

Chatwoot

Aplicativo de chatbots

- [Instalação e configurações do Chatwoot](#)
 - [Docker Chatwoot Production deployment guide](#)
 - [Chatwoot User Guide](#)
- [Customizações no Chatwoot](#)
 - [Liberação de versão Enterprise do Chatwoot](#)
 - [Disparador de Campanhas WhatsApp no Chatwoot](#)

Instalação e configurações do Chatwoot

Instalação e configurações do Chatwoot

Docker Chatwoot Production deployment guide

Link: <https://developers.chatwoot.com/self-hosted/deployment/docker>

Deploy Chatwoot using Docker containers for production environments

Pre-requisites

Before proceeding, make sure you have the latest version of `docker` and `docker-compose` installed.

As of now [at the time of writing this doc], we recommend a version equal to or higher than the following.

```
$ docker --version
Docker version 20.10.10, build b485636
$ docker compose version
Docker Compose version v2.14.1
```

Container name uses dashes instead of underscores by default with new docker/compose versions. If you are using an older version of docker/compose, replace `-` with `_`. Also, use `docker-compose` instead of `docker compose`.

Steps to deploy Chatwoot using docker-compose

1. Install Docker on your VM

■

Copy

```
# example in ubuntu
apt-get update
apt-get upgrade
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
apt install docker-compose-plugin
```

2. Download the required files

▪

Copy

```
# Download the env file template
wget -O .env https://raw.githubusercontent.com/chatwoot/chatwoot/develop/.env.example
# Download the Docker compose template
wget -O docker-compose.yaml
https://raw.githubusercontent.com/chatwoot/chatwoot/develop/docker-compose.production.yaml
```

3. Configure environment variables

Tweak the `.env` and `docker-compose.yaml` according to your preferences. Refer to the available [environment variables](#). You could also remove the dependant services like `Postgres`, `Redis` etc., in favor of managed services configured via environment variables.

▪

Copy

```
# update redis and postgres passwords
nano .env
# update docker-compose.yaml same postgres pass
```

```
nano docker-compose.yaml
```

4. Prepare the database

■

Copy

```
docker compose run --rm rails bundle exec rails db:chatwoot_prepare
```

5. Start the services

■

Copy

```
docker compose up -d
```

6. Access your installation

Your Chatwoot installation is complete. Please note that the containers are not exposed to the internet and they only bind to the localhost. Setup something like Nginx or any other proxy server to proxy the requests to the container.

If you want to verify whether the installation is working, try `curl -I localhost:3000/api` to see if it returns `200`. Also, you could temporarily drop the `127.0.0.1:3000:3000` for rails to `3000:3000` in the compose file to access your instance at `http://<your-external-ip>:3000`. It's recommended to revert this change back and use Nginx or some proxy server in the front.

Additional Steps

1. Have an `Nginx` web server acting as a reverse proxy for Chatwoot installation. So that you can access Chatwoot from `https://chat.yourdomain.com`
2. Run `docker compose run --rm rails bundle exec rails db:chatwoot_prepare` whenever you decide to update the Chatwoot images to handle the migrations.

Configure Nginx and Let's Encrypt

1. Configure Nginx to serve as a frontend proxy

```
sudo apt-get install nginx
cd /etc/nginx/sites-enabled
nano yourdomain.com.conf
```

2. Use the following Nginx config

Use the following Nginx config after replacing the `yourdomain.com` in `server_name`.

```
server {
    server_name <yourdomain.com>;

    # Point upstream to Chatwoot App Server
    set $upstream 127.0.0.1:3000;

    # Nginx strips out underscore in headers by default
    # Chatwoot relies on underscore in headers for API
    # Make sure that the config is set to on.
    underscores_in_headers on;
    location /.well-known {
        alias /var/www/ssl-proof/chatwoot/.well-known;
    }

    location / {
        proxy_pass_header Authorization;
        proxy_pass http://$upstream;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
```

```
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Ssl on; # Optional

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

proxy_http_version 1.1;
proxy_buffering off;

client_max_body_size 0;
proxy_read_timeout 36000s;
proxy_redirect off;
}
listen 80;
}
```

