Administration using python management commands

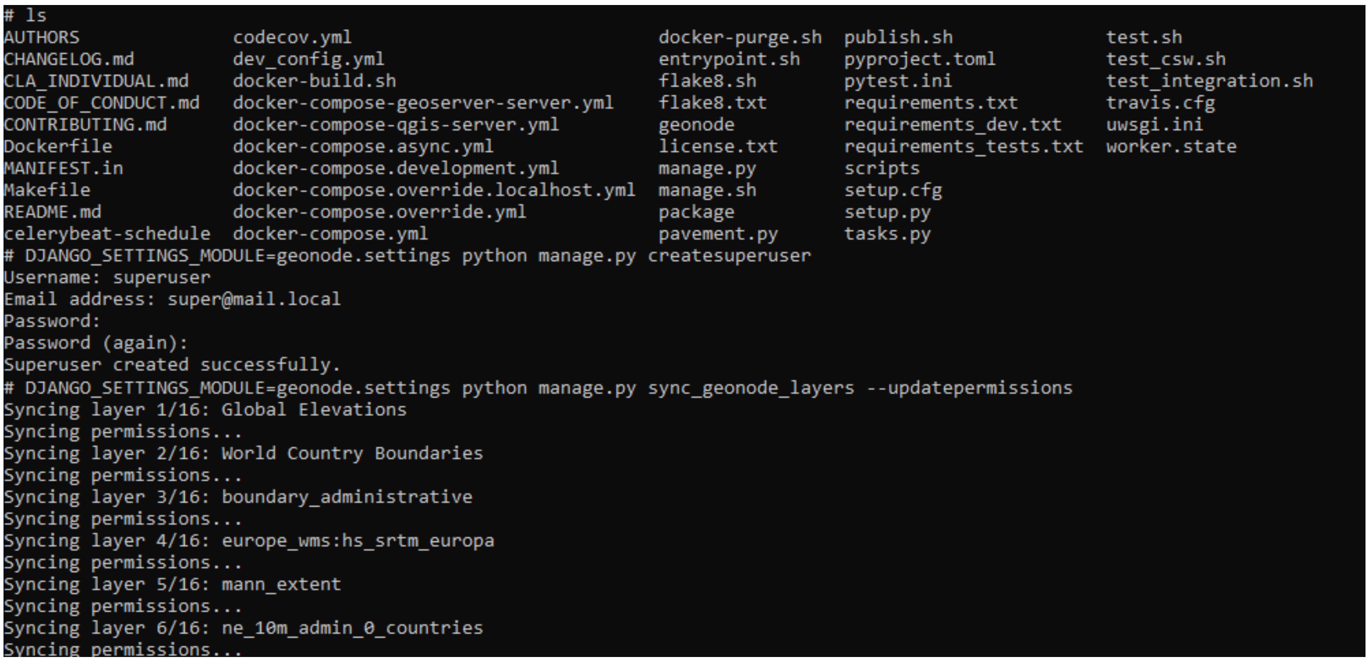
“GeoNode provides additional management operations which are available from the commandline”

GeoNode provides command-line options for doing various administrative tasks in GeoNode. These are provided from the command line interface and utilize the python administration utilities which are only accessible on the host where GeoNode is installed.

These commands can be run by an administrator with programming knowledge and are an extension of the admin interface as they allow users to do additional tasks that are not available in the admin interface. Some of the functions that might be performed using this interface may include:

- Changing advanced options such as disabling self-registration
- Performing backup and restore operations
- Changing the current server URL to fix broken references
- Helping with the automatic ingestion of large raster datasets into GeoNode

