

Omnichannel WhatsApp CRM — Complete Offline Documentation & Setup Manual

Product: Omnichannel WhatsApp CRM

Version: 1.9.0

Vendor / Engine: Powered by Dork Industry

License: Dork Industry Commercial Enterprise

Offline Edition: Fully self-contained local reference manual

Table of Contents

1. [Platform Architecture & System Overview](#)
2. [Prerequisites & Server System Requirements](#)
3. [Server Installation & Production Deployment](#)
 - [Automated Web Installer](#)
 - [CLI / Terminal Manual Installation](#)
 - [Nginx & HTTPS SSL Configuration](#)
 - [Background Queue Worker \(Supervisor\)](#)
 - [Cron Job Task Scheduler](#)
4. [Environment Configuration Reference \(.env\)](#)
5. [Super Admin Administration Guide](#)
 - [Client & Workspace Management](#)
 - [SaaS Pricing Plans & Feature Quotas](#)
 - [Payment Gateway Configuration & Webhooks](#)
 - [System SMTP Email Gateway](#)
 - [Global AI Providers & Vector DB Configuration](#)
6. [Client Workspace Operations Manual](#)
 - [Omnichannel Shared Live Chat Inbox](#)
 - [Meta WhatsApp Cloud API Channel Setup](#)

- [Contacts, Tags & Dynamic Segmentation](#)
- [Broadcast Campaigns & WhatsApp Template Management](#)
- [Visual Drag-and-Drop Automation Flow Builder](#)
- [AI Chatbots, Custom Knowledge Bases & RAG](#)
- [B2B Leads Discovery \(Google Places Scraper\)](#)
- [E-Commerce Automation \(Shopify & WooCommerce\)](#)

7. [Third-Party Integration Guides & UI Workflows](#)

- [Meta WhatsApp Cloud API](#)
- [Real-Time WebSocket Engine \(Pusher / Reverb\)](#)
- [Artificial Intelligence Models & Qdrant](#)
- [Payment Gateways \(Stripe, Razorpay, PayPal, Cashfree\)](#)
- [Transactional Mail \(SMTP / SES / Mailgun\)](#)

8. [REST API v1 & Webhooks Reference](#)

9. [Maintenance, Diagnostics & Troubleshooting FAQ](#)

1. Platform Architecture & System Overview

Omnichannel WhatsApp CRM is a cloud-native, multi-tenant SaaS communication and marketing automation engine built for enterprises, agencies, and high-growth businesses.

Core Technologies

- **Backend Framework:** Laravel 12.x (PHP 8.2+)
- **Frontend Layer:** Vue 3 with Inertia.js & Tailwind CSS
- **Database Support:** MySQL 8.0+, MariaDB 10.5+, or SQLite 3.35+
- **Asynchronous Task Queue:** Redis with Laravel Horizon or standard database queues
- **Real-Time Messaging:** Pusher Channels or self-hosted Laravel Reverb
- **Vector Knowledge Base:** Qdrant Vector Search engine for enterprise RAG chatbots

2. Prerequisites & Server System Requirements

Ensure your server (VPS, Dedicated, or Cloud Instance such as AWS EC2, DigitalOcean Droplet, or Linode) meets the following requirements:

Hardware Specifications

Deployment Tier	CPU Cores	RAM	Storage	Suitable For
Development / Test	1 vCPU	2 GB	20 GB SSD	Local testing, < 5 concurrent agents
Standard Production	2-4 vCPU	4-8 GB	60 GB NVMe	Up to 100k monthly messages, 30 agents
High-Volume Enterprise	8+ vCPU	16-32 GB	150+ GB NVMe	1M+ monthly messages, 100+ agents

Software & PHP Extensions

- **Operating System:** Ubuntu 22.04 LTS / 24.04 LTS or Debian 12
- **Web Server:** Nginx (recommended) or Apache 2.4 with `mod_rewrite`
- **PHP Version:** PHP 8.2 or PHP 8.3
- **Mandatory PHP Modules:**

```
bash
php8.2-bcmath php8.2-ctype php8.2-curl php8.2-dom php8.2-fileinfo \
php8.2-gd php8.2-mbstring php8.2-openssl php8.2-pdo php8.2-mysql \
php8.2-sqlite3 php8.2-tokenizer php8.2-xml php8.2-zip php8.2-redis
```

- **Process Manager:** `supervisor`
- **Node.js:** Node.js v18 LTS or v20 LTS with NPM

3. Server Installation & Production Deployment

Automated Web Installer

If deploying to a fresh server:

1. Upload and extract the project files into your web root directory (e.g. `/var/www/whatsapp-crm`).
2. Point your web server root to the `public/` folder.
3. Open `https://your-domain.com/install` in your browser.
4. The web installer will automatically check server requirements, write your `.env` database parameters, run database migrations, seed default admin credentials, and generate your application key.