3. Verify and reload Nginx config

```
nginx -t
systemctl reload nginx
```

4. Run Let's Encrypt to configure SSL certificate

```
apt install certbot
apt-get install python3-certbot-nginx
mkdir -p /var/www/ssl-proof/chatwoot/.well-known
certbot --webroot -w /var/www/ssl-proof/chatwoot/ -d yourdomain.com -i nginx
```

5. Access your installation

Your Chatwoot installation should be accessible from the `https://yourdomain.com` now.

Steps to build images yourself

We publish our base images to the Docker hub. You should be able to build your Chatwoot web/worker images from these base images.

Web

■

Copy

```
FROM chatwoot/chatwoot:latest
RUN chmod +x docker/entrypoints/rails.sh
ENTRYPOINT ["docker/entrypoints/rails.sh"]
CMD bundle exec bundle exec rails s -b 0.0.0.0 -p 3000
```

Worker

▪

Copy

```
FROM chatwoot/chatwoot:latest
RUN chmod +x docker/entrypoints/rails.sh
ENTRYPOINT ["docker/entrypoints/rails.sh"]
CMD bundle exec sidekiq -C config/sidekiq.yml
```

The app servers will run available on port `3000`. Ensure the images connect to the same database and Redis servers. Provide the configuration for these services via [environment variables](#).

Initial database setup

To set up the database for the first time, you must run `rails db:chatwoot_prepare`. You may get errors if you try to run `rails db:migrate` at this point.

Upgrading

If you're not using the `latest` or `latest-ce` tag, you first need to change the desired tag in your docker-compose file.

After that you can pull the new image and start using them:

■

Copy

```
docker compose pull
docker compose up -d
```

Finally you may need to update the database:

■

Copy

```
docker compose run --rm rails bundle exec rails db:chatwoot_prepare
```

Running Rails Console

■

Copy

```
docker exec -it $(basename $(pwd))-rails-1 sh -c 'RAILS_ENV=production bundle exec rails c'
```

Chatwoot CE edition docker images

If you want to run Chatwoot CE edition, replace the docker image tag with equivalent foss version tag. Docker tag for current `master` would be `latest-ce`. Version specific tags would follow the pattern `v*-ce`. For example the docker ce edition tag for Chatwoot `v2.3.2` would be `v2.3.2-ce`.

Instalação e configurações do Chatwoot

Chatwoot User Guide

Link: <https://chatwoot.help/hc/user-guide/en>

Search for the articles here or browse the categories below.

Featured Articles

[How to enhance conversations with OpenAI integration?](#)

[How to setup a WhatsApp channel?](#)

[How to use Agent bots?](#)

[Website live chat settings explained](#)

[How to create an API channel inbox?](#)

[How to use webhooks?](#)



Customizações no Chatwoot

Implementações no Chatwoot

Liberação de versão Enterprise do Chatwoot

Link: <https://github.com/CHypeTools/Dchat>

? Dchat - Chatwoot Enterprise Unlocker

Script para desbloquear funcionalidades enterprise do Chatwoot, removendo limitações da versão community.

? Uso Rápido

Execute diretamente no container do Chatwoot:

```
wget -q0- https://raw.githubusercontent.com/LuizBranco-ClickHype/Dchat/main/unlock.rb | bundle e
```

? O que o script faz

? Configurações do Banco de Dados

- Define o plano como `enterprise`
- Configura limite de usuários para 9.999.999
- Remove alertas de limitação do Redis

? Atualização de Fallbacks

- Modifica `lib/chatwoot_hub.rb`

- Cria backup automático do arquivo original
- Atualiza valores padrão para enterprise

? Funcionalidades Desbloqueadas

Após executar o script, seu Chatwoot terá:

- **Usuários ilimitados** (9.999.999)
- **Funcionalidades enterprise** ativadas
- **Sem alertas** de limitação
- **Configurações persistentes**

