🚀 AWS Deployment Guide

Complete Guide for Deploying Frappe + MariaDB + Lending App + API Server

For DevOps Team

🔐 CRITICAL CREDENTIALS - KEEP SECURE

Local MariaDB Database:

```
Database: _af6374d4ed93f504
User: _af6374d4ed93f504
Password: zTmiKxBrhzpoetXi
Host: 127.0.0.1
Port: 3306
```

Frappe Site Configuration:

```
Site Name: lending.localhost
Database Name: _af6374d4ed93f504
Database Password: zTmiKxBrhzpoetXi
Database User: _af6374d4ed93f504
Encryption Key: jIwCPOgIRn0oEUmS09wmDXEujy4vFykRqcfBm7y_WmA=
```

Frappe API Credentials:

```
API Key: 64726967de821d4
API Secret: 18fe12924de8f23
```

```
Base URL: http://127.0.0.1:8000
Site Name: lending.localhost
```

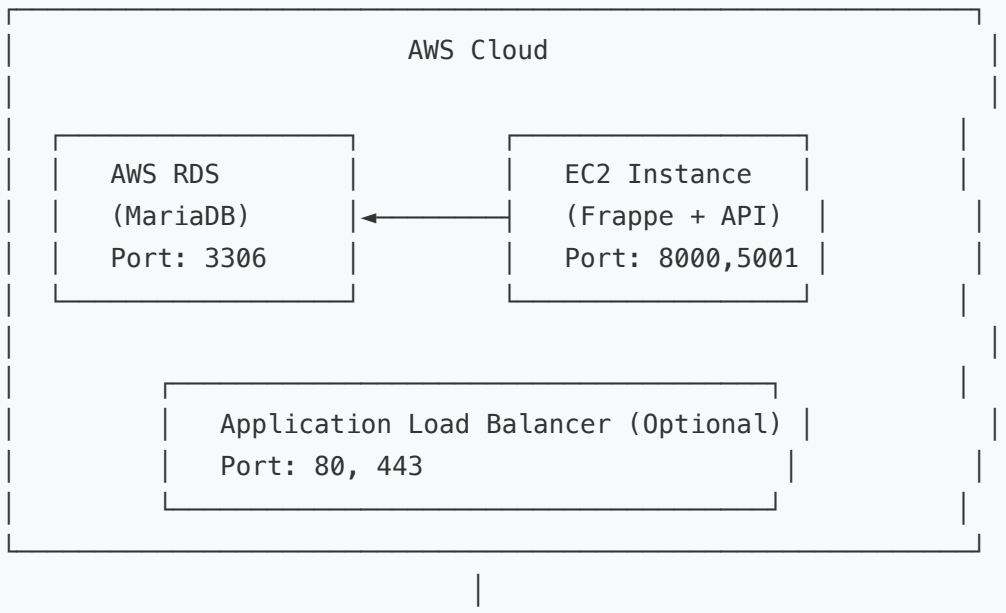
⚠️ **IMPORTANT:** These credentials are for local development. You MUST create new secure credentials for AWS production environment!

📋 AWS Hosting Complete Guide

📌 **Prerequisites:**

- AWS Account
- AWS CLI installed (optional but helpful)
- Domain name (optional, can use IP initially)
- SSH key pair for EC2

🏗️ Architecture Overview



```
                    ▼
        ┌─────────────────────┐
        │   Your Clients      │
        └─────────────────────┘
```

1 Create AWS RDS MariaDB Instance

1.1 Create RDS Database

1. **Go to AWS Console** → RDS → Create database
2. **Database Configuration:**
   - **Engine:** MariaDB
   - **Version:** 10.11 or latest
   - **Template:** Production (or Dev/Test for testing)
   - **DB instance identifier:** `frappe-lending-db`
   - **Master username:** `frappe_admin`
   - **Master password:** `[Create strong password - save it!]`
   - **DB instance class:** `db.t3.medium` (2 vCPU, 4GB RAM)
   - **Storage:**
     - Type: General Purpose SSD (gp3)
     - Allocated storage: 20 GB
     - Enable storage autoscaling: Yes
     - Maximum storage threshold: 100 GB
3. **Connectivity:**
   - **VPC:** Default VPC (or create new)
   - **Subnet group:** default
   - **Public access:** Yes (for initial setup, change later)
   - **VPC security group:** Create new
     - Name: `frappe-rds-sg`
   - **Availability Zone:** No preference

- **Database port:** 3306

4. **Database authentication:** Password authentication
5. **Additional configuration:**
   - **Initial database name:** `frappe_lending`
   - **Backup retention:** 7 days
   - **Enable encryption:** Yes (recommended)
6. **Click "Create database"** (takes 5-10 minutes)

