

Build and Deploy Documentation

SightScan | AMOS Project Team 2

SERVICES

Crawler

The crawler is responsible of crawling images for a given city and inserting them into the data warehouse.

Environment Variables

PGHOST → ****URL of the data warehouse

PGDATABASE → ****database name you want to access

PGPORT → **port** of the data warehouse

PGUSER → **user** of the data warehouse

PGPASSWORD → **password** of the data warehouse

apikey → **open maps api key**

maps_key → **open maps key**

Docker

- `cd ./amos/crawler`
- `docker build -t crawler:v1 .`
- `docker run -e PGHOST=<PGHOST> -e PGDATABASE=<PGDATABASE> -e PGPORT=<PGPORT> -e PGUSER=<PGUSER> -e PGPASSWORD=<PGPASSWORD> -e apikey=<apikey> -e maps_key=<maps_key> crawler:v1 "city_name"`

Locally

1. `cd ./amos/crawler`
 2. `pip install -r requirements.txt`
 3. `python main.py --<argument>=<argument_value>`
- **Command Line Arguments**
 - **--skip=true** Skips keyword if downloaded directory already exists. This is needed when re-downloading.

BUILD AND DEPLOY DOCUMENTATION

SightScan | AMOS Project Team 2

--threads=4 Number of threads to download.
--google=true Download from [google.com](https://www.google.com) (boolean)
--full=false Download full resolution image instead of thumbnails (slow)
--face=false Face search mode
--no_gui=auto No GUI mode. (headless mode) Acceleration for full_resolution mode, but unstable on thumbnail mode. Default: "auto" - false if full=false, true if full=true (can be used for docker linux system)
--limit=0 Maximum count of images to download per site. (0: infinite)
--no_driver=false Whether a driver should be used
--location='Berlin' The location keywords need to be found for.
--sights_limit The limit of sights to be found by the collector api

IMPORTANT

You need an .env-file in the **respective directory** to run it **locally**. Otherwise the env-variables won't be readable for the program.

Data Mart Refresher

The Data Mart Refresher is responsible for three things. First checking if enough labeled images are available in the DWH if there are it starts a model training process, if not it starts the image labeling process and updating our data marts.

Data Marts can be seen in the technical documentation

Environment Variables

DATA_MART_MTS_ENDPOINT_URL → URL of the data warehouse.

DATA_MART_ILS_ENDPOINT_URL → Database name you want to access.

DATA_MART_REFRESH_DATA_MARTS_EVERY_SECONDS → Interval of seconds to refresh the data marts

DATA_MART_ENABLE_MODEL_TRAINING_EVERY_SECONDS → Interval of seconds to check and trigger Model Training Service

DATA_MART_ENABLE LABELLING_REQUESTS_EVERY_SECONDS → Interval of seconds to check and trigger Image Labeling Service

PGPORT → **Port** of the data warehouse

PGUSER → **User** of the data warehouse

PGPASSWORD → **Password** of the data warehouse

Docker

BUILD AND DEPLOY DOCUMENTATION

SightScan | AMOS Project Team 2

1. `cd ./amos/data_mart_refresher`
2. `docker build -t data_mart_refresher:v1 .`
3. `docker run -e DATA_MART_MTS_ENDPOINT_URL=<DATA_MART_MTS_ENDPOINT_URL> -e DATA_MART_ILS_ENDPOINT_URL=<DATA_MART_ILS_ENDPOINT_URL> -e DATA_MART_REFRESH_DATA_MARTS_EVERY_SECONDS=5 -e DATA_MART_ENABLE_MODEL_TRAINING_EVERY_SECONDS=10 -e DATA_MART_ENABLE LABELLING REQUESTS EVERY_SECONDS=10 -e PGHOST=<PGHOST> -e PGDATABASE=<PGDATABASE> -e PGPORT=<PGPORT> -e PGUSER=<PGUSER> -e PGPASSWORD=<PGPASSWORD> data_mart_refresher:v1`

IMPORTANT

If you have environment variables in an .env-file you can remove the variables from the docker command.

