

Installing and Operating FORT 1.7.0.experimental

Ubuntu Server 26.04 - Verified Step-by-Step Guide

Source build, RPKI validation, systemd service, and RTR access from the local network

Autor: Nicolas Antoniello (Git: 65007)

Scope. This guide consolidates the steps that were tested successfully on Ubuntu Server 26.04 to install FORT 1.7.0.experimental. The official .deb package was not used because it declares legacy dependency package names that Ubuntu 26.04 no longer provides.

1. Verified environment

Component	Verified value
Operating system	Ubuntu Server 26.04 (resolute)
Architecture	amd64 / x86_64
FORT	1.7.0.experimental
Installation method	Build from the official source tarball
Configuration directory	/etc/fort
Local RPKI repository	/var/lib/fort/repository
VRP/ROA output	/var/lib/fort/output/validated-roas.csv
Service	fort.service
RTR port	TCP 8323
Maximum RTR version	1
Recommended time zone	UTC

2. Why FORT is built from source

The FORT 1.7.0.experimental Debian package was recognized correctly, but apt could not install it because it requires legacy package names:

```
fort : Depends: libmicrohttpd12
       Depends: libxml2
```

Ubuntu 26.04 provides the equivalent runtime packages under these names:

```
libmicrohttpd12t64
libxml2-16
```

The verified solution was to compile FORT against the native Ubuntu 26.04 libraries. Do not force dependencies or mix packages from older Ubuntu or Debian releases.

3. Install build dependencies

```
sudo add-apt-repository universe
sudo apt update

sudo apt install -y \
    build-essential \
    pkg-config \
    rsync \
    libjansson-dev \
    libssl-dev \
    libcurl4-openssl-dev \
    libxml2-dev \
    libmicrohttpd-dev \
    ca-certificates \
    wget \
    tar
```

Verify that Ubuntu resolved the expected packages:

```
apt-cache policy \
    libxml2-dev \
    libxml2-16 \
    libmicrohttpd-dev \
    libmicrohttpd12t64
```

4. Download, build, and install FORT 1.7.0.experimental

```
cd /tmp

wget https://github.com/NICMx/FORT-validator/releases/download/1.7.0.experimental/
fort-1.7.0.experimental.tar.gz

tar -xzf fort-1.7.0.experimental.tar.gz
cd fort-1.7.0.experimental

./configure
make -j"$(nproc)"
sudo make install
hash -r
```

Verify the installation:

```
command -v fort
fort --version
```

The expected binary location after make install is:

```
/usr/local/bin/fort
```

5. Verify linked libraries

```
ldd "$(command -v fort)" | grep -E 'xml2|microhttpd|ssl|crypto|curl|jansson'
```

The verified installation included, among others:

```
libjansson.so.4
libcurl.so.4
libxml2.so.16
libmicrohttpd.so.12
libcrypto.so.3
libssl.so.3
```

No library should be reported as not found.

6. Create the service account and directories

```
getent passwd fort || sudo useradd \
  --system \
  --home-dir /var/lib/fort \
  --create-home \
  --shell /usr/sbin/nologin \
  fort
sudo install -d -o root -g fort -m 0750 /etc/fort
sudo install -d -o fort -g fort -m 0750 /etc/fort/tal
sudo install -d -o fort -g fort -m 0750 /var/lib/fort/repository
sudo install -d -o fort -g fort -m 0750 /var/lib/fort/output
```

7. Initialize the TAL files

```
sudo -u fort /usr/local/bin/fort \
  --init-tals \
  --tal /etc/fort/tal
sudo find /etc/fort/tal \
  -maxdepth 1 \
  -type f \
  -name '*.tal' \
  -print
```

8. Run the first manual validation

```
sudo -u fort /usr/local/bin/fort \
  --mode standalone \
  --tal /etc/fort/tal \
  --local-repository /var/lib/fort/repository \
  --output.roa /var/lib/fort/output/validated-roas.csv
```

The first run can use both available vCPUs and download several gigabytes. This is expected. During the verified test, the repository reached approximately 2.7 GB during initial synchronization.

Recommended monitoring from another session:

```
pgrep -a fort
ps -o pid,stat,etime,%cpu,%mem,cmd -p "$(pgrep -n fort)"
sudo ss -tpn | grep fort
sudo du -sh /var/lib/fort/repository /var/lib/fort/output
vmstat 1
```

CPU usage above 100% means that the process is using more than one core. With 2 vCPUs, the theoretical maximum is approximately 200%.

9. Verify the validation result

```
echo $?
sudo ls -lh /var/lib/fort/output/validated-roas.csv
sudo head /var/lib/fort/output/validated-roas.csv
sudo wc -l /var/lib/fort/output/validated-roas.csv
sudo stat /var/lib/fort/output/validated-roas.csv
```

Observed result from the verified validation run:

```
Exit status: 0
File size: approximately 24 MB
Total lines: 887046
Header: ASN,Prefix,Max prefix length
```

10. Create the permanent configuration

Create the file:

```
sudo nano /etc/fort/config.json
```

Verified final content:

```
{
  "mode": "server",
  "tal": "/etc/fort/tal",
  "local-repository": "/var/lib/fort/repository",

  "output": {
    "roa": "/var/lib/fort/output/validated-roas.csv"
  },

  "server": {
    "address": [
      "127.0.0.1",
      "::1",
      "10.0.1.15"
    ],
    "port": "8323",
    "max-rtr-version": 1,
    "interval": {
      "validation": 3600,
      "refresh": 3600,
      "retry": 600,
      "expire": 7200
    }
  },

  "log": {
    "output": "console",
    "level": "info"
  }
}
```