1.2 Configure Security Group

1. Go to **EC2** → **Security Groups** → Find `frappe-rds-sg`
2. **Edit Inbound Rules:**
   - **Type:** MySQL/Aurora
   - **Port:** 3306
   - **Source:**
     - Your IP address (for initial setup)
     - Or EC2 security group (after EC2 is created)
3. **Save rules**

1.3 Get RDS Endpoint

After RDS is created:

- **Endpoint:** `frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com`
- **Port:** 3306
- **Save these details!**

2 Export Local Database

2.1 Backup Your Local Database

```
cd /Users/prom3/Desktop/regal/frappe-bench
```

```
# Export database
mysqldump -u _af6374d4ed93f504 -p'zTmiKxBrhzpoetXi' \
  _af6374d4ed93f504 > frappe_lending_backup_$(date +%Y%m%d_%H%M%S).sql

# Verify backup file
ls -lh frappe_lending_backup_*.sql
```

2.2 Test Backup File

```
# Check backup file size (should be ~50-100 MB)
du -h frappe_lending_backup_*.sql

# View first few lines
head -20 frappe_lending_backup_*.sql
```

3 Create EC2 Instance for Frappe

3.1 Launch EC2 Instance

1. **Go to AWS Console** → EC2 → Launch Instance
2. **Instance Configuration:**
   - **Name:** `frappe-lending-server`
   - **AMI:** Ubuntu Server 22.04 LTS (64-bit x86) or ARM64
   - **Instance type:** `t3.medium` (2 vCPU, 4GB RAM) or larger
   - **Key pair:** Create new or use existing
     - Name: `frappe-key`
     - Download `.pem` file and save securely
   - **Network settings:**
     - VPC: Default
     - Subnet: Public subnet
     - Auto-assign Public IP: Enable
     - Security group: Create new
       - Name: `frappe-ec2-sg`

- Rules:
  - SSH (22) - Your IP
  - HTTP (80) - Anywhere
  - HTTPS (443) - Anywhere
  - Custom TCP (8000) - Anywhere (Frappe)
  - Custom TCP (5001) - Anywhere (Your API)

3. **Storage:** 30 GB gp3 SSD
4. **Launch Instance**

3.2 Get EC2 Details

After launch:

- **Public IP:** `xx.xx.xx.xx`
- **Private IP:** `yy.yy.yy.yy`
- **Save these!**

4 Setup EC2 Instance

4.1 Connect to EC2

```
# From your local machine
chmod 400 frappe-key.pem
ssh -i frappe-key.pem ubuntu@[EC2-PUBLIC-IP]
```

4.2 Install Dependencies

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install required packages
sudo apt install -y \
  python3.11 python3.11-venv python3-pip \
  nodejs npm \
  redis-server \
```

```
  git curl wget \
  mariadb-client \
  nginx \
  supervisor \
  certbot python3-certbot-nginx

# Install Python 3.11 if not available
sudo apt install -y software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa -y
sudo apt update
sudo apt install -y python3.11 python3.11-venv python3.11-dev
```

4.3 Install Frappe Bench

```
# Install bench
sudo -H pip3 install frappe-bench

# Verify installation
bench --version
```

5 Import Database to RDS

5.1 Create Database on RDS

```
# From your local machine, connect to RDS
mysql -h frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com \
  -u frappe_admin -p \
  -e "CREATE DATABASE IF NOT EXISTS frappe_lending CHARACTER SET utf8mb4 COLLAT
```

5.2 Import Database

```
# From your local machine
mysql -h frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com \
  -u frappe_admin -p \
  frappe_lending < frappe_lending_backup_*.sql
```

> **Note:** This may take 5-15 minutes depending on database size.

5.3 Verify Import

```
mysql -h frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com \
  -u frappe_admin -p \
  -e "USE frappe_lending; SHOW TABLES;" | head -20
```

6 Setup Frappe on EC2

6.1 Initialize Frappe Bench

```
# On EC2
cd /home/ubuntu

# Initialize bench
bench init frappe-bench --frappe-branch version-15 --python python3.11
cd frappe-bench
```

6.2 Create New Site (Pointing to RDS)

```
# Create site with RDS connection
bench new-site lending.yourdomain.com \
  --db-host frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com \
  --db-port 3306 \
  --db-name frappe_lending \
  --db-user frappe_admin \
  --db-password '[YOUR-RDS-PASSWORD]' \
  --admin-password '[CREATE-ADMIN-PASSWORD]' \
  --no-mariadb-socket
```

> **Note:** Use your domain or EC2 public IP for site name.

6.3 Install Lending App

