

Web Mapping with Vector Tiles Using Geospatial Open Source Tools

Peter Keum - Paul McCombs

May 21, 2018

WA GIS 2018 Pre-Conference Workshop

Introduction

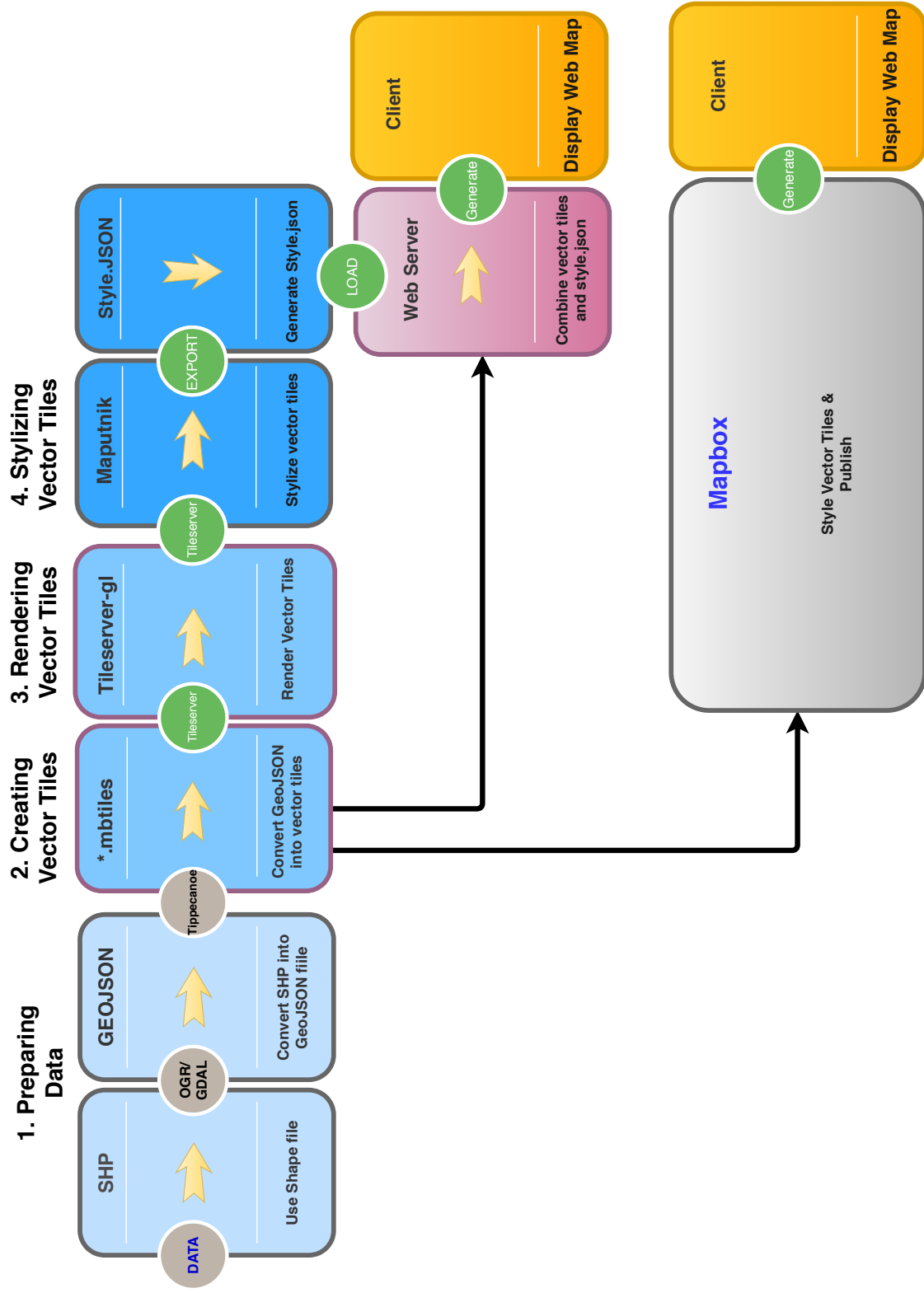
Hello there. This page contains materials for Washington URISA 2018 Workshop. This workshop describes and demonstrates various FOSS4G (Free Open Source Software for Geospatial) tools to create [vector tiles](#), to stylize vector tiles and to build web-maps.

We'll explore FOSS4G workflow to create web-maps from vector tiles. We'll have presentations with demonstrations which we will follow with hands-on exercises. We'll create vector tiles then style them to create web-maps. In addition, we'll be exploring [Mapbox Studio](#) workflow as well.

Who are we? We are two FOSS4G enthusiasts GIS professionals working for King County, Washington.

