

Argus

Compte les attaques contre vos serveurs et en garde l'historique en graphiques, sans service permanent ni base de données

Manuel

Version	1.0.0-14
Commit	`9622e76`
Date du commit	3 novembre 2026
Licence	MIT

Manuel hors ligne, identique aux guides du dépôt à la version et au commit indiqués. Le texte de référence est celui du dépôt.

Table des matières

Présentation **1**
- Installation en deux temps . . . 1
- Les trois vues . . . 1
- Une connexion par hôte, pas de nouvelle tentative . . . 2
- Configuration . . . 2
- Vérifier que cela fonctionne . . . 2
- Documentation . . . 2
- Crédits . . . 3
- Licence . . . 3

Installer Argus sur le serveur **4**
- Ce que vous obtenez . . . 4
- Prérequis . . . 4
- Installation . . . 4
- Essayer tout de suite . . . 4
- Changer un réglage . . . 5
- Désinstallation . . . 5

Copier les graphiques sur votre poste de bureau et les lire **6**
- Ce que vous obtenez . . . 6
- Prérequis . . . 6
- Déclarer les serveurs . . . 6
- Copie . . . 6
- Une connexion, pas de nouvelle tentative . . . 6
- Lire les graphiques . . . 7

Graphiques : fenêtres, découpage, heure, séries **8**
- Fenêtres . . . 8
- Découper une longue fenêtre . . . 8
- L'heure de la journée . . . 8
- Séries . . . 9
- Lire les nombres au lieu de l'image . . . 9
- Tranches à l'heure 00 (`T00`) . . . 9
- Langue des graphiques . . . 9

Dépannage **10**
- Lancer le contrôle . . . 10
- Ce que signifie chaque ligne . . . 10
- Lancer un collecteur à la main . . . 10
- Causes courantes . . . 10

Présentation

Pas de service permanent, pas de base de données, adapté aux machines de moins de 1 Go de RAM.

Argus compte les attaques contre vos serveurs, SSH de lui-même et, lorsqu'ils sont installés, les bannissements de Fail2Ban et de CrowdSec, et conserve l'historique sous forme de graphiques. Une tâche cron s'exécute chaque nuit, écrit des fichiers texte plats et dessine les graphiques. Rien ne tourne entre deux nuits.

Argus ne stocke ni n'affiche jamais l'adresse IP d'un attaquant, seulement des comptes par heure et par jour. Une **attaque** est une IP source qui atteint un nombre minimal de tentatives d'authentification échouées en une journée (5 par défaut), afin qu'un mot de passe mal saisi ne compte pas.

Grafana est une vue continue d'une infrastructure : un service qui tourne en permanence et peut être interrogé en direct. Argus est un instantané périodique de votre historique de sécurité : une image par nuit, lue quand vous le souhaitez. Grafana peut aussi agréger par semaine ou par année, au prix d'un service permanent, d'une base de données de séries temporelles et d'un tableau de bord à construire ; Argus n'a aucun des trois, ce qui lui permet de tenir sur un serveur de 1 Go.

Installation en deux temps

Si vous installez habituellement un programme sur l'ordinateur avec lequel vous travaillez, ici c'est l'inverse. Le paquet va sur le **serveur**, la machine attaquée. Votre propre ordinateur, le poste de bureau, n'installe rien du paquet et ne reçoit que les images.

1. **Sur le serveur** (la machine attaquée), installez le paquet. Il compte les attaques et dessine les graphiques chaque nuit. Il n'y a rien à lancer ensuite.
2. **Sur votre poste de bureau** (l'ordinateur depuis lequel vous lisez), lancez `argus-fetch`. Il copie les images des graphiques depuis vos serveurs vers votre ordinateur.