Replace 10.0.1.15 if the VM uses a different private address.

```
sudo chown root:fort /etc/fort/config.json
sudo chmod 0640 /etc/fort/config.json
```

Validate the JSON syntax and access permissions:

```
sudo python3 -m json.tool /etc/fort/config.json > /dev/null && \
  echo "JSON syntax is valid."

sudo -u fort test -r /etc/fort/config.json && \
  echo "FORT user can read the configuration."
```

11. Test the configuration outside systemd

```
sudo -u fort /usr/local/bin/fort \
  --configuration-file /etc/fort/config.json
```

The output should confirm at least the following values:

FORT 1.7.0.experimental on Ubuntu Server 26.04 - Verified guide

```
mode: server
server.address:
  127.0.0.1
  ::1
  10.0.1.15
server.port: 8323
server.max-rtr-version: 1
output.roa: /var/lib/fort/output/validated-roas.csv
```

It should also report that all configured sockets were opened successfully. Stop this foreground test with Ctrl+C before starting the systemd service.

12. Create the systemd service

Create the service unit file:

```
sudo nano /etc/systemd/system/fort.service
```

Place the following content in the file:

```
[Unit]
Description=FORT RPKI Validator
Documentation=https://nicmx.github.io/FORT-validator/
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=fort
Group=fort
Environment="MALLOC_ARENA_MAX=2"
ExecStart=/usr/local/bin/fort --configuration-file /etc/fort/config.json
Restart=on-failure
RestartSec=10
NoNewPrivileges=true
PrivateTmp=true
ProtectHome=true

[Install]
WantedBy=multi-user.target
```

Save the file, then reload systemd and enable the service:

```
sudo systemctl daemon-reload
sudo systemctl enable --now fort
```

Check the service status and recent logs:

```
sudo systemctl status fort
sudo journalctl -u fort -n 100 --no-pager
```

13. Verify RTR and local-network access

```
sudo ss -lntp | grep ':8323'
```

Listeners are expected on localhost and on the VM private address:

```
127.0.0.1:8323
[::1]:8323
10.0.1.15:8323
```

From another host or router on the same network:

```
nc -vz 10.0.1.15 8323
```

When the router connects, verify the established session on the validator:

```
sudo ss -tnp | grep ':8323'
```

Configure the following values on the router:

```
RTR server: 10.0.1.15
RTR port: 8323
RTR version: 1 or automatic negotiation
```

14. Firewall considerations

UFW was disabled in the verified installation. Filtering may still exist at the Proxmox datacenter, node, or VM level, on the router, or between VLANs and subnets.

```
sudo ufw status
sudo nft list ruleset
sudo iptables -S
```

If UFW is enabled, preferably allow only the router address:

```
sudo ufw allow from ROUTER_IP_ADDRESS to 10.0.1.15 port 8323 proto tcp
```

Access from another routed network requires forward and return routes. If access crosses the public Internet or NAT, use a VPN rather than exposing RTR directly.

15. Time and synchronization

RPKI validation depends on certificate, manifest, and CRL timestamps. The VM clock must remain synchronized.

```
timedatectl status
timedatectl timesync-status
```

Expected values:

```
System clock synchronized: yes
NTP service: active
```

UTC is recommended for infrastructure servers:

```
sudo timedatectl set-timezone UTC
```

To display Uruguay local time without changing the server time zone:

```
TZ=America/Montevideo date
```

16. Expected resource usage and behavior

- The first synchronization is the most expensive; later cycles reuse the local repository.
- With 2 vCPUs, FORT can approach 200% CPU during intensive processing stages.
- 4 GB of RAM was sufficient in the verified test; only a small amount of swap was observed, without sustained memory pressure.
- A brief period with zero CPU and I/O immediately before completion can be normal.
- Do not interrupt the first validation while CPU, network activity, or repository growth continues.

17. Troubleshooting the issues encountered

The .deb package cannot be installed

```
Unsatisfied dependencies:
fort : Depends: libmicrohttpd12
        Depends: libxml2
```

Resolution: build from the source tarball against Ubuntu 26.04 libmicrohttpd-dev and libxml2-dev.

The service repeatedly auto-restarts

```
unable to open /etc/fort/config.json: No such file or directory
```

Resolution: create /etc/fort/config.json before starting fort.service.

Permission denied while binding [::]:323

```
[::]:323: Unable to bind the socket: Permission denied
```

Cause: FORT used the privileged default port 323. Verified resolution: use the nested JSON structure and set `server.port` to 8323.

Python cannot read config.json

```
PermissionError: [Errno 13] Permission denied: '/etc/fort/config.json'
```

Cause: the file permissions are 0640 and ownership is root:fort. Validate it with `sudo` or as the fort user:

```
sudo python3 -m json.tool /etc/fort/config.json > /dev/null
```

18. Final checklist

- `fort --version` reports 1.7.0.experimental.
- `ldd` reports no missing libraries.
- TAL files exist under `/etc/fort/tal`.
- The local repository exists and is owned by fort:fort.
- `validated-roas.csv` exists and contains data.
- `/etc/fort/config.json` is valid JSON and readable by the fort account.
- `fort.service` is active (running) and enabled.
- FORT listens on 10.0.1.15:8323.
- The router can establish TCP connectivity to 10.0.1.15:8323.
- The system clock is synchronized and the server time zone is UTC.