[Peter Keum](#), GISP. GIS Specialist with King County Wastewater Treatment Division/KC GIS Center.
[LinkedIn](#)

[Paul McCombs](#) GIS Specialist with King County GIS Center.



Pre-Installation for Vector Tile Workshop

Please bring your laptop with the following items pre-installed. Instructions are found below. Please feel free to contact us for any questions or issues with following the installation instructions.

- Create a [GitHub](#) account.
 - If you have an existing GitHub account please be sure you can successfully log in.
 - Please follow the [GitHub how-to](#). We will use the GitHub accounts to share the web maps we will build during the workshop.
- Create a Tilehosting account.
 - Please go to [Tilehosting.com](#) to create an account. Tilehosting will provide vector basemaps for our web maps.
- In order to install Docker and other software during the workshop, the user needs to have **Administrative level privileges**. You will not be able to complete the exercises without administrative privileges.
- Download and install Node for your operating system.
 - Follow the [Node Download Instructions](#).
- After Installing Node, please install http-server, which is very simple http server that we'll run locally for testing and demonstration purposes.
 - Follow [Installation Instructions](#)
- Install Docker
 - Please download Docker Community Edition for your Operating System.
 - Windows 10 Home Edition is not compatible with Docker Community Edition, you will need to install [Docker Toolbox](#)
 - If you have difficulty installing Docker, please contact us.
- Install Docker on MacOS
 1. [Docker Community Edition](#)
 2. Mount Docker.dmg
 3. Drag Docker App onto Applications Shortcut
 4. Launch Docker
 5. Agree to run application from Internet
 6. Supply admin password
- Windows users may need to [enable virtualization for your laptop in the system configuration](#).
- Install Docker on Windows (Pro and Enterprise)
 1. [Docker Community Edition](#)
 2. Requires Windows 10 Pro or Enterprise
 3. **Does Not Work with Windows 10 Home Edition**
- Install Docker on Windows 10 Home:
 1. [Docker Toolbox](#)
 2. Accept Defaults during install

3. Run the Quick Start Terminal
4. Execute the steps at the Quick Start Terminal

Generating Vector Tiles using the Docker Containers for Windows 10 (Pro and Enterprise)

In this lesson we will learn to use Docker containers to deploy various tools to create and serve vector tiles on your laptop running Windows Pro or Enterprise Edition.

If you are running Windows 10 Home please use the [Windows 10 Home instructions](#)

1. Enable Hyper-V in Windows

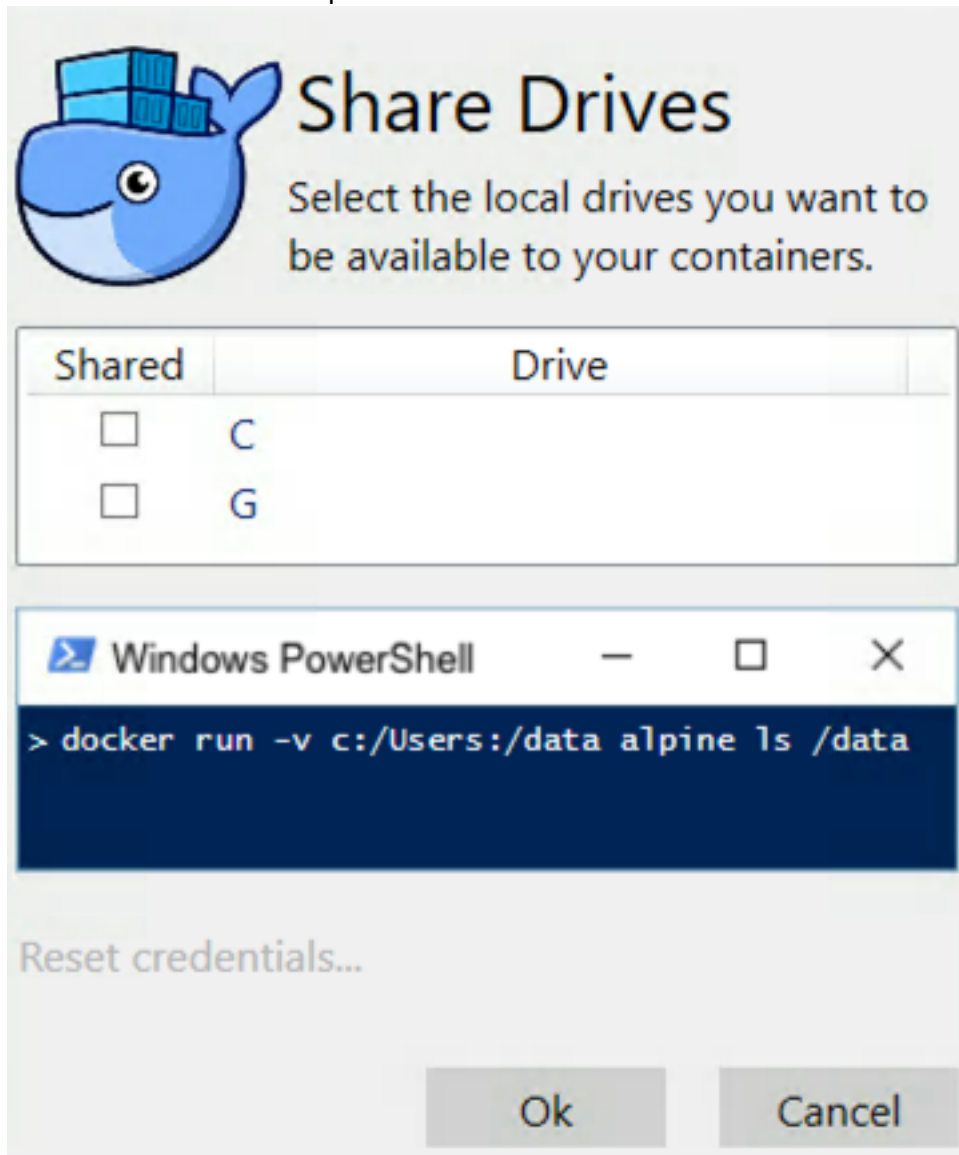
- **Requires Windows 10 Pro or Enterprise**
- [Hyper-V](#) provides hardware virtualization support in Windows
- Open a PowerShell terminal window as Administrator
- To enable Hyper-V temporarily, enter the following command at the Power Shell prompt

