

# Development of an Open Source Enterprise GIS

Russell Toler

Masters of Science in Geographic Information Science (MSGISci)  
Department of Geography, California State University, Long Beach

## Introduction

My objective was to test the viability of a department-wide enterprise open-source GIS. In my undertaking of this project I learned how to set up such a system, assessed its strengths and weaknesses, and determined what skills and knowledge are necessary on the part of the administrator. The prototype solution was made applicable to the City of El Segundo Planning Department.

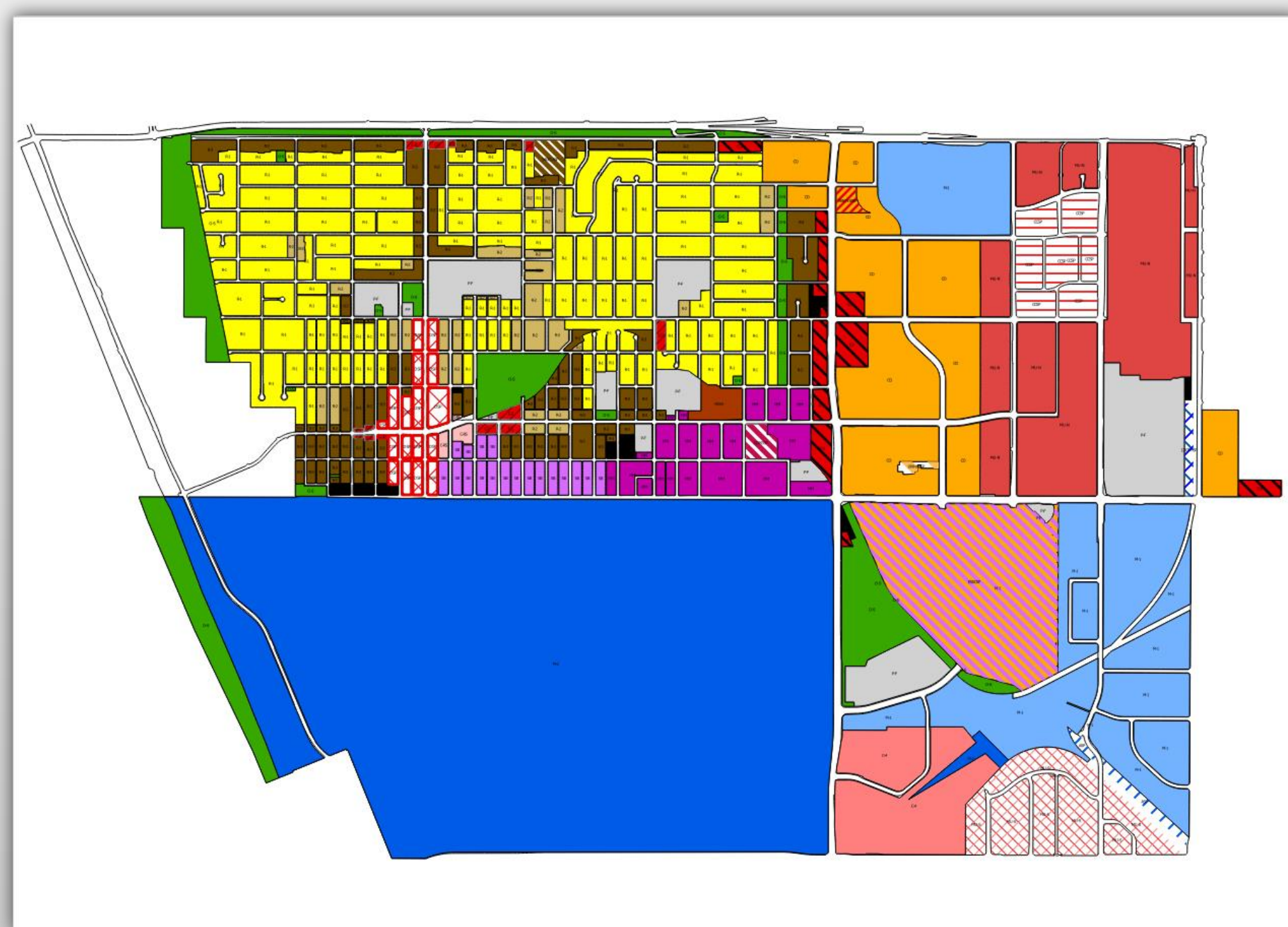


Figure 1. City of El Segundo Zoning Map

## Data and Data Sources

The data utilized for this project came from the Los Angeles County GIS Data Portal as well as the GIS Administrator at the City of El Segundo. All other software used in this project was downloaded freely from the web.

Table 1. List of data and data sources used in the project

Dataset	Source
City boundary	LA County GIS Portal
LA parcels	LA County GIS Portal
LA building footprints	LA County GIS Portal
LA streets	LA County GIS Portal
ES zoning	ES GIS Administrator
ES DSP	ES GIS Administrator
ES Specific Plans	ES GIS Administrator

## Methodology

My methodology was straightforward. I built the enterprise system step by step using all open source software. I first set up a new Amazon Web Server (AWS) EC2 instance, then installed PostgreSQL and PostGIS, then QGIS. Through PostGIS I loaded layers into the enterprise database, and within PostgreSQL I assigned user roles and privileges.