Où se trouve chaque partie :

```
server.example.org (the server)          your desk
─────────────────────────────────        ──────────────────────────────
/opt/argus/        scripts               ~/.config/argus/hosts.conf
/etc/argus/        argus.conf                the list of your servers
/srv/argus/<year>/ data and images       ~/argus-backup/<server>/<year>/images/
/usr/bin/argus-chart                         the copied charts
/usr/bin/argus-fetch   ── copies images ──►
```

Étape 1, sur le serveur (Debian ou Ubuntu) :

```
sudo apt install ./argus_1.0.0-14_all.deb
```

Étape 2, sur votre poste de bureau. Placez vos serveurs dans le fichier `~/.config/argus/hosts.conf`, un par ligne, sous la forme `<ssh alias>,<data directory>`, puis lancez `argus-fetch` :

```
mkdir -p ~/.config/argus
printf 'server.example.org,/srv/argus/\n' > ~/.config/argus/hosts.conf
argus-fetch
```

`argus-fetch` est installé par le même paquet. Sur un poste qui n'est pas une machine Debian, copiez le script unique `argus-fetch` depuis ce dépôt ; il ne lui faut que `bash`, `ssh` et `rsync`. Détails : `docs/install.md` et `docs/desk.md`.

Les trois vues

`argus-chart` dessine un graphique pour la fenêtre que vous choisissez. Lancez-le sur le serveur. Chaque commande écrit un PNG dans `/srv/argus/<year>/images/` (ajoutez `--out DIR` pour écrire ailleurs, ce dont un utilisateur ordinaire a besoin).

- `argus-chart daily` est l'historique de l'année, un point par jour.
- `argus-chart period` est le compte de chaque série sur une fenêtre.
- `argus-chart hourly` est le compte par heure de la journée sur une fenêtre, de 00 à 23.

Dix fenêtres à copier :

```
argus-chart period --day 2026-10-30                     # one day, one point per hour
argus-chart period --week 2026-W44                      # ISO week, Monday to Sunday
argus-chart period --month 2026-10                      # one calendar month
argus-chart period --quarter 2026-Q4                    # one quarter
argus-chart period --year 2025                          # one year
argus-chart period --from 2025-12-01 --to 2026-01-31    # a range over two years
argus-chart period --month 2026-10 --split week         # one PNG per week of the month
argus-chart period --week 2026-W44 --series ssh_direct  # one series only
argus-chart hourly --week 2026-W44                      # which hours are attacked
argus-chart hourly --month 2026-10 --series ssh_direct,ip_ban_sshd
```

Les séries sont dessinées côte à côte, jamais additionnées : SSH, chaque jail Fail2Ban et chaque scénario CrowdSec gardent leur propre ligne et leur propre couleur. Détails : `docs/charts.md`.

Une connexion par hôte, pas de nouvelle tentative

`argus-fetch` ouvre **une connexion par hôte** et **ne réessaie jamais**. Beaucoup de serveurs bannissent une adresse après quelques tentatives de connexion échouées, si bien qu'une seconde tentative pourrait vous verrouiller dehors. Un hôte qui ne répond pas est signalé une fois, les autres hôtes sont quand même copiés, et le code de sortie est 1. Corrigez la cause (clé, alias, pare-feu, machine éteinte), puis relancez la commande à la main.

Configuration

Tout fonctionne avec les valeurs par défaut. Pour en changer une, modifiez `/etc/argus/argus.conf` sur le serveur ; le fichier est lu par chaque script et livré avec chaque valeur commentée :

```
# /etc/argus/argus.conf
ARGUS_SSH_THRESHOLD=5          # failed attempts of one IP in one day that make an attack
ARGUS_LANG=en                  # chart language: en, fr, de or es
ARGUS_DATA_DIR="/srv/argus"    # where the data and the images are written
```

Les autres variables (chemins de journaux, taille des images, couleurs des séries) sont listées dans `man argus`.

Vérifier que cela fonctionne

```
sudo /opt/argus/argus-doctor.sh
```

Il indique si chaque source (SSH, Fail2Ban, CrowdSec) a été collectée la veille. Voir `docs/troubleshooting.md`.

Documentation

- `docs/install.md` : installer sur le serveur, arborescence, désinstallation
- `docs/desk.md` : copier les graphiques sur votre poste de bureau et les lire
- `docs/charts.md` : fenêtres, découpage, heure, séries, fuseau horaire, tranches `T00`
- `docs/troubleshooting.md` : ce que dit `argus-doctor.sh` et que faire
- Pages de manuel : `man argus`, `man argus-chart`, `man argus-fetch` (sources dans `man/en/`)

Crédits

Argus est le fruit d'une collaboration entre un humain et des modèles de langage, avec les rôles consignés tels qu'ils ont réellement été tenus. La liste complète est dans `CONTRIBUTORS.md`.

