

Deploying Geoserver

Enabling geoserver will add these 3 extractors

- **ncsa.geotiff.preview**, "geotiff preview extractor takes .tif input file to communicate with GeoServer to retrieve WMS metadata"
- **ncsa.geotiff.metadata** "Extractor to perform Geotiff."
- **ncsa.geoshp.preview** "gepshp extractor takes .zip input file to communicate with geoserver to retrieve WMS metadata"

In order to do this, the docker-compose.yml file should be modified in the following way

docker-compose.yml - geoserver

```
# Geoserver application geoserver:  
image: clowder/geoserver:${VERSION:-latest} networks:  
  - clowder  
ports:  
  - 8080:8080  
volumes:  
- geoserver:/data_dir restart: unless-stopped  
# deploy:  
# # # # #  
mode: replicated replicas: 1 placement:  
constraints:  
- node.role == manager  
restart_policy: condition: any
```

The location of the actual geoserver volume on the server is specified in the docker-compose.override.yml file:

docker-compose.override.yml

```
geoserver: driver_opts:  
type: none  
o: bind  
device: /clowder/data/geoserver
```

Geoserver service configuration

The geoserver extractor is in the docker-compose-extractors.yml file.

The default credentials for Geoserver isadmin/geoserver So, once the docker container starts, go to the geoserver service which is running on:[http://hostname.dept.illinois.edu:8080/geoserver/web](http://hostna me.dept.illinois.edu:8080/geoserver/web) and create a new admin user, and reset the password for the default admin account, and then disable the default admin account.

This interface is running over http, not https, I havent figured out how to make it run over https yet, so dont login/change passwords very much to try to minimize how much the password is passed around in clear text.

Security



Settings



Authentication



Passwords



Users, Groups, Roles



Data



Services

Then click the Users/Groups tab, and from here you can create new users and add them to the admin group. Note: You do not need to add the GROUP_ADMIN Role, the default admin does not even have that.

Users, Groups, and Roles

Manage user group and role services

Username	Enabled	Has Attributes
admin	✓	

Groupname	Enabled
clowder_user	✓

Now you can configure the extractor with the user account and password to run as. In the above example I created a "clowder_user" and then I will put that account in the following environment options below:

- GEOSERVER_USERNAME=clowder_user
- GEOSERVER_PASSWORD=SETADIFFERENTPASSWORD! - GEOSERVER_USER=clowder_user

If you ever need to reset the admin password, you can bring down the docker stack, then rename the directory that the clowder/geoserver volume points to, then start the container and then the geoserver service will be set to all the defaults.

Initially I tried to get the extractor to work with a non-admin account, and I don't think it does, so renaming and adding a different password is a good idea.

Note: If the password includes a \$ it will generate an error when you bring up the stack, so if the password has a \$, you will need to escape the \$ with an additional \$. So in the below example the real password is actually SETADIFFERENT\$PASSWORD!

docker-compose-extractors file geoshp

```
# Geoserver_shp extractor ncsa_geo_shp:  
image: clowder/extractors-geoshp-preview:${VERSION:-latest} networks:  
- clowder environment:  
- RABBITMQ_URI=${RABBITMQ_URI:-amqp://guest:guest@rabbitmq/%2F} - RABBITMQ_EXCHANGE=${RABBITMQ_EXCHANGE:-  
clowder}  
- REGISTRATION_ENDPOINTS=${REGISTRATION_ENDPOINTS}  
- PROXY_ON=true  
- PROXY_HOST=https://loan-compute-03.engr.illinois.edu/  
- PROXY_URL=https://loan-compute-03.engr.illinois.edu/api/proxy/ - GEOSERVER_USERNAME=clowder_user  
# Remember to escape $ with a \$  
- GEOSERVER_PASSWORD=SETADIFFERENT\$\$PASSWORD!  
- GEOSERVER_WORKSPACE=  
- GEOSERVER_USER=clowder_user  
- GEOSERVER_URL=http://geoserver:8080/geoserver/  
depends_on:  
    - geoserver  
restart: unless-stopped # deploy:  
## ## ## ##  
mode: replicated replicas: 1 placement:  
constraints:  
- node.role == worker  
restart_policy: condition: any
```