Ginan Workshop - Windows Set-up - 2025-06-23

Prerequisites:

- Installed Docker Desktop on computer
  - https://docs.docker.com/desktop/setup/install/windows-install/#install-docker-desktop-on-windows
- At least 10GiB of free space

Docker Desktop - Set-up network and Mongo database container

First, open up the Docker Desktop App and open a PowerShell terminal - the button is found on the bottom right of the window.
Next create a docker network for our containers to live in:

```
$ docker network create ginan-net
```

Next, spin up a container with a running Mongo database, ensuring it sits on the network we just created above:

```
$ docker run -d --network ginan-net  --name ginan-mongodb mongo:latest --noauth
```

The above container (and the Mongo database within) is where we will send results from our Ginan processing runs.

Find IP address of Mongo container

To ensure results end up in Mongo, we will need to find the IP address of the container we just created.
This can be done by reading off the address using the `docker inspect` command.
In your Docker Desktop terminal, pass the following command:

```
$ docker inspect ginan-mongodb
```

This will print out `json` formatted text. Under the field "IPAddress" (near the end of the print-out) you should have something that looks like this:

```
"IPAddress": "172.18.0.2"
```

Take note of this IP address, as this will need to be later input this manually into the YAML file, under `"mongo:primary_uri"`

Now that the MongoDB is running in the background, we can move to creating the Ginan container and a place for our files to live locally.

Create Local Directory for input / output files

First, you can create a local directory on your Windows machine for your input and output data to live.
In your favourite File System explorer, create a directory.
Mine is at: `C:\Users\ronal\geo-wkshop`
We will bind mount this location onto the Docker container used for Ginan, so that we can access it via the standard File Explorer on Windows.

Docker Desktop - Set-up Ginan container

Back in the Docker Desktop Powershell terminal, run the following docker command:

```
$ docker run -it --network ginan-net -p 5023:5023 -v C:\Users\ronal\geo-wkshop:/data gnssanalysis/ginan:1e62225 bash
```

This will download the image, start the container, bind your local directory to /data, and run an interactive bash terminal for you to run Ginan in.
We have also opened up port 5023 - this will be important to visualise the data locally later.

The bash terminal that has come up should have a somewhat funny prefix, looking something like:

```
root@e94a4dca32f9:/tmp#
```

In this terminal, we will install some necessary packages:

```
$ apt update
$ apt install wget
$ apt install git
$ apt install python3.12-venv
```

Now change directory to the one that has been locally mounted:

```
$ cd /data
```

Next, clone the Ginan repository from GitHub (taking only the relevant `develop` branch, which is most up to date):

```
$ git clone -b develop-weekly --single-branch https://github.com/GeoscienceAustralia/ginan.git
```

And lastly, download the necessary input data and products:

```
$ cd ginan/inputData/data
$ ./getData.sh
$ cd ../products
$ ./getProducts.sh
```

This final step may take some time, as quite a bit of data needs to be downloaded.

Set Up YAML (config) file - Point at our Mongo database

Now, before we run our first example, `ppp_example.yaml`, we need to edit the file to tell Ginan where to find the MongoDB.
Because all these files live on your local Windows File System, you can open up your favourite text editor and open the YAML in Windows itself (not in the terminal).
For me, this YAML lives here: `C:\Users\ronal\geo-wkshop\ginan\exampleConfigs\ppp_example.yaml`

After opening, scroll down to the field `"primary_uri"` under `"mongo:"` (which should be Line 292).
Replace `"localhost"` with the IP address you found earlier for your container with the Mongo database.

For me, the original field: `primary_uri: mongodb://localhost:27017` Would become --> `primary_uri: mongodb://172.18.0.2:27017`

Remember to save!

Docker Desktop - Run First example

Next, we will return to our bash terminal in Docker Desktop, and run our first example.
Change to the correct directory:

```
$ cd /data/ginan/exampleConfigs/
```

And then call the pea (Parameter Estimation Algorithm) to run this example:

```
$ pea --config ppp_example.yaml
```

This will print out all messages to screen, so that you cna keep track of where things are up to.

Once this is complete (it took about 20 - 25 min testing on my machine), you can investigate the results Ginan has an excellent visualisation tool called the GinanEDA (EDA = exploratory data analysis).

