Índice

01. Información importante

02. Enlace a GitHub

03. Datos del servidor y de la instalación

04. Actualizar el servidor y reiniciar - Paso 01

05. Instalador de Whaticket - Paso 02

06. Proceso de instalación en el terminal SSH – Paso 03

07. Arreglar nginx.conf

08. Arreglar Baileys, reinicio programado de aplicaciones de nodo

09. Comandos de nodo/pm2

10. Detalles de VPSDime: realice este procedimiento antes de la instalación.

11. Habilite Efi, personalización y pruebas de Redis

12. Canales de ayuda

13. Instalación con Docker / Ubuntu 22

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

01. Información importante

Utilizar Ubuntu 20.04

Mínimo requerido: 4 vCPU y 6 GB de RAM

Recomendado: 4 vcpu y 8 gb de RAM

Sugerencia de VPS, Estándar.

https://control.peramix.com/?affid=14

Solo se instala cuando las direcciones de frontend y backend hacen ping a la IP del servidor.

Compruebe https://dnschecker.org/ si se propaga en todas las regiones.

O a través del símbolo del sistema de Windows\* Tecla de Windows + R = CMD y comandos

Ping seudominio.com.br

No use caracteres especiales ni letras mayúsculas en el proceso de instalación, no use números en el nombre de la aplicación.

Compruebe si su proveedor de servicios VPS tiene algún cortafuegos y la necesidad de vaciar los puertos en ellos.

Los puertos en uso son 22, 80, 443, 3000, 4000, 5000, 5432 y 6379.

Si su servidor no accede directamente con el usuario root, habilite el privilegio con el comando sudo su, o sudo passwd root (establezca una contraseña de root), sudo su e inicie sesión con la contraseña que registró.

Nunca elimine el usuario administrador, id 1 y la empresa creada en la instalación, al editar los datos del usuario superadministrador no olvide completar la contraseña actual o una nueva contraseña, de lo contrario, guarda la contraseña en blanco en la base 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 eliminar vincular un nuevo plan a la empresa 1, si no, no podrá agregar conexiones, colas y nuevos usuarios.

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.

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

02 . Link do GitHub

\* Github - Instalador

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

\*Código

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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

03. Datos de instalación

Reemplace [[]] con sus datos personalizados.

Servidor:

IPV4: [[1.1.1.1]]

IPV6:

Contraseña: [[contraseña]]

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

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

Aplicación: [[appdoname]]

Contraseña de implementación: [[alphanumeric-password]]

Usuario superadministrador: admin@admin.com

Contraseña: 123456

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

04. Paso 01 - Actualizar el servidor

Actualice los paquetes del servidor Ubuntu - usando el privilegio de root:

sudo apt -y update && apt -y upgrade

Reinicie el servidor para completar las actualizaciones:

reiniciar

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

05. Paso 02: descargue el instalador de github

\*\* Ejecute estos comandos en su cliente ssh (recomendado por bitvise)

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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

06. Paso 03 - Ejecute la configuración

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: app.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 = redis por defecto es 6379 / en la instalación con docker use 5000

5000

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

07. Arreglar nginx.conf

Corregir errores de actualización de página.

Edite el archivo de configuración de nginx:

nano /etc/nginx/nginx.conf

Agregue el encabezado a continuación: (inmediatamente encima de #server\_tokens off;)

underscores\_in\_headers en;

+ Ctrl y X para cerrar el terminal, e Y para guardar los cambios.

Pruebe los cambios de nginx:

nginx -t

Reinicie el servicio:

Servicio de recarga de nginx

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

08. Arreglar Baileys, reinicio programado de aplicaciones de nodo.

Utilizamos las tareas programadas, cron jobs, para programar el comando pm2 restart all:

crontab -e

Si tiene varios editores de texto instalados, el sistema le pedirá que seleccione un editor para actualizar la lista de tareas cron. Use el número entre paréntesis para elegir su opción preferida. Usaremos la opción predeterminada, nano. (Elección numérica).

En el modelo de abajo, cada 12 horas tendremos un reinicio de reinicio de frontend y reinicio de backend.