CLI / Terminal Manual Installation

For automated CI/CD and terminal deployments:

```
bash
# 1. Clone or copy project to web root
cd /var/www/whatsapp-crm

# 2. Set file permissions for the web server user
sudo chown -R www-data:www-data /var/www/whatsapp-crm
sudo chmod -R 775 /var/www/whatsapp-crm/storage
sudo chmod -R 775 /var/www/whatsapp-crm/bootstrap/cache

# 3. Install composer dependencies (no dev packages in production)
composer install --no-dev --optimize-autoloader

# 4. Copy environment file and generate key
cp .env.example .env
php artisan key:generate

# 5. Configure your database in .env, then run the installer
php artisan saas:install

# 6. Run database migrations & seeders
php artisan migrate --force
php artisan db:seed --force

# 7. Create storage symlink
php artisan storage:link

# 8. Build production frontend assets (if compiling locally)
npm ci
npm run build

# 9. Optimize Laravel for production
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

Nginx & HTTPS SSL Configuration

Create `/etc/nginx/sites-available/whatsapp-crm.conf`:

```

nginx
server {
    listen 80;
    server_name crm.yourcompany.com;
    root /var/www/whatsapp-crm/public;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    index index.php index.html;
    charset utf-8;

    client_max_body_size 64M;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt  { access_log off; log_not_found off; }

    error_page 404 /index.php;

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
        fastcgi_hide_header X-Powered-By;
    }

    location ~ /\.(!well-known).* {
        deny all;
    }
}

```

Enable site and install SSL certificate:

```

bash
sudo ln -s /etc/nginx/sites-available/whatsapp-crm.conf /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
sudo certbot --nginx -d crm.yourcompany.com

```

Background Queue Worker (Supervisor)

Real-time message routing, WhatsApp webhooks, scheduled broadcasts, and AI streaming require an active background queue worker.

Create `/etc/supervisor/conf.d/whatsapp-crm-worker.conf`:

```
ini
[program:whatsapp-crm-worker]
process_name=%(program_name)s_%(process_num)02d
command=php /var/www/whatsapp-crm/artisan queue:work redis --sleep=3 --tries=3 --max-time=3600 --timeout=120
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=www-data
numprocs=4
redirect_stderr=true
stdout_logfile=/var/www/whatsapp-crm/storage/logs/worker.log
stopwaitsecs=3600
```

Start and verify workers:

```
bash
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start whatsapp-crm-worker:*
sudo supervisorctl status
```

Cron Job Task Scheduler

Scheduled campaigns, subscription renewal reminders, contact syncs, and automatic cleanups run through the Laravel scheduler.

Add this entry to your crontab (`crontab -e -u www-data`):

```
bash
* * * * * cd /var/www/whatsapp-crm && php artisan schedule:run >> /dev/null 2>&1
```

4. Environment Configuration Reference (.env)

The `.env` file in your root folder governs the behavior of your installation.

```
ini
# Application Setup
APP_NAME="Omnichannel WhatsApp CRM"
APP_ENV=production
APP_KEY=base64:...
APP_DEBUG=false
APP_URL=https://crm.yourcompany.com

# Database Connection (MySQL / MariaDB)
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=whatsapp_crm_db
DB_USERNAME=crm_user
DB_PASSWORD="your_strong_password"

# Cache & Queues (Redis recommended for high concurrency)
BROADCAST_CONNECTION=pusher
CACHE_STORE=redis
QUEUE_CONNECTION=redis
SESSION_DRIVER=redis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

# Real-Time WebSocket Notifications (Pusher)
PUSHER_APP_ID="your_pusher_app_id"
PUSHER_APP_KEY="your_pusher_key"
PUSHER_APP_SECRET="your_pusher_secret"
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER="mt1"

VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"

