========================================================================================================================================================

INSTALACIÓN DE UBUNTU 22 CON DOCKER

========================================================================================================================================================

Servidor>>

IP:

IPV6:

Usuario: root

Senha:

Frente = dirección principal del chatbot: [[app.dominio.com]]

Backend/API = dirección de api interna: [[api.dominio.com]]

Aplicación: CeltaInfo [[Nombre de la aplicación, nombre corto - sin letras mayúsculas ni caracteres especiales]]

Implementar contraseña: [[usuario implementar contraseña, solo letras y números, sin letras mayúsculas ni caracteres especiales; se recomiendan 8 caracteres]]

Usuario superadministrador: admin@admin.com

Senha: 123456

Nunca elimine este usuario, y al editar los datos no olvide completar la contraseña actual o una nueva contraseña.

Github - Código

https://github.com/launcherbr/whaticketsaasfree.git

========================================================================================================================================================

ACTUALIZA TU VPS

sudo apt update && sudo apt -y upgrade

INSTALAR NODE Y POSTGRES

sudo su root

curl -fsSL https://deb.nodesource.com/setup\_20.x | sudo -E bash -

apt-get install -y nodejs

npm install -g npm@latest

sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg

sudo apt update -y

sudo apt-get update -y && sudo apt-get -y install postgresql-16

sudo timedatectl set-timezone America/Sao\_Paulo

INTALAR DE DOCKER

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal estable"

política de apt-cache docker-ce

sudo apt install docker-ce

INSTALAR TITIRITERO

apt-get install -y libxshmfence-dev libgbm-dev wget descomprimir fontconfig locales gconf-service libasound2 libatk1.0-0 libc6 libcairo2 libcups2 libdbus-1-3 libexpat1 libfontconfig1 libgcc1 libgconf-2-4 libgdk-pixbuf2.0-0 libglib2.0-0 libgtk-3-3-20 libnspr4 libpango-1.0-0 libpangocairo-1.0-0 libstdc++6 libx11-6 libx11-xcb1 libxcb1 libxcomposite1 libxcursor1 libxdamage1 libxext6 libxfixes3 libxi6 libxrandr2 libxrender1 libxss1 libxtst6 ca-certificates fonts-liberation libappindicator1 libnss3 lsb-release xdg-utils

INSTALAR PM2

npm install -g pm2

INSTALAR SNAPD

apt install -y snapd

Núcleo de instalación instantánea

Núcleo de actualización instantánea

INSTALAR O NGINX

apt install -y nginx

rm /etc/nginx/sites-enabled/default

Reinicio del servicio nginx

sudo nano /etc/nginx/nginx.conf

client\_max\_body\_size 100 metros;

underscores\_in\_headers en;

INSTALAR CERTBOT

apt-get eliminar certbot

Instalación rápida --Certbot clásico

ln -s /snap/bin/certbot /usr/bin/certbot

CRIAR USUARIO DEPLOY =

Implementación de adduser

UserMod -AG Sudo Deploy

Rellene solo la contraseña, introduzca los campos de datos personales.

==============================================================================================================================================

INSTALAR USANDO EL INSTALADOR

========================================================================================================================================================

cd /inicio

ls

sudo apt install -y git && git clone https://github.com/launcherbr/instalador.git instalador && sudo chmod -R 777 instalador && cd instalador && sudo ./install\_primaria

========================================================================================================================================================

Siga las instrucciones paso a paso del terminal que se muestra - Tenga cuidado al copiar y pegar para no tener espacios al final de la expresión:

[0] Instalar WhatsPainel/Whaticket

Introduzca la contraseña para el usuario de implementación y la base de datos (no utilice caracteres especiales ni letras mayúsculas)

[[Senhadeploy]]

Introduce el enlace de GITHUB de tu Whaticket que quieres instalar:

https://github.com/launcherbr/whaticketsaasfree.git

Introduzca un nombre para la Instancia/Empresa que se instalará (No utilice espacios, caracteres especiales ni letras mayúsculas, utilice solo letras minúsculas);

