
TerraVisu

Release 2023.4.3

Autonomens

Apr 13, 2023

CONTENTS:

1	Install instructions	3
1.1	Requirements	3
1.2	Install	3
1.3	Update	4
2	Configuration	5
2.1	Environment variables	5
3	Usage	9
3.1	Configure instance settings	9
3.2	Sources, layers and views	9
4	Development	11
4.1	Prepare stack	11
4.2	Init database	11
4.3	Load initial data	11
4.4	Create your superuser	11
4.5	Prepare admin if required	11
4.6	Prepare frontend if required	12
4.7	Launch stack	12
4.8	Access	12
4.9	Linting	12
5	Troubleshooting	13
5.1	Elastic search container doesn't start	13
6	CHANGELOG	15
6.1	2023.4.3 (2023-04-13)	15
6.2	2023.4.2 (2023-04-11)	15
6.3	2023.4.1 (2023-04-07)	15
7	Indices and tables	17
	Index	19



La visualisation de données cartographiques
au service de vos applications métiers

INSTALL INSTRUCTIONS

1.1 Requirements

- **You need docker installed. Compose plugin is recommended in the configuration below.**
See [Docker](#).
- **Optional** : if you want to use external database, prepare a postgresql 11+ (15 recommended) postgres2.5 (3.3 recommended) database with postgis enabled, and a dedicated user.

You can use external database by commenting postgres container and volume references in docker-compose.yml, and set variables in your conf/visu.env file :

- POSTGRES_HOST
- POSTGRES_PORT
- POSTGRES_USER
- POSTGRES_PASSWORD
- POSTGRES_DB

Add local IPs in *pg_hba.conf* to allow connection from docker containers to your database.

- You can use external nginx proxy. Edit provided nginx conf file and comment nginx references in docker-compose.yml. Fix web:8000 to 127.0.0.1:8000 in nginx.conf.

1.2 Install

- Download [zip package](#)
- Unzip it where you want

```
unzip install.zip  
cd terra_visu
```

- Prepare environment variables

```
./conf/visu.env
```

-> Set or change all required values

at least:

- ALLOWED_HOST # list of your final host(s), comma separated values

- SECRET_KEY # unique key for your project. See <https://djecrety.ir/>
- POSTGRES_USER # a dedicated user for your database
- POSTGRES_PASSWORD # a dedicated password for your database
- Pull images

```
docker compose pull
```

- Init database and project config

```
docker compose run --rm web update.sh
```

- Create your super user

```
docker compose run --rm web ./manage.py createsuperuser
```

- Load initial data

```
docker compose run --rm web ./manage.py loaddata project/fixtures/initial.json
```

- Launch stack

```
docker compose up -d
```

- ... and access to TerraVisu

```
http://<your_domain>/
```

You can change port mapping by using a .env file in terra_visu directory :

```
# .env  
NGINX_PORT=8080
```

1.3 Update

- Read [release notes](#) about bugfix, news and breaking changes.
- Backup your data (database, public/media and var/ folder)
- Pull latest image

```
docker compose pull
```

- Run post update script

```
docker compose run --rm web update.sh
```

- Relaunch your stack

```
docker compose down  
docker compose up -d
```


CONFIGURATION

2.1 Environment variables

Add your environment variables in `app.env` file.

2.1.1 General

ALLOWED_HOSTS

domains allowed to be used by your instance. Support comma separated values.

Example:

```
ALLOWED_HOSTS=mysite.fr # ALLOWED_HOSTS=mysite.fr,my.other.site.fr
```

SECRET_KEY

unique secret key for your instance. (<https://djecrety.ir/>)

Example:

```
SECRET_KEY=zbesj@t3_&u75&l=xk@ftg1yh4wy)i)9!z+(v$ig7*-*lkd6om
```

SSL_ENABLED

Set true if your site is behind ssl proxy.

Example:

```
SSL_ENABLED=True
```

Default:

```
False
```

2.1.2 OIDC Connect

To allow OIDC login, you should configure these settings.

OIDC_ENABLE_LOGIN

Enable OIDC connect login.

Example:

```
OIDC_ENABLE_LOGIN=True
```

Default:

```
False
```

OIDC_DISABLE_INTERNAL_LOGIN

Disable internal login if OIDC enabled. (direct redirection to OIDC login)

Example:

```
OIDC_DISABLE_INTERNAL_LOGIN=True
```

Default:

```
False
```

OIDC_AUTH_SERVER

Set your OIDC Realm URL.

Example:

```
OIDC_AUTH_SERVER=https://your.openid.com/realms/master
```

OIDC_AUTH_CLIENT_ID

Set your OIDC Client ID.

Example:

```
OIDC_AUTH_CLIENT_ID=your-client-id
```

OIDC_AUTH_CLIENT_SECRET

Set your OIDC Client secret.