```
# Get your lending app
cd /home/ubuntu/frappe-bench

# Option 1: If app is in git repo
bench get-app lending https://github.com/your-repo/lending.git

# Option 2: If you need to copy from local
# (We'll do this in next step)

# Install app to site
bench --site lending.yourdomain.com install-app lending
```

7 Deploy Your Code to EC2

7.1 Upload Your Code

Option A: Using Git (Recommended)

```
# On EC2
cd /home/ubuntu/frappe-bench

# Clone your repo or pull updates
git clone https://github.com/your-repo/frappe-bench.git .
# OR if already cloned
git pull origin main
```

Option B: Using SCP (From Local Machine)

```
# From your local machine
cd /Users/prom3/Desktop/regal/frappe-bench

# Upload server folder
scp -i frappe-key.pem -r server/ ubuntu@[EC2-IP]:~/frappe-bench/

# Upload apps if needed
scp -i frappe-key.pem -r apps/ ubuntu@[EC2-IP]:~/frappe-bench/
```

7.2 Setup API Server on EC2

```
# On EC2
cd /home/ubuntu/frappe-bench/server

# Create virtual environment
python3.11 -m venv venv
source venv/bin/activate

# Install dependencies
pip install flask requests python-dotenv

# Create .env file
nano .env
```

Add to .env:

```
FRAPPE_BASE_URL=http://127.0.0.1:8000
FRAPPE_SITE_NAME=lending.yourdomain.com
FRAPPE_API_KEY=your-api-key
FRAPPE_API_SECRET=your-api-secret
```

7.3 Update server/utils.py

```
# On EC2
cd /home/ubuntu/frappe-bench/server
nano utils.py
```

Update FRAPPE_BASE_URL:

```
FRAPPE_BASE_URL = os.getenv('FRAPPE_BASE_URL', 'http://127.0.0.1:8000')
FRAPPE_SITE_NAME = os.getenv('FRAPPE_SITE_NAME', 'lending.yourdomain.com')
```

8 Update Frappe Site Config

8.1 Update site_config.json

```
# On EC2
cd /home/ubuntu/frappe-bench/sites/lending.yourdomain.com
nano site_config.json
```

Update to:

```
{
  "db_name": "frappe_lending",
  "db_password": "[YOUR-RDS-PASSWORD]",
  "db_type": "mariadb",
  "db_user": "frappe_admin",
  "db_host": "frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com",
  "db_port": 3306,
  "encryption_key": "[COPY-FROM-LOCAL-OR-GENERATE-NEW]"
}
```

8.2 Generate New Encryption Key (If Needed)

```
# On EC2
cd /home/ubuntu/frappe-bench
bench setup config
```

9 Start Services

9.1 Start Frappe

```
# On EC2
cd /home/ubuntu/frappe-bench

# Start Frappe (development)
bench start

# OR setup production mode
bench setup production
```

9.2 Setup API Server as Service

```
# On EC2
sudo nano /etc/systemd/system/frappe-api.service
```

Add:

```
[Unit]
Description=Frappe API Gateway
After=network.target

[Service]
Type=simple
User=ubuntu
WorkingDirectory=/home/ubuntu/frappe-bench/server
Environment="PATH=/home/ubuntu/frappe-bench/server/venv/bin"
ExecStart=/home/ubuntu/frappe-bench/server/venv/bin/python app.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Enable and start:

```
sudo systemctl daemon-reload
sudo systemctl enable frappe-api
sudo systemctl start frappe-api
sudo systemctl status frappe-api
```

10 Setup Nginx Reverse Proxy

10.1 Configure Nginx for Frappe

```
# On EC2
sudo nano /etc/nginx/sites-available/frappe
```

Add:

```
server {
    listen 80;
    server_name lending.yourdomain.com;

    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

10.2 Configure Nginx for API Server

```
sudo nano /etc/nginx/sites-available/api
```

Add:

```
server {
    listen 80;
    server_name api.yourdomain.com;

    location / {
        proxy_pass http://127.0.0.1:5001;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

10.3 Enable Sites

```
sudo ln -s /etc/nginx/sites-available/frappe /etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/api /etc/nginx/sites-enabled/
```

```
sudo nginx -t
sudo systemctl restart nginx
```

10.4 Setup SSL (Let's Encrypt)

```
# For Frappe
sudo certbot --nginx -d lending.yourdomain.com

# For API
sudo certbot --nginx -d api.yourdomain.com
```

11 Security Configuration

11.1 Update RDS Security Group

1. Go to **EC2** → **Security Groups** → `frappe-rds-sg`
2. **Edit Inbound Rules:**
   - Remove "My IP" rule
   - Add rule:
     - **Type:** MySQL/Aurora
     - **Port:** 3306
     - **Source:** `frappe-ec2-sg` (EC2 security group)

