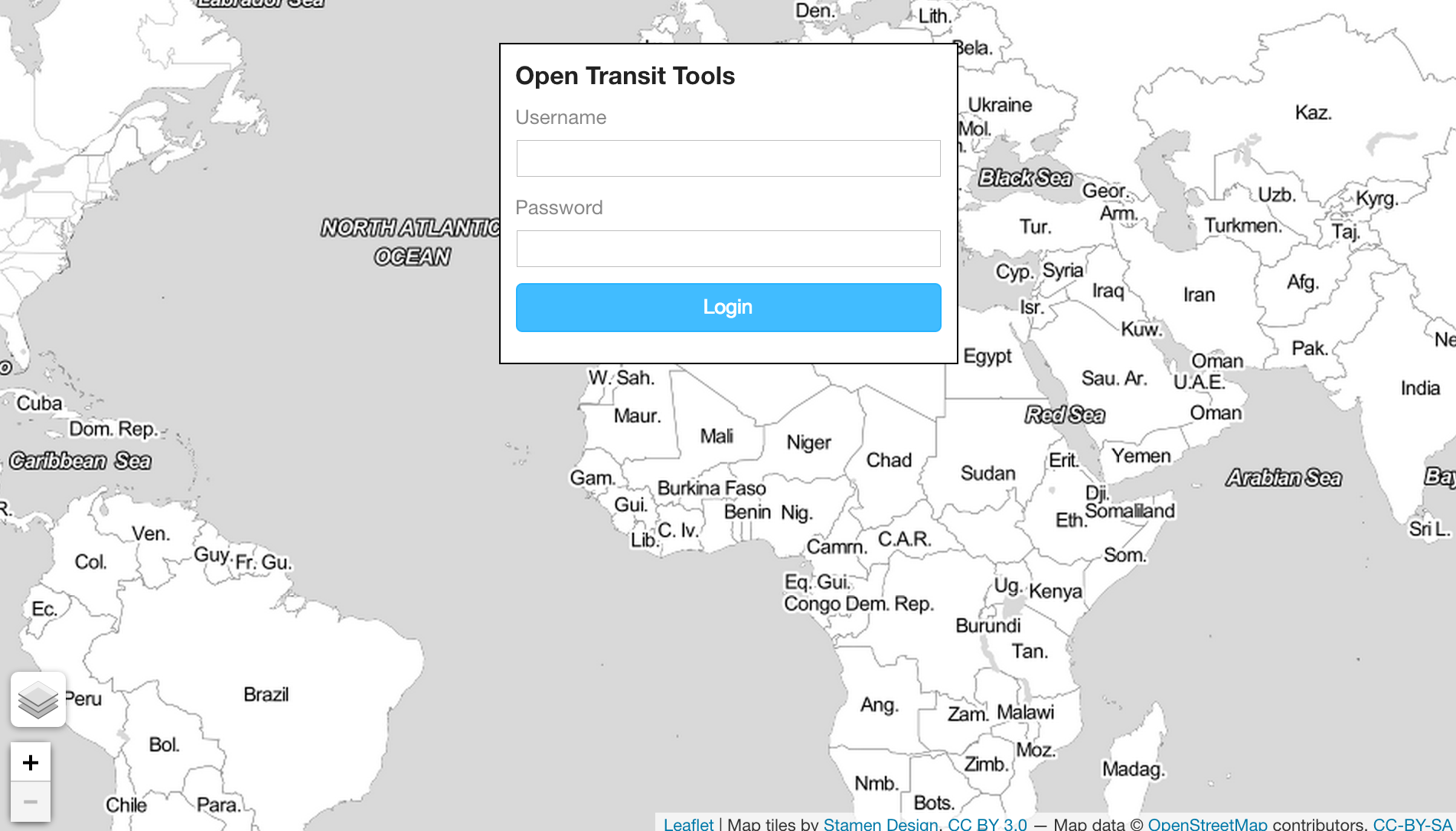
# Open Transit Indicators

Server Administrators’ Guide



|  |  |
| --- | --- |
| Prepared for:  http://www.worldbank.org/content/dam/Worldbank/WBG_Horizontal-RGB-web_300x59.jpg  1818 H Street, NW  Washington, DC 20433 | By:  **Description: Description: azavea_RGB_72dpi_white_sm**  340 N 12th St, Suite 402  Philadelphia, PA 19107  (215) 925-2600  <http://www.azavea.com> |

# Table of Contents

[Open Transit Indicators 1](#_Toc407386686)

[Table of Contents 2](#_Toc407386687)

[Introduction 3](#_Toc407386688)

[System Requirements 3](#_Toc407386689)

[Installation 3](#_Toc407386690)

[Standard Server Installation 4](#_Toc407386691)

[Vagrant Test Machine 9](#_Toc407386692)

[Deployment to Amazon EC2 10](#_Toc407386693)

[Application Configuration 10](#_Toc407386694)

[User Management 10](#_Toc407386695)

[System Monitoring 11](#_Toc407386696)

[Troubleshooting 12](#_Toc407386697)

[Acknowledgements 13](#_Toc407386698)

Open Transit Indicators

System Administrators’ Guide

# Introduction

This manual is intended for the system administrator responsible for installing and maintaining the Open Transit Indicators application. Users of the application may refer to the End Users’ Manual.

# System Requirements

The Open Transit Indicators application runs on AMD 64 bit Ubuntu 12.04. The server should have the following minimum resources:

* 8GB RAM
* 2 core CPU
* 24GB Storage

Additional software may be required on the administrator’s client machine. This guide assumes familiarity with SSH and administering Linux OS servers. Windows users may require PuTTy or similar software in order to SSH into the application Server.

Installation times will vary depending on local internet speed. There are several large code libraries upon which the application depends. With a high-speed connection, the installation process takes approximately 30 minutes.

# Installation

There are multiple options for deploying the application.

For testing purposes, installing a virtual machine (VM) using Vagrant (<https://www.vagrantup.com>) is a fast and simple way to run the application. Note that the machine is configured to use 8GB of RAM, so the test VM should have at least that much allocated.