`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All`
- **TIP:** [Step-by-step guide](#) to enable Hyper-V permanently on your computer. This includes a discussion of how to enable Virtualization in BIOS.

2. Install Docker on Windows 10

- **Requires Windows 10 Pro or Enterprise**
- Download [Docker Community Edition for Windows](#)
- Double Click the Docker for Windows Installer.exe file
 - Follow the instructions and accept default settings
 - You may be asked to provide your password during installation
 - You may be asked to restart your computer to complete the installation. If your computer restarts, you will need to reenble Hyper-V as explained in step 1 above.
- Start Docker for Windows from the Start Menu

- Turn on Drive Sharing
 - Open the settings for Docker
 - Select the *Share Drives* option



[image source](#)

- Check the box next to "C"
 - Click *OK* and you will be prompted for credentials with admin privileges
- If Docker was installed with an admin account different from your user account, you will need to check that your user is in the local docker-users group for your machine

3. Install a GDAL/OGR tools container

- Start a PowerShell Terminal with your normal user privileges. In other words do not start as admin as you did before.
- Go to [Docker Hub](#)
- Search for gdal
- Select [klokantech/gdal](#)

- Copy the Docker Pull Command & run it at PowerShell prompt

```
docker pull klokantech/gdal
```

4. Convert Shape file into GeoJSON file

- Download zipped files of [King County 2000 Census Block Groups](#)
- Place zip file into your local user directory (eg: C:)
 - "keump" will be replaced with your local user name
- Unzip zip file
- Use OGR tools at the PowerShell prompt

- *ogrinfo*: check the shape file's information

```
docker run -it --rm -v C:\Users\keump:/data klokantech/gdal ogrinfo
KingCo_2000_Census_BlockGroups.shp -al -so
```

- *ogr2ogr*: convert shape file to GeoJSON file

```
docker run -it --rm -v C:\Users\keump:/data klokantech/gdal ogr2ogr -t_srs
EPSG:4326 -f GeoJSON KingCo_2000_Census_BlockGroups.geojson
KingCo_2000_Census_BlockGroups.shp -Progress
```

5. Locate a GeoJSON file

- You can download the following GeoJSON file for next step if needed
- [King County 2000 Census Block Groups GeoJSON](#)

6. Install a Tippecanoe container which is a utility tool to create vector tiles

- Search for Tippecanoe on [Docker Hub](#)
- Select the [jskeates/tippecanoe repository](#)
- Copy the appropriate command from the *Docker Pull Command* section of the page
- Paste it at the PowerShell prompt, and hit enter to run it

```
docker pull jskeates/tippecanoe
```

7. Create some vector tiles

- Ensure Docker is running on your computer
- Start Tippecanoe container in interactive mode at the PowerShell prompt

```
docker run -it -v c:\users\keump:/home/tippecanoe jskeates/tippecanoe:latest
```

- You will see your command prompt change to look like bash-4.3\$
- Use the tippecanoe command at the PowerShell prompt to create vector tiles from the geoJSON file

```
tippecanoe -o KingCo_2000_Census_BlockGroups.mbtiles
KingCo_2000_Census_BlockGroups.geojson
```

- Exit the container when it is done

```
exit
```

- The vector tiles will be \$HOME/KingCo_2000_Census_BlockGroups.mbtiles
 - \$HOME represents your user directory at a unix style command prompt.

