Migracja z Zimbry do MailCow bez utraty poczty

Typ: procedura

Dotyczy: Zimbra (zmprov, zmmailbox), MailCow Dockerized, imapsync wbudowany w zadania synchronizacji (sprawdzono w dokumentacji 3.10.2026)

Czas: ~2 tygodnie kalendarzowe, ~6 godzin pracy

Efekt: wszystkie skrzynki, aliasy, listy dystrybucyjne, kalendarze i kontakty z `zimbra01` działają w MailCow na `docker01`, rekord MX wskazuje nową pocztę, a stara Zimbra zostaje do odczytu, dopóki nikt nie potwierdzi, że niczego nie brakuje.

Skróty użyte na tej stronie (11)

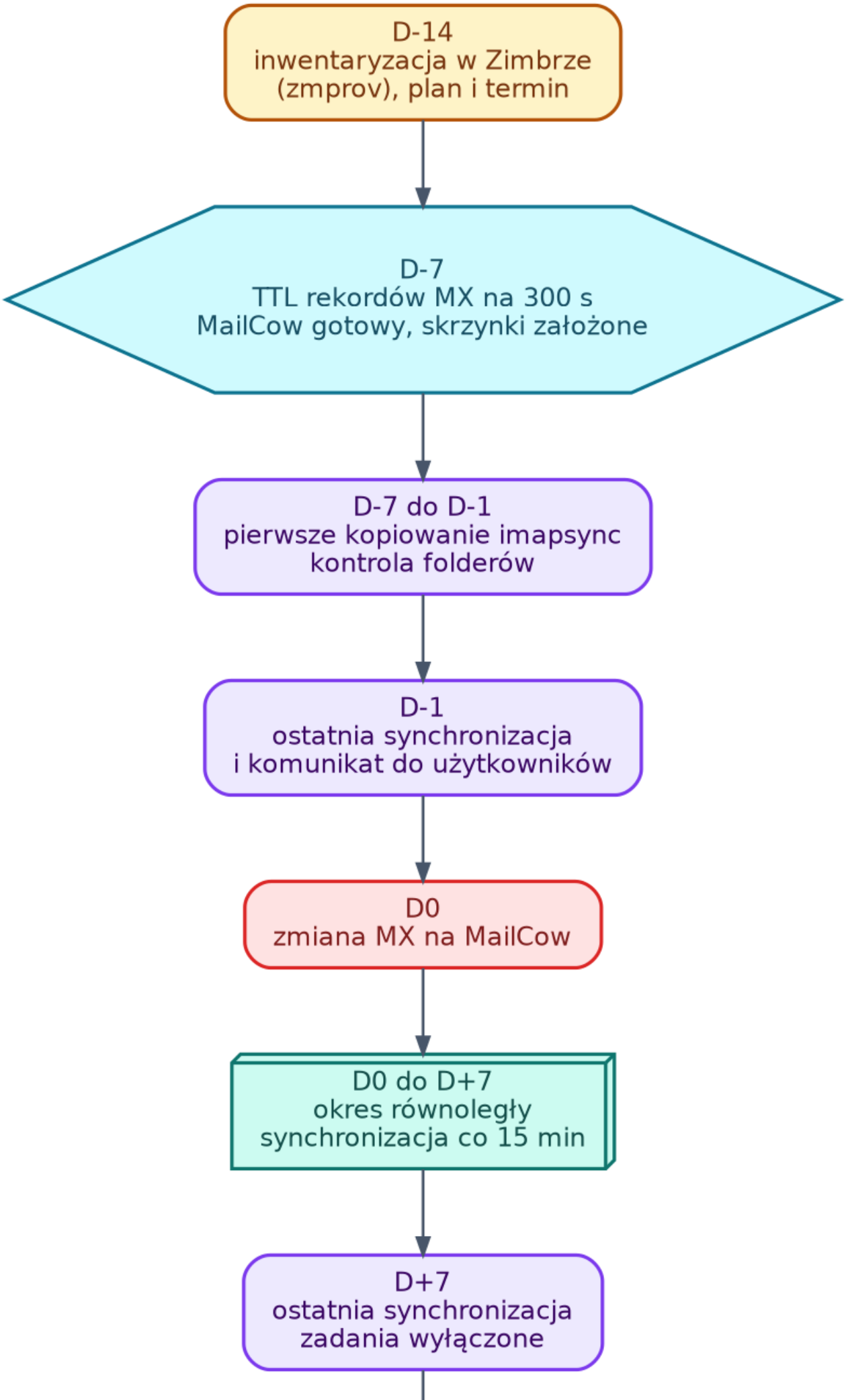
Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
CSV	w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
ICS	iCalendar: format pliku z terminami kalendarza, który potrafią wczytać prawie wszystkie programy kalendarzowe
IMAP	Internet Message Access Protocol: protokół, którym klient pocztowy pobiera wiadomości z serwera, zostawiając je tam zamiast kasować
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
LDIF	LDAP Data Interchange Format: tekstowy plik z wpisami katalogu, tu używany do przenoszenia kontaktów
MX	Mail Exchanger: rekord DNS wskazujący serwer poczty domeny
SOGo	webmail z kalendarzem i kontaktami, który w MailCow jest dostępny w przeglądarce obok panelu administracyjnego
SSL	Secure Sockets Layer: dawna nazwa szyfrowania, dziś formalnie TLS
TTL	Time To Live: czas życia rekordu lub danych