If installing on a webserver directly, a provisioning shell script is included that will gather and install all of the dependencies, set various configurations of the server software, and launch several processes so that the app is ready to run when provisioning is complete.

If deployment to Amazon EC2 is desirable, there is a configuration template that makes building and deploying an Amazon Machine Image (AMI) using Packer (<http://www.packer.io/>) very straightforward.

## Standard Server Installation

SSH into the server that has had Ubuntu 12.04 (AMD64) freshly installed.

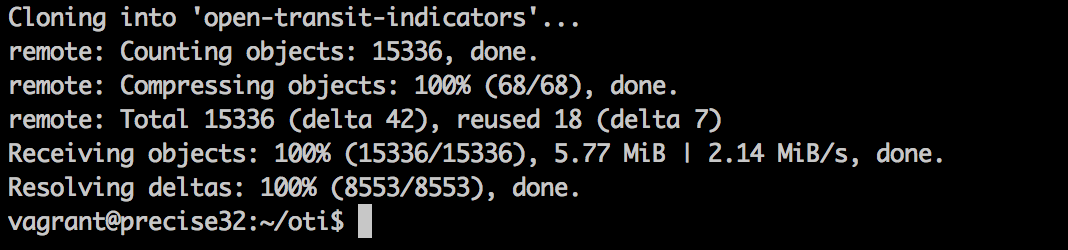
First, update the installation and install Git and your preferred text editor (Vim, Nano, etc.).

sudo apt-get update

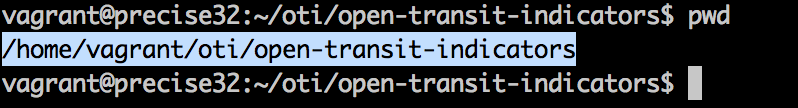
sudo apt-get -y install git



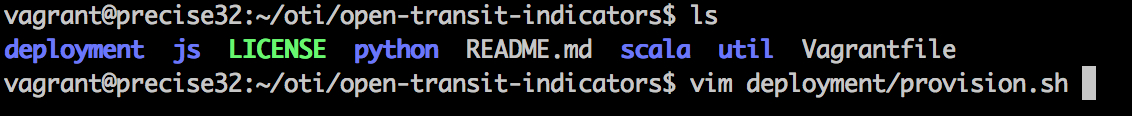
Create a directory in which to install the application. This could be /projects, ~/app, or anything else – this location will be set in the deployment script later. In the example, the application will be installed to the ~/oti directory. Change directories into the install location and clone the git repository.



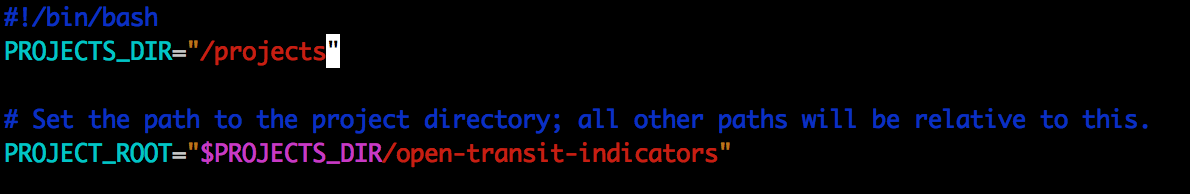
The project code will be pulled down from GitHub and cloned into the application directory in the server. In this case, it will be in ~/oti/open-transit-indicators as shown in the screenshot below.



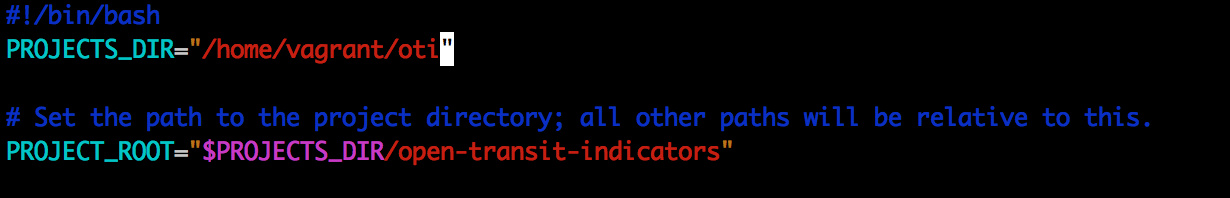
Double-check the location of the installation using the command ‘pwd’. This location will be used in the next step.



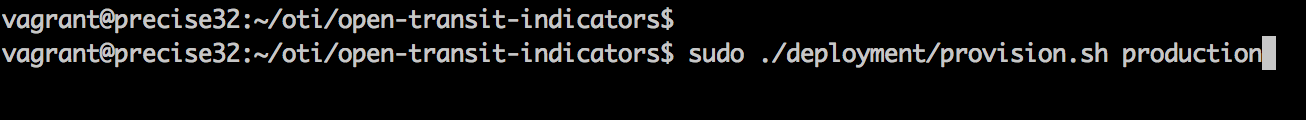
Before installing the software on the server, modify the deployment/provision.sh script to reflect the location of the codebase using your preferred text editor.



The 2nd line of the script will look like the above screenshot initially.