In this module, we explore the management interface from the command line.



```
# ls
AUTHORS              codecov.yml                        docker-purge.sh  publish.sh              test.sh
CHANGELOG.md         dev_config.yml                     entrypoint.sh    pyproject.toml          test_csw.sh
CLA_INDIVIDUAL.md    docker-build.sh                    flake8.sh        pytest.ini              test_integration.sh
CODE_OF_CONDUCT.md   docker-compose-geoserver-server.yml flake8.txt      requirements.txt        travis.cfg
CONTRIBUTING.md      docker-compose-qgis-server.yml     geonode          requirements_dev.txt    uwsgi.ini
Dockerfile           docker-compose.async.yml           license.txt      requirements_tests.txt  worker.state
MANIFEST.in          docker-compose.development.yml     manage.py        scripts
Makefile             docker-compose.override.localhost.yml manage.sh     setup.cfg
README.md            docker-compose.override.yml        package          setup.py
celerybeat-schedule  docker-compose.yml                 pavement.py      tasks.py
# DJANGO_SETTINGS_MODULE=geonode.settings python manage.py createsuperuser
Username: superuser
Email address: super@mail.local
Password:
Password (again):
Superuser created successfully.
# DJANGO_SETTINGS_MODULE=geonode.settings python manage.py sync_geonode_layers --updatepermissions
Syncing layer 1/16: Global Elevations
Syncing permissions...
Syncing layer 2/16: World Country Boundaries
Syncing permissions...
Syncing layer 3/16: boundary_administrative
Syncing permissions...
Syncing layer 4/16: europe_wms:hs_srtm_europa
Syncing permissions...
Syncing layer 5/16: mann_extent
Syncing permissions...
Syncing layer 6/16: ne_10m_admin_0_countries
Syncing permissions...
```

You try:

Goal: To explore the some management commands

Docker or Kubernetes

- Navigate to the server where the GeoNode is installed.
- Locate the running **django** container and exec the container.

- Run the following command

```
python manage.py --help
```

- Download the raster using the command line tool like get or curl

```
mkdir raster; wget https://github.com/kartoza/docker-mapserver/blob/master/map/E020N40.tif - O raster/E020N40.tif
```

- Inspect the command to use to ingest the raster into GeoNode

```
python manage.py importlayers -n "East Africa" -v 3 raster
```

- Inspect the layer in GeoNode to see if has been published.

Virtual Env

- Change directory to where the source code for GeoNode is installed
- Activate the virtual env.
- Run the management command as mentioned in the docker section

Check your results

When you are in the Django container you should be able to run a command without getting an error message. Execute python manage.py createsuperuser to ensure that you set up a master account on GeoNode.

Name	Expectation
GeoNode Install method	Rancher, Docker-compose, Ansible, Virtual Env
Docker container lookup	docker ps -a
Django Container Log in (Docker)	docker exec it django bash
Python Management Commands lookup	python manage.py --help

More about management commands

GeoNode is built with Django and Python, with a backend of Geoserver and a PostgreSQL database. The application framework provides an admin interface that allows users to manipulate all the data and other related components relating to permissions for layers, however, a management commandline interface provides a set of management tools that extend the functionality of the administrative interface with additional commands.

Management commands are executed with the python interpreter and modify the Django project configuration settings. Typically, this will be executed using the command formatted similar to `python manage.py <command_name>`, or with the settings object explicitly stated with additional command flags similar to the format of `DJANGO_SETTINGS_MODULE=geonode.settings python manage.py <command_name> --<flags>`.