8. Install a TileServer GL container

- Go to [Docker Hub](#)
- Search for tileserver-gl
- Select [klokantech/tileserver-gl](#)
- Copy the Docker pull command & run it at the PowerShell prompt

```
docker pull klokantech/tileserver-gl
```

9. Run TileServer GL

- Ensure Docker is running on your computer
- From the command line change into the directory where you have placed your mbtiles file.
- Start the TileServer GL container from the PowerShell prompt

```
docker run --rm -it -v C:\users\keump:/data -p 8080:80 klokantech/tileserver-gl
```

- Windows may prompt you to create a firewall exception, depending on your security settings. If asked, agree to the exception
- Test that the vector tiles are being served by entering <http://localhost:8080/> into your browser's address bar
- After testing, hit **ctl-C** to quit TileServer GL

Generating Vector Tiles using the Docker Containers for macOS

In this lesson we will learn to use Docker containers to create and serve vector tiles on your Macintosh laptop.

1. Install Docker on macOS

- Download [Docker Community Edition for Mac](#)
- Mount Docker.dmg
- Drag Docker App onto Applications Shortcut
- Launch Docker (double-click Docker Icon)
- Agree to run application from Internet
- Supply admin password when prompted

2. Install a GDAL/OGR tools container

- Go to [Docker Hub](#)
- Search for gdal
- Select [klokantech/gdal](#)
- Copy the Docker Pull Command & run it at the Terminal prompt

```
docker pull klokantech/gdal
```

3. Convert shape file into GeoJSON file

- Download zipped files of [King County Census Blocks](#)
- Place zip file into your local directory
- Unzip zip file
- Use OGR tools at the Terminal prompt

- *ogrinfo*: check the shape file's information

```
docker run -ti --rm -v $(pwd):/data klokantech/gdal ogrinfo  
KingCo_2000_Census_BlockGroups.shp -al -so
```

- *ogr2ogr*: convert shape file to GeoJSON file

```
docker run -ti --rm -v $(pwd):/data klokantech/gdal ogr2ogr -t_srs  
EPSG:4326 -f GeoJSON KingCo_2000_Census_BlockGroups.geojson  
KingCo_2000_Census_BlockGroups.shp -Progress
```

4. Locate a geoJSON file

- You can download the following GeoJSON file for next step if needed
 - [King County 2000 Census Block Groups GeoJSON](#)

5. Install a Tippecanoe container which is a utility tool to create vector tiles

- Search for Tippecanoe on [Docker Hub](#)
- Select the [jskeates/tippecanoe repository](#)
- Copy the appropriate command from the *Docker Pull Command* section of the page
- Paste it at the Terminal prompt, and hit enter to run it


```
docker pull jskeates/tippecanoe
```

6. Create some vector tiles

- Ensure Docker is running on your computer
- Start Tippecanoe container in interactive mode at the Terminal prompt

```
docker run -it -v $(pwd):/home/tippecanoe jskeates/tippecanoe:latest
```

- You will see your command prompt change to look like bash-4.3\$
- Use the tippecanoe command at the Terminal prompt to create vector tiles from the geoJSON file

```
tippecanoe -o KingCo_2000_Census_BlockGroups.mbtiles  
KingCo_2000_Census_BlockGroups.geojson
```

- Exit the container when it is done

```
exit
```

- The vector tiles will now be in your directory

7. Install a TileServer GL container

- Go to [Docker Hub](#)
- Search for tileserver-gl
- Select [klokantech/tileserver-gl](#)
- Copy the Docker pull command & run it at the Terminal prompt `docker pull klokantech/tileserver-gl`

8. Run TileServer GL

- Ensure Docker is running on your computer
- From the command line change into the directory where you have placed your mbtiles file
- Start the TileServer GL container from the Terminal prompt

```
docker run --rm -it -v $(pwd):/data -p 8080:80 klokantech/tileserver-gl
```

- Test that the vector tiles are being served by entering <http://localhost:8080/> into your browser's address bar
- After testing, hit **ctl-C** to quit TileServer GL

Generating Vector Tiles using the Docker Containers for Windows 10 (Home)

In this lesson we will learn to use Docker containers to deploy various tools to create and serve vector tiles on your laptop running Windows Home Edition.