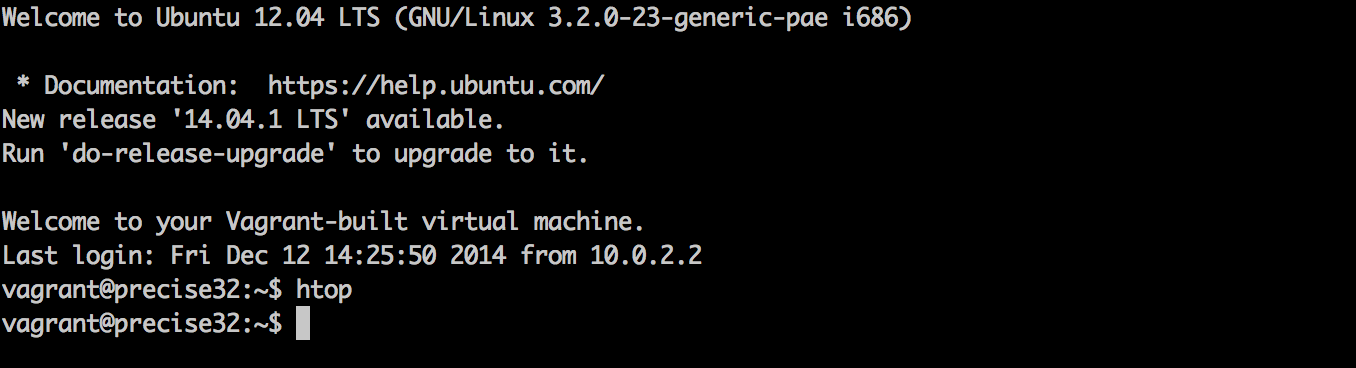
Modify the PROJECTS\_DIR value to match the directory that contains /open-transit-indicators.



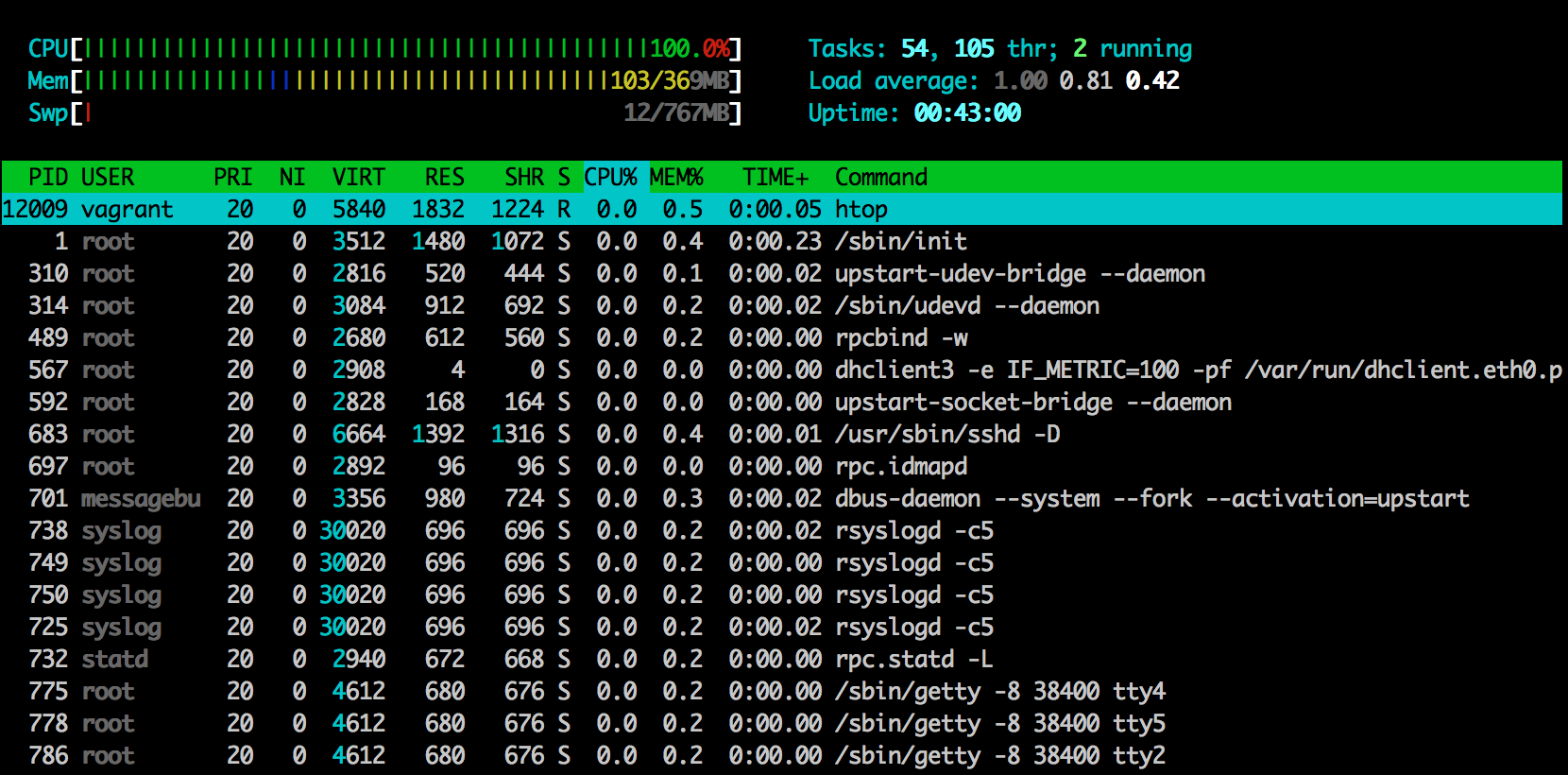
After modifying the provisioning script, run it using the above command with the production modifier.

The provisioning script will automatically configure the server with software and settings required to run the application. On a high-speed connection, this process will take approximately 30 minutes. If errors are encountered, it may be due to a repository being down or to a network connection issue. Starting over after a failure will skip the parts that were completed and continue from that point.

