



F10wer Evaluation ISO image

The F10wer Evaluation ISO image (shortly, the “eval”) is a quick way to deploy the F10wer platform solution in your virtualized environment to quickly evaluate it.

The ISO has been tested on virtual systems both BIOS both EFI boot.

The ISO is a standard stable Debian 12 with the following software customizations that already installed and preconfigured :

- F10wer 3.5 runtime and development environment with Python 3.12 (evaluation version)
- Clickhouse OLAP database server 25.12.2.54
- Apache 2.4.66-1~deb12u1

The scope of the “eval” image is to test F10wer to see if it can suit your environment and *could* not be the best deployment solution. Its sole scope is this: evaluate; if you want to put it in your production environment, unless you have a support contract and a software license, you are on your own.

Evaluation version limitations

The evaluation version has the following limitations:

1. Multi-threading is disabled. You will have only 1 thread processing incoming data.
2. The Flow Matrix is limited to 50 entries (no limits in the full version)
3. Does not handle Netflow V9 and IPFIX (no limits in the full version)
4. Does not handle IPv6 Flows (no limits in the full version)
5. Allows only Dynamic SSLv23 In-Memory Certificates for the API (regenerated after each daemon restart). The full version uses static TLS 1.2 certificate for the API.

Apart from this, it has no time limits, no limits on the number of flow exporters or anything else.

For more information about F10wer, refer to the website <https://f10wer.me> and the related user manual.



Hardware requirements

Minimum Hardware

- CPU: 64bit (Intel 64 or AMD64)
- 4 GB RAM
- Bootable CD-ROM-drive or USB boot support
- Monitor with a resolution of 640x480 for the installation
- Hard disk with at least 30 GB of disk space
- Ethernet network interface card

The Evaluation runs ok with 4Gb of RAM memory (my lab runs on an 8Gb RAM VM in Proxmox). As a rule of thumb: the more resources you give it, the better the performance.

You should tune-up Clickhouse to better use all the memory you will provide it with.

Regarding the storage, as a personal suggestion, in 2026, don't use nothing worse than an SSD. NVME storage is much way better. Worse will work, but will be slow as death.

Also regarding the CPU, my homelab runs on an ODROID-H3 which has an Intel(R) Pentium(R) Silver N6005 @ 2.00GHz, I am sure you have better hardware than me :-)

System services listening:

Port 80/tcp: F10wer Web Interface (you go with your browser here)

Port 8000/tcp: Raw Streamlit web interface (using web sockets) – you can use this too.

Port 7443/tcp: F10wer API

Port 8123/tcp: ClickHouse Server API

Port 22/tcp: SSH for remote server management

Port 2055/udp: F10wer Netflow/IPfix Listener (traffic here is redirected by samplicate)

Port 6344/udp: F10wer sFlow V5 Listener (traffic here is redirected by samplicate)

Port 2056/udp: Samplicator Netflow/IPfix Listener (forwards to F10wer, you can add other targets if you want)

Port 6343/udp: Samplicator sFlow V5 Listener (forwards to F10wer, you can add other targets if you want)

To add other targets, just check `/usr/local/bin/samplicate.sh` and eventually `systemctl restart samplicate`.

System users:

root – asked on install

f10wer – password f10werr0x

You can change the passwords using the standard Linux command `passwd`. For remote access, login via SSH as f10wer user and use the `su –` command. Hey, it's a Debian 12, you can mess around as you want! Remove the other unused users if needed but not the system ones like f10wer, clickhouse, apache, etc.

Application user:

F10wer user f10wer – password: f10werr0x (you can manager users with the `/opt/f10wer/bin/flcli-*user*` commands.

Remember to adjust any script that contains the password.

Clickhouse server user default password: f10werr0x – The password hash is stored in `/etc/clickhouse-server/users.d/default-password.xml` – Follow the Clickhouse procedures to change it if you need so.

System services to check:

The ISO, to work, needs some services that are defined in the directory `/etc/systemd/system` and are:

- `samplicate.service`: The Samplicate process for forwarding the flow traffic to F10wer or other systems of your choiche if needed



- `flower.service` (-> `/opt/flower/systemd/flower.service`): The F10wer product – receives and processes all Netflow/Ipflix/sFlow traffic to be feeded in Clickhouse by the `flowerpumper`
- `flowerpumper.service` (-> `/opt/flower/systemd/flowerpumper.service`): Takes in input the CSV files created by F10wer for injection in Clickhouse, ELK, etc. Full source code available for easy customization.
- `flower-webui.service` (-> `/opt/flower/systemd/flower-webui.service`): The F10wer dashboard. Full source code available for easy customization.
- In `/lib/systemd/system/clickhouse-server.service` you will find the clickhouse server process, which is where all the data is going to be stored.

All services are managed by `systemd`.

Cleanup

For a quick & easy cleanup of the system, in the `/usr/local/bin` folder there is the script `initdb.sh`.

You can run it in order to cleanup the Clickhouse database and recreate Database and tables.

Example:

```
root@flower-eval:~# /usr/local/bin/initdb.sh
```

Note: if you change the password for the Clickhouse or the F10wer user, remember to fix the script too.

F10wer customization

Refer to the user manual, anyway all files are under `/opt/flower/etc` and `/opt/flower/iplist`.

As a **minimum rule of thumb**, at least customize the `flower_internal_networks.conf` providing it with the network subnets you have in your site.

Storage space

The ISO image is quite small, estimating a 30Gb disk space should be sufficient (in most cases) to showcase the F10wer Platform. Anyway if you have an army of Flow Exporters and tons of data incoming, I suggest you to add a virtual disk (better in form of NVME storage) and using standard linux LVM commands to enlarge the filesystem. F10wer is self-contained in `/opt/flower` and the `data` directory grows, specially if you enable realtime LUA scripts, maybe you want/can move it onto its separate space, you are free to do what you prefer. Consider that where the storage also really grows is in the `/var/lib/clickhouse`, where the DB is stored, so having a single filesystem could not be a so bad idea at all. Do your math.

Network Configuration

The network configuration is normally asked in the installation phase during the Debian image install.

It's Debian, you can adjust it for your needs. Use a fixed IPv4 address and point all your flow exporters to it on the 2056 and 6343 UDP ports.

Centralized Syslog

The rsyslog installed in this distribution can receive log messages on standard port 514 and store it in the Clickhouse database server, just check in `/etc/rsyslog.d/50-clickhouse.conf` and uncomment all the lines. It is disabled by default to avoid filling up your storage, but you can use it and it's integrated in the F10wer dashboard.



Packet filtering

A local packet filter is enabled and installed by using `firewall-cmd` (which in turns uses `nftables`).

This is the default configuration:

```
root@flower-eval:~# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports: 2056/udp 2055/udp 6343/udp 6344/udp 80/tcp 443/tcp 7443/tcp 9000/tcp 514/tcp 514/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Hacking

Except for the core of FLOWer daemon, everything is provided with sources. You can model the FLOWer Platform as best as you want. The dashboard is in `/opt/flower/web` and it's written in `Python 3.12` and `Streamlit`. Create the views you want and enjoy, and if you use it seriously, buy the license !

Note: Sometimes, after installation the `flowerpumper` service is not started, check it out with:

```
# systemctl status flowerpumper
```

and eventually start it with:

```
# systemctl start flowerpumper
```

Hope you will enjoy FLOWer !

gilberto