

Despliegue de OpenEMR

Instrucciones técnicas del despliegue de la plataforma OpenEMR

- [Instalación](#)
- [Creación de Pacientes de Prueba](#)

Instalación

1. Requisitos Previos (Prerequisites)

Antes de que el usuario clone o descargue nada, debe saber si su sistema es compatible.

- **Sistema Operativo:** Linux
- **Software/Herramientas:** Versiones específicas de lenguajes (ej. Node.js v24, Composer 2.7.1)
- **Gestores de paquetes o bases de datos:** Docker, Composer, nodejs, npm

“ **Tip:** Si es posible, incluye los enlaces oficiales para descargar estos requisitos.

2. Clonación o Descarga del Proyecto

El primer paso práctico. Indica cómo obtener el código fuente.

```
git clone https://github.com/openemr/openemr.git
git checkout v8_0_0_3
```

3. Configuración del Entorno (Environment Setup)

```
sudo apt install composer
sudo apt install nodejs npm
sudo npm install -g n
sudo n 24
composer install --no-dev
npm install
npm run build
```

3.1 Configuración de Compose

Actualizar el archivo **openemr/docker/development-easy/docker-compose.yml** con la **imagen de openemr**(línea 29)

```
services:
  mysql:
    restart: always
    image: mariadb:11.8
    command: ['mariadb', '--character-set-server=utf8mb4', '--ssl-ca=/etc/ssl/ca.pem', '--ssl_cert=/etc/ssl/server-cert.pem', '--ssl_key=/etc/ssl/server-key.pem']
    ports:
      - 8320:3306
    volumes:
      - ../library/sql-ssl-certs-keys/easy/ca.pem:/etc/ssl/ca.pem:ro
      - ../library/sql-ssl-certs-keys/easy/server-cert.pem:/etc/ssl/server-cert.pem:ro
      - ../library/sql-ssl-certs-keys/easy/server-key.pem:/etc/ssl/server-key.pem:ro
      - databasevolumev8:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: root
    healthcheck:
      test:
        - CMD
        - /usr/local/bin/healthcheck.sh
        - --su-mysql
        - --connect
        - --innodb_initialized
      start_period: 1m
      start_interval: 10s
      interval: 1m
      timeout: 5s
      retries: 3
  openemr:
    restart: always
    image: openemr/openemr:flex //Actualizar
    ports:
      - 8300:80
      - 9300:443
    volumes:
      - ${OPENEMR_DIR:-../..}:/openemr:ro
      - ${OPENEMR_DIR:-../..}:/var/www/localhost/htdocs/openemr:rw
      - assetvolumev8:/var/www/localhost/htdocs/openemr/public/assets:rw
```

```

- themevolumev8:/var/www/localhost/htdocs/openemr/public/themes:rw
- sitesvolumev8:/var/www/localhost/htdocs/openemr/sites:rw
- nodemodulesv8:/var/www/localhost/htdocs/openemr/node_modules:rw
- vendordirv8:/var/www/localhost/htdocs/openemr/vendor:rw
- ccdanodemodulesv8:/var/www/localhost/htdocs/openemr/ccdaservice/node_modules:rw
- ccdanodemodules2v8:/var/www/localhost/htdocs/openemr/ccdaservice/packages/oe-cqm-
service/node_modules:rw
- logvolumev8:/var/log
- couchdbvolumev8:/couchdb/data
environment:
  DEBUG_COLORS: "true"
  TERM: xterm-256color
  COLORTERM: truecolor
  MYSQL_HOST: mysql
  MYSQL_ROOT_PASS: root
  MYSQL_USER: openemr
  MYSQL_PASS: openemr
  OE_USER: admin
  OE_PASS: pass
  EASY_DEV_MODE: "yes"
  EASY_DEV_MODE_NEW: "yes"
  DEVELOPER_TOOLS: "yes"
  XDEBUG_ON: 1
  XDEBUG_PROFILER_ON: 1
  # setting xdebug client host for cases where xdebug.discover_client_host fails
  XDEBUG_CLIENT_HOST: host.docker.internal
  GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0
  GITHUB_COMPOSER_TOKEN_ENCODED:
Z2hwX0NpbmJ4aXlnN0NzcGZXWG1UWFUwcXhGa040e1FKSDJoZGJXVw==
  GITHUB_COMPOSER_TOKEN_ENCODED_ALTERNATE: '103 104 112 95 57 54 108 115 88 116 87 72
51 75 81 105 69 88 88 119 97 79 80 78 109 69 66 115 85 97 106 78 112 71 49 81 90 102 74
121'
  SELENIUM_USE_GRID: "true"
  SELENIUM_HOST: selenium
  SELENIUM_FORCE_HEADLESS: "false"
  SELENIUM_BASE_URL: "http://openemr"
  OPENEMR_DOCKER_ENV_TAG: easy-dev-docker
  OPENEMR_SETTING_site_addr_oath: 'https://localhost:9300'