? Detalhes Técnicos

Arquivos Modificados

- `installation_configs` (banco de dados)
- `lib/chatwoot_hub.rb` (fallbacks)
- Cache Redis (limpeza de alertas)

Configurações Aplicadas

```
INSTALLATION_PRICING_PLAN = 'enterprise'  
INSTALLATION_PRICING_PLAN_QUANTITY = 9999999
```

Backups Automáticos

O script cria backups automáticos antes de modificar arquivos:

```
lib/chatwoot_hub.rb.backup.YYYYMMDD_HHMMSS
```

? Compatibilidade

- Container Docker do Chatwoot

- Instalações Rails padrão
- Versões recentes do Chatwoot

?? Importante

Este script modifica configurações do Chatwoot para remover limitações comerciais. Use de acordo com os termos de licença do software.

? Após a Execução

1. Reinicie o container do Chatwoot
2. Acesse a interface web
3. Verifique se as limitações foram removidas

??? Autor

Dchat desenvolvido por **LuizBranco-ClickHype**

? Repositório: [LuizBranco-ClickHype/Dchat](https://github.com/LuizBranco-ClickHype/Dchat)

Disparador de Campanhas WhatsApp no Chatwoot

Link: <https://github.com/rodtanci/Disparador-de-Campanha-Chatwoot-Evolution>

git clone <https://github.com/rodtanci/Disparador-de-Campanha-Chatwoot-Evolution.git>

? Disparador de Campanhas via WhatsApp [SAAS]?

? Introdução ?

O disparador de campanhas oferece uma forma eficiente e eficaz de gerenciar e enviar mensagens de campanha utilizando o ChatWoot. Esta solução é projetada para facilitar a comunicação com seus clientes, permitindo agendamento de campanhas, envio de mensagens personalizadas e acompanhamento de desempenho.

Funcionalidades

? Campanhas dentro do ChatWoot

Integração completa com o ChatWoot para gerenciar suas campanhas diretamente dentro da plataforma.

? Agendamento e Disparo Imediato

- Agende suas campanhas para uma data e hora específicas.

- Opção para disparo imediato, permitindo envios em tempo real.

?? Envio de Imagem

- Possibilidade de incluir imagens nas mensagens da campanha.
- Suporte para múltiplos formatos de imagem (JPEG e PNG).
- Usar a variável `&img=linkdaimagem.jpg`

? Envio de PDF

- Possibilidade de incluir PDF junto com uma mensagem.
- Usar a variável `&doc=linkdopdf.pdf`

? Envio de Video

- Possibilidade de incluir video nas mensagens da campanha.
- Usar a variável `&vid=linkdovideo.mp4`

? Envio de Audio

- Possibilidade de incluir audio nas mensagens da campanha.
- Usar a variável `&audio=linkdoaudio.mp3`

? Limitador de Envios Diário por Empresa

- Defina um limite diário de envios para cada empresa.
- Evite sobrecarregar a utilização da VPS com o alto fluxo de mensagens.

? Temporizador Randômico entre os Envios

- Adicione um temporizador randômico entre os envios para evitar detecção como spam e bloquear o número.
- (Lembrando que isso não é uma garantia de que você não possa perder seu número e sim uma solução para reduzir os riscos.)

? Contador de Envios e Falhas

- Conta número de envios `[]` .
- Conta número de falhas `[]` .

? Mensagem de Excesso de Limite Diário

- Envio automático de uma mensagem quando o limite diário de envios for excedido.
- Mantenha os administradores informados sobre o status das campanhas.

? Mensagem de Relatório da Campanha

- Receba uma mensagem avisando que o disparo da campanha foi iniciado.
- Receba um relatório detalhado ao final de cada campanha.
- Inclui número total de envios e falhas.
- Inclui número de disparos que restam no dia.
- Inclui os números que não enviaram com o nome do contato.

? Personalização com Nome e Email do Contato

- Use a variável (&nome) para personalizar as mensagens com o nome do contato.
- Use a variável (&email) para personalizar as mensagens com o email do contato.
- Aumente a personalização e eficácia das suas mensagens de campanha.