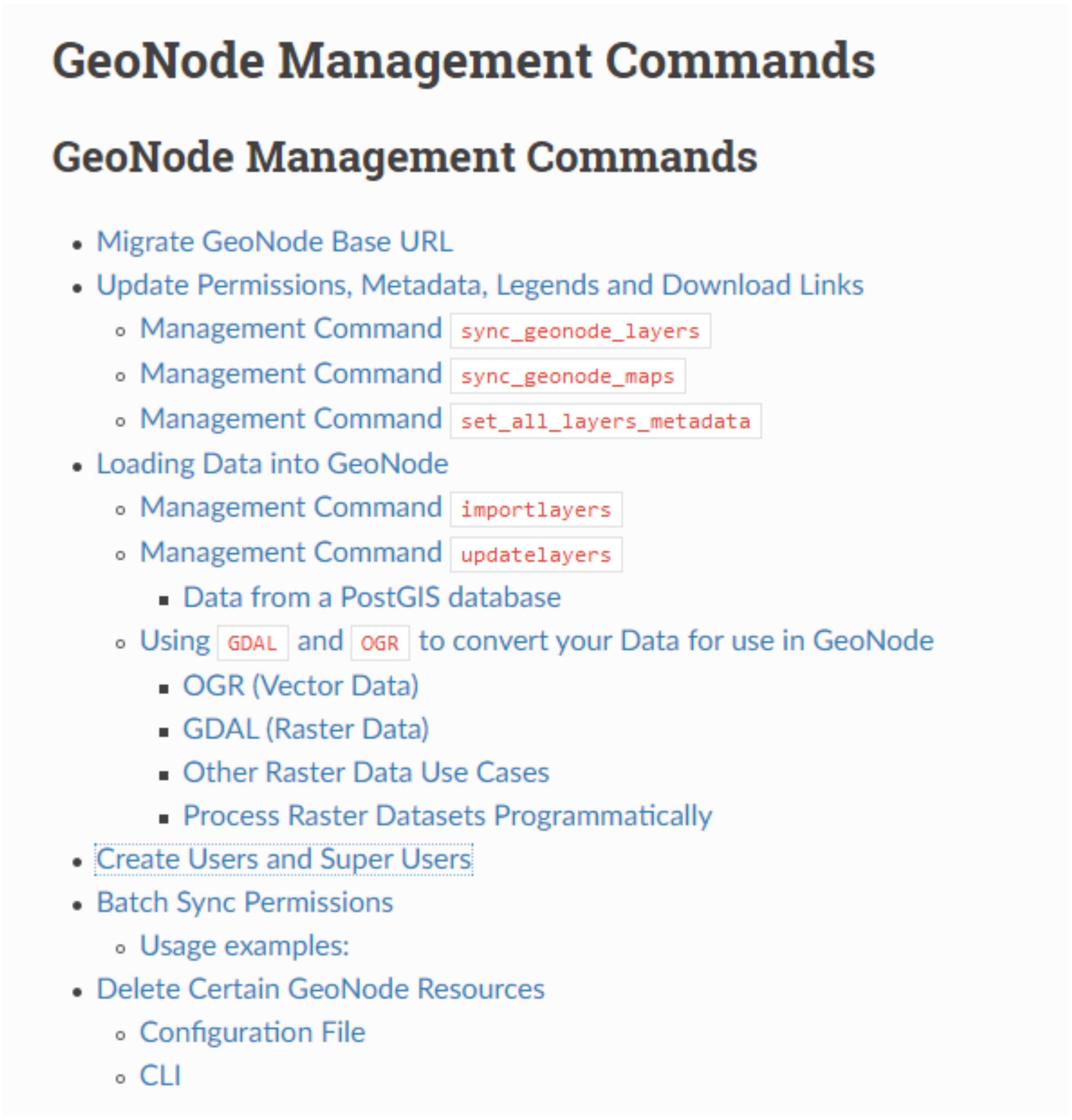
The application supports multiple management commands inclusing:

- createsuperuser
- importlayers
- updatelayers
- sync_geonode_maps
- set_all_layers_metadata

For a full list of all the management commands execute `python manage.py -help`. Each individual command may require additional parameters which may be reviewed e.g. `python manage.py importlayers -help`

The two most common applications of the management commands are typically:

- To upload and publish very large layers, that time out or return other errors when attempted via the web interface
- The publication of layers that are available in the backend but are not yet available from the GeoNode frontend interface



GeoNode Management Commands

GeoNode Management Commands

Check your knowledge:

1. What components make up GeoNode :
    a. *QGIS Desktop, Django and Celery*
    b. *GeoServer, Django, and PostgreSQL*
    c. *Django, python3, and QGIS Desktop*
2. Why would you want to use the command line management commands vs admin interface accessible on the browser:
    a. *It does not make sense, the browser is much more friendly*
    b. *The command-line should not be used because it is complex*
    c. *The command-line allows a user to do more tasks in an automated way which the admin interface will not be able to achieve*
    d. *The admin interface is simpler, easier to use, and has more commands*
3. Can users execue the command line management tools:
    a. *Yes, but only only if you have installed some form of management software like rancher*
    b. *It is impossible, you need server access*

Further reading:

- Management commands documentation https://docs.geonode.org/en/master/admin/index.html#geonode-management-commands