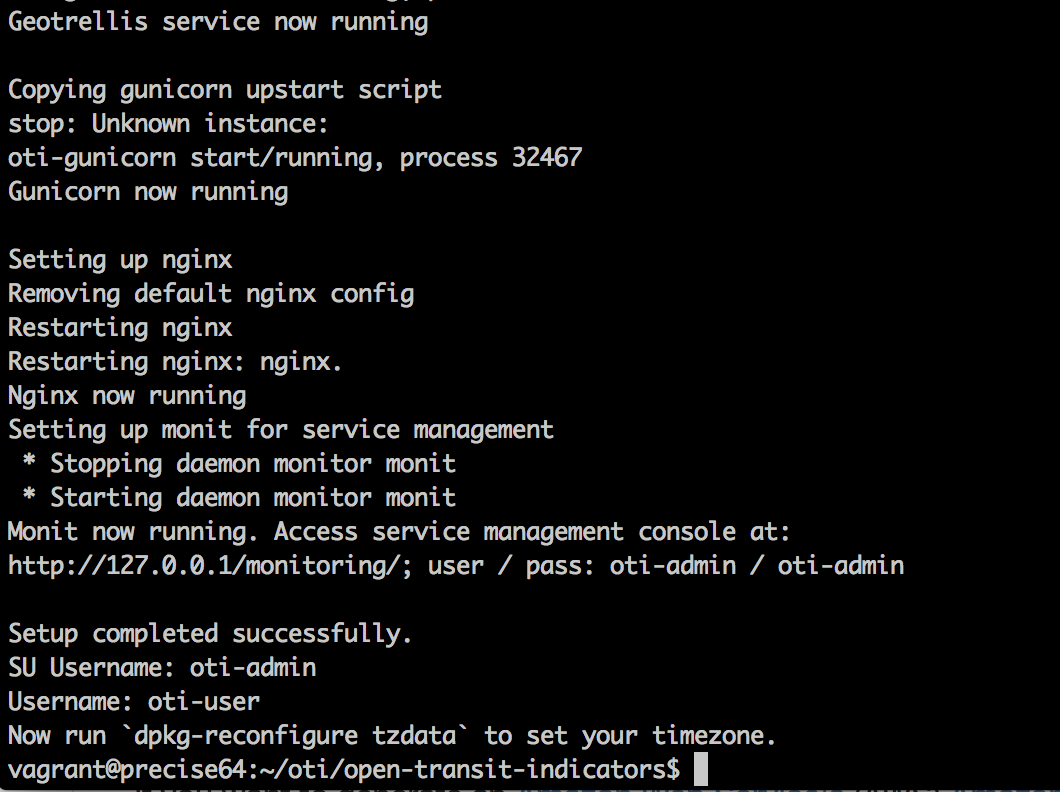
Make sure the system is configured with the appropriate memory and compute resources to ensure a successful installation.



Watching the system processes with ‘htop’ will show that the installation is working. It should take nearly 100% of the CPU while installing, unless it is waiting for files to download.



HTOP displays resource usage during installation.



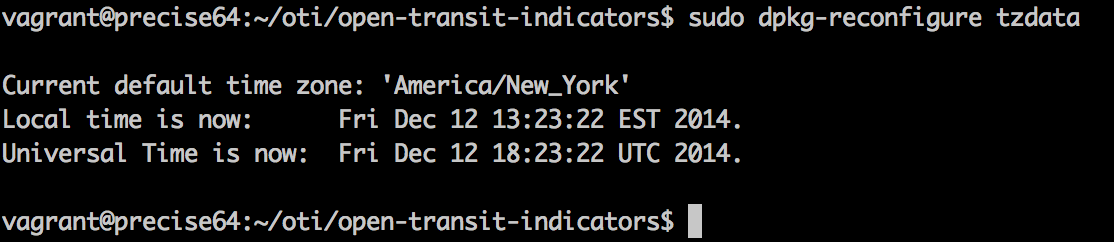
When the provisioning script is complete, it will display success messages as well as the default administrator user name and password. There is also a message recommending time zone configuration for the server. This is an important configuration to support scenario design.

Macintosh HD:Users:jbranigan:Desktop:oti-install:Screen Shot 2014-12-12 at 1.22.34 PM.png

With ‘sudo’, run the command noted at the end of the provisioning script.