If you are running Windows 10 Pro or Window 10 Enterprise please use the [Windows 10 Pro instructions](#)

1. Install Docker Toolbox

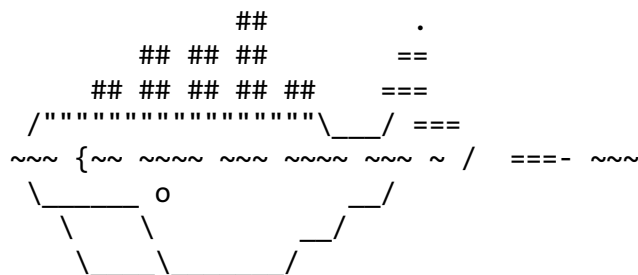
- Download [Docker Toolbox for Windows](#)
- Run *DockerToolbox.exe*
 - Accept Defaults during install

2. Run the Quick Start Terminal

- Double-click the Quick Start Terminal installed by Docker Toolbox
 - You will execute all the docker commands in the steps below at the *Docker Quickstart Terminal* prompt
 - For help getting started, check out the docs at <https://docs.docker.com>

Docker is up and running!

To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: `C:\Program Files\Docker Toolbox\docker-machine.exe env default`



- Docker Toolbox is configured to present it's default machine as IP 192.168.99.100 to the browser running natively on your computer

3. Install a GDAL/OGR tools container

- Go to [Docker Hub](#)
- Search for gdal
- Select [klokantech/gdal](#)
- Copy the Docker Pull Command & run it at Quick Start Terminal prompt

```
docker pull klokantech/gdal
```

4. Convert Shape file into GeoJSON file

- Download zipped files of [King County 2000 Census Block Groups](#)
- Place zip file into your local user directory (eg: C:)
 - "mccombsp" will be replaced with your local user name
- Unzip zip file. Be sure that the files are in your local user directory, not in a subdirectory.
- Use OGR tools at the Quick Start Terminal prompt

- *ogrinfo*: check the shape file's information

```
docker run -it --rm -v $HOME:/data klokantech/gdal ogrinfo
KingCo_2000_Census_BlockGroups.shp -al -so
```

- *ogr2ogr*: convert shape file to GeoJSON file

```
docker run -it --rm -v $HOME:/data klokantech/gdal ogr2ogr -t_srs
EPSG:4326 -f GeoJSON KingCo_2000_Census_BlockGroups.geojson
KingCo_2000_Census_BlockGroups.shp -Progress
```

5. Locate a GeoJSON file

- You can download the following GeoJSON file for next step if needed
 - [King County 2000 Census Block Groups GeoJSON](#)

6. Install a Tippecanoe container which is a utility tool to create vector tiles

- Search for Tippecanoe on [Docker Hub](#)
- Select the [jskeates/tippecanoe repository](#)
- Copy the appropriate command from the *Docker Pull Command* section of the page
- Paste it at the Quick Start Terminal prompt, and hit enter to run it

```
docker pull jskeates/tippecanoe
```

7. Create some vector tiles

- Ensure Docker is running on your computer
- Start Tippecanoe container in interactive mode at the Quick Start Terminal prompt

```
docker run -it -v $HOME:/home/tippecanoe jskeates/tippecanoe:latest
```

- You will see your command prompt change to look like bash-4.3\$
- Use the tippecanoe command at the Terminal prompt to create vector tiles from the geoJSON file

```
tippecanoe -o KingCo_2000_Census_BlockGroups.mbtiles
KingCo_2000_Census_BlockGroups.geojson
```

- Exit the container when it is done

```
exit
```

- The vector tiles will be \$HOME/KingCo_2000_Census_BlockGroups.mbtiles
 - \$HOME represents your user directory at a unix style command prompt.

8. Install a TileServer GL container

- Go to [Docker Hub](#)
- Search for `tileserver-gl`
- Select [klokantech/tileserver-gl](#)
- Copy the Docker pull command & run it at the Quick Start Terminal prompt

```
docker pull klokantech/tileserver-gl
```

9. Run TileServer GL

- Ensure Docker is running on your computer
 - From the command line change into the directory `cd $HOME` where you have placed your `mbtiles` file
 - Start the TileServer GL container from the Quick Start Terminal prompt
- ```
docker run --rm -it -v $(pwd):/data -p 8080:80 klokantech/tileserver-gl
```
- Test that the vector tiles are being served by entering <http://192.168.99.100:8080/> into your browser's address bar
  - After testing, hit **ctl-C** to quit TileServer GL