# File Storage
FILESYSTEM_DISK=public
```

5. Super Admin Administration Guide

The Super Admin portal is accessible at <https://crm.yourcompany.com/admin>.

Client & Workspace Management

- **Tenant Isolation:** Each business subscriber receives an isolated Workspace with independent contacts, WhatsApp phone channels, team member permissions, automation flows, and webhook endpoints.
- **Login Impersonation:** Super Admins can securely impersonate any client workspace with a single click to troubleshoot configuration or view user reports without requiring customer passwords.
- **Suspend / Reactivate:** Easily lock workspaces upon non-payment or policy violation while preserving all underlying message data.

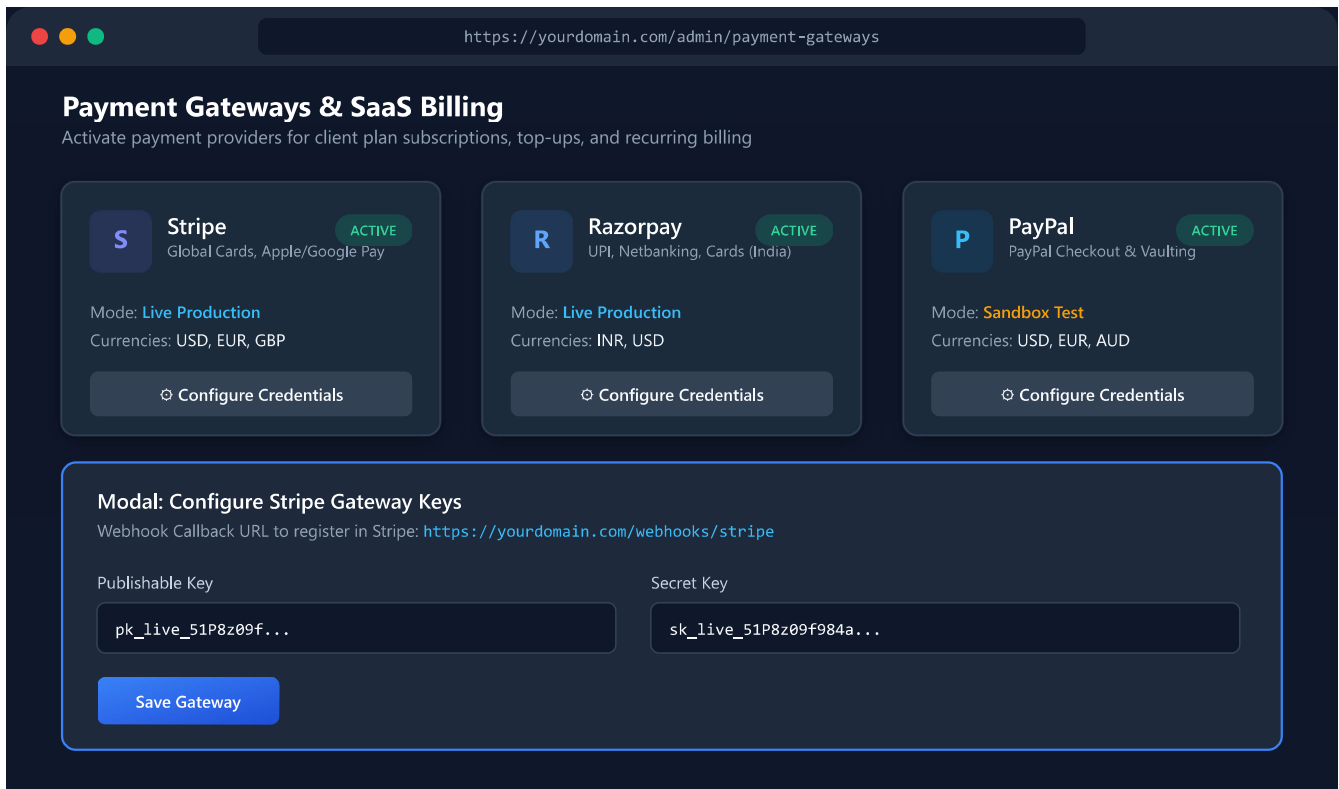
SaaS Pricing Plans & Feature Quotas

In **Admin Panel** → → **Plans**:

- Create monthly, quarterly, and annual subscription tiers.
- Configure granular limits:
 - Max Team Members / Agent Seats.
 - Max Contacts stored in CRM.
 - Max WhatsApp Connected Numbers.
 - Monthly Outbound Message / Broadcast Quota.
 - AI Assistant Credits (OpenAI / Gemini tokens).
 - Feature access toggles: Automation Flow Builder, Web Chat Widget, Google Places Scraper, Shopify Integration.

Payment Gateway Configuration & Webhooks

Navigate to **Admin Panel** → → **Settings** → → **Payment Gateways**:



Payment Gateways Setup

1. Stripe:

- Enter Publishable Key and Secret Key .
- Webhook endpoint: `https://crm.yourcompany.com/webhooks/stripe` .
- Listen for events: `checkout.session.completed` , `invoice.payment_succeeded` , `customer.subscription.deleted` .

2. Razorpay:

- Enter Key ID and Key Secret .
- Webhook URL: `https://crm.yourcompany.com/webhooks/razorpay` .

3. PayPal:

- Set Client ID, Secret, and Mode (`sandbox` or `live`).

4. Cashfree:

- Provide App ID, Secret Key, and Environment URL.

5. Bank Transfer / Offline:

- Provide custom bank instructions; admin manually approves uploaded payment receipts.

System SMTP Email Gateway

Navigate to **Admin Panel** → **Settings** → **Email Setup**:

https://yourdomain.com/admin/email-system

Email System & SMTP Configuration

Configure transactional emails for passwords, team invites, client subscriptions, and alerts

Mail Driver	SMTP Host
SMTP (Standard Server) ▼	smtp.mailgun.org
SMTP Port	Encryption Protocol
587	TLS (Recommended) ▼
SMTP Username	SMTP Password
postmaster@mg.yourdomain.com	
Sender "From" Email Address	Test Email Recipient
notifications@yourdomain.com	admin@yourdomain.com

Save Email Config

Send Test

SMTP Email Setup

- Supported drivers: Standard SMTP, Amazon SES, Mailgun, Postmark.
- Use the built-in **Send Test Email** button to verify outgoing deliverability before activating user email verification.

Global AI Providers & Vector DB Configuration

Navigate to **Admin Panel** → **Settings** → **AI Settings**:

https://yourdomain.com/admin/integrations (AI Provider Tab)

AI Intelligence & Knowledge Base Providers

Configure LLM engines (OpenAI, Gemini, Anthropic) and Vector Database for smart bot replies

**OpenAI (ChatGPT)**
• Enabled (Primary)
GPT-4o, GPT-4o-mini, Embeddings

**Anthropic Claude**
Available
Claude 3.5 Sonnet, Haiku

**Google Gemini**
Available
Gemini 1.5 Flash, Pro Vision

OpenAI API Configuration

API Secret Key
sk-proj-78a019bd823fa908b87c1092e48fa...

Default Chatbot Model
gpt-4o-mini (Recommended) ▼

Vector Knowledge Base Engine (RAG)
Qdrant Cloud: https://xyz-cluster.qdrant.tech:6333

Max Tokens per Response
4,096 tokens

Save AI Config

⚡ Test AI Completion

AI Configuration

- **OpenAI:** Supply API Key and default model (`gpt-4o` , `gpt-4o-mini`).
- **Google Gemini:** Set Gemini API Key and model (`gemini-1.5-flash` or `gemini-1.5-pro`).
