poste.io-addons



Antonio Sanna
poste.io-addons
Copyright © 2026 Antonio Sanna

poste.io-addons

Úvod

- poste.io-addons 6
  - Funkce 7
    - Externí schránky 7
    - Tmavý motiv 9
    - Kalendáře a kontakty přes ActiveSync 9
    - Logy ActiveSync 9
    - Zálohování a obnova 10
    - Opravy upstreamu 10
  - Rychlý start 11
  - Licence 11

Instalace

- Instalace 12
  - Požadavky 12
  - 1. Sestavení image 12
  - 2. Spuštění 12
  - 3. Stávající instalace: svazek /data 13
  - 4. Kontrola funkčnosti 13
  - Aktualizace Poste.io 13
  - Odinstalace 14

Výběr doplňků

- Výběr doplňků 15
  - Možnost 1: select-addons.sh 15
  - Možnost 2: úprava Dockerfilu 15

- Než ze stávající instalace doplněk vynecháte 15

Externí schránky

- Externí schránky 17
  - Co doplněk dělá 17
  - Jak to funguje 20
    - Rotace logů 21
  - Zabezpečení 21
  - Poznámky a omezení 22

Tmavý motiv

- Tmavý motiv administrace 23
  - Chování 23
  - Jak to funguje 23
  - Našli jste stránku, která zůstává světlá? 24

Kalendáře a kontakty přes ActiveSync

- Kalendáře a kontakty přes ActiveSync 25
  - Co uživatelé dostanou 25
  - Zapnutí kalendářů a kontaktů pro schránku 25
  - Je kolega zaneprázdněn? Sdílení jen pro čtení 25
  - Výchozí kalendář 25
  - Nastavení telefonu 26
  - Když je server DAV vypnutý nebo nefunkční 26
  - Instalace do existující instalace: zařízení se musí znovu synchronizovat 26
  - Odebrání doplňku 26
  - Aktualizace Poste.io 26
  - Omezení 26

- Jak to funguje 27
- Obejité problémy Z-Push 27
- Řešení potíží 27
- Testováno 27

Logy ActiveSync

- Logy ActiveSync v administraci 28
  - Stránka 28
  - Rotace logů 29
  - Stávající instalace 29
  - Soubory 29
  - Odebrání 29

Zálohování a obnova

- Zálohování a obnova 30
  - Co se zálohuje 31
  - Jak to funguje 31
  - Šifrovací klíč 32
  - Cíle 32
    - SMB / Samba 32
    - SFTP 32
    - S3 32
    - Svazek připojený hostitelem (NFS, druhý disk, ...) 33
    - Server REST, jen přidávání (rest-server) 33
    - Ověření 34
  - Úlohy zálohování 34
  - Sledování záloh 34

- Obnova 35
  - Seznam zpráv 35
  - Odstranění záloh úlohy 36
  - Z příkazového řádku 36
- Čtení záloh bez serveru 36
- Znovuvytvoření celého serveru 38
- Osvědčené postupy 39
- Záloha běžícího serveru 39
- Zabezpečení 39
- Pokud je poštovní server kompromitován 40
- Soubory a složky 40
- Řešení problémů 40
- Omezení 40
- Odebrání 41

Opravy upstreamu

- Opravy upstreamu 42
  - 1. Učicí skripty jsou zapisovatelné pro všechny 42
  - 2. Učení z Junk jde do úložiště, které nikdo nečte 42
  - 3. Každá naučená zpráva zůstane v /tmp 42
  - Jsou ještě potřeba? 42
  - Ověření, že učení funguje 43

Testy

- Testy 44
  - Co se spouští 44
  - Jak probíhá běh 44
  - Co se kontroluje 44

- Vlastní spuštění testů 46

poste.io-addons





(https://github.com/atsanna/poste.io-addons/blob/main/LICENSE)



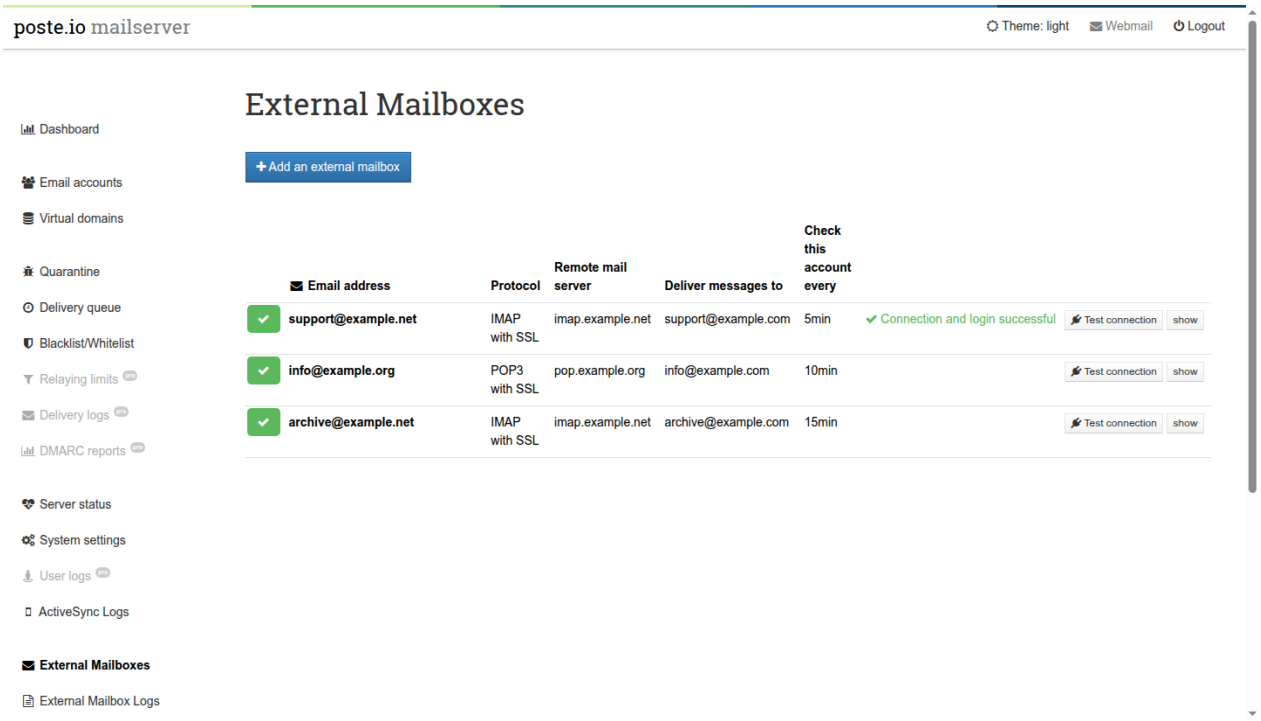
Doplňky pro poštovní server Poste.io *(https://poste.io)*, sestavené jako image Dockeru nad oficiálním image:

- **Externí schránky** — podle plánu stahují poštu z účtů POP3/IMAP hostovaných jinde a doručují ji do Poste.io; spravují se z administrace.
- **Tmavý motiv administrace** — automatický (řídí se systémem), s přepínačem v horní liště.
- **Kalendáře a kontakty přes ActiveSync** — účet Exchange v telefonu dostane kromě pošty i kalendáře a kontakty ze serveru DAV v Poste.io.
- **Logy ActiveSync v administraci** — čtení logů Z-Push s filtry a jejich rotace (Poste.io je nikdy nerotuje).
- **Zálohování a obnova** — šifrované, plánované zálohy konfigurace a pošty na SMB, SFTP, S3 nebo připojený svazek, nastavované a sledované z administrace.
- Do verze 2.5.19 **tři opravy upstreamu**, díky nimž „přesunout do Junk“ rspamd skutečně učí, aniž by v `/tmp` zůstávala kopie každé zprávy — převzato Poste.io ve 2.5.20, doplněk tedy skončil s vydáním 1.2.0.

Vezměte všechny, nebo jen ty, které chcete: každý doplněk je jedna sekce Dockerfilu (výběr doplňků).

V angličtině a češtině, stejně jako administrace Poste.io. Dockerfily pro edici Pro i open source: v produkci na **Poste.io 2.5.24 PRO**; image open source se při každém push sestavuje a celý testuje, se všemi doplňky i s každým zvlášť (testy).

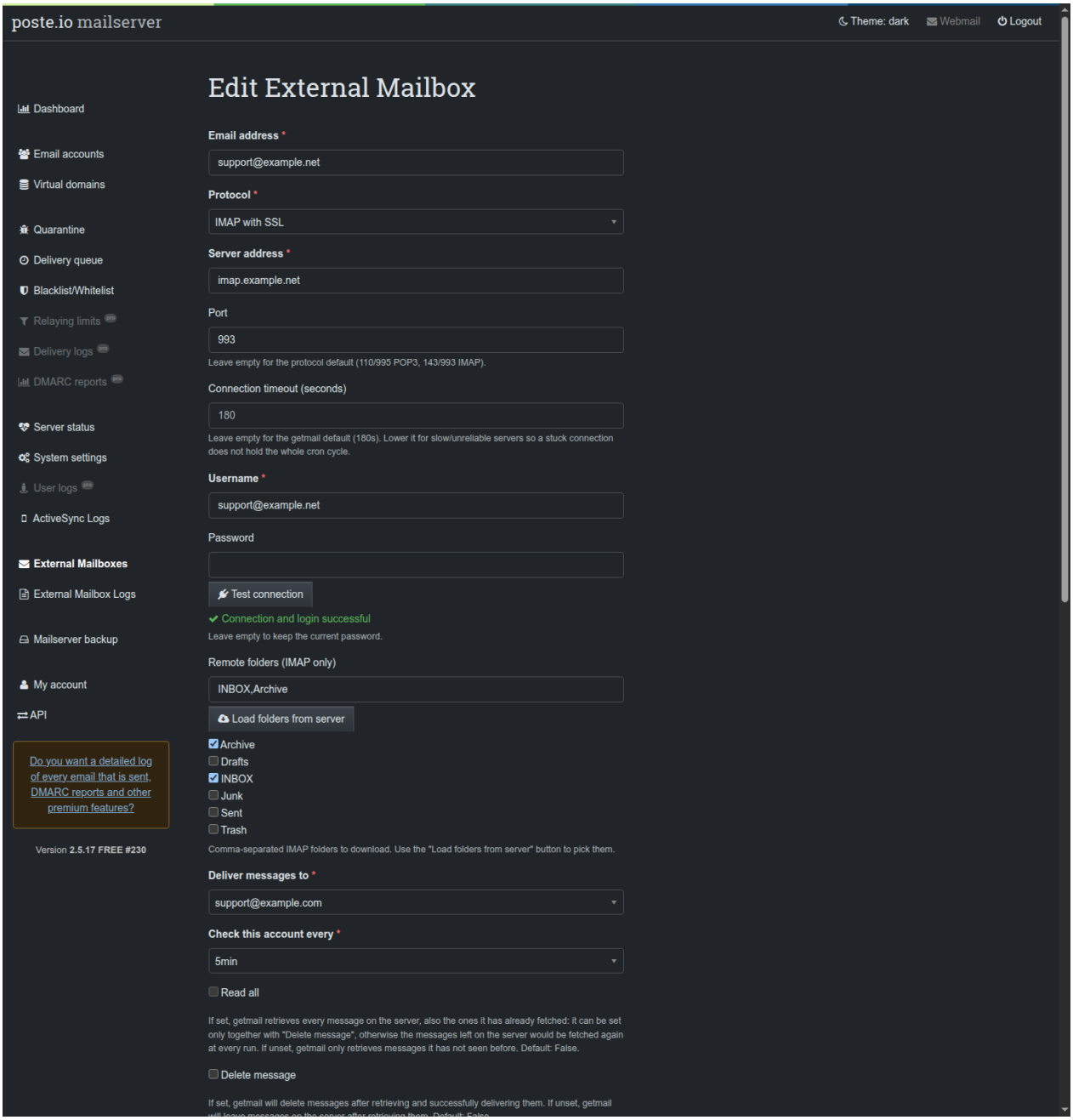
> *Projekt není nijak spojen s Poste.io / Analogic ani jimi podporován. „Poste.io“ je ochranná známka svých vlastníků; tento projekt obsahuje pouze doplňky a žádný kód Poste.io.*



Copyright © 2026 Antonio Sanna

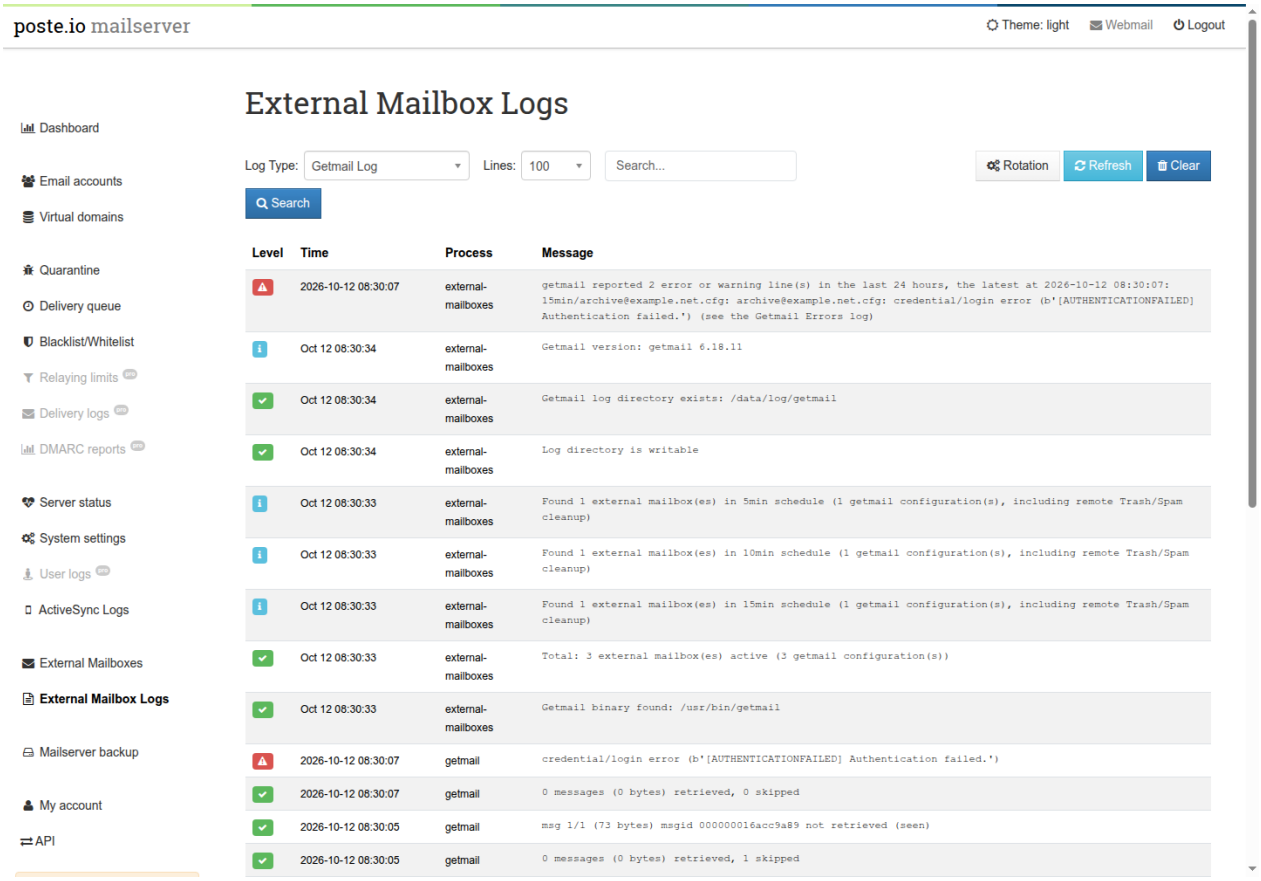
Funkce

Externí schránky



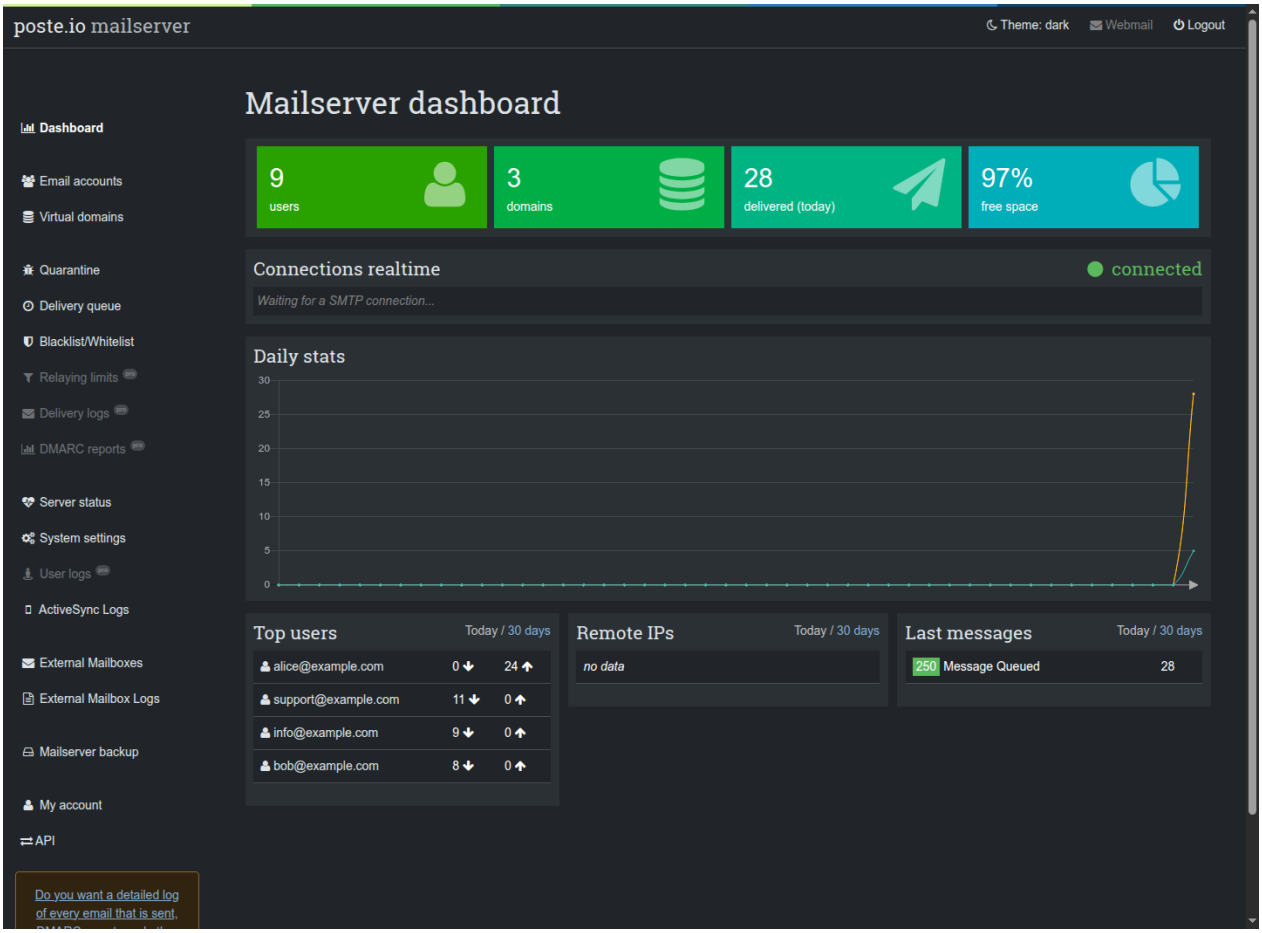
- POP3, POP3 přes SSL, IMAP a IMAP přes SSL, pomocí getmail6 *(https://github.com/getmail6/getmail6)*, každých 5–60 minut.
- Doručení do libovolné místní schránky **nebo přesměrování**: externí schránku `support@` lze doručovat do přesměrování a poštu dostane každý, kdo je v něm.
- Výběr složek IMAP (**Načíst složky ze serveru**) a volitelné čištění vzdálených složek Koš/Spam s dobou uchování.
- **Otestovat připojení** z formuláře i z každého řádku seznamu.
- Volby getmailu ve formuláři: stahovat vše/nové, mazat, mazat po N dnech, limity na jeden běh, mazání zpráv nad určitou velikost, port, časový limit. „Stahovat vše“ jen spolu s „Mazat zprávy“ (jinak by se vše, co zůstalo na serveru, stahovalo znovu při každém běhu).
- **Kontrola spamu**: každá stažená zpráva projde před doručením přes rspamd, stejně jako pošta přijatá přes SMTP (hlavičky, spam do Junk, Bayes); bez ní Poste.io staženou poštu vůbec nekontroluje.
- Stránka logů s logem getmailu, **chybami getmailu** (s červeným řádkem nad zobrazením getmailu, pokud jsou nějaké nedávné), logem cronu a přehledem stavu; **rotace logů** se nastavuje na téže stránce.
- Zabezpečení: getmail běží jako `mail`, nikdy jako root; konfigurační soubory `0600`; certifikáty TLS se ověřují; hesla se nikdy neposílají zpět do prohlížeče; explicitní tokeny CSRF; přísná validace každé hodnoty zapisované pro getmail.

Copyright © 2026 Antonio Sanna



Copyright © 2026 Antonio Sanna

Tmavý motiv



- Řídí se nastavením systému, nebo lze světlý/tmavý vynutit z horní lišty (volbu si pamatuje prohlížeč).
- Žádné bílé probliknutí při načítání; přihlašovací stránka se také řídí systémem.
- Grafy se při změně motivu přebarví a znovu vykreslí.
- Světlý motiv zůstává nedotčen: každé pravidlo platí jen pro tmavý motiv.

→ Tmavý motiv

Kalendáře a kontakty přes ActiveSync

- Jeden účet Exchange/ActiveSync v telefonu: pošta, kalendáře, úkoly a kontakty, oběma směry, z vestavěného serveru CalDAV/CardDAV v Poste.io.
- Zapíná se pro každou schránku v administraci, jako dosud (*Create calendar*, *Create address book*); schránka bez nich má dál jen poštu.
- Kalendář kolegy sdílený **jen pro čtení** se zobrazí jako další kalendář: před domluvením schůzky uvidíte, zda je dotyčný obsazen.
- Pokud je server DAV vypnutý nebo nefunkční, pošta funguje dál.
- Opravuje tři problémy Z-Push nalezené při testování (událost odmítnutá serverem potvrzená jako uložená, pád u kontaktů bez adresáře, ActiveSync nefunkční spolu se serverem DAV).

