

GEOOs – Componentes e Instalación

Componentes

La base de GEOOs son tres componentes. Uno encargado de la presentación de los datos y la interacción con los usuarios (Portal), otro especializado en datos raster (GeoServer, matrices georreferenciadas de variables) y el tercero (ZRepo) especializado en DataWarehousing, BI y repositorio para herramientas (externas) de analytics.

Los usuarios que ingresan a GEOOs desde la web, utilizan directamente el componente Portal, pudiendo utilizar las funcionalidades allí establecidas. Los usuarios especializados o sistemas que acceden a los datos mediante las APIs REST que se proveen, están accediendo a alguno de los otros componentes. Cada componente se explica en su propia documentación.

Además de los tres componentes anteriores (Portal, GeoServer y ZRepo), que conforman el GEOOs-Core, existen otros componentes de soporte, que se encargan de recopilar información externa e inyectarla dentro de GeoServer o ZRepo. Dentro de estos otros componentes se encuentran NOAA-GFS4, RIE (estaciones meteorológicas y oceanográficas), Copernicus, NOAA-WW3.

Los componentes finales que se deban instalar dependerán de las necesidades específicas que se tengan. Por ejemplo, es posible implementar una instalación de GEOOs para proveer los resultados de un modelo (como archivos NetCDF) hacia un Portal GEOOs ya existente en otro servidor. Para este caso, bastaría con instalar el módulo GeoServer y algún software que automatizara la generación de los archivos de salida de los modelos y los dejara en una carpeta de entrada del GeoServer.

Si se desea proveer de un portal propio (una URL de acceso con dominio propio) se debe instalar el componente Portal.

Instalación de los Componentes de GEOOs

El sistema GEOOs se basa en una arquitectura de microservicios, los que se comunican entre sí utilizando APIs de tipo REST. (HTTP / JSON). La comunicación entre estos microservicios puede ser local o remota.

Los componentes de GEOOs-Core (Portal, GeoServer y ZRepo) están desarrollados en NodeJS 14. Se recomienda utilizar las imágenes docker disponibles para tener rápidamente funcionando una instalación personalizada. En particular, si se hará una instalación en servidores locales o de prueba, se recomienda usar docker-swarm como orquestador de los servicios. Si se desea hacer una instalación en la nube, será necesario crear imágenes docker

a medida (basadas en las ya publicadas) que incluyan los archivos de configuración para cada caso.

Para ambientes de prueba, es posible instalar GEOOs-Core en cualquier sistema operativo (Windows 10, mac OSX y Linux) sin embargo, para ambientes de producción, se recomienda el uso de Linux, ya que la ejecución de los contenedores docker no requiere una capa de virtualización, mejorando el rendimiento y el consumo de recursos.

NOTA: En windows y Mac Docker posee interfaz gráfica (Docker Desktop), en Linux se maneja solo mediante consola.

Instalación de docker

Como antes se mencionó, se recomienda por simpleza y facilidad de configuración, la instalación de GEOOs-Core como un stack de servicios de docker swarm.

Un stack de servicios corresponde a un grupo de micro-servicios que se ejecutan dentro de una red privada (por software) que se comunican entre ellos, y que exponen algunos puertos al exterior (a la red de la máquina host).

Los detalles de la instalación de docker dependerán del sistema operativo de la máquina host que se utilice. Las instrucciones detalladas se pueden obtener en:

<https://docs.docker.com/engine/install/>

Como se utilizará docker engine con el orquestador swarm, es necesario primero iniciar el swarm, mediante el comando:

docker swarm init

Para configurar el stack de servicios que se usará en cada caso, se recomienda usar un archivo **“.yml”** con la especificación de los elementos dentro del stack.

Los contenedores docker corresponden a instancias de las imágenes en ejecución dentro de una máquina. Los contenedores ejecutando las imágenes del GEOOs-Core, requieren acceso a archivos de configuración y un directorio para dejar archivos de **“log”** con los resultados de sus ejecuciones. Si se utiliza docker swarm como orquestador de los micro-servicios, entonces es posible utilizar volúmenes de tipo **“bind”** para compartir un directorio de la máquina host dentro de un directorio del contenedor docker.