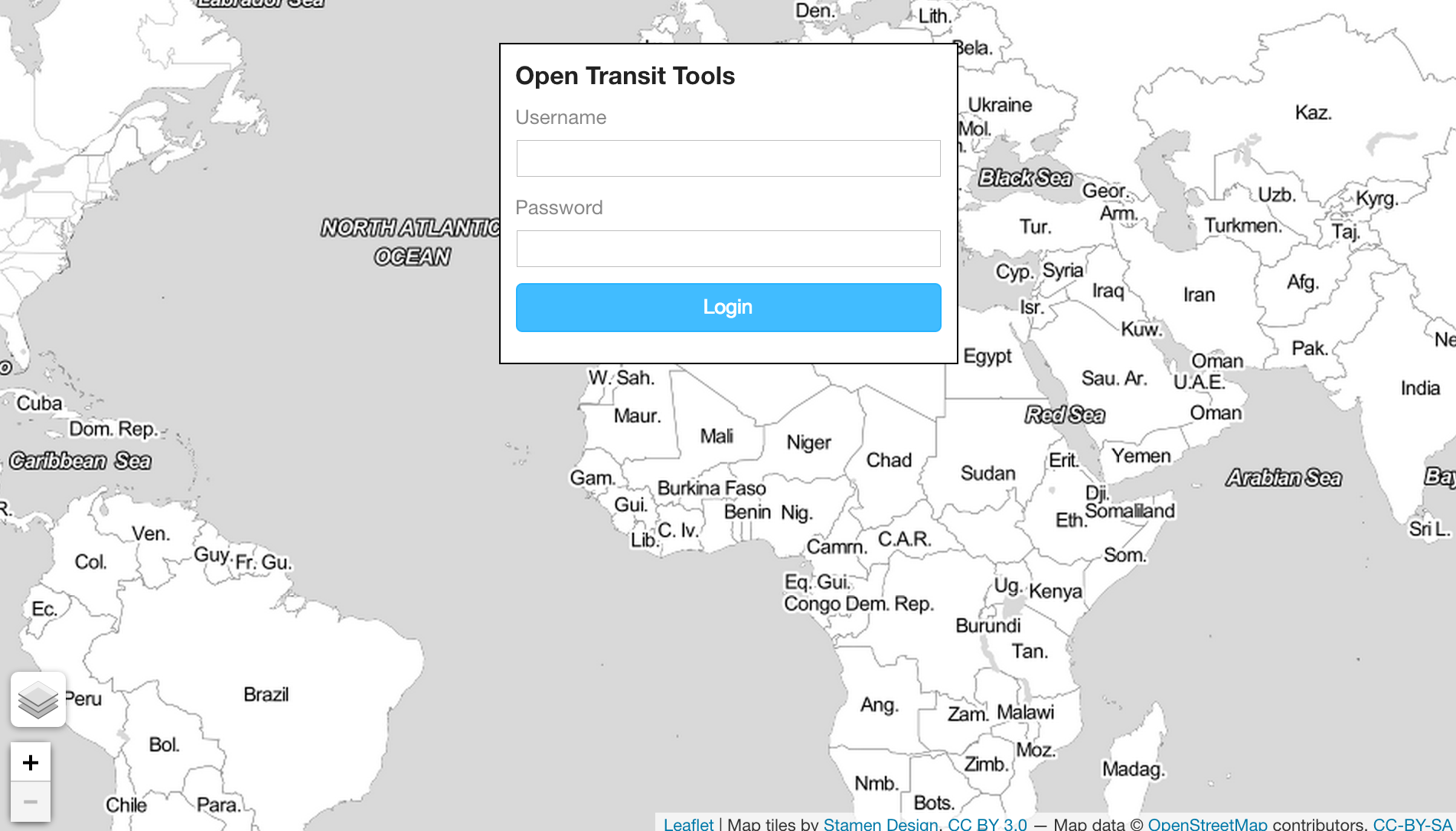


Choose the time zone to match the users’ browser settings. For example, if the server is in Seattle, but the users are in New York, set the time zone to New York.

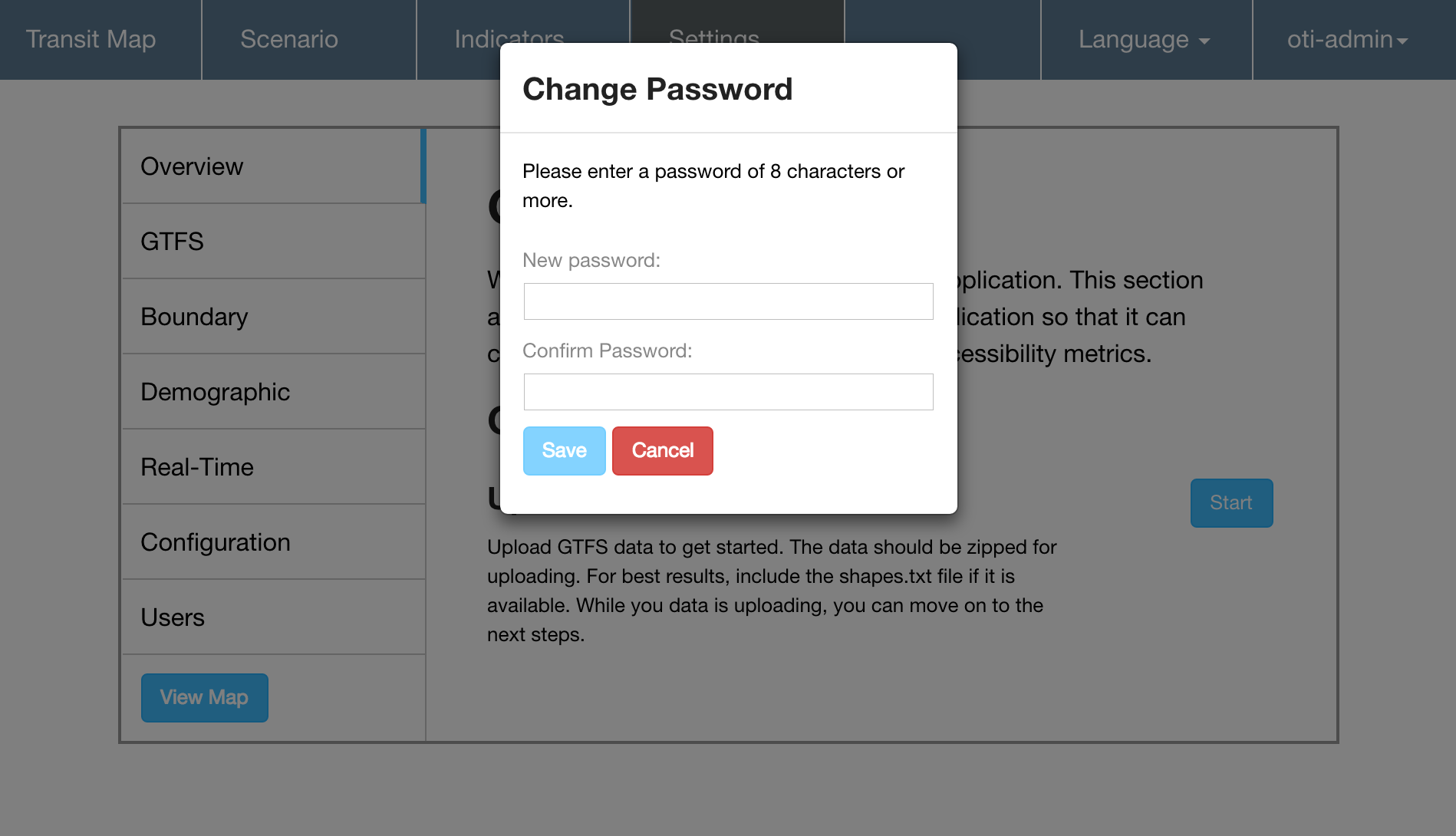


When complete, the server will confirm the time zone setting has been changed.

The installation process is complete. Navigate in the browser to the IP address or URL of the webserver.



Log in with the default credentials. Click on the ‘oti-admin’ button at the top right after logging in, and change the password.



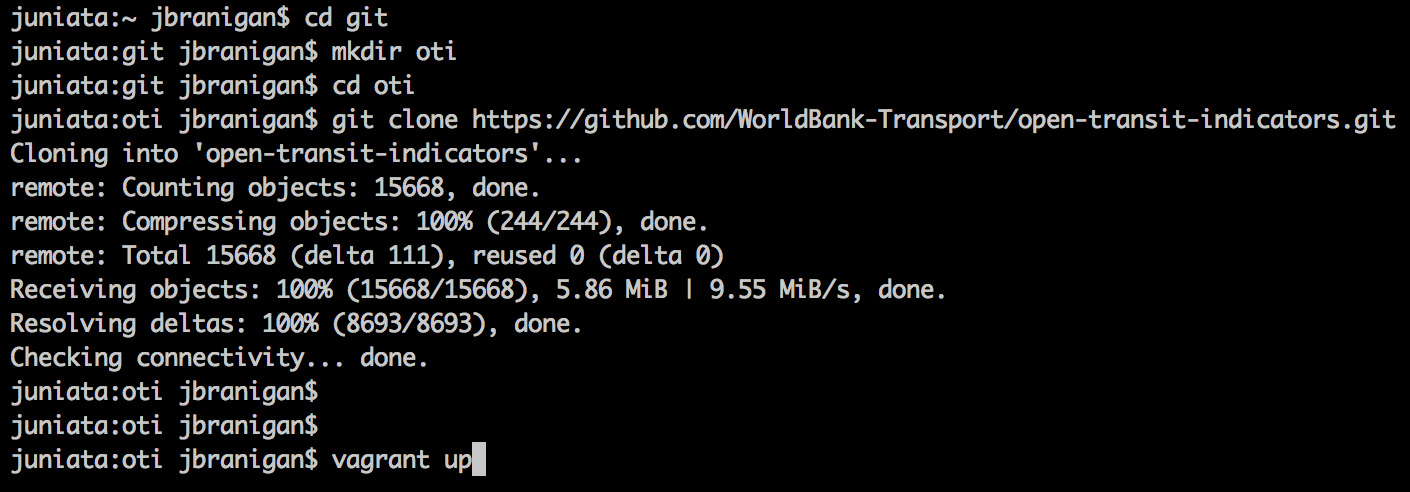
After changing the password, you are now ready to configure the software with your city’s GTFS and GIS data.

## Vagrant Test Machine

If you're using Vagrant, then installation is as simple as cloning the repo and then issuing vagrant up from the root of the repository directory.

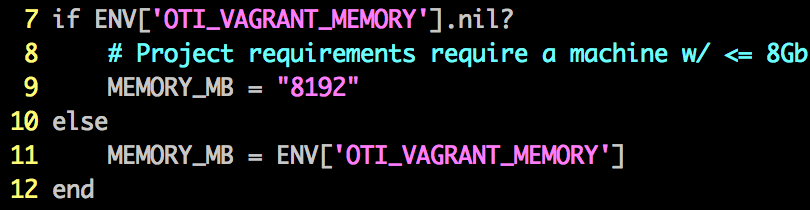
Vagrant is a virtual machine integration tool that provides an integrated development environment on a local development machine. To install Vagrant with VirtualBox (both software packages are free), follow the instructions here: <https://docs.vagrantup.com/v2/getting-started/index.html>

After installing Vagrant, clone the repository from GitHub (<https://github.com/WorldBank-Transport/open-transit-indicators>) to a directory, navigate to it, and issue the command ‘vagrant up’.



Vagrant will download the appropriate Ubuntu machine image and then provision the machine with all necessary dependencies, just like the provision.sh script in the full server installation. Once the provisioning is complete, the application will be available at http://localhost:8067 in the browser.

The default amount of memory that Vagrant will allocate to the VM is 8GB (8192MB). To change the default amount of memory allocated to the vagrant machine, modify the Vagrantfile in the project root directory. Set the environment variable OTI\_VAGRANT\_MEMORY on line 9 to the preferred size, in MB.



## Deployment to Amazon EC2

If you want to generate an Amazon Web Service's (AWS) Amazon Machine Image (AMI) you can use Packer to handle the provisioning.

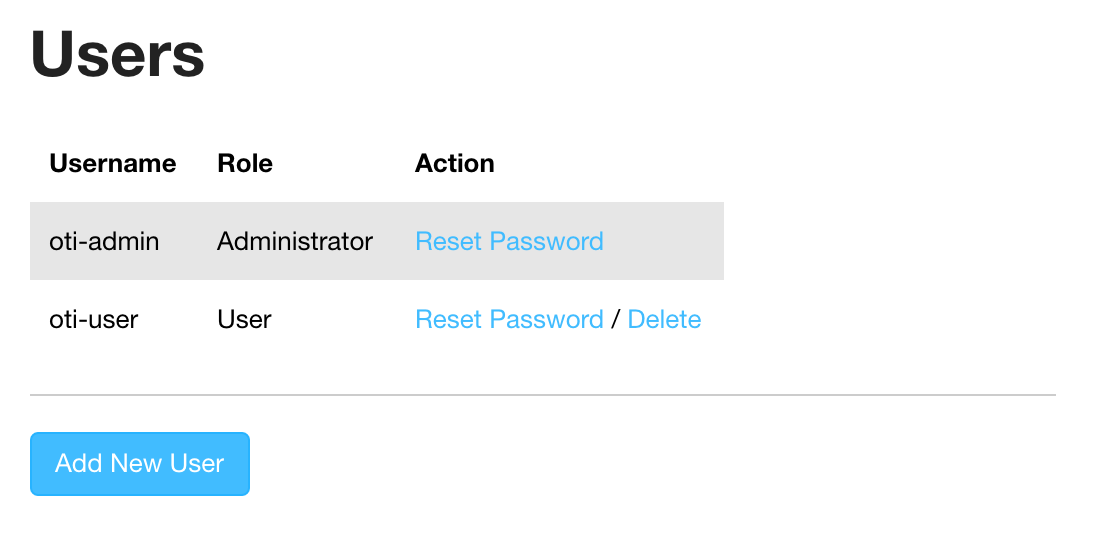
Use the following instructions to do so:

1. Create an AWS account if you do not have one already
2. Get the API keys for a new user or an existing user in your AWS account
3. Download and install Packer (<https://www.packer.io>)
4. Clone the repository to a local directory
5. Navigate to the deployment/packer directory
6. Copy the open-transit-vars.json.example file and save it as open-transit-vars.json
7. Edit open-transit-vars.json with the API keys downloaded in step 2
8. Run the following command to generate a new AMI: packer build -var-file=open-transit-vars.json open-transit-indicators.json PLEASE NOTE: Running this command will cause resources to be created in AWS and will cost money
9. Once the process finishes installing (could take up to an hour), make note of the AMI ID
10. Launch the AMI using the AWS EC2 management console and browse to the public DNS hostname provided by Amazon

# Application Configuration

Please refer to the End Users’ Guide for details on application configuration.

