

Inventory and Gap Analysis (v 4.0)

Installation Guide

29 November 2023

Supported by wVSSM Team

By: Hamid R. Shahriari (hshahriari@gmail.com)

1 REQUIREMENTS and PREPERATION

To install IGA v4, a server (virtual or physical) is needed:

Hardware specification:

- CPU: at least 2-core
- RAM: 4 GB or higher
- Storage: at least 20 GB (preferable SSD)

Software specification:

- OS: Linux Ubuntu Server 20.04 or 22.04 LTS
- DBMS: PostgreSQL 12.3 or higher
- Other needed software packages are listed in the next section.

Other requirements:

- A domain name specific for the application frontend and its assigned valid IP address
- A domain name specific for the application backend and its assigned IP address (may be same as frontend IP address)
- SSH access to the server

You have to define two domain names: the first one for the web application and the second one for its backend API. For example: mycountry.invgap.org and myapi.invgap.org. Both may use a single valid IP address.

⚠ NOTE: The current version of IGA (4.x) is dependent to **Python 3.8** (due to a used python libraries). You can not use Python 3.10 !

2 DNS Setup

You have to setup two domains, first for main URL, and second for backend URL.

For example: invgap.org and myapi.invgap.org. Of course, they may share a single IP. Please send the domain names to support team, after that we could create and build project file.

3 DEPENDENCIES

It is needed to have the following software packages:

1. Ubuntu 20.04 LTS (or higher)
2. Python 3.8 and python3-venv
3. Django 3.0.3
4. Django Rest Framework 3.11.0
5. Node 14.4.0 or higher
6. npm 6.14.5 or higher
7. PostgreSQL 12.3 or higher
8. Nginx web server

3.1 Install dependencies

3.1.1 Ubuntu 20.04 LTS

```
sudo apt-get update
sudo apt-get install python3.8 python3-pip python3.6-dev postgresql
postgresql-contrib libpq-dev
```

3.1.2 Django

```
pip3 install django
```

3.1.3 Node

```
sudo apt install node npm
```

3.1.4 PostgreSQL

```
sudo apt install postgresql postgresql-contrib
```

4 INSTALLING BACKEND

4.1 Dependencies

To install dependencies please read [first section](#) of this guide.

4.2 Installation

1. Select a folder for IGA installation. For example: /var/www/iga
2. Unzip the project file which has been built for you (Please refer to [DNS Setup](#) section).
3. Install the requirements:
 - a. `cd /var/www/iga/iga-backend`
 - b. `python3.8 -m venv venv`
 - c. `source venv/bin/activate`
 - d. `pip3 install -r requirements.txt`
4. To use PostgreSQL as DBMS, you must create PostgreSQL database.

4.3 PostgreSQL installation

1. Login as postgres user: `su - postgres`
2. Run PostgreSQL database: `psql`
3. Create database: `create database iga_db;`
4. Set password for PostgreSQL user: `\password DBPassword;`
5. Exit the PostgreSQL console: `\q`
6. Exit from postgres user: `exit`

4.4 Edit settings file

1. `cd iga-backend`
2. Using an editor, open the file `.env` (please note to the dot in the initial of filename!)
3. Edit the following lines, to indicate database and its password:

```
db_name=iga_db

db_user=postgres

db_password=DBPassword

...
```
3. Edit `inventory/settings.py` and add your frontend and backend address to `CSRF_TRUSTED_ORIGINS`.

```
CSRF_TRUSTED_ORIGINS = [
    'https://mycountry.invgap.org'
    'https://myapi.invgap.org'
]
```
4. For security reasons, it is recommended to change `SECRET_KEY` value.

4.5 Load data

If you have a database dump file (such as backup file), use the following command and skip section Load initial data.

```
pg_restore -U postgres -d iga_db --single-transaction --clean backupfile.bak
```

(assuming iga_db is the database name, and backupfile.bak is the file name of the backup file.)

4.5.1 Load initial data

First you have to setup virtual environment (if not active):

```
source venv/bin/activate
```

To load database tables and please use these commands:

```
python3 manage.py makemigrations
```

```
python3 manage.py migrate
```

then you have to load initial data:

```
python3 manage.py loaddata initial.json
```

Now you have user admin with password admin by default.

4.6 Run the backend service

To run the project:

```
screen -S IGA python3 manage.py runserver 0.0.0.0:8000
```

Of course, you can use `tmux` tool, or run using `nohup` command to run in the background.

You can check `localhost:8000` or your server IP for successful setup.

5 INSTALLING FRONT-END

5.1 Dependencies

To install dependencies, please read [related section](#) of this guide.

5.2 Installation

1. If you have extracted file of IGA, you can skip step 2.
2. Unzip the project file which has been built for you.
3. Change the directory to frontend directory e.g. run `cd iga/iga-frontend`
4. Install the requirements of project:

```
npm i --force
```

5. You can run the project by running the command:

```
serve -s build
```

If you do not have `serve` package installed, you can install it by running the command:

```
npm install -g serve
```

5.3 Running in background

It's recommended to run in background by `pm2` tool. You can install `pm2` by running the command:

```
npm install pm2 -g
```

then run the command (we assume that prefer to use port 3000):

```
pm2 serve build 3000 --name iga --spa
```

6 CONFIGURATION of NGINX

It's needed to configure Nginx web server. Follow these steps:

1. Copy the default config file from backend folder, with a new name such as `myinvgap`

```
cp IGA3-backend/nginx.conf /etc/nginx/sites-enabled/iga
```
2. Edit the new file `iga` and replace the domain names with your new domains (main domain, and API domain).
3. Edit the new file `iga` and replace the media folder according to the the folder you have installed the backend.
4. Please note that if the number of facilities or items is too large, you should increase some parameters in the Nginx server. So, please edit your nginx default config file (usually `/etc/nginx/nginx.conf`) and set these parameters:

```
http {  
    proxy_connect_timeout      300;  
    proxy_send_timeout        300;  
    proxy_read_timeout        300;  
    send_timeout               300;  
  
    client_max_body_size      20M;  
  
    ...  
}
```

5. Create site TLS certificates and config it. You can do it manually or run the following command, and follow the steps:

```
certbot -nginx
```

Note: Please note that certbot needs that the port 80 to be open. So, if your firewall bans the traffic to port 80, reconfigure it.

6. Configure media folder:

The backend needs a folder to save some static files such as logo, exported excel files and so on. It needs write permission to be accessed by nginx to serve static files and writeable by backend process.

Supposing that the backend have been installed in `/var/www/iga/iga-backend`, you may add something like this in your backend service configuration of Nginx:

```
server {  
    server_name api.invgap.org;  
    location /media {  
        alias /var/www/iga/iga-backend/media;  
    }  
    ...  
}
```

Then you have to give permission to its folder:

```
cd /var/www/iga/iga-backend  
chmod g+w,o+w media  
chmod g+w,o+w media/help
```

7. Check the Nginx configuration and if it is ok, restart the Nginx web server:

```
nginx -t  
systemctl restart nginx
```