Zanim zaczniemy

> **□□ Z ŻYCIA WZIĘTE**
>
> Poniedziałek, 8:05, tydzień po przełączeniu poczty. Pani Halina dzwoni, że w nowej skrzynce nie ma folderu „Spam od dostawców (do sprawdzenia)", w którym od lat trzyma faktury wpadające do spamu, a właśnie zamyka miesiąc. Tomek otwiera zadanie synchronizacji jej skrzynki i widzi w polu wykluczeń wzorzec, który przepisał z innej instrukcji: pomijał wszystkie foldery ze słowem „spam" w nazwie. Wzorzec znika, zadanie rusza ponownie, a folder wraca po kilku minutach, bo Zimbra wciąż działa w okresie równoległym. Gdyby ją wygaszono tydzień wcześniej, faktur nie dałoby się odzyskać.

Komendy Zimbry wykonujemy w bash na `zimbra01` jako użytkownik `zimbra` (`sudo su - zimbra`), komendy MailCow na `docker01` z katalogu `/docker/apps/mailcow`. Zakładamy, że MailCow jest już zainstalowany według Instalacja MailCow na docker01 od zera, a domena `zielona.example` założona w panelu.

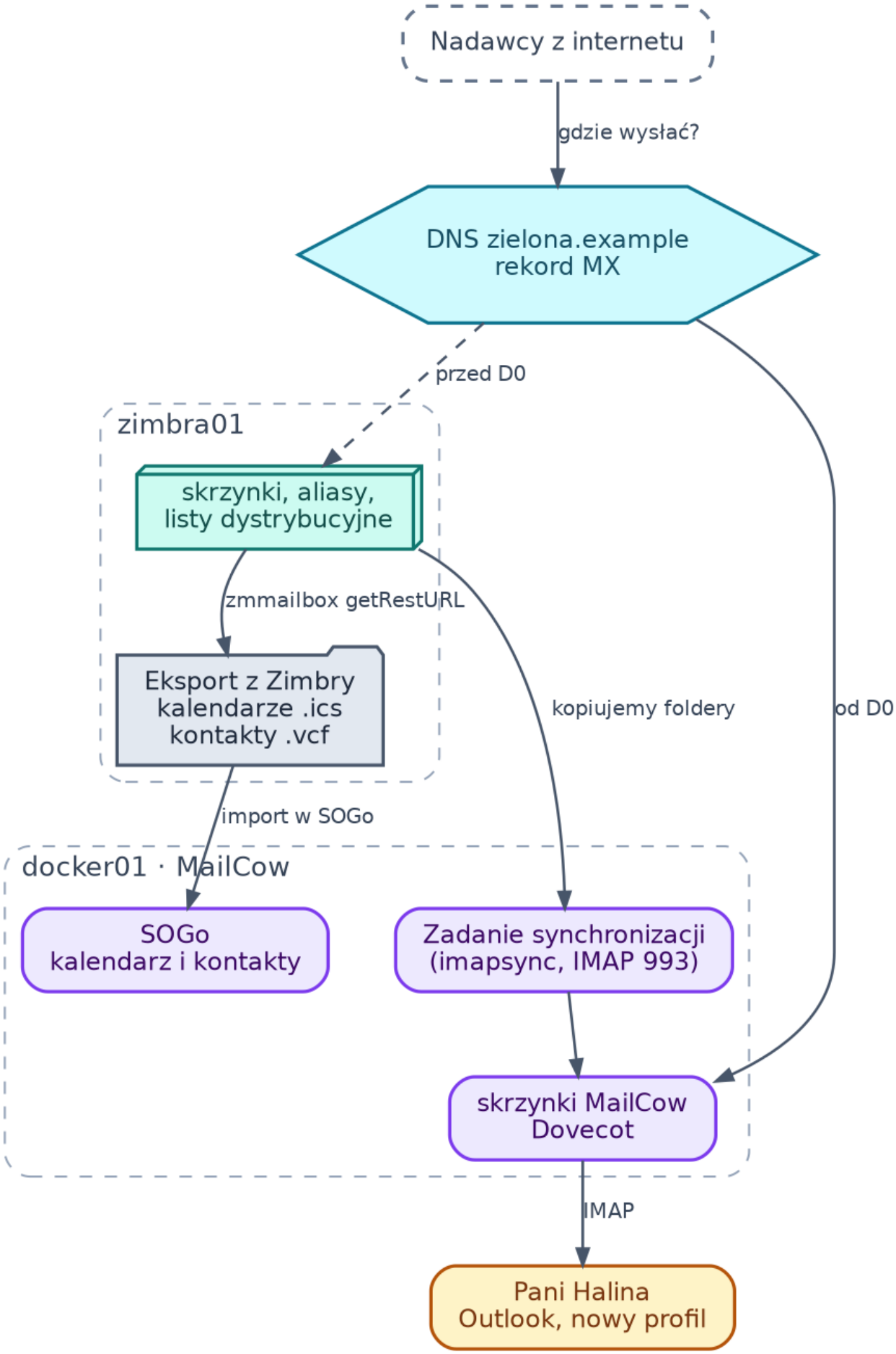
Migracja poczty to nie jedno kopiowanie, tylko plan rozłożony na dwa tygodnie. Przez pierwszy tydzień kopiujemy pocztę, ale ludzie nadal pracują w Zimbrze. W dniu przełączenia zmieniamy kierunek nowej poczty, a przez następny tydzień Zimbra jeszcze żyje i dogrywa to, co spóźnione.