- **Ronan Davalan** — Architecte et arbitre : scripts d'origine (2024), décisions, validation et essais.
- **Claude Code (Anthropic)** — Développeur principal : collecteurs, rendu des graphiques, `argus-doctor.sh`, paquet Debian, documentation.
- **Gemini (Google)** — Synthétiseur et co-rédacteur stratégique : planification et cadrage de la raison d'être du projet.
- **Gemini (Google), GLM-5.3-Flash Free et Gemini 4 Argon** — Graphistes : identité visuelle du projet (marque, icône, favicon), adoptée par Ronan Davalan le 2026-11-02.
- **DeepSeek** — Compagnon de terrain : installation de CrowdSec sur un serveur réel ayant servi à valider `collect-crowdsec.sh`.
- **DeepSeek Flash** — Contenu du site en anglais, puis traductions française, allemande et espagnole de la documentation, des pages de manuel et du site.
- **Muse Spark 1.3 Contributor** — Habillage du site : intégration du logo, du favicon et de l'image du graphe.
- **gd2 :** frère défunt de Ronan Davalan. Il a appris Git à Ronan et son usage, conseillé sur ce projet, et fait la première modification extérieure à son README (2024-08-07).

Licence

MIT — voir `LICENSE`.

Installer Argus sur le serveur

Le paquet va sur le **serveur**, la machine dont Argus lit les journaux. Vous ne l'installez pas sur votre poste de bureau : le poste de bureau ne fait que copier les graphiques (voir `desk.md`).

Ce que vous obtenez

```
/opt/argus/
├── collect-ssh.sh         SSH attacks, from auth.log or journald
├── collect-fail2ban.sh    Fail2Ban bans, one series per jail (optional)
├── collect-crowdsec.sh    CrowdSec alerts, one series per scenario (optional)
├── chart-daily.sh         history of the year, one point per day
├── chart-period.sh        count over a window (day, week, month, quarter, year, range)
├── chart-hourly.sh        count per hour of the day over a window
├── chart-*.gp             the gnuplot drawing of each view
├── argus-lib.sh           code shared by the chart scripts
└── argus-doctor.sh        health check of the collection

/usr/bin/argus-chart       one command for the three views
/usr/bin/argus-fetch       copies the charts to a desk
/etc/argus/argus.conf      optional settings, every line commented out
/etc/cron.d/argus          the nightly jobs
/srv/argus/                data and charts, readable by everyone
└── 2026/
    ├── ssh_direct_2026.txt      one line per hour with an attack
    ├── ip_ban_sshd_2026.txt     one file per Fail2Ban jail, if present
    ├── .daily/                  daily totals, rebuilt by chart-daily.sh
    ├── .heartbeat/              daily proof that each collector ran
    └── images/                  the PNG charts
```

`/srv/argus/` ne contient que des comptes, jamais une adresse IP, il est donc lisible par tous les utilisateurs : vous copiez les graphiques sans `sudo`.

Prérequis

Un serveur Debian ou Ubuntu avec cron. Le paquet dépend de `gawk` et de `gnuplot-nox` ; `apt` les installe. `rsync` et `openssh-client` sont recommandés (ils servent à `argus-fetch`). Fail2Ban, CrowdSec et `jq` sont optionnels. Un serveur de 1 Go de RAM suffit : rien ne reste en cours d'exécution.

Installation

Téléchargez `argus_1.0.0-14_all.deb` depuis la page des versions du dépôt, puis, sur le serveur :

```
sudo apt install ./argus_1.0.0-14_all.deb
```

Les tâches cron démarrent cette nuit-là : collecte à 00:05, graphiques à 00:15, contrôle de santé à 00:20. Le premier graphique apparaît dans `/srv/argus/2026/images/` après la première nuit.

Essayer tout de suite

Pour sauter la première nuit, exécutez les étapes à la main sur un serveur qui possède déjà un historique de journaux :

```
sudo /opt/argus/collect-ssh.sh
sudo /opt/argus/chart-daily.sh
ls /srv/argus/2026/images/
```

Dessinez ensuite une vue :

```
sudo argus-chart period --month 2026-10
```

Changer un réglage

`/etc/argus/argus.conf` est un fichier shell lu par chaque script. Décommentez et modifiez seulement ce dont vous avez besoin :

```
# /etc/argus/argus.conf
ARGUS_SSH_THRESHOLD=10         # failed attempts of one IP in one day that make an attack
ARGUS_LANG=en                  # chart language: en, fr, de or es
ARGUS_IMG_WIDTH=3840           # image size in pixels
ARGUS_IMG_HEIGHT=2160
ARGUS_COULEUR_ssh_direct=orange   # colour of one series, any gnuplot colour name
```

Les noms de variables utilisent des majuscules et des traits de soulignement, jamais des traits d'union. La série `caddy-xmlrpc` est donc colorée par `ARGUS_COULEUR_caddy_xmlrpc`. Toutes les variables sont listées dans `man argus`.

Désinstallation

```
sudo apt remove argus     # removes the scripts, keeps /srv/argus and /etc/argus
sudo apt purge argus      # also removes /srv/argus and /etc/argus
```

La purge supprime aussi votre historique. Copiez-le d'abord avec `argus-fetch` (voir `desk.md`).

Copier les graphiques sur votre poste de bureau et les lire

Argus tourne sur vos serveurs et y dessine les graphiques. `argus-fetch` copie les images sur l'ordinateur avec lequel vous travaillez. Rien n'est envoyé depuis le serveur : c'est vous qui venez chercher les images, quand vous le souhaitez.

Ce que vous obtenez

```
~/.config/argus/hosts.conf              the servers to copy, one per line
~/argus-backup/
└── server.example.org/                 one folder per ssh alias
    └── 2026/
        └── images/
            ├── 2026-11-02_argus_4k.png
            └── 2026-11-02_argus_period_week-2026-W44.png
```

Prérequis

`bash`, `ssh` et `rsync` sur le poste de bureau. Le paquet installe `argus-fetch` dans `/usr/bin` ; sur un autre système, copiez le fichier unique `argus-fetch` depuis le dépôt vers un dossier de votre `PATH`. Votre `ssh` doit déjà se connecter au serveur avec une clé, sans demande de mot de passe : `argus-fetch` n'en demande jamais.

Déclarer les serveurs

Un serveur par ligne, `<ssh alias>,<data directory>`. L'alias est un nom de `Host` de `~/.ssh/config` (ou `user@host`) ; le répertoire est `ARGUS_DATA_DIR` sur ce serveur, `/srv/argus/` par défaut.

```
mkdir -p ~/.config/argus
printf 'server.example.org,/srv/argus/\nbackup.example.org,/srv/argus/\n' > ~/.config/argus/hosts.conf
cat ~/.config/argus/hosts.conf
```

Les lignes vides et les lignes commençant par `#` sont ignorées. Le dépôt fournit `hosts.conf.example`.

Copie

```
argus-fetch
```

Pour chaque serveur, il ouvre **une** connexion, copie les fichiers PNG de `<directory>/<year>/images/` , puis la ferme. Un fichier déjà présent sur votre poste de bureau n'est jamais écrasé et rien n'est supprimé. Lancez-le aussi souvent que vous le souhaitez : seules les nouvelles images sont copiées.

Pour ne copier qu'un serveur, ou placer les copies ailleurs :

```
argus-fetch --only server.example.org
argus-fetch --dest ~/Documents/argus
```

Une connexion, pas de nouvelle tentative

Si un serveur ne répond pas, `argus-fetch` le signale une fois sur la sortie d'erreur, continue avec les autres serveurs et se termine avec le code de sortie 1. Il ne réessaie jamais, car beaucoup de serveurs

bannissent une adresse après quelques tentatives de connexion échouées. Trouvez la cause (clé, alias, pare-feu, serveur éteint), puis relancez la commande vous-même.

Lire les graphiques

Ouvrez les images avec n'importe quelle visionneuse :

```
xdg-open ~/argus-backup/server.example.org/2026/images/
```

Un graphique montre une ligne par série. Sur une fenêtre d'au plus 31 points, le compte exact est écrit à chaque point. Une vue qui n'existe pas encore (une autre semaine, un mois par heure) se dessine sur le serveur, en root pour que l'image arrive dans le dossier des images :

```
ssh server.example.org
sudo argus-chart period --week 2026-W44
```

Le prochain `argus-fetch` copie la nouvelle image avec les autres. Voir `charts.md` pour les fenêtres que vous pouvez demander.

Graphiques : fenêtres, découpage, heure, séries

`argus-chart` a trois vues. Lancez-le sur le serveur, en root pour écrire dans `/srv/argus/<year>/images/`, ou en tant qu'utilisateur ordinaire avec `--out DIR`.

```
argus-chart daily  [YEAR]              one point per day, the whole year
argus-chart period [window] [options]  one line per series over a window
argus-chart hourly [window] [options]  24 positions, 00 to 23, per series
```

Chaque fichier est nommé `<today>_argus_<view>_<window>.png`, par exemple `2026-11-02_argus_period_week-2026-W44.png`.

Fenêtres

`period` et `hourly` acceptent les mêmes fenêtres. Sans fenêtre, ils utilisent l'année en cours.

Option	Fenêtre	Exemple
`--day YYYY-MM-DD`	un jour	`--day 2026-10-30`
`--week YYYY-Www`	semaine ISO, du lundi au dimanche	`--week 2026-W44`
`--month YYYY-MM`	mois calendaire	`--month 2026-10`
`--quarter YYYY-Qn`	trimestre, `n` de 1 à 4	`--quarter 2026-Q4`
`--year YYYY`	année calendaire	`--year 2025`
`--from D --to D`	plage, les deux jours inclus	`--from 2025-12-01 --to 2026-01-31`

Les semaines suivent la norme ISO 8601 : la semaine 1 de 2027 commence le lundi 2027-01-04, et 2026 compte une semaine 53. Une fenêtre qui couvre deux années lit les données des deux.

```
argus-chart period --day 2026-10-30 --out ~/argus-out
argus-chart period --from 2025-12-01 --to 2026-01-31 --out ~/argus-out
```

Découper une longue fenêtre

`--split` dessine un graphique par sous-période au lieu d'un seul long graphique :

```
argus-chart period --month 2026-10 --split week --out ~/argus-out
argus-chart period --year 2026 --split month --out ~/argus-out
```

`--split week` convient à un mois ou à un trimestre, `--split month` à un trimestre ou à une année. La première commande écrit un fichier par semaine ISO touchant octobre, chacun découpé à la fenêtre, nommé `…_month-2026-10_week-2026-W41.png`.

L'heure de la journée

`hourly` additionne, pour chaque série, les attaques vues à chaque heure sur la fenêtre, et dessine 24 positions. Il montre à quelles heures de la journée vous êtes attaqué, et non quels jours.

```
argus-chart hourly --week 2026-W44 --out ~/argus-out
```

L'heure est l'**heure locale du serveur**, telle que l'écrivent les collecteurs. C'est l'heure à laquelle l'attaque a été vue dans les journaux, non l'heure à laquelle elle a commencé. Le titre indique « local server time ». Un changement d'heure d'été ajoute ou retire une heure sur une journée ; sur une semaine ou plus, l'effet ne se voit pas.

Séries

Une série est une ligne : `ssh_direct` pour SSH, `ip_ban_<jail>` pour chaque jail Fail2Ban, `ip_ban_<scenario>` pour chaque scénario CrowdSec. Le nom est celui du fichier de données sans l'année, donc `ssh_direct_2026.txt` est la série `ssh_direct`.

Les séries sont superposées, jamais additionnées : Argus n'invente pas un score à partir d'outils différents. Chaque série garde la même couleur d'une année à l'autre (définissez-la avec `ARGUS_COULEUR_<series>`, sinon elle vient d'un hachage du nom).

```
argus-chart period --week 2026-W44 --list-series
argus-chart period --week 2026-W44 --series ssh_direct
argus-chart hourly --month 2026-10 --series ssh_direct,ip_ban_sshd --out ~/argus-out
```

Un nom inconnu dans `--series` arrête la commande avec le code de sortie 2 et liste les noms disponibles.

Lire les nombres au lieu de l'image

`--print-data` imprime sur la sortie standard le tableau qui sous-tend le graphique (séparé par des tabulations, en-tête commençant par #) et ne dessine rien :

```
argus-chart period --week 2026-W44 --print-data
argus-chart hourly --day 2026-10-30 --print-data
```

Tranches à l'heure 00 (`T00`)

Une ligne de données est `<count> <date>T<hour>`, par exemple `492 2026-10-30T14`. Les fichiers antérieurs à la collecte horaire contiennent une ligne par jour, toutes écrites en `T00`. Une série composée uniquement de tranches `T00` n'a aucune information d'heure, si bien que `hourly` l'écarte au lieu de dessiner une barre à minuit. Il imprime le message `series 'legacy' excluded: all its slices are at hour 00 (daily data, no hourly depth)` sur la sortie d'erreur et écrit `excluded series: no hourly data: legacy` dans le titre du graphique. `period` et `daily` le dessinent encore, comme points quotidiens.

Langue des graphiques

`ARGUS_LANG` définit les titres et les axes : `en` (par défaut), `fr`, `de` ou `es`.

```
ARGUS_LANG=de argus-chart period --week 2026-W44 --out ~/argus-out
```

Dépannage

Un graphique sans attaque peut signifier une journée calme ou un collecteur qui s'est arrêté. Les fichiers de données ne permettent pas de les distinguer, car un jour sans attaque n'écrit aucune ligne. `argus-doctor.sh` le peut : chaque collecteur écrit une ligne par jour dans `/srv/argus/<year>/.heartbeat/<source>.txt`, `<date> <number of events>`, même lorsque le nombre vaut 0.

Lancer le contrôle

```
sudo /opt/argus/argus-doctor.sh
```

La tâche cron l'exécute chaque nuit à 00:20 et ajoute la sortie à `/var/log/argus.log`. Le code de sortie est 0 lorsqu'il n'y a rien à signaler, 1 sinon. Un serveur sain affiche :

```
== Source availability ==

== Heartbeats (2026) ==
  ssh : ok
    -> up to date, yesterday: 12 event(s)
  fail2ban : absent
  crowdsec : absent

Nothing to report.
```

Ce que signifie chaque ligne

`<source> : ok` ou `absent` indique si la source peut être lue maintenant : `auth.log` ou `journald` pour `ssh`, `fail2ban-client` et son journal pour `fail2ban`, `cscli` et `jq` pour `crowdsec`. `absent` pour Fail2Ban ou CrowdSec est normal lorsque vous ne les exécutez pas.

Ligne sous la source	Signification	Que faire
`up to date, yesterday: N event(s)`	collectée hier, N événements (0 est une journée calme)	rien
`last confirmed on DATE (no entry for yesterday)`	le collecteur ne s'est pas exécuté ou a échoué depuis DATE	regardez `/var/log/argus.log`, lancez le collecteur à la main
`never collected`	aucun fichier de battement de vie, bien que la source soit lisible	lancez le collecteur à la main, vérifiez `/etc/cron.d/argus`

Les lignes qui demandent une action sont répétées sous `TO CHECK` sur la sortie d'erreur.

Lancer un collecteur à la main

```
sudo /opt/argus/collect-ssh.sh
sudo /opt/argus/argus-doctor.sh
tail -n 20 /var/log/argus.log
```

Causes courantes

- **Aucune image après la première nuit.** Vérifiez que `/etc/cron.d/argus` existe et que cron s'exécute (`systemctl status cron`). Lancez `sudo /opt/argus/chart-daily.sh` et lisez son message.

- **`chart-daily.sh: no data file for 2026 in /srv/argus/2026 — nothing to plot.`** Aucun collecteur n'a écrit de fichier cette année. Un serveur sans aucune attaque n'en écrit pas. Lancez `sudo /opt/argus/collect-ssh.sh` pour collecter la veille.
- **SSH affiche 0 événement chaque jour.** `auth.log` est absent et `journald` contient les journaux sous un autre nom d'unité. Définissez `ARGUS_SSHD_UNIT="sshd"` dans `/etc/argus/argus.conf` (la valeur par défaut est `ssh`).
- **Moins d'attaques SSH que de bannissements Fail2Ban.** Argus compte une IP seulement après `ARGUS_SSH_THRESHOLD` tentatives échouées en une journée (5 par défaut), et Fail2Ban peut bannir plus tôt. Abaissez le seuil pour comparer.
- **`argus-chart: unknown view.`** Les vues sont `period`, `hourly` et `daily`.
- **`unknown series.`** Listez les noms avec `argus-chart period --list-series`.
- **`argus-fetch` signale un hôte.** Le serveur n'a pas répondu au premier essai, et `argus-fetch` ne fait pas de nouvelle tentative. Vérifiez l'alias, la clé et le pare-feu, puis relancez-le. Voir `desk.md`.
- **Permission refusée à l'écriture d'une image.** `/srv/argus` appartient à root : utilisez `sudo`, ou `--out DIR` dans votre répertoire personnel.