

Documentación GeoNode con Docker

Se siguieron algunos pasos de la siguiente guía:

<https://docs.geonode.org/en/master/install/advanced/project/index.html#docker>

Índice

Documentación GeoNode con Docker.....	1
Índice.....	1
Requisitos para la instalación.....	2
Programas utilizados.....	2
Requisitos mínimos de sistema.....	2
Pasos para Instalar GeoNode.....	3
1. Creación de usuario geonode.....	3
2. Clonar repositorio.....	3
Notas Adicionales.....	3
Instalación Manual de GeoNode.....	4
Creación de usuario geonode.....	4
Instalación de Python y otros software y creación de directorio de trabajo.....	4
Clonación del repositorio geonode-project y creación del entorno virtual de Python.....	5
Instalación y creación de un proyecto de Django.....	5
Creación de archivo de variables de entorno.....	6
Instalación de Docker y permisos.....	6
Compilación de imágenes de Docker y creación de contenedores con docker compose..	7
Volver a acceder por consola.....	7
Ajustes de variables de entorno.....	8
Archivo .json y .env.....	8
Configuraciones adicionales del archivo .env:.....	8
Reinicio.....	8
Personalización.....	9
Cambiar miniatura.....	9
Agregar capa base.....	9
Cambiar previsualización.....	11
Problemas encontrados.....	13
Redirección a localhost/geoserver/j_spring_security_check.....	13
Corregir autenticación GeoServer en la administración de GeoNode.....	13
Modificación de contraseña.....	14
Problemas al cargar capas.....	17
Posible pérdida de personalización de GeoNode.....	20

Requisitos para la instalación

Programas utilizados

Máquina virtual: VirtualBox 7.0.14

Servidor: Ubuntu 22.04.3 Live Server amd64

Requisitos mínimos de sistema

Procesadores: 4 núcleos de 2 GHz o superior. (Es posible que se requiera potencia de procesamiento adicional para múltiples renderizados de estilos simultáneos)

RAM: 8 GB o 16 GB para implementación en producción.

Arquitectura: Se recomienda 64 bits.

Uso de disco: de 30 GB o 50 GB de mínimo (reservado solo para el sistema operativo y el código fuente).

Espacio en el disco: Espacio en disco adicional para cualquier dato alojado con GeoNode, datos almacenados en la base de datos y mosaicos almacenados en caché con GeoWebCache. Para bases de datos, datos espaciales, mosaicos en caché y "espacio temporal" útiles para la administración, un tamaño de referencia decente para implementaciones de GeoNode es entre 50 GB y 100 GB.

Pasos para Instalar GeoNode

1. Creación de usuario geonode

Crear usuario, darle permisos e ingresar con el mismo.

```
sudo adduser geonode
```

```
sudo usermod -aG sudo geonode
```

```
su geonode
```

Luego de haber creado el usuario geonode

2. Clonar repositorio

Se puede clonar este repositorio en "\$HOME" y ejecutar directamente el script dentro de la carpeta del repositorio:

```
sudo chmod +x install_geonode.sh
```

```
sudo ./install_geonode.sh
```

Notas Adicionales

- El proceso de instalación le pedirá al usuario ingresar una contraseña para el usuario "admin" de Geoserver.
- El script guarda un log completo con todas las salidas por consola. Este se encuentra en /var/log y tiene esta estructura:

```
geonode_instalacion_$(date +%Y-%m-%d_%H-%M-%S).log
```

- La instalación completa puede llevar varios minutos.
- Para más información ver el script completo (**install_geonode.sh**) en el repositorio de github.

Instalación Manual de GeoNode

Esta guía sirve como alternativa al script. Describe todos los pasos que se realizaron manualmente para la instalación del software. Para una alternativa más simple, ver el repositorio.

