

Volterra 11/02/2019

Versione: 1

INSTALLAZIONE DI G3W-SUITE

Di seguito le istruzioni per installare G3W-SUITE dockerizzato

INSTALLAZIONE DI DOCKER E DOCKER-COMPOSE

fonte:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04>

INSTALLAZIONE DI DOCKER

Si aggiungono le chiavi per i repository apt docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

si aggiunge il repository apt docker alla nostra lista:

```
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu focal stable"
```

installiamo la versione 'Community Edition' di Docker:

```
sudo apt install docker-ce
```

Verifichiamo che l'installazione sia andata buon fine e che il demone del servizio docker sia attivo:

```
sudo systemctl status docker
```

dovremmo vedere una risposta del genere:

```
docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor  
   preset: enabled)  
   Active: active (running) since Tue 2020-05-19 17:00:41 UTC; 17s ago  
   TriggeredBy: ● docker.socket  
     Docs: https://docs.docker.com
```

```
Main PID: 24321 (dockerd)
  Tasks: 8
  Memory: 46.4M
  CGroup: /system.slice/docker.service
          └─24321 /usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.sock
```

aggiungiamo il nostro utente al gruppo di sistema 'docker' per evitare di dover fare tutto da super utente:

```
sudo usermod -aG docker ${USER}
```

INSTALLAZIONE DI DOCKER-COMPOSE

fonte: <https://docs.docker.com/compose/install/>

Come da manuale docker installiamo lo script docker-compose:

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(un
ame -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

rendiamo eseguibile lo script:

```
sudo chmod +x /usr/local/bin/docker-compose
```

verifichiamo che docker-compose funzioni:

```
docker-compose --version
```

```
docker-compose version 1.25.4, build 8d51620a
```

INSTALLAZIONE DI G3W-SUITE-DOCKER

Cloniamo la serie di script, dockerfile e docker-compose file yml per avviare e installare g3w-suite.

Cloniamo l'ultima versione stabile v3.1.x

```
git clone -b v3.1.x https://github.com/g3w-suite/g3w-suite-docker.git
```

entriamo dentro alla cartella clonata e verifichiamo che il branch è quello giusto:

```
cd g3w-suite-docker
git branch
* v3.1.x
```

Creiamo un file `.env` per le variabili di ambiente che i container docker utilizzeranno al loro interno, copiando quello `.example` fornito e andandolo a modificare

```
cp .env.example .env
nano .env
```

il file risulta così strutturato

```
WEBGIS_PUBLIC_HOSTNAME=dev.g3wsuite.it

# Shared volume mount (docker internal: shared-volume)
WEBGIS_DOCKER_SHARED_VOLUME=/tmp/shared-volume-g3w-suite-dev

# Suite
# Docker internal DB
G3WSUITE_POSTGRES_USER_LOCAL=g3wsuiteudb
G3WSUITE_POSTGRES_PASS=gwte#85utha76u
G3WSUITE_POSTGRES_DBNAME=g3wsuite
G3WSUITE_POSTGRES_HOST=postgis
G3WSUITE_POSTGRES_PORT=5432

G3WSUITE_QDJANGO_SERVER_URL=http://g3w-suite/ows/

# Caching
G3WSUITE_TILECACHE_PATH=/shared-volume/tile_cache/
TILESTACHE_CACHE_BUFFER_SIZE=256
TILESTACHE_CACHE_TOKEN=wtegdhnr572364yrhfteg

# Unicorn workers (default to 8)
G3WSUITE_GUNICORN_NUM_WORKERS=8
```

E' sufficiente modificare le prime due variabili (e volendo anche `G3WSUITE_POSTGRES_PASS`, per cambiare la password per il db postgres) come segue:

```
WEBGIS_PUBLIC_HOSTNAME=g3wsuite.provincia.biella.it

# Shared volume mount (docker internal: shared-volume)
WEBGIS_DOCKER_SHARED_VOLUME=/home/walter/shared-volume-g3w-suite
...
```

Installiamo e avviamo g3w-suite:

```
docker-compose up -d
```

se tutto è andato a buon fine la suite sarà disponibile al seguente indirizzo:

`http://<ip_server>:8080`

nel caso volessimo farla partire sulla porta 80 dobbiamo aprire il file `docker-compose.yml` e modificare alla riga 65 come segue:

```
...
nginx:
  image: nginx:1.19.5
  ports:
    - "8080:8080"
    - "443:443"
  expose:
    - "8080"
  volumes:
  ...
```

```
...
nginx:
  image: nginx:1.19.5
  ports:
    - "80:8080"
    - "443:443"
  expose:
    - "8080"
  volumes:
  ...
```

ed eseguire nuovamente:

```
docker-compose up -d
```