[[nomedoapp]]

Informar al Qt. de Conexiones/What's que puede registrarlo: 1 a 9999

9999

Informar a los Usuarios/Asistentes Cantidad que puede registrarlo: del 1 al 9999

9999

Introduzca el dominio del FRONTEND/PANEL; Ej: appbot.cloud

[[app.dominio.com]]

Introduzca el dominio del BACKEND/API; Ej: api.appbot.cloud

[[api.dominio.com]]

Introduzca el puerto FRONTEND para el appbot; Ej: 3000 A 3999

3000

Introduzca el puerto BACKEND para esta instancia; Ej: 4000 A 4999

4000

Introduzca el puerto de la PROGRAMACIÓN REDIS/MSG al appbot; Ej: 5000 A 5999

5000

Compruebe si es necesario liberar los puertos del firewall, por lo general, la propia instalación libera los puertos.

========================================================================================================================================================

========================================================================================================================================================

========================================================================================================================================================

INSTALACIÓN MANUAL

========================================================================================================================================================

BACKEND

========================================================================================================================================================

CREACIÓN DE REDIS Y BANCO DE POSTGRES

La contraseña del banco está entre '' (comillas simples)

sudo su root

usermod -aG docker deploy

docker run --name redis-[[nomeredis]] -p 5000:6379 --restart always --detach redis redis-server --requirepass [[senharedis]]

sudo su - postgres

createdb [[nomebanco]];

psql

CREATE USER [[USERBANCO]] SUPERUSER HERIT CREATEDB CREATEROLE;

ALTER USER [[userbanco]] PASSWORD '[[senhauser]]';

\q

salida

========================================================================================================================================================

CREAR VARIABLE DE ENTORNO

---- Uso de la implementación de usuarios en el directorio de back-end ----

cd /home/deploy/[[nome do app]]/backend

ls

sudo nano .env

------------ Edite el archivo con los datos a continuación --------------

https://github.com/launcherbr/saascodigo/edit/master/backend/.env.example

NODE\_ENV=

BACKEND\_URL=https://[[dominio do backend]]

FRONTEND\_URL=https://[[dominio do frontend]]

PROXY\_PORT=443

PUERTO=4000

DB\_DIALECT=postgres

DB\_HOST=anfitrión local

DB\_TIMEZONE=-03:00

DB\_PORT=5432

DB\_USER=[[userbanco]]

DB\_PASS=[[senhauser]]

DB\_NAME=[[nombre del banco]]

rootPath=/var/www/backend

JWT\_SECRET=kZaOTd+YZpjRUyyuQUpigJaEMk4vcW4YOymKPZX0Ts8=

JWT\_REFRESH\_SECRET=dBSXqFg9TaNUEDXVp6fhMTRLBysP+j2DSqf7+raxD3A=

REDIS\_URI=redis://[[senharedis]]@127.0.0.1:5000

REDIS\_OPT\_LIMITER\_MAX=1

REDIS\_OPT\_LIMITER\_DURATION=3000

USER\_LIMIT=10000

CONNECTIONS\_LIMIT=100000

CLOSED\_SEND\_BY\_ME=verdadero

========================================================================================================================================================

INSTALACIÓN DEL LOCAL

---- Uso de la implementación de usuarios en el directorio de back-end ----

cd /home/deploy/[[nome do app]]/backend

Instalación de npm

========================================================================================================================================================

COMPILAR EL CÓDIGO DE BACKEND

---- Uso de la implementación de usuarios en el directorio de back-end ----

cd /home/deploy/[[nome do app]]/backend

Compilación de ejecución de npm

========================================================================================================================================================