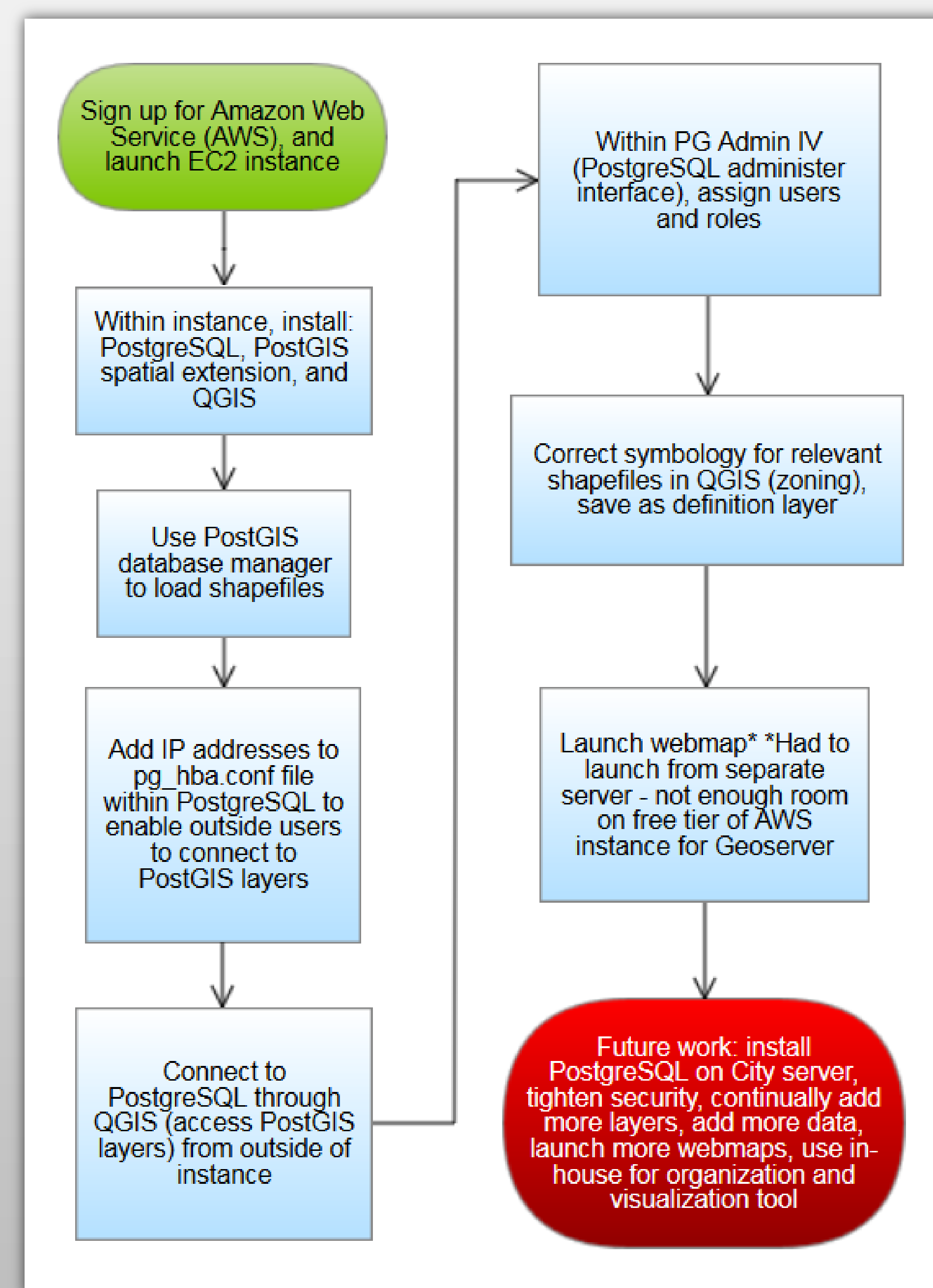


Figure 2. Project workflow

## Timeline

Project Timeline			
Task	Planned	Actual	
Acquire and organize data	5/1	4/28	
Configure AWS virtual server instance	5/8	5/7	
Install and configure PostgreSQL and QGIS, load layers	5/15	5/28	
Resolve firewall issue (enable connection to RDBMS from outside of virtual server)	5/22	6/2	
Collect and add new data (downtown parking inventory)	5/29	Did not happen	
Experiment assigned user login privileges on co-worker	6/5	Did not happen	
Run advanced GIS analysis on new data	6/12	Did not happen	
Launch web map	6/19	Did not happen*	

\*The web map was launched, but never hosted. The map exists in a number of qgis2web files, but is inaccessible through a web browser.

## Results

The result of my project was a working prototype of an open source enterprise GIS. The point of an enterprise system is for multiple users to be able to access and work with the same shared data. I was able to successfully configure such a system, with nineteen layers and five users.