- **Anthropic Claude:** Set Claude API Key (`claude-3-5-sonnet`).
- **Qdrant Vector DB:**
 - Connect your Qdrant cluster host (e.g., `https://xyz.cloud.qdrant.io:6333`) and API Key.
 - Used for real-time document vector embeddings and Semantic Search for AI Agent RAG.

6. Client Workspace Operations Manual

Subscribers interact with their team workspace at `https://crm.yourcompany.com/app/dashboard` .

Omnichannel Shared Live Chat Inbox

- **Multi-Agent Collaboration:** Multiple human operators can view incoming customer messages in real-time.
- **Assignment & Routing:** Assign conversations to specific team members or auto-assign via round-robin.
- **Canned Responses:** Insert frequently used message templates with quick keyboard shortcuts (e.g. `/welcome` , `/pricing`).

- **Internal Notes & Mentions:** Leave private notes on any ticket visible only to your internal team.

Meta WhatsApp Cloud API Channel Setup

Navigate to **Workspace** → → **Channels** → → **WhatsApp Cloud API**:

The screenshot shows a web browser window with the URL `https://yourdomain.com/app/whatsapp/setup`. The page is titled "WhatsApp Channel Setup" and includes a subtitle "Connect Meta WhatsApp Business Cloud API & Real-time Webhooks". A green button labeled "+ Connect WhatsApp" is in the top right corner. Below the header, there is a section for "Omnichannel Official WABA" with a WABA ID of "104829104928190" and an "Active" status. To the right of this section are two buttons: "Sync Numbers" and "Sync Templates". The main content area is divided into two columns. The left column is titled "Meta Webhook Configuration (Copy to Meta App)" and contains a "Callback URL" field with the value `https://yourdomain.com/webhooks/whatsapp/waba_live_token` and a "Verify Token" field with the value `crm_meta_wh_9a8f27c31e40a`. Below these fields is a green checkmark icon and the text "Webhook Subscription Fields: Subscribe to: messages, message_deliveries, message_reads". The right column is titled "Connected Phone Numbers (Meta Cloud)" and contains two phone number entries. The first entry is "+1 (555) 019-2831" with a display name of "Customer Support Team" and a quality of "HIGH". The second entry is "+1 (555) 048-9120" with a display name of "VIP Sales Desk" and a quality of "HIGH". Both entries have an "ONLINE" status. At the bottom of the right column is a dashed box with the text "+ Add Another Phone Number ID".

WhatsApp Cloud API Setup

To connect an official WhatsApp Business phone number:

1. Log in to [Meta for Developers](#).
2. Create an App → → Select Type **Business**.
3. Add the **WhatsApp** product to your app.
4. From **WhatsApp** → → **API Setup**, copy:
 - **WhatsApp Business Account ID (WABA ID)**
 - **Phone Number ID**
5. Generate a **Permanent System User Access Token** in Meta Business Manager with permissions:
 - `whatsapp_business_messaging`
 - `whatsapp_business_management`
6. Paste these credentials into the CRM and click **Save Settings**.
7. Copy the **Webhook Callback URL** and **Verify Token** from the CRM and paste them into the Meta App Configuration page.

8. Subscribe to the `messages` webhook field.

Contacts, Tags & Dynamic Segmentation

- **Custom Attributes:** Store custom fields (e.g. Company Name, Order ID, VIP Status).
- **Tagging:** Categorize customers with colored badges for quick filtering.
- **CSV Import / Export:** Bulk import up to 500,000 contacts with header mapping.
- **Segments:** Create dynamic lists based on filters (e.g. "Purchased in last 30 days" or "Has tag #Interested").

Broadcast Campaigns & WhatsApp Template Management

