# Perforce i Lore: porównanie dróg do wysokiej dostępności na Windowsie

**Poziom: średnio**

**Dotyczy: P4 Server 2026.1 (dawniej Helix Core), Lore 0.10.0, Windows Server 2022, klaster Hyper-V hv01 i hv02 na S2D, Hyper-V Replica, Storage Replica**

**Czas: ~60 min**

*Poprzednia strona opisuje dwie drogi do rezerwy dla Perforce i Lore: maszynę wirtualną jako rolę klastra Hyper-V (a) i rolę w klastrze gościnnym z dyskiem VHD Set (b). Tu zestawiamy je z kolejnymi pomysłami (replika standby, kopia na trzeci host, magazyn S3 dla Lore, zimna rezerwa), mówimy, co użytkownik zobaczy w czterech sytuacjach, i wybieramy, od czego Zielona zaczyna. Wracamy tu, gdy ktoś pyta „a może by tak inaczej” albo gdy test przełączenia pokaże wynik gorszy od oczekiwanego.*

> Ta strona to dodatek z przykładem praktycznym. Zasady ogólne: Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi

> Nic z tej strony nie było uruchomione na prawdziwych hostach Zielonej. Czasy podajemy jako rzędy wielkości do zmierzenia testem, a wszystko, czego nie udało się potwierdzić w dokumentacji producenta, oznaczamy „do potwierdzenia”. Stan źródeł: 3 października 2026.

> **ZAŁOŻENIA SCENARIUSZA**
>
> Zielona ma tylko dwa hosty Windows, `hv01` i `hv02`, w jednej serwerowni w Pamiątkowie i nie ma trzeciego hosta. Swarzędz jako trzecie miejsce zakłada osobną lokalizację z tunelem IPsec do `fw01`: IPsec site-to-site: Pamiątkowo i Swarzędz między FortiGate a routerem innej firmy, ale to decyzja o sprzęcie i łączu, a nie założenie tej strony.

**Skróty użyte na tej stronie (19)**

| Skrót | Co znaczy |
|---|---|
| **AWS** | Amazon Web Services: chmura Amazonu |
| **CNAME** | Canonical Name: rekord DNS będący aliasem, drugą nazwą wskazującą na już istniejący rekord A |
| **DNS** | Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP |
| **DR** | Disaster Recovery: plan przywrócenia działania po poważnej awarii |
| **DynamoDB** | baza danych typu klucz-wartość w chmurze AWS; Lore trzyma w niej fragmenty, stan, magazyn zmienny i blokady, gdy działa na AWS |
| **FAQ** | Frequently Asked Questions: zbiór najczęściej zadawanych pytań z odpowiedziami |
| **HA** | High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu |
| **HAProxy** | High Availability Proxy: program, który rozdziela połączenia między kilka serwerów i omija te, które nie odpowiadają |
| **IPsec** | Internet Protocol Security: zestaw protokołów szyfrujących i uwierzytelniających ruch sieciowy, na nim stoi tunel VPN typu IKEv2 |
| **MinIO** | program do własnego magazynu obiektowego zgodnego z S3; wersja społecznościowa jest od 2026 roku zarchiwizowana i dostępna tylko jako kod źródłowy |
| **RGW** | RADOS Gateway: brama w systemie Ceph, która udostępnia jego magazyn obiektowy przez interfejs zgodny z S3 |
| **RPO** | Recovery Point Objective: ile danych wolno stracić przy awarii |
| **RTO** | Recovery Time Objective: jak długo wolno nie działać przy awarii |
| **S2D** | Storage Spaces Direct: programowa macierz Windows Server z dysków wielu serwerów |
| **S3** | Simple Storage Service: magazyn plików w chmurze AWS |
| **SDP** | Server Deployment Package: zestaw skryptów i układ katalogów Perforce do wdrażania serwera P4 na Linuksie, z usługami nazwanymi p4d_N |
| **TCP** | Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia |
| **TLS** | Transport Layer Security: szyfrowanie połączeń, np. w HTTPS |
| **VHD** | Virtual Hard Disk: plik, który Hyper-V pokazuje maszynie wirtualnej jako dysk; wspólny dysk dwóch maszyn ma format VHD Set (.vhds) |

# Fiszka

Polecenia `p4` wykonujemy na stacji administratora jako konto z uprawnieniem super, a polecenia PowerShell na `hv01` lub `hv02` w sesji administratora. Karty dotyczą drogi z repliką standby (c1) i drogi z kopią na trzeci host (c2), więc przydadzą się dopiero po ich wdrożeniu. Adres `10.20.10.85` to propozycja dla standby `p4d02`, nazwa `perforce` to alias z poprzedniej strony.

🔒 TYLKO ODCZYT

## Podgląd przełączenia na standby bez wykonania: p4 failover

```
p4 -p p4d02.ad.zielona.example:1666 failover
```

Polecenie działa na serwerze standby. Bez opcji `-y` tylko raportuje, co by zrobiło, więc służy do sprawdzenia gotowości (monitoring na standby, specyfikacje serwerów) [2]. Treść odpowiedzi zapisujemy z laboratorium, bo dokumentacja jej nie podaje.

🔒 TYLKO ODCZYT

## Sprawdzenie, na jakim punkcie dziennika jest standby: p4 pull -l -j

```
p4 -p p4d02.ad.zielona.example:1666 pull -l -j
```

Pokazuje stan dziennika repliki i serwera nadrzędnego oraz czas ostatniej zmiany pliku stanu [8]. Gdy numery sekwencji się różnią i różnica rośnie, standby nie nadąża.

Przykład (kształt odpowiedzi z dokumentacji, liczby przykładowe):

```
Current replica journal state is: Journal 41, Sequence: 1200345.
Current master journal state is: Journal 41, Sequence: 1200345.
The statefile was last modified at: 2026/10/02 07:12:03.
The Server time is currently: 2026/10/02 07:12:04 +0200 CEST
```

🔒 TYLKO ODCZYT

## Sprawdzenie zaległych transferów plików na standby: p4 pull -l -s

```
p4 -p p4d02.ad.zielona.example:1666 pull -l -s
```

Wypisuje podsumowanie oczekujących transferów plików wersjonowanych [8]. Metadane jadą przez dziennik, a pliki osobno (`pull -u`), więc po awarii mogą brakować najświeższych plików. Tę lukę sprawdzamy w teście przez `p4 verify -q //...`.

🔒 TYLKO ODCZYT

### Sprawdzenie stanu wszystkich replik na serwerze centralnym: p4 servers -J

```
p4 -p perforce.ad.zielona.example:1666 servers -J
```

Z opcją `-J` pokazuje dla każdej repliki, do którego miejsca dziennika doszła ona w zapisie i w odtworzeniu [9]. Rosnąca różnica między polami `PersistedSequence` i `AppliedSequence` oznacza, że replika zostaje w tyle.

**TYLKO ODCZYT**

### Sprawdzenie stanu replikacji maszyny Hyper-V: Measure-VMReplication

```
Measure-VMReplication -VMName "p4d01"
```

Wypisuje statystyki replikacji maszyny: stan, kondycję (`Normal`, `Warning`, `Critical`), czas ostatniej replikacji, średnie opóźnienie i rozmiar zaległości [15]. Wykonujemy na hoście, który prowadzi maszynę, po wdrożeniu drogi (c2).

