**Exporting OpenStreetMap to UBUNTU tileserver-gl-light**

Tileserver-gl-light uses of both raster and vector files to display maps. Raster files are images while vector files are generated on the fly with styles to modify colour, etc. Due to generation on the fly, vector files are slower than raster files in rendering maps.

Raster files via Open Street Map (OSM) is typically served from [*https://tile.openstreetmap.org/{z}/{x}/{y}.png*](https://tile.openstreetmap.org/%7bz%7d/%7bx%7d/%7by%7d.png)

The implication is that the server serving this link must be connected to the internet. To wearn off this internet dependency, the best way is for the tileserver to serve tiles from a local MBtile file. There are no easily available and updated downloadable Singapore MBTile file. Even the SG Gov OneMap is not updated. As you can see outlines of the new SIT campus in my generated OSM file on the right vs what is from SGonemap.

|  |  |
| --- | --- |
| GOV One Map | Generated OSM MBTiles |
|  |  |

Geographic Information Systems (GIS) is a highly specialised skillset. What this means is that the information is specialised that there is hardly any books or resources that are skilled in this area. But then again, not every app needs a map.

The context of this exploration is to use NVIDIA Xavier NX (<https://www.nvidia.com/en-sg/autonomous-machines/embedded-systems/jetson-xavier-nx/>) to display local maps. The Xavier NX is an edge computer running off UUNTU on ARM64 CPU. ARM64 is not a common generated target for libraries. This implies building libraries from scratch if a version is not available, a highly tedious trial and error process.

**The thought process is as follows.**

Make use of Windows 10 tools to generate OSM mbtiles file. Use windows to examine and confirm the mbtile file is ok before moving it to UBUNTU.

A balance is made between having low level details ( ie. Big ZOOM level ) vs higher level ( less details but smaller file) There is no need to have zoom level starting from 2. 12 is good enough as a start. Also, I chose 19 as the max level. What these numbers means is that the higher the zoom level, the more tiles will be downloaded into the MBTILES files making it overly big. Too small is also useless. I even though I had the EPSG coordinates system mixed up, so I couldn’t see the map in Ubuntu. So, using the DEVELOPER tools on Chrome Browser for debugging will aid you to see what is done properly. Due to the size of the MBfile, it takes awhile to load. There are other integration tools like LOADING spinner that can be added.

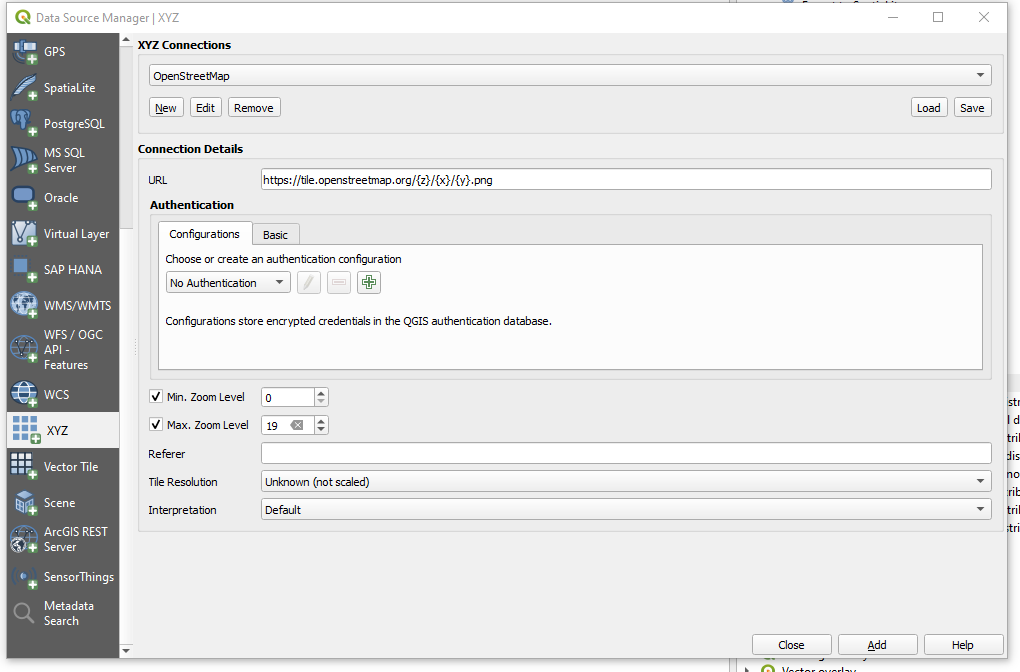
The tools I used are:

|  |  |
| --- | --- |
| MapTiler Engine 13.2 | To just view the MBtiles file |
| DB Browser for SQLite | MBtiles file is basically an SQL database. Using SQLite 3, you can examine the contents of the MBtiles file. I got a corrupted file on UBUNTU and discovered it too late after hours of generation. Hence, this confirmation step helps reduce wasted hours. |
| QGIS 3.26.2 desktop | This is the key tool. Overall steps is to create a new project, add a data source manager layer with inbuilt OSM XYZ connection |

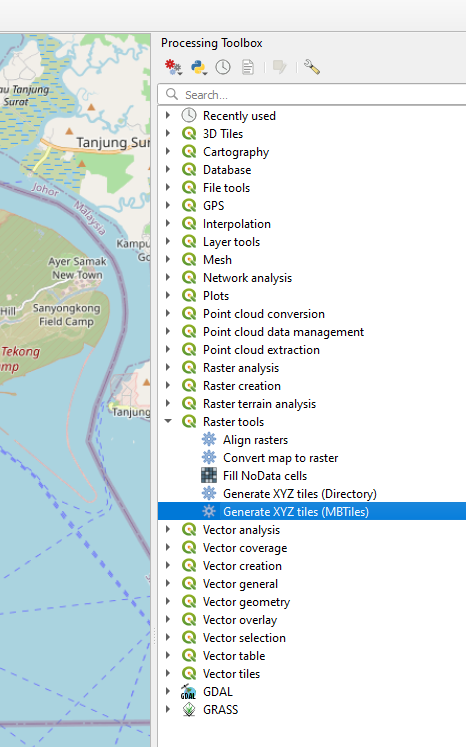
1. **Create a new project with OSM layer**