Los volúmenes del tipo **“bind”** permiten dejar los archivos de configuración de los componentes de GEOOs en un directorio modificable y los logs de ejecución dentro de otro directorio que se puede consultar también desde la máquina host.

Para utilizar kubernetes como orquestador en la nube, se recomienda crear imágenes nuevas de los servicios que incluyan sus archivos de configuración (no serán modificables “en caliente”). Los logs se pueden desactivar o redireccionar a un sistema de archivos montado en el punto correspondiente.

A continuación se muestran una serie de ejemplos de posibles instalaciones de GEOOs, en donde se va aumentando la complejidad en cada uno de ellos, agregando nuevos servicios al stack.

Instalación del Portal Vacío

El componente Portal de GEOOs utiliza un servidor de base de datos MongoDB como proveedor de las funcionalidades de persistencia de usuarios, sus sesiones, links, etc. Una instalación mínima del Portal requiere además instalar un servidor MongoDB.

El servidor MongoDB se levantará como un contenedor desde la imagen mongo:4.4.1-bionic. Este contenedor se debe configurar con el nombre de usuario / contraseña del usuario MongoDB administrador y un volumen para los datos persistentes de las bases de datos.

Para este ejemplo y los que se mostrarán a continuación se asume que se creará un directorio **/opt/geoos** en la máquina (Linux, en el ejemplo, pero aplicable a Windows 10 y mac OSX). En este directorio se trabajará sobre un archivo llamado “**geoos.yml**” con la especificación de los servicios que conformarán el stack llamado “**geoos**”.

Para instalar una primera versión de un Portal GEOOs vacío, completamos los siguientes pasos:

NOTA: Los siguientes comandos se deben realizar bajo administrados **ROOT**.

- Crear un directorio config bajo /opt/geoos:
sudo mkdir /opt/geoos
- Se ingresa a la carpeta:
cd /opt/geoos
- Luego crear el archivo **config/portal.hjson** con el siguiente contenido:

```
{
  webServer:{http:{port:8090}}

  #Available base Maps
  maps:[{
    code:"esri-world-physical", name:"Esri - World Physical Map"
```

```

url:'https://server.arcgisonline.com/ArcGIS/rest/services/World_Physical_Map/MapServer/tile/{z}/{y}/{x}'
  options:{
    attribution: 'Tiles &copy; Esri &mdash; Source: US National Park Service'
    maxZoom: 8
  }
}, {
  code:"open-topo", name:"OSM - Open Topo"
  url:'https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png'
  options:{
    maxZoom: 17
    attribution:
      href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
      href="http://viewfinderpanoramas.org">SRTM</a>
      href="https://opentopomap.org">OpenTopoMap</a>
      href="https://creativecommons.org/licenses/by-sa/3.0/">CC-BY-SA</a>'
      data:
      style:
      contributors,
      Map
  }
  includeLabels:true
}, {
  code:"esri-world-imagery", name:"Esri - World Imagery"

url:'https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}'
  options:{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community'
  }
}, {
  code:"stamen-terrain", name:"Stamen - Terrain"
  url:'https://stamen-tiles-{s}.a.ssl.fastly.net/terrain/{z}/{x}/{y}{r}.{ext}'
  options:{
    attribution: 'Map tiles by <a href="http://stamen.com">Stamen Design</a>, <a href="http://creativecommons.org/licenses/by/3.0">CC BY 3.0</a> &mdash; Map data &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
    subdomains: 'abcd',
    minZoom: 0,
    maxZoom: 18,
    ext: 'png'
  }
  includeLabels:true
}, {
  code:"esri-world-street", name:"Esri - World Street Map",

url:'https://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer/tile/{z}/{y}/{x}',
  options:{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, DeLorme, NAVTEQ, USGS, Intermap, iPC, NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), TomTom, 2012'
  }
  includeLabels:true
}, {
  code:"esri-world-terrain", name:"Esri - World Terrain",

url:'https://server.arcgisonline.com/ArcGIS/rest/services/World_Terrain_Base/MapServer/tile/{z}/{y}/{x}',
  options:{
    attribution: 'Tiles &copy; Esri &mdash; Source: USGS, Esri, TANA, DeLorme, and NPS',
    maxZoom:13
  }
}, {
  code:"esri-ocean-basemap", name:"Esri - Ocean Base Map",

url:'https://server.arcgisonline.com/ArcGIS/rest/services/Ocean_Basemap/MapServer/tile/{z}/{y}/{x}',
  options:{

```

```

        attribution: 'Tiles &copy; Esri &mdash; Sources: GEBCO, NOAA, CHS, OSU, UNH,
CSUMB, National Geographic, DeLorme, NAVTEQ, and Esri',
        maxZoom:13
    }
    includeLabels:true
}, {
    code:"esri-natgeo", name:"Esri - NatGeo World Map",
url:'https://server.arcgisonline.com/ArcGIS/rest/services/NatGeo_World_Map/MapServer/tile/{
z}/{y}/{x}',
    options:{
        attribution: 'Tiles &copy; Esri &mdash; National Geographic, Esri, DeLorme,
NAVTEQ, UNEP-WCMC, USGS, NASA, ESA, METI, NRCAN, GEBCO, NOAA, iPC',
        maxZoom:16
    }
    includeLabels:true
}]

plugins:["base-plugin"]

# Here
hereAPIKey:"eG2NCop0Fa_k1JUQUaiZvkt-pQKlSLY0DS_UwKqaD6c"

# Servers
geoServers:[]
zRepoServers:[]

# Variables Clasification
varSubjects:[{
    code:"meteo", name:"Metereología"
}, {
    code:"oce", name:"Oceanografía"
}, {
    code:"geo", name:"Geología"
}, {
    code:"geopolitica", name:"División Geopolítica"
}, {
    code:"senso", name:"Datos desde Senso"
}]

varTypes:[{
    code:"sat", name:"Telemetría Satelital"
}]

varRegions:[{
    code:"centro", name:"Zona Centro"
}]
}

```

- Crear el directorio “**mongo-data**” bajo **/opt/geoos**
- Crear el directorio “**log**” bajo **/opt/geoos**
- Si bien las imágenes necesarias para los servicios se descargan automáticamente al instalar (deploy) el stack de Swarm, para poder monitorear el avance de la descarga de esas imágenes, es posible descargarlas antes usando el comando “pull” de docker. Para el Portal vacío se requieren las imágenes de MongoDB y del servicio del Portal. Éstas se descargan usando los comandos:
 - **docker pull mongo:4.4.1-bionic**
 - **docker pull docker.homejota.net/geoos/portal**

- Crear el archivo **geoos.yml** en **/opt/geoos** con el siguiente contenido:

```
version: '3.6'
services:
  db:
    image: mongo:4.4.1-bionic
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin-user
      MONGO_INITDB_ROOT_PASSWORD: admin-password
    volumes:
      - ./mongo-data:/data/db
    ports:
      - 27017:27017

  portal:
    image: docker.homejota.net/geoos/portal
    deploy:
      restart_policy:
        condition: on-failure
    ports:
      - 8091:8090/tcp
    environment:
      MONGO_URL: "mongodb://admin-user:admin-password@db:27017"
      MONGO_DATABASE: geoos
    volumes:
      - .config:/home/config
```

- Instalar (deploy) el stack de servicios “geoos” usando el comando:

docker stack deploy -c geoos.yml geoos

```
root > docker stack deploy -c geoos.yml geoos
Creating network geoos_default
Creating service geoos_portal
Creating service geoos_db
root >
```

Luego de un momento (depende de las capacidades de la máquina host) los servicios deberían estar levantados y escuchando por sus puertos. Esto se puede verificar usando el comando:

docker service ls

ID	NAME	MODE	REPLICAS	IMAGE
zgx1h9xl8kpo	geoos_db	replicated	1/1	mongo:4.4.1-bionic
23e111e47tzw	geoos_portal	replicated	1/1	docker.homejota.net/geoos/portal:latest
	PORTS			
	*:27017->27017/tcp			
	*:8091->8090/tcp			