D-14
inwentaryzacja w Zimbrze
(zmprov), plan i termin
D-7
TTL rekordów MX na 300 s
MailCow gotowy, skrzynki założone
D-7 do D-1
pierwsze kopiowanie imapsync
kontrola folderów
D-1
ostatnia synchronizacja
i komunikat do użytkowników
D0
zmiana MX na MailCow
D0 do D+7
okres równoległy
synchronizacja co 15 min
D+7
ostatnia synchronizacja
zadania wyłączone

Oś czasu migracji: od inwentaryzacji w D-14, przez przełączenie MX w D0, do wygaszenia Zimbry w D+30 (kliknięcie otwiera pełny rozmiar)

Dni liczymy od przełączenia (D0). Dzień D0 wybieramy według kalendarza Zielonej: nie w ostatnie trzy dni robocze miesiąca, nie w piątek po 15:00 i nie po 15 listopada, kiedy obowiązuje zakaz zmian w infrastrukturze. Najlepsza jest sobota rano.

Wymaganie	Uwaga
Działający MailCow z domeną `zielona.example`	patrz Instalacja MailCow na docker01 od zera
Dostęp do `zimbra01` jako `zimbra` i do panelu MailCow jako administrator	komendy `zmprov` i `zmmailbox` wymagają konta `zimbra`, patrz Zimbra
Klucz API MailCow z prawem zapisu	do masowego zakładania skrzynek i aliasów, patrz krok 6
IMAP Zimbry osiągalny z `docker01` na porcie `993`	sprawdzamy w kroku 5
Wolne miejsce w `/docker` równe około 1,3 rozmiaru wszystkich skrzynek	liczymy w kroku 3
Prawo zmiany rekordów DNS domeny `zielona.example` i reguł przekierowania na `fw01`	zmiany robimy w krokach 10 i 13
Pani Halina, Marek i reszta uprzedzeni o terminie	komunikat w kroku 12


Nadawcy z internetu
gdzie wysłać?
DNS zielona.example
rekord MX
przed D0
od D0
zimbra01
skrzynki, aliasy,
listy dystrybucyjne
zmmailbox getRestURL
Eksport z Zimbry
kalendarze .ics
kontakty .vcf
kopiujemy foldery
import w SOGo
docker01 · MailCow
SOGo
kalendarz i kontakty
Zadanie synchronizacji
(imapsync, IMAP 993)
skrzynki MailCow
Dovecot
IMAP
Pani Halina
Outlook, nowy profil

Przepływ migracji: DNS decyduje, dokąd trafia nowa poczta, a zadania synchronizacji kopiują skrzynki z zimbra01, podczas gdy kalendarze i kontakty jadą przez pliki (kliknięcie otwiera pełny rozmiar)

Poczta wędruje dwiema drogami. Skrzynki kopiuje zadanie synchronizacji, które samo loguje się do Zimbry po IMAP i przepisuje foldery do MailCow. Kalendarze i kontakty nie są pocztą, więc przenosimy je plikami eksportu i importujemy w SOGo.

> **Pojęcie: imapsync.** Program, który kopiuje skrzynkę z jednego serwera IMAP na drugi, folder po folderze, wiadomość po wiadomości, i potrafi robić to wielokrotnie bez tworzenia duplikatów, bo rozpoznaje wiadomości po nagłówkach. MailCow ma go wbudowanego i udostępnia w panelu jako zadania synchronizacji.

> **Pojęcie: TTL rekordu DNS.** Liczba sekund, przez które inne serwery mogą pamiętać odpowiedź na pytanie o ten rekord. Jeśli rekord MX ma TTL jednego dnia, po jego zmianie część internetu przez dobę nadal wysyła pocztę starą drogą, dlatego TTL obniżamy z wyprzedzeniem.