**Layer - Data Source Manager – XYZ**

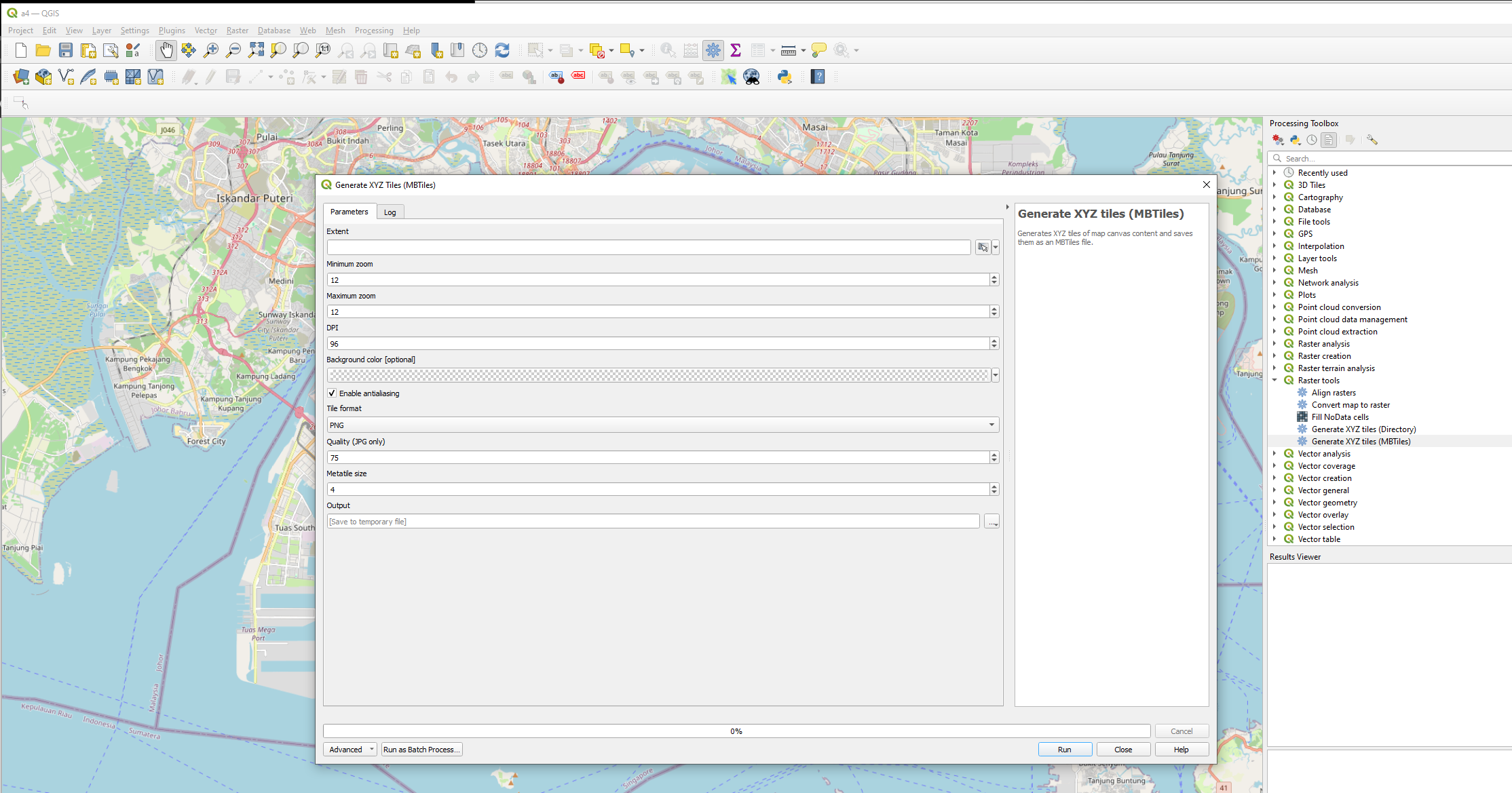
1. Activate QGIS Desktop and create a new project.
2. Create a new layer with Open street Map via Data Source Manager



1. Generate Raster mbtiles file



It is important to check the right extent. You can “draw” on the map to get the right extents. Use a fixed file instead of temp file. Set max zoom to 19. Leave the rest of the parameters as it is.