# Installing and Using Maputnik, Vector Tile Style Editor.

## What is Maputnik?

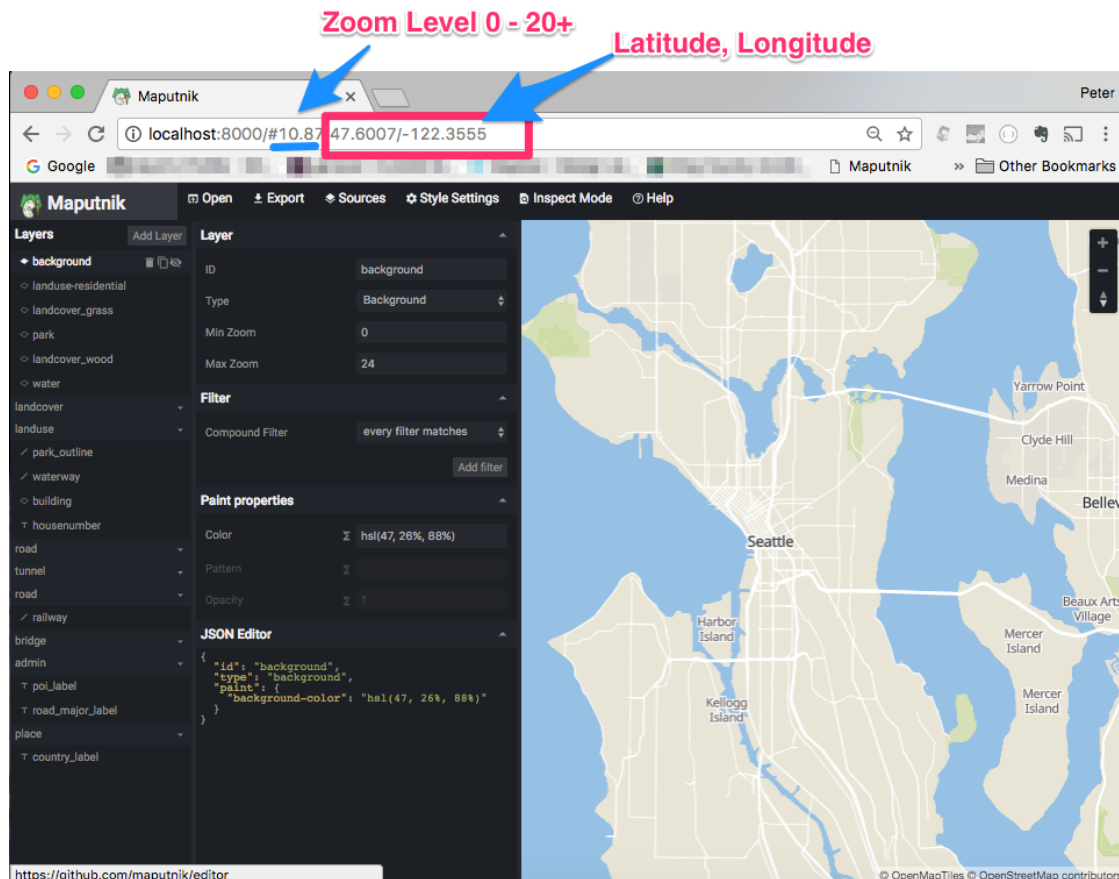
Maputnik is a vector tile visual style editor that is easy to use, free and open source. It is a style editor with immediate feedback to create Mapbox GL styles using the [Mapbox GL style specification](#). Maputnik does not require a cloud based deployment and will run on your local machine.