Przebieg

1. Wejście na konto zimbra: sudo su

```
sudo su - zimbra
```

Wszystkie komendy z kroków 2 do 4 wykonujemy z tego konta na `zimbra01`.

2. Inwentaryzacja skrzynek: zmprov gaa

```
zmprov -l gaa NazwaDomeny | grep -v -E '^(spam|ham|virus-quarantine|galsync)\.' | sort >
/tmp/skrzynki.txt
wc -l /tmp/skrzynki.txt
```

(gdzie `NazwaDomeny` to `zielona.example`, a przełącznik `-l` każe `zmprov` czytać katalog LDAP bezpośrednio, co opisuje `zmprov --help`)

Lista kont zawiera też konta techniczne Zimbry (filtry spamu, kwarantanna wirusów), które odfiltrowujemy. Konta `admin` i ewentualne konta usługowe usuwamy z pliku ręcznie, jeśli nie mają

trafić do MailCow.

Przykład:

```
87 /tmp/skrzynki.txt
```

3. Rozmiary skrzynek: zmprov gqu

```
zmprov gqu NazwaSerwera | sort -k3 -n -r | head -20
```

Wynik pokazuje zajętość skrzynek na serwerze. Kolejność kolumn sprawdzamy w `zmprov help gqu`, bo od niej zależy `sort -k3`. Dwadzieścia największych skrzynek dostaje od nas osobną uwagę: kopiują się godzinami i najczęściej mają duże foldery, które łatwo przegapić przy sprawdzaniu.

4. Zapisanie aliasów i list dystrybucyjnych: zmprov ga i gdl

Aliasy każdego konta zapisujemy do pliku, w formacie `alias;konto`:

```
while read -r konto; do
  zmprov -l ga "$konto" zimbraMailAlias | grep '^zimbraMailAlias:' | sed "s|^zimbraMailAlias:
||; s|\$|;${konto}|"
done < /tmp/skrzynki.txt > /tmp/aliasy.csv
```

Listy dystrybucyjne zapisujemy w formacie `lista;adres1,adres2,...`:

```
for lista in $(zmprov gadl); do
  czlonkowie=$(zmprov gdl "$lista" | grep -E '^\s+\S+@\S+$' | tr -d ' ' | paste -sd, -)
  echo "${lista};${czlonkowie}"
done > /tmp/listy.csv
```

Format wyniku `gdl` sprawdzamy na jednej liście, zanim puścimy pętlę na wszystkie, bo zależy od wersji Zimbry. W MailCow lista dystrybucyjna odpowiada aliasowi z wieloma adresami docelowymi.

Przykład (`/tmp/aliasy.csv` i `/tmp/listy.csv`):

```
ksiegowosc@zielona.example;halina.kowalska@zielona.example
magazyn@zielona.example;marek.zawadzki@zielona.example,swarzedz@zielona.example
```

5. Sprawdzenie, czy IMAP Zimbry odpowiada: openssl s_client

Wracamy na `docker01` i sprawdzamy, czy do Zimbry da się dotrzeć po IMAP z szyfrowaniem:

```
openssl s_client -connect zimbra01.ipa.zielona.example:993 </dev/null 2>/dev/null | head -3
```

Gdy połączenie się nie udaje, sprawdzamy zaporę między `docker01` a `zimbra01` i to, czy usługa IMAP w Zimbrze jest włączona.

6. Założenie skrzynek w MailCow: API add/mailbox

Klucz API tworzymy w panelu MailCow w ustawieniach dostępu, w sekcji API, i nadajemy mu prawo zapisu. Klucz podajemy zmienną środowiskową, nie wpisujemy go w skrypt.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: panel MailCow, ustawienia dostępu, sekcja API, wiersz z nowym kluczem w trybie odczyt i zapis. Klucz zamazujemy.

Skrypt zakłada skrzynkę dla każdego adresu z listy, z losowym hasłem (hasła i tak zastąpi logowanie domenowe, patrz MailCow: konta z Active Directory):

```
export MAILCOW_API_KEY="KluczApi"
while read -r adres; do
  czesc="${adres%@*}"
  domena="${adres#*@}"
  haslo="$(openssl rand -base64 18)"
  curl -s -X POST "https://poczta.zielona.example/api/v1/add/mailbox" \
    -H "X-API-Key: ${MAILCOW_API_KEY}" -H "Content-Type: application/json" \
    -d
"{\"local_part\":\"${czesc}\",\"domain\":\"${domena}\",\"name\":\"${czesc}\",\"quota\":5120,\"
password\":\"${haslo}\",\"password2\":\"${haslo}\",\"active\":true,\"force_pw_update\":false}"
  echo
done < skrzynki.txt
```

(gdzie `quota` to limit skrzynki w megabajtach, który dla większych skrzynek ustawiamy ręcznie na podstawie kroku 3, a `skrzynki.txt` to plik z kroku 2 skopiowany na `docker01`)

Aliasy i listy zakładamy przez ten sam interfejs, osobnym punktem `add/alias`:

```
while IFS=';' read -r alias cel; do
  curl -s -X POST "https://poczta.zielona.example/api/v1/add/alias" \
    -H "X-API-Key: ${MAILCOW_API_KEY}" -H "Content-Type: application/json" \
    -d "{\"address\":\"${alias}\",\"goto\":\"${cel}\",\"active\":true}"
  echo
done < aliasy.csv
```

Dla listy dystrybucyjnej uruchamiamy tę samą pętlę na pliku `listy.csv`, bo pole `goto` przyjmuje adresy rozdzielone przecinkami. Typy pól API sprawdzamy w dokumentacji API na `https://poczta.zielona.example/api`, bo mogą różnić się między wersjami MailCow.

Przykład:

```
[{"type":"success","log":["mailbox","add","mailbox",{"local_part":"halina.kowalska","domain":"
zielona.example"}],"msg":["mailbox_added","halina.kowalska@zielona.example"]}]
```

7. Zapamiętanie skrótu hasła i ustawienie tymczasowego: zmprov sp

Zadanie synchronizacji musi zalogować się do skrzynki w Zimbrze, a my nie znamy haseł użytkowników. Dla jednego konta zapamiętujemy skrót hasła i ustawiamy hasło tymczasowe:

```
umask 077
zmprov -l ga NazwaSkrzynki userPassword | grep '^userPassword:' | awk '{print $2}' >
/tmp/NazwaSkrzynki.hash
zmprov sp NazwaSkrzynki HasloTymczasowe
```

Po zakończeniu synchronizacji oryginalne hasło przywracamy z zapisanego skrótu:

```
zmprov ma NazwaSkrzynki userPassword "$(cat /tmp/NazwaSkrzynki.hash)"
```

> Metodę najpierw sprawdzamy na jednym koncie testowym: ustawiamy hasło tymczasowe, przywracamy skrót i logujemy się starym hasłem. Pliki ze skrótami są wrażliwe, więc po całej migracji je usuwamy. Zmianę wykonujemy wieczorem, bo w oknie między ustawieniem a przywróceniem hasła klient pocztowy użytkownika nie zaloguje się do Zimbry.

Do dnia D0 robimy to wieczorem i przywracamy hasło następnego ranka. Po D0 użytkownicy nie pracują już w Zimbrze, więc hasła tymczasowe zostawiamy do końca okresu równoległego i przywracamy je dopiero wtedy, gdy Zimbra zostaje do odczytu.

8. Utworzenie zadania synchronizacji: panel MailCow

```
E-Mail → Configuration → Synchronizations
```

Dla każdej skrzynki wybieramy nazwę z listy „Username", wpisujemy hosta `zimbra01.ipa.zielona.example` i port `993`, w polach „Username" i „Password" podajemy dane konta w Zimbrze (hasło tymczasowe z kroku 7), szyfrowanie ustawiamy na SSL, zaznaczamy „Active" i klikamy „Add". Do dnia D0 zadanie włączamy wieczorem i wyłączamy rano, razem z hasłem tymczasowym z kroku 7. Po D0 zostaje włączone i działa ciągle, z interwałem 15 minut ustawionym w polu interwału. Pola wykluczeń i mapowania folderów czytamy przed pierwszym uruchomieniem: jeśli pole wykluczeń zawiera wzorzec pomijający foldery, nasze foldery o takiej nazwie nie przejdą.

□□ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: panel MailCow, E-Mail, Configuration, Synchronizations, formularz nowego zadania wypełniony dla konta `halina.kowalska@zielona.example` z hostem `zimbra01.ipa.zielona.example`, portem 993 i szyfrowaniem SSL, oraz widoczne pole wykluczeń.

Nazwy pól mogą różnić się od naszego opisu w zależności od wersji MailCow, więc w razie wątpliwości sprawdzamy dokumentację rozdziału „Sync job migration". Zadanie wykonuje się samo w odstępach, które ustawiliśmy w polu interwału.

9. Próba bez kopiowania na jednym koncie: imapsync --dry

Jeśli na `docker01` mamy zainstalowany sam program imapsync, możemy na koncie testowym sprawdzić, jak ułożą się foldery, zanim cokolwiek zostanie skopiowane:

```
imapsync --host1 zimbra01.ipa.zielona.example --user1 NazwaSkrzynki --passfile1
/tmp/haslo1.txt --host2 poczta.zielona.example --user2 NazwaSkrzynki --passfile2
/tmp/haslo2.txt --ssl1 --ssl2 --automap --dry
```

Opcja `--dry` sprawia, że imapsync niczego nie robi naprawdę, tylko wypisuje, co by zrobił, a `--passfile1` i `--passfile2` czytają hasła z plików, żeby nie zostawiały śladu w historii poleceń. Folder

o nazwie z kropką albo ukośnikiem sprawdzamy tu szczególnie uważnie, bo oba serwery mogą inaczej zapisywać hierarchię folderów. To krok opcjonalny, bo ten sam efekt daje zadanie testowe w panelu na koncie `test.migracja`.

10. Obniżenie TTL rekordów MX: sprawdzenie dig

Obniżamy TTL rekordu MX i rekordów `autodiscover` oraz `autoconfig` do 300 sekund w panelu DNS domeny `zielona.example`. Robimy to co najmniej tyle czasu przed przełączeniem, ile wynosił poprzedni TTL, czyli przy TTL jednego dnia najpóźniej w D-1, w praktyce w D-7. Sprawdzamy wynik z zewnątrz sieci Zielonej:

```
dig +noall +answer MX zielona.example
```

Przykład:

```
zielona.example.    300    IN    MX    10 mail.zielona.example.
```

11. Eksport kalendarza i kontaktów: zmmailbox getRestURL

Kalendarz każdego użytkownika eksportujemy do pliku w formacie ICS:

```
zmmailbox -z -m NazwaSkrzynki getRestURL '/Calendar?fmt=ics' > /tmp/NazwaSkrzynki-kalendarz.ics
```

Kontakty eksportujemy tak, żeby SOGo mógł je wczytać. SOGo importuje kontakty z plików vCard i LDIF, nie z CSV, więc wybieramy archiwum z osobnymi plikami vCard:

```
zmmailbox -z -m NazwaSkrzynki getRestURL '/Contacts?fmt=zip' > /tmp/NazwaSkrzynki-kontakty.zip
```

Zawartość archiwum sprawdzamy na koncie testowym. Pliki vCard łączymy w jeden plik, a w SOGo importujemy go z menu przy „Personal Address Book" (opcja „Import"), kalendarz z menu przy „Personal Calendar".

Eksport z Zimbry nie przenosi się sam, więc tabela pokazuje, co dzieje się z resztą.

Element	Przenosi się?	Jak postępujemy

Foldery i wiadomości	tak	zadanie synchronizacji (imapsync)
Flagi przeczytania i gwiazdki	tak	idą razem z wiadomościami przez IMAP
Tagi kolorowe Zimbry	nie	użytkownik ustawia kategorie od nowa
Aliasy i listy dystrybucyjne	nie samo	skrypty z kroków 4 i 6
Kalendarz	po eksporcie	plik ICS importujemy w SOGo, udostępnienia ustawiamy od nowa
Kontakty	po eksporcie	plik vCard importujemy w SOGo
Filtry i reguły poczty	nie	odtwarzamy ręcznie z listy zrzuconej przez użytkownika
Odpowiedzi automatyczne i przekierowania	nie	ustawiamy ręcznie
Udostępnione foldery i delegacje	nie	ustawiamy ręcznie po stronie MailCow
Zadania, Teczka, Notatki	nie	eksportujemy archiwum pełne (krok 14) i zostawiamy w archiwum
Hasła	nie	użytkownicy logują się hasłem domenowym

12. Powiadomienie użytkowników: wiadomość e-mail

Wysyłamy w D-7 i przypominamy w D-1. Treść trzymamy krótką, bo ludzie czytają dwa pierwsze zdania.

```
Temat: Zmiana poczty firmowej w sobotę, co robimy w poniedziałek

W sobotę przenosimy pocztę firmową na nowy serwer. Wszystkie Wasze wiadomości,
foldery, kontakty i kalendarze przenosimy my, nie musicie kopiować niczego sami.

W poniedziałek rano:
1. Zamykamy Outlooka i uruchamiamy go ponownie. Jeśli poprosi o hasło,
   wpisujemy hasło domenowe, takie jak do komputera.
2. Gdy Outlook nie zgłosi się sam, dzwonimy do IT, a my zakładamy nowy profil
   przy Waszym biurku.
3. Po przejrzeniu skrzynki zgłaszamy brakujące foldery od razu, w poniedziałek.

Stara poczta zostaje do odczytu jeszcze przez dwa tygodnie.
```

```
Zespół IT
```

Panią Halinę prosimy dodatkowo o listę folderów, na których jej szczególnie zależy, i sprawdzamy je razem z nią w poniedziałek rano.

13. Przełączenie MX: rekord DNS i przekierowanie na fw01

W dniu D0 robimy dwie zmiany w jednym oknie. Pierwsza to rekord MX domeny `zielona.example`: z `mail.zielona.example` na `poczta.zielona.example`, priorytet `10`. Druga to przekierowania portów na `fw01`, które dziś kierują port `25` (i pozostałe porty pocztowe) na `zimbra01`, a po zmianie mają kierować na `docker01`.

> Zielona ma jeden adres publiczny `203.0.113.10`, więc rekord MX sam niczego nie przełączy: obie nazwy wskazują ten sam adres. O tym, czy pocztę odbiera `zimbra01`, czy `docker01`, decyduje przekierowanie portu `25` na `fw01`. Dlatego obie zmiany robimy razem, a rekord `autodiscover` przestawiamy w tym samym oknie.

