Installer script srvstatus

- **Notes** :
  - le script ratibor78/srvstatus nous permet de surveiller le status de services *Systemd* via *Grafana*.
  - le plugin `inputs.systemd_status` ne fonctionne pas dans un container Docker. Voir : https://github.com/influxdata/telegraf/issues/7689
- TODO :
  - tester l'utilisation de la commande suivante pour générer un fichier json de status à la place du script srvstatus:

```
systemctl list-units   --type service   --full   --all   --plain   --no-legend   --no-pager   | sed 's/ \{1,\}/,/g'   | jq     --raw-input     --slurp     'split("\n") | map(split(",")) | .[0:-1] | map( { "unit": .[0], "load": .[1], "active": .[2], "sub": .[3], "description": .[4] } )
```

  - Sur Bullseye la commande suivante fonctionne :

```
 systemctl list-units -t service --full --all --output=json --no-pager
```

  - **Problème** la commande n'affiche pas le temps de fonctionnement alors que le script *srvstatus* le fait...

Installer le script

- Se placer dans /opt : `cd /opt`
- Cloner le dépôt : `git clone https://github.com/ratibor78/srvstatus.git`
- Se placer dans le dossier du script : `cd /opt/srvstatus`
- Installer le paquet : `apt install python3-venv`
- Créer un *venv* avec *Python 3* : `python3 -m 'venv' ./venv`
- Activer le *venv* : `source venv/bin/activate`
- Installer les paquets requis : `pip install -r requirements.txt`
- Rendre exécutable le script : `chmod +x ./service.py`

Mettre à jour le script

- Se placer dans le dossier du script : `cd /opt/srvstatus`
- Mettre à jour le dépôt : `git pull`
- Activer le *venv* : `source venv/bin/activate`
- Installer les paquets requis : `pip install -r requirements.txt`
- Désactiver le *venv* : `deactivate`
- Vérifier le bon fonctionnement : `vi status.json`

Configurer le script

- Copier le fichier *setting.ini* depuis le dépôt Github *sinp-<region>-srv* : `wget https://raw.githubusercontent.com/cbn-alpin/sinp-<region>-srv/main/<instance>-srv/opt/srvstatus/settings.ini`
    - Suivant le SINP et le serveur, remplacer `<region>` (par `paca` ou `aura`) et `instance` (par `web` ou `db`)
- Si le fichier n'est pas dispo dans le dépôt, vous pouvez créer le fichier de config puis le stocker dans le dépôt Git *sinp-<region>-srv* : `cp settings.ini.back settings.ini`
- Vérifier ou définir les noms des services *Systemd* à surveiller dans le fichier *setting.ini* : '' vi settings.ini ''

Préparer le Cron du script

- Copier le fichier *srvstatus.cron* depuis le dépôt Github *sinp-<region>-srv* : `wget https://raw.githubusercontent.com/cbn-alpin/sinp-<region>-srv/main/<instance>-srv/opt/srvstatus/srvstatus.cron`
    - Suivant le SINP et le serveur, remplacer `<region>` (par `paca` ou `aura`) et `instance` (par `web` ou `db`)
- Si le fichier n'est pas disponible dans le dépôt, vous pouvez créer le fichier puis le stocker dans le dépôt Git *sinp-<region>-srv* : '' vi srvstatus.cron ''
- Y placer le contenu suivant :

```
# /etc/cron.d/srvstatus: crontab entries for the srvstatus script
# Copy this file into /etc/cron.d/ without .cron extension : cp srvstatus.cron /etc/cron.d/srvstatus
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

* * * * * root (sleep 10 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
* * * * * root (sleep 20 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
* * * * * root (sleep 30 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
* * * * * root (sleep 40 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
* * * * * root (sleep 50 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
* * * * * root (sleep 60 ; /opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json)
```

- Copier le fichier de *Cron* pour l'activer : `cp srvstatus.cron /etc/cron.d/srvstatus`

Configurer Telegraf

- Avant lancer *Telegraf*, assurez vous d'avoir lancer manuellement une première fois le script

srvstatus en *root* afin de créer le fichier *status.json* afin qu'il soit correctement pris en compte par le volume de Docker (voir ci-dessous) : `/opt/srvstatus/venv/bin/python /opt/srvstatus/service.py > /opt/srvstatus/status.json`
- Penser à ajouter le volume correspondant au fichier *status.json* dans le *docker-compose.yml* afin que *Telegraf* y est accès :

```
services:
  telegraf:
    volumes:
      - /opt/srvstatus/status.json:/opt/srvstatus/status.json
```

- Ajouter dans le fichier *telegraf.conf* utiliser par le container *Docker* la configuration suivante :

```
[[inputs.exec]]
  commands = ["cat /opt/srvstatus/status.json"]
  timeout = "5s"
  name_override = "services_stats"
  data_format = "json"
  tag_keys = ["service"]
```

Stocker les logs du Cron dans leur propre fichier

- **Objectif** : séparer les logs des Cron dans un fichier de log à part afin qu'ils ne surchargent pas le fichier syslog. Le script `srvstatus` écrit plusieurs lignes de log toutes les minutes.
- Éditer le fichier de config de Rsyslog :

```
vi /etc/rsyslog.conf
```

  - Remplacer la ligne :

    ```
    *.*;auth,authpriv.none     -/var/log/syslog
    ```

    par

    ```
    *.*;cron,auth,authpriv.none     -/var/log/syslog
    ```

  - Dé-commenter la ligne :

    ```
    # cron.*              /var/log/cron.log
    ```

- Redémarrer le service Rsyslog avec :

```
systemctl restart rsyslog.service
```

- Redémarrer le service Cron avec :

```
systemctl restart cron.service
```

- Vérifier :
  - la présence du fichier `cron.log` et l'ajout de nouveaux logs avec :

```
vi /var/log/cron.log
```

- l'absence de log du Cron dans le fichier `syslog` avec :

```
vi /var/log/syslog
```

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

Permanent link:
https://wiki-sinp.cbn-alpin.fr/serveurs/installation/script-srvstatus

Last update: **2023/06/08 08:14**