11.2 Configure Firewall (UFW)

```
# On EC2
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

11.3 Disable RDS Public Access (After Setup)

1. Go to **RDS** → Your database → **Modify**
2. **Connectivity** → **Public access:** No

3. **Apply immediately**

12 Verify Everything Works

12.1 Test Frappe

```
# On EC2
curl http://localhost:8000

# From browser
http://lending.yourdomain.com
```

12.2 Test API Server

```
# On EC2
curl http://localhost:5001/api/loan-categories

# From browser
http://api.yourdomain.com/api/loan-categories
```

12.3 Test Database Connection

```
# On EC2
mysql -h frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com \
  -u frappe_admin -p \
  -e "SELECT COUNT(*) as table_count FROM information_schema.tables WHERE table
```

13 Monitoring & Maintenance

13.1 Check Logs

```
# Frappe logs
cd /home/ubuntu/frappe-bench
bench --site lending.yourdomain.com logs
```

```
# API server logs
sudo journalctl -u frappe-api -f

# Nginx logs
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log
```

13.2 Setup Auto Backup

```
# On EC2
crontab -e
```

Add:

```
# Daily database backup at 2 AM
0 2 * * * mysqldump -h frappe-lending-db.xxxxx.us-east-1.rds.amazonaws.com -u f
```

🎯 Quick Reference Commands

Database Backup

```
mysqldump -h [RDS-ENDPOINT] -u frappe_admin -p frappe_lending > backup.sql
```

Restart Services

```
# Frappe
cd /home/ubuntu/frappe-bench
bench restart

# API Server
sudo systemctl restart frappe-api

# Nginx
sudo systemctl restart nginx
```

Update Code

```
# On EC2
cd /home/ubuntu/frappe-bench
git pull
bench migrate
bench restart
```

💰 Estimated AWS Costs (Monthly)

Service	Cost
RDS (db.t3.medium)	~$50-70
EC2 (t3.medium)	~$30-40
Storage (20GB)	~$2-5
Data Transfer	~$5-20
Total	**~$90-140/month**

🆘 Troubleshooting

Database Connection Issues

```
# Test connection
mysql -h [RDS-ENDPOINT] -u frappe_admin -p

# Check security group
# Ensure EC2 security group is allowed in RDS security group
```

Frappe Not Starting

```
# Check logs
bench --site [site-name] logs

# Check Redis
sudo systemctl status redis

# Check database connection
bench --site [site-name] console
```

API Server Not Responding

```
# Check service status
sudo systemctl status frappe-api

# Check logs
sudo journalctl -u frappe-api -n 50

# Test locally
curl http://localhost:5001/api/loan-categories
```

✅ Deployment Checklist

- □ RDS MariaDB created and accessible
- □ Local database exported
- □ Database imported to RDS
- □ EC2 instance created
- □ Frappe installed on EC2
- □ Site created pointing to RDS
- □ Lending app installed

- ☐ API server deployed
- ☐ Nginx configured
- ☐ SSL certificates installed
- ☐ Security groups configured
- ☐ Services running
- ☐ Everything tested

📜 Deployment Helper Script

The `deploy_to_aws.sh` script automates the initial deployment preparation:

Script Overview

This script helps you:

1. Backup your local database
2. Prepare files for AWS deployment
3. Generate deployment commands

Script Configuration

```
# Configuration
BACKUP_DIR="./backups"
DB_USER="_af6374d4ed93f504"
DB_PASS="zTmiKxBrhzpoetXi"
DB_NAME="_af6374d4ed93f504"
```

What the Script Does

Step 1: Creates Database Backup

```
mysqldump -u "$DB_USER" -p"$DB_PASS" "$DB_NAME" > "$BACKUP_FILE"
```

Step 2: Creates Deployment Package

- Copies `server/` folder
- Copies database backup
- Creates `DEPLOYMENT_INFO.txt`

Step 3: Generates Upload Commands

Creates `UPLOAD_COMMANDS.sh` with SCP commands to upload files to EC2.

Step 4: Generates Import Commands

Creates `IMPORT_DATABASE.sh` with MySQL commands to import database to RDS.

Running the Script

```
cd /Users/prom3/Desktop/regal/frappe-bench
chmod +x deploy_to_aws.sh
./deploy_to_aws.sh
```

Output

The script creates a `deploy_package/` directory containing:

- ✅ `server/` - Your API server code
- ✅ `frappe_lending_backup_*.sql` - Database backup
- ✅ `DEPLOYMENT_INFO.txt` - Deployment information
- ✅ `UPLOAD_COMMANDS.sh` - Upload script

- ✅ `IMPORT_DATABASE.sh` - Database import script

AWS Deployment Guide

Version 1.0 | Last Updated: December 2025

For DevOps Team - Keep This Document Secure