Po zmianie sprawdzamy z zewnątrz sieci Zielonej (z hotspotu telefonu), dokąd trafia poczta:

```
swaks --to AdresOdbiorcy@zielona.example --from AdresNadawcy
```

Przykład:

```
=== Trying poczta.zielona.example:25...
=== Connected to poczta.zielona.example.
```

14. Pełne archiwum skrzynki przed wygaszeniem: zmmailbox getRestURL tgz

```
zmmailbox -z -m NazwaSkrzynki getRestURL '/?fmt=tgz' > /mnt/backup/zimbra/NazwaSkrzynki.tgz
```

Archiwum zawiera całą skrzynkę razem z kalendarzem i kontaktami w formacie Zimbry. Nie wgrywamy go do MailCow, tylko zostawiamy na `bkp01` jako zabezpieczenie na wypadek brakującego folderu, którego nikt wcześniej nie zauważył.

15. Pilnowanie okresu równoległego: kto jeszcze pisze do Zimbry

```
sudo grep -o 'client=[^,]*' /var/log/zimbra.log | sort | uniq -c | sort -rn | head -20
```

Po D0 do Zimbry nie powinna już dochodzić nowa poczta zewnętrzna, ale mogą pisać do niej urządzenia i aplikacje skonfigurowane na stałe, jak skaner w Swarzędzu czy aplikacje Oli. Adresy z tego zestawienia sprawdzamy i przestawiamy na `poczta.zielona.example`.

Przykład:

```
    412 client=skaner-sw01.ad.zielona.example[10.20.30.15]
     38 client=app01.ad.zielona.example[10.20.10.41]
      3 client=mail-ed1-f43.google.com[209.85.208.43]
```

16. Sprawdzenie, czy nic nie zginęło: doveadm mailbox status

W MailCow liczymy wiadomości w każdym folderze skrzynki i porównujemy z liczbami w Zimbrze:

```
cd /docker/apps/mailcow
sudo docker compose exec dovecot-mailcow doveadm mailbox status -u
halina.kowalska@zielona.example messages '*'
```

Po stronie Zimbry liczby folderów odczytujemy poleceniem `zmmailbox -z -m halina.kowalska@zielona.example gaf`. Porównujemy folder po folderze, a nie tylko sumy, bo brakujący mały folder gubi się w sumie.

Przykład:

```
INBOX messages=2418
Sent messages=1960
Faktury dostawcow messages=1287
Spam od dostawcow (do sprawdzenia) messages=64
```

17. Wygaszenie Zimbry: zmcontrol stop

Siedem dni po D0 (D+7) wykonujemy ostatnią synchronizację i wyłączamy zadania w panelu MailCow. Czternaście dni po D0 przywracamy hasła z kroku 7 i przestawiamy Zimbrę na tylko do odczytu, czyli zamykamy jej porty pocztowe na `fw01` i zostawiamy dostęp do panelu z sieci wewnętrznej, a użytkowników prosimy o potwierdzenie, że niczego nie brakuje. Trzydzieści dni po D0, po kopii końcowej z kroku 14, zatrzymujemy usługi:

```
sudo su - zimbra -c "zmcontrol stop"
```

Maszynę wirtualną `zimbra01` zostawiamy wyłączoną, a po okresie przechowywania kopii usuwamy rekordy DNS i reguły przekierowania, które jej dotyczyły.

Weryfikacja

Sprawdzamy, że wszystko, co obiecaliśmy w efekcie, działa:

1. Liczba skrzynek w MailCow jest równa liczbie w `/tmp/skrzynki.txt`.
2. Dla trzech wylosowanych skrzynek liczba wiadomości w każdym folderze (krok 16) zgadza się z Zimbrą.
3. Wiadomość wysłana z zewnątrz na `AdresOdbiorcy@zielona.example` trafia do MailCow (krok 13).
4. Alias `ksiegowosc@zielona.example` dostarcza pocztę do skrzynki pani Haliny.
5. Kalendarz pani Haliny w SOGo zawiera terminy z zamknięcia miesiąca z Zimbry.

Wycofanie zmian

Do dnia D+14 powrót jest prosty, bo Zimbra nie została tknięta. Cofamy przekierowania portów na `fw01` na `zimbra01` i rekord MX na `mail.zielona.example`; ponieważ TTL wynosi 300 sekund, całość wraca w kilka minut.

Wiadomości, które w międzyczasie trafiły do MailCow, kopiujemy z powrotem do Zimbry programem imapsync w odwrotnym kierunku (`--host1` to MailCow, `--host2` to `zimbra01`), a skrzynki zostawiamy w MailCow bez zmian, żeby po usunięciu przyczyny przełączyć je ponownie. Po D+30 i zatrzymaniu Zimbry powrót wymaga jej uruchomienia (`zmcontrol start`) i odtworzenia stanu z kopii.