# Z życia wzięte

**Z ŻYCIA WZIĘTE**

Czwartek, 9:20. Tomek kładzie na stole wydruk strony o klastrze Perforce i Lore, a Ola pyta, dlaczego nikt nie sprawdził repliki standby, skoro producent sam ją poleca. Pan Zbigniew chce wiedzieć jedno: czy Lore da się zrobić tak samo. Tomek zapisuje oba pytania na tablicy i okazuje się, że żadnego z nich nie umie rozstrzygnąć z głowy.

Autor poprosił o porównanie dróg (a) i (b) dla obu usług oraz o poszukiwanie innych. Strona zbiera wyniki takiego poszukiwania: co działa, co działa z zastrzeżeniami i co odpada, każde z odnośnikiem do dokumentacji. Na końcu stoi rekomendacja w dwóch wariantach dla Perforce i osobno dla Lore.

# Jak to działa

**Rozwiń: jak to działa pod spodem**

# Przypomnienie dróg (a) i (b)

Droga (a) to maszyna wirtualna `p4d01` albo `lore01` jako rola klastra Hyper-V: planowe wyłączenie hosta przechodzi w migracji na żywo, a po awarii hosta maszyna startuje na drugim hoście od zera. Droga (b) to dwie maszyny `p4n1` i `p4n2` w klastrze gościnnym `ftgc01` z dyskami VHD Set: po awarii przechodzi tylko usługa, bez rozruchu systemu. Procedury, polecenia i testy obu dróg są na stronie Perforce i Lore w klastrze: cztery drogi, rekomendacja dla Zielonej i test przełączenia [13], a pojęcia roli, zasobu i kworum na stronie Klaster Windows dla własnych usług: rola, zasób, zależności, kworum i testy przełączenia. Tu ich nie powtarzamy.

Obie drogi mają wspólną cechę, o której łatwo zapomnieć: dane serwera leżą w jednym egzemplarzu na wspólnym dysku. Lustro S2D chroni przed utratą dysku fizycznego, ale nie przed uszkodzeniem zawartości plików ani skasowaniem danych (zasada „Jedna kopia danych to nie wysoka dostępność” na stronie Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi).

# Cztery sytuacje, które porównujemy

Każdą drogę oceniamy w czterech sytuacjach. Planowe wyłączenie hosta to aktualizacja albo przegląd, awaria hosta to nagła utrata zasilania, awaria systemu gościa to zawieszenie lub restart Windows w maszynie, a uszkodzenie bazy to stan, w którym serwer nie chce wstać albo zwraca błędy tabel. Do tego liczymy RPO, RTO, liczbę maszyn, licencje i złożoność. Różnicę między planowym wyłączeniem a awarią tłumaczą zasady przekrojowe ze strony Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi.

> **Pojęcie: HA i DR w słowniku Perforce.** Perforce rozróżnia dwa przypadki przełączenia. HA (wysoka dostępność) to zaplanowane prace, przy których serwer centralny bierze udział w przełączeniu i nie tracimy danych. DR (odtwarzanie po katastrofie) to sytuacja, w której serwer centralny i jego standby nie mogą pracować [1].

> **Pojęcie: replika standby.** Drugi serwer P4, który na bieżąco kopiuje dziennik serwera głównego (`journalcopy`), odtwarza go (`pull -L`) i pobiera pliki wersjonowane (`pull -u`). Poleceniem `p4 failover` zamienia się w serwer główny. Przełączenie nie jest automatyczne: wykonuje je ręcznie administrator z uprawnieniem super [1][3].

Hyper-V Replica i Storage Replica to technologie odtwarzania po awarii, a nie wysokiej dostępności [14][16]; różnicę opisuje sekcja „HA a odtwarzanie po katastrofie” na stronie Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi.

# Lista dróg, które oceniamy

| Oznaczenie | Droga | Dla której usługi |
|---|---|---|
| (a) | maszyna wirtualna jako rola klastra Hyper-V | Perforce i Lore |
| (b) | rola Generic Service albo Generic Application w klastrze gościnnym z VHD Set | Perforce i Lore |
| (c1) | replika standby i `p4 failover` | Perforce |
| (c1b) | forwarding replica, commit-edge, P4 Proxy i P4 Broker | Perforce |
| (c2) | Hyper-V Replica albo Storage Replica na trzeci host | Perforce i Lore |
| (c3) | magazyn zgodny z S3 (i baza zgodna z DynamoDB) zamiast dysku, serwery Lore bezstanowe za HAProxy | Lore |
| (c3b) | replika odczytu Lore w dwupoziomowej hierarchii serwerów | Lore |
| (c3c) | Lore z S3 i DynamoDB w AWS | Lore |
| (c4) | zimna rezerwa: kopia i odtworzenie na nowej maszynie | Perforce i Lore |
| (c5) | kopia bazy `offline_db` ze skryptów SDP dla Windows (dodatek do (a) i (b)) | Perforce |

# Perforce: porównanie dróg

Duża tabela ma jedenaście kolumn, których nie da się czytać na jednym ekranie, więc dzielimy ją na dwie połowy z tymi samymi wierszami. Pierwsza pokazuje, co widzi użytkownik, druga ile to kosztuje i jak to sprawdzić.

# Co widzi użytkownik

| Droga | Planowe wyłączenie hosta | Awaria hosta | Awaria systemu gościa | Uszkodzenie bazy |
|---|---|---|---|---|
| (a) maszyna jako rola klastra | migracja na żywo: niewidoczne albo pauza rzędu sekund | błąd połączenia, rozruch maszyny na drugim hoście: minuty, operacje w toku do powtórzenia | zawieszony system klaster hostów widzi tylko częściowo, zwykle potrzebny ręczny restart (do potwierdzenia) | serwer nie startuje albo zgłasza błędy tabel, odtwarzamy z checkpointu i dziennika, z `offline_db` krócej |
| (b) klaster gościnny, VHD Set | maszyny węzłów przechodzą na żywo, niewidoczne, o ile węzeł nie wypadnie z klastra (test T-b7), aktualizacja gościa to przełączenie roli z przerwą | rola startuje na drugim, już działającym węźle: sekundy do minuty, połączenia zerwane | węzeł wypada z klastra, rola przechodzi, zawieszenie procesu przy działającej usłudze widzi tylko monitoring | jedna kopia na VHD Set, skutek jak w (a), kontrola przed startem nie pozwala wstać na uszkodzonych tabelach |
| (c1) droga (a) plus standby `p4d02` | (a) niewidoczne, a planowe przełączenie na standby (`p4 failover` z udziałem serwera głównego) daje krótką przerwę bez utraty danych | maszyna startuje w (a) albo ręczny `p4 failover` na standby: czas decyzji plus kilka minut, zmiany niezreplikowane mogą przepaść | jak awaria hosta: ręczny `p4 failover` | standby ma własną kopię metadanych odtwarzaną z dziennika, więc awans omija uszkodzone pliki `db.*`, ale skasowanie i błąd logiczny dotrą na standby |
| (c1b) forwarding replica, commit-edge, proxy | nie zmieniają nic: to rozwiązania wydajnościowe | serwer centralny nadal jeden | jak wyżej | jak wyżej |
| (c2) kopia na trzeci host | nic się nie zmienia, replika nie przejmuje pracy | nic, dopóki administrator nie uruchomi repliki: minuty do godzin | nic | replika dostaje uszkodzenie z opóźnieniem, a punkty odtwarzania Hyper-V Replica (do 24 godzinnych) pozwalają cofnąć maszynę [14] |
| (c4) zimna rezerwa | przerwa na czas prac, nic nie przejmuje | przerwa na czas budowy nowej maszyny i odtworzenia: godziny | jak awaria hosta | odtwarzanie z kopii na `bkp01` |