Locally

1. `cd ./amos/data_mart_refresher`
2. `pip install -r requirements.txt`
3. `python ./cron_jobs/main.py`

IMPORTANT

You need an .env-file in the **respective directory** to run it **locally**. Otherwise the env-variables won't be readable for the program.

Postgres

Is an open source object-relational database, which runs our data warehouse.

Docker

1. `cd ./postgres`
2. `docker build -t postgres:v1 .`
3. `docker run postgres:v1 -p 5432:5432`

Important

If you run a PostgreSQL without our Dockerfile, run the `database_init.sql` file to initialise the data warehouse.

Deployment

- The data warehouse is deployed using Amazon RDS and creating a PostgreSQL database instance using the free tier version from AWS. It is as se

BUILD AND DEPLOY DOCUMENTATION

SightScan | AMOS Project Team 2

The screenshot shows the AWS 'Create database' console. The 'Choose a database creation method' section has 'Easy create' selected. The 'Configuration' section shows 'PostgreSQL' selected as the engine type. Under 'DB instance size', the 'Free tier' option is selected, which provides 1 vCPU, 1 GiB RAM, and 20 GiB storage for 0.029 USD/hour. The 'DB instance identifier' is set to 'database-1', the 'Master username' is 'postgres', and 'Auto generate a password' is checked. The 'Master password' field is empty.

Create database

Choose a database creation method [Info](#)

- ☐ Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- ☒ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Configuration

Engine type [Info](#)

- ☐ Amazon Aurora
- ☐ MySQL
- ☐ MariaDB
- ☒ PostgreSQL
- ☐ Oracle
- ☐ Microsoft SQL Server

DB instance size

- ☐ Production
db.r4.large
2 vCPUs
15.25 GiB RAM
500 GiB
0.797 USD/hour
- ☐ Dev/Test
db.m4.large
2 vCPUs
8 GiB RAM
100 GiB
0.236 USD/hour
- ☒ Free tier
db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB
0.029 USD/hour

DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-1
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)
Type a login ID for the master user of your DB instance.
postgres
1 to 16 alphanumeric characters. First character must be a letter

☒ Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

1. Simply choose PostgreSQL as a Database engine and the Free tier version

BUILD AND DEPLOY DOCUMENTATION

SightScan | AMOS Project Team 2

PostgreSQL

DB instance size

- ☐ Production
 - db.r4.large
 - 2 vCPUs
 - 15.25 GiB RAM
 - 500 GiB
 - 0.797 USD/hour
- ☐ Dev/Test
 - db.m4.large
 - 2 vCPUs
 - 8 GiB RAM
 - 100 GiB
 - 0.236 USD/hour
- ☒ Free tier
 - db.t2.micro
 - 1 vCPUs
 - 1 GiB RAM
 - 20 GiB
 - 0.029 USD/hour

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)

Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm password [Info](#)

► **View default settings for Easy create**

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

[Cancel](#) [Create database](#)

2. Add your dataset password and username and create the database.
3. Once the database is created connect to the database via the credentials and run the database_init.sql file.

Desktop App

The desktop app, is the user interface for the SightScan project it allows to interact with images and download trained models to use them for identifying images.

- Locally
 1. `cd ./amos/gui`
 2. `pip install -r requirements.txt`
 3. `python main.py`

Sight Detector

The Sight Detector is our Model Training Service which uses a YOLOv5 algorithm to detect sights in a given image.

- Locally
 1. `cd ./amos/sight_detector`
 2. `pip install -r requirements.txt`
 3. `python -m pip install -r requirements.txt`
 4. `python detect.py --weights weights/best.pt --save-txt`

DEPLOYMENT

Right now deployment is happening locally, since we don't have the correct configured access to AWS to deploy our docker images.