# Scripting System

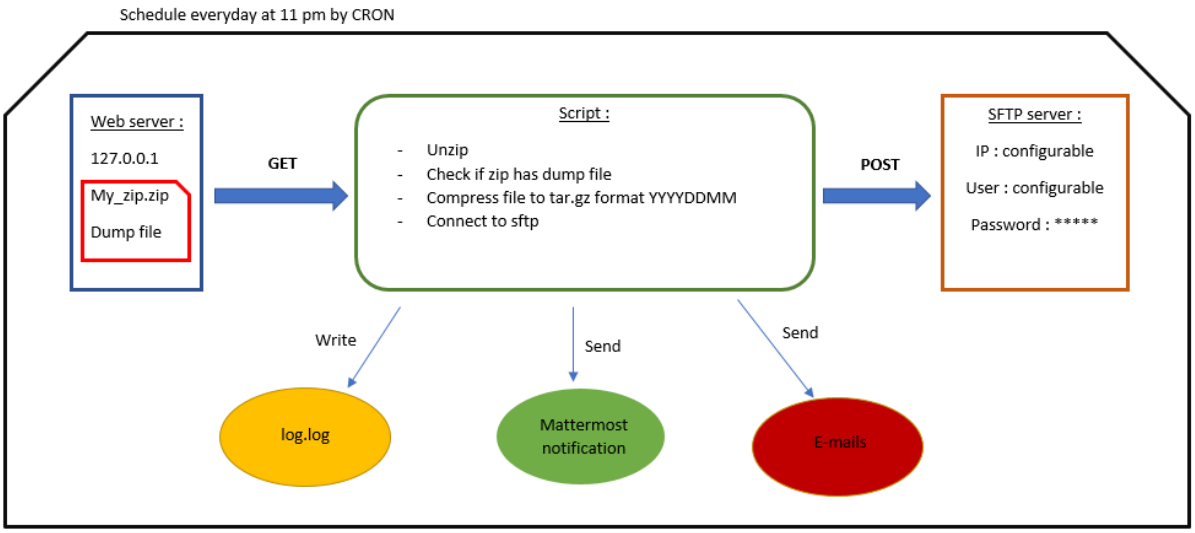
## Summary



# Introduction

This program is an archival program. It requests a zip file on a web server (unconfigurable). Then unzip file, check if a specific dump file is in it. If there is the dump file, then get its last modification date and compare with today. If dump file was changed today, then upload it on sftp server (configurable). Program removes all expired files too.

Program write every thing in logs and can send e-mails or/and Mattermost notification.



# 1 – Create SSH keys

To set up installation, please make sure you have "sudo" access, otherwise you will not be able to install and configure all.

This script requires SFTP Connection, that is why you need to create public & private key pair.

To process open a terminal and type "ssh-keygen -t rsa", and let a blank and press enter for all questions send by terminal.

You should get (for your session):


# 2 - Download project

All bash files are set up for a Linux distribution, if you are on Windows or other, then you will have to configure this section manually (cf Annexes).

This project is enabled at https://gitlab.com/JujuDeFruit/Scripting-System.

Download project at this URL, and unzip this where you want on your computer.

By default, Python 3 is already installed on all Linux distribution.

Type "python3 --version" anywhere in terminal to be sure. If this command returns nothing, please download and install python3 there: https://www.python.org/downloads.

After this, you can notice you have 3 bash (.sh) files. We will use them to launch our program.

- install_dependencies.sh: this file permits to install everything this project need, such as libraries, CRON file and package managing pip.
- server.sh: launch web server in which program will get zip file.
- gen_config.sh: generate a template of config.json file you will need later if you delete this one accidentally, or to regenerate this file format. Please notice, if you run this file, your current configuration will be erased.

Be careful, for all this configuration, please just modify "config. json" file. You normally cannot modify other files with a standard permission.

