

Ubuntu Linux Installation

Audience: Client IT, Developers

See the up-to-date documentation at <https://github.com/Ecotrust/TEKDB/wiki/installation> .

Production

Installation on Ubuntu 20.04 LTS (22.04 LTS and 18.04 LTS should be very similar)

1. Create the directory /usr/local/apps/

```
sudo mkdir /usr/local/apps
cd /usr/local/apps
sudo chmod 775 /usr/local/apps
sudo chgrp adm /usr/local/apps
```

- NOTE: The above assumes that your user has sudo privileges and is part of the adm group

2. Install Dependencies

```
sudo apt-get update
sudo apt-get install git python3 python3-pip python3-virtualenv
virtualenv gcc make uwsgi uwsgi-plugin-python3 gdal-bin
python3-gdal python3-dev build-essential -y
```

3. Clone repository into /usr/local/apps/

```
git clone https://github.com/Ecotrust/TEKDB.git
```

4. Create and activate a Python3 Virtual Environment

```
cd /usr/local/apps/TEKDB/
python3 -m pip install --user virtualenv
virtualenv env --python=python3
source /usr/local/apps/TEKDB/env/bin/activate
```

5. Install Postgres + PostGIS

< FOR Ubuntu 20.04 ('Focal') >

```
sudo apt install postgresql-12 postgresql-contrib
postgresql-server-dev-12 postgis postgresql-12-postgis-3 -y
```

< FOR Ubuntu 22.04 ('Jammy') >

```
sudo apt install postgresql-14 postgresql-contrib
postgresql-server-dev-14 postgis postgresql-14-postgis-3 -y
```

6. Create Database and DB User

Replace USERNAME with an appropriate username for your database user below. You will be prompted to provide a new password as well.

```
sudo -u postgres createuser -s -P USERNAME
sudo -u postgres createdb -O USERNAME tekdb
sudo -u postgres psql -c "CREATE EXTENSION postgis; CREATE
EXTENSION postgis_topology; CREATE EXTENSION pg_trgm;" tekdb
```

7. Install Python dependencies

```
pip install -r /usr/local/apps/TEKDB/TEKDB/requirements.txt
pip install -r /usr/local/apps/TEKDB/TEKDB/requirements_linux.txt
```

8. Enable External Access to Postgres Enable authentication from remote access

< FOR Ubuntu 18.04 >

```
sudo vim /etc/postgresql/10/main/pg_hba.conf
```

< FOR Ubuntu 20.04 >

```
sudo vim /etc/postgresql/12/main/pg_hba.conf
```

< FOR Ubuntu 22.04 >

```
sudo vim /etc/postgresql/14/main/pg_hba.conf
```

Find the section of uncommented lines near the bottom.

Add the following line to the Authentication section at the bottom:

```
host      tekdb          USERNAME    0.0.0.0/0          md5
```

Enable remote access to PostgreSQL server:

< FOR Ubuntu 18.04 >

```
sudo vim /etc/postgresql/10/main/postgresql.conf
```

< FOR Ubuntu 20.04 >

```
sudo vim /etc/postgresql/12/main/postgresql.conf
```

< FOR Ubuntu 22.04 >

```
sudo vim /etc/postgresql/14/main/postgresql.conf
```

Uncomment the 'listen_addresses' line and change it to read as such:

```
listen_addresses = '*'
```

Restart PostgreSQL Server:

```
sudo service postgresql restart
```

9. Install Proj.4 You can install from source, [as GeoDjango recommends](#), or just use a package manager (untested):

```
sudo apt install libproj-dev proj-bin -y
```

10. Update Local Settings

```
cp /usr/local/apps/TEKDB/TEKDB/TEKDB/local_settings.py.template  
/usr/local/apps/TEKDB/TEKDB/TEKDB/local_settings.py  
vim /usr/local/apps/TEKDB/TEKDB/TEKDB/local_settings.py
```

- You may use whatever text editor you prefer: emacs, nano, vim, etc...
- Set the following:

- **ALLOWED_HOSTS**: add your domain name to the list, i.e.:

```
[  
    'localhost',  
    'your.site.com'  
]
```

- NOTE: a value of '*' will allow any requests to and all urls

- **SECRET_KEY**: make something long and complex - you will not need to remember or enter it, just be sure not to share it.

- **DATABASES**:

```
{  
    'default': {  
        'ENGINE':  
'django.contrib.gis.db.backends.postgis',  
        'NAME': 'tekdb',  
        'USER': 'USERNAME',  
        'PASSWORD': 'PASSWORD',  
        'HOST': 'localhost',  
        'PORT': 5432  
    }  
}
```

11. [OPTIONAL] DJ Shortcut Many of the commands below assume an activated virtual environment and explicitly call the manage.py script. You can save yourself some typing in the future by replacing these commands with the alias 'dj' for 'manage.py':

```
sudo vim /etc/bash.bashrc
```

Then add the following to the bottom of the file:

```
alias dj="/usr/local/apps/TEKDB/env/bin/python  
/usr/local/apps/TEKDB/TEKDB/manage.py"  
alias djrun="dj runserver 0.0.0.0:8000"
```

12. Run migration

```
python /usr/local/apps/TEKDB/TEKDB/manage.py migrate
```

13. Run static file collection

```
python /usr/local/apps/TEKDB/TEKDB/manage.py collectstatic
```

14. Create Superuser (or import legacy data)

If building a **new database** without data to load into it:

```
python /usr/local/apps/TEKDB/TEKDB/manage.py loaddata  
/usr/local/apps/TEKDB/TEKDB/TEKDB/fixtures/default_users_f  
ixture.json
```

- This will create the default groups:
 - Administrator
 - Editor
 - ReadOnly
- This will also create three default users (with corresponding permissions)
 - admin
 - editor
 - readonly
- Reach out to your administrator for default passwords and edit these as soon as possible

If migrating from a **prior version of the ITKDB/TEKDB tool**, follow the steps here:

[Migrating from TEKDB \(2017\) to ITKDB \(2022\)](#)

If migrating data from **MTKEDB** in, follow steps in this document.