Creación de usuario geonode

Creamos el usuario **geonode** y le asignamos una contraseña:

```
sudo apt update
```

```
sudo adduser geonode
```

```
sudo usermod -aG sudo geonode
```

```
su geonode
```

Instalación de Python y otros software y creación de directorio de trabajo

```
sudo add-apt-repository -y ppa:ubuntugis/ppa
```

```
sudo apt update
```

```
sudo apt install -y python3-gdal gdal-bin libgdal-dev
```

```
sudo apt install -y python3-pip python3-dev python3-virtualenv  
python3-venv virtualenvwrapper
```

```
sudo apt install -y libxml2 libxml2-dev gettext
```

```
sudo apt install -y libxslt1-dev libjpeg-dev libpng-dev libpq-dev  
libmemcached-dev
```

```
sudo apt install -y software-properties-common build-essential
```

```
sudo apt install -y git unzip gcc zlib1g-dev libgeos-dev libproj-dev
```

```
sudo apt install -y sqlite3 spatialite-bin libsqlite3-mod-spatialite
```

```
sudo apt update -y
```

```
sudo apt autoremove -y
sudo apt autoclean -y
sudo apt purge -y
sudo apt clean -y
sudo apt install -y virtualenv virtualenvwrapper
sudo apt install -y vim
sudo mkdir -p /opt/geonode_custom/
sudo usermod -a -G www-data geonode
sudo chown -Rf geonode:www-data /opt/geonode_custom/
sudo chmod -Rf 775 /opt/geonode_custom/
```

Clonación del repositorio geonode-project y creación del entorno virtual de Python

```
cd /opt/geonode_custom/
git clone --branch 4.4.x
https://github.com/GeoNode/geonode-project.git
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
mkvirtualenv --python=/usr/bin/python3 my_geonode
```

Instalación y creación de un proyecto de Django

```
pip install Django==3.2.13
django-admin startproject --template=./geonode-project -e
py,sh,md,rst,json,yml,ini,env,sample,properties -n monitoring-cron -n
Dockerfile my_geonode
cd /opt/geonode_custom/my_geonode
```

Creación de archivo de variables de entorno

Crear el archivo JSON con los valores para las variables de entorno:

```
vim envs-personalizadas.json
```

Dentro del .json se agregan los valores deseados. Ejemplo:

```
{  
  "hostname": "direcciónIp",  
  "geonodepwd": "contraseñaGeoNode",  
  "geoserverpwd": "contraseñaGeoNode",  
  "pgpwd": "contraseñaGeoNode",  
  "dbpwd": "contraseñaGeoNode",  
  "geodbpwd": "contraseñaGeoNode"  
}
```

Reemplazar “direcciónIp” y “contraseñaGeoNode” por los datos necesarios.

Luego creamos el archivo .env:

```
python create-envfile.py -f envs-personalizadas.json
```

Instalación de Docker y permisos

```
curl -fsSL https://get.docker.com -o install-docker.sh
```

```
sudo sh install-docker.sh
```

```
sudo usermod -aG docker geonode
```

```
exit
```

```
su geonode
```

```
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
```

```
mkvirtualenv --python=/usr/bin/python3 my_geonode
```

```
cd /opt/geonode_custom/my_geonode
```

Compilación de imágenes de Docker y creación de contenedores con docker compose

```
sudo apt install docker-compose
```

```
docker compose -f docker-compose.yml build --no-cache
```

```
docker compose -f docker-compose.yml up -d
```

NOTA: Este proceso puede tardar varios minutos.

Con estos pasos, la configuración básica del GeoNode está terminada.

Volver a acceder por consola

Antes de realizar cualquier acción debemos acceder con el usuario geonode que previamente hemos creado:

```
su geonode
```

```
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
```

```
mkvirtualenv --python=/usr/bin/python3 my_geonode
```

Otra opción:

Se puede ingresar directamente con el usuario geonode previamente creado

```
ssh geonode@direccionip
```

Luego, debemos posicionarnos en la carpeta correcta:

```
cd /opt/geonode_custom/my_geonode
```

Ajustes de variables de entorno

Archivo .json y .env

Una opción para modificar el archivo .env es primero crear un nuevo .json con variables que serán inyectadas al .env.

```
vim envs-personalizadas.json
```

Luego se debe construir el nuevo archivo .env basado en .json anteriormente creado.

```
python create-envfile.py -f envs-personalizadas.json
```

Configuraciones adicionales del archivo .env:

De ser necesario, se debe agregar la dirección ip correspondiente, más el número de puerto utilizado en todas las líneas necesarias. más las siguientes variables como aparecen a continuación:

Importante: Reemplazar **direcciónIp** por la dirección ip que corresponda.

```
NGINX_BASE_URL=http://direcciónIp/
```

```
ALLOWED_HOSTS=["django", '*', 'direcciónIp']"
```

```
HTTP_HOST=direcciónIp
```

```
HTTPS_HOST=
```

```
HTTP_PORT=80
```

```
HTTPS_PORT=
```

Reinicio

Luego de tener el archivo .env configurado ejecutar docker compose una vez más:

```
docker compose -f docker-compose.yml build --no-cache
```

```
docker compose down && docker compose up -d
```


Personalización

Con esta guía podremos reemplazar el mapa base de OpenStreetMap con la versión Argenmap. Además, se cambiará las miniaturas y las previsualizaciones por su equivalente del IGN.

Cambiar miniatura

Listamos los contenedores

```
docker ps
```

Entrar al contenedor de django

```
docker exec -ti django4my_geonode bash
```

Y ejecutar para descargar la miniatura

```
wget
```

```
https://wms.ign.gov.ar/geoserver/gwc/service/tms/1.0.0/capabaseargenmap@EPSG%3A3857@png/0/0/0.png -O  
/mnt/volumes/statics/static/mapstorestyle/img/argenmap.png
```

Agregar capa base

Luego editar el archivo settings.py

```
cd /usr/local/lib/python3.10/dist-packages/geonode/
```

```
apt upgrade
```

```
apt install vim
```

```
vim settings.py
```

Buscar el array DEFAULT_MS2_BACKGROUNDS y agregar el bloque del mapa base nuevo:

```
{  
    "type": "wms",  
    "format": "image/png",  
    "title": "Argenmap",  
    "name": "capabaseargenmap",  
    "id": "capabaseargenmap",
```

```

        "url":
"https://wms.ign.gob.ar/geoserver/ows?service=WMS&version=1.3.0&request=GetCa
pabilities",
        "group": "background",
        "thumbURL": f"{SITEURL}static/mapstorestyle/img/argenmap.png",
        "visibility": True
    },

```

Importante: observar que la indentación sea correcta y hecha con espacios y que el bloque esté abierto y cerrado por llaves con las comas al final o principio dependiendo de en qué parte del array se encuentre.

Luego cambiar el parámetro “visibility” a False para el mapa base por defecto anterior, qué es OpenStreetMap. Ejemplo:

```

DEFAULT_MS2_BACKGROUNDS = [
    {
        "type": "wms",
        "format": "image/png",
        "title": "Argenmap",
        "name": "capabaseargenmap",
        "id": "capabaseargenmap",
        "url": "https://wms.ign.gob.ar/geoserver/ows?service=WMS&version=1.3.0&request=GetCapabilities",
        "group": "background",
        "thumbURL": f"{SITEURL}static/mapstorestyle/img/argenmap.png",
        "visibility": True
    },
    {
        "type": "osm",
        "title": "Open Street Map",
        "name": "mapnik",
        "source": "osm",
        "group": "background",
        "visibility": False,
    },
    {
        "type": "tileprovider",

```

Guardar el cambio, salir del contenedor y reiniciar todos usando docker compose y su archivo Yaml.

docker compose restart

El reinicio de los servicios puede demorar varios minutos, para verificar se puede leer el log de Django con **docker compose logs -f django** hasta que aparezca lo siguiente:

```
django4my_geonode | 0 static files copied to '/mnt/volumes/statics/static', 3858 unmodified.
django4my_geonode | statics tasks done
django4my_geonode | *****geoserver*****
django4my_geonode | GeoServer is available for HTTP calls!
django4my_geonode | waitforgeoserver tasks done
django4my_geonode | *****geoserver fixture*****
django4my_geonode | Setting GeoServer Admin Password..
django4my_geonode | Response Code: 200
django4my_geonode | GeoServer admin password updated SUCCESSFULLY!
django4my_geonode | geoserverfixture tasks done
django4my_geonode | Executing UWSGI server uwsgi --ini /usr/src/my_geonode/uwsgi.ini for Production
django4my_geonode | -----
django4my_geonode | FINISHED DJANGO ENTRYPOINT -----
django4my_geonode | -----
django4my_geonode | got command uwsgi --ini /usr/src/my_geonode/uwsgi.ini
django4my_geonode | [uWSGI] getting INI configuration from /usr/src/my_geonode/uwsgi.ini

```

Después de estos cambios se va a observar el nuevo mapa base al abrir los diferentes recursos pero resta modificar el mapa base de las previsualizaciones.