```

```
OPENEMR_SETTING_oauth_password_grant: 3
OPENEMR_SETTING_rest_system_scopes_api: 1
OPENEMR_SETTING_rest_api: 1
OPENEMR_SETTING_rest_fhir_api: 1
OPENEMR_SETTING_rest_portal_api: 1
OPENEMR_SETTING_portal_onsite_two_enable: 1
OPENEMR_SETTING_ccda_alt_service_enable: 3
OPENEMR_SETTING_couchdb_host: couchdb
OPENEMR_SETTING_couchdb_port: 6984
OPENEMR_SETTING_couchdb_user: admin
OPENEMR_SETTING_couchdb_pass: password
OPENEMR_SETTING_couchdb_dbase: example
OPENEMR_SETTING_couchdb_ssl_allow_selfsigned: 1
OPENEMR_SETTING_gbl_ldap_host: 'ldap://openldap:389'
OPENEMR_SETTING_gbl_ldap_dn: 'cn={login},dc=example,dc=org'
OPENEMR_SETTING_EMAIL_METHOD: SMTP
OPENEMR_SETTING_SMTP_HOST: mailpit
OPENEMR_SETTING_SMTP_PORT: 1025
OPENEMR_SETTING_SMTP_USER: openemr
OPENEMR_SETTING_SMTP_PASS: openemr
OPENEMR_SETTING_SMTP_SECURE: ''
OPENEMR_SETTING_SMTP_Auth: 'TRUE'
OPENEMR_SETTING_practice_return_email_path: 'noreply@openemr.local'
OPENEMR_SETTING_Patient_Reminder_Sender_Name: 'OpenEMR Reminders'
depends_on:
  mysql:
    condition: service_healthy
healthcheck:
  test:
    - CMD
    - /usr/bin/curl
    - --fail
    - --insecure
    - --location
    - --show-error
    - --silent
    - https://localhost/meta/health/readyz
  start_period: 3m
```

```
    start_interval: 10s
    interval: 1m
    timeout: 5s
    retries: 3
selenium:
  restart: always
  image: selenium/standalone-chromium:4.40.0
  ports:
    - 4444:4444    # Selenium WebDriver interface
    - 7900:7900    # VNC port for viewing browser during tests (to work, will require
SELENIUM_FORCE_HEADLESS to be "false" in openemr service)
  environment:
    SE_VNC_PASSWORD: openemr123
    SE_SCREEN_WIDTH: 1920
    SE_SCREEN_HEIGHT: 1080
    SE_SCREEN_DEPTH: 24
    SE_START_XVFB: true
  volumes:
    - /dev/shm:/dev/shm
  healthcheck:
    test: ["CMD", "curl", "-f", "http://localhost:4444/wd/hub/status"]
    start_period: 30s
    interval: 30s
    timeout: 5s
    retries: 3
phpmyadmin:
  restart: always
  image: phpmyadmin
  ports:
    - 8310:80
  environment:
    PMA_HOSTS: mysql
  depends_on:
    mysql:
      condition: service_healthy
couchdb:
  restart: always
  image: couchdb
```

```
ports:
- 5984:5984
- 6984:6984
volumes:
- ../library/couchdb-config-ssl-cert-keys/local.ini:/opt/couchdb/etc/local.ini:rw
- ../library/couchdb-config-ssl-cert-keys/easy/ca.pem:/etc/ssl/ca.pem:ro
- ../library/couchdb-config-ssl-cert-keys/easy/server-cert.pem:/etc/ssl/server-
cert.pem:ro
- ../library/couchdb-config-ssl-cert-keys/easy/server-key.pem:/etc/ssl/server-
key.pem:ro
- couchdbvolumev8:/opt/couchdb/data
environment:
  COUCHDB_USER: admin
  COUCHDB_PASSWORD: password
openldap:
restart: always
image: openemr/dev-ldap:easy
environment:
  LDAP_TLS_VERIFY_CLIENT: try
  LDAP_TLS_CA_CRT_FILENAME: ca.pem
  LDAP_TLS_CRT_FILENAME: server-cert.pem
  LDAP_TLS_KEY_FILENAME: server-key.pem
mailpit:
restart: unless-stopped
image: axllent/mailpit:latest
ports:
- 8025:8025    # Web UI and API
- 1025:1025    # SMTP server
environment:
  MP_SMTP_AUTH_ACCEPT_ANY: 1
  MP_SMTP_AUTH_ALLOW_INSECURE: 1
  MP_MAX_MESSAGES: 5000
  MP_DATABASE: /data/mailpit.db
volumes:
- mailpitvolumev8:/data
healthcheck:
  test: ["CMD", "wget", "--quiet", "--tries=1", "--spider",
"http://localhost:8025/api/v1/messages"]
```

```
    start_period: 10s
    interval: 30s
    timeout: 5s
    retries: 3
volumes:
  databasevolumev8: {}
  assetvolumev8: {}
  themevolumev8: {}
  sitesvolumev8: {}
  nodemodulesv8: {}
  vendordirv8: {}
  ccdanodemodulesv8: {}
  ccdanodemodules2v8: {}
  logvolumev8: {}
  couchdbvolumev8: {}
  mailpitvolumev8: {}
```

4. Ejecución de la Aplicación (Running the App)

```
docker compose -f docker/development-easy/docker-compose.yml up -d
```

4.1 Acceso

- url: <https://localhost:9300/>
- usuario: admin
- password: pass

Creación de Pacientes de Prueba

Prerequisitos: [Instalación de OpenEMR](#)

Para crear 100 pacientes con datos.

```
docker exec -it development-easy-openemr-1 bash
apk add --upgrade openjdk17-jre
/root/devtools import-random-patients 100
```