Gdy coś nie zadziała

Objaw	Przyczyna	Co robimy

Zadanie synchronizacji kończy się błędem logowania	hasło tymczasowe nie zostało ustawione albo ma znaki, które psują zapis	sprawdzamy logowanie do `zimbra01:993` ręcznie, ustawiamy proste hasło tymczasowe
Brakuje folderów w MailCow	wzorzec w polu wykluczeń albo nietypowe znaki w nazwie folderu	czytamy pole wykluczeń, uruchamiamy zadanie ponownie, sprawdzamy foldery z kroku 16
Poczta nadal ląduje w Zimbrze po D0	urządzenie albo aplikacja ma wpisany adres `zimbra01`	zestawienie z kroku 15, zmiana hosta SMTP na `poczta.zielona.example`
Outlook pyta o hasło i nie widzi folderów	stary profil trzyma ustawienia z Zimbry	zakładamy nowy profil, sprawdzamy rekord `autodiscover`
Zadanie synchronizacji trwa godzinami dla jednej skrzynki	skrzynka ma kilka dużych folderów	zostawiamy do końca, kontrolujemy w logach `sudo docker compose logs --tail 100 dovecot-mailcow`
Po przywróceniu hasła użytkownik nie loguje się do Zimbry	skrót hasła zapisany z błędem albo zmieniony w międzyczasie	ustawiamy hasło ponownie przez `zmprov sp`, użytkownik zmienia je sam
Kontakty importują się bez pierwszego wpisu	znany błąd importu LDIF w SOGo	sprawdzamy po imporcie pierwszy kontakt i dopisujemy go ręcznie

□□ Ćwiczenia

Ćwiczenie 1: kiedy obniżyć TTL

Rekord MX ma dziś TTL jednego dnia (86400 s), a przełączenie planujemy na sobotę rano. Najpóźniej kiedy musimy obniżyć TTL i dlaczego robimy to wcześniej?

Odpowiedź

Najpóźniej w piątek rano, bo stary TTL musi zdążyć wygasnąć w pamięciach innych serwerów, zanim zmienimy rekord. W praktyce robimy to tydzień wcześniej, żeby zostawić zapas na błąd, na przykład gdy zmiany w DNS publikują się z opóźnieniem.

Ćwiczenie 2: dlaczego nie kopiujemy plików

Dlaczego nie przekopiować po prostu katalogów skrzynek z `zimbra01` do MailCow, zamiast używać imapsync?

Odpowiedź

Zimbra trzyma wiadomości we własnym magazynie z osobną bazą metadanych, a MailCow (Dovecot) w katalogach Maildir. Formaty są różne, więc skopiowane pliki nic by w MailCow nie znaczyły. Imapsync rozmawia z oboma serwerami po IMAP, czyli w języku, który rozumieją oba.

Ćwiczenie 3: lista kontrolna dla pani Haliny

Pani Halina ma 60 folderów po dostawcach, kalendarz z terminami zamknięcia miesiąca i regułę poczty przenoszącą faktury do osobnego folderu. Co z tego przeniesie się samo, a co ustawiamy ręcznie?

Odpowiedź

Foldery z wiadomościami przeniesie zadanie synchronizacji (po sprawdzeniu pola wykluczeń). Kalendarz przenosimy plikiem ICS, importowanym w SOGo. Reguły poczty nie przenoszą się, więc odtwarzamy ją ręcznie z panią Haliną w poniedziałek, najlepiej z jej opisem, jakie faktury ma wpadać do jakiego folderu.

Ćwiczenie 4: przełączenie przy jednym adresie publicznym

Zmieniliśmy rekord MX, ale testowa wiadomość z hotspotu nadal ląduje w Zimbrze. Co sprawdzamy w pierwszej kolejności?

Odpowiedź

Przekierowanie portu `25` na `fw01`. Zielona ma jeden adres publiczny, więc oba rekordy MX prowadzą w to samo miejsce i o tym, która maszyna odbiera pocztę, decyduje reguła przekierowania. Dopóki wskazuje ona na `zimbra01`, zmiana rekordu niczego nie zmienia.

Powiązane: Zimbra · MailCow · Instalacja MailCow na docker01 od zera · MailCow: konta z Active Directory · Jak działa poczta: MTA, MDA, MUA, SMTP, kolejka · Diagnostyka dostarczania

Źródła: docs.mailcow.email, rozdział „Sync job migration" · docs.mailcow.email i `https://poczta.zielona.example/api` (API MailCow: `add/mailbox`, `add/alias`) · imapsync.lamiral.info i github.com/imapsync/imapsync (opcje `--dry`, `--passfile1`, `--automap`) · community.zextras.com, artykuły o eksporcie i imporcie danych kont Zimbry oraz o poleceniach `zmprov` · dokumentacja SOGo i jej instrukcje importu (kontakty vCard i LDIF, kalendarz ICS)

Revision #18
Created 2026-10-03 09:39:01 UTC by Tomasz Konieczny
Updated 2026-10-26 14:20:35 UTC by Tomasz Konieczny