Cambiar previsualización

Buscar en settings.py la variable THUMBNAIL_BACKGROUND y cambiarlo de el estado por defecto:

```
THUMBNAIL_BACKGROUND = {
    # class generating thumbnail's background
    # 'class': 'geonode.thumbs.background.WikiMediaTileBackground',
    "class": "geonode.thumbs.background.OSMTileBackground",
    # 'class': 'geonode.thumbs.background.GenericXYZBackground',
    # initialization parameters for generator instance, valid only for generic classes
    "options": {
        # 'url': URL for the generic xyz / tms service
        # 'tms': False by default. Set to True if the service is TMS
        # 'tile_size': tile size for the generic xyz service, default is 256
    },
    # example options for a TMS service
    # 'class': 'geonode.thumbs.background.GenericXYZBackground',
    # 'options': {
    #     'url': 'http://maps.geosolutionsgroup.com/geoserver/gwc/service/tms/1.0.0/osm%3Aosm_simple_light@EPSG%3A900913/png/{z}/{x}/{y}.png',
    #     'tms': True
    # },
}
```

Comentar "'class' : "geonode.thumbs.background.OSMTileBackground";" y descomentar "class": 'geonode.thumbs.background.GenericXYZBackground',

Luego, en **options** se debe modificar las variables. A continuación un ejemplo:

```
THUMBNAI_BACKGROUND = {
  # class generating thumbnail's background
  # 'class': 'geonode.thumbs.background.WikiMediaTileBackground',
  # "class": "geonode.thumbs.background.OSMTileBackground",
  "class": "geonode.thumbs.background.GenericXYZBackground",
  # initialization parameters for generator instance, valid only for generic classes
  "options": {
    'url': 'http://wms.ign.gob.ar/geoserver/gwc/service/tms/1.0.0/capabaseargenmap@EPSG%3A3857@png/{z}/{x}/{y}.png',
    'tms': True,
    'tile_size': 256
  },
  # example options for a TMS service
  # 'class': 'geonode.thumbs.background.GenericXYZBackground',
  # 'options': {
  #   'url': 'http://maps.geosolutionsgroup.com/geoserver/gwc/service/tms/1.0.0/osm%3Aosm_simple_light@EPSG%3A900913@png/{z}/{x}/{y}.png',
  #   'tms': True
  # }
```

Al finalizar los cambios, debemos reiniciar el contenedor.

docker compose restart

Problemas encontrados

Redirección a localhost/geoserver/j_spring_security_check

En primer lugar se debe corroborar de tener el archivo .env bien configurado. Una vez que este sea corregido es posible que se necesite eliminar los volúmenes y reiniciar el docker compose para que se apliquen los cambios.

```
docker compose down
```

Eliminación de volúmenes

```
docker volume prune -a -f
```

```
docker compose up -d
```

Esto debería solucionar el redireccionamiento

Nota: En caso de también necesitar eliminar las imágenes se debe hacer un build nuevo para volver a crearlas desde cero.

```
docker compose down
```

Esto elimina imágenes y volúmenes

```
docker system prune -a -f -volumes
```

```
docker compose -f docker-compose.yml build --no-cache
```

```
docker compose up -d
```

Corregir autenticación GeoServer en la administración de GeoNode

Ingresar a la administración de Geonode y luego en la sección: Inicio > Django OAuth Toolkit > Applications > GeoServer

Modificar Redirect uris y reemplazar localhost por la IP o nombre de host del servidor.