# 3 – Install dependencies

Still in terminal, navigate to "Scripting-System" folder, and execute "install-dependencies.sh" file. To do this, type "./install-dependencies.sh" in Scripting-System folder.

```
julien@julien:~/Desktop/Scripting-System$ ./install_dependencies.sh
Python 3.7.3
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  python3-pip
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0 o/171 ko dans les archives.
Après cette opération, 707 ko d'espace disque supplémentaires seront utili
sés.
Sélection du paquet python3-pip précédemment désélectionné.
(Lecture de la base de données... 463457 fichiers et répertoires déjà inst
allés.)
Préparation du dépaquetage de .../python3-pip_18.1-5_all.deb ...
Dépaquetage de python3-pip (18.1-5) ...
Paramétrage de python3-pip (18.1-5) ...
Traitement des actions différées (« triggers ») pour man-db (2.8.5-2) ...
pip 18.1 from /usr/lib/python3/dist-packages/pip (python 3.7)
Requirement already satisfied: email_validator in /home/julien/.local/lib/
python3.7/site-packages (1.1.2)
Requirement already satisfied: dnspython>=1.15.0 in /home/julien/.local/li
b/python3.7/site-packages (from email_validator) (2.0.0)
Requirement already satisfied: idna>=2.0.0 in /usr/lib/python3/dist-packag
```

```
es (from email_validator) (2.6)
Requirement already satisfied: pysftp in /home/julien/.local/lib/python3.7
/site-packages (0.2.9)
Requirement already satisfied: paramiko>=1.17 in /home/julien/.local/lib/p
ython3.7/site-packages (from pysftp) (2.7.2)
Requirement already satisfied: cryptography>=2.5 in /usr/lib/python3/dist-
packages (from paramiko>=1.17->pysftp) (2.6.1)
Requirement already satisfied: pynacl>=1.0.1 in /home/julien/.local/lib/py
thon3.7/site-packages (from paramiko>=1.17->pysftp) (1.4.0)
Requirement already satisfied: bcrypt>=3.1.3 in /home/julien/.local/lib/py
thon3.7/site-packages (from paramiko>=1.17->pysftp) (3.2.0)
Requirement already satisfied: cffi>=1.4.1 in /home/julien/.local/lib/pyth
on3.7/site-packages (from pynacl>=1.0.1->paramiko>=1.17->pysftp) (1.14.4)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from
 pynacl>=1.0.1->paramiko>=1.17->pysftp) (1.12.0)
Requirement already satisfied: pycparser in /home/julien/.local/lib/python
3.7/site-packages (from cffi>=1.4.1->pynacl>=1.0.1->paramiko>=1.17->pysftp
) (2.20)
no crontab for julien
Successfully crontab created in /var/spool/cron/crontabs.
Logs for this crontab are in /var/log/syslog.
To restart service, type sudo service cron restart/start.
```

Make sure any error appeared.

If an error occurred, you will have to install manually dependencies, please check in annexes of the documentation. (be sure pip, libraries and cron are installed or created).

# 4 – Configuration

Now, if everything happened well, we can configure program.

Edit configuration file: "config.json".

You now have this:

config.json 504 Bytes

```
{
        "zip": "my_zip",
        "file": "dumpfile",
        "time-to-save": "1",
        "sftp": {
                "ip": "my_sftp_ip",
                "user": "my_sftp_user",
                "password": "my_sftp_password"
        },
        "email": {
                "send-emails": "yes",
                "auth": {
                        "email": "exemple@exemple.com",
                        "password": "my_password_for_mail"
                },
                "server": {
                        "ip": "smtp.mail.yahoo.com",
                        "port": "465"
                },
                "log-file-attached": "yes",
                "title": "My Daily report",
                "dest": [
                        "first@email.com",
                        "second@email.com"
                ]
        },
        "notification": "never"
}
```

In this file, you will can configure program that will run.

If you accidently erase this file, you can regenerate buy running "gen_config.sh" file.

This file is at JSON format, so you have to respect a format when writing data in this one.

Here https://jsonformatter.curiousconcept.com/, a JSON formatter to avoid errors when writing in it.

## Description of configuration file

| Key | Type | Value |
|---|---|---|
| zip | String | Name of zip to get on web server (ex: "my_zip"). |
| file | String | Name of dump file to get on web server contained in zip file and archived in SFTP Server (ex: "my_dump_file"). |
| time-to-save | String | Time in days to store zip files on SFTP server (ex: "2"). Default is 10. |
| sftp | Dictionary | Set of data to configure access to SFTP server. Description here. |
| email | Dictionary | Set of data to configure e-mails. Description here. |

| | | |
|---|---|---|
| notification | String | "**always**": Mattermost notification is always sent at the end of the program.<br>"**error**": Mattermost notification is sent at the end only if error(s) occurred.<br>"**never**": never send Mattermost notification.<br>Default is "always". |

### SFTP configuration (set of data)

| Key | Type | Value |
|---|---|---|
| ip | String | IP of SFTP server to connect. (ex: "10.25.64.32") |
| user | String | Username of user to connect on server. (ex: "guest") |
| password | String | Password of user to connect to. (ex: "password") |

