Despliegue y Operación – ETL Meta Ads (SOLVO)

Despliegue y Operación – ETL Meta Ads (SOLVO)  
Última actualización: 2025-08-12 11:10

1. Propósito  
Este documento describe cómo desplegar y operar el proceso ETL que extrae métricas de Meta Ads y las inserta en la BBDD de Lepho. Incluye requisitos, preparación del entorno, configuración de variables de entorno, pruebas, programación (cada 30 minutos) y pautas de operación.

2. Estructura de archivos  
- API\_META\_SCRIPT\_SOLVO.py → lógica ETL (extracción, carga por lotes, control de sincronización)  
- RUN.py → punto de entrada  
- requirements.txt → dependencias de Python  
- .env (no incluido) → claves y parámetros sensibles

3. Requisitos del servidor  
- Python 3.x con acceso a Internet (Graph API) y conectividad de red a Artemisa (BBDD Lepho).  
- Acceso a MySQL con permisos de lectura/escritura sobre el esquema destino.  
- Posibilidad de programar la ejecución cada 30 minutos (cron en Linux o Task Scheduler en Windows).

4. Preparación del entorno (venv + dependencias)  
En el directorio del proyecto:  
 - Crear y activar entorno virtual (venv).  
 - Instalar dependencias del requirements.txt.

Linux/Mac:  
 python -m venv venv  
 source venv/bin/activate  
 pip install -r requirements.txt

Windows (PowerShell):  
 python -m venv venv  
 .\venv\Scripts\Activate.ps1  
 pip install -r requirements.txt

5. Variables de entorno (.env)  
Tambien se pasará un archivo .env con la siguiente estructura. Necesario que este en la carpeta raíz del proyecto.  
 DECRYPTION\_KEY=\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 DB\_HOST=lepho-host  
 DB\_PORT=3306  
 DB\_NAME=digital  
 DB\_USER=usuario  
 DB\_PASSWORD=\*\*\*\*\*\*\*\*\*  
Notas:  
- Usamos cifrado AES para las credenciales almacenadas (tabla api\_credentials\_solvo). El valor DECRYPTION\_KEY debe coincidir con el usado al insertar la configuración.  
- En entornos con proxy/inspección TLS, puede requerirse REQUESTS\_CA\_BUNDLE/SSL\_CERT\_FILE apuntando al bundle de CAs corporativas.

6. Prueba manual  
Con el venv activado:  
 python RUN.py  
Verifique en logs la marca de tiempo de time\_update\_solvo y registros insertados/actualizados.

7. Programación cada 30 minutos  
Linux (cron):  
 crontab -e  
 \*/30 \* \* \* \* /ruta/proyecto/venv/bin/python /ruta/proyecto/RUN.py >> /var/log/solvo\_meta.log 2>&1

Windows (Task Scheduler, línea de comando):  
 schtasks /create /tn "Solvo Meta ETL" /sc minute /mo 30 ^  
 /tr "\"C:\ruta\proyecto\venv\Scripts\python.exe\" C:\ruta\proyecto\RUN.py" ^  
 /f

8. Operación  
- Logs: redirigir a fichero (ej. /var/log/solvo\_meta.log) o revisar historial del Programador de Tareas.  
- Control de sincronización: la tabla time\_update\_solvo guarda el último timestamp de ejecución correcta.  
- Cargas rápidas: el ETL usa inserciones por lotes (executemany / multi-VALUES) con ON DUPLICATE KEY UPDATE para acelerar la escritura.  
- Rate limiting: el ETL implementa backoff ante throttling del Marketing API y reduce llamadas mediante filtros incrementales (updated\_time).

9. Seguridad  
- Mantener .env fuera de control de código y con permisos restringidos.  
- Rotar tokens/secretos según la política interna (Esto lo hago yo de manera remota).  
- Validar que el servidor tenga acceso a Artemisa únicamente desde cuentas/segmentos autorizados.

10. Solución de problemas  
- Certificados (SSLError self-signed): configurar bundle de CAs corporativas (REQUESTS\_CA\_BUNDLE / SSL\_CERT\_FILE).  
- FKs en MySQL: se incluyen triggers de “stub” para campañas/adsets/ads que garantizan integridad referencial en cargas incrementales.  
- Rendimiento: verificar max\_allowed\_packet si se sube el tamaño de lote; revisar índices/PK en tablas DIM/FACT.

11. Anexos  
A. Ejemplo de .env (plantilla mínima) ubicado en la carpeta del proyecto.  
 DECRYPTION\_KEY=\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 DB\_HOST=lepho-host  
 DB\_PORT=3306  
 DB\_NAME=digital  
 DB\_USER=usuario  
 DB\_PASSWORD=\*\*\*\*\*\*\*\*\*  
 # Opcional en redes con inspección TLS corporativa  
 # REQUESTS\_CA\_BUNDLE=C:\certs\corp\_cabundle.pem  
 # SSL\_CERT\_FILE=C:\certs\corp\_cabundle.pem

B. Referencias  
- Python venv (entorno virtual) – documentación oficial.  
- Cron (crontab) – manual.  
- Programación de tareas en Windows (schtasks) – documentación oficial.  
- Meta Marketing API – referencia.

Contacto  
Cualquier duda, por favor contactar con David Zerpa: jose.batista@becallgroup.com