## Installing Maputnik

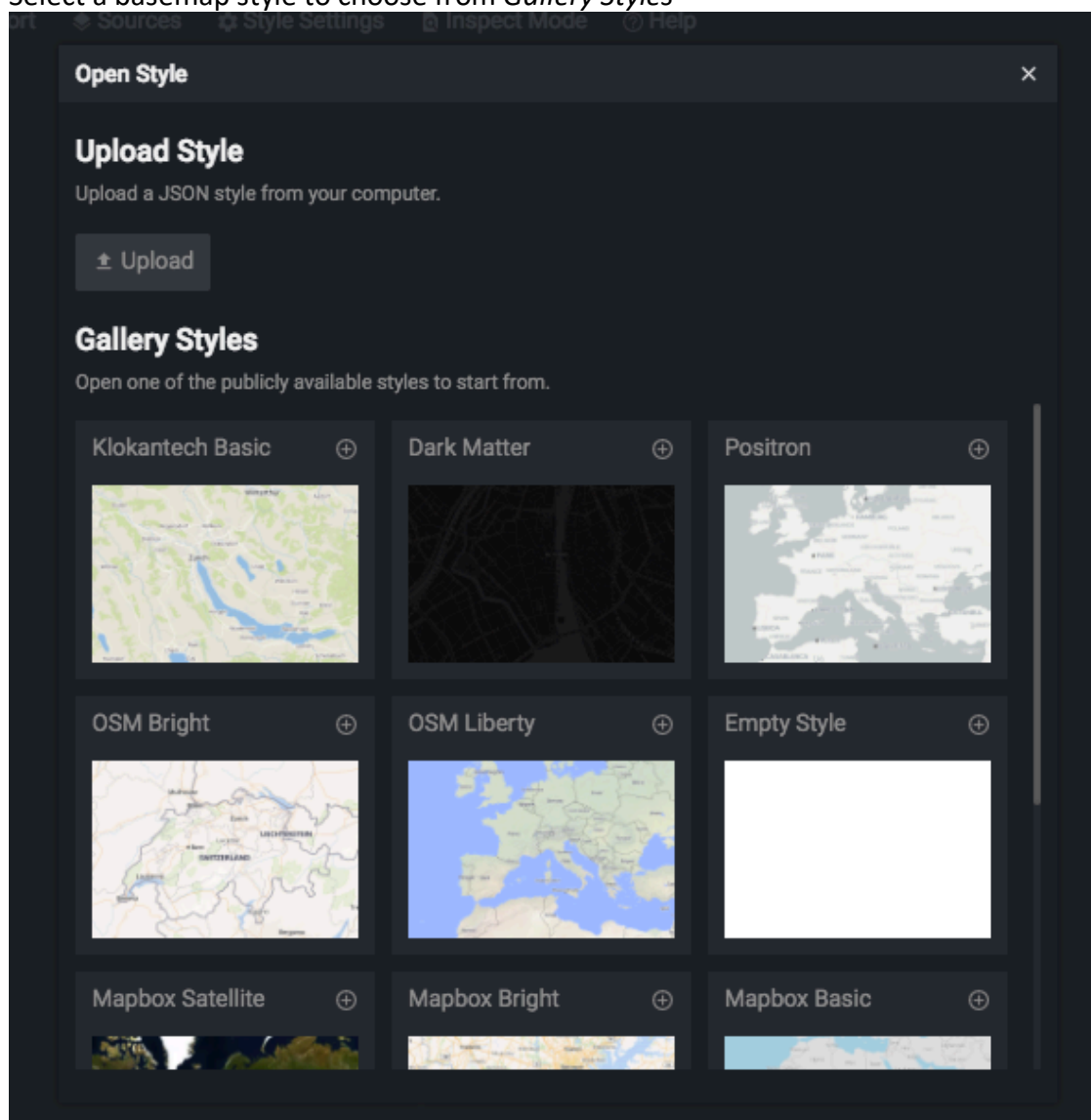
1. Download the latest Maputnik release from [here](#).
  - Under *Assets*, click on and download *public.zip* to your local drive.
2. Move the unzipped contents of *public.zip* into a directory (i.e \$Home/Document/Maputnik/public).
3. Open your terminal window, and be sure you are in the directory with those contents.
4. Run your web-server to start the Maputnik style editor: `http-server -p 8000`

## Using Maputnik

1. Go open a browser and enter [localhost:8000](#) to start styling vector tiles with Maputnik.



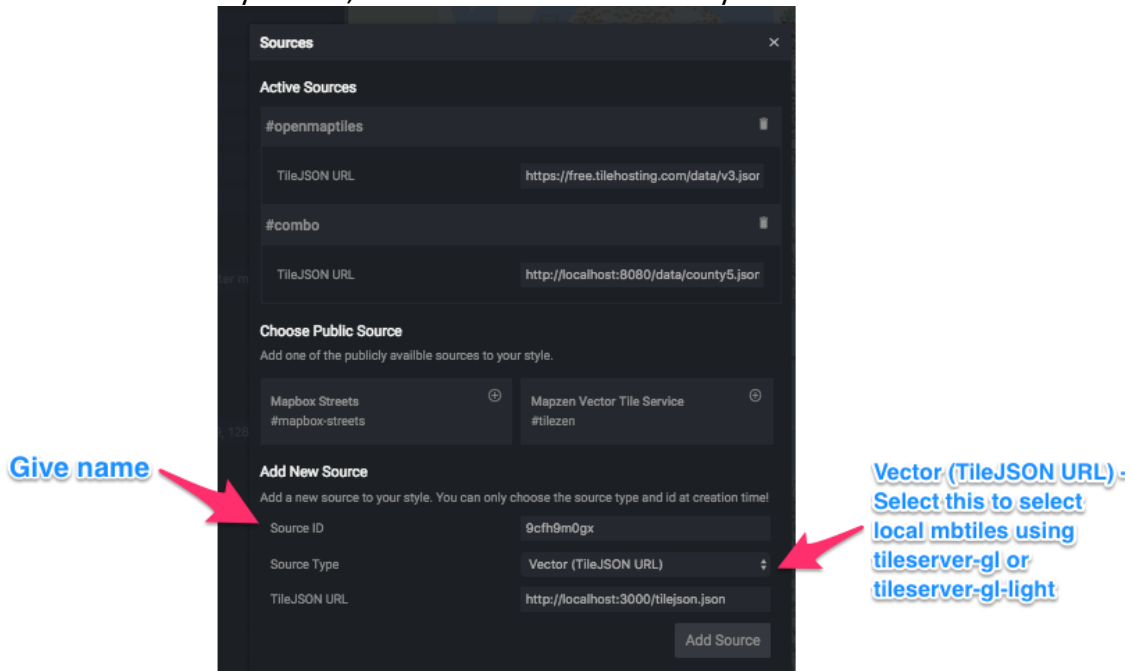
2. Select a basemap style to choose from *Gallery Styles*



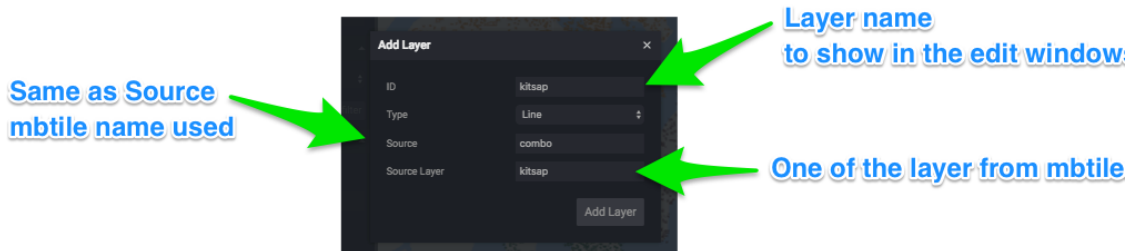
3. Start Tilesserver-gl to serve the previously generated vector tiles.
  - See previous Windows Pro Instructions [step 9](#)
  - See previous MacOS Instructions [step 8](#)
  - See previous Windows Home Instructions [step 9](#)
4. Load the King County Block Group vector tiles.

5. Load the data by clicking on *Sources*

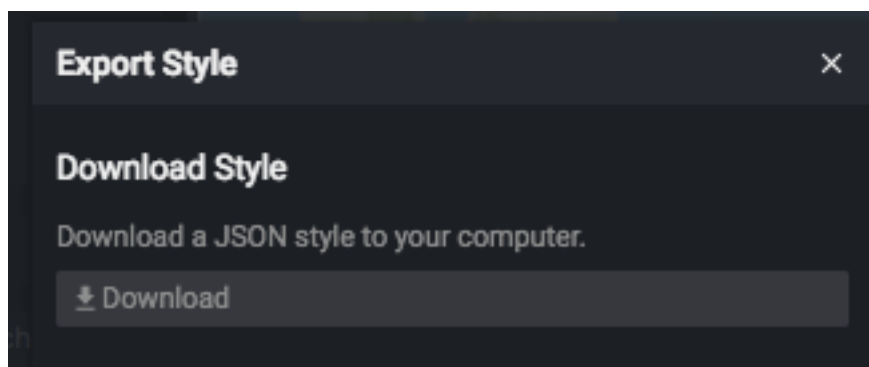
- **Hint:** To delete any source, one must remove it from layer list first.



- *Source ID*: Give a name associate with the data
- *Source Type*: Select GeoJSON or Vector (TileJSON URL)
- *TileJSON URL*: Copy url of locally hosted vector tile



6. Export Style JSON file locally. This includes references to the basemap, and your own styled vector tiles.



## Creating Simple Web Map from Vector Tiles

1. Download *index.html* from [PKgeo's GitHub page](#) and save into new clean directory.
2. Copy the previously exported *style.json* file into the same directory location created in step 1 above.
3. Open *style.json* in a text editor.

- Edit line 13 and 14. Change the coordinates to 47.613 and -122.3988.

```
"center": [
 47.613,
 -122.3988
],
```

- Make sure to enter your own [Tilehosting](#) Maps API key.

```
"sources": {
 "openmaptiles": {
 "type": "vector",
 "url": "https://free.tilehosting.com/data/v3.json?key=type your API key"
 }
},
"sprite": "https://openmaptiles.github.io/klokantech-basic-gl-style/sprite",
"glyphs": "https://free.tilehosting.com/fonts/{fontstack}/{range}.pbf?key=type your API key",
```

4. Go to [Bl.ock Builder](#)
  - Click the `_Start Coding+` button at the top right.
  - Click *login* at the top right and login with your GitHub credentials.
  - Delete the existing code in the *index.html* tab, and copy in the contents of your *index.html* file from step 1 above.
  - Press the `+` button at the top far right to add a new file then name it *style.json*.
  - Copy in the contents of your *style.json* file from step 2 above.
  - Click the *Save* button at the top.
  - Once saved, click *view bl.ock* and a browser new tab will open with a map view showing your Maputnik style.