→ Kalendáře a kontakty přes ActiveSync (stávající instalace: zařízení se musí jednou znovu synchronizovat)

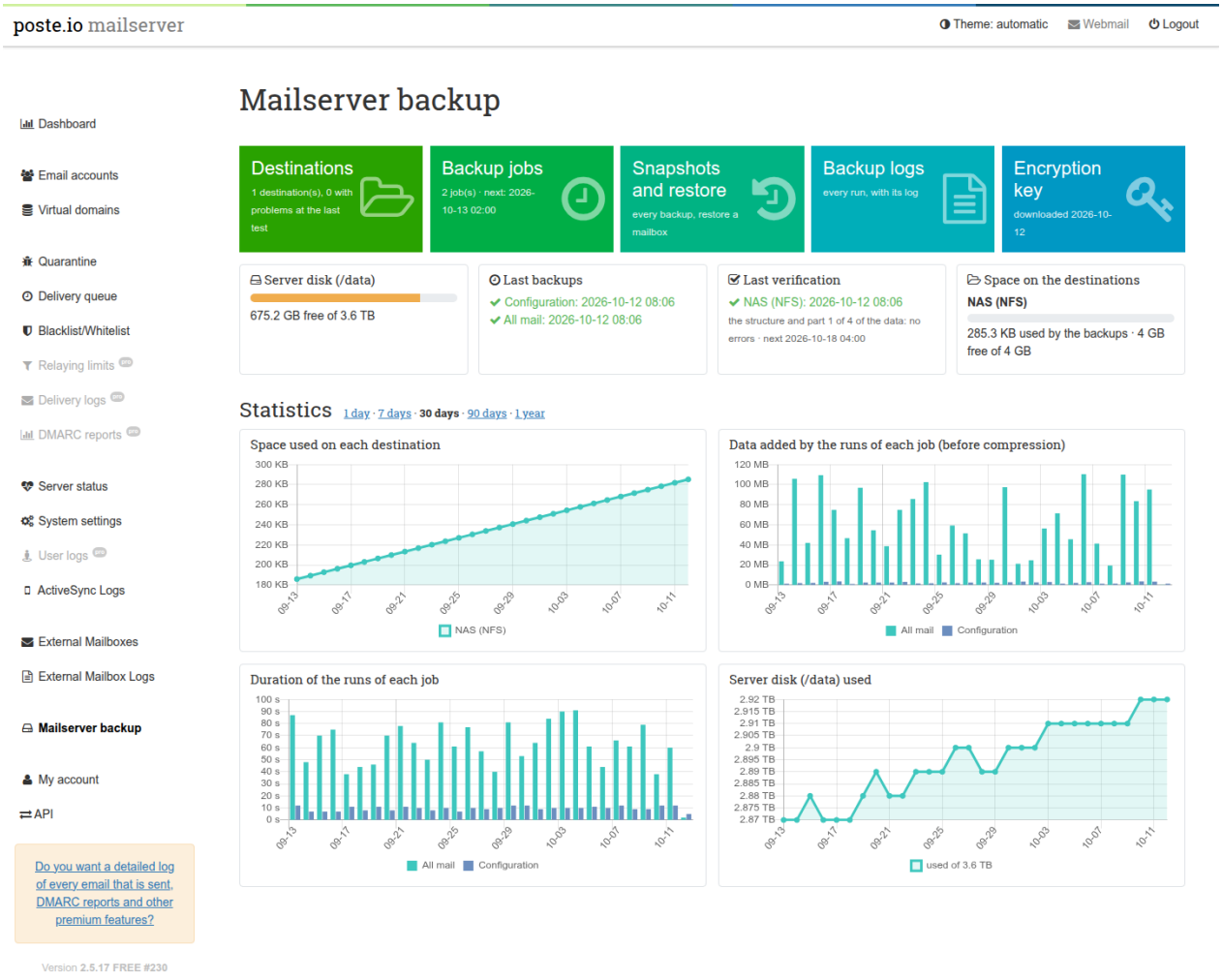
Logy ActiveSync

- Stránka hned za *User logs* s logem Z-Push a jeho logem chyb: filtr podle úrovně, účtu a textu, nejnovější řádky nahoře.
- Rotace logů se nastavuje na téže stránce (Poste.io tyto logy nikdy nerotuje): každý den, týden nebo měsíc, staré kopie se uchovávají komprimované.
- Záměrně jen pro čtení: logy zůstávají Z-Push (`www-data`), jak je Poste.io chce mít.

→ Logy ActiveSync

Copyright © 2026 Antonio Sanna

Zálohování a obnova



- **Konfigurace** (účty, domény, nastavení, certifikáty, klíče DKIM, kalendáře a adresáře, nastavení webmailu, trénink spamového filtru) a **veškerá pošta nebo vybrané schránky**, každou hodinu, den, týden nebo měsíc.
- Do jednoho nebo více cílů najednou: **SMB/Samba, SFTP, S3**, svazek připojený hostitelem (NFS, druhý disk) nebo **REST server resticu v režimu append-only**, kde kompromitovaný poštovní server nemůže zálohy smazat; u každého cíle před uložením **Otestovat** (dosažitelnost, prázdná složka, volné místo, zápis).
- Vytvářeno pomocí restic *(https://restic.net)*: šifrováno ještě před opuštěním serveru, deduplikováno a komprimováno, takže časté zálohy ukládají jen to, co se změnilo; pravidla uchování pro každou úlohu (hodinové, denní, týdenní, měsíční).
- Průběh v reálném čase, log každého běhu, místo v každém cíli (žlutě a červeně podle limitů, které nastavíte), **statistiky** za 30 dní až rok a plánované **ověření** každého cíle, které uložená data znovu přečte, vždy po částech.
- **Obnova schránky z administrace** do nové složky téže nebo jiné schránky: celé, některých složek, zpráv odpovídajících filtru (data, odesílatel, předmět) nebo **jednotlivých zpráv** nalezených v prohledávatelném seznamu; nic z toho, co už tam je, se nezmění; odstranění záloh úlohy; zálohy čtené z **jiného stroje jen se souborem klíče**; **celý server obnovený** na nové instalaci; každý postup testy provádějí přesně tak, jak je zdokumentován.
- Zabezpečení: modul běží jako `mail`, nikdy jako root; nastavení se před použitím znovu validuje; SFTP jen na server, jehož klíč SSH byl potvrzen; hesla se nikdy neposílají zpět do prohlížeče.

→ Zálohování a obnova

Opravy upstreamu

Do verze 2.5.19 přesunutí zprávy do Junk rspamd neučilo: učicí skripty byly zapisovatelné pro všechny (Dovecot je odmítá spustit), učily do Bayesova úložiště, které kontrola pošty nikdy nečte, a nechávaly v `/tmp` kopii každé zprávy. **Poste.io 2.5.20 obsahuje všechny tři opravy**, doplněk tedy zmizel; na starším základním image sestavujte z vydání 1.2.0. End-to-end testy chování učení dál ověřují při každém sestavení.

Copyright © 2026 Antonio Sanna

→ Opravy upstreamu

Rychlý start

```
git clone https://github.com/atsanna/poste.io-addons.git
cd poste.io-addons

# Pro (requires docker login to the Poste.io registry)
docker build -f Dockerfile-pro --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .
# or open source
docker build -f Dockerfile     --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .

# only some add-ons, e.g. the dark theme and calendars/contacts
sh select-addons.sh Dockerfile-pro dark-theme activesync-dav > Dockerfile.custom
docker build -f Dockerfile.custom --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .
```

Image spouštějte stejně jako oficiální (viz `docker-compose.example.yml` *(https://github.com/atsanna/poste.io-addons/blob/main/docker-compose.example.yml)*). Na **stávající** instalaci si nejprve přečtěte Instalace: svazek `/data` zakrývá složky, které image připravuje, a telefony používající ActiveSync se musí jednou znovu synchronizovat.

Licence

MIT *(https://github.com/atsanna/poste.io-addons/blob/main/LICENSE)* © 2026 Antonio Sanna. Licence se vztahuje pouze na soubory tohoto repozitáře; Poste.io samotné licencují jeho vlastníci.

Copyright © 2026 Antonio Sanna

Instalace

Tento repozitář nedodává hotový image: vlastní image si sestavíte nad oficiálním image Poste.io, takže vždy víte, co v něm je.

Požadavky

- Docker 20.10+ (BuildKit ve výchozím stavu zapnutý).
- **Edice open source**: nic dalšího, základní image `analogic/poste.io` je veřejný.
- **Edice Pro**: licence Poste.io Pro a přístup do registru Poste.io (`docker login` s přihlašovacími údaji, které vám Poste.io poskytne), protože základní image je `poste.io/mailserver`.
- **Poste.io 2.5.24 nebo novější**: jeho administrace běží na Symfony 7, pro které jsou stránky doplňků napsány. Pro starší Poste.io použijte vydání 2.0.1.
- Testováno s **Poste.io 2.5.24**: Pro v produkci, open source (`analogic/poste.io:2.5.24`) sestavený a spuštěný v testovacím kontejneru. Jiné verze mohou fungovat; sestavení se záměrně zastaví, pokud Poste.io změní soubory, do kterých se doplňky zapojují (viz Aktualizace Poste.io).
- **Starší verze, open source**: Docker Hub uchovává jen poslední tag `analogic/poste.io`, předchozí zmizí, jakmile vyjde nová verze. Pro sestavení staršího vydání tohoto repozitáře pro edici open source potřebujete jeho základní image stále v místní cache Dockeru. Registr Pro uchovává všechny verze.

1. Sestavení image

```
git clone https://github.com/atsanna/poste.io-addons.git
cd poste.io-addons

# open source
docker build -f Dockerfile     --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .

# Pro (after docker login to the Poste.io registry)
docker build -f Dockerfile-pro --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .
```

Jak sestavit jen některé doplňky, viz Výběr doplňků.

Sestavení, jedna sekce Dockerfilu na každý doplněk:

1. **Externí schránky**: nainstaluje `getmail6`, zkopíruje stránky administrace do `/opt/admin` (jen nové soubory: žádný soubor Poste.io se nenahrazuje), znovu sestaví classmap Composeru (Pro a open source 2.5.20 používají autoloader typu classmap-authoritative, bez toho jsou nové třídy neviditelné), připojí anglické a české překlady, přidá dvě položky menu, připraví složky getmailu, `cron.sh` a crontab uživatele `mail`;
2. **Tmavý motiv**: zkopíruje své soubory do `/opt/admin` a zapojí je do `base.html.twig`;
3. **Kalendáře a kontakty přes ActiveSync**: nainstaluje `libawl-php`, nakonfiguruje Z-Push a přidá jeho dva backendy;
4. **Logy ActiveSync**: zkopíruje svou stránku do `/opt/admin` (a znovu sestaví classmap), přidá svou položku menu a překlady, skript pro rotaci, svou noční položku cronu a spouštěcí skript;
5. **Opravy upstreamu**: převzato Poste.io ve 2.5.20, už se nic nepoužívá (opravy upstreamu); na starším základním image použijte vydání 1.2.0.

2. Spuštění

Image používejte přesně jako oficiální. Minimální příklad je v `docker-compose.example.yml` *(https://github.com/atsanna/poste.io-addons/blob/main/docker-compose.example.yml)*:

```
cp docker-compose.example.yml docker-compose.yml   # edit hostname, TZ, data path
docker compose up -d
```

Pak otevřete `https://<your host>/admin/`. Nové položky **Externí schránky**, **Logy externích schránek**, **Logy ActiveSync** a **Zálohy poštovního serveru** jsou v levém menu (jen pro superadministrátory); přepínač motivu je v horní liště.

Copyright © 2026 Antonio Sanna

3. Stávající instalace: svazek `/data`

Vše, co Poste.io uchovává (pošta, konfigurace, logy), žije ve svazku `/data` a **svazek zakryje vše, co image umístí pod `/data`**. Do zcela nového, prázdného pojmenovaného svazku Docker obsah image zkopíruje; u bind mountu (`./data:/data`) nebo u existujícího svazku ne. Na stávající instalaci proto jednou vytvořte složky getmailu a zkopírujte spouštěč:

```
docker exec mailserver sh -c '
  mkdir -p /data/var/getmail/5min /data/var/getmail/10min /data/var/getmail/15min \
           /data/var/getmail/30min /data/var/getmail/60min /data/var/getmail/disabled \
           /data/log/getmail &&
  chown -R mail:mail /data/var/getmail /data/log/getmail &&
  find /data/var/getmail /data/log/getmail -type d -exec chmod 0750 {} +'
docker cp external-mailboxes/cron.sh mailserver:/data/var/getmail/cron.sh
docker exec mailserver sh -c 'chown mail:mail /data/var/getmail/cron.sh && chmod 0750 /data/var/getmail/cron.sh'
```

`cron.sh` zůstává na svazku: když ho nové vydání tohoto repozitáře změní, zkopírujte ho znovu stejným způsobem.

Konfigurační soubory getmailu jsou také na svazku; zapisuje je administrace při uložení schránky. Když vydání změní to, co do nich administrace zapisuje (například kontrolu stažené pošty přes rspamd, která je sekcí každého souboru), stávající schránky to získají, až je z administrace **znovu uložíte**: otevřete každou externí schránku, heslo nechte prázdné (zůstane uložené) a uložte. CHANGELOG uvádí, kdy je to potřeba.

S **kalendáři a kontakty přes ActiveSync** změní první spuštění nového image identifikátory složek v Z-Push: vyžádejte na každém zařízení novou synchronizaci, jinak telefony skončí se zdvojenými poštovními složkami. Příkazy jsou v Kalendáře a kontakty přes ActiveSync. Pozdější nová sestavení a aktualizace Poste.io s doplňkem uvnitř to nepotřebují.

Logy ActiveSync nevyžadují nic ručně: jejich složku na svazku vytvoří spouštěcí skript při každém startu kontejneru.

Ani **zálohování** nevyžaduje nic ručně: jeho složku na svazku (`/data/var/backup`) a šifrovací klíč vytvoří spouštěcí skript (klíč jen jednou, při prvním startu). Pak si klíč stáhněte v *Zálohy poštovního serveru* > *Šifrovací klíč* a uložte ho mimo server (viz Zálohování a obnova).

4. Kontrola funkčnosti

```
# the mail user owns the schedule; root has none
docker exec mailserver crontab -u mail -l
docker exec mailserver crontab -u root -l      # "no crontab for root"

# after the first run (every 5 minutes for the 5min folder)
docker exec mailserver tail /data/log/getmail/getmail.log
```

V administraci zobrazuje **Logy externích schránek** log getmailu, log cronu a přehled stavu (verze getmailu, složky, počet schránek).

```
# backup: the last results of the jobs (JSON), and the engine ticking
docker exec mailserver poste-backup status
```

V administraci **Zálohy poštovního serveru** na své nástěnce upozorní, pokud modul neběžel déle než deset minut nebo pokud spouštěcí skript nemohl připravit jeho složku.

Aktualizace Poste.io

Sestavte image s novou verzí a kontejner vytvořte znovu:

```
docker build -f Dockerfile-pro --build-arg POSTE_IO_VERSION=<new> -t my/poste.io:<new> .
```

Sestavení je navrženo tak, aby **hlasitě selhalo**, místo aby vytvořilo napůl funkční image:

- `dark-theme-install.sh` se zastaví, pokud `base.html.twig` už nemá `</head>` na samostatném řádku nebo seznam v pravé části navigační lišty právě jednou; pro Junk (kontroly `grep -q` na konci bloku);
- `activesync-dav-install.sh` se zastaví, pokud některý z 19 řádků konfigurace Z-Push, které mění, není přesně takový, jak se očekává;
- `activesync-logs-install.sh` se zastaví, pokud položku menu *User logs* nenajde právě jednou a následovanou jejím `</li>`;
- `backup-install.sh` se zastaví, pokud chybí nástroj, který potřebuje (restic, rclone, nástroje SSH, smbclient, sqlite3), nebo rozšíření PHP, nebo pokud konec menu administrátorů nenajde právě jednou.

Copyright © 2026 Antonio Sanna

Před každou aktualizací se podívejte na opravy upstreamu: od 2.5.20 je Poste.io dodává a bloky už v Dockerfilech nejsou.

Odinstalace

Znovu sestavte/spusťte oficiální image. Doplňky pouze přidávají soubory; data, která zanechávají ve svazku, jsou `/data/var/getmail` (konfigurace schránek včetně hesel), `/data/log/getmail`, `/data/var/activesync-logs`, staré kopie logů Z-Push a `/data/var/backup` (nastavení záloh včetně hesel k cílům a šifrovací klíč). Pokud je už nepotřebujete, odstraňte je; šifrovací klíč si ponechte, dokud můžete zálohy potřebovat. Pokud jste používali kalendáře a kontakty přes ActiveSync, po přechodu synchronizujte každé zařízení znovu (identifikátory složek se změní zpět). Jak odebrat jen některé doplňky, viz Výběr doplňků.

Copyright © 2026 Antonio Sanna

Výběr doplňků

Doplňky jsou nezávislé: vezměte všechny, nebo jen ty, které chcete. Každý z nich je **jedna sekce** Dockerfilu mezi dvěma značkami `ADD-ON` a má v repozitáři vlastní složku.

Doplněk	Název	Složka	Co v Poste.io mění
Externí schránky	`external-mailboxes`	`external-mailboxes/`	přidává stránky administrace a dvě položky menu, instaluje getmail6 a filtr `getmail-rspamd` (`/usr/local/bin`), přidává crontab pro uživatele `mail` a noční rotaci jeho logů (`/etc/cron.d`, `/etc/logrotate.d`)
Tmavý motiv administrace	`dark-theme`	`dark-theme/`	přidává styl a dvě šablony, dva řádky `include` v `base.html.twig`
Kalendáře a kontakty přes ActiveSync	`activesync-dav`	`activesync-dav/`	19 řádků konfigurace Z-Push, dva backendy Z-Push, instaluje `libawl-php`
Logy ActiveSync v administraci	`activesync-logs`	`activesync-logs/`	přidává stránku administrace a jednu položku menu (za *User logs*), spouštěcí skript (`/etc/cont-init.d`) a noční rotaci logů Z-Push (`/etc/cron.d`, `/etc/logrotate.d`)
Zálohování a obnova	`backup`	`backup/`	přidává stránky administrace a jednu položku menu, modul (`/usr/local/bin/poste-backup` a jeho knihovnu), položku cronu a spouštěcí skript (`/etc/cron.d`, `/etc/cont-init.d`), instaluje restic, rclone, klienta OpenSSH a smbclient

Žádný doplněk nepotřebuje jiný. Testy sestavují image open source s každým doplňkem zvlášť i se všemi dohromady (viz Testy).

Možnost 1: `select-addons.sh`

Skript vypíše Dockerfile jen s doplňky, které uvedete, vše ostatní beze změny:

```
# Pro edition, only the dark theme and the backup
sh select-addons.sh Dockerfile-pro dark-theme backup > Dockerfile.custom
docker build -f Dockerfile.custom --build-arg POSTE_IO_VERSION=2.5.24 -t my/poste.io:2.5.24 .

# open source edition, only External Mailboxes
sh select-addons.sh Dockerfile external-mailboxes > Dockerfile.custom
```

U neznámého názvu skončí chybou a vypíše názvy, které zná. Se všemi šesti názvy vypíše Dockerfile přesně tak, jak je.

Možnost 2: úprava Dockerfilu

Každá sekce vypadá takto:

```
##############################################################################
#  ADD-ON: Dark theme for the admin  [dark-theme]
#  docs/en/docs/dark-theme.md
#  To leave it out, delete this whole section (down to the next ADD-ON).
##############################################################################
...
```

Smažte vše od prvního řádku `#####` až po řádek před další hlavičkou `#####` / `ADD-ON` (u poslední sekce až do konce souboru). Část nad první sekcí (základní image, labels) ponechte.

Než ze stávající instalace doplněk vynecháte

Odebrání doplňku z image neuklidí to, co zanechal ve svazku `/data`, a některé doplňky při přidání nebo odebrání vyžadují krok navíc:

- **Externí schránky**: konfigurace schránek (včetně hesel) zůstávají v `/data/var/getmail`, log v `/data/log/getmail`. Pokud je už nepotřebujete, smažte je.
- **Kalendáře a kontakty přes ActiveSync**: přidání nebo odebrání změní identifikátory složek ActiveSync, takže **každé zařízení se musí znovu synchronizovat** (Kalendáře a kontakty přes ActiveSync). Samotné kalendáře a kontakty zůstávají na serveru DAV.

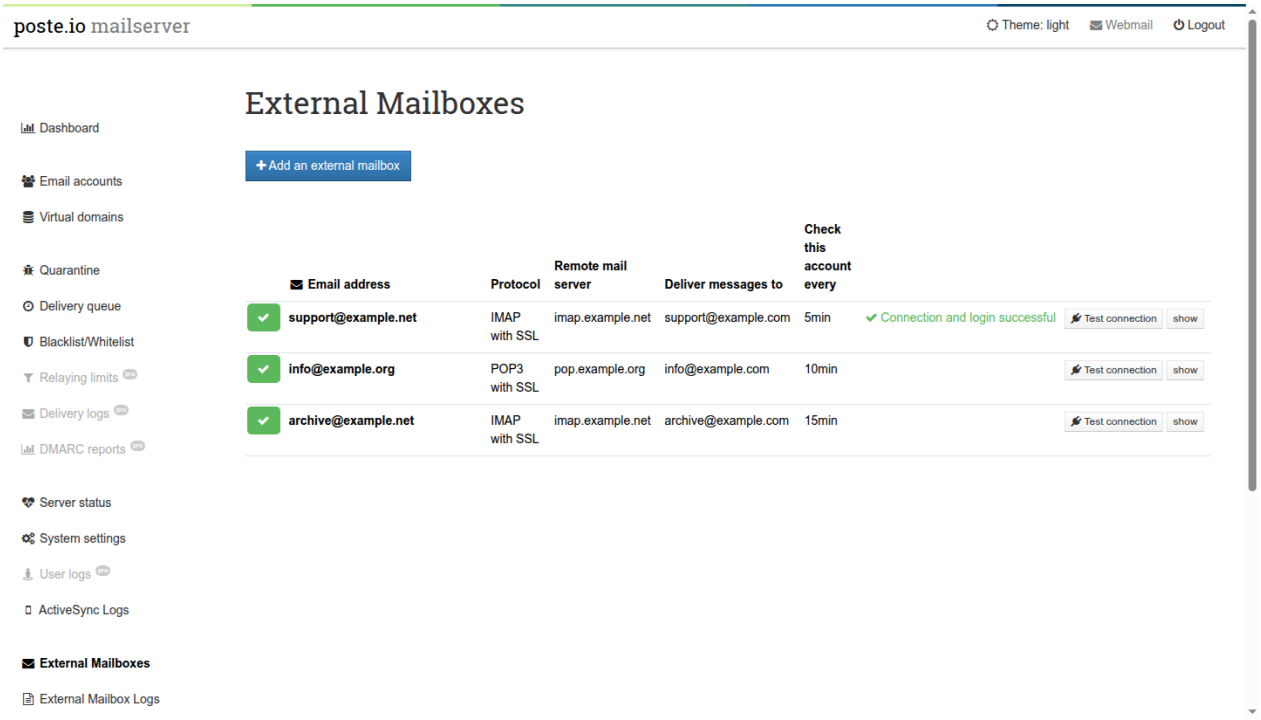
Copyright © 2026 Antonio Sanna

- **Logy ActiveSync**: nastavení rotace zůstává v `/data/var/activesync-logs`, staré kopie logů Z-Push v `/data/log/z-push` (`*.log.1`, `*.log.N.gz`). Pokud je už nepotřebujete, smažte je.
- **Zálohování**: nastavení (včetně hesel k cílům) a šifrovací klíč zůstávají v `/data/var/backup`. Zálohy v cílech zůstanou nedotčeny: klíč si ponechte, dokud je můžete potřebovat (Zálohování a obnova).
- **Tmavý motiv**: není třeba nic dělat.