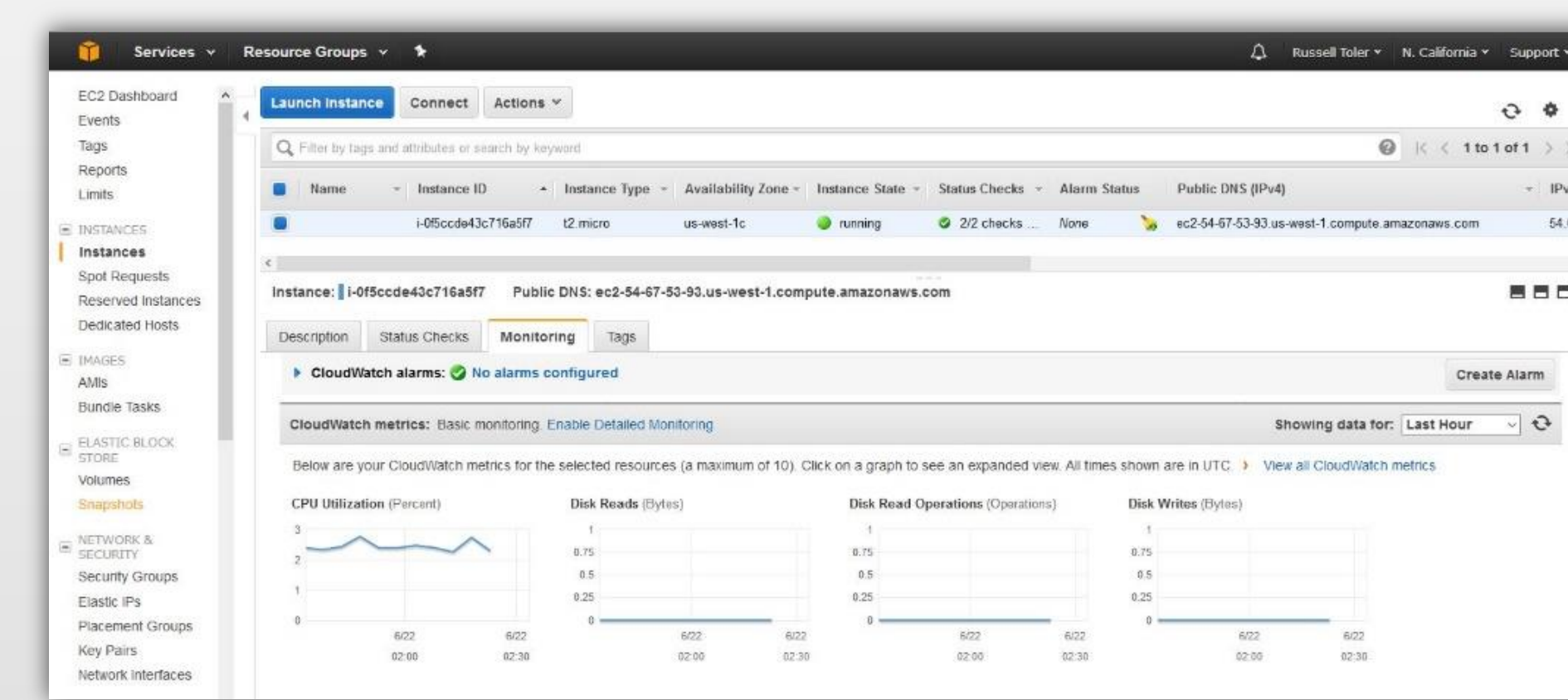


Figure 3. The control console for my AWS EC2 instance

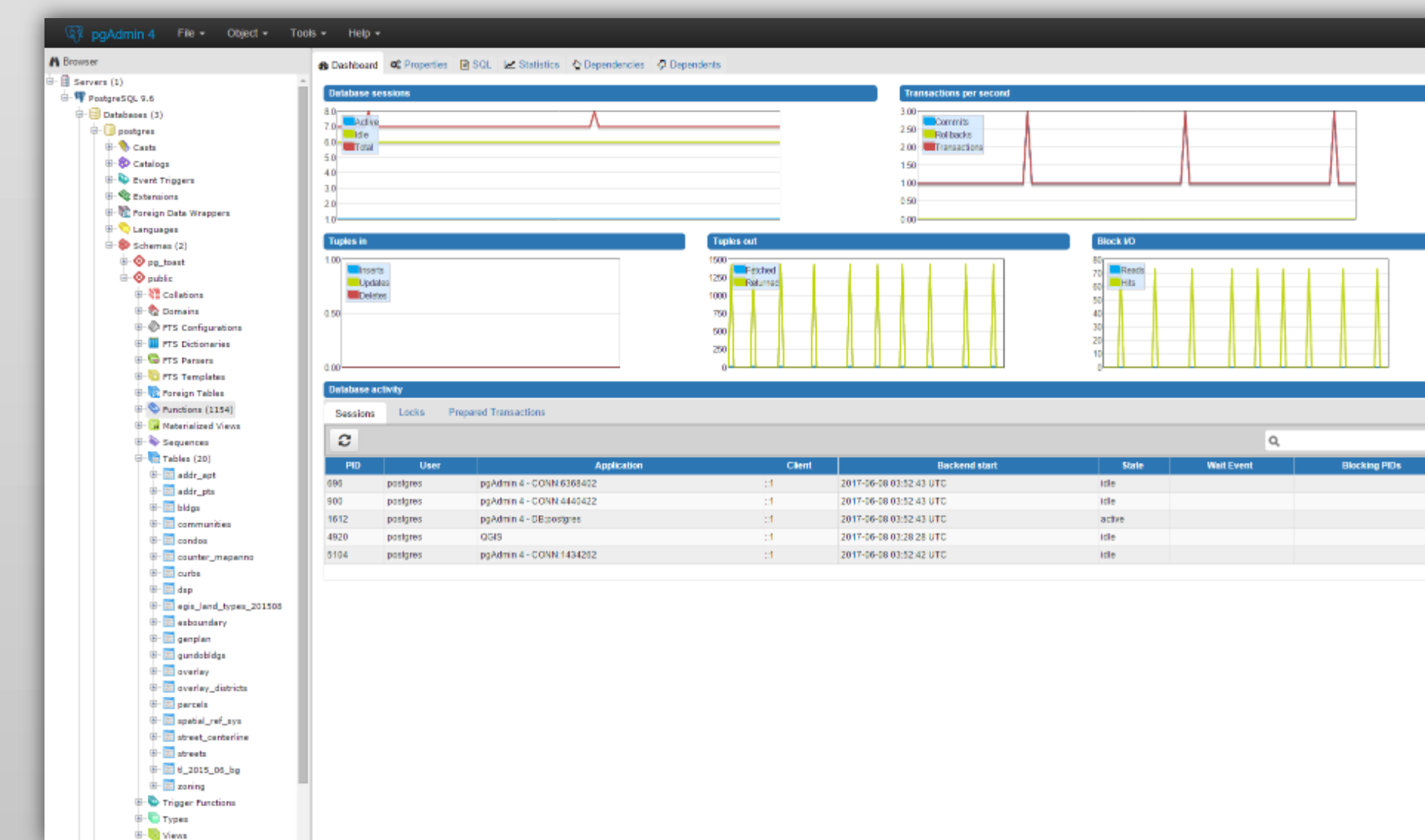


Figure 4. PostgreSQL administrator dashboard

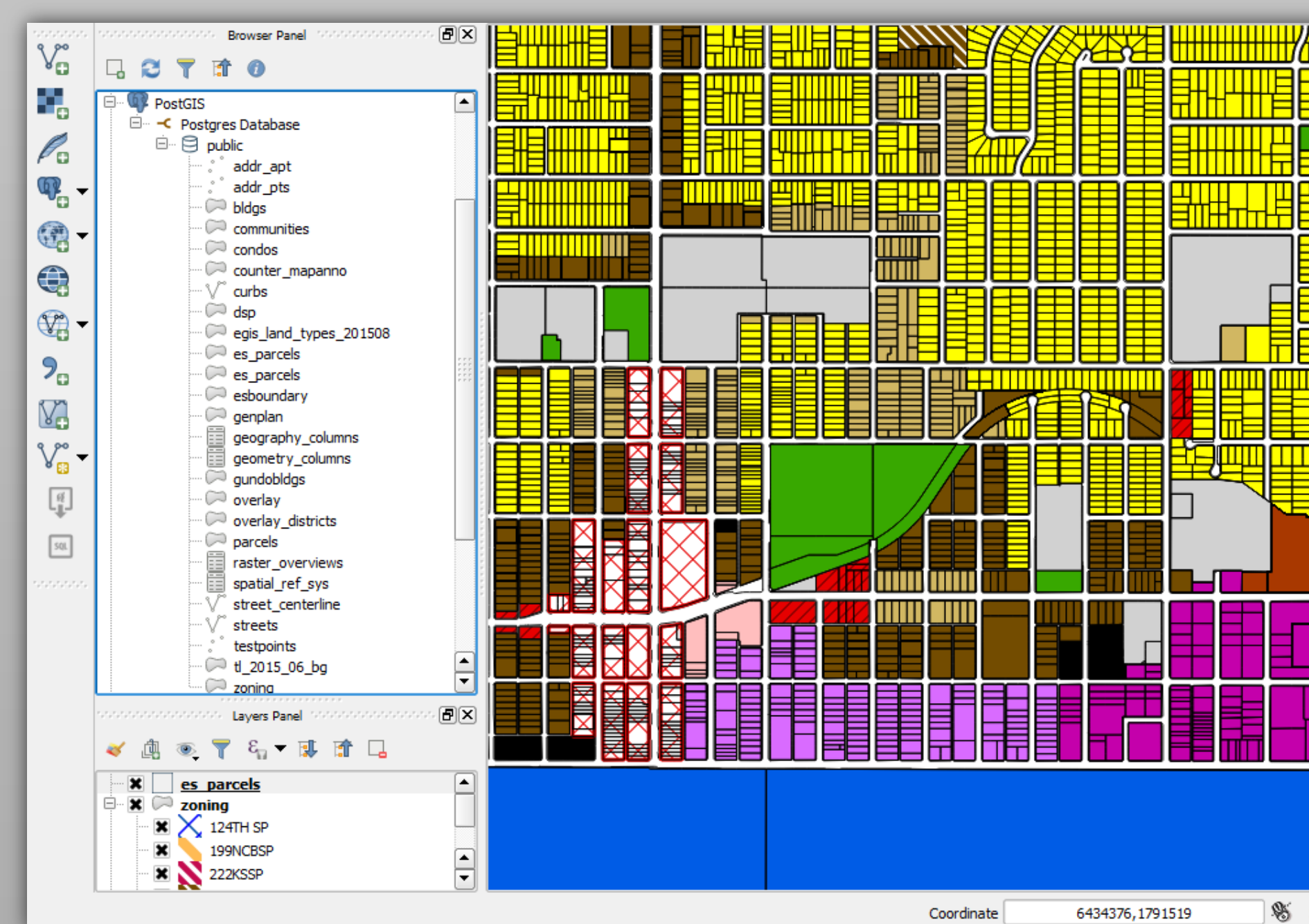


Figure 5. Successful QGIS connection to shared layers stored in PostgreSQL database



Figure 6. Individual and group logins, each tailored to have separate viewing and editing rights to database layers

## Discussion

Overall, I found an open source enterprise GIS solution to be viable in a small municipal context. However, this was not able to be tested with the rigor that I had intended. It may not be an impressive replacement to a proprietary counterpart, but can raise a department's levels of organization and efficiency at the very least. Under the management of a capable administrator, much more than this is possible, including