Valor por defecto = http://localhost/geoserver/index.html

Corregido:

Client id	ZCkY8ulon3hS12w
User	1000 <input type="button" value="Q"/> admin
Redirect uris	http://172.20.205.53/geoserver/index.html Allowed URIs list, space separated
Client type	<input checked="" type="radio"/> Confidential <input type="radio"/> Publico
Authorization grant type	<input checked="" type="radio"/> Authorization code

Modificación de contraseña

Solución basada en esta guía:

https://docs.google.com/document/d/1g_sD8NCzKRfIkrbXZp0ggHmELG_B9hz8xCqKHaK-O8/e/dit

Debemos modificar varios parámetros en Geoserver:

Proxy Base URL

Global > URL base del proxy = <http://direcciónIp:8080/geoserver>

Global Settings
Settings that apply to all OGC services and control the internal

Settings

- Global
- Image Processing
- Raster Access

OGC Services
Service Settings

Proxy Base URL

Enable global services

→ **Service Request Settings**

Base Server URL

Users, Groups, and Roles > geonode REST role service > Base Server URL = <http://direcciónIp>

The screenshot displays the Geonode administration interface. On the left, the 'Security' menu is visible with options: Settings, Authentication, Passwords, and Users, Groups, Roles. Below the menu is a dropdown menu showing 'ROLE_ADMIN'. The main content area shows a table of REST role services with two entries: 'default' and 'geonode REST role service'. The 'geonode REST role service' entry is selected. Below the table, the 'REST Role Service Settings' are shown for the selected service. The 'Base Server URL' field contains 'http://192.168.1.108' and the 'Roles REST Endpoint' field contains '/api/roles'.

Security

- Settings
- Authentication
- Passwords
- Users, Groups, Roles

ROLE_ADMIN

REST Role Service Settings

Base Server URL

Roles REST Endpoint

REST Role Service Settings

→ Add new
 Remove selected

<input type="checkbox"/>	Name
<input type="checkbox"/>	default
<input type="checkbox"/>	<u>geonode REST role service</u>

← << < 1 > >> → Results 1 to 2 (out of 2 items) →

Authentication URL

Authentication > geonode-oauth2 > Modificar todas las ip por la nuestra: http://**direcciónIp**

Name	<input type="text" value="geonode-oauth2"/>
OAuth2 provider connection	
Enable Redirect Authentication EntryPoint	<input type="checkbox"/>
Login Authentication EndPoint	<input type="text" value="/j_spring_oauth2_geonode_login"/>
Logout Authentication EndPoint	<input type="text" value="/j_spring_oauth2_geonode_logout"/>
Force Access Token URI HTTPS Secured Protocol	<input type="checkbox"/>
Access Token URI	<input type="text" value="http://192.168.1.108/o/token/"/>
Force User Authorization URI HTTPS Secured Protocol	<input type="checkbox"/>
User Authorization URI	<input type="text" value="http://192.168.1.108/o/authorize/"/>
Redirect URI	<input type="text" value="http://192.168.1.108/geoserver"/>
Check Token Endpoint URL	<input type="text" value="http://192.168.1.108/api/o/v4/tokeninfo/"/>
Logout URI	<input type="text" value="http://192.168.1.108/account/logout/"/>
Scopes	<input type="text" value="write"/>
Client ID	<input type="text" value="Jrchz2oPY3akmzndmgUTYrs9gczlgoV20YPSvqaV"/>
Client Secret	<input type="text" value="rCnp5txobUo83EpQEblM8fvj3QT5zb5qRfxNsuPzCqZai"/>
Role source	<input type="text" value="Role service"/> <input type="text" value="geonode REST role service"/>

Admin Password

Finalmente cambiamos la contraseña del admin por la nuestra:

Edit user

You can update the password, enable/disable the user or cl

▼ default

- ➕ Add new user
- ➖ Remove Selected
- ➖ Remove Selected and remove role associations

<input type="checkbox"/>	Username
<input type="checkbox"/>	admin

<< < 1 > >> Results 1 to 1 (out of 1 items)

User name
admin

Enabled

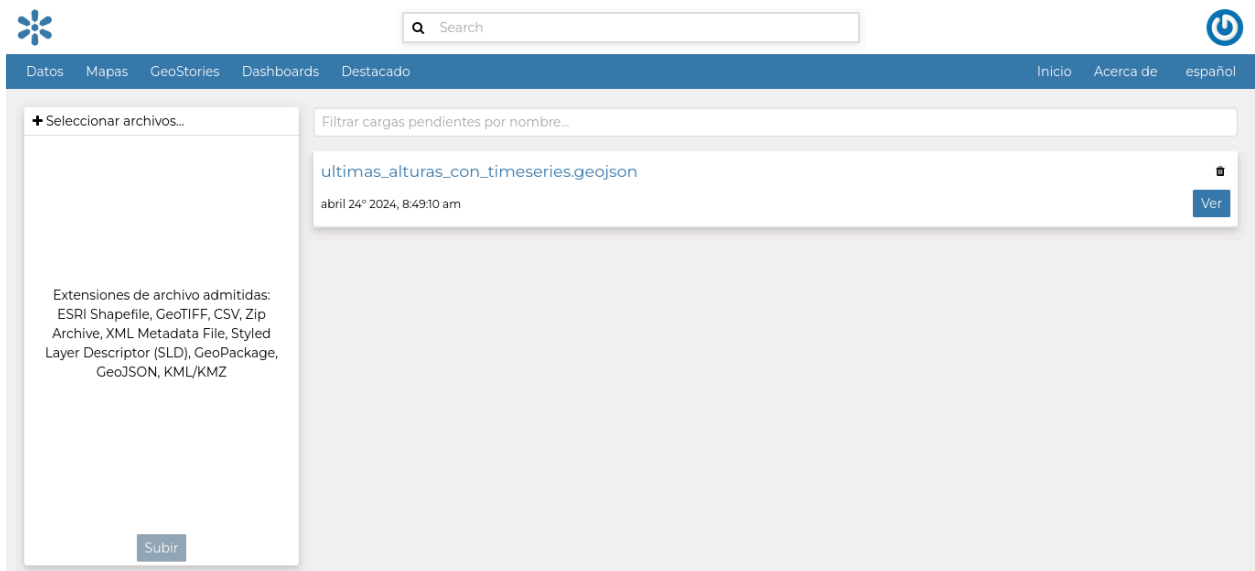
Password
[.....]

Confirm password
[.....]

Problemas al cargar capas

Solución basada en esta guía:

https://docs.google.com/document/d/1-fcEGyB1usF4VQSdPD6OB-HF9pjpgw4c-n1lgNFHr_4U/edit



Vista de la capa

The screenshot shows a GeoNode interface for a layer titled "ultimas_alturas_con_timeseries". The main map area displays a map of Chile with red dots representing data points. A legend is visible on the left. The right sidebar contains the following information:

Info

Título	ultimas_alturas_con_timeseries
Propietario	admin
Publicación	2024-04-24 08:49
Creado	2024-04-24 08:49
Última modificación	2024-04-24 08:49
Tipo de recurso	dataset
Punto de contacto	admin
Idioma	eng
Información Suplementaria	No information provided

[Ver metadatos completos](#)

© OpenStreetMap contributors.

geonode.org Developers About

The screenshot shows a GeoNode search results page. The top navigation bar includes "Datos", "Mapas", "GeoStories", "Dashboards", and "Destacado". The search bar contains "Search". The results section shows "2 Recursos encontrados". Two resource cards are displayed:

- ultimas_alturas_con_timeseries**: A map showing red dots on a coastal area. Owner: admin. [Ver](#)
- polygon_1**: A map showing a grey polygon. Owner: admin. [Ver](#)

geonode.org Developers About

Esto indica que la capa no puede ser accedida por WMS, al intentar verla directamente desde el request GetMap se observa el siguiente error:

HTTP Status 500 – Internal Server Error

Type Exception Report

Description: The server encountered an unexpected condition that prevented it from fulfilling the request.

