Lloyd's Market News Digest Deployment Guide

Target host: `automation-runner-01`

VM details: `192.168.1.30` / `100.90.70.30`

Schedule: Daily at `08:00` (Europe/London)

1. Prerequisites

- Ubuntu/Debian host with `sudo` access
- Docker running (for existing local Postgres container)
- MongoDB Atlas connection string
- OpenAI API key
- GitHub repo write access (SSH key or PAT)

2. Connect to the VM

```
ssh <user>@100.90.70.30
```

3. Install system packages

```
sudo apt-get update
sudo apt-get install -y git python3 python3-venv python3-pip postgresql-client cron
sudo systemctl enable --now cron
```

4. Clone repository

```
sudo mkdir -p /opt/automation
sudo chown -R "$USER":"$USER" /opt/automation
cd /opt/automation
git clone https://github.com/poovannanrajendran/lloyds-market-news-digest.git
cd lloyds-market-news-digest
```

5. Create Python virtual environment

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip wheel setuptools
pip install -r requirements.txt
```

6. Configure environment file

Copy template and fill values:

```
cp .env.example .env
```

Required settings for your topology:

```
# MongoDB Atlas
MONGODB_URI="<atlas-uri>"
MONGO_DB_NAME="lloyds_digest_raw"
# Local Postgres container (same VM)
```

```
POSTGRES_HOST="127.0.0.1"
POSTGRES_PORT="5432"
POSTGRES_DB="lloyds_digest"
POSTGRES_USER="<postgres-user>"
POSTGRES_PASSWORD="<postgres-password>"

# OpenAI models and Flex service tier
OPENAI_API_KEY="<openai-api-key>"
OPENAI_MODEL="gpt-5-mini"
LLOYDS_DIGEST_LLM_RELEVANCE_MODEL="gpt-5-nano"
LLOYDS_DIGEST_LLM_CLASSIFY_MODEL="gpt-5-nano"
LLOYDS_DIGEST_LLM_SUMMARISE_MODEL="gpt-5-mini"
OPENAI_SERVICE_TIER="flex"
OPENAI_LINKEDIN_MODEL="gpt-5-mini"
OPENAI_LINKEDIN_SERVICE_TIER="flex"

# Optional but recommended
GITHUB_PAGES_BASE_URL="https://poovannanrajendran.github.io/lloyds-market-news-digest/digests/"
SMTP_ENABLED="false"
```

7. Verify Postgres container is reachable

```
docker ps
ss -ltnp | grep 5432
```

If port `5432` is not exposed to host, publish it in Docker first.

8. Initialize Postgres schema

```
set -a
source .env
set +a
bash scripts/db_init_postgres.sh
```

9. Initialize MongoDB collections and indexes

Install `mongosh` if needed, then run:

```
MONGO_DB_NAME="$MONGO_DB_NAME" mongosh "$MONGODB_URI" --file scripts/db_init_mongo.js
```

10. Configure Git for automated publish

```
git config user.name "automation-runner-01"
git config user.email "<your-email>"
```

Recommended auth for cron:

```
git remote set-url origin git@github.com:poovannanrajendran/lloyds-market-news-digest.git
ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519_lloyds -N ""
cat ~/.ssh/id_ed25519_lloyds.pub
```

Add that public key as a write-enabled deploy key on the repository, then test:

```
ssh -T git@github.com
```

11. Smoke test database connectivity

```
source .venv/bin/activate
export PYTHONPATH="$PWD/src"
python scripts/smoke_test_connections.py
```

Expected output includes:

- `Postgres OK`
- `Mongo OK`

12. Run full daily flow once manually

```
source .venv/bin/activate
export PYTHONPATH="$PWD/src"
./scripts/run_daily.sh
```

This script runs:

- pipeline (`python -m lloyds_digest run --now --verbose`)
- digest render (`scripts/render_digest_llm_compare.py`)
- GitHub Pages publish (`scripts/publish_github_pages.sh`)
- LinkedIn post render (`scripts/render_linkedin_post.py`)
- dashboard render (`scripts/render_run_dashboard.py`)

13. Validate outputs

- `output/digest_YYYY-MM-DD.html` exists
- `docs/digests/digest_YYYY-MM-DD.html` exists
- `docs/index.html` updated
- `logs/run_YYYY-MM-DD.log` exists

14. Configure cron for 8:00 AM daily

Edit crontab:

```
crontab -e
```

Add:

```
CRON_TZ=Europe/London
0 8 * * * cd /opt/automation/lloyds-market-news-digest && /bin/bash -lc 'source .venv/bin/activate &
```

Verify:

```
crontab -l
```

15. Operational checks

- Confirm GitHub Pages reflects new digest after run
- Check `output/dashboard/index.html` for run metrics and failures
- Rotate API keys immediately if ever exposed

- Keep `.env` out of version control

Troubleshooting quick checks

- OpenAI errors: verify `OPENAI_API_KEY`, model names, and `OPENAI_SERVICE_TIER=flex`
- Postgres errors: verify host/port/user/password and Docker port mapping
- Mongo errors: verify Atlas URI and IP allowlist
- Git push failures: verify deploy key or token auth for non-interactive cron