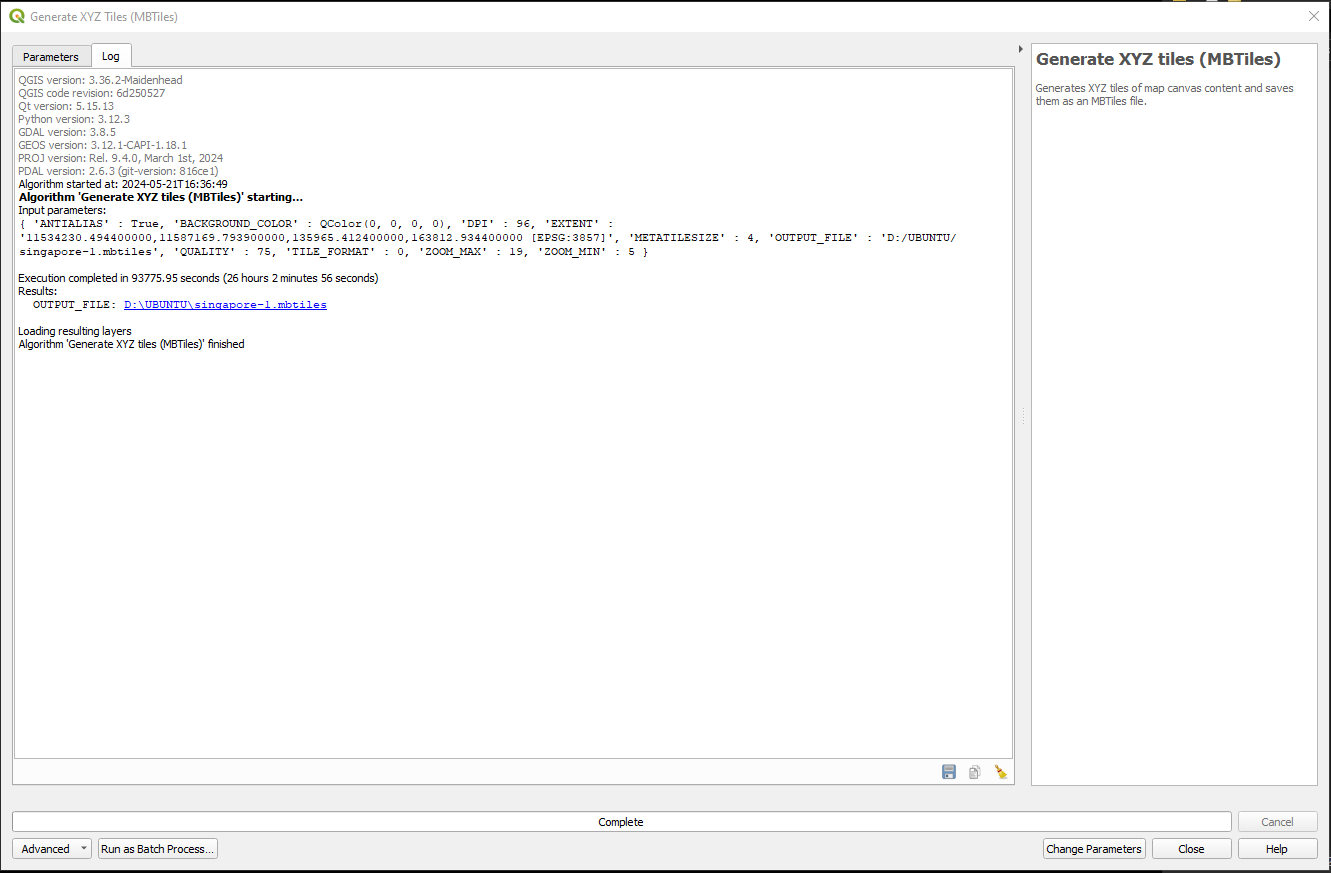
Set Max zoom to 19

Set the name of the specific file to generate

Set the extents to the area of the map

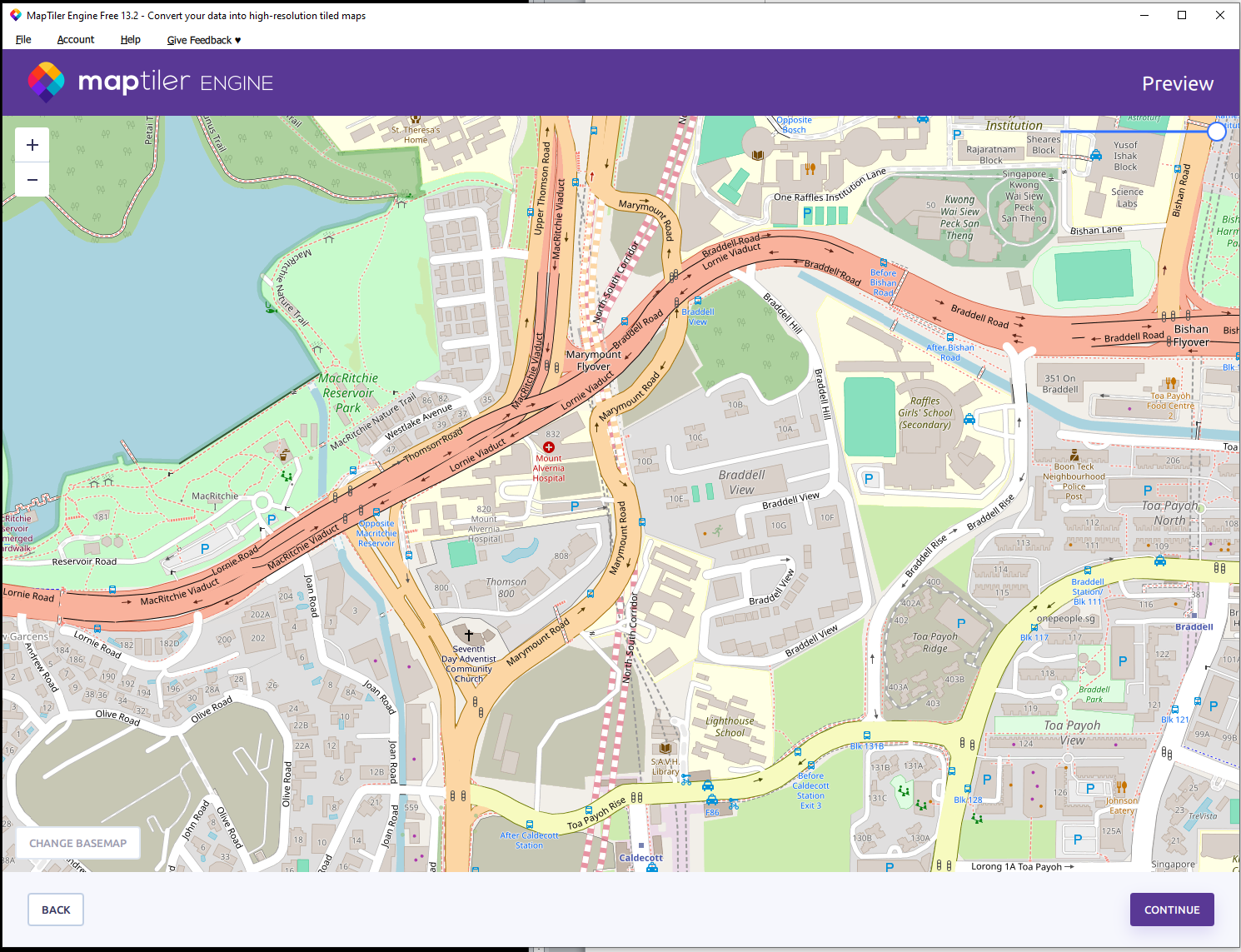
**processing.run("native:tilesxyzmbtiles", {'EXTENT':'11520707.115800001,11588546.141500000,122150.430800000,166024.781900000 [EPSG:3857]','ZOOM\_MIN':12,'ZOOM\_MAX':16,'DPI':96,'BACKGROUND\_COLOR':QColor(0, 0, 0, 0),'ANTIALIAS':True,'TILE\_FORMAT':0,'QUALITY':75,'METATILESIZE':4,'OUTPUT\_FILE':'D:/UBUNTU/a3.mbtiles'})**

It takes quite a long ( more than 5 hours ) to generatea 19 level zoom. Shorter processing time and smaller MBtile file if you reduce the ZOOM max level. One way to tell if the file is still being generated is to look at the folder