CREACIÓN DE LAS TABLAS EN LA BASE DE DATOS

----Uso de la implementación de usuario en el directorio de backend----

cd /home/deploy/[[nome do app]]/backend

npx sequelize db:migrate

ABARROTANDO LAS MESAS DE LOS BANCOS

----Uso de la implementación de usuario en el directorio de backend----

cd /home/deploy/[[nome do app]]/backend

npx sequelize db:seed:all

INICIO DEL SERVICIO PM2

----Uso de la implementación de usuario en el directorio de backend----

cd /home/deploy/[[nome do app]]/backend

pm2 start dist/server.js --name [[nome do app]]-backend

Guardar PM2 --force

CONFIGURACIÓN DE NGINX

----Usando la raíz de usuario en el directorio, etc----

sudo nano /etc/nginx/sites-available/[[nome do app]]-backend

servidor {

server\_name [[Dominio al backend]];

Ubicación / {

proxy\_pass http://127.0.0.1:4000;

proxy\_http\_version 1.1;

proxy\_set\_header Actualización \$http\_actualización;

proxy\_set\_header "Actualización" de la conexión;

proxy\_set\_header Host \$host;

proxy\_set\_header X-Real-IP \$remote\_addr;

proxy\_set\_header X-forwarded-Proto \$scheme;

proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for;

proxy\_cache\_bypass \$http\_actualización;

}

}

---- Generar vínculos simbólicos ----

ln -s /etc/nginx/sites-available/[[nome do app]]-backend /etc/nginx/sites-enabled

========================================================================================================================================================

FRONTEND

========================================================================================================================================================

CREAR VARIABLE DE ENTORNO

---- Uso de la implementación de usuario en el directorio frontend----

cd /home/deploy/[[nome do app]]/frontend

sudo nano .env

REACT\_APP\_BACKEND\_URL=https://[[api.dominio.com]]

REACT\_APP\_HOURS\_CLOSE\_TICKETS\_AUTO = 24

INSTALANDO DEPENDENCIAS DEL FRONTEND

---- Uso de la implementación de usuarios en el directorio front-end ----

cd /home/deploy/[[nome do app]]/frontend

Instalación de npm

COMPILANDO EL CÓDIGO DE LA INTERFAZ

---- Uso de la implementación de usuarios en el directorio front-end ----

cd /home/deploy/[[nome do app]]/frontend

Compilación de ejecución de npm

INICIO PM2

---- Uso de la implementación de usuarios en el directorio front-end ----

cd /home/deploy/[[nome do app]]/frontend

pm2 start server.js --name [[nome do app]]-frontend

Guardar PM2 --force

----Configuración para PM2 para iniciar con Ubuntu----

Puesta en marcha de PM2

sudo env PATH=$PATH:/usr/bin /usr/lib/node\_modules/pm2/bin/pm2 startup systemd -u deploy --hp /home/deploy

CONFIGURACIÓN DE NGINX

---- Uso de la raíz de usuario en el directorio, etc. ----

sudo nano /etc/nginx/sites-available/[[nome do app]]-frontend

servidor {

server\_name [[Dominio do frontend]];

Ubicación / {

proxy\_pass http://127.0.0.1:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Actualización \$http\_actualización;

proxy\_set\_header "Actualización" de la conexión;

proxy\_set\_header Host \$host;

proxy\_set\_header X-Real-IP \$remote\_addr;

proxy\_set\_header X-forwarded-Proto \$scheme;

proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for;

proxy\_cache\_bypass \$http\_actualización;

}

}

---- Generar vínculos simbólicos ----

sudo ln -s /etc/nginx/sites-available/[[nome do app]]-frontend /etc/nginx/sites-enabled

EJECUCIÓN DE CERTBOT PARA GENERAR SSL

certbot -m seuemail@gmail.com --nginx --agree-tos --non-interactive --domains [[dominio do frontend]],[[dominio do backend]]

========================================================================================================================================================

========================================================================================================================================================

========================================================================================================================================================