# Ile to kosztuje i jak to sprawdzamy

| Droga | RPO | RTO | Maszyny | Licencje | Złożoność | Co przetestować |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| (a) | bliskie zeru po awarii hosta (dane na S2D), do godziny po utracie puli (obrót dziennika co godzinę) | minuty do kilkunastu (rozruch plus start serwera), do zmierzenia | 1 maszyna ( `p4d01` ) | jedna licencja serwera | niska | T1, T2 i T3 z poprzedniej strony, odtworzenie z kopii |
| (b) | jak (a) | sekundy do minut plus kontrola przed startem, do zmierzenia | 2 maszyny ( `p4n1` , `p4n2` ) i świadek na `bkp01` | jedna licencja serwera | wysoka | T-b1 do T-b7 z poprzedniej strony |
| (c1) | planowo zero, po awarii to, co nie dotarło przez `journalcopy` (do zmierzenia) [1] | planowo minuty, po awarii czas decyzji plus przełączenie plus zmiana aliasu (TTL jedna minuta) | 2 maszyny ( `p4d01` , `p4d02` na drugim hoście) | osobna licencja dla standby [6][7] | wysoka | T4 z poprzedniej strony i testy T-c1 do T-c4 niżej |
| (c1b) | nie dotyczy | nie dotyczy | dodatkowe serwery bez roli rezerwy | nie wdrażamy | wysoka | nie wdrażamy |
| (c2) | Hyper-V Replica: 30 sekund, 5 lub 15 minut do wyboru, Storage Replica: zero (synchroniczni e, do 5 ms) albo bliskie zeru [14][16] | start ręczny: godziny | trzeci host Windows, którego Zielona dziś nie ma | Hyper-V Replica bez dopłaty w Windows Server [14], Storage Replica w Standard ma limity (1 partnerstwo, 1 wolumen, 2 TB) [16] | średnia | test failover ( `Test Failover` ), `Measure-VMReplication` |
| (c4) | do ostatniej kopii: checkpoint co noc, dziennik co godzinę | godziny | brak dodatkowych | jak (a) | niska | kwartalne odtworzenie na maszynie pomocniczej |

Czasy w tabelach to rzędy wielkości, a nie obietnice. Pewne są tylko zdania z dokumentacji (przełączenie ręczne, osobna licencja standby, bez utraty danych przy udziale serwera głównego), a każdą liczbę zastępujemy wynikiem testu.

# Lore: porównanie dróg

Lore jest przed wersją 1.0 (najnowsze wydanie 0.10.0), a dokumentacja uprzedza, że interfejsy, formaty i protokoły mogą się zmieniać między wydaniami, choć dane zapisane w Lore mają pozostać czytelne dla przyszłych wydań [20]. Wydanie 0.9.0 zmieniło protokół replikacji i schemat tabel DynamoDB, więc replikę i jej serwer nadrzędny trzeba aktualizować razem, a wdrożenie na AWS wymagało migracji metadanych [26]. Przy takim tempie zmian prostota drogi ma większą wagę niż przy Perforce.

# Co widzi użytkownik

| Droga | Planowe wyłączenie hosta | Awaria hosta | Awaria systemu gościa | Uszkodzenie magazynu |
|---|---|---|---|---|
| (a) maszyna jako rola klastra | migracja na żywo: niewidoczne albo pauza | błąd połączenia, rozruch na drugim hoście: minuty, przesyłanie od początku | jak w Perforce (do potwierdzenia) | serwer odmawia startu albo zwraca błędy, odtwarzamy z kopii katalogu magazynów |
| (b) klaster gościnny, Generic Application albo NSSM | maszyny węzłów przechodzą na żywo, niewidoczne albo pauza | program startuje na drugim węźle: sekundy do minut, zdobyte blokady znikają | węzeł wypada, rola przechodzi | jedna kopia na VHD Set, skutek jak w (a) |
| (c3) magazyn zgodny z S3 i dwie instancje za HAProxy | niewidoczne: HAProxy wyłącza instancję z rotacji | HAProxy zauważa awarię po około dziewięciu sekundach (kontrola co 3 s, trzy porażki), połączenia zerwane, klient ponawia | jak awaria hosta | dane leżą w magazynie, nie na serwerze, więc awaria magazynu jest awarią usługi |
| (c3b) replika odczytu | odczyty z repliki działają, zapis (push) czeka na serwer nadrzędny | jak obok, a czy replika odpowiada przy niedostępnym serwerze nadrzędnym, dokumentacja nie mówi (do potwierdzenia) | jak obok | replika jest kopią serwera nadrzędnego, nie źródłem prawdy |
| (c3c) S3 i DynamoDB w AWS | niewidoczne | instancje wymienne, magazyn trwały po stronie AWS | jak awaria hosta | magazyn zarządzany przez AWS |
| (c2) kopia na trzeci host | nic się nie zmienia | nic, dopóki administrator nie uruchomi repliki | nic | kopia jest spójna jak po nagłym wyłączeniu zasilania |
| (c4) zimna rezerwa | przerwa na czas prac | godziny | godziny | odtwarzanie katalogu magazynów z `bkp01` |

# Ile to kosztuje i jak to sprawdzamy

| Droga | RPO | RTO | Maszyny | Licencje | Złożoność | Co przetestować |
|---|---|---|---|---|---|---|
| (a) | do kilkunastu sekund po awarii hosta (zapis magazynów co `flush_delay_seconds`, u nas 10), do ostatniej kopii po utracie puli (założenie macierzy: 24 godziny) | minuty | 1 maszyna (`lore01`) | licencja MIT, bez opłat [19] | niska | T1 i T2 z poprzedniej strony, odczyt i zapis testowego repozytorium |
| (b) | jak (a) | sekundy do minut | 2 maszyny i trzeci dysk `.vhds` | MIT | wysoka | testy z poprzedniej strony, sposób zatrzymania programu przy przenoszeniu |
| (c3) | zero, jeśli magazyn jest trwały | sekundy | 2 instancje Lore, `lb01` i `lb02`, co najmniej trzy węzły magazynu zgodnego z S3 i trzy węzły bazy zgodnej z DynamoDB | MIT, a ewentualne licencje magazynu i bazy do sprawdzenia | bardzo wysoka | nie do wdrożenia, patrz niżej |
| (c3b) | zapis: bez zmian, odczyt: aktualność repliki do zmierzenia | odczyt: sekundy, zapis: jak (a) | 1 dodatkowa maszyna (`lore02`, propozycja `10.20.10.86`) | MIT | średnia, procedury w dokumentacji brak | odczyt przy zatrzymanym serwerze nadrzędnym, wersje po aktualizacji |
| (c3c) | zero | sekundy | instancje w AWS | opłaty AWS | średnia | poza zakresem |

| Droga | RPO | RTO | Maszyny | Licencje | Złożoność | Co przetestować |
|---|---|---|---|---|---|---|
| (c2) | Hyper-V Replica: 30 sekund do 15 minut, kopia spójna jak po awarii zasilania | godziny | trzeci host Windows | jak w Perforce | średnia | test failover, odczyt i zapis po starcie repliki |
| (c4) | do ostatniej kopii (doba), kopia przy zatrzymanym serwerze | godziny | brak dodatkowych | MIT | niska | kwartalne odtworzenie |