Copyright © 2026 Antonio Sanna

Externí schránky

Stahuje poštu ze schránek hostovaných jinde (POP3 nebo IMAP, s SSL i bez něj) a podle plánu ji doručuje do Poste.io pomocí getmail6 *(https://github.com/getmail6/getmail6)*. Vše se nastavuje v administraci Poste.io (jen pro superadministrátory): **Externí schránky** a **Logy externích schránek** v levém menu.



Co doplněk dělá

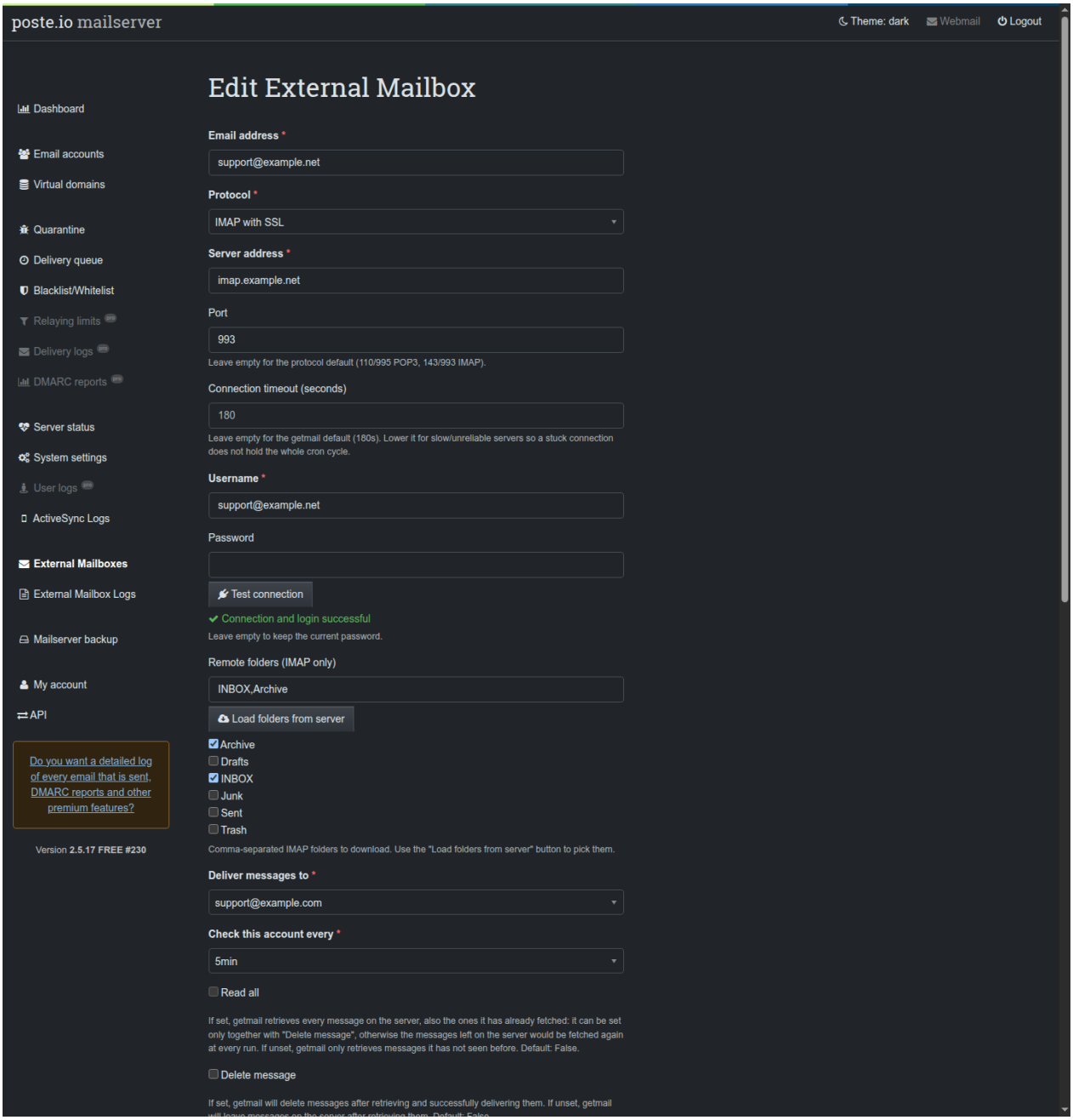
- **Protokoly**: POP3, POP3 přes SSL, IMAP, IMAP přes SSL.
- **Plán**: každých 5, 10, 15, 30 nebo 60 minut, nebo vypnuto.
- **Doručení**: do libovolné místní schránky **nebo přesměrování** (aliasu). Doručení prochází samotným Poste.io (`sendmail` → Haraka, uvnitř kontejneru), takže se uplatní filtry a přesměrování: když externí schránku `support@` doručíte do místního přesměrování, každý člen přesměrování dostane kopii. V menu cíle doručení se přesměrování zobrazují jako `support@example.com → alice@example.com (redirect)`.
- **Složky IMAP**: zvolte, které vzdálené složky se mají stahovat; **Načíst složky ze serveru** vypíše skutečné složky účtu.
- **Vzdálené čištění (IMAP)**: volitelně vyprazdňuje vzdálené složky Koš a/nebo Spam po uplynutí doby uchování (ihned, 1 den … 1 měsíc). Pokud se složka zároveň importuje i čistí, „Smazat ihned“ se prodlouží na 1 den, aby zpráva nikdy nemohla být smazána dřív, než se naimportuje.
- **Volby getmailu**: stahovat všechny / jen nové zprávy, mazat po stažení, mazat po N dnech, přidat hlavičku `Received:`, logovat do syslogu, limity na běh (zprávy a bajty, užitečné pro první import velké schránky), mazat na serveru zprávy větší než N bajtů, vlastní port a časový limit spojení.
- **„Stahovat vše“ jen s „Mazat zprávy“**: se „Stahovat vše“ getmail stáhne všechny zprávy na serveru, i ty, které už stáhl; pokud na serveru zůstávají, getmail varuje, že „all messages will be retrieved each time getmail is run“. Proto lze „Stahovat vše“ zapnout jen spolu s „Mazat zprávy“: formulář volbu vypne, když je „Mazat zprávy“ vypnuto, a administrace takovou kombinaci odmítne. U nové schránky je vypnutá (stahují se jen zprávy, které dříve staženy nebyly a které si getmail pamatuje ve svých souborech `oldmail-*`).
- **Otestovat připojení**: jedno přihlášení na vzdálený server, z formuláře (s právě upravovanými hodnotami) i z každého řádku seznamu (s uloženou konfigurací). Nikdy automaticky a jediný pokus na jedno kliknutí: poskytovatelé po opakovaných neúspěších účty zamykají.
- **Stránka logů**: log getmailu, chyby getmailu a log cronu, s vyhledáváním, limitem řádků, automatickým obnovováním a stavovým zobrazením (verze getmailu, složky, počet externích schránek pro každý plán). **Vymazat** vyprázdní log getmailu nebo chyby getmailu. Syslog a log pošty se nabízejí jen tehdy, když je administrace může číst; na standardním Poste.io nemůže (`root:adm`, `0640`), takže se neobjeví.
- **Chyby getmailu**: co getmail hlásí (chyby konfigurace, neúspěšná přihlášení, nedostupné servery, pády, varování ke konfiguraci), zapisuje `cron.sh` do vlastního logu, každý řádek s časem a souborem schránky, ke kterému patří, a na stránce logů se to

Copyright © 2026 Antonio Sanna

zobrazuje jako **Chyby getmailu**. Pokud existují řádky z posledních 24 hodin, výchozí zobrazení getmailu na to upozorní červeným řádkem nahoře (kolik jich je a poslední z nich). Samostatný soubor proto, že administrace nemůže číst `/data/log/mail.log` (`root:adm`, `0640`), kam chyby jdou také, a protože některé z nich (konfigurace, kterou getmail nedokáže přečíst, pád) se do vlastního logu getmailu nikdy nedostanou.

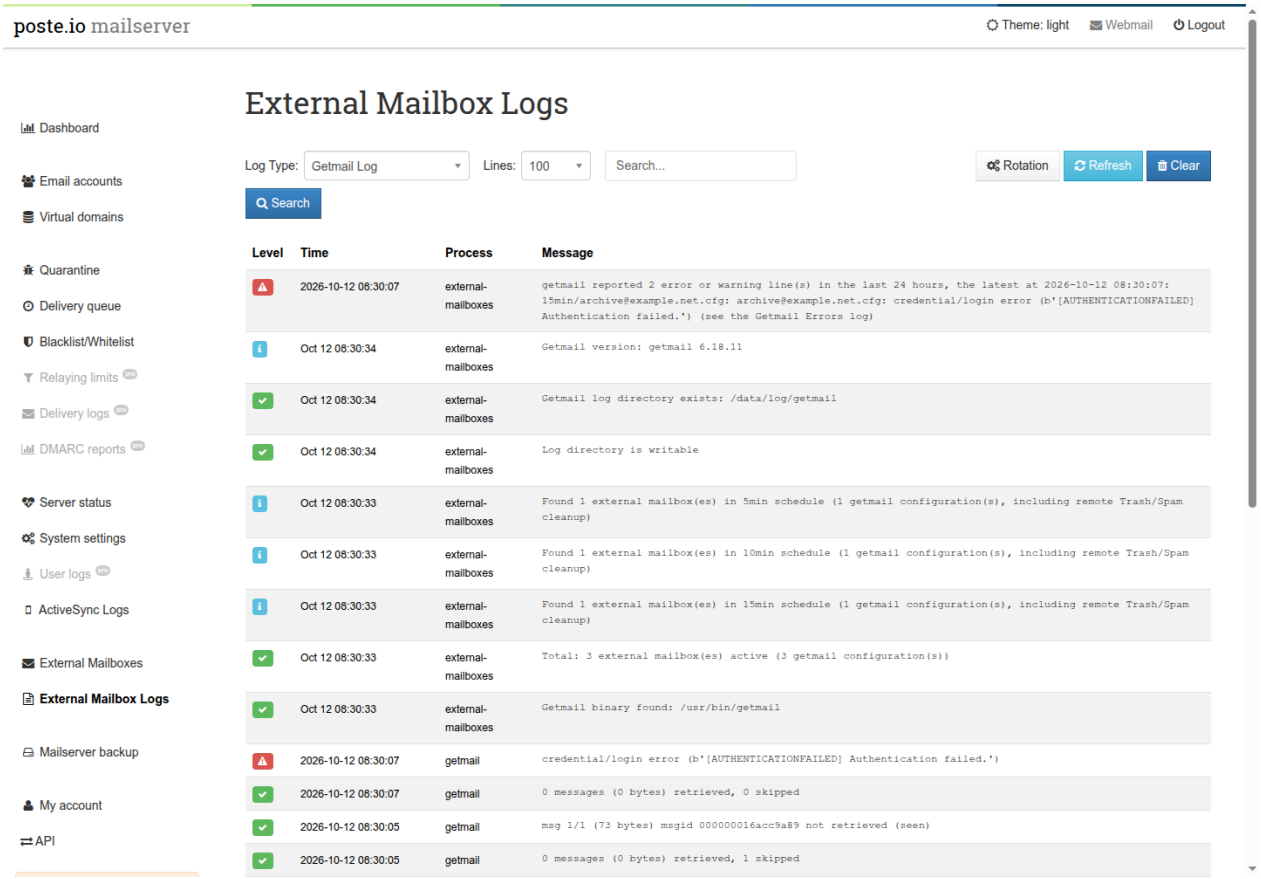
- **Rotace logů**: `getmail.log` a `errors.log` rotuje noční logrotate Poste.io s nastavením zvoleným na stránce logů (tlačítko **Rotace**): rotace zapnuta nebo vypnuta, jak často se rotuje log getmailu (den, týden, měsíc), velikost, nad kterou se rotují chyby getmailu (512 KB až 10 MB), a kolik starých kopií se uchovává, komprimovaných (1 až 12). Výchozí: týdně, 1 MB, 4 kopie. Chyby se rotují podle velikosti, a ne podle data, protože červený řádek zobrazení getmailu čte jen aktuální `errors.log`. Viz Jak to funguje níže.
- **Kontrola spamu (rspamd)**: každou staženou zprávu před doručením zkontroluje rspamd, prostřednictvím filtru getmailu (`getmail-rspamd`, součást image). Zpráva dostane stejné hlavičky `X-Rspamd-Score`, `X-Rspamd-Bar` a `X-Rspamd-Report` jako pošta přijatá přes SMTP (existující se nahradí, takže je nelze podvrhnout), a od skóre, při kterém Poste.io označuje spam (`munge_subject` v souboru `quarantine_decide.ini` serveru Haraka, výchozí 6), také prefix předmětu `*****SPAM*****`, podle kterého ji pravidlo sieve schránky zařadí do **Junk**. Nad skóre karantény Poste.io jde zpráva také do Junk: nic staženého se nezadržuje. Bez filtru se stažená pošta vůbec nekontroluje: getmail doručuje přes lokální soket pro odesílání serveru Haraka, kde plugin rspamd lokální spojení přeskakuje (`local_ip=false`). Bayesovský klasifikátor se učí do úložiště domény příjemce, stejně jako u pošty přijaté přes SMTP. **Omezení**: zpráva přichází ze schránky, ne od klienta SMTP, takže kontroly adresy a jména klienta se neuplatní (ta, která by se spustila u každé zprávy, neznámé jméno hostitele, je vypnuta), SPF nelze zkontrolovat a DMARC se opírá jen o DKIM (nepodepsaná zpráva z domény s politikou `reject` dostane `DMARC_POLICY_REJECT`, +2). Vzdálený poskytovatel kontroluje SPF a DMARC se skutečnou adresou, když poštu přijímá. **Pokud rspamd neodpovídá**, zpráva se doručí beze změny a důvod se zapíše do chyb getmailu (níže): kvůli filtru se pošta nikdy neztratí ani nezpozdí.
- **Překlady**: angličtina a čeština, stejně jako zbytek administrace Poste.io (jazyk vybírá prohlížeč).

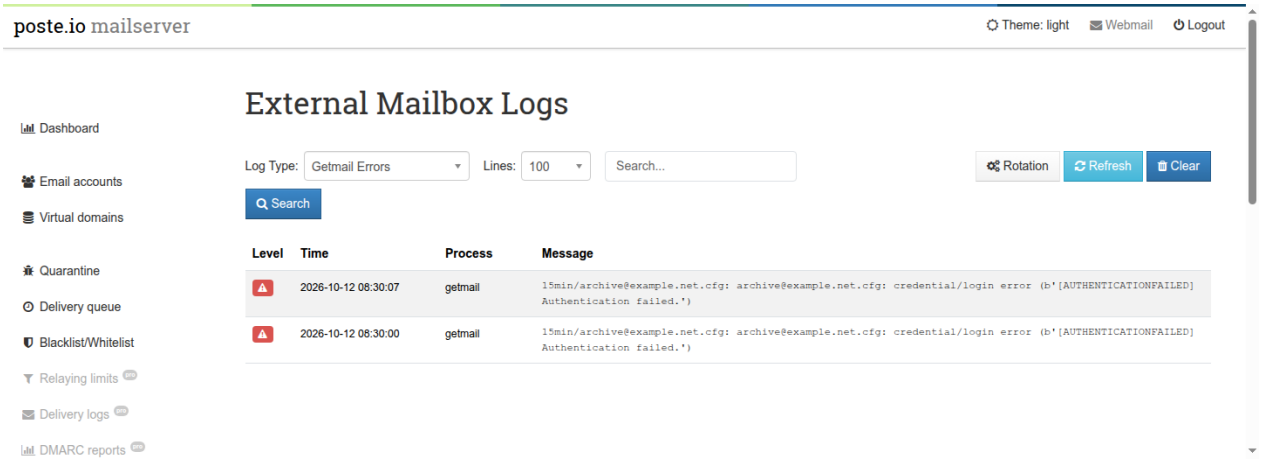
Copyright © 2026 Antonio Sanna



Stránka logů: výchozí zobrazení getmailu s červeným řádkem nahoře, když jsou k dispozici nedávné chyby, a zobrazení **Chyby getmailu** (ukázková data).

Copyright © 2026 Antonio Sanna





Jak to funguje

„Databází“ je souborový systém, uvnitř svazku `/data`:

```
/data/var/getmail/
├── 5min/ 10min/ 15min/ 30min/ 60min/   one .cfg per mailbox; the folder is the schedule
├── disabled/                            mailboxes not scheduled
├── cron.sh                              the runner
└── <schedule>/oldmail-*                 getmail's memory of already fetched messages
/data/log/getmail/getmail.log            operational log (written by getmail)
/data/log/getmail/errors.log             what getmail reports as errors (written by cron.sh)
```

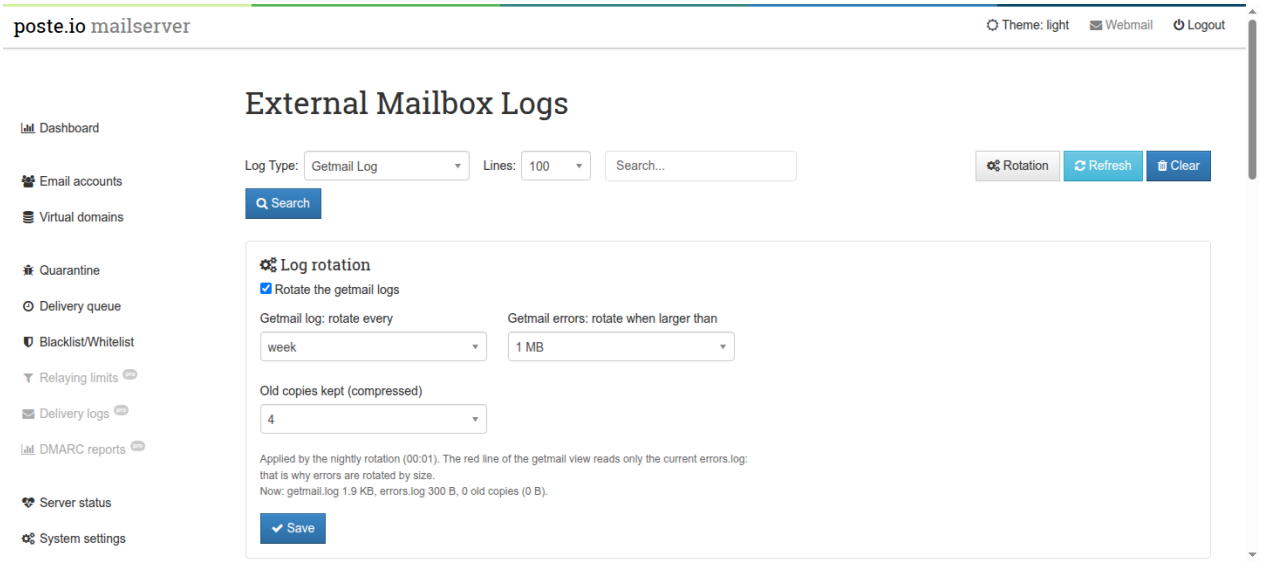
Schránka se vzdáleným čištěním má ve své složce až tři soubory: `user@example.com.cfg` (stahování), `user@example.com.trash.cfg` a `user@example.com.spam.cfg` (stažení → zahození do `/bin/true` → smazání na serveru). Soubory čištění se v seznamu neuvádějí zvlášť: vytvářejí se, aktualizují a odstraňují spolu se svou schránkou.

Crontab uživatele **`mail`** (`/etc/external-mailboxes.crontab`) spouští pro každý plán `cron.sh <folder>`; `cron.sh` spouští getmail na každý `.cfg` s `flock -n`, takže se pomalá schránka nikdy nepřekrývá sama se sebou. getmail nikdy neběží jako root. Každá

Copyright © 2026 Antonio Sanna

stažená zpráva projde filtrem `getmail-rspamd` (sekce `[filter1]` v `.cfg`) a pak se doručí pomocí `sendmail` z Poste.io; co getmail vypíše jako chyby, jde do `errors.log`.

Rotace logů



Administrace běží jako `mail` a nemůže zapisovat pravidla logrotate, která musí patřit rootovi. Nastavení se proto ukládá do `/data/var/getmail/log-rotation.ini` (na svazku: přežije nové sestavení i aktualizace) a skript spouštěný jako root, `/usr/local/bin/external-mailboxes-logrotate`, z něj každou noc v 00:00 (`/etc/cron.d/external-mailboxes-logrotate`) vytvoří `/etc/logrotate.d/external-mailboxes`, minutu před logrotate Poste.io. Změna provedená v administraci platí od příští noci. Image obsahuje pravidlo s výchozím nastavením, zapsané týmž skriptem při sestavení.

Soubor nastavení zapisuje uživatel s nižšími oprávněními a čte ho root: skript z něj nic nespouští a přijímá jen hodnoty nabízené formulářem (`enabled` 0/1, `getmail_every` daily/weekly/monthly, `errors_size` 512k/1M/5M/10M, `keep` 1/2/4/8/12); cokoli jiného znamená výchozí hodnotu. Administrace kontroluje tytéž hodnoty před uložením (a také svůj vlastní token CSRF).

```
/data/var/getmail/log-rotation.ini          settings (admin, mail, 0640)
/usr/local/bin/external-mailboxes-logrotate writes the rule (root)
/etc/cron.d/external-mailboxes-logrotate    runs it at 00:00
/etc/logrotate.d/external-mailboxes         the rule, used by logrotate at 00:01
```

Zabezpečení

- **Soubory**: `/data/var/getmail` a `/data/log/getmail` patří `mail:mail`, složky `0750`, soubory `.cfg` a `oldmail-*` `0600`. Administrace (php-fpm běží jako `mail`) je zapisuje, cron je čte jako `mail`.
- **Hesla**: getmail je potřebuje v souborech `.cfg` jako čistý text, takže jsou uložena tam (čitelná jen pro `mail`). **Nikdy se neposílají zpět do prohlížeče**: při úpravě znamená prázdné pole hesla „ponechat uložené“ a uložené heslo se přečte a zapíše zpět nedotčené.
- **Uložené heslo a testovací tlačítka**: s prázdným polem hesla použijí „Otestovat připojení“ a „Načíst složky ze serveru“ uložené heslo **jen tehdy, když protokol, server, port a uživatelské jméno jsou přesně ty uložené**; jinak o heslo požádají. Bez tohoto pravidla by stačilo změnit server ve formuláři a uložené heslo by se odeslalo na libovolného hostitele.
- **TLS**: u protokolů SSL nastavuje vygenerovaná konfigurace `ca_certs`, takže getmail ověřuje certifikát serveru i jeho jméno (bez toho getmail používá `CERT_NONE`). „Otestovat připojení“ a „Načíst složky ze serveru“ ho ověřují také. Protokoly bez SSL přenášejí čistý text, jak říká jejich název.
- **CSRF**: Poste.io vypíná ochranu formulářů Symfony proti CSRF (`fos_rest.yaml`, `disable_csrf_role: PUBLIC_ACCESS`), takže každý formulář a každé volání AJAX modulu nese vlastní explicitní token, který kontroluje controller.
- **Validace**: protokol, plán a doba uchování musí být hodnoty nabízené formulářem; e-mailová adresa se stává názvem souboru a kontroluje se proti přísnému vzoru (žádné `/`, žádné `..`); název serveru také; cíl doručení musí být existující schránka nebo přesměrování; hodnoty zapisované do souboru `.cfg` nesmějí obsahovat zalomení řádku (vložily by do něj konfigurační řádky).

Copyright © 2026 Antonio Sanna

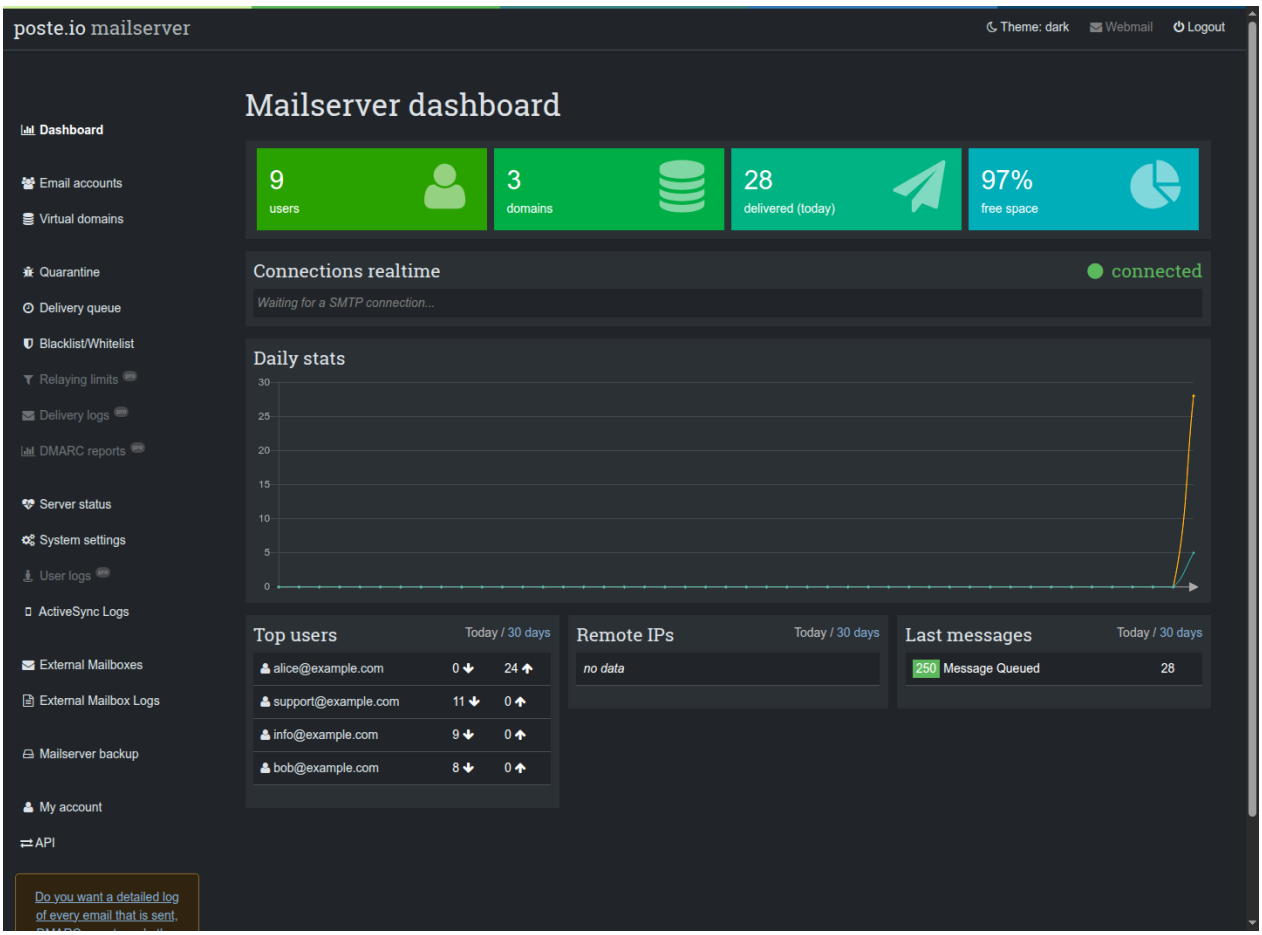
Poznámky a omezení

- **Gmail / Microsoft 365**: OAuth2 není podporováno. Gmail přijímá IMAP s *app password* (vyžaduje dvoufázové ověření) při použití protokolu IMAP s SSL a `imap.gmail.com`; autor to netestoval, svůj účet si ověřte pomocí **Otestovat připojení**. Složky Gmailu se jmenují například `[Gmail]/Spam`.
- **Multidrop** (jedna vzdálená schránka typu catch-all tříděná podle původního příjemce) se nenabízí: chcete-li jednu externí schránku posílat více lidem, doručujte ji do přesměrování.
- **Test POP3**: c-client nepřijímá ani polootevřené relace POP3, ani relace jen pro čtení, takže test otevře `INBOX`; zavírá ho bez expunge, takže se žádná zpráva nesmaže.
- Stavové řádky stránky logů jsou výstupem logu a zůstávají v angličtině.

Copyright © 2026 Antonio Sanna

Tmavý motiv administrace

Tmavý motiv pro stránky administrace Poste.io (`/admin`), automatický a s přepínačem.



Chování

- **Automaticky**: řídí se nastavením operačního systému / prohlížeče (`prefers-color-scheme`), i když se změní při otevřené stránce.
- **Přepínač**: horní lišta dostane položku *Motiv* vedle *Webmail* a *Logout*; každé kliknutí přepne **automatický → tmavý → světlý → automatický**. Volba se uchovává v prohlížeči (`localStorage` , klíč `poste-admin-theme`); *automatický* ji odstraní.
- **Žádné bílé probliknutí**: o motivu rozhodne malý skript v `<head>` ještě před vykreslením stránky.
- **Přihlašovací stránka**: je tmavá také, když je tmavý systém (přepínač tam není: přihlašovací stránka nahrazuje horní lištu).
- **Grafy** (nástěnka, stav serveru): Chart.js kreslí vlastními barvami, proto se upravuje text, mřížka a okraje prstencových grafů a grafy, které už na stránce jsou, se při změně motivu znovu vykreslí.
- **Světlý motiv nedotčen**: každé pravidlo je omezeno na `[data-bs-theme=dark]` . Ověřeno porovnáním vypočtených barev stránky se stylem a bez něj ve světlém režimu: žádný rozdíl.

Webmail (Roundcube, vzhled Elastic) už má vlastní tmavý režim a zůstává nedotčen.

Jak to funguje

Administrace Poste.io je postavena na Bootstrapu 5.3, jehož tmavý barevný režim potřebuje jen `data-bs-theme="dark"` na `<html>` : pozadí, text, okraje, tabulky, vstupní pole a upozornění se přizpůsobí. Zbývají barvy, které Poste.io napevno zadává pro světlý vzhled, opravené v `dark-theme/admin/public/dark-theme.css` *(https://github.com/atsanna/poste.io-addons/blob/main/dark-theme/admin/public/dark-theme.css)*:

- horní lišta a levé menu, „pěkné“ přechody tlačítek, stránkování;
- rozbalovací nabídky Chosen, rámečky s nápovědou, rámečky sekcí nástěnky a indikátor načítání;
- černý text zapsaný inline v několika šablonách (`style="color: black"`).

Copyright © 2026 Antonio Sanna

Soubory:

Soubor	Úloha
`dark-theme/admin/templates/Base/DarkTheme/_head.html.twig`	styl + skript motivu, vložené před `</head>`
`dark-theme/admin/templates/Base/DarkTheme/_toggle.html.twig`	položka *Motiv* v horní liště
`dark-theme/admin/public/dark-theme.css`	opravy jen pro tmavý motiv
`dark-theme/dark-theme.{en,cs}.yml`	překlady přepínače
`dark-theme/dark-theme-install.sh`	přidá oba `include` do `base.html.twig` v Poste.io a překlady

`dark-theme-install.sh` je jediná část, která se dotýká souboru Poste.io, a to dvěma řádky `include`. Odmítne se spustit, pokud svá místa zapojení nenajde právě jednou, takže aktualizace Poste.io, která je přesune, rozbije sestavení, místo aby motiv bez povšimnutí zmizel.

Našli jste stránku, která zůstává světlá?

Poste.io někdy zapisuje barvy přímo do šablony. Založte issue se snímkem obrazovky a URL stránky; oprava je obvykle jedno pravidlo v `dark-theme.css`.

Copyright © 2026 Antonio Sanna

Kalendáře a kontakty přes ActiveSync

Poste.io obsahuje dvě věci, které se do verze 2.5.23 nikdy nepotkaly:

- **Z-Push**, server ActiveSync, který telefony používají pro typ účtu „Exchange“, nastavený **jen pro poštu**: přepínače kalendáře a kontaktů v telefonu jsou zapnuté, ale nikdy nic nedorazí;
- **server CalDAV/CardDAV** (SabreDAV, na `https://<your host>/dav/`), kde každá schránka může mít kalendáře a adresáře, vytvořené v administraci.

S jedním účtem ActiveSync dostane telefon poštu **i** kalendáře a kontakty ze serveru DAV: stejná data, jaká vidí klient CalDAV/CardDAV, a to oběma směry. **Od verze 2.5.24 je Poste.io propojuje samo**, se standardními backendy CalDAV a CardDAV ze Z-Push. Tento doplněk (Poste.io 2.5.24 nebo novější) to zachovává a nahrazuje jen tyto dva backendy opravenými: událost odmítnutá serverem se telefonu nehlásí jako uložená a vypnutý server DAV nevyřadí ActiveSync (viz obejité problémy níže).

Co uživatelé dostanou

Schránka má (v administraci)	Telefon zobrazí
nic	poštu + „Calendar“ a „Contacts“, které pro ni vytvoří Poste.io (viz níže)
kalendář	poštu + tento kalendář (jako **výchozí** kalendář) + jeho úkoly
adresář	poštu + kontakty
kalendář sdílený kolegou	tento kalendář jako **další** kalendář

Vše funguje oběma směry: událost nebo kontakt vytvořené v telefonu se uloží na serveru a změny na serveru se dostanou do telefonu.

Zapnutí kalendářů a kontaktů pro schránku

Kalendáře a adresáře se vytvářejí pro každou schránku v administraci. Od verze 2.5.24, když je DAV zapnutý, server DAV Poste.io navíc dá schránce, která žádné nemá, prázdný „Calendar“ a „Contacts“ při jejím prvním přístupu k DAV (kalendář webmailu, ActiveSync a pozvánky potřebují kam ukládat): doplněk tomu nemůže a ani nebrání.

1. **Settings → DAV (Experimental) → Enabled** musí být zaškrtnuto (zapíná samotný server DAV).
2. **Mailboxes →** příslušná schránka **→ DAV**, sekce: *Create calendar* a/nebo *Create address book*.
3. V telefonu musí být v nastavení účtu ActiveSync zapnutá synchronizace kalendáře a kontaktů; pak synchronizujte (nebo počkejte na příští synchronizaci).
4. Po zapnutí nebo vypnutí DAV **restartujte kontejner**: od verze 2.5.24 Poste.io vybírá backend ActiveSync (jen pošta, nebo pošta + kalendáře a kontakty) při startu kontejneru.

Schránka bez kalendáře a bez adresáře má v telefonu dál jen poštu. Když se kalendář nebo adresář v administraci vytvoří nebo smaže, telefon složku přidá nebo odebere při příští synchronizaci složek.

Je kolega zaneprázdněn? Sdílení jen pro čtení

V sekci DAV schránky kolegy otevřete u kalendáře stránku *share*, vyberte schránku, se kterou se má sdílet, a zvolte **read only**. V telefonu tohoto uživatele se kalendář kolegy objeví jako další kalendář, s událostmi a jejich názvy (server DAV nemá úroveň „jen volno/obsazeno“).

Zápis do kalendáře jen pro čtení z telefonu je odmítnut: nová událost dostane chybu, změna dostane konflikt (vyhraje verze na serveru), smazání se na serveru neprovede a událost se při příští synchronizaci vrátí. Funguje i sdílení pro čtení i zápis.

Výchozí kalendář

ActiveSync má jeden výchozí kalendář na účet. Poste.io pojmenovává kalendáře náhodnými identifikátory, takže doplněk nehádá: ptá se serveru přes `schedule-default-calendar-URL` (RFC 6638), tutéž vlastnost, kterou klienti CalDAV používají pro pozvánky. SabreDAV odpoví **prvním kalendářem, který uživatel vlastní**; kalendáře sdílené kolegy nejsou nikdy výchozí.

Copyright © 2026 Antonio Sanna

Nastavení telefonu

Přidejte účet **Exchange / ActiveSync** s adresou a heslem schránky a názvem hostitele vašeho poštovního serveru a pak ověřte, že je pro tento účet zapnutá synchronizace kalendáře a kontaktů. Testováno s aplikací Gmail v Androidu; ostatní klienti ActiveSync by se měli chovat stejně, ale testováni nebyli.

Když je server DAV vypnutý nebo nefunkční

Kombinovaný backend Z-Push odmítne celé přihlášení, pokud ho odmítne kterýkoli z jeho backendů, takže bez opatření by server DAV, který neodpovídá, vyřadil ActiveSync pro všechny uživatele, **včetně pošty**. Doplněk tomu brání:

- pokud server DAV při přihlášení neodpoví tak, jak má (například když je DAV v nastavení vypnutý), backendy kalendářů a kontaktů přihlášení přijmou **bez složek**;
- telefon dál synchronizuje poštu (nová pošta se doručí ve stejné odpovědi) a kalendáře a kontakty odstraní při příští synchronizaci složek;
- když server DAV znovu funguje, kalendáře a kontakty se vrátí samy.

Log Z-Push pak ukazuje `the DAV server is not answering ... mail goes on`.

Instalace do existující instalace: zařízení se musí znovu synchronizovat

Sestavte a spusťte image podle Instalace. Doplněk mění konfiguraci Z-Push a s ní i identifikátory poštovních složek; telefon, který už synchronizoval, by dostal nové složky, **aniž by se odstranily ty staré** (zdvojené poštovní složky). Hned po prvním spuštění s doplňkem požádejte každé zařízení o úplnou novou synchronizaci (resync):

```
# every user/device pair known to Z-Push
docker exec <container> sh -c 'cd /opt/z-push && runuser -u www-data -- php z-push-admin.php -a list'

# one pair (device ids are listed without dashes)
docker exec <container> runuser -u www-data -- php /opt/z-push/z-push-admin.php -a resync -u <user> -d <device>

# or all pairs at once
docker exec <container> sh -c 'cd /opt/z-push && runuser -u www-data -- php z-push-admin.php -a list | awk "NF == 2
&& \$1 != \"Device\" && \$1 !~ /^-/" | while read -r d us; do for u in $(echo "$us" | tr , " "); do runuser -u www-
data -- php z-push-admin.php -a resync -u "$u" -d "$d"; done; done'
```

Každý telefon pak při příští synchronizaci začne znovu od prázdného seznamu složek (může být potřeba otevřít poštovní aplikaci). Zprávy zůstávají na serveru; telefon je jen znovu stáhne.

Nové sestavení image nebo aktualizace Poste.io s doplňkem uvnitř resync nevyžaduje: konfigurace zůstává stejná a stav Z-Push je uložen na svazku `/data` (`/data/zpush`).

Odebrání doplňku

Odeberte z Dockerfile blok *ActiveSync: calendars and contacts*, sestavte image znovu, znovu vytvořte kontejner a pak znovu synchronizujte všechna zařízení jako výše (identifikátory složek se změní zpět).

Aktualizace Poste.io

`activesync-dav-install.sh` kontroluje každý řádek, který v konfiguraci Z-Push mění, a **zastaví sestavení**, pokud některý z nich není přesně takový, jak se očekává. Vydání Poste.io, které změní Z-Push, tedy sestavení hlasitě přeruší, místo aby vytvořilo napůl nakonfigurovaný image.

Omezení

- **Složitá opakování se v telefonu zjednoduší** (omezení Z-Push): například „měsíčně 2. a 3. den“ dorazí jako „měsíčně 2. den“. Událost na serveru zůstane nedotčená.
- **Žádné zjišťování volna/obsazenosti při zvaní**: Z-Push ho implementuje jen pro Kopano, ne pro CalDAV. Ke zobrazení kalendáře kolegy použijte sdílení jen pro čtení.
- **Žádný sdílený ani globální adresář**: server DAV v Poste.io sdílí kalendáře, ne adresáře.
- **Všechny adresáře uživatele se zobrazí jako jedna složka kontaktů** (backend CardDAV v Z-Push); nové kontakty z telefonu jdou do prvního adresáře.

Copyright © 2026 Antonio Sanna

- **Úkoly**: Z-Push zobrazuje každý kalendář dvakrát, jako události a jako úkoly.

Jak to funguje

Konfigurace měněná skriptem `activesync-dav/activesync-dav-install.sh` (3 řádky ve 2 souborech; každý, i nastavení Poste.io, na která backendy spoléhají, se zkontroluje dřív, než se cokoli zapíše):

Soubor (v `/opt/z-push`)	Změna
`config.php`	žádná: Poste.io nastavuje `BACKEND_PROVIDER` při každém startu (`BackendCombined`, když je DAV zapnutý)
`backend/combined/config.php`	backend kalendářů `c` → `BackendPosteCalDAV`, backend kontaktů `d` → `BackendPosteCardDAV` (mapování složek z Poste.io zůstává)
`backend/caldav/config.php`	žádná: server, port a cesta jsou ty, které zapisuje Poste.io
`backend/carddav/config.php`	výchozí cesta `/dav/addressbooks/%u/` místo `…/default/`: nové kontakty jdou do prvního adresáře uživatele, žádný „default“ se nevytváří

Soubory nainstalované v `/opt/z-push/backend/`:

Soubor	Role
`postecaldav/postecaldav.php`	`BackendPosteCalDAV`, rozšiřuje `BackendCalDAV` ze Z-Push
`postecarddav/postecarddav.php`	`BackendPosteCardDAV`, rozšiřuje `BackendCardDAV` ze Z-Push
`postecaldav/postedav.php`	pomocník pro PROPFIND sdílený oběma

Oba backendy komunikují se serverem DAV uvnitř kontejneru, na adrese, kterou nastavuje Poste.io, pomocí přihlašovacích údajů, se kterými se přihlásil telefon (HTTP Basic, heslo schránky). Každé přihlášení začíná jedním PROPFIND, který je zároveň kontrolou stavu serveru DAV. Balíček Debianu `libawl-php` (potřebuje ho klient CalDAV v Z-Push) je součástí Poste.io od verze 2.5.24; Dockerfile ho přesto instaluje, neškodně, aby doplněk na tom nezávisel; jeho adresář přidává do include path PHP pouze backend.

Obejité problémy Z-Push

Nalezeny při testování Z-Push 2.7.5 proti serveru DAV v Poste.io a opraveny ve dvou backendech bez úprav vlastních souborů Z-Push:

1. **Událost odmítnutá serverem byla telefonu potvrzena jako uložená.** `BackendCalDAV` nikdy nekontroluje výsledek PUT, takže událost zapsaná do sdíleného kalendáře jen pro čtení zůstala v telefonu, zatímco ji nikdo jiný neviděl. Nyní je odmítnutý zápis nebo smazání pro zařízení chybou.
2. **Kontakt přidaný uživatelem bez adresáře shodil Z-Push** (nedostatek paměti, HTTP 500 pro celou synchronizaci včetně pošty, telefon to zkoušel donekonečna znovu). Nyní uživatel bez adresáře nemá složku kontaktů a kontakt, který se přesto odešle, dostane chybu.
3. **Vypnutý server DAV vyřadil ActiveSync**, viz výše.

Řešení potíží

- Logy Z-Push: `/var/log/z-push/z-push.log` a `z-push-error.log` v kontejneru. Řádky doplňku začínají `BackendPosteCalDAV`, `BackendPosteCardDAV` nebo `PosteDAV`.
- Co má zařízení: `z-push-admin.php -a list -u <user> -d <device>` (spuštěno jako `www-data`, viz výše) ukáže složky a synchronizovaná data (`Emails`, `Calendars`, `Tasks`, `Contacts`).
- Kalendáře chybí všem: zkontrolujte, že je DAV v nastavení zapnutý; s vypnutým DAV odpovídá `/dav/` *Enable DAV at your server settings*.
- Kalendáře chybí jednomu uživateli: tato schránka nemá v administraci kalendář, nebo má telefon pro tento účet vypnutou synchronizaci kalendáře.

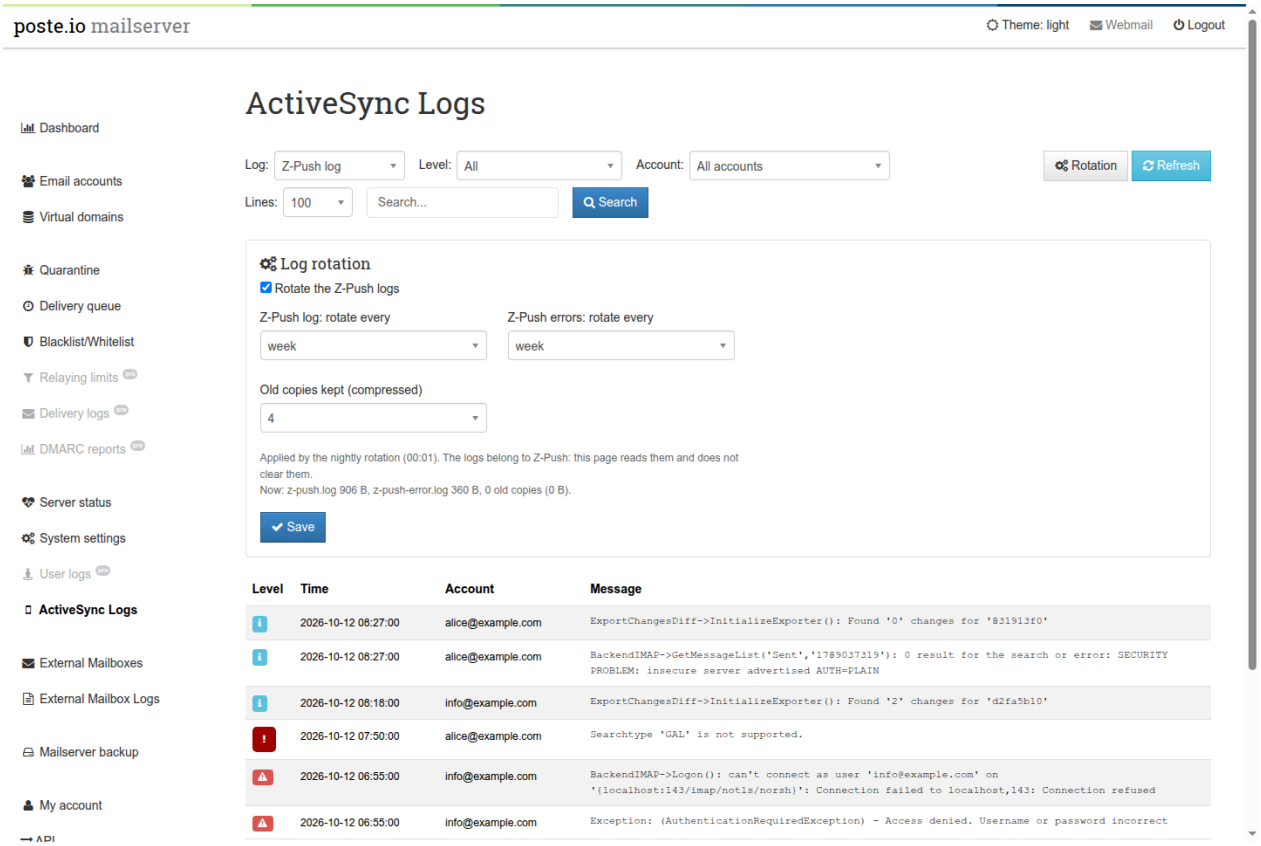
Testováno

Poste.io 2.5.24 se Z-Push 2.7.5, Pro (v produkci) a open source (sestaveno a otestováno v kontejneru). V testovacím kontejneru sestaveném z Dockerfile: typy složek a výchozí kalendář, server ↔ telefon pro události a kontakty, sdílení jen pro čtení, uživatelé bez zapnutých funkcí, adresář odstraněný, zatímco ho telefon ještě měl jako složku, DAV vypnutý a znovu zapnutý a migrace již synchronizovaného telefonu s resynchronizací i bez ní.

Copyright © 2026 Antonio Sanna

Logy ActiveSync v administraci

Stránka administrace Poste.io, **Logy ActiveSync**, pro čtení logů Z-Push (serveru ActiveSync, se kterým komunikují telefony) a pro nastavení jejich rotace. Bez ní jsou tyto logy jen uvnitř kontejneru a Poste.io je nikdy nerotuje: na serveru s několika telefony roste `z-push.log` o pár megabajtů měsíčně, donekonečna.



Doplněk je nezávislý na ostatních: Z-Push a jeho logy jsou součástí každého Poste.io, s kalendáři a kontakty přes ActiveSync i bez nich.

Stránka

Položka je v levém menu hned za **User logs**, jen pro superadministrátory.

- **Log**: `z-push.log` (vše, co Z-Push zapisuje na své úrovni logování, v Poste.io INFO) nebo `z-push-error.log` (jen jeho varování a chyby).
- **Úroveň**: všechny řádky, varování a chyby, nebo jen chyby (`ERROR` a `FATAL` Z-Push).
- **Účet**: účty nalezené v posledních 20 000 řádcích logu; vyberte jeden, chcete-li vidět jen jeho řádky.
- **Řádky** a **Hledat** (prostý text, bez rozlišování velikosti písmen).

Filtry se uplatní na celý soubor, poté se zobrazí posledních 25–500 odpovídajících řádků, nejnovější nahoře. Každý řádek ukazuje úroveň Z-Push (varování žlutě, chyby červeně), čas, účet a zprávu; zpráva, kterou Z-Push zapíše na dva řádky, se zobrazí jako jedna. Stránka se sama obnovuje každých 30 sekund.

Co tam obvykle najdete: telefon se špatným heslem (`AuthenticationRequiredException`), telefon, který žádá něco, co server nenabízí (`Searchtype 'GAL' is not supported`), IMAP odmítající spojení Z-Push (poštovní server nefunguje) a při každé synchronizaci řádek INFO *"SECURITY PROBLEM: insecure server advertised AUTH=PLAIN"*: Z-Push komunikuje s IMAP v Poste.io na `localhost` bez TLS, což je šum, ne riziko zvenčí.

Tlačítko **Vymazat** zde **není**, a to záměrně. Logy patří Z-Push (uživatel `www-data`, režim `0644`) a Poste.io je při každém startu vrací uživateli `www-data`; administrace běží jako `mail` a může je číst, ne měnit. Aby do nich administrace mohla zapisovat, musela by se změnit oprávnění, která nastavuje Poste.io. Malé je místo toho udržuje rotace.

Copyright © 2026 Antonio Sanna

Rotace logů

Rotace otevře nastavení: zapnuto/vypnuto, jak často se každý log rotuje (každý den, týden nebo měsíc), kolik starých kopií se uchovává, komprimovaných (1–12). Výchozí hodnoty jsou týdně pro oba a 4 kopie. Řádek pod nastavením ukazuje aktuální velikost obou logů a starých kopií.

Funguje stejně jako rotace logů Externích schránek. Administrace běží jako `mail` a nemůže zapisovat pravidla logrotate, která musí patřit rootovi, takže:

1. administrace uloží nastavení do `/data/var/activesync-logs/log-rotation.ini` (na svazku: přežijí nová sestavení i aktualizace), s vlastním tokenem CSRF a jen s hodnotami, které nabízí formulář;
2. skript roota, `/usr/local/bin/activesync-logs-logrotate`, z nich každou noc v 00:00, minutu před logrotate Poste.io, a při každém startu kontejneru vytvoří `/etc/logrotate.d/activesync-logs`. Změna provedená v administraci platí od příští noci.

Soubor s nastavením zapisuje uživatel s nižšími oprávněními a čte ho root: skript z něj nic nespouští a přijímá jen povolené hodnoty (`enabled` 0/1, `zpush_every` a `errors_every` daily/weekly/monthly, `keep` 1/2/4/8/12); cokoli jiného znamená výchozí hodnotu.

Pravidlo rotuje oba logy jako `www-data` (`su www-data www-data`) a znovu je vytvoří s režimem `0644` a vlastníkem `www-data`, jak je má Poste.io. Není potřeba `copytruncate`: Z-Push otevírá soubor znovu pro každý řádek, který zapisuje, takže hned pokračuje v novém souboru.

```
/data/var/activesync-logs/log-rotation.ini   settings (admin, mail, 0640)
/usr/local/bin/activesync-logs-logrotate     writes the rule (root)
/etc/cron.d/activesync-logs-logrotate        runs it at 00:00
/etc/cont-init.d/90-activesync-logs.sh       at every start: the settings folder, then the rule
/etc/logrotate.d/activesync-logs             the rule, used by logrotate at 00:01
```

Stávající instalace

Nic není třeba dělat ručně. `/data` je svazek, takže složka připravená v image by byla na instalaci, která už svazek má, zakrytá; složku s nastavením proto při každém startu kontejneru vytváří `/etc/cont-init.d/90-activesync-logs.sh` (spouštěcí skript, který jako root spouští init image, po skriptech samotného Poste.io), který zároveň zapíše pravidlo: od první noci se logy rotují s výchozím nastavením, dokud ho na stránce nezměníte.

Soubory

Soubor	Úloha
`activesync-logs/admin/src/Base/Controller/ActiveSyncLogsController.php`	stránka a ukládání nastavení
`activesync-logs/admin/templates/Base/ActiveSyncLogs/`	stránka a její tabulka (tabulka se znovu načítá každých 30 sekund)
`activesync-logs/activesync-logs.{en,cs}.yml`	překlady
`activesync-logs/activesync-logs-logrotate` + `.cron`	skript roota a jeho noční spouštění
`activesync-logs/90-activesync-logs.sh`	spouštěcí skript: složka s nastavením a pravidlo
`activesync-logs/activesync-logs-install.sh`	nainstaluje vše výše uvedené a přidá položku menu

`activesync-logs-install.sh` je jediná část, která se dotýká souboru Poste.io: přidá tři řádky do šablony menu hned za **User logs**. Odmítne se spustit, pokud tuto položku nenajde právě jednou a následovanou jejím `</li>`, takže aktualizace Poste.io, která menu změní, rozbije sestavení, místo aby stránku nechala bez její položky.

Odebrání

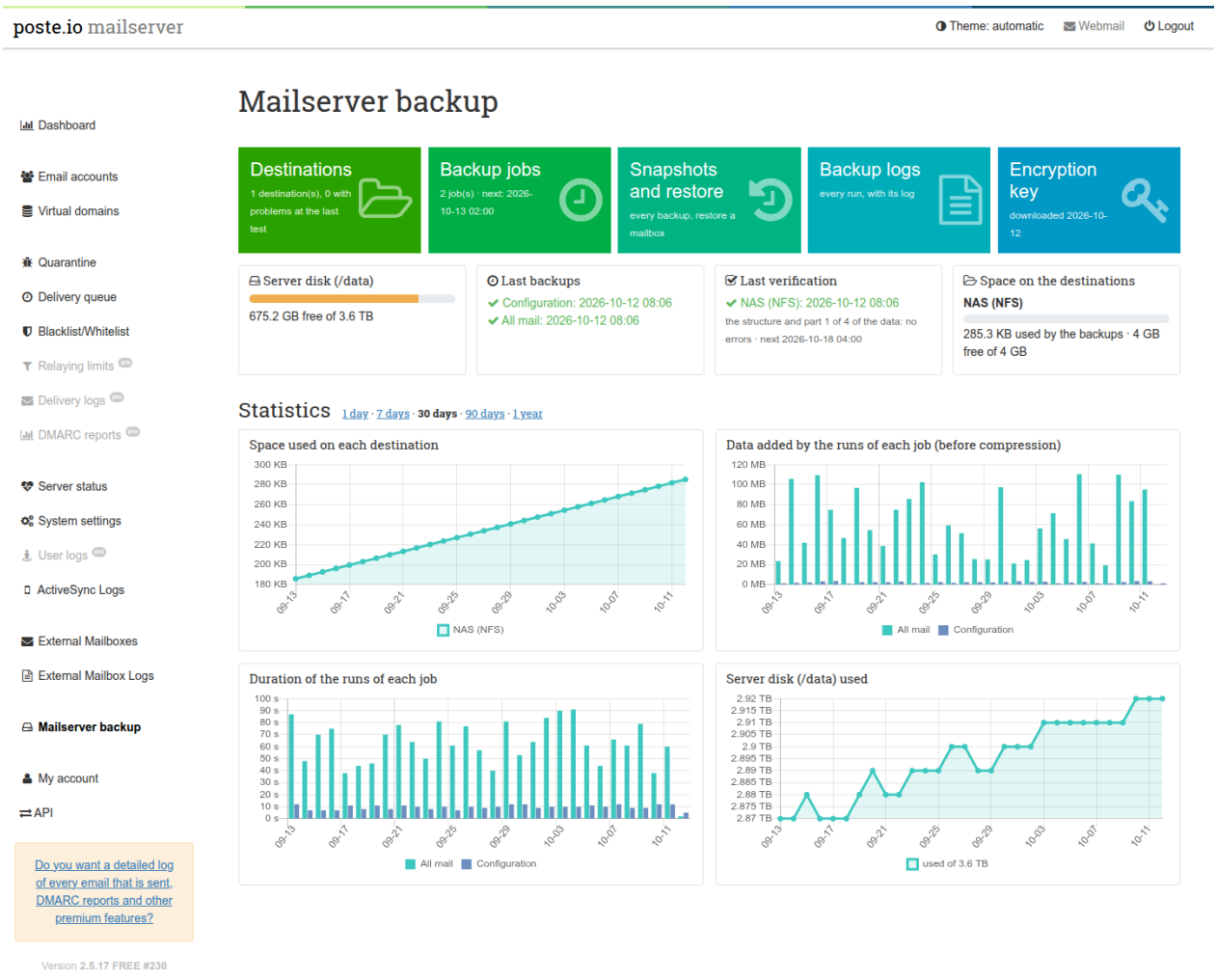
Sestavte image bez této sekce. Ve svazku zůstane `/data/var/activesync-logs` (nastavení) a v `/data/log/z-push` staré kopie vytvořené rotací (`*.log.1`, `*.log.N.gz`); pokud je už nepotřebujete, odstraňte je. Samotné logy Z-Push patří Poste.io a zůstanou, kde byly.

Copyright © 2026 Antonio Sanna

Zálohování a obnova

Šifrované plánované zálohy serveru Poste.io, vytvářené a kontrolované z administrace: konfigurace serveru, veškerá pošta nebo jen některé schránky, odesílané do jednoho nebo více cílů současně (SMB/Samba, SFTP, S3 nebo svazek připojený hostitelem, například NFS).

> ***Stav tohoto doplňku.*** *Tato stránka popisuje, co je nyní v repozitáři. Doplněk vzniká po etapách; co některá etapa ještě nedělá, je uvedeno tam, kde na tom záleží (například upozornění, která přijdou jako další).*



Doplněk je nezávislý na ostatních doplňcích i na čemkoli mimo kontejner: nepotřebuje na hostiteli žádný zálohovací systém. Pokud hostitel už zálohuje celý stroj (snímek hypervizoru, záloha VM), tento doplněk přidává to, co ony poskytnout nemohou: zálohy samotného poštovního serveru, mimo stroj, šifrované, časté, s možností obnovit jednotlivé schránky.

- Co se zálohuje
- Jak to funguje
- Šifrovací klíč
- Cíle
- Úlohy zálohování
- Sledování záloh
- Obnova
- Čtení záloh bez serveru
- Znovuvytvoření celého serveru
- Osvědčené postupy
- Záloha běžícího serveru

Copyright © 2026 Antonio Sanna

- Zabezpečení
- Pokud je poštovní server kompromitován
- Soubory a složky
- Řešení problémů
- Omezení
- Odebrání

Co se zálohuje

Úloha zálohuje **konfiguraci serveru**, **poštu**, nebo obojí.

Konfigurace (několik megabajtů):

Co	Odkud
účty, domény, aliasy, hesla (hašovaná), limity	`/data/users.db`, kopírováno za provozu vlastním zálohovacím příkazem SQLite a poté zkontrolováno kontrolou integrity
kalendáře a adresáře kontaktů (DAV)	`/data/dav.db`, stejným způsobem
nastavení webmailu, identity, adresáře kontaktů	`/data/roundcube/roundcube.db` (patří webmailu: kopíruje se databáze a její žurnál WAL (write-ahead log), znovu, pokud se mezitím změnily, a kopie se zkontroluje)
nastavení serveru	`/data/server.ini`, `/data/instance-id`
certifikáty TLS (i Let's Encrypt)	`/data/ssl`
klíče DKIM jednotlivých domén	`/data/domains/<domain>/private`, `public`, `selector`
učení spamového filtru (Bayes v rspamd, uložený v redis)	`/data/redis/dump.rdb`
stavy zařízení ActiveSync	`/data/zpush`
nastavení doplňků a Poste.io v `/data/var`	vše, co tam je, kromě antivirové databáze (stáhne se znovu) a vlastní složky tohoto doplňku
seznam zpráv v zálohách (viz Obnova)	`/data/var/backup/engine/index.db`, kopírováno stejně jako databáze

Pošta: buď veškerá (`/data/domains`, každá schránka se svými složkami a filtry a navíc zprávy zadržené v karanténě), nebo jen schránky vybrané v úloze (celá jejich domovská složka: pošta a filtry).

Nikdy se nezálohuje: šifrovací klíč a nastavení doplňku (obsahují hesla cílů: viz Zabezpečení), záznamy, fronta odchozí pošty, antivirová databáze, mezipaměť map pravidel spamového filtru (stahují se znovu) a klíčenky PGP webmailu (`roundcube/enigma`: čitelné jen pro webmail; uživatelé, kteří v Roundcube používají PGP, si své klíče exportují odtud). Obecněji: soubor konfigurace, který uživatel `mail` nemůže číst, se záměrně vynechá a záznam každého běhu uvádí, kolik takových souborů bylo.

Jak to funguje

Dvě části:

- **administrace** ukládá nastavení a požadavky;
- **zálohovací modul** (`/usr/local/bin/poste-backup`, spouštěný cronem každou minutu) provádí zálohy. Čte nastavení a před použitím čehokoli z nich je **znovu ověří**.

Obě části běží jako uživatel `mail`, pod kterým už běží administrace Poste.io a kterému patří pošta. Záloha nikdy neběží jako root: ať nastavení administrace říká cokoli, modul nemůže udělat víc, než by mohla sama administrace (viz Zabezpečení).

Zálohy vytváří restic *(https://restic.net)*: šifrované, s deduplikací (soubor, který už je uložen, se nikdy neukládá znovu, takže hodinová záloha velkého serveru uloží jen to, co se změnilo), kontrolované podle obsahu. Ke vzdáleným cílům se přistupuje přes rclone *(https://rclone.org)*. Oba nástroje pocházejí z balíčků Debianu v obrazu.

Každý **cíl** má svůj vlastní **repozitář** restic a každá úloha zapisuje do každého ze svých cílů úplnou, nezávislou kopii: ztrátou jednoho cíle se nic neztratí. Jeden cíl může sloužit více úlohám.

Každý běh úlohy:

1. s poštou: aktualizuje seznam zpráv (viz Seznam zpráv); poté zkopíruje databáze v konzistentním stavu a kopie zkontroluje;

Copyright © 2026 Antonio Sanna

2. pro každý cíl: cíl zkontroluje (místní složka musí být připojený svazek, ne na disku `/data`), vytvoří v něm repozitář **jen tehdy, je-li složka prázdná** (do složky s jinými soubory se nikdy nezapisuje), provede zálohu a poté odstraní staré snímky této úlohy podle jejích pravidel uchovávání;
3. zaznamená výsledek pro administraci (stav a historii).

Cíl, který selže, nezastaví ostatní. Vždy jen jeden běh: dlouhá záloha nechá další minuty čekat (zámek), nikdy se nespustí dvakrát. Po úlohách, které jsou na řadě, spustí modul ověření, která jsou na řadě.

Výsledek běhu na cíli je **v pořádku**, **varování** nebo **chyba**:

- *v pořádku*: snímek je úplný. Zpráva přesunutá nebo smazaná v době, kdy záloha četla schránky (zpráva přesunutá do jiné složky, změněný příznak: Dovecot dá souboru nový název), nevadí: běh v poznámce uvede, kolik takových zpráv bylo, a další záloha je zachytí tam, kde jsou nyní.
- *varování*: snímek byl vytvořen, ale některý soubor nešlo přečíst (například kvůli oprávněním), schránka vybraná v úloze už neexistuje, nebo nešlo odstranit staré snímky. Běh uvede, co se stalo; podívejte se do jeho záznamu.
- *chyba*: na tomto cíli není žádný snímek.

Šifrovací klíč

Každá záloha je zašifrována klíčem vytvořeným při prvním startu kontejneru (`/data/var/backup/settings/restic.key`, 48 náhodných bajtů). **Bez klíče nelze žádnou zálohu přečíst**, ani vy, ani nikdo, kdo by se zmocnil cíle.

Klíč není součástí žádné zálohy (záloha chráněná klíčem uloženým uvnitř ní by byla k ničemu). **Stáhněte si ho** v *Zálohy poštovního serveru* > *Šifrovací klíč* a uložte ho mimo server, tam, kde máte ostatní hesla. Administrace zobrazuje varování, dokud nebude alespoň jednou stažen.

Klíč se nikdy nenahrazuje: pokud se soubor smaže, při dalším startu se vytvoří nový a tento nový klíč nedokáže přečíst zálohy vytvořené předtím. Místo toho vraťte zpět starý klíč (stejný soubor, vlastník `mail`, režim `0600`).

Cíle

Zálohy poštovního serveru > *Cíle záloh* > *Přidat cíl*. Tlačítko **Otestovat** vedle *Uložit* vyzkouší hodnoty formuláře **dříve**, než se uloží: ověří, že je cíl dosažitelný, že je složka prázdná nebo už je repozitářem, zjistí volné místo a zapíše a odstraní malý soubor (server REST volné místo neuvádí a nic se do něj nezapisuje: tam test ověří certifikát, uživatele a repozitář). Stránka se znovu nenačítá, takže zadané heslo zůstane ve svém poli. Uložený cíl lze znovu otestovat z jeho stránky. Používejte **jednu složku pro každý poštovní server**.

Dejte zálohám v cíli vlastní účet s přístupem jen k této složce (viz Osvědčené postupy).

SMB / Samba

Server, sdílená složka, složka uvnitř sdílené složky, uživatel, heslo a doména, pokud ji server vyžaduje. Funguje se Sambou, se sdílenými složkami Windows a se sdílenými složkami NAS (SMB 2 nebo novější). Volné a celkové místo sdílené složky se čte pomocí `smbclient`: klient SMB v rclone 1.60 je uvádí vydělené faktorem, který závisí na serveru (na Sambě a na NAS QNAP polovinu), takže jeho čísla se nepoužívají.

SFTP

Server, port, uživatel, složka a přihlášení **heslem** nebo **klíčem**:

- *klíčem*: do formuláře se nic nevkládá. Při uložení cíle pro něj administrace vytvoří pár klíčů (ed25519), uchová soukromý klíč a zobrazí veřejný klíč spolu s připraveným příkazem, který se na serveru spustí jako daný uživatel (vytvoří `~/.ssh` s režimem 700 a přidá klíč do `~/.ssh/authorized_keys` s režimem 600), a s tlačítkem pro jeho zkopírování. Na NAS musí mít uživatel domovskou složku a zapnutou službu SFTP, jinak se klíč nenačte. Poté cíl otestujte;
- **vlastní klíč SSH serveru musí být potvrzen** dříve, než se cokoli odešle. První test ukáže otisky, které server nabízí; porovnejte je s otisky serveru (`ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub` na serveru) a potvrďte. Od té chvíle jiný klíč zálohy zastaví, místo aby je odeslal komukoli, kdo odpoví. Změna serveru nebo portu vyžaduje nové potvrzení.

Složka je povinná (repozitář nesmí skončit mezi vlastními soubory uživatele).

S3

Koncový bod (`https://...`), region (pokud ho poskytovatel vyžaduje), bucket, složka uvnitř bucketu, přístupový klíč a tajný klíč. Funguje s Amazon S3 a se službami kompatibilními s S3 (MinIO, Backblaze B2, Wasabi, Ceph, …). Pokud bucket neexistuje, vytvoří se, má-li k tomu klíč oprávnění. S3 neuvádí volné místo.

Copyright © 2026 Antonio Sanna

Svazek připojený hostitelem (NFS, druhý disk, …)

Kontejner nemůže sám připojovat síťové souborové systémy (potřeboval by oprávnění, která by ho učinila mnohem méně bezpečným), takže sdílenou složku připojí hostitel a předá ji kontejneru jako svazek pod `/backup`, `/mnt`, `/srv` nebo `/media`. Například s Docker Compose a exportem NFS:

```
services:
  mailserver:
    # ...
    volumes:
      - ./data:/data
      - nfs-backup:/backup/nfs

volumes:
  nfs-backup:
    driver: local
    driver_opts:
      type: nfs
      o: "addr=nas.example.lan,rw,nfsvers=4"
      device: ":/export/mail-backup"
```

Poté přidejte cíl typu *Složka* s cestou `/backup/nfs`. Modul odmítne složku, která **není přípojným bodem** (záloha by zůstala uvnitř kontejneru a zmizela by s ním), která je **na stejném disku jako `/data`** (nepřežila by ztrátu tohoto disku), nebo do které **nemůže zapisovat**: zálohy zapisuje uživatel `mail`, **uid 8, gid 8**. Přidělte složku svazku tomuto uživateli (u exportu NFS: adresář na serveru vlastněný 8:8 a export, který tohoto uživatele nepřemapovává (squash)).

Server REST, jen přidávání (`rest-server`)

Vlastní server nástroje restic, `rest-server` *(https://github.com/restic/rest-server)*, spuštěný s `--append-only`, **odmítá mazat**: tento poštovní server tam může zálohy jen přidávat a ani ten, kdo ho ovládne, tam zálohy nemůže smazat ani přepsat (viz Pokud je poštovní server kompromitován). Staré zálohy se pak odstraňují **na straně serveru**, podle vlastního plánu serveru a s klíčem. Pouze HTTPS.

Na NAS (nebo jiném stroji) s Dockerem (QNAP Container Station, Synology Container Manager, …): složka pro data, certifikát vystavený na název, který bude používat poštovní server (název hostitele nebo adresa IP), a uživatel:

```
mkdir -p /share/backup/rest && cd /share/backup/rest
# a certificate for the name used by the mail server (here nas.example.lan; for an
# address use subjectAltName=IP:192.0.2.10), valid ten years
openssl req -x509 -newkey ed25519 -nodes -keyout key.pem -out cert.pem -days 3650 \
    -subj "/CN=nas.example.lan" -addext "subjectAltName=DNS:nas.example.lan"
openssl x509 -in cert.pem -noout -fingerprint -sha256      # compare it in the admin
docker run -d --name rest-server --restart unless-stopped -p 8000:8000 \
    -v /share/backup/rest:/data -e OPTIONS="--append-only --tls --tls-cert /data/cert.pem --tls-key /data/key.pem" \
    restic/rest-server:0.12.1
docker exec -it rest-server create_user mailserver          # asks for the password
```

V administraci: *Přidat cíl*, typ **Server REST**, název serveru, port `8000`, název repozitáře (například `mailserver`), uživatel a heslo, zaškrtnuté **Přijímá jen nové zálohy (jen přidávání)**. První test tlačítkem **Otestovat** ukáže certifikát, který server předkládá: porovnejte jeho otisk s otiskem vypsaným výše a pak ho potvrďte. Od té chvíle je přijímán jen tento certifikát, i při vlastních kontrolách administrace (jiný je odmítnut, stejně jako u klíčů SSH u SFTP, i kdyby byl podepsán veřejnou certifikační autoritou), a uživatel a heslo se nikdy neobjeví na příkazovém řádku (restic je dostává v proměnných prostředí). Certifikát s prošlou platností je odmítnut.

Nový certifikát (obnovený, nebo nový server pod stejným názvem): zálohy do tohoto cíle se zastaví s hlášením „certifikát serveru se změnil“. Test tlačítkem **Otestovat** na stránce cíle pak ukáže nový certifikát s jeho otiskem, který je třeba porovnat a potvrdit stejně jako ten první; předtím se na server nic neodešle. Adresa může být název nebo adresa IP (i IPv6); certifikát ji musí uvádět.

Na cíli s režimem jen přidávání tento server nikdy neodstraňuje snímky: pravidla uchovávání úloh se tam nepoužívají a *Odstranit tyto zálohy* se nenabízí (a modul to odmítne). Odstraňujte je na straně serveru, například jednou týdně, se stejnými pravidly jako úloha a s **klíčem** (poslední řádek souboru staženého z administrace), uloženým **mimo** složku, kterou server REST obsluhuje:

```
tail -n 1 poste-backup-key.txt > /share/backup/restic.key && chmod 600 /share/backup/restic.key
docker run --rm -v /share/backup/rest:/data -v /share/backup/restic.key:/key:ro -e RESTIC_PASSWORD_FILE=/key \
    restic/restic:0.14.0 -r /data/mailserver forget --host poste --group-by host,tags \
    --keep-daily 7 --keep-weekly 4 --keep-monthly 6 --prune
```

Soubor s klíčem nechte čitelný jen pro správce NAS. Server REST volné místo neuvádí (stejně jako S3). Pokud server neběží s `--append-only`, zaškrtnutí zrušte: tento server pak v tomto cíli odstraňuje staré snímky jako v kterémkoli jiném cíli.

Copyright © 2026 Antonio Sanna

Ověření

Záloha, kterou nelze přečíst, se odhalí až ve chvíli, kdy je potřeba, pokud se nepřečte dříve. Každý cíl se **ověřuje** podle vlastního plánu (na jeho stránce, *Ověření*): restic zkontroluje strukturu repozitáře a **znovu přečte část uložených dat**, pokaždé jinou, takže se za určitý počet ověření přečte všechno jednou. Ve výchozím nastavení: **každou neděli ve 04:00, čtvrtina dat** (všechna data přečtena zhruba za čtyři týdny). Může to být každý den, týden nebo měsíc, nebo nikdy, a pokaždé od všech dat až po 1/52 z nich.

Poškozený nebo chybějící soubor na cíli se odhalí právě takto: samotná kontrola struktury data nečte, čtení ano (testy změní jeden bajt datového souboru na cíli a očekávají, že ho ověření najde). Výsledek se zobrazí na přehledu (*Poslední ověření*, a mezi problémy, pokud ověření našlo chyby), na stránce cílů a na stránce cíle s posledními ověřeními; *Ověřit nyní* spustí ověření v příští minutě. První plánované ověření nového cíle proběhne v jeho dalším termínu, ne hned. Během ověření neběží žádná záloha: ta, která je na řadě, se spustí hned potom (restic po dobu kontroly repozitář uzamkne); ověření se zastaví nejpozději po 6 hodinách. Část, ve které byly nalezeny chyby, se při dalším ověření čte znovu, dokud není čistá: chyba do té doby zůstává na přehledu a nezmizí jen proto, že jiná část byla v pořádku. Ověření, které nemohlo doběhnout do konce (například proto, že cíl nebyl dosažitelný), se zobrazí jako neúspěšné a zkusí se znovu o hodinu později.

Úlohy zálohování

Zálohy poštovního serveru > Úlohy zálohování > Přidat úlohu zálohování:

- **Co**: konfigurace serveru a žádná pošta / veškerá pošta / jen některé schránky;
- **Kdy**: každou hodinu (v určitou minutu), každý den, každý týden (v určitý den) nebo každý měsíc (den 1 až 28) v daný čas, v **časovém pásmu plánů** (viz níže);
- **Cíle**: jeden nebo více, každý dostane úplnou kopii;
- **Uchovávat**: kolik hodinových, denních, týdenních a měsíčních snímků uchovávat. Po každém běhu se odstraní starší snímky **této úlohy**: pro každou hodinu, den, týden a měsíc se ponechá poslední snímek, až do těchto počtů (`forget --keep-*` nástroje restic, pak `--prune`). Alespoň jedno číslo musí být větší než 0. Příklad: 24 hodinových, 7 denních, 4 týdenní, 12 měsíčních = každou hodinu po dobu jednoho dne, každý den po dobu týdne, každý týden po dobu měsíce, každý měsíc po dobu roku.

Časové pásmo plánů, které se volí pod seznamem úloh: v tomto pásmu jsou hodiny úloh a časy zobrazené na stránkách záloh. Bez volby je to časové pásmo kontejneru (jeho `TZ`, jinak `/etc/timezone`), což je v obrazu Poste.io UTC, pokud není nastaveno `TZ`. Volba mění jen zálohy: server a jeho záznamy si ponechávají svůj vlastní čas (obvykle UTC, což je pro záznamy dobře: při změně času žádná hodina nechybí ani se neopakuje). Vlastní záznamy záloh zapisují každý čas s jeho posunem (`+02:00`), takže je lze porovnat s ostatními. Po změně pásma se úloha, jejíž čas v novém pásmu dnes už uplynul, může jednou spustit, a to v příští minutě.

Úloha se spustí, když přijde její čas; **Spustit nyní** (nebo *Uložit a spustit nyní*) ji spustí do minuty. Běh zmeškaný kvůli vypnutému serveru proběhne jednou, jakmile server znovu běží (ne jednou za každou zmeškanou hodinu). Nová úloha se nespustí pro čas, který už v okamžiku jejího vytvoření uplynul.

Stránka u každé úlohy ukazuje poslední běh na každém cíli: kolik nových dat uložil, jak dlouho trval, nebo chybu, s odkazem na jeho záznam.

Smazání úlohy (nebo cíle) ji odebere jen z nastavení: zálohy, které vytvořila, zůstávají v cílech, šifrované, a lze je stále obnovit. Žádná pravidla uchovávání je už neprořeďují: odstraňte je v části *Snímky a obnova*, až nebudou potřeba (viz Odstranění záloh úlohy).

Sledování záloh

- **Přehled**: nalezené problémy (neúspěšný běh, nikdy nestažený klíč, neběžící modul), právě běžící úloha s jejím **průběhem** (procenta, hotová data a soubory z celkového počtu, poté odstraňování starých snímků a měření místa), disk serveru, poslední zálohy a **místo v každém cíli**: kolik tam zálohy zabírají a, pokud to server uvádí, volné místo (S3 ne). Místo se měří po každé záloze. Pruhy disku a cílů se zbarví **žlutě** od 80 % obsazenosti a **červeně** od 90 %: *Upozornění na místo* na stránce cílů nastavuje tyto dva limity (jsou to zároveň limity upozornění příští etapy).
- **Poslední ověření** každého cíle a kdy bude další (viz Ověření).
- **Statistiky** za poslední den (jedna hodnota za hodinu), nebo za 7 dní, 30 dní, 90 dní či rok (jedna hodnota za den): obsazené místo na každém cíli a na disku serveru (poslední měření dne), data přidaná běhy každé úlohy (běh zapisuje stejná data do každého ze svých cílů: počítají se jednou) a jak dlouho běhy trvaly. Vycházejí z historie běhů (ve výchozím nastavení uchovávané 400 dní, viz níže); hodnoty místa jsou k dispozici od této verze.
- Přehled, úlohy a záznamy se **aktualizují samy**: každých několik sekund, když běží úloha, jinak každých 15 sekund; stránka se znovu načte, když úloha začne nebo skončí.
- **Záznamy záloh**: jeden řádek na běh a cíl, nejnovější nahoře, s odkazem na záznam běhu (co modul udělal, v angličtině, s výstupem nástroje restic). Pod seznamem se nastavuje, jak dlouho se uchovávají: záznamy běhů ve výchozím nastavení 90 dní a historie (jeden řádek na běh, pro statistiky) 400 dní, v rozsahu od 1 do 3650 dní. Starší se odstraňují po každém běhu a jednou denně; samotných záloh se to netýká.

Copyright © 2026 Antonio Sanna

- Stránky administrace jsou v **angličtině a češtině** (dva jazyky Poste.io, volené podle prohlížeče), včetně hlášení testů, validace a běhů. Samotné záznamy zůstávají v angličtině.

Zatím nejsou k dispozici žádná upozornění (e-mail, Telegram, webhook, stav pro monitoring): jsou to další etapy tohoto doplňku. Do té doby je neúspěšná záloha vidět jen v administraci, takže se do ní čas od času podívejte, nebo sledujte `poste-backup status` (JSON) ze svého monitoringu.

Obnova

Zálohy poštovního serveru > *Snímky a obnova* vypisuje snímky cíle (načtené z něj při otevření stránky), nejnovější nahoře: kdy, která úloha, co obsahuje (konfiguraci, veškerou poštu, některé schránky). Odtud:

- **Obnovit schránku…** (snímek s poštou), ve třech krocích na jedné stránce:
- **schránka**: schránky ve snímku, které obsahují zprávy, s jejich počtem; ta, která už na tomto serveru není, je označena. Prázdné složky (Poste.io vytváří jednu pro každou adresu, včetně přesměrování) se jen počítají, v řádku pod seznamem: není v nich co obnovit;
- **co**, jedním ze dvou způsobů:
  - **celé složky, nebo zprávy odpovídající filtru**: složky této schránky s jejich zprávami, všechny zaškrtnuté (celá schránka), nebo jen některé z nich; a pokud chcete jen některé zprávy, filtr: přijato od / do (data tak, jak je počítá server, oba dny včetně), odesílatel obsahuje, předmět obsahuje (na velikosti písmen nezáleží). Složky a filtr lze kombinovat, například *jen Sent, odesílatel customer.com, od 1. do 7. října*;
  - **jednotlivé zprávy**: zprávy ve snímku, nejnovější nahoře, 50 na stránku, s datem přijetí, složkou, odesílatelem a předmětem; vyberte složku nebo všechny, hledejte podle odesílatele nebo předmětu (na velikosti písmen nezáleží, na diakritice také ne), zaškrtněte zprávy, které chcete, napříč stránkami i hledáními (až 1 000 najednou);
- **kam**: schránka tohoto serveru, do které se má obnovit (stejná se vybere, pokud existuje; jinak se nevybere žádná a nic se nespustí, dokud ji nevyberete), název **nové složky** (*Restored <čas snímku>*, pokud ho nezměníte) a kde ji vytvořit: na nejvyšší úrovni schránky, nebo uvnitř některé z jejích složek.

Před potvrzením stránka uvede, kam zprávy půjdou. Uloží se do nové složky se složkami snímku (… / *INBOX*, … / *Sent*), se svými příznaky (přečteno, …) a s přihlášeným odběrem, aby je poštovní programy zobrazily: **nic, co už ve schránce je, se nezmění**. Pokud je název obsazený, složka dostane *(2)*. Zprávy se nikdy nevracejí do svých původních složek: import Dovecotu nerozpozná zprávu, která tam stále je, takže každá z nich by se objevila dvakrát. Co potřebujete, přesuňte z obnovené složky svým poštovním programem a pak ji smažte.

Jednotlivé zprávy se stejným způsobem uloží do nové složky, každá ve své složce (… / *INBOX*, … / *Sent*), a jen ty.

Obnova běží na pozadí (modul ji převezme do minuty: stránka to ukáže a pak i její výsledek a záznam). Z cíle se čtou jen vybrané složky, do pracovní složky na `/data`, a pak je importuje Dovecot, takže se zapisují dvakrát: modul nejprve sečte jejich velikost a obnovu odmítne, pokud na `/data` není místo pro dvojnásobek plus 256 MB (plný disk by zastavil poštu). Zprávy se započítávají do kvóty schránky. Symbolické odkazy v obnovených složkách se před importem odstraní, nikdy se jimi neprochází. - **Obnovit server…**: celý server se obnovuje na **novou instalaci**, ne na běžícím serveru (jeho soubory by se nahrazovaly za provozu). Stránka ukazuje příkazy ze sekce Znovuvytvoření celého serveru vyplněné pro tento cíl a tento snímek (jen u snímků s konfigurací); hesla se zadávají při spuštění příkazů a nikdy se nezobrazují. U SFTP jsou klíče serveru ty, které byly potvrzeny v administraci, a cíl, který se přihlašuje klíčem, potřebuje váš vlastní klíč: pár klíčů administrace v zálohách není.

Seznam zpráv

Snímek obsahuje soubory: odesílatel, předmět a datum každé zprávy pocházejí ze seznamu, který modul vede na serveru, `/data/var/backup/engine/index.db` (čitelný jen pro `mail`, stejně jako zbytek `/data/var/backup`). Před každým během úlohy s poštou modul projde soubory schránek, přečte hlavičku souborů, které přibyly od posledního běhu (odesílatel, předmět; datum přijetí a velikost se berou ze souboru), a poznamená si zprávy, které už ve své schránce nejsou. První běh přečte každou zprávu jednou; další běhy čtou jen nové. Problém se seznamem se zapíše do záznamu běhu a nikdy nezpůsobí selhání zálohy.

Zpráva se identifikuje názvem svého souboru v Maildiru bez části s příznaky: ten je stejný v každém snímku i po přesunu do jiné složky. Zpráva, která už ve své schránce není, zůstává v seznamu, protože ji starší snímky stále obsahují; odstraní se, jakmile ji už nemůže obsahovat žádný snímek uchovávaný na žádném cíli (opustila svou schránku před nejstarším snímkem s poštou na každém cíli úloh s poštou).

Seznam je součástí zálohy konfigurace jako konzistentní kopie (krok 4 sekce Znovuvytvoření celého serveru ho vrátí zpět). Zpráva, kterou seznam nezná, například smazaná dříve, než se seznam začal vést, se přesto zobrazí a lze ji obnovit: s datem a velikostí, bez odesílatele a předmětu. Upozorňujeme, že odesílatel a předmět smazané zprávy zůstávají v seznamu (a v zálohách konfigurace) tak dlouho, dokud samotnou zprávu může obsahovat některá záloha.

Copyright © 2026 Antonio Sanna

Pokud se seznam někdy poškodí, modul ho odloží stranou (`index.db.damaged`) a vytvoří ho znovu ze schránek a záloha konfigurace pokračuje bez něj: seznam nikdy nezpůsobí selhání zálohy. První běh přečte hlavičku každé zprávy; na velmi velkém serveru se zastaví po 20 minutách a zbytek přečtou další běhy (záznam uvádí, kolik jich ještě zbývá).

Stránka přečte soubory snímku jednou a uchová je v mezipaměti (`/var/cache/poste-backup/listing.db`, posledních pět prohlížených schránek), takže stránky a hledání odpovídají okamžitě.

Odstranění záloh úlohy

Pod seznamem *Zálohy jednotlivých úloh* uvádí počet snímků každé úlohy na tomto cíli, i úloh, které už neexistují (*smazaná úloha*). **Odstranit tyto zálohy** (po zaškrtnutí potvrzení) odstraní z tohoto cíle všechny snímky této úlohy a uvolní data, která obsahují jen ony (`forget --prune` nástroje restic); zálohy ostatních úloh zůstanou nedotčeny. Nelze to vrátit zpět. Úloha, která stále existuje, vytvoří při dalším běhu nové zálohy.

Z příkazového řádku

`poste-backup restic <destination id> <restic command>` spustí restic nad repozitářem cíle s jeho nastavením a klíčem. Identifikátory jsou v adresách administrace (`/admin/backup/destinations/<id>/edit`) a v `/data/var/backup/settings/config.json`.

```
# the snapshots on a destination
docker exec mailserver poste-backup restic 0a1b2c3d4e5f snapshots

# what is in the latest snapshot of a job
docker exec mailserver poste-backup restic 0a1b2c3d4e5f ls latest --tag job:<job id>

# one mailbox back, into a separate folder (nothing is overwritten)
docker exec mailserver poste-backup restic 0a1b2c3d4e5f restore latest \
    --target /tmp/restore --include /data/domains/example.com/alice
```

Obnovená schránka je pak v `/tmp/restore/data/domains/example.com/alice`; zprávy, které potřebujete, zkopírujte zpět do `Maildir` schránky (vlastník `mail`) nebo do její nové složky, například:

```
docker exec mailserver sh -c 'H=/data/domains/example.com/alice; U=alice@example.com
  mkdir -p "$H/Maildir/.Restored/cur" "$H/Maildir/.Restored/new" "$H/Maildir/.Restored/tmp" &&
  cp -a "/tmp/restore$H/Maildir/cur/." "$H/Maildir/.Restored/cur/" &&
  chown -R mail:mail "$H/Maildir/.Restored" &&
  doveadm force-resync -u "$U" "*" && doveadm mailbox subscribe -u "$U" Restored'
```

Tím se zkopírují zprávy doručené pošty (`Maildir/cur`); ostatní složky jsou `Maildir/.<name>/cur` (Poste.io ukládá poštu v uspořádání Maildir++: `.Sent`, `.Junk`, …). Tento postup spouštějí testy doplňku přesně tak, jak je zde napsán.

Databáze jsou ve snímku jako konzistentní kopie pod `/data/var/backup/engine/staging/` (`users.db`, `dav.db`, `roundcube.db`).

`poste-backup` vždy běží jako uživatel `mail` (spuštěný jako root se přepne), takže obnovené soubory patří uživateli `mail`. To je správně pro poštu a pro vlastní soubory Poste.io, které vlastní `mail`; souboru, který na serveru patří jinému uživateli (například `roundcube.db` webmailu, který vlastní `www-data`), musí root po zkopírování na místo vrátit vlastníka (`chown www-data:www-data ...`).

Když je ztracen samotný server, viz další dvě sekce.

Čtení záloh bez serveru

Zálohy jsou běžné repozitáře restic *(https://restic.net)*. Lze je číst z libovolného stroje s **restic 0.14** nebo novějším a **rclone 1.60** nebo novějším (na Debianu nebo Ubuntu: `apt install restic rclone openssh-client`; oba jsou také v obrazu sestaveném z tohoto repozitáře): když je poštovní server ztracen, nebo když chcete nahlédnout do záloh, aniž byste se ho dotkli.

Co potřebujete:

- **soubor s klíčem** stažený v *Zálohy poštovního serveru > Šifrovací klíč*;
- **cíl** tak, jak byl zadán v administraci: jeho typ, server, sdílenou složku nebo bucket, složku, uživatele a heslo (nebo přístupový klíč a tajný klíč). Nastavení doplňku v zálohách nejsou, protože obsahují tato hesla: poznamenejte si je spolu s klíčem.

Příkazy jsou pro bash. Nejprve klíč, kterým je poslední řádek staženého souboru (soubor jako celek klíčem není):

```
```

Copyright © 2026 Antonio Sanna

```
tail -n 1 poste-backup-key-mail.example.com.txt > restic.key
chmod 600 restic.key
export RESTIC_PASSWORD_FILE=$PWD/restic.key
```

Pak cíl, jeden z následujících. rclone ho čte z proměnných (`RCLONE_CONFIG_BK_...` definují remote s názvem `BK`), s heslem upraveným pomocí „obscure“, jak rclone vyžaduje; `read -rsp` se na heslo zeptá, aniž by ho zobrazil. `RESTIC_REPOSITORY` je složka cíle: u SMB sdílená složka a složka v ní, u SFTP složka, u S3 bucket a složka v něm.

SMB / Samba (přidejte `RCLONE_CONFIG_BK_DOMAIN=...`, pokud server vyžaduje doménu):

```
export RCLONE_CONFIG_BK_TYPE=smb RCLONE_CONFIG_BK_HOST=nas.example.lan RCLONE_CONFIG_BK_USER=mailbackup
read -rsp 'Password: ' P; echo
export RCLONE_CONFIG_BK_PASS=$(printf '%s\n' "$P" | rclone obscure -) && unset P
export RESTIC_REPOSITORY=rclone:BK:share/folder
```

SFTP: klíč SSH serveru se nejprve ověří, stejně jako to dělá administrace: načtěte ho, porovnejte jeho otisky s otisky serveru (`ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub` na serveru) a předejte ho rclone. Server, který ukáže jiný klíč, je pak odmítnut.

```
ssh-keyscan -p 22 backup.example.com > known_hosts
ssh-keygen -lf known_hosts
export RCLONE_CONFIG_BK_TYPE=sftp RCLONE_CONFIG_BK_HOST=backup.example.com RCLONE_CONFIG_BK_PORT=22 \
       RCLONE_CONFIG_BK_USER=mailbackup RCLONE_CONFIG_BK_KNOWN_HOSTS_FILE=$PWD/known_hosts
read -rsp 'Password: ' P; echo
export RCLONE_CONFIG_BK_PASS=$(printf '%s\n' "$P" | rclone obscure -) && unset P
export RESTIC_REPOSITORY=rclone:BK:/srv/backup/mail
```

Cíl, který se přihlašoval klíčem: pár klíčů vytvořený administrací zůstal na ztraceném serveru (v zálohách nikdy není). Přihlaste se heslem účtu, nebo přidejte k účtu na serveru SFTP svůj vlastní klíč a použijte ho místo hesla: `export RCLONE_CONFIG_BK_KEY_FILE=~/.ssh/id_ed25519`.

S3 (přidejte `RCLONE_CONFIG_BK_REGION=...`, pokud poskytovatel vyžaduje region):

```
export RCLONE_CONFIG_BK_TYPE=s3 RCLONE_CONFIG_BK_PROVIDER=Other RCLONE_CONFIG_BK_ENDPOINT=https://s3.example.com \
       RCLONE_CONFIG_BK_ACCESS_KEY_ID=AKIAEXAMPLE
read -rsp 'Secret key: ' P; echo
export RCLONE_CONFIG_BK_SECRET_ACCESS_KEY=$P && unset P
export RESTIC_REPOSITORY=rclone:BK:bucket/folder
```

Složka (svazek připojený hostitelem): připojte na tomto stroji stejnou sdílenou složku nebo disk a pak:

```
export RESTIC_REPOSITORY=/mnt/mail-backup
```

Server REST: certifikát serveru v souboru (`cert.pem` vytvořený na serveru, nebo ten, který administrace ukazuje v *Obnovit server*) a každý příkaz restic níže s `--cacert server.pem` (heslo obsahující `@ : / %` nebo mezery se do adresy zapisuje v procentovém kódování: `@` je `%40`):

```
read -rsp 'Password of the REST server: ' P; echo
export RESTIC_REPOSITORY="rest:https://mailserver:$P@nas.example.lan:8000/mailserver" && unset P
```

Pak restic, jako obvykle:

```
restic snapshots --group-by tags --latest 1    # the latest snapshot of each job
restic snapshots                               # all of them: time, paths, tags
restic ls <snapshot>                           # what one of them holds
restic restore <snapshot> --target ./restore --include /data/domains/example.com/alice
restic check                                   # the repository is intact
```

Štítek `job:<id>` snímku označuje jeho úlohu a jeho cesty říkají, co úloha obsahuje: mezi nimi `/data/var/backup/engine/staging` u konfigurace, `/data/domains` u veškeré pošty. Obnovené soubory patří uživateli, který je obnovuje. Zpráva je jeden soubor v obvyklém internetovém formátu zpráv: po přejmenování na `.eml` ji otevře většina poštovních programů.

Čtení repozitář nemění. `restic check` si ho po dobu běhu vyhradí (výhradní zámek): server do něj mezitím nemůže zapisovat, takže ho spouštějte, když není na řadě žádná záloha. Nikdy nespouštějte `forget` ani `prune` z jiného stroje: server odstraňuje své staré snímky sám.

Copyright © 2026 Antonio Sanna

Testy doplňku spouštějí tyto příkazy přesně tak, jak jsou zde napsány, pro každý druh cíle na stroji jen s Debianem 12, restic, rclone a ssh.

Znovuvytvoření celého serveru

Když je server ztracen, nová instalace ze záloh získá zpět vše: účty a jejich hesla, domény, poštu, nastavení serveru, certifikáty, klíče DKIM, kalendáře a adresáře kontaktů, nastavení webmailu, učení spamového filtru, nastavení ostatních doplňků. Potřebujete úlohu, která zálohuje **konfiguraci**, a úlohu, která zálohuje **veškerou poštu** (nebo jednu úlohu s obojím), soubor s klíčem a cíl (viz předchozí sekce).

1. **Nainstalujte nový server** tak, jak byl nainstalován ten starý: obraz sestavený z tohoto repozitáře, stejná verze Poste.io (aktualizujte až potom), stejný název hostitele a porty a nová, **prázdná** datová složka. Spusťte ho a počkejte, dokud neodpoví stránka administrace (Poste.io připraví datovou složku); **nenastavujte** ho. Pak ho zastavte:

```
docker stop mailserver
```

1. **Uložte soubor s klíčem do složky** hostitele, například `/root/restore`, a otevřete shell jako root v kontejneru stejného obrazu s datovou složkou nového serveru a touto složkou. Neběží v něm žádná služba:

```
docker run --rm -it -v /srv/mailserver/data:/data -v /root/restore:/restore --entrypoint bash <image>
```

Použijte datovou složku (nebo svazek) a obraz své instalace a přidejte `--network <name>`, pokud je cíl dosažitelný jen ze sítě Docker serveru.

1. **Obnovte** poslední snímek konfigurace a poslední snímek pošty (stejný snímek, pokud obojí obsahuje jedna úloha) přímo do `/data`. V tomto shellu:

```
cd /restore
# the key and the destination: as in "Reading the backups without the server"
restic snapshots --group-by tags --latest 1
restic restore <configuration snapshot> --target /
restic restore <mail snapshot> --target /
```

Pod rootem vrátí restic každý soubor tam, kde byl, s jeho vlastníkem a oprávněními, přes soubory nové instalace.

1. **Databáze**: záloha obsahuje kopie vytvořené za provozu, konzistentní a zkontrolované. Umístěte je na místo (spolu se seznamem zpráv v zálohách, pokud ho záloha obsahuje) a odstraňte žurnálové soubory databází nové instalace:

```
S=/data/var/backup/engine/staging
cp "$S/users.db" /data/users.db
cp "$S/dav.db" /data/dav.db
cp "$S/roundcube.db" /data/roundcube/roundcube.db
if [ -f "$S/index.db" ]; then install -o mail -g mail -m 600 "$S/index.db" /data/var/backup/engine/index.db; fi
rm -f /data/users.db-wal /data/users.db-shm /data/dav.db-wal /data/dav.db-shm \
      /data/roundcube/roundcube.db-wal /data/roundcube/roundcube.db-shm "$S"/*.db
```

1. **Klíč**, aby server mohl staré zálohy číst a navázat na ně; pak shell opusťte:

```
install -o mail -g mail -m 600 restic.key /data/var/backup/settings/restic.key
exit
```

1. **Spusťte server** (`docker start mailserver`) a přihlaste se do administrace jako správce starého serveru. Pak v *Zálohy poštovního serveru* znovu přidejte cíle (jejich test tlačítkem **Otestovat** oznámí, že složka už obsahuje repozitář záloh; u SFTP znovu potvrďte klíč serveru a při přihlašování klíčem vložte na server nový veřejný klíč) a úlohy. Nové úlohy pokračují ve stejných repozitářích a to, co tam už je, se znovu neukládá. Snímky starých úloh zůstávají a už se neprořeďují: odstraňte je ručně, až nebudou potřeba (viz Úlohy zálohování).

Nevrací se: záznamy, pošta, která ještě čeká v odchozí frontě, antivirová databáze (stáhne se znovu), vlastní nastavení doplňku (krok 6) a to, co je uvedeno jako nikdy nezálohované v sekci Co se zálohuje. Pokud má nový server jinou adresu IP, aktualizujte záznamy DNS.

Testy doplňku spouštějí tento postup přesně tak, jak je zde napsán: záloha se obnoví na novou instalaci obrazu a poté se každý účet, zpráva, nastavení, databáze a vlastník porovnají se starým serverem.

Copyright © 2026 Antonio Sanna

Osvědčené postupy

- **3-2-1**: uchovávejte alespoň jednu kopii **mimo stroj** a ještě lépe jednu mimo budovu (NAS v jiné místnosti není mimo budovu; bucket S3 nebo server jinde ano). Úloha může zapisovat do několika cílů najednou.
- **Dvě úlohy** obvykle stačí: **konfigurace** často (každou hodinu: je malá) a **veškerá pošta** každý den, v noci. Příklad pravidel uchovávání: konfigurace 24 hodinových, 7 denních; pošta 7 denních, 4 týdenní, 12 měsíčních.
- **Stáhněte šifrovací klíč** hned a uchovávejte ho na **dvou místech** mimo server (například ve správci hesel a jako tištěnou kopii). Bez něj nelze žádnou zálohu přečíst. Uchovávejte spolu s ním údaje každého cíle (server, složka, uživatel, heslo): nastavení doplňku v zálohách nejsou a po ztrátě serveru jsou klíč a tyto údaje vše, co potřebujete (Čtení záloh bez serveru).
- **Vlastní účet** v každém cíli, s přístupem jen ke složce záloh. Ne účet správce NAS: kdo ovládne poštovní server, ovládl by celý NAS.
- **Chraňte zálohy před samotným serverem**: viz Pokud je poštovní server kompromitován.
- **Občas vyzkoušejte obnovu** (schránku do testovací schránky, v části *Snímky a obnova*, viz Obnova; zálohy čtené z jiného stroje se souborem s klíčem, viz Čtení záloh bez serveru): záloha, která nikdy nebyla obnovena, je naděje, ne záloha.
- **Podívejte se na přehled** po změnách (nový cíl, nová úloha, aktualizace) a čas od času: dokud nepřijdou upozornění, selhávající záloha nikomu nic neřekne.
- **Místo**: první záloha uloží vše; další ukládají jen to, co se změnilo (restic deduplikuje: data, která už jsou uložena, se nikdy neukládají znovu). Dimenzujte cíl pro poštu plus uchovávanou historii a sledujte místo zobrazené na přehledu. Pro představu o řádu, změřeno na skutečném serveru: první záloha 9,7 GB pošty trvala osm a půl minuty přes SMB na NAS ve stejné síti, další 18 sekund (1,8 MB nových dat) a repozitář zabral na NAS 3,5 GB (restic data komprimuje).

Záloha běžícího serveru

Záloha běží, zatímco server pracuje; nic se nezastavuje.

- **Databáze** (účty, kalendáře, webmail) se kopírují vlastním zálohovacím příkazem SQLite, který poskytuje konzistentní kopii používané databáze, a každá kopie se před uložením zkontroluje.
- **Pošta** je ve formátu Maildir, který je navržen tak, aby se dal kopírovat za provozu: každá zpráva je soubor, který se nikdy nemění. Jediné, co se může stát, je, že se zpráva přesune nebo smaže, zatímco záloha čte její složku: tato záloha ji vynechá (běh v poznámce uvede, kolik takových zpráv bylo) a zachytí ji ta příští. Indexové soubory Dovecotu Dovecot znovu sestaví, pokud jejich kopie není aktuální.
- **Snímek disku** (jeden okamžik pro všechno) může vytvořit jen hostitel, se souborovým systémem, který snímky podporuje (ZFS, LVM, btrfs), nebo zálohováním celého virtuálního stroje; kontejner to nedokáže bez oprávnění, o která tento doplněk nežádá. Pokud je hostitel už vytváří, jsou dobrým doplňkem: tento doplněk přidává kopie samotného poštovního serveru, mimo stroj, šifrované, s možností obnovit jednotlivé schránky.

Zabezpečení

- **Žádný root.** Administrace i modul běží jako `mail`. V Poste.io patří samotný `/data` uživateli `mail`, takže program, který by tam běžel jako root, by mohl být uživatelem `mail` sveden (symbolickým odkazem, přejmenovanou složkou, změněným řádkem databáze) ke čtení nebo zápisu souborů, ke kterým by přistupovat neměl. Spouštění záloh jako `mail` odstraňuje celou tuto třídu problémů: nic, co administrace zapíše, nemůže modul přimět číst nebo zapisovat víc, než už může `mail`. Root při startu jen vytvoří `/data/var/backup` (po ověření, že `/data/var` je skutečná složka vlastněná rootem) a předá ji uživateli `mail`.
- **Složky schránek** pocházejí z databáze Poste.io: složka se použije, jen pokud je přesně dvě úrovně pod `/data/domains`, bez části `.` / `..` a bez symbolického odkazu kdekoli v cestě.
- **Vše se kontroluje dvakrát.** Administrace odmítá neplatné hodnoty; modul soubor nastavení před použitím znovu ověří. Názvy serverů, složky, uživatelé a klíče se kontrolují podle přísných vzorů (složka nikdy nemůže vypadat jako volba, nic nesmí obsahovat `..`) a příkazy se nikdy nespouštějí přes shell.
- **Tajné údaje.** Hesla cílů a šifrovací klíč jsou na svazku, čitelné jen pro `mail` a root, nikdy se neposílají zpět do prohlížeče (pole hesla ponechané prázdné zachová uložené heslo) a rclone je dostává v proměnných prostředí (upravené pomocí „obscure“ přes stdin, nikdy na příkazovém řádku). Nastavení s hesly není součástí záloh: po znovuvytvoření serveru ze zálohy zadejte cíle znovu.
- **Zálohy** jsou zašifrovány klíčem dříve, než opustí server; cíl vždy vidí jen šifrovaná data. Obsahují veškerou poštu a hašovaná hesla účtů: chraňte cíle i klíč.
- **SFTP** komunikuje jen se serverem, jehož klíč SSH byl potvrzen.
- **Jen superadministrátoři** vidí tyto stránky.

Copyright © 2026 Antonio Sanna

Pokud je poštovní server kompromitován

Aby mohl vytvářet zálohy a odstraňovat staré, má server přihlašovací údaje k cílům s právem tam **zapisovat a mazat**. Kdo převezme kontrolu nad poštovním serverem (s heslem administrace nebo přes zranitelnost), může proto zálohy také smazat nebo zašifrovat, pokud je cíl nechrání. Chraňte je **na straně cíle**, kde poštovní server nemá co říct:

- **NAS**: plánované **snímky** sdílené složky se zálohami, pouze pro čtení (snímky Btrfs nebo ZFS na Synology, QNAP, TrueNAS…), uchovávané déle, než je doba, za kterou byste si problému všimli;
- **S3**: **verzování** bucketu, nebo lépe **Object Lock** s dobou uchování, aby bylo možné smazaný nebo přepsaný objekt obnovit;
- **server SFTP**: snímky jeho souborového systému, nebo kopie složky se zálohami, kterou ten server vytváří na místo, kam poštovní server nedosáhne.
- **server REST v režimu jen přidávání**: poštovní server tam může jen přidávat, ať je přinucen dělat cokoli; viz Server REST, jen přidávání.

Soubory a složky

```
/usr/local/bin/poste-backup               the engine: tick, run, status, restic (as mail;
                                          started as root it switches to mail)
/usr/local/lib/poste-backup/lib/          its PHP library (shared with the admin's validation)
/etc/cron.d/poste-backup                  runs "poste-backup tick" every minute, as mail
/etc/cont-init.d/90-backup.sh             at every start: the folder (root), then folders
                                          inside and the key (once) as mail

/data/var/backup/                         mail:mail 0750
  settings/                               (the admin)
    config.json                           destinations and jobs (0600)
    restic.key                            the encryption key (0600)
    key-downloaded, tests.json            when the key was downloaded, last tests
    queue/                                requests: run now, restore a mailbox, remove
                                          the backups of a job
    ssh/<id>, <id>.pub                    key pairs of SFTP destinations
  engine/                                 (the engine)
    state.json                            last run of each job, last restores and removals
    history.jsonl                         one line per run and destination (with the space
                                          used there and on /data), and per verification
    logs/                                 log of each run (kept for the days set in the admin)
    index.db                              the list of the messages of the backups (0600)
    staging/                              consistent copies of the databases, during a run
    restore/                              a mailbox being restored, before it is imported
/var/cache/poste-backup                   restic's cache, and listing.db: the files of the
                                          snapshots last shown in the admin (rebuilt when needed)
/run/poste-backup.lock                    one run at a time
```

Řešení problémů

```
docker exec mailserver poste-backup status          # last results
docker exec mailserver ls -t /data/var/backup/engine/logs | head
docker exec mailserver poste-backup run <job id>    # run a job now and wait for it
```

Přehled upozorní, když modul neběžel déle než deset minut (a přitom neběží žádná záloha): cron pak v kontejneru neběží. Také ukáže, proč spouštěcí skript nemohl připravit složku záloh, pokud se to stalo (`/run/poste-backup.start-error`): například `/data/var` nahrazený něčím, co není složka vlastněná rootem.

Omezení

- **Jeden repozitář na server**: nesměrujte dva poštovní servery do stejné složky.
- Zatím žádná upozornění: přijdou s dalšími etapami.
- Statistiky jsou po dnech a ve zobrazení za 1 den po hodinách (skutečné hodiny: v den změny času se popisek hodiny zobrazí dvakrát nebo se přeskočí).
- Schránka se obnovuje do nové složky: celá, některé její složky, zprávy odpovídající filtru (data, odesílatel, předmět), nebo až 1 000 jednotlivých zpráv najednou vybraných ze seznamu. Celý server se obnovuje z příkazového řádku, na novou instalaci.
- restic a rclone jsou ve verzích z vydání Debianu, na kterém je obraz postaven (restic 0.14, rclone 1.60 v Poste.io 2.5.24).
- První záloha velkého serveru chvíli trvá (vše je nové); další ukládají jen to, co se změnilo.
- S3 neuvádí volné místo.

Copyright © 2026 Antonio Sanna

Odebrání

Sestavte bez této sekce. Ve svazku zůstane `/data/var/backup` (nastavení, klíč, stav). Zálohy v cílech zůstanou nedotčeny: uchovejte klíč, pokud je můžete potřebovat.

Copyright © 2026 Antonio Sanna

Opravy upstreamu

> ***Převzato upstreamem.*** *Poste.io **2.5.20** obsahuje všechny tři opravy (učicí skripty volají `rspamc --rcpt`, mažou svůj dočasný soubor a už nejsou zapisovatelné pro všechny), takže doplněk `upstream-fixes` z Dockerfilů zmizel. Na základním image starším než 2.5.20 sestavujte z vydání 1.2.0 (https://github.com/atsanna/poste.io-addons/releases/tag/v1.2.0). End-to-end testy chování učení dál ověřují při každém sestavení. Zbytek této stránky popisuje tři problémy tak, jak byly.*

Tři problémy image Poste.io kolem učení spamu: přesunutí zprávy do **Junk** (nebo z ní) má rspamd učit, a do verze 2.5.19 to nedělalo (opravy 1 a 2); a každá naučená zpráva zůstala ležet v `/tmp` (oprava 3). První dva byly nahlášeny Poste.io; vydání do 1.2.0 obsahovala obejití, označené `UPSTREAM-FIX`, odstraněné teď, když je základní image opravuje.

1. Učicí skripty jsou zapisovatelné pro všechny

V základním image mají `/etc/dovecot/sieve` a skripty roury `spam` / `ham` režim `0777`. Dovecot odmítá spustit program zapisovatelný pro všechny (`executable ... is world-writable` v `/data/log/mail.log`), takže k učení nikdy nedojde.

Oprava: `chmod 0755` na složku a oba skripty, `0644` na soubory `.sieve`.

2. Učení z Junk jde do úložiště, které nikdo nečte

Poste.io konfiguruje Bayesův klasifikátor rspamd **pro každou doménu** (`per_user` vrací doménu příjemce), takže kontrola pošty čte klíč `RS<domain>`. Skripty pro Junk však volají `rspamc learn_spam` / `learn_ham` **bez příjemce**: rspamd pak učí do `RS<address>`, který žádná kontrola nikdy nečte.

Oprava: imapsieve předá skriptům uživatele IMAP (`pipe :copy "spam" ["${IMAPUSER}"]`) a skripty volají `rspamc --rcpt "$1" learn_spam`, takže učení skončí v úložišti domény. Blok končí kontrolami `grep -q`: pokud Poste.io skripty změní, sestavení selže, místo aby dodalo opravu, která už neplatí.

3. Každá naučená zpráva zůstane v /tmp

Skripty `spam` / `ham` zkopírují zprávu do dočasného souboru (`input_tmpfile=$(mktemp)`), aby ji mohly předat `rspamc` dvakrát, a nikdy ho neodstraní. Každá zpráva přesunutá do Junk nebo z ní tak zanechá svou kopii v `/tmp` kontejneru, čitelnou pro uživatele `mail`, dokud se kontejner znovu nevytvoří.

Oprava: hned za řádek s `mktemp` dostanou skripty

```
trap 'rm -f "$input_tmpfile"' EXIT
trap 'exit 1' HUP INT TERM
```

takže se soubor odstraní, když skript skončí, i když je zastaven. Stejně jako u opravy 2 kontrola `grep` nechá sestavení selhat, pokud se skripty změní.

Jsou ještě potřeba?

Ne, od Poste.io 2.5.20 (ověřeno na `poste.io/mailserver:2.5.20` a `analogic/poste.io:2.5.20`, skripty z 2026-10-21). Na starším základním image se podívejte do základního image:

```
docker run --rm --entrypoint sh poste.io/mailserver:<VERSION> -c \
  'ls -l /etc/dovecot/sieve; grep -n "pipe :copy" /etc/dovecot/sieve/*.sieve; grep -n "rspamc.*learn_\|tmpfile" /
etc/dovecot/sieve/spam /etc/dovecot/sieve/ham'
```

(`analogic/poste.io:<VERSION>` pro edici open source.)

- skripty `-rwxr-xr-x` → odstraňte opravu 1;
- `pipe` s argumentem a `rspamc` s `--rcpt` (nebo rovnocenné řešení) → odstraňte opravu 2;
- skripty svůj dočasný soubor odstraňují (`rm`, `trap` nebo žádný dočasný soubor) → odstraňte opravu 3.

Pokud místo toho sestavení selže na kontrolách `grep`, skripty se změnily: než blok upravíte, přečtěte si je.

Copyright © 2026 Antonio Sanna

Ověření, že učení funguje

Přesuňte spamovou zprávu do Junk a pak v kontejneru:

```
redis-cli -s /var/run/redis/redis.sock hget RS<your-domain> learns_spam
```

Počítadlo **vaší domény** musí vzrůst. (`rspamc stat` sčítá všechny klíče a může být zavádějící.) Bayes začne klasifikovat po 200 naučených zprávách každého druhu.

Copyright © 2026 Antonio Sanna

Testy



(https://github.com/atsanna/poste.io-addons/actions/workflows/tests.yml)

Každý push a pull request sestaví image **open source** a spustí end-to-end testy proti skutečnému kontejneru Poste.io. Odznak výše (i ten v README) ukazuje výsledek posledního běhu.

Edice **Pro** se zde netestuje: její základní image pochází z registru Poste.io a vyžaduje licenci. Doplňky jsou v obou edicích tytéž soubory; image Pro se před každým vydáním testuje ručně.

Co se spouští

`.github/workflows/tests.yml` *(https://github.com/atsanna/poste.io-addons/blob/main/.github/workflows/tests.yml)* spouští paralelně sedm úloh. Každá sestaví image z `Dockerfile` s verzí Poste.io, která je v něm pevně nastavena (`ARG POSTE_IO_VERSION`), a pak na něm spustí `tests/run.sh` *(https://github.com/atsanna/poste.io-addons/blob/main/tests/run.sh)*:

Úloha	Image
`all`	Dockerfile tak, jak je, všechny doplňky
`external-mailboxes`, `dark-theme`, `activesync-dav`, `activesync-logs`, `backup`	jen jeden doplněk, vytvořený pomocí `select-addons.sh`

Úlohy s jediným doplňkem dokazují, že se každý doplněk sestaví a funguje bez ostatních; testy doplňků, které v image nejsou, se hlásí jako `SKIP`.

Jak probíhá běh

`tests/run.sh <image>`:

1. spustí z image dočasný kontejner: bez publikovaných portů, s vlastním svazkem, s názvem hostitele `mail.example.test`; počká, až odpoví administrace, ActiveSync a IMAP. Testy jsou napsané v Pythonu (jen standardní knihovna): základní image Poste.io nemá `python3` (Externí schránky ho přinášejí spolu s getmail6), takže u image bez tohoto doplňku se doinstaluje do dočasného kontejneru;
2. vytvoří doménu `example.test` a tři schránky (`alice`, superadministrátor, `bob` a `carol`) vlastními konzolovými příkazy Poste.io; hesla jsou náhodná, generují se uvnitř kontejneru a nikdy se nevypisují;
3. exportuje tabulky WBXML ze Z-Push obsaženého v image, aby testovací klient mluvil přesně tím kódováním ActiveSync, které používá Z-Push;
4. spustí uvnitř kontejneru `tests/test_addons.py` a `tests/test_activesync.py`; každá kontrola vypíše `PASS` nebo `FAIL` spolu s tím, co zjistila;
5. s doplňkem zálohování: spustí ve stejné privátní síti další dva dočasné kontejnery jako cíle záloh, server Samba + SFTP (`tests/backup-dest` *(https://github.com/atsanna/poste.io-addons/tree/main/tests/backup-dest)*) a server S3 (vlastní image rclone s pevnou verzí); `tmpfs` v `/backup/local` hraje roli svazku připojeného hostitelem. Spustí `tests/test_backup.py` a potom `tests/test_restore.sh` *(https://github.com/atsanna/poste.io-addons/blob/main/tests/test_restore.sh)*, který čte zálohy z jiného kontejneru, jen s restic, rclone a ssh (`tests/backup-reader` *(https://github.com/atsanna/poste.io-addons/tree/main/tests/backup-reader)*), a obnoví z nich celý server v novém kontejneru ze stejného image;
6. odstraní kontejnery, svazky a síť. Návratový kód je počet neúspěšných kontrol.

Administrace se ovládá jako v prohlížeči (`tests/panel.py` *(https://github.com/atsanna/poste.io-addons/blob/main/tests/panel.py)*): stránky se načítají, formuláře se čtou ze stránky (včetně jejich tokenu CSRF), vyplňují a odesílají. Telefon se simuluje skutečnými požadavky ActiveSync 14.0 (`tests/eas.py` *(https://github.com/atsanna/poste.io-addons/blob/main/tests/eas.py)*): Provision, FolderSync, Sync a Ping, tak jak je posílá zařízení. Nic se nenastavuje za zády Poste.io.

Co se kontroluje

Externí schránky (`tests/test_addons.py`) - položky menu, tři stránky, nainstalovaný getmail; - plán je v crontabu uživatele `mail` (pět intervalů), nic pro roota; složky getmailu `0750` s vlastníkem `mail`; - formulář odeslaný bez tokenu CSRF je odmítnut (HTTP 403, nic se neuloží); - schránka vytvořená v administraci se zapíše jako soubor getmailu `0600` s vlastníkem `mail`, se seznamem složek IMAP jako platnou n-ticí Pythonu; - heslo se nikdy neposílá zpět do prohlížeče; - end-to-end: zprávu vloženou do schránky uživatele bob stáhne `cron.sh` (spuštěný jako `mail`, stejně jako z cronu) přes IMAP a doručí ji uživatelce alice; stránka logů ji ukazuje; - rspamd u stažené pošty: soubor getmailu má sekci `[filter1]`; stažená zpráva dorazí s `X-Rspamd-Score` (její report se vypíše); stažená

Copyright © 2026 Antonio Sanna

testovací zpráva GTUBE dorazí do Junk s předmětem `*****SPAM*****`; - „Stahovat vše“ jen s „Mazat zprávy“: u nové schránky vypnuto; s vypnutým „Mazat zprávy“ odmítnuto (HTTP 400, nic se neuloží), se zapnutým uloženo; komentář nad `read_all` ve vygenerovaném souboru říká, co volba dělá; - rotace logů: výchozí pravidlo (týdně, 1 MB, 4 kopie) a jeho noční položka cronu; nastavení na stránce logů; uložení bez tokenu CSRF je odmítnuto; uložené nastavení (`0640`, vlastník `mail`) použije skript spouštěný jako root; hodnota, kterou formulář nenabízí, je odmítnuta; soubor nastavení změněný mimo administraci dá výchozí hodnoty a nic nespustí; vypnutá rotace nenechá v pravidle žádný log getmailu; vynucená rotace ponechá `getmail.log.1`, znovu vytvoří `getmail.log` a getmail do něj zapisuje; stránka ukazuje staré kopie; - chyby getmailu: po úspěšném běhu je `errors.log` prázdný (`0640`, vlastník `mail`) a zobrazení getmailu nemá žádný chybový řádek; v administraci se přidá schránka se špatným heslem a spustí se: chyba přihlášení je v `errors.log` s příslušnou schránkou, nahoře v zobrazení **Chyby getmailu** jako chyba a shrnutá v červeném řádku nahoře v zobrazení getmailu; po smazání schránky v administraci je **Vymazat** bez tokenu CSRF odmítnuto a s ním soubor vyprázdní, a červený řádek zmizí.

Logy ActiveSync (`tests/test_addons.py`) - položka menu hned za *User logs*; složka nastavení vytvořená startovacím skriptem na novém svazku (`0750`, vlastník `mail`); výchozí pravidlo (oba logy týdně, 4 kopie, rotované jako `www-data`, bez `copytruncate`) a jeho noční položka cronu; - skutečná chyba Z-Push: požadavek ActiveSync se špatným heslem je nahoře na stránce jako chyba, s příslušným účtem; logy Z-Push zůstávají `www-data 0644`; - se známými řádky ve formátu Z-Push: nejnovější první, zpráva zapsaná na dva řádky zobrazená jako jedna, text logu escapovaný (žádné HTML z logu), filtry úrovně, menu a filtr účtů (účet, který v menu není, se ignoruje), vyhledávání bez ohledu na velikost písmen, hledaný text, který shell nespustí, log chyb; - žádný formulář pro vymazání; nastavení rotace: bez tokenu CSRF odmítnuto, uloženo `0640` s vlastníkem `mail`, použito skriptem spouštěným jako root, hodnota, kterou formulář nenabízí, odmítnuta, soubor nastavení změněný mimo administraci dá výchozí hodnoty a nic nespustí, vypnutá rotace nenechá v pravidle žádný log Z-Push; vynucená rotace ponechá `z-push.log.1`, znovu vytvoří `z-push.log` `0644` s vlastníkem `www-data`, Z-Push do něj při dalším požadavku zapisuje a stránka ukazuje staré kopie; - `lint:yaml` a `lint:twig` projdou.

Tmavý motiv (`tests/test_addons.py`) - oba řádky `include` jsou v `base.html.twig`, každý jednou; - šablona stylů se servíruje jako `text/css` a obsahuje tmavá pravidla; - přihlašovací stránka načítá šablonu stylů a skript motivu v `<head>`; - přepínač je v horní liště; překlady (en, cs) jsou přítomny; - `lint:yaml` a `lint:twig` projdou.

Opravy upstreamu (`tests/test_addons.py`) - složka sieve a učicí skripty nejsou zapisovatelné pro všechny; - zpráva se přes IMAP přesune do Junk: spustí se učicí skript, `rspamc` dostane příjemce (`--rcpt`), rspamd se ji naučí jako spam a tokeny skončí v bayesovském úložišti **domény** (v tom, které se používá při kontrole), ne v úložišti jednotlivé adresy.

Kalendáře a kontakty přes ActiveSync (`tests/test_activesync.py`) - nastavení z administrace: DAV zapnutý v nastavení, kalendář a adresář pro uživatelku alice, kalendář uživatele bob sdílený s alice **jen pro čtení**, nic pro carol; - složky v telefonu: pošta beze změny; kalendář uživatelky alice je výchozí, sdílený kalendář uživatele bob je další; kontakty jen pro ty, kdo mají adresář; carol dostane jen poštu; - server → telefon: události a kontakt zapsané přes CalDAV/CardDAV dorazí; opakování „2. a 3. den v měsíci“ dorazí zjednodušené (známé omezení Z-Push, kontrolované proto, aby se případná změna poznala); - telefon → server: událost a kontakt vytvořené v telefonu se uloží; zápis do kalendáře jen pro čtení je odmítnut (nová událost 6, změna 7, smazání 8) a kalendář na serveru zůstane beze změny; - adresář smazaný v administraci, zatímco telefon ještě má jeho složku: kontakt do ní odeslaný dostane chybu (ne HTTP 500) a složka se odstraní; - DAV vypnutý v nastavení: pošta dál funguje (nová zpráva se doručí, jen kalendář dostane „hierarchy changed“), Ping odpovídá, FolderSync odstraní jen kalendáře a kontakty, nový telefon dostane jen poštu; po opětovném zapnutí: kalendáře, kontakty a jejich obsah se vrátí.

Zálohování (`tests/test_backup.py`, `tests/backup_unit.php`) - instalace: položka menu, stránky, složky a šifrovací klíč (`mail`, `0750` / `0700` / `0600`; 48 náhodných bajtů), cron spouštějící modul jako `mail` každou minutu; jednotkové kontroly plánů (i přes přechody na letní a zimní čas), validace, zpráv a jejich překladů; - cíle: bez tokenu CSRF odmítnuto; neplatné hodnoty odmítnuty se zprávou (složka na `/data`, `..`, název serveru se `;`, složka, která vypadá jako volba); místní svazek (skutečné připojení) testem projde, složka, která není přípojným bodem, ne; SMB se skutečnou velikostí sdílené složky (`df` na serveru Samba: rclone 1.60 hlásí polovinu); SMB se špatným heslem uložené, ale v testu neúspěšné; S3; SFTP: zobrazí se otisky klíčů serveru, ničemu se nedůvěřuje až do potvrzení, pak přesně klíčům serveru; SFTP s párem klíčů vytvořeným administrací; hesla se nikdy neposílají zpět do prohlížeče; - tlačítko **Otestovat** ve formuláři: otestují se zadané hodnoty a nic se neuloží ani nezapamatuje; špatné heslo, chybějící token CSRF, neplatný server, uložené heslo znovu použité jen pro tentýž server; klíče SFTP potvrzené ve formuláři; - úlohy a běhy: úloha do pěti cílů spuštěná z administrace, na každém cíli snímek se štítkem úlohy, jeden řádek historie na cíl; ve snímku kopie databází, redis, klíče DKIM a pošta, a **ne** klíč ani nastavení s hesly; smazaná zpráva obnovená identicky; obnova schránky ze stránky Zálohování a obnova provedená tak, jak je tam napsána; databáze účtů obnovená ze SFTP neporušená; pravidla uchovávání pro každou úlohu; jen vybraná schránka; jeden selhávající cíl nezastaví ostatní, neprázdná složka se nikdy nepoužije; soubor, který záloha nemůže přečíst, vyvolá varování s uvedením počtu; - zabezpečení: nastavení změněná mimo administraci (vložený příkaz, neplatné id) se nahlásí a nic z nich se nespustí; soubory modulu patří uživateli `mail`; domovská složka schránky změněná v `users.db` (`..`, symbolický odkaz) nikdy nevede mimo `/data/domains`; startovací skript odmítne `/data/var`, který nepatří rootovi; klíč, který je symbolickým odkazem, je odmítnut; - místo na každém cíli, stránka logů a log běhu (názvy, které nejsou logy, ani odkazy se nenajdou), průběh v reálném čase, zatímco úloha běží, stránky v češtině, časové pásmo kontejneru a pásmo zvolené v administraci, počet dní uchovávání logů, vždy jen jeden běh najednou; - snímky a obnova: snímky cíle s jejich úlohou a obsahem, chyba cíle, který nelze přečíst; stránka obnovy: krok 1 vypíše schránky snímku, které obsahují zprávy, s jejich počtem a časem snímku, označí schránku, která už na serveru není, a prázdné složky a přesměrování jen spočítá; krok 2 zaškrtne složky se zprávami a zvolí tutéž schránku, nebo žádnou u schránky, která už na serveru není, a upozorní, když adresa ve snímku není; schránka obnovená z administrace do jiné schránky a dvakrát do téže: nová složka *Restored* (pak *(2)*) se stejným počtem zpráv, přihlášená k odběru, zpráva identická s originálem, nic z vlastní pošty schránky

Copyright © 2026 Antonio Sanna

nezměněno, pracovní složka modulu odstraněna; jen složka Sent do nové složky uvnitř existující; filtr podle předmětu, podle data přijetí (zpráva z ledna 2020 odeslaná přes IMAP se svým datem), rozsah dat bez zprávy a text začínající „-“: nic neobnoveno, žádná chyba; odmítnuto s příslušnou zprávou: nezvolená schránka, cíl obnovy, který není schránkou, žádná složka, složka, která ve snímku není, název složky s tečkou, nadřazená složka, která neexistuje, data v obráceném pořadí, chybějící token CSRF, id snímku, které žádnému snímku nepatří; požadavky změněné mimo administraci (cesta jako adresa nebo jako složka) odmítnuté modulem, neexistující snímek vede k chybě; symbolický odkaz v obnovené složce odstraněn, nikdy nenásledován; seznam zpráv: uchovávaný jen uživatelem `mail`, v záloze konfigurace jako kopie, předmět a odesílatel kódované jako hlavičky se dekódují, každá zpráva schránky uvedena jednou, zpráva přesunutá do jiné složky je tatáž položka, smazaná označená a ponechaná, log běhu říká, jak byl seznam aktualizován; zprávy snímku v administraci: hledání (na velikosti písmen a diakritice nezáleží) najde zprávu s její složkou, odesílatelem, předmětem, datem a velikostí, nic nenalezeno dává 0, jen jedna složka, nejnovější první, zpráva, kterou seznam nezná, zobrazená bez podrobností, složka, která je cestou, nebo stránka pod 0 odmítnuty; dvě zprávy vybrané z mnoha obnovené do jiné schránky, a jen ty; odmítnuto s příslušnou zprávou: žádná vybraná, více než 1 000, cesta mimo Maildir; zpráva, která ve snímku není, odmítnutá modulem; čištění: zpráva zmizelá před nejstarší uchovávanou zálohou se odstraní, zpráva zmizelá od té doby zůstane; výběr v češtině; stránka příkazů k obnově serveru (hodnoty v uvozovkách, žádné heslo, u SFTP s klíčem vlastní klíč a potvrzené klíče serveru), nabízená jen u snímků s konfigurací; zálohy smazané úlohy odstraněné po potvrzení, zálohy ostatních úloh nedotčené; - ověřování, statistiky, upozornění na místo: nový cíl se ověřuje každou neděli ve 04:00, čtvrtina dat; *Ověřit nyní* přečte první čtvrtinu, pak druhou; nastavení uložené z formuláře; všechna data, je-li tak nastaveno; **jeden bajt datového souboru změněný na cíli (přes rclone) ověření najde**, zobrazí se mezi problémy na přehledu, na stránce cíle a v logech; ověření, na které podle plánu přišla řada, spustí pravidelný tik modulu; řádek historie zálohy obsahuje obsazené místo na cíli a na `/data`; statistiky za 30 dní, 90 dní a rok (jiná hodnota znamená 30), s místem, přidanými daty a dobou trvání úlohy; upozornění na místo odmítnutá bez tokenu CSRF a když žlutá není pod červenou, pak uložená a převzatá pruhy; v češtině; - REST server v režimu append-only (oficiální `rest-server` od restic, HTTPS s certifikátem vytvořeným `run.sh`): Test ve formuláři nejprve ukáže certifikát serveru (jeho otisk porovnaný s `openssl`), pak po potvrzení oznámí, že je dostupný a prázdný; špatné heslo, jiný certifikát a adresa, která v certifikátu není, jsou odmítnuty s uvedením důvodu; tři běhy s pravidlem „Uchovávat hodinových: 1“ tam ponechají všechny tři snímky (z tohoto serveru se nic neodstraňuje) a log říká proč; v logu žádné heslo; stránka snímků ukazuje poznámku a žádné odstranění tam, a modul odmítne odstranění zapsané mimo administraci; ověření funguje; příkazy k obnově serveru obsahují certifikát a `--cacert`, žádné heslo; jiný certifikát uložený pro server způsobí, že záloha selže dřív, než se cokoli odešle; v češtině. Jednotkové kontroly: nastavení ověřování a jejich limity, kdy je ověření na řadě, požadavek, grafy vypočtené z řádků historie (jeden běh započtený jednou, starší řádky a ověření vynechány); cíl REST (přijatý zatím bez certifikátu, odmítnuté hodnoty, adresa s kódovaným heslem v prostředí a nikdy na příkazovém řádku, `--cacert`, IPv6 v hranatých závorkách).

Obnova (`tests/test_restore.sh`, `tests/read_backups.sh`, `tests/rebuild_server.sh`, `tests/test_restore.py`): dva postupy ze stránky Zálohování a obnova pro ztracený server, provedené tak, jak jsou tam napsány. - čtení záloh bez serveru, v kontejneru Debian 12 jen s restic, rclone a ssh a se souborem klíče staženým z administrace: z SMB, SFTP (s heslem i s vlastním klíčem), S3 a ze složky se vypíšou snímky a zpráva se obnoví identicky; `restic check` nenajde žádné chyby; jiný klíč, celý stažený soubor (místo jeho posledního řádku) a server SFTP s jiným klíčem jsou odmítnuty; - obnova celého serveru: nová instalace image jednou spuštěná a zastavená, záloha obnovená jako root bez běžící služby, server znovu spuštěn; poté účty a jejich (hashovaná) hesla, domény, všechny zprávy, klíče DKIM, certifikáty, kalendáře a adresáře, databáze webmailu, redis a vlastníci a oprávnění odpovídají starému serveru, seznam zpráv záloh je zpět; uživatel se přihlásí přes IMAP starým heslem a najde starou zprávu; přihlásí se starý administrátor; starý klíč je zpět a nový server čte staré zálohy z cíle zadaného znovu.

Vlastní spuštění testů

S Dockerem, z repozitáře:

```
docker build -f Dockerfile --build-arg POSTE_IO_VERSION=2.5.24 -t poste-addons-test .
sh tests/run.sh poste-addons-test
```

Se všemi doplňky trvá běh po sestavení asi patnáct minut (měřeno na stolním počítači), většinu z toho zaberou testy zálohování a obnovy; bez doplňku zálohování asi tři. Stejný skript funguje i s image Pro sestaveným z `Dockerfile-pro`.

Copyright © 2026 Antonio Sanna