1. **Check the generated MBfile is viewable in Windows**

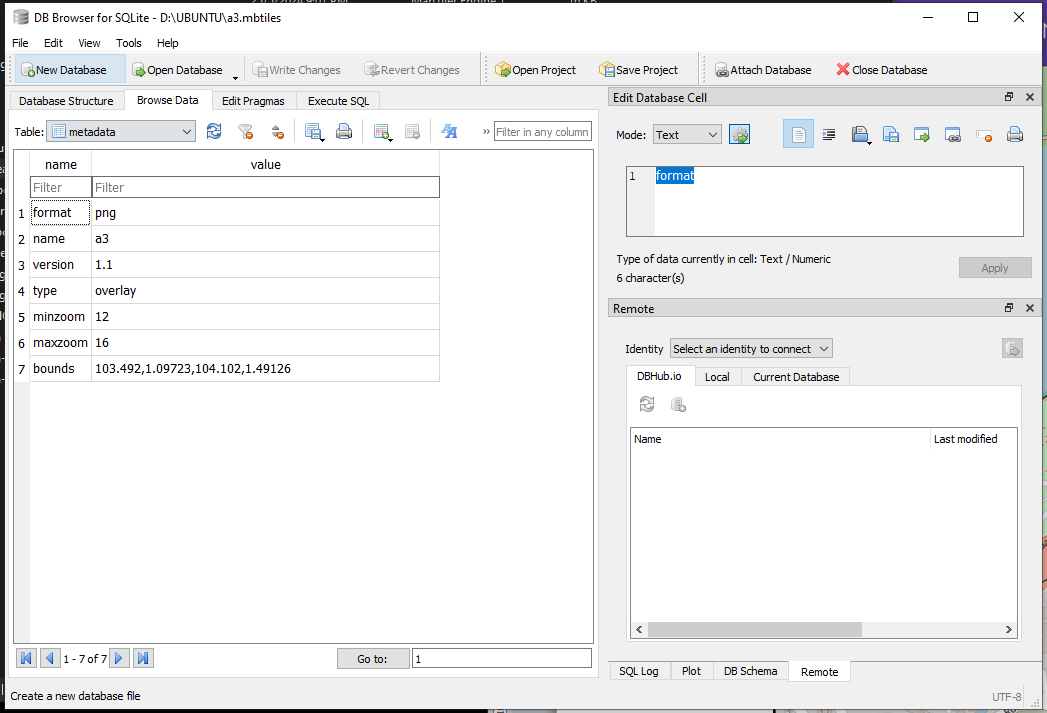
Double click on the file to render ( not serve ) in maptiler engine ( in windows )



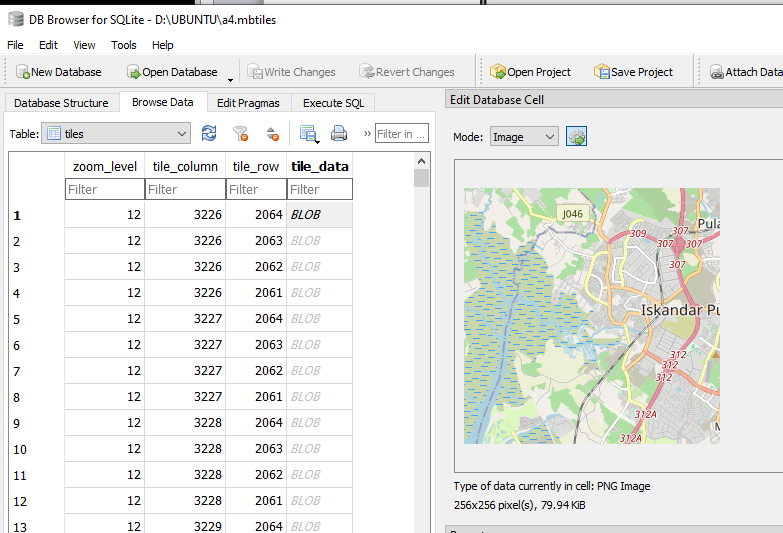
Use **DB Browser** on the generated MBTILES DB file to see zoom levels and bounds. Check that the tiles tables can have the generated images BLOB.

MBfiles is basically an SQL file. There are only two tables – **Metadata** and **Tiles**.

Bounds denote boundary of the map in lat , lon of the top left corner to the bottom right corner.



Looking at the **tiles** table, you can see the generated zoom level followed by column and row plus a BLOB (Binary Large Object) containing the generated image. The bigger the zoom level, the more rows and columns and hence more BLOBs are generated and kept in the MBtiles file. So tile\_column 3266, row 2064 plus 2063 combined with column 3267 row 2064 and 2063 will form 4 squares at zoom 12.



|  |  |  |
| --- | --- | --- |
|  | 3226 | 3227 |
| 2063 |  |  |
| 2064 |  |  |

1. **Sending mbfile to UBUNTU**

I used OUTLOOK mail file attachment with OneDrive to save the MBtiles file. Make sure, the file is properly loaded and downloaded from UBUNTU. There was a case where download was an issue resulting in a corrupted file.

1. Copy xxx.mbtiles file over to ~/.npm/etc ..
2. Modify tkh1.html to adjust extends plus ZOOM levels
3. View in **~/tileserver/tileserver-gl-light**