almost anything proprietary GIS is used for. However, if one is going to administer such a system, and especially if they intend to maximize its potential, they should have a firm grasp on PostgreSQL, QGIS, Python, and internet security.

Alternative methodologies may have caused less pain. The utilization of a proprietary software would have undoubtedly provided a smoother ride - but this would have gone against my specific desire to learn about OSS and would have been expensive for my department to adopt.

Given the nature of OSS, enterprise systems have a higher chance of being vulnerable, configured poorly or otherwise sloppy. This sort of situation puts more pressure on the administrator to take full charge of and responsibility for the system and ensure that the best quality work possible is always being done. This applies to both the quality of the system itself and to the results that the system enables the production of. In my case, given the pressure of time and my impatience with my slow web server, I tended to move on to the next step as soon as I was able to complete the previous step without first determining if I had found the optimal method for completing that step. Further testing would have been ideal.

## Conclusion

The project was a success, but I would do some things differently if I were to do it again. The lowest (free) tier of Amazon's EC2 options is not powerful enough for the demands that this project required. Rather than use a cloud-based server, I could have set up on a home or portable server. Since the project was an example prototype only, this would have been a viable option. However, given the rise in popularity of cloud-based hosting, the use of AWS added an interesting detail to the project.

The open-source software used (PostgreSQL and QGIS) are powerful, but less intuitive than their proprietary counterparts. This means that a successful system depends heavily on the skills of the administrator. Practically, it is unlikely that an agency that uses open source software to avoid licensing fees has at the same time personnel sharp enough to administer and to be trained in a more onerous system.

But there are other reasons that a public agency might take the open source route, including the appearance of responsible use of taxpayer money, along with more customizability of the software.

Submitted in partial fulfillment of the requirements of the Masters of Science in Geographic Information Science (MSGISci), August 12, 2017.

For additional information please contact: Russell Toler  
russell.toler@gmail.com