Monitorer Gunicorn GeoNature

- Modifier la *stack Docker telegraf* du serveur *web-srv* :
  - Modifier la configuration de *Telegraf* pour le serveur *web-srv* en ajoutant le support de *statsd* comme indiquer dans la doc du plugin.
  - Ajouter la publication du port UDP 8125 sur l'hôte uniquement en local (*127.0.0.1*) dans le fichier `docker-compose.yml` de *telegraf*
  - Envoyer sur le serveur les modification à l'aide de *rsync*
- Se connecter au serveur en tant qu'*admin* et se rendre dans le dossier *~/docker* : `ssh -t admin@web-<region>-sinp 'cd docker; exec bash -l' '' * Arrêter puis redémarrer le stack *telegraf* pour prendre en compte les changement : ''docker-compose down ; docker-compose up -d`
- Se connecter au serveur en tant que *admin* : '' ssh admin@web-<region>-sinp ''
  - Passer en root : `sudo -i`
  - Nous allons surcoucher les fichiers service Systemd de GeoNature, UsersHub, TaxHub et Atlas:
    - Créer une fichier *overide* : `systemctl edit geonature.service`
    - Ajouter le contenu suivant (nous ajoutons aussi des options de redémarrage auto) :

```
[Unit]
StartLimitIntervalSec=6min
StartLimitBurst=5

[Service]
ExecStart=
ExecStart=/home/geonat/www/geonature/backend/venv/bin/gunicorn geonature:create_app() \
                --statsd-host "localhost:8125" \
                --statsd-prefix "geonature"  \
                --name "${GUNICORN_PROC_NAME}" --workers "${GUNICORN_NUM_WORKERS}" \
                --bind "${GUNICORN_HOST}:${GUNICORN_PORT}" --timeout="${GUNICORN_TIMEOUT}"
Restart=on-failure
RestartSec=1min
```

    - Prendre en compte les changements : `systemctl daemon-reload`
    - Relancer le service : `systemctl restart geonature.service`
  - Répéter cette procédure pour l'Atlas, UsersHub et TaxHub :
    - Pour l'Atlas, le service se nomme `geonature-atlas.service` :

```
[Unit]
StartLimitIntervalSec=6min
StartLimitBurst=5

[Service]
ExecStart=
ExecStart=/home/geonat/www/atlas/venv/bin/gunicorn atlas.wsgi:app \
```

```
                --statsd-host "localhost:8125" \
                --statsd-prefix "atlas"  \
                --name "${GUNICORN_PROC_NAME}" --workers
"${GUNICORN_NUM_WORKERS}" \
                --bind "${GUNICORN_HOST}:${GUNICORN_PORT}" --
timeout="${GUNICORN_TIMEOUT}"
Restart=on-failure
RestartSec=1min
```

- Pour TaxHub, le service `taxhub.service` :

```
[Unit]
StartLimitIntervalSec=6min
StartLimitBurst=5

[Service]
ExecStart=
ExecStart=/home/geonat/www/taxhub/venv/bin/gunicorn
apptax.app:create_app() \
                --statsd-host "localhost:8125" \
                --statsd-prefix "taxhub"  \
                --name "${GUNICORN_PROC_NAME}" --workers
"${GUNICORN_NUM_WORKERS}" \
                --bind "${GUNICORN_HOST}:${GUNICORN_PORT}" --
timeout="${GUNICORN_TIMEOUT}"
Restart=on-failure
RestartSec=1min
```

- Pour UsersHub, , le service `usershub.service` :

```
[Unit]
StartLimitIntervalSec=6min
StartLimitBurst=5

[Service]
ExecStart=
ExecStart=/home/geonat/www/usershub/venv/bin/gunicorn
app.app:create_app() \
                --statsd-host "localhost:8125" \
                --statsd-prefix "usershub"  \
                --name "${GUNICORN_PROC_NAME}" --workers
"${GUNICORN_NUM_WORKERS}" \
                --bind "${GUNICORN_HOST}:${GUNICORN_PORT}" --
timeout="${GUNICORN_TIMEOUT}"
Restart=on-failure
RestartSec=1min
```

- Vérifier la prise en compte des nouvelles métriques dans *InfluxDB*. Leurs noms doivent débuter par "geonature_...", "atlas_...", etc.
  - En utilisant l'interface disponible avec InfluxDB v2 : https://influxdb.<domaine-sinp>/
  - En passant par le container :

        - Se connecter à l'instance hébergeant InfluxDB, normalement "bkp-srv" : '' ssh admin@bkp-<region>-sinp ''
        - Se connecter à InfluxDB :
            - pour la v2 : '' docker exec -it monitor-influxdb influx v1 shell ''
            - pour la v1 : '' docker exec -it monitor-influxdb influx ''
        - Sélectionner la base stockant les métriques : '' USE "telegraf" ''
        - Afficher les métriques : '' SHOW MEASUREMENTS ''
- Pour le Dashboard, nous avons adapté à InfluxDB le Dashboard Gunicorn suivant initialement prévu pour Prometheus :
    - https://gist.github.com/dmyerscough/59896aa752ba48794d2aef4c7a0fdd6e
    - https://github.com/pilosus/prometheus-client-python-app-grafana-dashboard/blob/master/flask-web-app.json

From:
https://sinp-wiki.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
https://sinp-wiki.cbn-alpin.fr/serveurs/installation/web-srv/monitorer-gunicorn

Last update: **2023/06/01 19:01**