\* Corrección, si el estado de PM2 no enumera las aplicaciones después de un reinicio:

sudo su - deploy

cd /home/deploy/[[nome do app]]/frontend

sudo PM2 start server.js --name [[nombre de la aplicación]]-frontend

sudo pm2 guardar --force

sudo su - deploy

cd /home/deploy/[[nome do app]]/backend

sudo pm2 start dist/server.js --name [[nombre de la aplicación]]-backend

sudo pm2 guardar --force

\* caso guardado arriba, y no aparece después del nuevo reinicio, reintente e incluya estos comandos:

sudo su - root

Puesta en marcha de PM2

sudo env PATH=$PATH:/usr/bin /usr/lib/node\_modules/pm2/bin/pm2 startup systemd -u deploy --hp /home/deploy

========================================================================================================================================================

Inicia sesión:

Usuario estándar

admin@admin.com

123456

Forma. Registro

/registro

INFORMACIÓN IMPORTANTE:

No elimine el usuario superadministrador, id 1: al cambiar el correo electrónico o el nombre, no olvide completar el campo de contraseña, de lo contrario, guardará la contraseña en blanco en la base de datos.

No elimine el ID de la empresa principal 1, realice el cambio de datos.

Ve a Configuración y empresas para habilitar el envío de campañas. Después de la activación, cierre la sesión e inicie sesión para que aparezca el menú.

El plan 1 se puede eliminar, no acepta cambios de nombre, pero al eliminarlo vincule un nuevo plan a la empresa 1 si no podrá agregar conexiones, colas y nuevos usuarios.

En nuestra área de base de conocimiento tienes las instrucciones que hice sobre el chatbot, en Google Drive en el archivo de texto hay una recopilación de videos de YouTube de Whaticket en general tanto gratuitos como SAAS.

En el tipo de chatbot, solo texto. WhatsApp no permite el funcionamiento de botones y listas más que a través de la api oficial de Whatsapp Business, que es de pago.

Al cambiar la gestión de tiempos, vacíe los campos rellenados de la opción actual, ejemplo al cambiar de una gestión de tiempos por cola a por empresa, vaya a los tiempos de cola y deje los tiempos en blanco (para que la información de tiempo se borre de la base de datos), haga lo mismo si va a cambiar una gestión de tiempo de empresa a cola, Desactive los campos de tiempo de la empresa o si deshabilita el tiempo borrando los campos activos actualmente.

Medios vacíos con más de 30 días de antigüedad:

find /home/deploy/\*/backend/public -type f -mtime +30 -delete

========================================================================================================================================================

Comando para compilar/reconstruir

sudo npm run build

========================================================================================================================================================

Comando para actualizar bibliotecas

Actualización de npm --force

========================================================================================================================================================

Comando para reiniciar pm2

PM2 Reiniciar todo

========================================================================================================================================================

Gerencianet, actualmente Efi

Para la devolución automática se necesita una aplicación en PC, Insomnia o usar Postman:

backend / .env

Agregar estas líneas

GERENCIANET\_SANDBOX=falso

GERENCIANET\_CLIENT\_ID=Client\_Id\_Gerencianet

GERENCIANET\_CLIENT\_SECRET=Client\_Secret\_Gerencianet

GERENCIANET\_PIX\_CERT=certificado-Gerencianet

GERENCIANET\_PIX\_KEY=Clave Pix Gerencianet

En backend\certs

Salvar o certificado no formato .p12

Actualizar

./actualizar

(dentro del directorio de la aplicación, desde el permiso 777 antes a través del terminal ssh con el comando chmod 777 update)

========================================================================================================================================================

Ayuda (múltiples canales)

\*no son específicos de este código

https://help.whaticket.com/pt-br

https://whaticket.online/

https://www.youtube.com/@whaticketapp

https://www.youtube.com/@multiconversaoficial

https://www.youtube.com/@melissatreinamentos

https://www.youtube.com/@astraonlineweb

https://www.youtube.com/@equipechat

https://blog.melissatreinamentos.tec.br/

========================================================================================================================================================

========================================================================================================================================================