<https://github.com/ECOTRUST/TEKDB/wiki/Migrating-From-Access-to-Postgres>

15. Load Initial Lookup Data

- The database administration forms include numerous dropdown choices. While all of these choices may be populated by hand, it's better to start with them primed with some of the more common choices. To do this, simply run:

```
python /usr/local/apps/TEKDB/TEKDB/manage.py loaddata  
/usr/local/apps/TEKDB/TEKDB/TEKDB/fixtures/default_lookups_fixtur  
e.json
```

16. Update your Database iterators A problem on older installs is that after importing data into the database, the iterators may not be updated (or maybe are overwritten by the data import). Rather than check that every single iterator sequence in the database matches the IDs that exist in their sister tables, we've written a handy script:

- If the database schema has changed since you last generated the sequence updater script, create a new one:

```
cd /usr/local/apps/TEKDB/deployment
sudo -u postgres psql -Atq tekdb -f
    /usr/local/apps/TEKDB/deployment/generate_db_iterator_reset.sql -o
    /usr/local/apps/TEKDB/deployment/reset_db_iterators_YYYYMMDD.sql
```

Be sure to replace YYYYMMDD with today's date and remove any old versions of the script.

- If the 'reset_db_iterators...' script is up to date, run it:

```
sudo -u postgres psql tekdb -f
    /usr/local/apps/TEKDB/deployment/reset_db_iterators_YYYYMMDD.sql
```

17. Test your installation with Django's Dev Server

- If you need to debug and test here:
 - Make sure network traffic is allowed on port 8000
 - add `DEBUG=True` to
/usr/local/apps/TEKDB/TEKDB/TEKDB/local_settings.py
 - run `python /usr/local/apps/TEKDB/TEKDB/manage.py runserver 0.0.0.0:8000`
 - Know the IP address (or URL if you have DNS set up) of your server
 - punch your site's address at port 8000 into a browser, i.e.:
your.domain.com:8000
- **Be sure to either remove your 'DEBUG=' entry into local settings** or set it to False when done.

18. Install and configure NGINX

```
sudo apt-get install nginx -y
sudo cp /usr/local/apps/TEKDB/deployment/tekdb_nginx.conf
    /etc/nginx/sites-available/tekdb
sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/tekdb
    /etc/nginx/sites-enabled/tekdb
sudo cp /usr/local/apps/TEKDB/deployment/uwsgi_params /etc/nginx/
```

19. Configure UWSGI and boot processes:

```
sudo cp /usr/local/apps/TEKDB/deployment/emperor.ini /etc/uwsgi/  
sudo ln -s /usr/local/apps/TEKDB/deployment/uwsgi.service  
/etc/systemd/system/  
sudo ln -s /usr/local/apps/TEKDB/deployment/tekdb.ini  
/etc/uwsgi/apps-enabled/  
sudo service uwsgi start  
sudo service uwsgi restart  
sudo cp /usr/local/apps/TEKDB/deployment/rc.local /etc/rc.local  
sudo cp /usr/local/apps/TEKDB/deployment/rc.local.service  
/etc/systemd/system/rc.local.service  
sudo chmod 744 /etc/rc.local  
sudo systemctl enable rc.local
```

20. Set Media Folder Permissions:

```
sudo groupadd mediausers  
sudo adduser www-data mediausers  
sudo chgrp -R mediausers /usr/local/apps/TEKDB/TEKDB/media  
sudo chmod -R 770 /usr/local/apps/TEKDB/TEKDB/media
```

21. Reboot

```
sudo reboot 0
```

22. Test Server and Networking/Ports

Automatic (Unattended) Security Updates

From the document [Using the "unattended-upgrades" package](#)

Install the unattended-upgrades package if it isn't already installed (sudo apt-get install unattended-upgrades).

To enable it, do:

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

(it's an interactive dialog) which will create /etc/apt/apt.conf.d/20auto-upgrades with the following contents:

- APT::Periodic::Update-Package-Lists "1";
- APT::Periodic::Unattended-Upgrade "1";

To have the server automatically reboot when necessary to install security updates:

1. install the package update-notifier-common

```
sudo apt-get install update-notifier-common
```

2. edit the file /etc/apt/apt.conf.d/50unattended-upgrades near the bottom you will find the line

```
//Unattended-Upgrade::Automatic-Reboot "false";
```

uncomment it and set value to true:

```
Unattended-Upgrade::Automatic-Reboot "true";
```

To tell the server what time is most safe to reboot (when needed), uncomment the line

```
//Unattended-Upgrade::Automatic-Reboot-Time "02:00";
```

And set the time to your desired restart time.

Read the source document for more details.

Debugging

- startup scripts logs: /var/log/rc.local.log
- nginx error log: /var/log/nginx/tekdb.error.log
- nginx access log: /var/log/nginx/tekdb.access.log