Instructions to Run - AWS Localstack Content Moderation

July 30, 2025

Prerequisites (verify, don't install)

- **Docker** running locally (Desktop or Engine)
- **Python 3.11+** available in `PATH`
- **Git** CLI (If to be cloned via GIT)

Use two terminals for the following steps: Terminal 1 runs project commands; Terminal 2 runs LocalStack. All commands are intended for **Linux** or **macOS** environments.

Terminal 1 – Create & Activate Virtual Env
Creates a dedicated Python virtual environment and installs the project dependencies.

```
python -m venv venv_DIC_Assignment3
source venv_DIC_Assignment3/bin/activate
pip install -r requirements.txt
```

Terminal 2 – Activate Virtual Env & Start LocalStack
Activates the same environment and starts LocalStack, which emulates AWS services; leave this terminal running.

```
source venv_DIC_Assignment3/bin/activate
LOCALSTACK_ACTIVATE_PRO=0 LOCALSTACK_DEBUG=1 localstack start
```

Terminal 1 – Deploy Lambdas and Resources
Runs the provisioning script to deploy Lambda stubs, create SSM parameters, provision the `reviews-input` bucket, the tables `reviews`, `sentiment` and `users` and configure all event triggers.

```
python scripts/setup_resources.py
```

Terminal 1 – Sanity Check (optional)
Verifies that the bucket, functions and table exist.

```
awslocal s3 ls
awslocal lambda list-functions
awslocal dynamodb list-tables
```

Terminal 1 – Choose one of the following two options
You can either run the full dev-set (optional) *or* upload a single test review. Both will trigger the pipeline.

Option A – run `reviews_devset.json` through pipeline
Processes either 20% or the entire dev-set for bulk testing.

```
python scripts/prepare_devset.py    # only needs to be run once initially
python scripts/run_devset.py 20%    # or use the command below to run the full
↪  dev-set
python scripts/run_devset.py all
```

Option B – Upload a sample review (recommended for quick test)
Triggers the pipeline with a single review, ideal for quick verification.

```
awslocal s3 cp test_review.json s3://reviews-input/sample_review.json
```

Confirm DynamoDB Ingestion
Checks that the review appears in the `reviews` table.

```
awslocal dynamodb scan --table-name reviews
```

You should see an item whose `reviewId` matches `sample_review.json`, confirming the pipeline is operational.

Terminal 1 – Summarise results
Runs the final analysis script to count the number of reviews tagged as `bad`, `neutral` and `good` and reports how many users were banned.

```
python scripts/get_results.py
```

Terminal 1 – Run Integration Test
Sets temporary environment variables to allow local AWS access and runs integration tests with verbose output.

```
AWS_ACCESS_KEY_ID=test \
AWS_SECRET_ACCESS_KEY=test \
AWS_DEFAULT_REGION=us-east-1 \
```

```
AWS_ENDPOINT_URL=http://localhost:4566 \
pytest -m integration -v
```