Docker Desktop - Visualise Data from First example - GinanEDA

Create Python Vritual Environment:

To use the GinanEDA, we will need to create a python virtual environment to install the necessary packages.
In your Ginan container bash terminal (the one with the funny prefix - mine was `root@e94a4dca32f9:`), run the following:

```
$ cd /data
$ python3 -m venv ginan-env
```

This will create a directory called `"ginan-env"` which contains the virtual environment.
To activate the environment, we use the following command:

```
$ source /data/ginan-env/bin/activate
```

You should now see a label at the start of your next line in your terminal, specifically `(ginan-env)`
For me this looks like this: `(ginan-env) root@e94a4dca32f9:`. This tells you that you are in a virtual environment

Install Python Packages in Virtual Environment:

Next, install the necessary packages using the requirements file:

```
$ cd /data/ginan/scripts/GinanEDA/
$ pip install -r requirements.txt
```

This may take some time, do not exit the terminal early.

Edit the index.py File to Display the GinanEDA in Your Browser

Once this is complete, we will edit the `index.py` file located in the `ginan/scripts/GinanEDA` directory.
Again, this will live in your local Windows File system, so you can open this file in your favourite editor, and make the following change:

`app.run()` --> `app.run(host='0.0.0.0', port=5023)`

Make sure to leave the line with the same indentation(4 spaces from the start of the line).

Run the GinanEDA

Now you can run the GinanEDA:

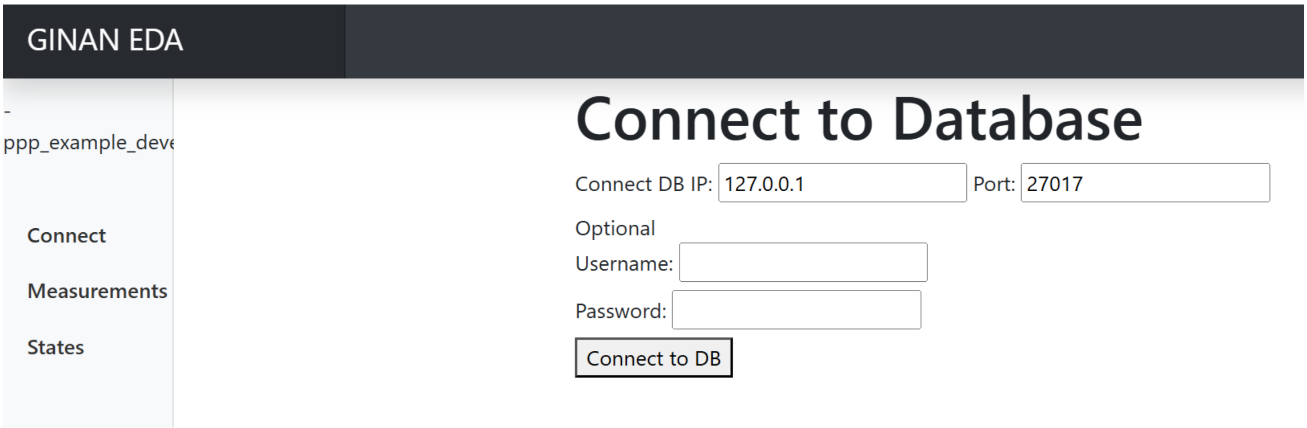
```
$ python3 index.py
```

This will print out something like the following to screen:

```
 * Serving Flask app 'index'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5023
 * Running on http://156.14.0.3:5023
Press CTRL+C to quit
```

Open the GinanEDA and Results from the First Example

You can click on the first `localhost` address (`http://127.0.0.1:5023`), which should open up in your default browser.
If that doesn't work, you can open up the browser yourself and copy past the address yourself. You should see something like this:



Connect to Database

Connect DB IP: 127.0.0.1 Port: 27017

Optional

Username:

Password:

The defaults on the GinanEDA will not work for our set-up with Docker.
Use the following steps:

1. Change the `Connect DB IP` box from `127.0.0.1` --> The IP you obtained for your Mongo container (in my case `172.18.0.2`)
2. Click `Connect to DB`
3. A dropdown should appear below where you can choose the relevant database - choose `ppp_example_develop`
4. Click `Load DB` This will load the relevant database from our first example

You will now be able to follow along with the rest of Workshop, as we go through how to use the GinanEDA and run other examples.