**Texas Underground Installation Manual**

caves.tacc.utexas.edu

This manual explains how to install the Texas Underground website from scratch, as well as how to update an existing site for changes to the software or database schema.

# Assumptions

This manual makes the following assumptions when installing the site from scratch:

* You are installing on a brand of Linux. These instructions were tested for Ubuntu.
* NGINX is installed and configured for HTTPS.
* The site is configured at /etc/nginx/sites-available/caves.conf.
* PostgreSQL database is installed and running.
* A ‘caves’ database has been created for PostgreSQL.
* Your account as full permissions to the ‘caves’ database.
* Your account can run as a superuser.
* You are backing up the ‘caves’ database by some means not described here.

# Preparing the Linux Box

SSH in to the Linux box and install Node.js, yarn, and pm2 as follows:

sudo apt install npm

sudo npm install -g n

sudo n stable

hash -r

sudo npm install -g yarn

sudo npm install -g pm2

Add the following to /etc/nginx/sites-available/caves.conf, at the very start of the file, before "server {" (e.g. sudo vim /etc/nginx/sites-available/caves.conf):

upstream app\_yourdomain {

server 127.0.0.1:3000;

keepalive 8;

}

Add the following to /etc/nginx/sites-available/caves.conf before the final bracket of "server { ... }":

root /var/www/html;

location /api {

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-NginX-Proxy true;

proxy\_pass http://127.0.0.1:3000/api;

proxy\_redirect off;

}

location /static {

root /var/www/html;

expires 1d;

add\_header Cache-Control "public, no-transform";

}

location /build {

root /var/www/html;

expires 1h;

add\_header Cache-Control "public, no-transform";

}

location / {

expires 1d;

add\_header Cache-Control "public, no-transform";

try\_files /index.html /dev/null =404;

}

Feel free to change the cache expiration timeouts. The one hour (“1h”) timeout for builds may only be suitable during development.

Run "sudo nginx -t" to check the syntax of caves.conf and correct any errors.

Restart NGINX with the new configuration: sudo /etc/init.d/nginx restart

You can test the setup at this point by going to https://caves.tacc.utexas.edu/api and making sure you get a "502 bad gateway" error. This is the correct error prior to installing the website.

# Configure the Website

Create /var/www/.env based on the following (e.g via "sudo vim /var/www/.env"):

CAVESITE\_BASE\_URL=https://caves.tacc.utexas.edu

CAVESITE\_PORT=3000

CAVESITE\_HIDDEN\_TABS=comma-delimited-names-of-tabs-you-want-to-hide

CAVESITE\_LOG\_SERVER\_RESTART=on

SENDGRID\_API\_KEY=sender-sendgrid-key

MAPBOX\_ACCESS\_TOKEN=your-mapbox-access-token

CAVESITE\_LOG\_DIR=/var/www/logs

CAVESITE\_DB\_HOST=localhost

CAVESITE\_DB\_PORT=5432

CAVESITE\_DB\_NAME=caves

CAVESITE\_DB\_USER=postgres-user-name

CAVESITE\_DB\_PASSWORD=postgres-user-password

You can leave out CAVESITE\_HIDDEN\_TABS or leave its value blank if you don’t want to hide any of the website tabs. If you do want to hide one or more tabs, list them here with only the first letter capitalized.

# Build, Deploy, and Launch the Website

If you have not already installed a git repo for the website, do so as follows:

cd ~

git clone https://github.com/ut-entomology/cavesite.git

cd cavesite

Otherwise, update the existing repo:

cd ~/cavesite

git pull

From within ~/cavesite, type the following to build the website:

yarn

rm -rdf build # this prevents intermittent build problems

yarn build

You have to use yarn and not npm because the version dependencies were established with yarn.

Now it’s time to deploy the site, but if you previously deployed the site, you should take down the live site before doing so. In this case, type the following to shut down the site:

sudo pm2 stop server

Deploy the site as follows. Note that this deployment process tears down all of the ‘caves’ database except for the users, key\_data, and logs tables:

./bin/setup-db

sudo ./bin/deploy

If the database schema did not change, you need only run the second of the above commands.

If this is your first time starting up the site, and if this will be the live deployment, run the following commands. This sets the server up to run in the background and to automatically restart on reboots and crashes:

sudo pm2 start --name server -u www-data --cwd /var/www node build/backend/server.js

sudo pm2 startup

You can instead run the server for testing purposes, and only for the duration of your login:

cd /var/www

sudo -u www-data node build/backend/server.js

You can start, stop, and resume server operation as follows:

sudo pm2 stop server

sudo pm2 start server

sudo pm2 restart server

Note that at this point, the website contains no data.

# Prepare Website for Importing Data

Before you can import data into the website, you must first prepare it for importing. The first step is to create an admin user as follows:

cd /var/www

node build/tools/create-admin.js

You will be asked to create a user and a password. The tool requires that the password be a strong one and may ask you to enter a different password.

Now go to the website and login as this user:

<https://caves.tacc.utexas.edu>

At the top right of the page you’ll see a switch for selecting ‘Data’ or ‘Admin’. Select ‘Admin’.

From the ‘Schedule’ tab, select the days and times at which you’d like to import from GBIF. The import should occur after GBIF has imported from Specify and had time to process the import for publication.

From the ‘Files’ tab, add cave localities, cave obligates, karst regions, Texas species status, and federal species status as required. Save your changes before leaving the website.

The site is now ready to import from CSV or GBIF.

# Importing from CSV

You can import the data from a CSV file instead of GBIF if you want. The format of this CSV file must be that of the report output by the -rW switch of the Python program in the cleancave repo. This CSV file uses column names that can be uploaded to Specify.

To import from such a CSV file, type the following:

cd /var/www

node build/tools/load-csv.js path-to-csv-file.csv

# Importing from GBIF

To manually import the data from GBIF, type the following, and the data will import immediately upon pressing return:

cd /var/www

node build/tools/import-gbif.js –force

To instead schedule regular imports from GBIF..

TBD

# Note on Importing

The import process is designed to work without having to first purge the database, so you can repeatedly import, and upon completion of each import, the newly imported data will replace the prior data. Each import replaces all data in the database except for users, logs, and admin key\_data files (such as cave localities, cave obligates, karst regions, Texas species status, and federal species status).