

$ mysql -u root -p

Enter password:

ERROR 1698 **(**28000**)**: Access denied **for** user 'root'**@**'localhost'

ENTONCES:

**sudo** mysql -u root -p

@Changeme12345

**USE** mysql;

**SELECT** user, plugin **FROM** user;

**UPDATE** user **SET** plugin="mysql**\_**native**\_**password" **WHERE** user="root";

**SELECT** user, plugin **FROM** user;

FLUSH **PRIVILEGES**;

exit;

mysql -u root -p

@Changeme12345

CREATE DATABASE otrs CHARACTER SET utf8 COLLATE utf8\_general\_ci;

GRANT ALL PRIVILEGES ON otrs.\* TO otrs\_user@localhost IDENTIFIED BY 'passwd';

FLUSH PRIVILEGES;

QUIT

sudo apt install apache2 libapache2-mod-perl2

sudo apt install libdatetime-perl libcrypt-eksblowfish-perl libcrypt-ssleay-perl libgd-graph-perl libapache-dbi-perl libsoap-lite-perl libarchive-zip-perl libgd-text-perl libnet-dns-perl libpdf-api2-perl libauthen-ntlm-perl libdbd-odbc-perl libjson-xs-perl libyaml-libyaml-perl libxml-libxml-perl libencode-hanextra-perl libxml-libxslt-perl libpdf-api2-simple-perl libmail-imapclient-perl libtemplate-perl libtext-csv-xs-perl libdbd-pg-perl libapache2-mod-perl2 libtemplate-perl libnet-dns-perl libnet-ldap-perl libio-socket-ssl-perl libmoo-perl

sudo apachectl -M | grep perl

sudo a2enmod perl

sudo systemctl restart apache2

sudo systemctl status apache2

(PULSAR Q PARA SALIR)

sudo useradd -d /opt/otrs -c 'OTRS user' otrs

sudo usermod -aG www-data otrs

id otrs

sudo apt-get install wget

wget https://download.znuny.org/releases/otrs-latest.tar.gz

!!! NO USAR ESTE ENLACE ES ANTIGUO!!! wget <http://ftp.otrs.org/pub/otrs/otrs-latest.tar.gz>

tar xvf otrs-latest.tar.gz

mv otrs-\*/ otrs

sudo mv otrs /opt

sudo /opt/otrs/bin/otrs.CheckModules.pl

sudo cp /opt/otrs/Kernel/Config.pm.dist /opt/otrs/Kernel/Config.pm

sudo nano /opt/otrs/Kernel/Config.pm

!!!LA CONTRASEÑA ES: passwd!!! (CTRL O: PARA GUARDAR Y LUEGO PULSA ENTER, CTRL X: PARA SALIR)

Texto

Descripción generada automáticamente

sudo nano /opt/otrs/scripts/apache2-perl-startup.pl

Texto

Descripción generada automáticamente

Quitar comentarios

use DBD::mysql ();

use Kernel::System::DB::mysql;

sudo /opt/otrs/bin/otrs.SetPermissions.pl –web-group=www-data

ls -ldh /opt/otrs/

sudo ln -s /opt/otrs/scripts/apache2-httpd.include.conf /etc/apache2/sites-enabled/otrs.conf

sudo perl -cw /opt/otrs/bin/cgi-bin/index.pl

sudo perl -cw /opt/otrs/bin/cgi-bin/customer.pl

sudo perl -cw /opt/otrs/bin/otrs.Console.pl

sudo systemctl restart apache2

<http://serveripaddress/otrs/installer.pl>

<http://20.0.0.1/otrs/installer.pl>

sudo ufw allow 80

Open the URL <http://server-ip-address/otrs/installer.pl>

[http://UUURRRRLLLL/otrs/index.pl](http://UUURRRRLLLL/otrs/index.pl%20)

<http://127.0.0.1/otrs/index.pl>

root@localhost

DlTWLBtUMTTz1t4H

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

otrs\_user

passwd

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

NUEVO.nhphswljcemetpnlx0ebows0ag.bx.internal.cloudapp.net

[support@yourhost.example.com](mailto:support@yourhost.example.com)

KONECTA

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Página de inicio:

<http://172.172.152.47/otrs/index.pl>

Usuario:

root@localhost

Contraseña:

veZ9cuo9MrI7fU1s

CONFIGURACION DE RED

**sudo** mysql -u root -p

GRANT ALL PRIVILEGES ON \*.\* TO 'otrs\_user'@'%' IDENTIFIED BY 'passwd' WITH GRANT OPTION;

FLUSH PRIVILEGES;

sudo apt install python3-pip

sudo nano /etc/mysql/mariadb.cnf

[mysqld]

bind-address = 0.0.0.0

Interfaz de usuario gráfica

Descripción generada automáticamente

Texto

Descripción generada automáticamente

sudo service mysql restart

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

!pip install mysql-connector-python

import time

import mysql.connector

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

import joblib

# Inicia el cronómetro para el script completo

start\_time\_total = time.time()

# Configura los datos de conexión a la base de datos OTRS

config = {

    'user': 'otrs\_user',

    'password': 'passwd',

    'host': '20.102.112.53',

    'database': 'otrs',

    'port': 3306,

}

try:

    # Crea una conexión a la base de datos OTRS

    connection = mysql.connector.connect(\*\*config)

    # Comprueba si la conexión se ha establecido

    if connection.is\_connected():

        print("Conexión establecida a la base de datos OTRS")

        # Consulta SQL para extraer los datos de Machine Learning incluyendo el ID del ticket

        cursor = connection.cursor(dictionary=True)

        cursor.execute(

            "SELECT ticket.id AS ticket\_id, REPLACE(article\_data\_mime.a\_body, '\\n', ' ') AS article\_body "

            "FROM ticket "

            "JOIN article ON ticket.id = article.ticket\_id "

            "JOIN article\_data\_mime ON article.id = article\_data\_mime.article\_id "

            "WHERE ticket.queue\_id = (SELECT id FROM queue WHERE name = 'machine learning')"

        )

        data = cursor.fetchall()

        df\_machine\_learning = pd.DataFrame(data)

        # Cargar el modelo entrenado y el vectorizador desde DBFS

        clf = joblib.load('/dbfs/models/modelo\_clasificacion.pkl')

        vectorizer = joblib.load('/dbfs/models/vectorizador\_tfidf.pkl')

        # Inicia el cronómetro para la predicción

        start\_pred = time.time()

        # Transformar y realizar predicciones

        X\_machine\_learning\_tfidf = vectorizer.transform(df\_machine\_learning['article\_body'])

        predicciones = clf.predict(X\_machine\_learning\_tfidf)

        # Finaliza el cronómetro para la predicción

        end\_pred = time.time()

        # Imprime el tiempo de predicción

        print(f"Tiempo de predicción: {end\_pred - start\_pred} segundos")

        # Inicializa un contador

        contador\_predicciones = 0

        # Inicia el cronómetro para la actualización de los tickets

        start\_update = time.time()

        # Actualizar el área de los tickets en la base de datos

        for i in range(len(predicciones)):

            ticket\_id = df\_machine\_learning['ticket\_id'][i]

            nueva\_area = predicciones[i]

            update\_query = (

                f"UPDATE ticket "

                f"SET queue\_id = (SELECT id FROM queue WHERE name = '{nueva\_area}') "

                f"WHERE id = {ticket\_id}"

            )

            cursor.execute(update\_query)

            connection.commit()

            contador\_predicciones += 1

            print(f"Área actualizada para el ticket machine learning con ID {ticket\_id} a '{nueva\_area}'")

        # Finaliza el cronómetro para la actualización

        end\_update = time.time()

        # Imprime el tiempo de actualización

        print(f"Tiempo de actualización de los tickets: {end\_update - start\_update} segundos")

        # Imprime el número total de predicciones realizadas

        print(f"Total de predicciones realizadas: {contador\_predicciones}")

except mysql.connector.Error as error:

    print(f"Error al conectar a la base de datos: {error}")

finally:

    # Cierra la conexión cuando hayas terminado de usarla

    if connection.is\_connected():

        connection.close()

        print("Conexión cerrada")

# Finaliza el cronómetro para el script completo

end\_time\_total = time.time()

# Imprime la duración total del script

print(f"Duración total del script: {end\_time\_total - start\_time\_total} segundos")