### E-mail configuration (set of data)

| Key | Type | Value | | |
|---|---|---|---|---|
| send-emails | String | "**yes/y**": send e-mails when the program is over.<br>"**no/n**": do not send e-mails when program is over.<br>Default is "yes". | | |
| auth | Dictionary | email | String | Email used to send e-mail. |
| | | password | String | Password of e-mail. |
| server | Dictionary | ip | String | IP of e-mail server. It can be its name (ex: "smtp.mail.yahoo.com"). It can be blank, then program use internal e-mail server. |
| | | port | String | Port to connect to e-mail server. (ex: "465") |
| log-file-attached | String | "**yes/y**": attach log file to send e-mails.<br>"**no/n**": do not attach log file to send e-mails.<br>Default is "yes". | | |
| title | String | Title of e-mails. (ex: "My daily report"). Default is "Scripting system rapport". | | |
| dest | Array of strings | List of e-mails addresses to send rapport e-mail. (ex: [firstname.lastname@domain.com, professional@email.com]). It can be empty. | | |

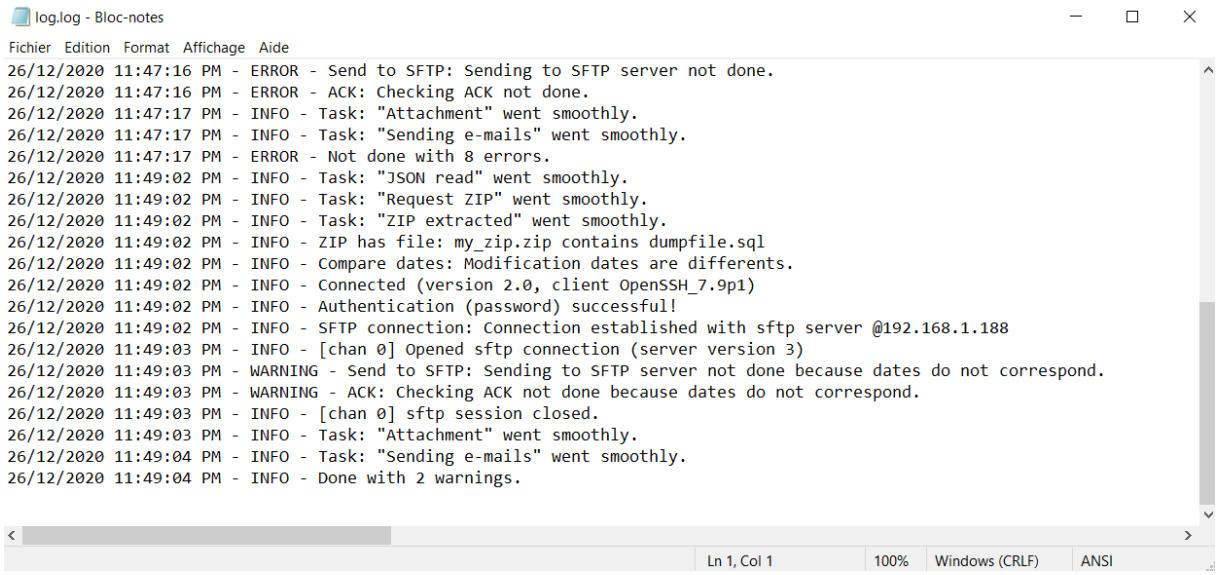
All underlined keys are the required keys. Otherwise, program raises an error.

# 5 – Get informed of my archival program

As explained in the introduction, this file takes user informed of its progress. Each information processed by script is written in log.log file. Moreover, if you configured correctly the configuration file, you can choose to receive an e-mail or/and a Mattermost notification to get in touch with the process.

Every time process does not work, check "Scripting-System/log.log" file. Errors are reported and can be fixed from this.

Example of content of log file:




# Annexes

## Install dependencies manually

We need to install pip (dependencies manager), email_validator and pysftp libraries required in the main program.

- Install pip: "sudo apt-get install python3-pip"
- Install libraries: "python3 -m pip email_validator pysftp"

If a dependency issue appears such as broken dependencies or anything else repair them manually with "sudo apt-get update –fix-missing", and retry installing packages.

To be sure all packages are correctly installed, be sure the following commands return something:

- "python3 -m pip --version",
- "python3 -m pip show email_validator"
- "python3 -m pip show pysftp"

## Create CRON manually

- Create default cron file (be careful: do not use sudo, or you will create cron for root):

  "crontab -e"

- Edit this one by writing without quotes:

  "0 23 * * * cd <path_of_your_folder>; python3 main.py"

- Verification crontab is created:
  "crontab -l" returns you what you wrote in file.