# User Management

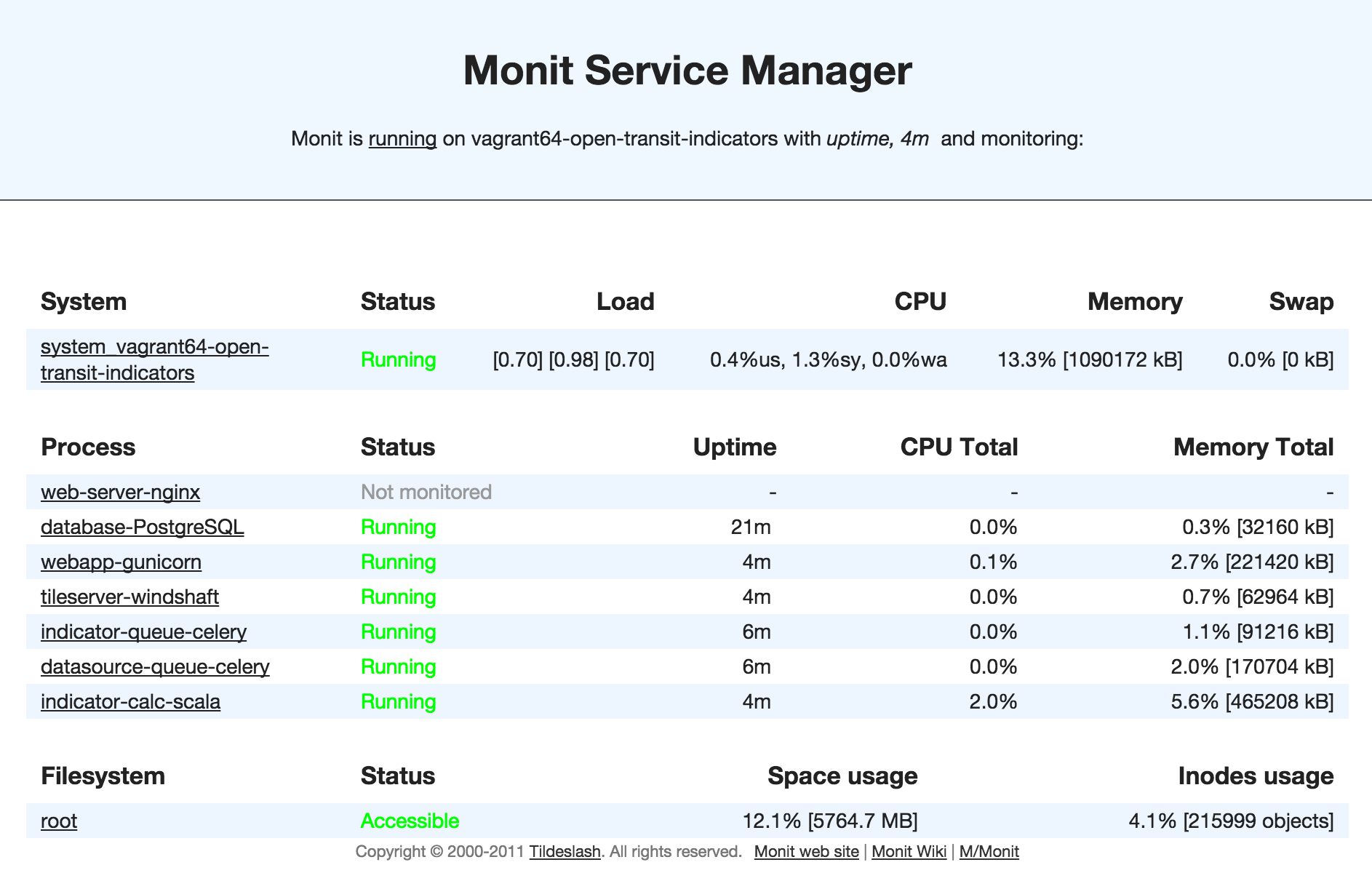


Admin users can add, delete, and manage users of the application. The system does not include email functionality, so an administrator would distribute temporary passwords to other users on account creation or password reset. There are two roles users can have:

* Administrator Capabilities
  + Configure the application datasets
  + Manage users
  + Create and view scenarios
* User Capabilities
  + View base indicator results
  + Create and view scenarios

# System Monitoring

There is a tool available for administrators to monitor various services used by the application. It can be reached at http://<host or IP or machine>/monitoring.



The application is designed to start all required services on server reboot, and to automatically restart any service that might fail during use. If an issue arises while the application is running, this page would be the first place to see if there are any errors. If one of the processes is marked as not running, click the name of the process, scroll to the bottom of the page, and click the “Start Service” button.

This page can also be consulted while the application is processing long-running tasks like indicator calculations. CPU and memory usage for “indicator-calc-scala”, in that example, should be near the top of the machine’s capacity.

# Troubleshooting

This application should be considered beta software, and may become unstable in situations that have not been encountered and tested. If the software becomes unusable even with all services running, it may be necessary to rebuild the database.