# Opis dróg poza (a) i (b)

Każda droga ma ocenę w trzech stopniach: działa, działa z zastrzeżeniami albo nie dla Zielonej. Ocena dotyczy wyłącznie naszego środowiska (dwa hosty Hyper-V z S2D, dwóch administratorów, Windows), a nie wartości samej techniki.

# (c1) Replika standby i p4 failover

Ocena: działa z zastrzeżeniami. To droga wysokiej dostępności, którą Perforce opisuje we własnej dokumentacji, i ta, która leczy słabość dróg (a) i (b): druga, niezależna kopia metadanych [1][12].

Dokumentacja ustala warunki: `p4 failover` działa tylko na standby lub forwarding-standby, standby musi mieć licencję odpowiednią do nowej roli, a monitoring na standby musi być włączony przy starcie serwera [1][2]. Standby do HA konfigurujemy ze specyfikacją `Services: standby`, poleceniami startowymi `journalcopy`, `pull -L` i `pull -u` oraz `rpl.journalcopy.location=1`, a serwer docelowy ma mieć poziom bezpieczeństwa 4 lub wyższy [3]. Perforce zaleca dla HA i DR właśnie standby z `rpl.journalcopy.location=1` [4]. Przełączenie nie jest automatyczne, a w jego trakcie klienci nie wykonują poleceń [1][2].

Utrata danych zależy od udziału serwera głównego: gdy bierze udział, nie tracimy nic, a gdy go nie ma, tracimy transakcje, które nie zdążyły dotrzeć przez `journalcopy` [1]. Na Windowsie po `p4 failover -y` dochodzą kroki ręczne: sprawdzenie ustawień usługi poleceniem `p4 set -S`, zmiana `P4NAME` i `P4LOG` na wartości roli serwera głównego oraz restart usługi poleceniem `svcinst` [10]. Przewodnik SDP Failover Guide obejmuje Unix i Windows [10], a własne wskazówki Perforce dla chmury też zalecają standby w osobnej strefie dostępności i zmianę DNS po przełączeniu [12].

Serwer zapasowy wymaga osobnej licencji, Perforce przewiduje taką formę [6][7].

Z drogą (a) standby łączy się dobrze: `p4d01` zostaje rolą klastra, a `p4d02` stoi na drugim hoście. Z drogą (b) techniczne połączenie jest możliwe, ale dwa mechanizmy przełączania tego samego serwera (klaster i `p4 failover`) dają dwa sposoby zepsucia aliasu `perforce`, który w (b) jest rekordem A punktu dostępu, a w (a) i (c1) rekordem CNAME. Łączymy więc (c1) tylko z (a).

# (c1b) Forwarding replica, commit-edge, P4 Proxy i P4 Broker

Ocena: nie dla Zielonej. Commit-edge daje „najlepszą wydajność, bo większość poleceń jest lokalna”, a dokumentacja nie przedstawia go jako rozwiązania HA; edge nie może być standby dla serwera commit, choć może mieć własny standby [4][5]. Forwarding replica filtruje metadane i odciąża serwer centralny, a dla DR i HA Perforce zaleca zwykły standby [4]. P4 Proxy buforuje wersje plików, P4 Broker kieruje polecenia, a producent wskazuje dla jego dawnych zastosowań nowsze repliki i serwery edge [4].

Forwarding-standby jest odmianą standby, więc może być celem `p4 failover`, ale przełączenie na zwykły standby jest szybsze [1]. Zielona ma jedną grupę użytkowników w jednym biurze, więc nie ma wydajnościowego powodu, żeby dokładać edge. Wiele serwerów to więcej do utrzymania bez korzyści dla dostępności.

# (c2) Hyper-V Replica albo Storage Replica na trzecie miejsce

Ocena: działa z zastrzeżeniami, ale jako odtwarzanie po katastrofie, nie wysoka dostępność. Hyper-V Replica replikuje maszynę asynchronicznie co 30 sekund, 5 minut albo 15 minut, trzyma do 24 godzinnych punktów odtwarzania, ma replikację rozszerzoną na trzeci poziom i nie wymaga dopłaty w Windows Server. Przełączenie jest ręczne, a przy awarii nieplanowanej możliwa jest utrata danych [14].

Microsoft pisze też, że pliki VHD Set wspierają Hyper-V Replica, ale zachowanie klastra gościnnego `ftgc01` po replikacji sprawdzamy w laboratorium (do potwierdzenia) [17]. Kopie takich maszyn na poziomie hosta sprawiają kłopoty (błędy 32770 i 32775), więc replikację klastra gościnnego traktujemy ostrożnie [18].

Storage Replica replikuje wolumeny synchronicznie (opóźnienie do 5 ms) albo asynchronicznie, ma tryby stretch cluster, klaster do klastra i serwer do serwera, ale nie jest kopią zapasową, bo replikuje także skasowanie. W wydaniu Standard ma limity: jedno partnerstwo, jeden wolumen do 2 TB [16]. Serwer Perforce ma co najmniej dwa wolumeny (dane i dziennik), więc tu potrzebna byłaby wersja Datacenter albo jeden wspólny wolumen.

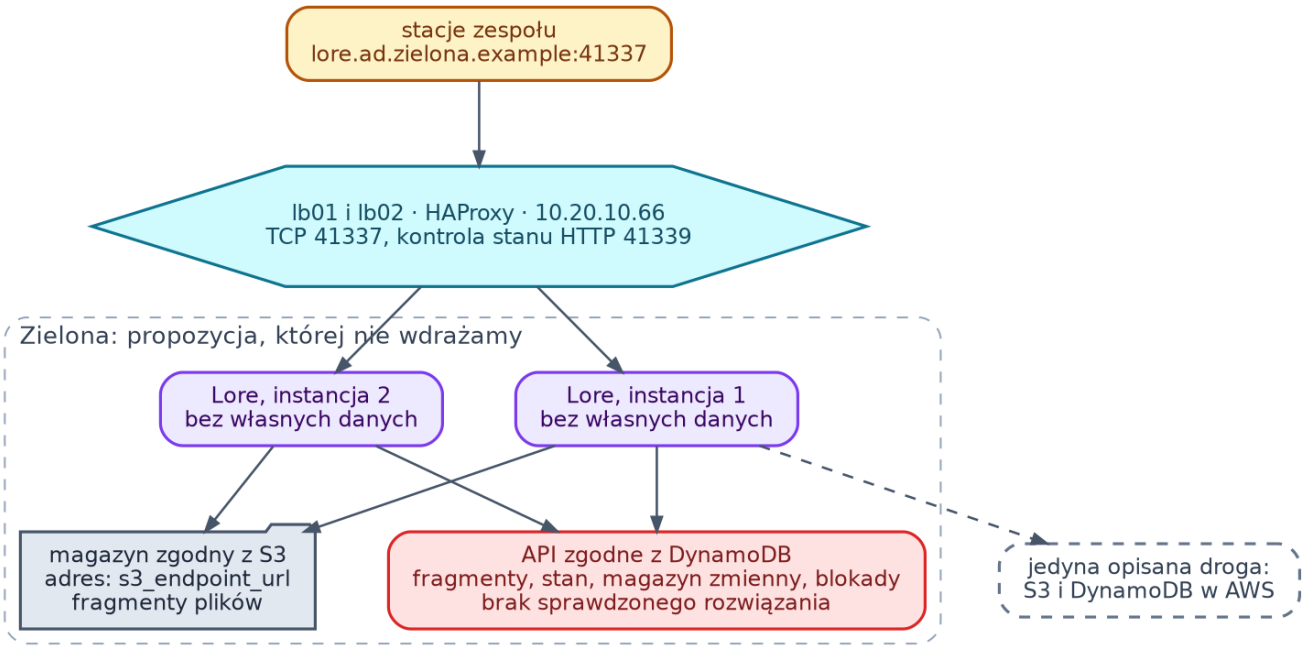
Zasadniczy problem jest organizacyjny: Zielona nie ma trzeciego hosta Windows. Dwa hosty stoją w jednej serwerowni, więc replika między nimi nie chroni przed pożarem ani zalaniem. Kandydatem na trzecie miejsce jest Swarzędz, ale to decyzja o sprzęcie i łączu, nie administratora (patrz „Decyzje autora”).

# (c3) Lore: magazyn zgodny z S3 i dwie instancje za HAProxy

Ocena: nie dla Zielonej. Odpowiadamy wprost na dwa pytania.

Czy Lore dopuszcza wiele instancji? Tak: dokumentacja pisze, że warstwa serwerów skaluje się poziomo, że serwery są bezstanowymi czytelnikami, a cały trwały stan leży w magazynach, i że w tym samym trybie pracują warstwy pamięci podręcznej i replik odczytu [20][22].

Czy Lore dopuszcza magazyn S3 poza AWS? Dokumentacja tego nie opisuje, a samo S3 i tak nie wystarcza.




*Rysunek 1. Najlepsza teoretyczna droga dla Lore: dwie bezstanowe instancje za HAProxy z wspólnym magazynem. Magazyn S3 można wskazać pod własnym adresem, ale stan serwera trzyma też DynamoDB, a zgodnego z nią rozwiązania sprawdzonego w Zielonej nie mamy. (kliknięcie otwiera pełny rozmiar)*

Szczegóły, na których opieramy ocenę:

- Referencyjna implementacja trwałego magazynu to wtyczka `aws`: S3 na niezmienne fragmenty, DynamoDB na magazyn zmienny i blokady. Dokumentacja konfiguracji mówi, że rozproszone blokady wymagają wtyczki takiej jak `aws` (DynamoDB), a tryb lokalny blokad działa w pamięci jednego procesu [19].
- Ta sama dokumentacja pisze, że stock `loreserver` nie rejestruje żadnych wtyczek, że pola wtyczek zależą od binarki, która je rejestruje, i że przewodnik o wtyczkach jest dopiero planowany [19]. Wtyczki dołącza się przez pochodny serwer, który rejestruje je przed uruchomieniem właściwego [25]. Z drugiej strony kod gałęzi main rejestruje wtyczki `aws` i `hashicorp`, oficjalny Dockerfile buduje program `loreserver`, a dwa niezależne opisy wdrożeń na AWS używają sekcji `[plugins.aws.*]` [27][28][29]. Tej sprzeczności nie rozstrzygamy na papierze: sprawdzamy w laboratorium na obrazie 0.10.0, czy `loreserver` przyjmuje taką konfigurację.
- W kodzie wtyczki (gałąź main, nie sprawdzaliśmy tagu 0.10.0) są pola `s3_endpoint_url` i `s3_force_path_style`, więc technicznie bucket może leżeć na Ceph RGW. Dokumentacja i decyzja projektowa o S3 nie wspominają o magazynach zgodnych z S3 [23][27].
- Samo S3 nie wystarcza. Konfiguracja wtyczki wymaga też tabel DynamoDB dla fragmentów i ich stanu, a magazyn zmienny i blokady są w DynamoDB w całości [27]. Decyzja projektowa wyjaśnia, że magazyn zmienny potrzebuje operacji compare-and-swap, której S3 nie daje, a Consul odrzucono jako zbyt wolny [24].
- Jedyna znana nam baza zgodna z DynamoDB, którą można postawić samemu, to ScyllaDB Alternator, a jego dokumentacja mówi, że transakcje wielu elementów ( `TransactWriteItems`) nie są zaimplementowane, podczas gdy kod wtyczki ma wywołanie `transact_write_items` [27][31]. Czy serwer używa go na ścieżce, którą potrzebujemy, nie wiemy. Alternator przy mniej niż trzech węzłach zalecałby współczynnik replikacji 1, a ten dokumentacja zostawia testom [31].
- MinIO odpada jako domyślny wybór: repozytorium jest zarchiwizowane od 25 kwietnia 2026, a wersja społecznościowa dystrybuowana jest tylko jako kod źródłowy [32]. Alternatywy (Ceph RGW, Garage, SeaweedFS) są wymienione w opracowaniach trzecich, a nie w dokumentacji Lore.

Wniosek: bezstanowe instancje Lore za HAProxy działają tak, jak opisuje dokumentacja, wyłącznie na S3 i DynamoDB w AWS. U nas do tego trzeba by dwóch instancji, magazynu obiektowego z co najmniej trzema węzłami i bazy zgodnej z DynamoDB także z trzema, czego dwa hosty Hyper-V nie dają jako niezależne awarie. Przeniesienie pojedynczego punktu awarii z dysku na magazyn i bazę nie jest wysoką dostępnością.

# (c3b) Replika odczytu Lore

Ocena: działa z zastrzeżeniami, do sprawdzenia w laboratorium, wyłącznie dla odczytu. Pytania o replikację Lore kierujemy tu, bo dokumentacja jednak ją opisuje, choć fragmentarycznie. FAQ mówi, że Lore skaluje się poziomo z warstwami pamięci podręcznej i replik odczytu przed trwałym magazynem, a pierwsze wydanie obsługuje dwupoziomową hierarchię serwerów [20]. W konfiguracji są tryby magazynu `remote`, punkty `[server.quic_internal]` i `[server.grpc_internal]` (port 41340, wzajemny TLS, domyślnie wyłączone) i lista partnerów `[topology.fixed]` z polem `locality` [19].

Ta sama dokumentacja zastrzega, że wnętrza trybów `composite` i `replicated` są poza jej zakresem, a przewodnik wdrożenia lokalnego pomija replikację i wiele hostów [19][21]. Nie znaleźliśmy więc instrukcji, jak postawić replikę na dyskach lokalnych. Nawet gdy ją postawimy, zapis (push) idzie przez serwer nadrzędny, więc replika nie jest rezerwą zapisu. Jej wartość to odciążenie odczytów i, być może, odczyt przy niedostępnym serwerze nadrzędnym, czego dokumentacja nie obiecuje.

# (c3c) Lore na S3 i DynamoDB w AWS

Ocena: nie dla Zielonej. Droga działa i jest opisana: Epic wskazuje ją jako implementację referencyjną [20], a opisy wdrożeń pokazują instancje w kontenerach za równoważnikiem TCP, z kontrolą stanu na porcie 41339 i magazynem w S3 i DynamoDB [28][29].

Serwer przy starcie sprawdza istnienie tabel i bucketu, więc potrzebuje do nich uprawnień, a jedno z wdrożeń działało na pojedynczym zadaniu [29][30]. Znanym kosztem są ograniczenia przepustowości tabel i wiaderek przy dużych importach [28].

Dla Zielonej oznacza to wynoszenie wszystkich zasobów graficznych do chmury publicznej, co jest decyzją o danych i budżecie, a nie administracyjną.

# (c4) Zimna rezerwa: kopia i odtworzenie

Ocena: działa. To punkt odniesienia: bez klastra i bez repliki, z nocnym checkpointem i godzinnym dziennikiem (Perforce) oraz kopią katalogu magazynów (Lore) na `bkp01`. RTO liczymy w godzinach, a RPO w czasie od ostatniej kopii. Zostaje na dnie każdej innej drogi, bo replika powiela błąd, a kopia go cofa [1] (zasada „Replikacja to nie kopia zapasowa” na stronie Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi).

# (c5) Kopia bazy offline_db ze skryptów SDP dla Windows

Ocena: działa, jako dodatek do (a) i (b). Skrypty SDP dla Windows utrzymują drugą kopię plików metadanych (`offline_db`): codzienna kopia skraca dziennik, odtwarza go w `offline_db`, tworzy z niej checkpoint i buduje `offline_db` od nowa, a to wszystko bez blokowania działającego serwera [11]. SDP wymaga Windows 10 albo Windows Server 2016 i PowerShell 5.x [11].

To ważne uzupełnienie poprzedniej strony: nasz nocny `p4d -jc` na działającym serwerze blokuje tabele na czas checkpointu, a `offline_db` robi tę pracę poza serwerem (skrypt nocny opisany na poprzedniej stronie). Przy uszkodzeniu bazy odtwarzanie z gotowej kopii jest szybsze niż budowa od zera. Własne wskazówki Perforce nazywają takie odtworzenie na tym samym hoście „local failover” [12]. Czy przyjmujemy skrypty SDP w całości, czy przenosimy do naszych skryptów sam

pomysł, rozstrzyga autor.

# Rekomendacja dla Zielonej

Na start dla obu usług i obu wariantów stoi to samo: droga (a), kopie z poprzedniej strony i świadek klastra `hv01` i `hv02`. Wariantów nie rozdzielamy dziś, tylko po teście, bo dopiero zmierzona przerwa i zmierzone odtwarzanie mówią, czy dodatkowa maszyna jest warta swojej ceny.

# Perforce bez p4 failover

Wariant, którego autor chciał wcześniej. Na start: (a) plus nocny checkpoint, godzinny obrót dziennika, restic na `bkp01`, a do decyzji autora kopia `offline_db` (c5) zamiast `p4d -jc` na żywo. Po teście: jeśli zmierzona przerwa po awarii hosta przekracza RTO, przechodzimy na (b). Trzeba pamiętać, co (b) daje, a czego nie: skraca czas o rozruch systemu, ale dane nadal są w jednym egzemplarzu na wspólnym dysku, więc uszkodzenia bazy nie leczy.

# Perforce z dodatkową repliką

Na start tak samo: (a) i kopie. Po pozytywnym teście (a): maszyna `p4d02` pod `10.20.10.85` na drugim hoście i droga (c1), z przełączeniem opisanym na poprzedniej stronie (krok 5) i testami T-c1 do T-c4. Droga (b) w tym wariancie odpada: standby daje to, czego (b) nie daje (drugą kopię metadanych), a mechanizmów przełączania nie mnożymy.

# Lore

Lore nie ma dziś drogi wysokiej dostępności z dokumentacją po stronie Windowsa poza (a) i (b). Na start: (a), kopie katalogu magazynów przy zatrzymanym serwerze i test przełączenia.

(b) rozważamy po teście, jeśli zmierzona przerwa jest zbyt długa, pamiętając, że sposób zatrzymania programu przy przenoszeniu roli nie jest opisany w dokumentacji.

Magazyn S3 poza AWS (c3) odrzucamy do czasu, gdy Epic opublikuje przewodnik o wtyczkach i potwierdzi zgodną z DynamoDB bazę. Replikę odczytu (c3b) sprawdzamy w laboratorium na `10.20.10.86`, bez wpływu na produkcję. Przy każdym nowym wydaniu czytamy zmiany o replikacji i magazynach.

# Co robimy teraz, a co po teście

| Krok | Kiedy | Perforce bez p4 failover | Perforce z repliką | Lore |
|---|---|---|---|---|
| Droga (a), kopie, świadek klastra | teraz | tak | tak | tak |
| Test przełączenia (T1, T2, T3) | zaraz po budowie | tak | tak | tak |
| Droga (c1), maszyna `p4d02` | po teście (a) | nie | tak | nie dotyczy |
| Droga (b) | po teście, gdy przerwa większa niż RTO | tak | nie | tak, jeśli RTO tego wymaga |
| Droga (c2), trzeci host | po decyzji autora o sprzęcie | opcja | opcja | opcja |
| Droga (c3b), laboratorium | po (a) | nie dotyczy | nie dotyczy | tak |
| Droga (c3), magazyn S3 | do czasu przewodnika Epic | nie dotyczy | nie dotyczy | nie |

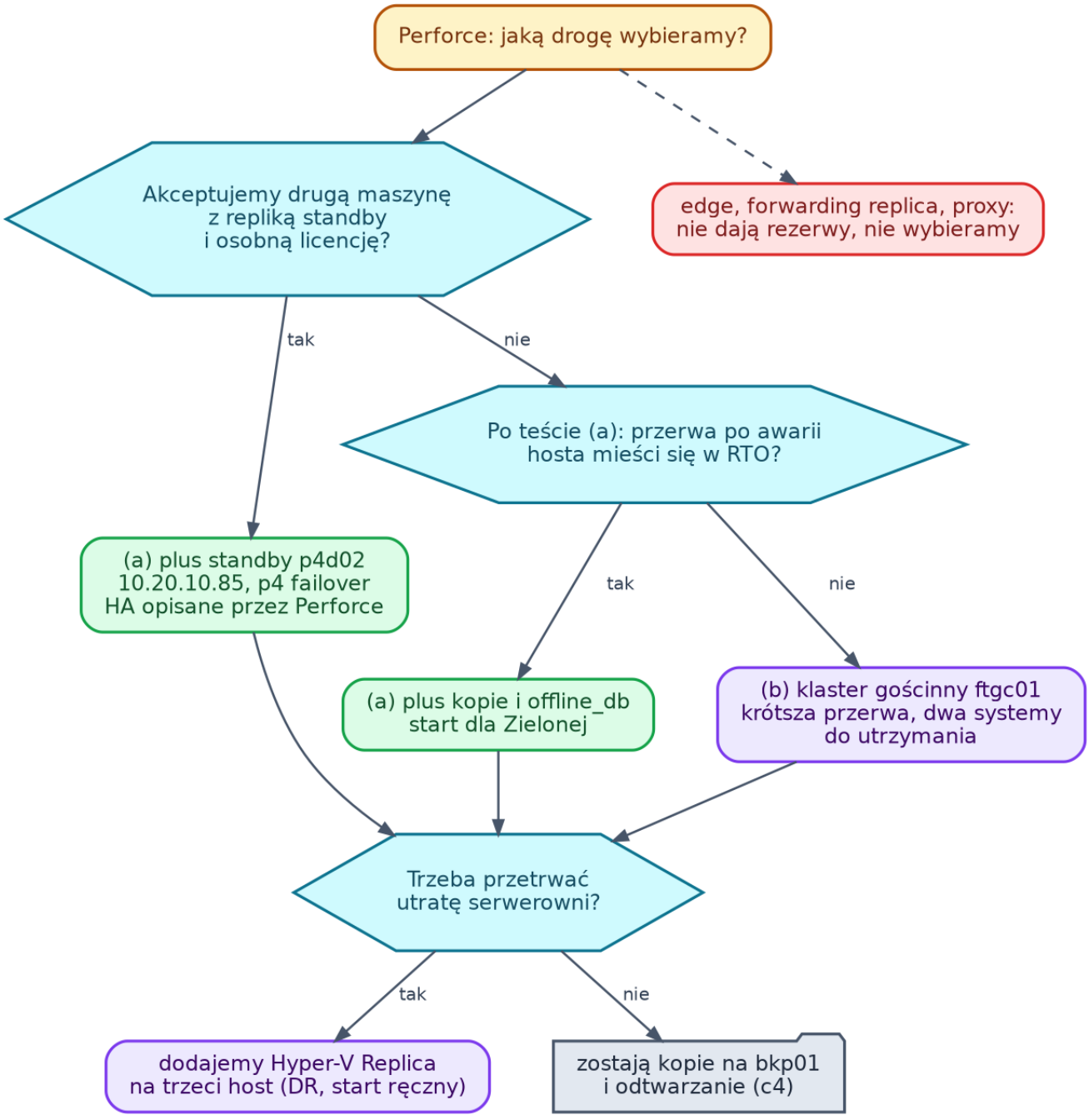
# Dodatkowe testy

Do testów z poprzedniej strony dochodzą testy dróg (c1) i (c3b). Wszystkie mierzymy tą samą pętlą portu i `p4 info` co poprzednio.

| Test | Jak wykonujemy | Co zapisujemy |
|---|---|---|
| T-c1: planowe przełączenie na standby | `p4 failover` w podglądzie, potem z `-y`, zmiana celu aliasu | czas od `-y` do działającego aliasu, kroki ręczne na Windowsie (`P4NAME`, `P4LOG`, restart usługi) |
| T-c2: awaria serwera głównego | wyłączamy `p4d01` bez udziału w przełączeniu, `p4 failover` na standby | liczba niezreplikowanych zmian, wynik `p4 verify -q //...` na nowym serwerze |
| T-c3: zerwanie `journalcopy` | odcinamy sieć standby na kilka minut, przywracamy | czy standby nadrabia sam, wynik `pull -l -j` i `pull -l -s` |
| T-c4: stan po awansie | po T-c1 i T-c2 uruchamiamy kopie, `p4 info` | czy monitoring i triggery działają na nowym serwerze |
| T-L1: replika odczytu Lore | zatrzymujemy serwer nadrzędny, próbujemy sklonować i wypchnąć | czy odczyt działa, czy zapis jest odrzucany z czytelnym błędem |

# Drzewo decyzji

Drzewo prowadzi od pytania o replikę do wyboru drogi dla Perforce. Dla Lore drzewa nie rysujemy, bo ma jedną dostępną drogę: (a), a po teście (b).




*Rysunek 2. Drzewo decyzji dla Perforce. Pytamy najpierw o dodatkową replikę standby, potem o zmierzoną przerwę drogi (a), na końcu o ochronę przed utratą serwerowni. (kliknięcie otwiera pełny rozmiar)*

# Decyzje autora

- Czy przyjmujemy kopię `offline_db` ze skryptów SDP, czy zostajemy przy własnych skryptach checkpointu.
- Czy Zielona kupuje trzeci host (Swarzędz) na kopię DR, czy do tego czasu `bkp01` jest jedynym miejscem kopii. Jeśli `bkp01` stoi w tej samej serwerowni co hosty, pożar zabiera wszystko (do potwierdzenia).
- Czy Lore w ogóle ma dostać drogę (b), czy zostajemy przy (a) do wydania 1.0.
- Czy zgadzamy się na adresy z puli: `10.20.10.85` (`p4d02`) i `10.20.10.86` (`lore02`, tylko laboratorium).

# Częste błędy

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| Po skasowaniu depotu znika on także na standby | replika powiela pomyłkę, nie jest kopią zapasową | restic i checkpointy na `bkp01` zostają przy każdej drodze |
| `p4 failover` odmawia pracy na standby | monitoring nie był włączony przy starcie standby | włączamy monitoring przy starcie serwera i sprawdzamy podglądem bez `-y` [1] |
| Dwa serwery P4 z rozbieżnymi danymi | `p4 failover -i` przy żywym serwerze głównym | opcji `-i` używamy tylko wtedy, gdy serwer główny na pewno nie wróci [2] |
| Po awansie standby na Windowsie serwer ma stare ustawienia | nie zmieniono `P4NAME` i `P4LOG` po przełączeniu | wykonujemy kroki Windows z przewodnika SDP Failover Guide [10] |
| Uznajemy Hyper-V Replica za wysoką dostępność | replika startuje ręcznie i odtwarza stan sprzed kilkunastu sekund do kwadransa | opisujemy ją jako DR i nie obiecujemy nią RTO [14] |
| Edge albo proxy dokładamy „dla pewności” | edge nie może być standby serwera commit | rezerwę robimy standby, a edge tylko dla wydajności [4] |
| Zakładamy, że Lore na Ceph RGW „po prostu działa”, skoro istnieje `s3_endpoint_url` | stan serwera trzyma też DynamoDB, a dokumentacja nie opisuje magazynów zgodnych z S3 | traktujemy to jako eksperyment w laboratorium, nie jako plan [19][27] |
| Wybieramy MinIO, bo „wszyscy tak robią” | repozytorium zarchiwizowane, dystrybucja tylko źródłowa | przy każdym wyborze magazynu sprawdzamy bieżący stan projektu [32] |
| Replika Lore nie startuje po aktualizacji | replika i serwer nadrzędny mają różne wersje protokołu | aktualizujemy je razem [26] |
| Łączymy (b) i (c1) | dwa mechanizmy przełączania jednego serwera, alias w DNS raz jako rekord A, raz jako CNAME | wybieramy jeden: (a) z (c1) albo samo (b) |

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| Test przełączenia w zakazie zmian | okno od 15 listopada do 6 stycznia | testy robimy poza zakazem i poza zamknięciem miesiąca |

# □□ Ćwiczenia

1. Dlaczego droga (b) skraca przerwę po awarii hosta, a mimo to nie leczy uszkodzenia bazy Perforce?

**Odpowiedź**

W (b) po awarii węzła rola startuje na drugim, już działającym węźle, więc odpada rozruch systemu. Dane serwera leżą jednak w jednym egzemplarzu na wspólnym dysku, więc uszkodzone pliki `db.*` zobaczy także drugi węzeł. Dopiero replika standby ma osobną kopię metadanych odtwarzaną z dziennika.

1. Czym różni się planowe przełączenie na standby od przełączenia po awarii serwera głównego?

**Odpowiedź**

Przy planowym serwer główny bierze udział w przełączeniu i według dokumentacji nie tracimy danych. Po awarii tracimy transakcje, które nie zdążyły dotrzeć na standby przez `journalcopy`. W obu przypadkach przełączenie wykonuje ręcznie administrator z uprawnieniem super, a w jego trakcie klienci nie wykonują poleceń.

1. Dlaczego Hyper-V Replica nie spełnia wymagania „wyłączenie jednego hosta ma być niewidoczne”?

**Odpowiedź**

Replika jest kopią asynchroniczną na drugim hoście i nie przejmuje pracy sama: start repliki wykonuje człowiek, a przy awarii nieplanowanej możliwa jest utrata zmian z ostatnich sekund lub minut. Niewidoczne wyłączenie planowe daje migracja na żywo w klastrze, czyli droga (a).

1. Dlaczego bezstanowe instancje Lore za HAProxy nie są dla Zielonej, mimo że dokumentacja dopuszcza wiele serwerów?

**Odpowiedź**

Wiele serwerów jest opisane tylko z trwałym magazynem w S3 i DynamoDB (implementacja referencyjna w AWS). W kodzie wtyczki można wskazać własny adres S3, ale stan trzyma też DynamoDB, a zgodnej z nią bazy sprawdzonej z Lore nie mamy. Do tego trzeba by magazynu i bazy z kilku węzłów, czego dwa hosty nie dają jako niezależne awarie.

1. Jaką wartość ma replika odczytu Lore, skoro zapis idzie przez serwer nadrzędny?

**Odpowiedź**

Odciąża serwer nadrzędny przy odczytach (klonowanie, synchronizacja) i być może pozwala odczytywać dane, gdy serwer nadrzędny jest niedostępny, ale tego dokumentacja nie obiecuje. Nie jest rezerwą zapisu, dlatego nie liczymy jej do wysokiej dostępności.

**Powiązane:** Perforce i Lore w klastrze: cztery drogi, rekomendacja dla Zielonej i test przełączenia · Klaster Windows dla własnych usług: rola, zasób, zależności, kworum i testy przełączenia · Wysoka dostępność: pojęcia, liczenie dostępności i poziomy rezerwy · Replikacja danych między hostami: DRBD, GlusterFS, Ceph, ZFS i replikacja w aplikacji · Równoważenie i adres wirtualny: keepalived i HAProxy na parze lb01 i lb02 · Macierz wysokiej dostępności usług Zielonej: poziom rezerwy każdej usługi, zależności i kolejność prac

**Źródła:** (odczyt 3 października 2026)


1. help.perforce.com: Failover
2. help.perforce.com: p4 failover
3. help.perforce.com: Standby and forwarding-standby server
4. help.perforce.com: Deployment architecture
5. help.perforce.com: Commit-edge overview
6. help.perforce.com: General guidelines for multi-server deployments (osobna licencja dla standby i replik)
7. help.perforce.com: Request a license (osobne licencje dla standby)
8. help.perforce.com: p4 pull (opcje `-l`, `-j`, `-s`, `-u`)
9. help.perforce.com: p4 servers (opcja `-J`)
10. workshop.perforce.com: SDP Failover Guide (dla Unix i Windows)
11. workshop.perforce.com: SDP Guide for Windows (`offline_db`, codzienna kopia, wymagania)

12. perforce.com: Technical guide: Azure best practices (standby w osobnej strefie, local failover, zmiana DNS)
13. Ta baza wiedzy: Perforce i Lore w klastrze: cztery drogi, rekomendacja dla Zielonej i test przełączenia, źródła 1 do 44 tamże
14. learn.microsoft.com: Hyper-V Replica overview
15. learn.microsoft.com: Measure-VMReplication
16. learn.microsoft.com: Storage Replica overview
17. learn.microsoft.com: Create Hyper-V VHD Set files (pliki VHD Set wspierają Hyper-V Replica)
18. learn.microsoft.com: Errors when you back up VMs that belong to a guest cluster
19. epicgames.github.io: Lore Server configuration reference
20. epicgames.github.io: FAQ (stabilność 0.x, skalowanie poziome, dwupoziomowa hierarchia, platformy)
21. epicgames.github.io: Deploy a local Lore Server
22. epicgames.github.io: The Lore Version Control System, system design
23. epicgames.github.io: ADR-00004: S3 storage options
24. epicgames.github.io: ADR-00007: AWS mutable store options
25. epicgames.github.io: ADR-00013: Plugin dependency strategy
26. github.com: Lore, wydania (v0.9.0: zmiana protokołu replikacji i tabel DynamoDB, v0.10.0: obrazy kontenerów, wymagane `jwt_issuer` i `jwt_audience`) oraz lore-server/DOCKER.md i lore-server/Dockerfile
27. github.com: kod Lore w gałęzi main: lore-aws/src/clients.rs, lore-aws/src/dynamodb.rs, lore-server/src/plugins/aws.rs i lore-server/src/plugins/mod.rs (pola konfiguracji wtyczki, rejestracja wtyczek; odczyt kodu, nie dokumentacji)
28. anchorpoint.app: How to Set Up Lore on AWS Using S3 as Storage
29. dev.classmethod.jp: I tried deploying Epic Games' Lore server to AWS Fargate + S3 + DynamoDB with Terraform
30. dev.classmethod.jp: I measured the VPC endpoints required to run Epic Games' Lore on a private subnet without NAT
31. docs.scylladb.com: Alternator i Alternator compatibility
32. github.com: minio/minio (repozytorium zarchiwizowane 25 kwietnia 2026, wersja społecznościowa tylko jako kod źródłowy)

---

Revision #13
Created 2026-10-03 17:41:06 UTC by Tomasz Konieczny
Updated 2026-10-26 14:22:04 UTC by Tomasz Konieczny