Exception

```
java.lang.NullPointerException
    org.geoserver.security.GeoServerRestRoleService.getRoleByName(GeoServerRestRoleService.java:351)
    org.geoserver.security.GeoServerRestRoleService.getAdminRole(GeoServerRestRoleService.java:453)
    org.geoserver.security.impl.RoleCalculator.addMappedSystemRoles(RoleCalculator.java:114)
    org.geoserver.security.impl.RoleCalculator.calculateRoles(RoleCalculator.java:106)
    org.geoserver.security.impl.RoleCalculator.calculateRoles(RoleCalculator.java:67)
    org.geoserver.security.filter.GeoServerPreAuthenticatedUserNameFilter.getRolesFromRoleService(GeoServerPreAuthenticatedUserNameFilter.java:184)
    org.geoserver.security.filter.GeoServerPreAuthenticatedUserNameFilter.getRoles(GeoServerPreAuthenticatedUserNameFilter.java:150)
    org.geoserver.security.oauth2.GeoServerOAuthAuthenticationFilter.doAuthenticate(GeoServerOAuthAuthenticationFilter.java:320)
    org.geoserver.security.oauth2.GeoServerOAuthAuthenticationFilter.doFilter(GeoServerOAuthAuthenticationFilter.java:150)
    org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:346)
    org.geoserver.security.filter.GeoServerCompositeFilter$NestedFilterChain.doFilter(GeoServerCompositeFilter.java:71)
    org.springframework.security.web.authentication.www.BasicAuthenticationFilter.doFilterInternal(BasicAuthenticationFilter.java:166)
    org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:117)
    org.geoserver.security.filter.GeoServerCompositeFilter$NestedFilterChain.doFilter(GeoServerCompositeFilter.java:75)
    org.geoserver.security.filter.GeoServerCompositeFilter.doFilter(GeoServerCompositeFilter.java:92)
    org.geoserver.security.filter.GeoServerBasicAuthenticationFilter.doFilter(GeoServerBasicAuthenticationFilter.java:81)
    org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:346)
    org.geoserver.security.filter.GeoServerCompositeFilter$NestedFilterChain.doFilter(GeoServerCompositeFilter.java:71)
    org.geoserver.security.filter.GeoServerSecurityContextPersistenceFilter$1.doFilterInternal(GeoServerSecurityContextPersistenceFilter.java:72)
    org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:117)
    org.geoserver.security.filter.GeoServerCompositeFilter$NestedFilterChain.doFilter(GeoServerCompositeFilter.java:75)
    org.geoserver.security.filter.GeoServerCompositeFilter.doFilter(GeoServerCompositeFilter.java:92)
    org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:346)
    org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:221)
    org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:186)
    org.geoserver.security.GeoServerSecurityFilterChainProxy.doFilter(GeoServerSecurityFilterChainProxy.java:142)
    org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:354)
    org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:267)
    org.geoserver.filters.XFrameOptionsFilter.doFilter(XFrameOptionsFilter.java:77)
    org.geoserver.filters.GZIPFilter.doFilter(GZIPFilter.java:48)
    org.geoserver.filters.SessionDebugFilter.doFilter(SessionDebugFilter.java:49)
    org.geoserver.filters.FlushSafeFilter.doFilter(FlushSafeFilter.java:42)
    org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:201)
    org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:117)
```

Note: The full stack trace of the root cause is available in the server logs.

Apache Tomcat/9.0.65

Después de los cambios la capa se observa pero sin iniciar sesión

Al loguearse en Geonode, el request de autenticación hacia GeoServer

http://172.16.0.13/proxy/?url=http://172.16.0.13:8080/geoserver/pdf/info.json?access_token=VkxWuy6STOWSUz4JhshaMYkMx1o6ep

Devuelve el siguiente error en el encabezado

HTTP 500

HTTPConnectionPool(host='172.16.0.13', port=8080): Max retries exceeded with url: /geoserver/pdf/info.json?access_token=VkxWuy6STOWSUz4JhshaMYkMx1o6ep (Caused by ResponseError('too many 500 error responses'))

Después de los cambios realizados se eliminaron los contenedores y se volvieron a crear, borrando previamente el valor de la variable de entorno OAUTH2_API_KEY en el archivo “.env”

Resultado exitoso con la sesión iniciada:

