MailCow: co to jest i z czego się składa

Poziom: średni **Dotyczy: MailCow Dockerized** **Czas: ~25 min**

MailCow to nie jeden program, tylko zestaw kilkunastu kontenerów współpracujących ze sobą, więc zanim zaczniemy cokolwiek zmieniać, warto wiedzieć, który kontener odpowiada za co.

Skróty użyte na tej stronie (10)

Skrót	Co znaczy
ACL	Access Control List: lista, kto ma jakie prawa do pliku lub zasobu
DKIM	DomainKeys Identified Mail: cyfrowy podpis wiadomości e-mail
IMAP	Internet Message Access Protocol: protokół, którym klient pocztowy pobiera wiadomości z serwera, zostawiając je tam zamiast kasować
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
MDA	Mail Delivery Agent: program wkładający pocztę do skrzynki
MTA	Mail Transfer Agent: serwer przekazujący pocztę dalej, np. Postfix
SMTP	Simple Mail Transfer Protocol: protokół wysyłania poczty
SOGo	webmail z kalendarzem i kontaktami, który w MailCow jest dostępny w przeglądarce obok panelu administracyjnego
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS

Fiszka

TYLKO ODCZYT

Sprawdzenie działających kontenerów: docker compose ps

```
cd /docker/apps/mailcow && sudo docker compose ps
```

Wypisujemy wszystkie kontenery należące do danej instalacji MailCow wraz z ich statusem. To pierwszy krok każdej diagnostyki, zanim zajrzymy w logi konkretnego kontenera.

Przykład:

```
NAME                                IMAGE                     STATUS
mailcowdockerized-postfix-mailcow-1 ghcr.io/mailcow/postfix   Up 4 days (healthy)
mailcowdockerized-dovecot-mailcow-1 ghcr.io/mailcow/dovecot   Up 4 days (healthy)
mailcowdockerized-mysql-mailcow-1   mariadb:10.11             Up 4 days (healthy)
mailcowdockerized-sogo-mailcow-1    ghcr.io/mailcow/sogo      Up 4 days (healthy)
mailcowdockerized-rspamd-mailcow-1  ghcr.io/mailcow/rspamd    Up 4 days (healthy)
mailcowdockerized-nginx-mailcow-1   ghcr.io/mailcow/nginx     Restarting (1) 12
seconds ago
```

cd = change directory: przejdź do katalogu

OSTROŻNIE · NIEODWRACALNE

Wykonanie pełnej kopii zapasowej: backup_and_restore.sh

```
cd /docker/apps/mailcow
sudo env MAILCOW_BACKUP_LOCATION=/docker/kopie/mailcow THREADS=2 ./helper-
scripts/backup_and_restore.sh backup all --delete-days 3
```

Katalog na kopie podajemy zmienną `MAILCOW_BACKUP_LOCATION`, a nie argumentem: bez zmiennej skrypt pyta o niego interaktywnie. Argument `all` obejmuje komponenty `vmail`, `crypt`, `redis`, `rspamd`, `postfix` i `mysql`. Pełny opis kopii, rotacji i wysyłki na `bkp01` jest na stronie MailCow: kopia zapasowa i aktualizacja.

Przykład (początek wyniku):

```
Using 2 Thread(s) for this run.
Using /docker/kopie/mailcow as backup/restore location.


Found project name mailcowdockerized
```

cd = change directory: przejdź do katalogu · **env** = environment: uruchamia program z dodatkowymi zmiennymi środowiskowymi, na przykład sudo env ZMIENNA=wartość polecenie

□□ ZMIENIA SYSTEM

Odtworzenie z kopii zapasowej: backup_and_restore.sh restore

```
cd /docker/apps/mailcow
sudo env MAILCOW_BACKUP_LOCATION=/docker/kopie/mailcow ./helper-
scripts/backup_and_restore.sh restore
```

Skrypt uruchomiony w trybie odtwarzania pyta interaktywnie, którą z zapisanych kopii i który komponent odtworzyć. Kontenery zatrzymuje sam: odtwarzanie bazy zatrzymuje cały MailCow, a odtwarzanie skrzynek zatrzymuje Dovecota, więc ręcznego zatrzymywania nie robimy. Mimo to nie uruchamiamy odtwarzania na działającej instalacji produkcyjnej bez decyzji i bez świeżej kopii tego, co zostanie zastąpione.

□□ **cd** = change directory: przejdź do katalogu · **env** = environment: uruchamia program z dodatkowymi zmiennymi środowiskowymi, na przykład sudo env ZMIENNA=wartość polecenie

□□ ZMIENIA SYSTEM

Aktualizacja instalacji: update.sh

```
cd /docker/apps/mailcow && sudo ./update.sh
```

Przykład:

```
Checking for update...
Pulling images...
Recreating containers...
Update complete.
```

□□ **cd** = change directory: przejdź do katalogu

□□ TYLKO ODCZYT

Zalogowanie się do kontenera bazy danych: docker exec

```
sudo docker exec -it mailcowdockerized-mysql-mailcow-1 mysql -p
```

Hasło do konta roota MySQL znajduje się w pliku `mailcow.conf` w katalogu instalacyjnym, pod zmienną `DBROOT`.

□□ **docker** = narzędzie do kontenerów

□□ TYLKO ODCZYT

Przełączenie się na bazę danych mailcow: USE

```
USE mailcow;
```

□□ TYLKO ODCZYT

Sprawdzenie istniejących reguł sender_acl: SELECT

```
SELECT * FROM sender_acl;
```

Przykład:

```
+----+--------------------------+-----------------------+----------+
| id | logged_in_as             | send_as               | external |
+----+--------------------------+-----------------------+----------+
| 10 | sekretariat@zielona.example   | zarzad@zielona.example     |        0 |
| 11 | biuro@zielona.example         | biuro@zielona.example      |        0 |
+----+--------------------------+-----------------------+----------+
2 rows in set (0.00 sec)
```

□□ ZMIENIA SYSTEM

Dodanie nowej reguły Send As dla skrzynki lokalnej: INSERT

```
INSERT INTO sender_acl (logged_in_as, send_as, external) VALUES
('NazwaSkrzynki@zielona.example', 'AdresWysylkowy@zielona.example', 0);
```

To samo robimy zwykle w panelu, a nie w SQL: E-Mail → Configuration → Mailboxes → (skrzynka) → Edit → Allow to send as, jak na stronie MailCow: codzienna praca z domenami, skrzynkami i spamem. Wpis SQL pokazuje, co panel zapisuje. Kolumna `external` ustawiona na `0` oznacza, że oba adresy należą do domen obsługiwanych przez tę instalację MailCow. Adresy z domen spoza instalacji wpisujemy w tym samym oknie w polu „External sender addresses” (panel zapisuje je z `external` równym `1`), a nie ręcznie w bazie.

□□ **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych

TYLKO ODCZYT

Sprawdzenie logów konkretnego kontenera: docker logs --tail

```
sudo docker logs --tail 100 mailcowdockerized-postfix-mailcow-1
```

Ograniczamy wynik do ostatnich 100 linii: logi kontenerów działających od dłuższego czasu potrafią być bardzo długie, a interesuje nas zwykle to, co wydarzyło się tuż przed zauważonym problemem.

docker = narzędzie do kontenerów

TYLKO ODCZYT

Śledzenie logów na żywo: docker logs -f

```
sudo docker logs -f mailcowdockerized-dovecot-mailcow-1
```

Widzimy typowy wpis przy problemie z uwierzytelnieniem.

Przykład:

```
dovecot: imap-login: Disconnected (auth failed, 3 attempts):
user=<uzytkownik@zielona.example>, method=PLAIN, rip=10.20.10.42
```

docker = narzędzie do kontenerów

TYLKO ODCZYT

Restart pojedynczego kontenera: docker compose restart

```
cd /docker/apps/mailcow && sudo docker compose restart nginx-mailcow
```

Restart pojedynczego kontenera nie rusza pozostałych. Używamy tego zamiast pełnego `docker compose down && up`, gdy problem dotyczy wyraźnie jednego komponentu.

cd = change directory: przejdź do katalogu

Z życia wzięte

Z ŻYCIA WZIĘTE

W piątek po południu panel administracyjny MailCow na `docker01` przestaje odpowiadać, a Ola już szykuje się do zgłoszenia dużej awarii poczty. Tomek sprawdza `docker compose ps` i widzi, że tylko kontener `nginx-mailcow` się restartuje w kółko, podczas gdy `postfix-mailcow` i `dovecot-mailcow` stoją zielone. Restart samego `nginx-mailcow` przywraca panel, a poczta przez cały ten czas była dostarczana bez przerwy.

Czym jest MailCow

Pojęcie: MailCow. Kompletny serwer pocztowy złożony z kilkunastu współpracujących ze sobą kontenerów Docker, uruchamianych razem przez `docker-compose`. W jednym miejscu łączy MTA (Postfix), MDA i serwer IMAP (Dovecot), bazę danych, webmail (SOGo), panel administracyjny, filtr antyspamowy i ochronę przed nadużyciami (fail2ban). Zamiast konfigurować i łączyć te elementy ręcznie, jak przy budowaniu serwera pocztowego od zera na bazie samego Postfixa, MailCow dostarcza je jako gotowy, wspólnie zarządzany zestaw.

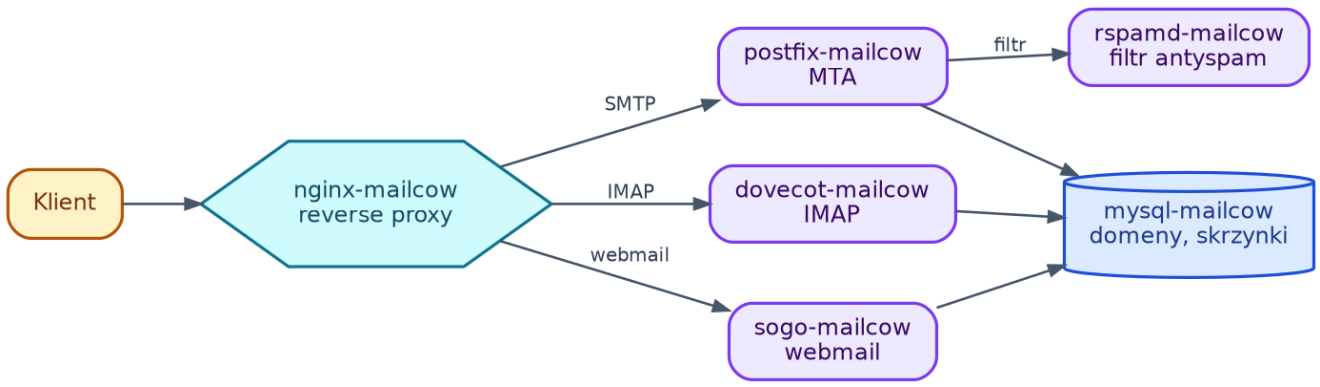
Układ docker-compose i mailcow.conf

Pojęcie: mailcow.conf. Główny plik konfiguracyjny instalacji, leżący w katalogu instalacyjnym MailCow (np. `/docker/apps/mailcow/mailcow.conf`). Zawiera zmienne środowiskowe czytane przez `docker-compose.yml`: nazwę hosta, port bazy danych, hasło roota MySQL, zakresy adresów IP dopuszczone do panelu administracyjnego i wersję gałęzi aktualizacji. Zmiana wartości w tym pliku nie działa sama z siebie, wymaga ponownego złożenia kontenerów, jak pokazujemy niżej przy aktualizacji.

Kontenery składowe i za co odpowiadają

> **Pojęcie: kontener.** Pojedynczy, izolowany proces uruchamiany z obrazu Docker, opisany szerzej na stronie o Dockerze w tym katalogu wiedzy. MailCow rozbija odpowiedzialności serwera pocztowego na osobne kontenery zamiast jednego monolitycznego procesu, między innymi po to, żeby awaria jednego komponentu (np. webmaila) nie wyłączała pozostałych (np. przyjmowania poczty).

Kontener	Rola
`postfix-mailcow`	MTA: przyjmowanie i wysyłanie poczty po SMTP
`dovecot-mailcow`	MDA i serwer IMAP/POP3: faktyczne przechowywanie skrzynek i dostęp klientów pocztowych
`mysql-mailcow` (MariaDB)	baza danych z domenami, skrzynkami, aliasami i regułami `sender_acl`
`sogo-mailcow`	webmail oraz serwer CalDAV/CardDAV (kalendarze, kontakty)
`rspamd-mailcow`	filtr antyspamowy i silnik reguł DKIM/SPF/DMARC po stronie serwera
`nginx-mailcow`	reverse proxy wewnętrzny, wystawiający panel administracyjny i webmail na zewnątrz
`redis-mailcow`	pamięć podręczna i magazyn stanu dla kilku pozostałych kontenerów
`dockerapi-mailcow`	API do zarządzania samymi kontenerami z poziomu panelu administracyjnego



Nginx-mailcow kieruje ruch do postfixa, dovecota albo sogo zależnie od protokołu, a wszystkie trzy trzymają dane we wspólnej bazie mysql-mailcow (kliknięcie otwiera pełny rozmiar)

Kopia zapasowa i odtworzenie

Wykonanie pełnej kopii zapasowej: backup_and_restore.sh

> **Pojęcie: backup_and_restore.sh.** Skrypt dołączony do instalacji MailCow, obsługujący zarówno tworzenie, jak i odtwarzanie kopii zapasowej wszystkich istotnych danych: bazy `mysql` (razem z kalendarzami i kontaktami SOGo), skrzynek pocztowych (`vmail`), Redisa z kluczami DKIM (`redis`), danych `rspamd` i `postfix` oraz danych szyfrowania (`crypt`). Argument `all` wykonuje kopię wszystkich tych komponentów naraz; da się też wskazać pojedynczy komponent (np. tylko `vmail`), gdy potrzebujemy odtworzyć jedynie same wiadomości bez ruszania konfiguracji.

Aktualizacja: update.sh

> **Pojęcie: update.sh.** Skrypt aktualizujący zarówno pliki konfiguracyjne (`docker-compose.yml`, `mailcow.conf`) do najnowszej wersji z repozytorium, jak i same obrazy kontenerów. Przed uruchomieniem zawsze wykonujemy pełną kopię zapasową, bo aktualizacja potrafi zmieniać schemat bazy danych i nie da się jej bezpiecznie cofnąć bez kopii sprzed zmiany.

Dodanie domeny i skrzynki

Codzienne zarządzanie domenami, skrzynkami, aliasami i limitami wykonujemy przez panel administracyjny WebUI, a nie ręczną edycją bazy danych: panel zapisuje te same tabele, ale przy okazji dba o spójność (np. tworzenie odpowiednich katalogów Maildir, wpisów DKIM). Bezpośredni dostęp do bazy, pokazany wyżej przy `sender_acl`, zostaje do diagnostyki i do operacji, których panel nie udostępnia.

> **Pojęcie: alias.** Dodatkowy adres przekierowujący pocztę do już istniejącej skrzynki, bez zajmowania osobnego miejsca na dysku: jedna skrzynka `biuro@zielona.example` może mieć dowolnie wiele aliasów, np. `kontakt@zielona.example` i `info@zielona.example`, wszystkie trafiające do tej samej fizycznej skrzynki.

> **Pojęcie: limit skrzynki (quota).** Maksymalny rozmiar skrzynki w megabajtach, ustawiany indywidualnie dla każdej skrzynki w panelu. Po przekroczeniu limitu Dovecot odrzuca kolejne przychodzące wiadomości z kodem błędu informującym nadawcę o przepełnionej skrzynce, zamiast po cichu je gubić.

Dodanie uprawnienia Send As z domeny zewnętrznej

> **Pojęcie: Send As (sender_acl).** Uprawnienie pozwalające jednej skrzynce wysyłać wiadomości z adresem nadawcy innym niż jej własny, np. skrzynka `sekretariat@zielona.example` wysyłająca w imieniu `zarzad@zielona.example`. Reguły tego typu leżą w tabeli `sender_acl` bazy danych MailCow, ale dodajemy je w panelu: w oknie edycji skrzynki pole „Allow to send as” obsługuje adresy z naszych domen, a pole „External sender addresses” adresy z domeny zewnętrznej (nienależącej do tej instalacji MailCow). To drugie pole wymaga uprawnienia „Allow to extend sender ACL by external addresses”, które administrator może odebrać administratorom domen. Dokładny opis jest na stronie MailCow: codzienna praca z domenami, skrzynkami i spamem.

Certyfikat i odnowienie

MailCow zarządza certyfikatem TLS dla panelu, webmaila i usług SMTP/IMAP samodzielnie przez wbudowaną integrację z Let's Encrypt, uruchamianą przez kontener `acme-mailcow`. W typowej instalacji nie wymaga to ręcznej ingerencji, o ile port 80 jest dostępny z internetu dla walidacji domeny. Na `docker01` porty 80 i 443 zajmuje Traefik, więc tam MailCow ma wyłączone własne Let's Encrypt (`SKIP_LETS_ENCRYPT=y`), a certyfikat dostarczamy według strony Instalacja MailCow na docker01 od zera. Ręczne wskazanie własnego certyfikatu (np. z wewnętrznego urzędu certyfikacji firmy) opisujemy na stronie Certyfikaty SSL: wystawienie, konwersja, weryfikacja, pliki podmieniamy w katalogu danych `data/assets/ssl/` i restartujemy kontenery `postfix-mailcow`, `dovecot-mailcow` i `nginx-mailcow`.

Częste błędy

Aktualizacja bez wcześniejszej kopii zapasowej. `update.sh` potrafi zmienić schemat bazy danych: bez świeżej kopii cofnięcie nieudanej aktualizacji jest znacznie trudniejsze niż samo jej wykonanie.

Edycja tabel bazy danych zamiast panelu tam, gdzie panel wystarcza. Bezpośrednia edycja SQL pomija logikę panelu (np. generowanie DKIM przy nowej domenie, tworzenie katalogów Maildir przy nowej skrzynce). Zostawiamy ją tylko do diagnostyki i operacji faktycznie niedostępnych z poziomu WebUI. Send As, także z domeny zewnętrznej, ustawiamy w oknie edycji skrzynki.

Restart całego stosu zamiast jednego kontenera. `docker compose down && up` zrywa wszystkie połączenia IMAP i sesje SMTP na całej instalacji, podczas gdy problem dotyczył jednego kontenera. Najpierw sprawdzamy logi, dopiero potem decydujemy o zakresie restartu.

□□ Ćwiczenia

> Ćwiczenia zakładają dostęp do testowej instalacji MailCow (np. w maszynie wirtualnej) i konto z uprawnieniami administratora panelu.

Ćwiczenie 1: pełny cykl kopii zapasowej i weryfikacji

```
cd /docker/apps/mailcow
sudo env MAILCOW_BACKUP_LOCATION=/docker/kopie/mailcow THREADS=2 ./helper-
scripts/backup_and_restore.sh backup all
ls -la /docker/kopie/mailcow/mailcow-*/
```

Wykonujemy kopię zapasową i sprawdzamy, jakie katalogi faktycznie powstały dla poszczególnych komponentów.

Odpowiedź

Powinniśmy zobaczyć jeden katalog ze znacznikiem czasu w nazwie (`mailcow-RRRR-MM-DD-GG-MM-SS`), a wewnątrz niego osobne archiwa dla każdego komponentu: `backup_mariadb.tar.zst`, `backup_vmail.tar.zst`, `backup_redis.tar.zst`, `backup_rspamd.tar.zst`, `backup_postfix.tar.zst`, `backup_crypt.tar.zst`, plus plik `mailcow.conf`. Brak któregoś z archiwów oznacza, że kopia danego komponentu się nie powiodła.

Powiązane: Postfix: konfiguracja · Jak działa poczta: MTA, MDA, MUA, SMTP, kolejka · Certyfikaty SSL: wystawienie, konwersja, weryfikacja

Źródła: dokumentacja mailcow.email/docs · `helper-scripts/backup_and_restore.sh --help`

Revision #38
Created 2025-03-13 13:29:58 UTC by Tomasz Konieczny
Updated 2026-10-26 14:20:29 UTC by Tomasz Konieczny