0 \*/6 \* \* \* /usr/bin/node /usr/bin/pm2 reiniciar todo

Para conocer otras opciones de modelos cron, visite uno de los siguientes sitios:

https://crontab.guru/

https://crontab-generator.org/

https://crontab.cronhub.io/

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

09. Comandos de nodo/pm2

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

Comando para construir/reconstruir:

sudo npm run build

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

Comando para actualizar bibliotecas;

Actualización de npm --force

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

Comando para reiniciar pm2:

PM2 Reiniciar todo

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

Registro:

Registro de PM2

Limpieza de registros:

PM2 Empotrar "ID"

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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

10. Particularidades de VPSDime: realice este procedimiento antes de la instalación.

Actualice los paquetes de su servidor:

sudo apt -y update && apt -y upgrade

El firewall sin complicaciones (UFW) no está instalado

Instale UFW, habilite y libere los puertos, tenga cuidado de liberar el puerto SSH para no perder el acceso al servidor.

sudo apt-get install ufw

sudo ufw enable

Libere los siguientes puertos: sudo ufw allow <port>/<protocol> =

sudo ufw allow 22/tcp

sudo ufw allow 80/tcp

sudo ufw permitir 443/tcp

sudo ufw allow 3000/tcp

sudo ufw permitir 4000/tcp

sudo ufw permitir 5000/tcp

sudo ufw allow 5432/tcp

sudo ufw allow 6379/tcp

Compruebe si los puertos están limpios:

Estado de sudo ufw

Instalar Curl:

sudo apt-get install curl

Establezca la versión de Node.js instalación en 20, de lo contrario, instalará la versión 10.

Ejecute este comando para instalar la versión correcta:

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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

11. Habilite Efi, personalización y pruebas de Redis

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)

Fazer o WebHook:

Revisa el pdf en Google Drive.

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

Cambiar nombre, color primario, logotipo y favicon:

Realice modificaciones mediante el usuario de implementación, no mediante la raíz.

\*\* Cambiar color primario: (#2DDD7F)

/frontend/src/App.js

/frontend/src/layout/index.js

\*\* Colores del chat interno: (validar)

/frontend\src\pages\Chat\ChatMessages.js

frontend\src\pages\Chat\ChatList.js

\*\* Colores de la lista de tareas (validar)

/frontend/src/pages/ToDoList/index.js

\*\* Logotipo e íconos de favicon:

/frontend/src/activos

/frontend/público/

/backend/public/logotipos

\*\* Empresa / copyright

/frontal/package.json

\*\* Comando para reconstruir, ruta absoluta /home/deploy/"name"/

cd /frontend npm run build

\*\* Medios vacíos con más de 30 días de antigüedad:

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

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

\*\* Habilitar el envío de la 2ª Copia de los Comprobantes de Asaas (No confundir con la facturación del Plan Saas):

Obtenga su clave api en el sitio web de Asaas e ingrésela en el campo de configuración;

Crear una cola de servicio:

Nombre: 2ª Copia del Boleto

Mensaje de saludo:

Por favor, proporciónenos el CPF/CNPJ para consultarlo en nuestra base de datos

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

\*\* Acceso a PostgreSQL:

Utilice PGAdmin4, con las credenciales de la base de datos guardadas en el archivo .env del backend y los datos de acceso raíz, IP y contraseña como túnel.

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

Redis - Base de datos de programación

A continuación se muestra una secuencia de comandos que se utilizan para probar si la contraseña de Redis funciona. El primer comando intenta establecer una clave en un valor antes de la autenticación:

Pruebas (con Docker)

Docker PS

docker exec -it CONTAINER ID bash

redis-cli

Auth sua\_senha

(Sin Docker)

redis-cli

Establecer tecla1 10

Esto no funcionará porque no se ha autenticado, por lo que Redis devuelve un error:

(error) Se requiere autenticación NOAUTH.

El siguiente comando se autentica con la contraseña especificada en el archivo de configuración de Redis:

Auth sua\_senha

Redis reconoce:

Señal

Reposta: Pong

Después de confirmar que puede ejecutar comandos en el cliente Redis después de la autenticación, puede salir de redis-cli:

renunciar

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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

12. Canales de ayuda

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/

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

13. Instalación en Ubuntu 22

No se puede crear el usuario de implementación, cree manualmente antes de la instalación.

Ve al terminal SSH y actualiza tu servidor:

sudo apt -y update && apt -y upgrade

Agregue manualmente el usuario de implementación:

Implementación de adduser

\* Ingrese la contraseña

\* Repetir contraseña

\* Entrar para los campos de datos personales, no rellenar.

UserMod -AG Sudo Deploy

o

adduser deploy sudo

Descargue 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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Prueba de Redis/Docker:

Docker PS

docker exec -it redis-[[nombre de la aplicación]] bash

redis-cli

Auth [[Senhadeploy]]

Señal

Reposta: Pong

\* 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

|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||