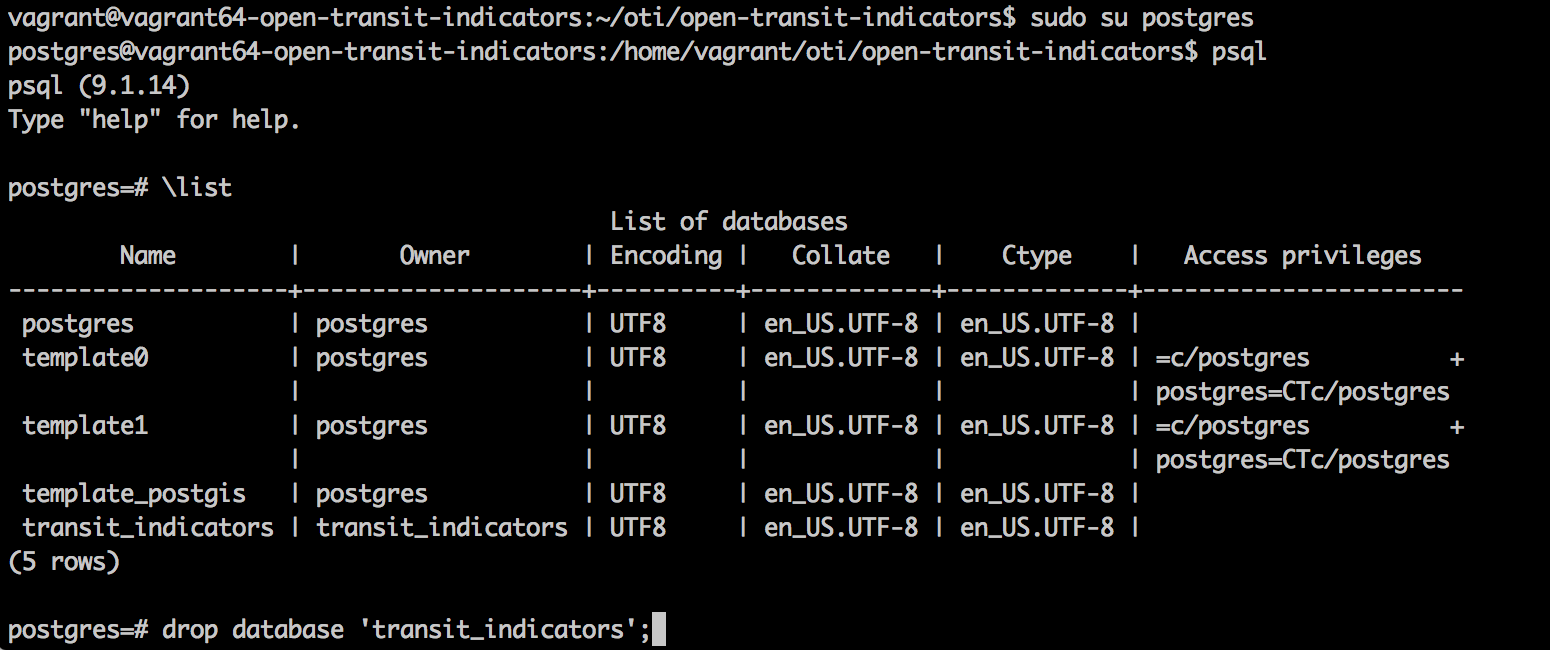
Note: this will erase the data and return the machine to default settings. Export indicator result CSVs, if possible, before following this process.

SSH into the server and switch user to “postgres” with

$ sudo su postgres

Then execute the ‘psql’ command to enter the PostgreSQL database.

List all tables with the PostgreSQL “\list” command as seen below:



To delete the database, issue the command:

drop database ‘transit\_indicators’;

Exit postgres with the “\q” command, then rerun the deployment script (/deployment/provision.sh).

# Acknowledgements

This administrators’ manual was developed to support the World Bank Open Transit Indicators project, with funding provided by AusAid.

The World Bank project team, led by **Ms. Holly Krambeck**, **Dr. Li QU**, and **Mr. Christopher DeSerio**, would like to express their sincere thanks to **Dr. Yulin JIANG**, Director of China Urban Sustainable Transport Research Center, for her unwavering support for the project and substantial inputs provided by her team, including **Dr. Cheng LI** and **Dr. Xianglong LIU**. The team would also like to thank **Mr. Lei YAN** from the Zhengzhou Bus Company for his diligent work in testing the platform and invaluable technical support he has provided to the other participating cities. And a special thanks to **Ms. Linghong ZOU**, the team intern during the summer of 2013, who helped build the critical foundations for the project.

The team would like to thank **Ms. Van Anh Thi Tran**, Senior Transportation Specialist and Task Team Leader for the Haiphong transport project, for her support for this initiative. The team would also like to express its appreciation for **Ms. Kate Chapman** and the Humanitarian OpenStreetMap Team, working with **Dr. Tran Khanh Toan** and **Professor Le Sy Xinh** from the Vietnam Maritime University’s GIS program, for their excellent work on open source digital mapping and GTFS data collection activities.

The team would also like to thank **Mr. Aaron Antrim**, founding principal of Trillium Solutions**; Mr. Michael Smith**, founding principal of Transitime; and **Ms. Bibiana McHugh**, IT Manager of Trimet – these three remarkable individuals provided critical guidance and technical knowledge to the participating pilot governments and to the team on GTFS and GTFS-RT.

The team would like to express its appreciation of **Mr. Kevin Webb**, founding principal of Conveyal, and **Mr. James Wong**, independent transit consultant, for inspiring the development of this project through their ground-breaking work in GTFS data analysis and open-source software development.

Finally, the team would like to thank **Mr. John Branigan**, Azavea GIS Project Manager, and his extensive team of talented and exceedingly hard working developers for truly going above and beyond the call of duty in building a software application to support better transit planning in developing countries.

Additional information on the project can be found from the following resources:

**Source Code Repository**

<https://github.com/WorldBank-Transport/open-transit-indicators>

**Transitime “stop\_times.txt” Generator**

***Tool for generating file required to evaluate transit system on-time performance in Open Transit Indicators***

<https://github.com/WorldBank-Transport/Transitime>

**International GTFS Training Materials: Link Repository**

<https://github.com/WorldBank-Transport/GTFS-Training-Materials/wiki/Link-repository-for-international-GTFS-training-materials>