Example:

```
OIDC_AUTH_CLIENT_SECRET=7GcKm7XiWIE6BRscGHZZku
```

OIDC_AUTH_SCOPE

Set your OIDC Client scope. Support comma separated values.

Example:

```
OIDC_AUTH_SCOPE=openid,email
```

Default:

```
openid
```

2.1.3 SENTRY

SENTRY_DSN

Set your SENTRY_DSN to enable sentry reporting.

Example:

```
SENTRY_DSN=https://your.sentry/dsn
```

Default:

```
None
```

SENTRY_TRACE_SAMPLE_RATE

Specify sample rate for your performance tracking.

Example:

```
SENTRY_TRACE_SAMPLE_RATE=1.0
```

Default:

```
0.2
```

SENTRY_SEND_DEFAULT_PII

Specify if sentry enable user informations.

Example:

```
SENTRY_SEND_DEFAULT_PII=False
```

Default:

```
True
```

2.1.4 API Schemas

API_SCHEMA

Set true if you want to expose API openapi schema. It expose /api/schema/ endpoint.

Example:

```
API_SCHEMA=True
```

Default:

```
False
```

API_SWAGGER

Set true if you want to expose API swagger. API_SCHEMA should be enabled. It expose /api/schema/swagger/ endpoint.

Example:

API_SWAGGER=**True**

Default:

False

API_REDOC

Set true if you want to expose API redoc. API_SCHEMA should be enabled. It expose /api/schema/redoc/ endpoint.

Example:

API_REDOC=**True**

Default:

False

3.1 Configure instance settings

- Need to have SuperUser privileges
- Go to `/config/`

3.2 Sources, layers and views

- Need to have required privileges
- Go to `/admin/`

3.2.1 Configure map base layers

3.2.2 Add data sources

3.2.3 Configure views and layers

DEVELOPMENT

4.1 Prepare stack

```
cp db.env.dist db.env
cp app.env.dist app.env
docker compose build
```

4.2 Init database

```
docker compose run --rm web ./manage.py migrate
```

4.3 Load initial data

```
docker compose run --rm web ./manage.py loaddata project/fixtures/initial.json
```

4.4 Create your superuser

```
docker compose run --rm web ./manage.py createsuperuser
```

4.5 Prepare admin if required

```
make build_admin
```

4.6 Prepare frontend if required

```
make build_front
```

4.7 Launch stack

```
docker compose up
```

4.8 Access

4.8.1 Frontend

<http://visu.localhost:8080>

4.8.2 Admin

<http://visu.localhost:8080/admin/>

4.8.3 Django admin (config / debug)

<http://visu.localhost:8080/config/>

4.9 Linting

We use flake8, isort and black rules. You can run :

```
make lint
```

to check them

TROUBLESHOOTING

5.1 Elastic search container doesn't start

If you have:

bootstrap check failure [1] of [1]: max virtual memory areas vm.max_map_count [xxx] is too low, increase to at least [yyy]

Then you need to increase the vm.max_map_count on your host machine.

```
sudo nano /etc/sysctl.conf  
vm.max_map_count=262144
```

Then reboot your machine.

CHANGELOG

6.1 2023.4.3 (2023-04-13)

Improvements:

- Allow to set group access to extra menu items

6.2 2023.4.2 (2023-04-11)

New features:

- Allow using style images patterns in polygon advanced styles

6.3 2023.4.1 (2023-04-07)

New Version

New Simplified Installation

New documentation

Bug fixes:

- Fix and allow date usage in source fields and imported data
- Fix group creation / edition in admin
- Fix LayerTree cache management
- Fix bug when no base layer defined in scene (#109)

New features:

- Use icon and patterns in point / polygon styles

Improvements:

- Direct use elasticsearch connector for data indexation instead of terra-bonobo-nodes
- Better layer duplication
- Some instance configuration managed in config panel (/config/)

Maintenance

- From Python 3.6 to 3.10

- From Django 2.2 to 4.1
- All python packages updated
- Admin node-js from 12 to 18

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

E

environment variable

- ALLOWED_HOSTS, 5
- API_REDOC, 8
- API_SCHEMA, 7
- API_SWAGGER, 7
- OIDC_AUTH_CLIENT_ID, 6
- OIDC_AUTH_CLIENT_SECRET, 6
- OIDC_AUTH_SCOPE, 6
- OIDC_AUTH_SERVER, 6
- OIDC_DISABLE_INTERNAL_LOGIN, 6
- OIDC_ENABLE_LOGIN, 6
- SECRET_KEY, 5
- SENTRY_DSN, 7
- SENTRY_SEND_DEFAULT_PII, 7
- SENTRY_TRACE_SAMPLE_RATE, 7
- SSL_ENABLED, 5