?? Disparos através das Etiquetas dos Contatos

- Utilize as etiquetas dos contatos para segmentar e direcionar suas campanhas de maneira eficiente.
- Agrupe contatos com base em características específicas e envie mensagens direcionadas.

? Benefícios ?

- **Automatização:** Reduza o trabalho manual com a automação de campanhas. ☐
 - **Personalização:** Melhore a experiência do cliente com mensagens personalizadas. ☐
 - **Eficiência:** Acompanhe o desempenho em tempo real e otimize suas campanhas. ☐
 - **Integração:** Dispare campanhas diretamente pelo ChatWoot sem precisar abrir outro sistema, otimizando tempo e recursos. ☐
-

? Tutorial de Automação de Disparo de Campanhas

Vamos começar o tutorial para fazer a automação do sistema de disparo de campanhas usando o n8n e a Evolution API junto ao ChatWoot.

Antes de iniciar, certifique-se de que você já tem instalado:

- ChatWoot
- n8n
- Evolution API
- pgAdmin ou outro de sua preferência para acessar o banco de dados do Postgres

Passo 1: Criar uma Caixa de Entrada de Canal SMS do Tipo Bandwidth

1. **Acesse o ChatWoot:** Faça login na sua conta do ChatWoot.
2. **Configurações:** Vá para a seção de configurações.
3. **Caixas de Entrada:** Selecione "Caixas de Entrada" no menu.
4. **Adicionar Nova Caixa de Entrada:** Clique no botão "Adicionar Nova Caixa de Entrada".
5. **Escolher Tipo de Canal:** Selecione "SMS" e escolha "Bandwidth" como o tipo de canal.
6. **Configurar Detalhes do Canal:**
 - Nome da Caixa de Entrada: Disparador (ou o nome que preferir).
 - Número de telefone: +741963 (NUMERAÇÃO ALEATÓRIA)
 - ID da Conta: 1 (ID DA CAIXA DE ENTRADA CADASTRADA NO CHATWOOT QUE VAI MANDAR AS MENSAGENS)
 - ID da aplicação: Instancia (NOME DA INSTACIA NA EVOLUTION)
 - Chave API: +5511934849643 (NUMERO DE WHATSAPP PARA RECEBER O RELATORIO)
 - Chave secreta API: email@exemplo.com (EMAIL PARA RECEBER O RELATORIO)
7. **Salvar Configurações:** Clique em "Criar canal Bandwidth" para criar a nova caixa de entrada.

Passo 2: Adicionar Colunas no Banco de Dados do ChatWoot

- **Acesse o Banco de Dados:** Use o pgAdmin ou outro software de sua preferência para acessar o banco de dados do ChatWoot.
- **Adicionar Coluna na Tabela Accounts:**
 - Execute o seguinte comando SQL para adicionar a coluna `limite_disparo`:

```
ALTER TABLE accounts
ADD COLUMN limite_disparo INTEGER NOT NULL DEFAULT 500;
```

Adicionar Colunas na Tabela campaigns:

- Execute os seguintes comandos SQL para adicionar as colunas status_envia, enviou e falhou na tabela campaigns:

```
ALTER TABLE campaigns
ADD COLUMN status_envia INTEGER NOT NULL DEFAULT 0;

ALTER TABLE campaigns
ADD COLUMN enviou INTEGER NOT NULL DEFAULT 0;

ALTER TABLE campaigns
ADD COLUMN falhou INTEGER NOT NULL DEFAULT 0;
```

Adicionar nova Tabela para guardar os envios que falharem:

- Execute o seguinte comando SQL para adicionar a tabela campaigns_failed:

```
-- Cria a sequência para a tabela campaigns_failed
CREATE SEQUENCE campaigns_failed_id_seq;

-- Cria a tabela com a coluna 'id' usando a sequência criada
CREATE TABLE campaigns_failed (
  id BIGINT PRIMARY KEY NOT NULL DEFAULT nextval('campaigns_failed_id_seq'::regclass),
  nomecontato CHARACTER VARYING(100),
  telefone CHARACTER VARYING(15),
  id_campanha INT);
```

Passo 3: Importar Workflows no n8n

1. **Acesse o n8n:** Faça login na sua instância do n8n.
2. **Adicionar Novo Workflow:**
 - Clique em "Add Workflow".
3. **Importar Workflow:**
 - Clique nos três pontinhos no canto superior direito.
 - Selecione "Import from File".
4. **Importar o Fluxo Disparador:**
 - Importe o arquivo de workflow disparador.json.

