

---

# **TerraVisu**

***Release 2023.4.5***

**Autonomens**

**Apr 20, 2023**



## CONTENTS:

<b>1</b>	<b>Install instructions</b>	<b>3</b>
1.1	Requirements . . . . .	3
1.2	Install . . . . .	3
1.3	Update . . . . .	4
<b>2</b>	<b>Configuration</b>	<b>5</b>
2.1	Environment variables . . . . .	5
<b>3</b>	<b>Usage</b>	<b>9</b>
3.1	Configure instance settings . . . . .	9
3.2	Sources, layers and views . . . . .	9
<b>4</b>	<b>Development</b>	<b>11</b>
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
<b>5</b>	<b>Troubleshooting</b>	<b>13</b>
5.1	Elastic search container doesn't start . . . . .	13
<b>6</b>	<b>CHANGELOG</b>	<b>15</b>
6.1	2023.4.5 (2023-04-20) . . . . .	15
6.2	2023.4.4 (2023-04-19) . . . . .	15
6.3	2023.4.3 (2023-04-13) . . . . .	15
6.4	2023.4.2 (2023-04-11) . . . . .	15
6.5	2023.4.1 (2023-04-07) . . . . .	16
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Index</b>	<b>19</b>





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 2023.4.5 (2023-04-20)

#### Improvements:

- Disable autocomplete / autofill on PostGIS source form in admin

#### Bugfix

- Allow PostGIS source form edition in admin without retype password

### 6.2 2023.4.4 (2023-04-19)

#### Improvements:

- Improve admin to define polygons patterns
- Filter frontend with non empty views
- Default view is now the first ordered for an user (authenticated or not)

### 6.3 2023.4.3 (2023-04-13)

#### Improvements:

- Allow to set group access to extra menu items
- Include basic certificates in docker image

### 6.4 2023.4.2 (2023-04-11)

#### New features:

- Allow using style images patterns in polygon advanced styles

## 6.5 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