- **Meta Official Templates:** Synchronize WhatsApp message templates directly with Meta. Submit new templates (with text, media headers, quick reply buttons, and call-to-action links) straight from the CRM without opening Facebook Business Manager.
- **Targeted Broadcasting:** Send bulk WhatsApp messages to any segment or contact tag.
- **Anti-Ban Intelligent Throttling:** Configure delay intervals between outgoing messages (e.g., 3-7 seconds random delay) to comply with WhatsApp rate limits and protect phone reputation.
- **Campaign Analytics:** Track delivery rate, read rate, and reply percentages in real-time.

Visual Drag-and-Drop Automation Flow Builder

Create automated conversational trees without code:

- **Triggers:** On keyword match (e.g. "order status", "demo"), when new conversation starts, or on webhook trigger.
- **Nodes Available:**
 - `Send Message` : Text, Image, Video, Document, or Interactive Button list.
 - `Time Delay` : Wait X minutes or hours before following up.
 - `Condition Branch` : Check if contact has specific tag, attribute, or language.
 - `Assign Agent` : Route to a specific team member or department.
 - `HTTP Webhook Request` : Send or fetch data from external APIs (CRM, Zapier, Make).
 - `AI Bot Handover` : Switch conversation to AI knowledge base bot.

AI Chatbots, Custom Knowledge Bases & RAG

Build 24/7 autonomous support agents that answer customer questions accurately using your proprietary documentation:

1. Go to **AI Chatbots** → → **Knowledge Base**.
2. Upload company documentation (PDFs, DOCX, TXT) or paste website URLs to scrape automatically.
3. The platform parses and chunks the content, generating vector embeddings stored in Qdrant.
4. When a customer asks a question, the AI retrieves relevant context passages (Retrieval-Augmented Generation) and provides a helpful, human-like response.
5. If the AI detects negative sentiment or the customer asks for a person, it automatically triggers a **Human Agent Fallback** notification in the live inbox.

B2B Leads Discovery (Google Places Scraper)

- Search any niche and city (e.g. "Dentists in Miami" or "Real Estate Agencies in London").
- Extract verified business names, telephone numbers, websites, ratings, and addresses.
- Auto-import verified numbers directly into your CRM contact segments for instant WhatsApp outreach campaigns.

E-Commerce Automation (Shopify & WooCommerce)

- **Shopify:** Connect your Shopify store to automatically sync customers and order statuses.
- **Automated Order Notifications:** Send instant WhatsApp alerts for:
 - Order Confirmation with order summary and tracking link.
 - Abandoned Cart Recovery reminders with one-click checkout buttons.
 - Out for Delivery & Fulfilled tracking updates.
- **WooCommerce:** Install the CRM webhook plugin or paste the webhook URL in WooCommerce settings to trigger automated messaging.

7. Third-Party Integration Guides & UI Workflows

Real-Time WebSocket Engine (Pusher / Reverb)

Real-time chat inbox updates and badge counters rely on WebSocket broadcasting:

Pusher Settings UI

1. Register at [Pusher.com](https://pusher.com) and create a **Channels** app.
2. Select your closest cluster (e.g. `mt1` US East, `eu` Europe, `ap2` India).
3. Copy **App ID**, **Key**, **Secret**, and **Cluster** into the CRM settings page.
4. Click **Test Realtime Connection**; the UI will emit a ping event to confirm end-to-end communication.

8. REST API v1 & Webhooks Reference

External software, ERPs, and custom landing pages can interact with Omnichannel WhatsApp CRM using the standard REST API.

Authentication

Authenticate requests using Bearer tokens generated from **Workspace Settings** → **API Keys**:

```
http
Authorization: Bearer YOUR_WORKSPACE_API_KEY
Content-Type: application/json
Accept: application/json
```

Send WhatsApp Message Endpoint

```
http
POST /api/v1/messages/send
Host: crm.yourcompany.com
```

Payload:

```
json
{
  "phone": "+1234567890",
  "type": "text",
  "message": "Hello! Your appointment is confirmed for tomorrow at 10:00 AM."
}
```

Response (200 OK):

```
json
{
  "status": "success",
  "message_id": "msg_98472918391",
  "delivered": true
}
```

Inbound Webhook Event Payloads

When enabled, the CRM sends webhook notifications to your external URL for events such as `message.received`, `message.delivered`, `contact.created`, and `lead.converted`.

9. Maintenance, Diagnostics & Troubleshooting FAQ

1. Database Compatibility (SQLite vs MySQL)

- When running on SQLite locally for testing or development, standard SQL date math functions like `TIMESTAMPDIFF` are adapted automatically via cross-database Laravel expressions.
- For high-volume production, always deploy on **MySQL 8.0+** or **MariaDB 10.5+**.

2. WhatsApp Webhook Returns 403 Forbidden

- Ensure the **Verify Token** string in your CRM channel settings exactly matches the Verify Token entered in the Meta App Dashboard under Webhooks → → WhatsApp.
- Check that your server has a valid SSL certificate (Meta will reject self-signed or expired certificates).

3. Outgoing WhatsApp Messages Show "Failed"

- Check that your Meta Permanent System User Access Token has not expired or been revoked.
- Check that your WhatsApp Business Account has a valid payment method on file in Meta Business Manager if your free conversation tier has been exhausted.
- If sending template messages, ensure the template is **APPROVED** in the Meta Business Manager before dispatching.

4. Background Queue Worker Is Not Processing

Check the worker status in terminal:

```
bash
sudo supervisorctl status whatsapp-crm-worker:*
```

If workers have died, review the log file:

```
bash
tail -n 100 /var/www/whatsapp-crm/storage/logs/worker.log
```

Restart workers after code or configuration updates:

```
bash
php artisan queue:restart
sudo supervisorctl restart whatsapp-crm-worker:*
```

5. Clearing System Cache

When making changes to configuration or `.env`:

```
bash
php artisan optimize:clear
php artisan config:cache
php artisan route:cache
```

Documentation compiled and maintained by Dork Industry. All rights reserved.