

# Installer l'agent Telegraf via Docker

- Se connecter en tant qu'admin : `ssh admin@db-<region>-sinp`
- Se placer dans le dossier : `cd ~/docker/telegraf`
  - Lancer le container afin d'afficher les logs : `docker-compose up`
  - Corriger les éventuels problèmes de configuration de Telegraf s'affichant dans les logs
  - Lorsque tout fonctionne correctement, lancer le container en tant que service (option -d) : `docker-compose up -d`

## Monitorer différents services spécifiques

- Nginx:
  - status : [Doc config du service status dans Nginx](#)
  - access log : ajouter l'utilisateur "telegraf" au groupe "adm" dans le fichier `docker-compose.yml` via `group_add`. Puis configurer Telegraf en ajoutant le support de l'input Tail et du parser Grok (voir `telegraf.conf` ci-dessous comme exemple).
- Postgresql : ajouter l'utilisateur "telegraf" aux roles Postgresql et au fichier `pg_hba.conf` ([Voir config Postgresql](#)).
- Unicorn : [Doc monitorer Unicorn GeoNature](#)
- Disques additionnels : ajouter les entrées complémentaires à l'input `diskio`.

## Monitorer InfluxDB v2

- Pour monitorer InfluxDB v2, il est nécessaire d'utiliser l'url `influxdb.<domaine-sinp>/metrics` qui retourne des infos au format *prometheus*.
- Il est conseillé de stocker ces métriques dans un bucket distinct nommé `oss_metrics`.
- Pour créer le bucket, il est possible via l'interface `influxdb.<domaine-sinp>` d'ajouter un template qui va le créer pour nous ainsi qu'un dashboard :
  - Aller sur le menu Settings > Templates
  - Ajouter l'url suivante : [https://raw.githubusercontent.com/influxdata/community-templates/master/influxdb2\\_oss\\_metrics/influxdb2\\_oss\\_metrics.yml](https://raw.githubusercontent.com/influxdata/community-templates/master/influxdb2_oss_metrics/influxdb2_oss_metrics.yml)
  - Vérifier la création du bucket "oss\_metrics" et du Dashboard
- Configurer ensuite Telegraf en ajoutant une seconde sortie vers le bucket "oss\_metrics", le plugin `inputs.prometheus` et utiliser la mécanique de tag du fichier de conf de Telegraf pour rediriger les métriques prometheus vers la sortie correspondant au bucket "oss\_metrics".
- Redémarrer la stack telegraf : `docker-compose down ; docker-compose up -d`
- Vérifier sur `manager.<domaine-sinp>` la sortie des logs de Telegraf.
- Vérifier dans `influxdb.<domaine-sinp>` que le plugin prometheus alimente bien le bucket "oss\_metrics".
- Les métriques renvoyées par le plugin prometheus doivent être en version 1 pour être compatible avec les Dashboard open source proposé pour Grafana et l'interface web d'InfluxDB.

## Exemple de fichier telegraf.conf complet

Exemple avec :

- le support d'InfluxDB v2 accessible sur le port 8086 de l'ip privée 10.0.1.10
- le statut de Nginx sur [http://172.18.5.1:9090/nginx\\_status](http://172.18.5.1:9090/nginx_status) (voir config Nginx),
- l'analyse des logs de Nginx via Tail et Grok,
- la surveillance de Postgresql
- la surveillance des service du système à l'aide du script *srvstatus*
- la surveillance de Gunicorn pour GeoNature à l'aide de Statsd

## Tester une métrique

- Pour tester une métrique, il est possible de rajouter une sortie au fichier config comme dans l'exemple ci-dessus présent à la fin du fichier.
- Il est aussi possible de :
  - se connecter au container : `docker exec -it telegraf /bin/bash`
  - d'exectuer un test avec la commande (Ex. ici avec la métrique *disk*) : `telegraf --input-filter=disk --test --debug`

From:

<http://sinp-wiki.cbn-alpin.fr/> - **CBNA SINP**

Permanent link:

<http://sinp-wiki.cbn-alpin.fr/serveurs/installation/docker-telegraf?rev=1685948553>

Last update: **2023/06/05 07:02**