5. Importar o Fluxo Reset-Limite-Campanhas:

- Repita os passos acima e importe o reset-limite-campanha.json.

Passo 4: Editar o Workflow Disparador no n8n

1. **Acesse o Workflow Disparador:** No n8n, abra o workflow Disparador que você importou.
2. **Editar o primeiro nó do postgres Buscar campanhas**
 - Alterar na linha "select * from campaigns c where campaign_type = 1 and status_envia = 0 and account_id = 1"
 - altere o valor e account_id para o ID da conta do chatwoot.
3. **Editar Nó Info_Base:**
 - Preencha os seguintes campos com suas informações:
 - **URL do ChatWoot**
 - **URL da Evolution API**
 - **URL do view Typebot**
 - **Token de acesso da conta do ChatWoot**
 - **Global API KEY da Evolution API**
 - **Email que vai enviar o relatório**
4. **Conectar Nós do Postgres ao Banco de Dados do ChatWoot:**
 - Conecte todos os nós do Postgres ao banco de dados do ChatWoot, garantindo que as informações fluam corretamente entre os sistemas.

Passo 5: Editar o Workflow reset-limite-campanha no n8n

1. **Acesse o Workflow reset-limite-campanha:** No n8n, abra o workflow reset-limite-campanha que você importou.
2. **Conectar Nós do Postgres ao Banco de Dados do ChatWoot:**
 - Conecte todos os nós do Postgres ao banco de dados do ChatWoot, garantindo que as informações sejam atualizadas corretamente para resetar o limite de disparo diário.
 - Se desejar altere o limite de envio diario dentro do nó do postgres.
 - Recomendação para não sobrecarregar o worflow 500 disparos seguidos.

Agora que os workflows foram importados, configurados e editados, sua automação de disparo de campanhas está pronta para funcionar!

? Cadastro de uma Nova Campanha no ChatWoot

Para criar uma campanha, siga os passos abaixo:

1. **Clique em Campanhas:** Acesse a seção de Campanhas no ChatWoot.
2. **Clique em Única:** Selecione a opção "Única".
3. **Clique em Criar uma campanha única:** Adicione as informações da sua campanha.
 - **Título:** Insira o título da campanha.
 - **Mensagem:** Digite a mensagem que quer enviar na campanha.
 - Para adicionar o nome do contato à mensagem, digite `&nome`.
 - Para adicionar o email do contato à mensagem, digite `&email`.
 - Para adicionar uma imagem, digite `&img=urldaimagem aqui .jpg`.
 - Para adicionar um video, digite `&vid=urldovideo.mp4`.
 - Para adicionar um pdf, digite `&doc=urldopdf.pdf`.

Exemplo de uso imagem:

"Olá &nome, tudo bem? Você já viu esta bela paisagem?"

`&img=https://img.freepik.com/fotos-gratis/paisagem-de-nevoeiro-matinal-e-montanhas-com-baloes-de-ar-quente-ao-nascer-do-sol_335224-794.jpg`"

Exemplo de uso pdf:

"Olá &nome, tudo bem? Segue o nosso catalogo em PDF?"

`&doc=https://evolution-api.com/files/evolution-api.pdf`"

Exemplo de uso video:

"Olá &nome, tudo bem? Você já viu esse vídeo??"

`&vid=https://evolution-api.com/files/video.mp4`"

Exemplo de uso typebot:

`&typebot=id_da_url_do_fluxo_do_typebot"`

Exemplo url: <https://chatbotapi.cubochat.com.br/fluxotypebot> Usar: fluxotypebot

4. **Selecionar Caixa de Entrada:** No campo "Selecionar caixa de entrada", selecione a caixa de SMS que você criou no início do tutorial.
5. **Público:** Selecione a etiqueta que está atribuída aos contatos que quer disparar a campanha.
6. **Horário Agendado:** Selecione o dia e a hora que quer disparar a campanha. Caso queira disparar a campanha na mesma hora, selecione a data e a hora atual.
7. **Clique em Criar:** Finalize a criação da campanha.

Agora tudo está pronto para enviar a sua campanha!

? Roadmap do Projeto

Versão 1.1 ?

Mensagem de Início de Disparo

- Implementar uma mensagem que indique o início do disparo da campanha para notificar os usuários sobre o processo inicial.

Correção no limitador

- Correção no limitador de disparo, foi encontrado um erro que ele rouba 1 limite da conta quando chega em 1.

Versão 1.2 ?

Relatório de Finalização por Email

- Adicionar funcionalidade para enviar um relatório por email ao final da campanha, detalhando métricas e resultados.

Inclusão do Limite de Disparos Diários no Relatório

- Introduzir no relatório o número restante de disparos diários disponíveis, fornecendo transparência sobre os limites e uso.

Versão 1.3 ?

Adicionando variável

- Vamos adicionar a variável &email para conseguir personalizar mais ainda a mensagem.

Versão 1.4 ?

Relatório de Falhas de Envio

- Desenvolver um relatório específico que liste os contatos que falharam no envio, incluindo nome e telefone, para análise e correção.

Versão 1.5 ?

Envio de PDF

- Incluir a inserção de um arquivo pdf dentro da campanha.

Versão 1.6 ?

Envio de VIDEO

- Incluir a inserção de video dentro da campanha.

Versão 1.7 ?

Envio de AUDIO

- Incluir a inserção de audio dentro da campanha.

Versão 1.8 ????

Multiplas etiquetas

- Disparar campanha para mais de uma etiqueta.

Versão 1.9 ?? (Já disponível)

Buscar etiquetas na conversa

- Agora alem de buscar a etiqueta do contato ele busca nas conversas. **Envia campanha para grupos**

- Agora suas campanhas também serão enviadas para grupos do whatsapp.

Versão 2.0 [SAAS] ?

Apenas um workflow por instancia

- Este workflow foi desenvolvido para funcionar de forma centralizada com toda a instância do Chatwoot, eliminando a necessidade de criar um novo fluxo no n8n para cada caixa de entrada individual.
- Com esse modelo SaaS, é possível disparar campanhas por todos os números conectados à plataforma de forma unificada. Basta apenas criar as caixas de entrada (SMS) desejadas no Chatwoot, configurar os respectivos disparadores e informar corretamente o Application ID, que deve ser o nome da caixa de entrada cadastrada na Evolution API.
- Dessa forma, o envio de mensagens é direcionado automaticamente para o número vinculado à caixa de entrada correspondente, garantindo escalabilidade, agilidade e padronização no processo de campanhas.

Versão 2.1 ? (Já disponível)

Envia fluxo de mensagem do typebot

- Agora você pode enviar fluxo de typebot ao invés de mensagem única.

Considerações Finais ??

- Este roadmap pode ser ajustado conforme novas ideias surjam ou prioridades se alterem durante o desenvolvimento. Cada etapa visa melhorar a funcionalidade e a eficiência do disparador de campanhas, proporcionando uma experiência mais completa e eficaz aos usuários.

? Apoio ao Projeto

Quem quiser apoiar o projeto com dicas de melhorias e reportar alguma falha pode me acionar pelo grupo do WhatsApp.

<https://chat.whatsapp.com/H2as2v9yHre8U2gjNaCWRC>

Canal do Youtube: <https://www.youtube.com/@RodrigoTanci>

Caso alguém queira contribuir de forma monetária, o chave PIX CNPJ: **36799434000140**