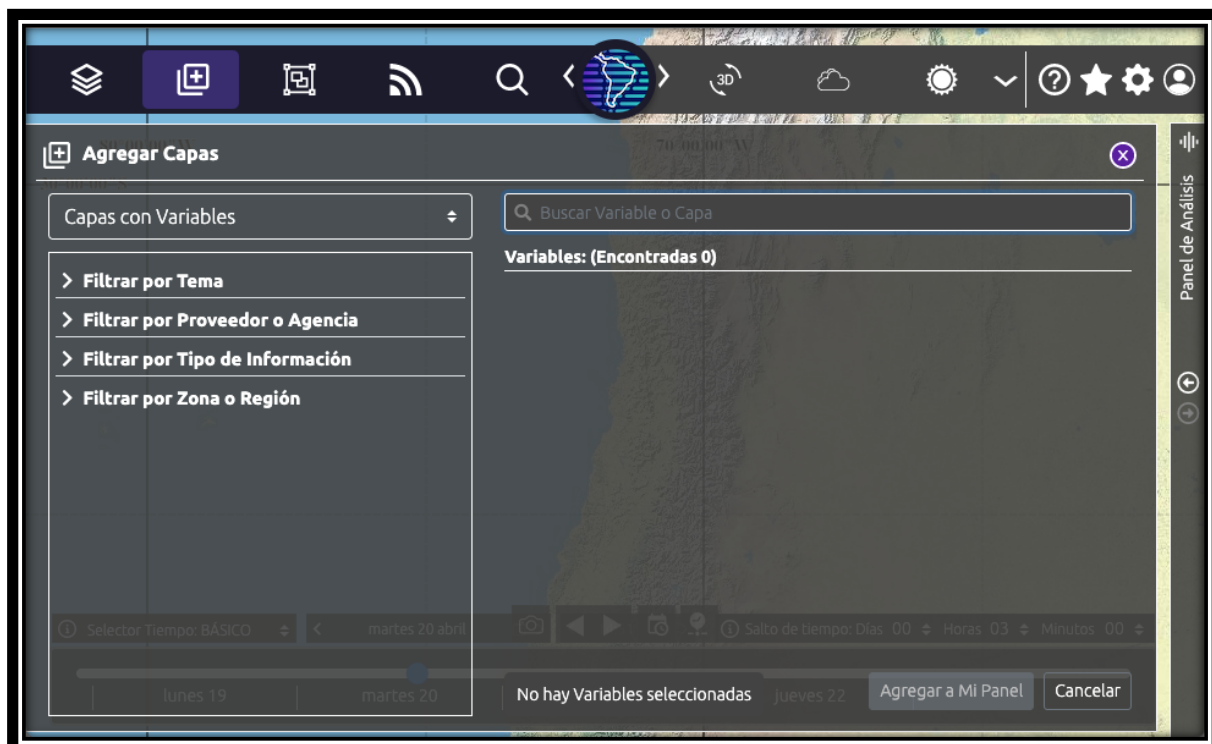
Para ver los logs específicos de un servicio (detectar problemas si no levantan, por ejemplo), se usa el nombre completo de éste, el que se compone del nombre del stack (geoos) y el del servicio dentro del stack. Por ejemplo, usando el comando:

```
docker service logs geoos_portal -f
```

Si todo está bien, se verá un log como este:

```
geoos_portal.1.t4miarxnn2yw@Huntress | [GEOOS HTTP Server 0.62] Listening at  
Port 8090
```

Para probar la instalación, es posible navegar desde un browser a la dirección IP del servidor (host), puerto 8091. Se debería cargar el Portal GEOOs, pero sin ninguna variable ni estación disponible:



Instalación de Componentes

GEOOs permite que sus componentes se puedan instalar de forma distribuida y que puedan ser compartidos (aquellos públicos) entre diferentes instalaciones. Desde la instalación de prueba del Portal, podemos configurarlo (**config/Portal.hjson**) para que utilice los componentes GeoServer y zRepo publicados en “geoos.org”. Para ellos modificamos el archivo “**portal.hjson**” para que incluya esos servidores:

```
# Servers
```

```
geoServers:["https://geoserver.geoos.org"]
```

```
zRepoServers:[{url:"https://zrepo.geoos.org", token:"geoos-public"}]
```

El componente Portal está diseñado para detectar los cambios de configuración “en caliente”, por lo que no es necesario reiniciar el servicio. Luego de aplicar los cambios y actualizar la página, el Portal muestra las mismas variables y estaciones que la instalación en geoos.org:

