Kopia fs01: restic z migawką VSS do repozytorium pliki i test odtworzenia pliku oraz całego udziału

Poziom: średnio

Dotyczy: Windows Server 2022, fs01 (10.20.10.15), restic 0.19.1, rest-server 0.14.0 na bkp01 (10.20.10.61), bkp02 (10.20.30.61), hvtest01

Czas: ~90 min

Ta strona zamyka lukę z Strategii kopii zapasowych Zielonej: co noc restic robi na `fs01` migawkę VSS, kopiuje z niej udziały do repozytorium `pliki` na `bkp01`, a dalej działa już to, co opisują strony o serwerach kopii (kopia krzyżowa, archiwum, retencja 365 dni). Opisuje też dwa testy: odtworzenie jednego pliku i całego udziału, z pomiarem czasu. Wracamy tu przy alarmie o wieku kopii, przed testem kwartalnym i po ataku na udziały.

ZAŁOŻENIA SCENARIUSZA

serwery kopii działają tak, jak na stronie Serwery kopii bkp01 i bkp02: restic, kopia krzyżowa i niezmienność: `bkp01` z rest-server w trybie tylko do dopisywania, `bkp02` w Swarzędzu za tunelem IPsec, który sam pobiera kopie z `bkp01`. Test całego udziału robimy na samodzielnym hoście `hvtest01` poza klastrem, przyjętym na stronie Testy odtwarzania: plik, maszyna, baza i cała lokalizacja; host jest w domenie i widzi `bkp01`. Gdy `bkp02` jeszcze nie działa, kopia `fs01` ma tylko dwie warstwy (`bkp01` i archiwum).

Skróty użyte na tej stronie (18)

Skrót	Co znaczy
AD	Active Directory: katalog Microsoftu z kontami użytkowników, komputerów i grup w firmie

Skrót	Co znaczy
DACL	Discretionary Access Control List: zwykła lista uprawnień pliku albo folderu, mówi, kto może co zrobić
DFS	Distributed File System: usługa łącząca udziały z wielu serwerów pod jedną wspólną ścieżką sieciową
DFSN	nazwa modułu PowerShella do przestrzeni nazw DFS (polecenia New-DfsnRoot, New-DfsnFolder i podobne)
FSRM	File Server Resource Manager: rola Windows Server do pilnowania serwera plików, ustawia limity miejsca, filtry plików, raporty i akcje po zdarzeniach
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01
gMSA	group Managed Service Account: konto usługi, któremu Windows sam zmienia hasło
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
IPsec	Internet Protocol Security: zestaw protokołów szyfrujących i uwierzytelniających ruch sieciowy, na nim stoi tunel VPN typu IKEv2
NTFS	New Technology File System: system plików Windows z wbudowanymi uprawnieniami ACL na poziomie pliku i folderu
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
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
SID	w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta
VHDX	Virtual Hard Disk v2: format pliku dysku wirtualnego maszyny Hyper-V, następca starszego VHD, obsługuje większe rozmiary i jest odporniejszy na uszkodzenie przy nagłym zaniku zasilania
VSS	Volume Shadow Copy Service: mechanizm Windows do tworzenia spójnych migawek danych mimo działających w tle usług, wykorzystywany też przy kopiach zapasowych kontrolera domeny

Fiszka

Pierwsze trzy fiszki wykonujemy w bashu na `bkp01` jako użytkownik z sudo, pozostałe w PowerShellu jako administrator na `fs01` albo `hvtest01`, zgodnie z opisem. Placeholdery: `NazwaRepozytorium` (u nas `pliki`), `NazwaUdzialu`, `SciezkaDocelowa`.

ZMIENIA SYSTEM

Utworzenie konta REST dla fs01: htpasswd

```
sudo htpasswd -B /etc/rest-server/htpasswd NazwaRepozytorium
```

Konto `pliki` zobaczy na `bkp01` tylko katalog `/backup/rest/pliki`, bo rest-server działa z opcją `--private-repos`. Plik `htpasswd` już istnieje, więc nie dajemy `-c` (to by go nadpisało). Hasło konta REST zapisujemy w Vaultwardenie; to inny sekret niż hasło repozytorium.

REST = sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu · **htpasswd** = Linux: tworzy i zmienia plik z kontami i skrótami haseł (tu dla serwera REST restic)

ZMIENIA SYSTEM

Plik z hasłem repozytorium na serwerze kopii

```
sudo install -m 600 -o restic -g restic /dev/null /etc/restic/haslo-NazwaRepozytorium
sudo -u restic nano /etc/restic/haslo-NazwaRepozytorium
```

Retencja i kontrola na `bkp01` (skrypt `restic-retencja.sh`) czytają hasło z `/etc/restic/haslo-pliki`, a kopia krzyżowa i archiwum z tego samego pliku. Wpisujemy tam to samo hasło repozytorium, którego użyje `fs01`. Repozytorium `pliki` jest już w `poziomy.sh` z poziomem 365 dni, więc niczego więcej w skryptach serwera kopii nie zmieniamy.

install = install: kopiuje pliki i tworzy katalogi z jednoczesnym ustawieniem uprawnień

ZMIENIA SYSTEM

Dopisanie repozytorium do stanu dla Zabbixa

```
echo 'pliki|/backup/rest/pliki|/etc/restic/haslo-pliki' | sudo tee -a
/etc/restic/repozytoria.lista
```

Od następnej godziny `restic-status.sh` zapisuje wiek ostatniej migawki `pliki`, a Zabbix podnosi alarm powyżej 26 godzin, tak jak dla innych repozytoriów. Element `custom.restic.wiek[pliki]` dokładamy na hoście `bkp01` w Zabbixie.

echo = od angielskiego echo: wypisuje podany tekst na standardowe wyjście, często łączone z przekierowaniem do pliku

□□ ZMIENIA SYSTEM

Pobranie restic na fs01: Invoke-WebRequest

```
$wersja = "0.19.1"
Invoke-WebRequest -Uri
"https://github.com/restic/restic/releases/download/v$wersja/restic_$($wersja)_windows_a
md64.zip" -OutFile "$env:TEMP\restic.zip"
Get-FileHash "$env:TEMP\restic.zip" -Algorithm SHA256
Expand-Archive -Path "$env:TEMP\restic.zip" -DestinationPath "$env:TEMP\restic"
New-Item -Path "C:\Program Files\restic" -ItemType Directory -Force | Out-Null
Copy-Item "$env:TEMP\restic\restic_$($wersja)_windows_amd64.exe" "C:\Program
Files\restic\restic.exe"
```

Komendy można wykonać jako blok. Wersja 0.19.1 jest ta sama co na serwerach kopii i na `docker01`. Sumę z `Get-FileHash` porównujemy z plikiem `SHA256SUMS` na stronie wydania, a dokładną nazwę archiwum i pliku w środku sprawdzamy na tej stronie przed pobraniem (do potwierdzenia).

□□ **SHA256** = funkcja skrótu kryptograficznego dająca 256-bitowy odcisk danych

□□ ZMIENIA SYSTEM

Katalog na sekrety i skrypt tylko dla SYSTEM i administratorów: icacls

```
New-Item -Path "C:\ProgramData\restic\logi" -ItemType Directory -Force | Out-Null
icacls "C:\ProgramData\restic" /inheritance:r /grant:r "*S-1-5-18:(OI)(CI)F" "*S-1-5-32-
544:(OI)(CI)F"
```

> (gdzie `*S-1-5-18` to konto SYSTEM, a `*S-1-5-32-544` grupa Administratorzy; zapis przez numer SID działa tak samo w polskim i angielskim Windowsie)

W tym katalogu leżą hasło repozytorium, hasło konta REST, plik wykluczeń, skrypt i logi. Zwykły użytkownik nie może ich odczytać ani podmienić skryptu, który działa jako SYSTEM.

□□ **SID** = w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta · **REST** = sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu · **icacls** = PowerShell/CMD: odczytaj albo nadaj uprawnienia NTFS na pliku lub folderze z poziomu

konsoli

PLIK KONFIGURACYJNY

Plik wykluczeń: wykluczenia.txt

```
# Pliki, ktorych nie kopiujemy z D:\Dane (wzorzec bez ukosnika pasuje do nazwy w kazdym folderze)
~$*
Thumbs.db
*.tmp
$RECYCLE.BIN
System Volume Information
```

Plik zapisujemy jako `C:\ProgramData\restic\wykluczenia.txt`. Pliki `~$*` to blokady Office, które istnieją tylko przy otwartym dokumencie. Folderów `0-pulapka` nie wykluczamy: wabiki są małe, a ich kopia pozwala porównać je po incydencie.

ZMIENIA SYSTEM

Inicjalizacja repozytorium z fs01: restic init

```
$env:RESTIC_REPOSITORY = "rest:https://bkp01.ad.zielona.example:8000/NazwaRepozytorium"
$env:RESTIC_REST_USERNAME = "NazwaRepozytorium"
$env:RESTIC_REST_PASSWORD = (Get-Content "C:\ProgramData\restic\rest-haslo.txt" -Raw).Trim()
$env:RESTIC_PASSWORD_FILE = "C:\ProgramData\restic\haslo-repozytorium.txt"
& "C:\Program Files\restic\restic.exe" init
```

Wykonujemy raz, jako administrator na `fs01`. Wcześniej zapisujemy dwa pliki z hasłami w `C:\ProgramData\restic` (z Vaultwardena). Certyfikat `bkp01` jest wystawiony przez wewnętrzny urząd, któremu ufają komputery domeny, więc opcja `--cacert` nie jest potrzebna (sprawdzamy pierwszym `init`).

PLIK KONFIGURACYJNY

Skrypt nocnej kopii: kopia-fs01.ps1

```
# Nocna kopia udzialow fs01 do repozytorium pliki na bkp01, z migawka VSS.
# Uruchamia zadanie zaplanowane jako SYSTEM. Kod wyjscia: 0 dobrze, 3 czesc plikow
nieodczytana, inne blad.
# Bez "Stop": w Windows PowerShell 5.1 komunikaty restic na stderr przy przekierowaniu
bylyby traktowane jak bledy.
$ErrorActionPreference = "Continue"
$Katalog = "C:\ProgramData\restic"
$Log = Join-Path $Katalog ("logi\kopia-" + (Get-Date -Format "yyyy-MM-dd") + ".log")
$Zrodlo = "restic-fs01"

$env:RESTIC_REPOSITORY = "rest:https://bkp01.ad.zielona.example:8000/pliki"
$env:RESTIC_REST_USERNAME = "pliki"
$env:RESTIC_REST_PASSWORD = (Get-Content (Join-Path $Katalog "rest-haslo.txt") -
Raw).Trim()
$env:RESTIC_PASSWORD_FILE = Join-Path $Katalog "haslo-repozytorium.txt"
$env:RESTIC_CACHE_DIR = Join-Path $Katalog "cache"

if (-not [System.Diagnostics.EventLog]::SourceExists($Zrodlo)) {
  New-EventLog -LogName Application -Source $Zrodlo
}

& "C:\Program Files\restic\restic.exe" backup "D:\Dane" `
  --use-fs-snapshot -o vss.timeout=5m `
  --host fs01 --tag pliki `
  --exclude-file (Join-Path $Katalog "wykluczenia.txt") *>> $Log
$Kod = $LASTEXITCODE

switch ($Kod) {
  0 { Write-EventLog -LogName Application -Source $Zrodlo -EventId 1000 -EntryType
Information -Message "Kopia fs01 zakonczona poprawnie." }
  3 { Write-EventLog -LogName Application -Source $Zrodlo -EventId 1003 -EntryType
Warning -Message "Kopia fs01 niepelna: czesc plikow nieodczytana, szczegoly w $Log" }
  default { Write-EventLog -LogName Application -Source $Zrodlo -EventId 1001 -EntryType
Error -Message "Kopia fs01 nieudana, kod $Kod, szczegoly w $Log" }
}

Get-ChildItem (Join-Path $Katalog "logi") -Filter "kopia-*.log" | Where-Object
LastWriteTime -lt (Get-Date).AddDays(-30) | Remove-Item
exit $Kod
```

Plik zapisujemy jako `C:\ProgramData\restic\kopia-fs01.ps1`. Opcja `--use-fs-snapshot` robi na czas kopii migawkę VSS i czyta pliki z niej, a nie z żywego wolumenu [1], więc otwarte w nocy pliki Excela trafiają do kopii w spójnym stanie. `-o vss.timeout=5m` wydłuża czas na utworzenie migawki z domyślnych 120 sekund [1]. Kod 3 oznacza, że części danych nie udało się odczytać, a każdy kod nieznany traktujemy jak błąd [2]. Skrypt nie wywołuje `forget` ani `prune`: serwer kopii odmówiłby kasowania, a retencję robi `bkp01`.

VSS = Volume Shadow Copy Service: mechanizm Windows do tworzenia spójnych migawek danych mimo działających w tle usług, wykorzystywany też przy kopiach zapasowych kontrolera domeny · **exit** = zakończ bieżącą powłokę albo skrypt z podanym kodem wyjścia

ZMIENIA SYSTEM

Zadanie zaplanowane o 3:15: Register-ScheduledTask

```
$akcja = New-ScheduledTaskAction -Execute "powershell.exe" -Argument '-NoProfile -File "C:\ProgramData\restic\kopia-fs01.ps1"'
$wyzwalacz = New-ScheduledTaskTrigger -Daily -At 03:15
$ustawienia = New-ScheduledTaskSettingsSet -ExecutionTimeLimit (New-TimeSpan -Hours 3) -StartWhenAvailable
Register-ScheduledTask -TaskName "Kopia fs01 restic" -Action $akcja -Trigger $wyzwalacz -Settings $ustawienia -User "NT AUTHORITY\SYSTEM" -RunLevel Highest
```

Godzina 3:15 pochodzi z harmonogramu nocy na stronie o serwerach kopii (po bazach i GLPI, przed kopią krzyżową o 4:00). Konto SYSTEM ma uprawnienia do migawek VSS i do odczytu pełnych deskryptorów zabezpieczeń plików, czego restic potrzebuje, żeby zapisać w kopii uprawnienia NTFS [1]. Limit trzech godzin chroni okno kopii krzyżowej.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **VSS** = Volume Shadow Copy Service: mechanizm Windows do tworzenia spójnych migawek danych mimo działających w tle usług, wykorzystywany też przy kopiach zapasowych kontrolera domeny · **NTFS** = New Technology File System: system plików Windows z wbudowanymi uprawnieniami ACL na poziomie pliku i folderu · **NT** = pierwotnie kryptonim N-Ten, od procesora, dla którego system miał powstać; popularnie dziś odczytywane jako New Technology, choć to rozwinięcie nie jest oficjalne: rdzeń dzisiejszego Windows Server

ZMIENIA SYSTEM

Ręczne uruchomienie i wynik ostatniego przebiegu: Start-ScheduledTask

```
Start-ScheduledTask -TaskName "Kopia fs01 restic"
Get-ScheduledTaskInfo -TaskName "Kopia fs01 restic" | Format-List LastRunTime,
LastTaskResult, NextRunTime
```

Pierwszy przebieg przenosi całe `D:\Dane`, więc robimy go w piątek wieczorem po 18:00, gdy biuro już nie pracuje, a nie w nocy przed kopią krzyżową. `LastTaskResult` równy 0 to kopia poprawna, 3 to kopia niepełna.

Przykład:

```
LastRunTime    : 26.10.2026 03:15:00
LastTaskResult : 0
NextRunTime    : 27.10.2026 03:15:00
```

TYLKO ODCZYT

Lista migawek fs01 w repozytorium: restic snapshots

```
sudo -u restic restic -r /backup/rest/NazwaRepozytorium --password-file
/etc/restic/haslo-NazwaRepozytorium snapshots --host fs01
```

Wykonujemy na `bkp01`, lokalnie na katalogu repozytorium. Każda nocna kopia to jedna migawka z datą; z niej odczytujemy RPO.

Przykład:

```
ID        Time                 Host   Tags   Paths
----------------------------------------------------
3f9a1c22  2026-10-25 03:15:07  fs01   pliki  D:\Dane
b71e0d94  2026-10-26 03:15:05  fs01   pliki  D:\Dane
----------------------------------------------------
2 snapshots
```

RPO = Recovery Point Objective: ile danych wolno stracić przy awarii

TYLKO ODCZYT

Odczyt ścieżek w migawce: restic ls

```
sudo -u restic restic -r /backup/rest/NazwaRepozytorium --password-file
/etc/restic/haslo-NazwaRepozytorium ls latest --host fs01 | head -20
```

Restic zapisuje ścieżki z Windowsa w postaci z ukośnikami i literą dysku jako pierwszym katalogiem. Postać, której używamy w `--include`, przepisujemy z tego wyniku, a nie zgadujemy.

Przykład:

```
/D
/D/Dane
/D/Dane/Handel
/D/Dane/Ksiegowosc
/D/Dane/Ksiegowosc/Faktury
```

□□ ZMIENIA SYSTEM

Test pliku z pomiarem czasu: test-plik.sh

```
sudo -u restic /usr/local/sbin/test-plik.sh /backup/rest/NazwaRepozytorium
/etc/restic/haslo-NazwaRepozytorium /D/Dane/NazwaUdzialu/NazwaFolderu
```

Skrypt ze strony o testach odtwarzania odtwarza wskazany folder do katalogu tymczasowego na `bkp01`, mierzy czas i dopisuje wynik do `testy-odtworzen.csv`. Na Linuksie restic odtwarza same pliki, bez uprawnień NTFS, więc ten test sprawdza dane, hasło i czas, a nie uprawnienia.

Przykład:

```
zaliczony: 412 plikow, 1873420288 bajtow, 61 s
```

□□ **NTFS** = New Technology File System: system plików Windows z wbudowanymi uprawnieniami ACL na poziomie pliku i folderu

□□ ZMIENIA SYSTEM

Odtworzenie pliku albo folderu na fs01 obok oryginału: restic restore

```
& "C:\Program Files\restic\restic.exe" restore latest --host fs01 --target
"SciezkaDocelowa" --include "/D/Dane/NazwaUdzialu/NazwaFolderu"
```

Wykonujemy na `fs01` jako administrator, ze zmiennymi środowiskowymi jak przy `init`. Restic odtwarza pełną ścieżkę pod katalogiem docelowym, czyli z `--target "D:\Odtworzenie"` plik trafia do `D:\Odtworzenie\D\Dane\...` [3]. Jako administrator restic odtwarza też uprawnienia NTFS (deskryptory zabezpieczeń), a bez uprawnień `SeRestorePrivilege`, `SeSecurityPrivilege` i `SeTakeOwnershipPrivilege` tylko listę DACL [3]. Zamiast `latest` możemy podać numer migawki z `restic snapshots`.

NTFS = New Technology File System: system plików Windows z wbudowanymi uprawnieniami ACL na poziomie pliku i folderu · **DACL** = Discretionary Access Control List: zwykła lista uprawnień pliku albo folderu, mówi, kto może co zrobić

ZMIENIA SYSTEM

Podgląd odtworzenia bez zapisu: restic restore --dry-run

```
& "C:\Program Files\restic\restic.exe" restore latest --host fs01 --target
"SciezkaDocelowa" --include "/D/Dane/NazwaUdzialu" --dry-run -vv
```

Wypisuje, co zostałoby odtworzone, bez zapisu na dysk [3]. Przed odtworzeniem całego udziału sprawdzamy nim, czy `--include` trafia w dobry folder i ile danych przyjdzie.

ZMIENIA SYSTEM

Test całego udziału na hvtest01 z pomiarem czasu

```
$t0 = Get-Date
& "C:\Program Files\restic\restic.exe" restore latest --host fs01 --target
"D:\Test\Odtworzenie" --include "/D/Dane/NazwaUdzialu"
$czas = New-TimeSpan -Start $t0 -End (Get-Date)
$pliki = Get-ChildItem "D:\Test\Odtworzenie\D\Dane\NazwaUdzialu" -Recurse -File
"{0:hh\:mm\:ss}  plikow: {1}  GB: {2:N1}" -f $czas, $pliki.Count, (($pliki | Measure-
Object Length -Sum).Sum / 1GB)
```

Wykonujemy na `hvtest01` jako administrator domenowy, z restic zainstalowanym jak na `fs01` i z hasłami pobranymi z Vaultwardena na czas testu (po teście pliki z hasłami usuwamy). Liczbę plików i rozmiar porównujemy z tym samym poleceniem `Get-ChildItem` uruchomionym na `fs01`, a uprawnienia kilku folderów z `icacls`. Wynik i czas wpisujemy do protokołu ze strony o testach.

Przykład:

```
01:12:40  plikow: 48211  GB: 231,4
```

GB = Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej

TYLKO ODCZYT

Porównanie sum kontrolnych wybranych plików: Get-FileHash

```
Get-FileHash "SciezkaPlikuOdtworzonego", "SciezkaPlikuNaFs01" -Algorithm SHA256 |
Format-Table Hash, Path
```

Bierzemy pięć plików z różnych folderów, niezmienianych od nocy (data modyfikacji przed godziną kopii). Takie same sumy znaczą, że kopia zawiera dokładnie te bajty.

SHA256 = funkcja skrótu kryptograficznego dająca 256-bitowy odcisk danych

ZMIENIA SYSTEM

Podmiana celu folderu w przestrzeni nazw po odtworzeniu: New-DfsnFolderTarget

```
New-DfsnFolderTarget -Path "\\ad.zielona.example\pliki\NazwaUdzialu" -TargetPath
"\\fs01\NazwaUdzialuOdtw"
Remove-DfsnFolderTarget -Path "\\ad.zielona.example\pliki\NazwaUdzialu" -TargetPath
"\\fs01\NazwaUdzialu" -Force
```

Używamy tylko przy prawdziwym odtworzeniu całego udziału na `fs01`, opisanym w sekcji „Odtworzenie udziału po ataku”. Użytkownicy dalej otwierają tę samą ścieżkę, a trafiają do odtworzonego folderu po wygaśnięciu skierowania w pamięci stacji.

Z życia wzięte

Z ŻYCIA WZIĘTE

Wtorek, 2 lutego 2027, 17:05, pierwszy kwartalny test `fs01`. Tomek odtwarza udział `Ksiegowosc` na `hvtest01` i zatrzymuje stoper po 1 godzinie 13 minutach dla 231 GB. Ola, która mierzy czas, liczy na kartce: przy tej prędkości całe 1,5 TB danych wróciłoby po prawie ośmiu

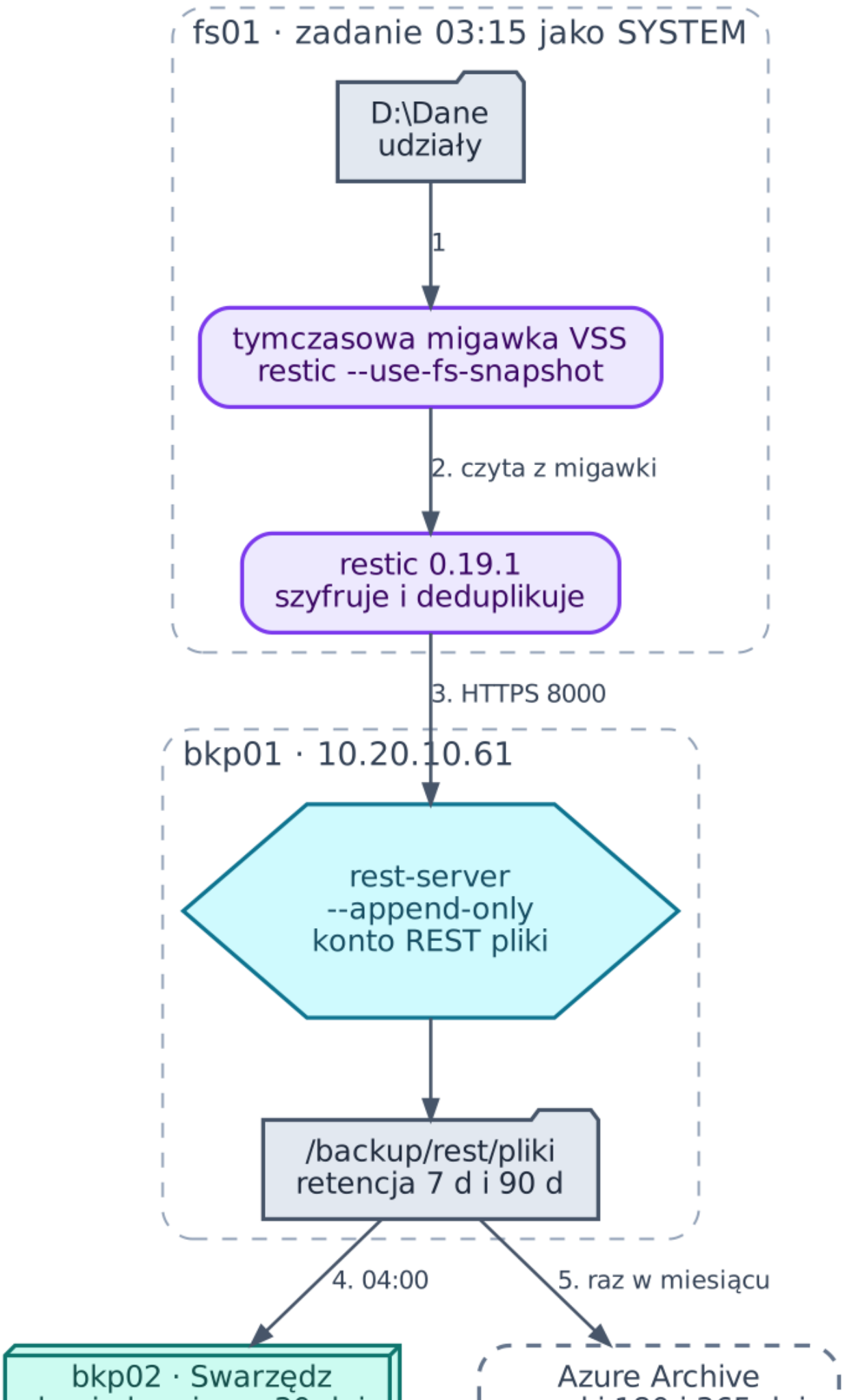
> godzinach, a macierz obiecuje cztery. Do protokołu idą dwa wnioski: kolejność odtwarzania udziałów (księgowość i wysyłki pierwsze) i rozmowa z panią Haliną o RTO dla reszty.

Jak to działa

Rozwiń: jak to działa pod spodem

Droga kopii

Kopia `fs01` składa się z dwóch kroków. Na `fs01` restic robi migawkę, czyta z niej pliki, szyfruje je i wysyła do `bkp01`; wszystko dalej dzieje się już bez udziału `fs01`.


fs01 · zadanie 03:15 jako SYSTEM
D:\Dane
udziały
1
tymczasowa migawka VSS
restic --use-fs-snapshot
2. czyta z migawki
restic 0.19.1
szyfruje i deduplikuje
3. HTTPS 8000
bkp01 · 10.20.10.61
rest-server
--append-only
konto REST pliki
/backup/rest/pliki
retencja 7 d i 90 d
4. 04:00
5. raz w miesiącu
bkp02 · Swarzędz
Azure Archive

Rysunek 1. Droga kopii fs01: zadanie o 3:15 jako SYSTEM robi tymczasową migawkę VSS, restic czyta z niej D:\Dane i wysyła przez HTTPS na rest-server bkp01 do repozytorium pliki, skąd o 4:00 kopię pobiera bkp02, a raz w miesiącu paczka idzie do Azure Archive (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: restic.** Program do kopii zapasowych, który dzieli pliki na kawałki, każdy kawałek zapisuje raz (deduplikacja) i szyfruje przed wysłaniem. Druga i każda następna kopia przenosi tylko zmienione kawałki, więc nocna kopia 2 TB danych trwa zwykle minuty, a nie godziny. Ten sam program działa na serwerach kopii, na `docker01` i tu, więc procedury są wspólne.

> **Pojęcie: migawka VSS na czas kopii.** Z opcją `--use-fs-snapshot` restic prosi usługę VSS o migawkę wolumenu i czyta pliki z niej [1]. Plik otwarty przez panią Halinę o 3:15 (bo zostawiła Excela na noc) trafia do kopii w stanie z chwili migawki, a nie w połowie zapisu. Migawka znika po kopii; z poprzednimi wersjami dla użytkowników ze strony Migawki VSS i poprzednie wersje na fs01: harmonogram, miejsce na różnice i dlaczego to nie jest kopia dzieli tylko miejsce na różnice.

Parametr	Wartość	Skąd
Repozytorium	`pliki` na `bkp01`, przez rest-server, konto REST `pliki`	Serwery kopii bkp01 i bkp02
Poziom retencji	365 dni: dzienne 7 dni, tygodniowe 90 dni, miesięczne 180 dni i kwartalne 365 dni w archiwum	Strategia kopii
Godzina	3:15, co noc	harmonogram nocy na stronie o serwerach kopii
RPO i RTO	24 h i 4 h (propozycja z macierzy wysokiej dostępności)	Macierz wysokiej dostępności
Kopia krzyżowa	`bkp02` pobiera najnowszą migawkę o 4:00, trzyma 30 dni	strona o serwerach kopii
Archiwum	paczka miesięczna i kwartalna do Azure Archive, z niezmiennością	Zimne archiwum w chmurze: Azure Blob Storage i Amazon S3 Glacier

Dlaczego pliki, a nie tylko kopia maszyny

Maszyna `fs01` jest też w repozytorium `maszyny` (eksport Hyper-V, retencja 30 dni). Eksport całej maszyny z dyskiem danych 2 TB co noc to dużo danych dla `bkp01`, a odtworzenie jednego pliku

z eksportu wymaga zaimportowania albo zamontowania całego dysku. Kopia plików restic daje pojedynczy plik w minutę i rok historii; eksport maszyny daje szybki powrót systemu, ról i ustawień FSRM. Częstotliwość eksportu `fs01` (co noc czy raz w tygodniu) to decyzja do podjęcia przy przeglądzie pojemności `bkp01`.

Gdyby `fs01` dostał kiedyś partnera z replikacją DFS, dokumentacja Microsoftu dla maszyn w Azure każe kopiować dane replikowane programem wewnątrz maszyny i nie odtwarzać takiego serwera z migawki maszyny [4]. Kopia restic wewnątrz `fs01` spełnia ten warunek już dziś.

Uprawnienia NTFS w kopii

Restic na Windowsie zapisuje deskryptory zabezpieczeń plików: pełne, gdy działa z uprawnieniem `SeBackupPrivilege` albo jako administrator, a bez tego tylko właściciela, grupę i listę DACL [1]. Dlatego zadanie działa jako SYSTEM. Przy odtwarzaniu na Windowsie jako administrator uprawnienia wracają razem z plikami [3]; na Linuksie (`bkp01`) wracają same pliki. Uprawnienia odnoszą się do numerów SID kont i grup domeny, więc na hoście spoza domeny zobaczymy w nich numery zamiast nazw.

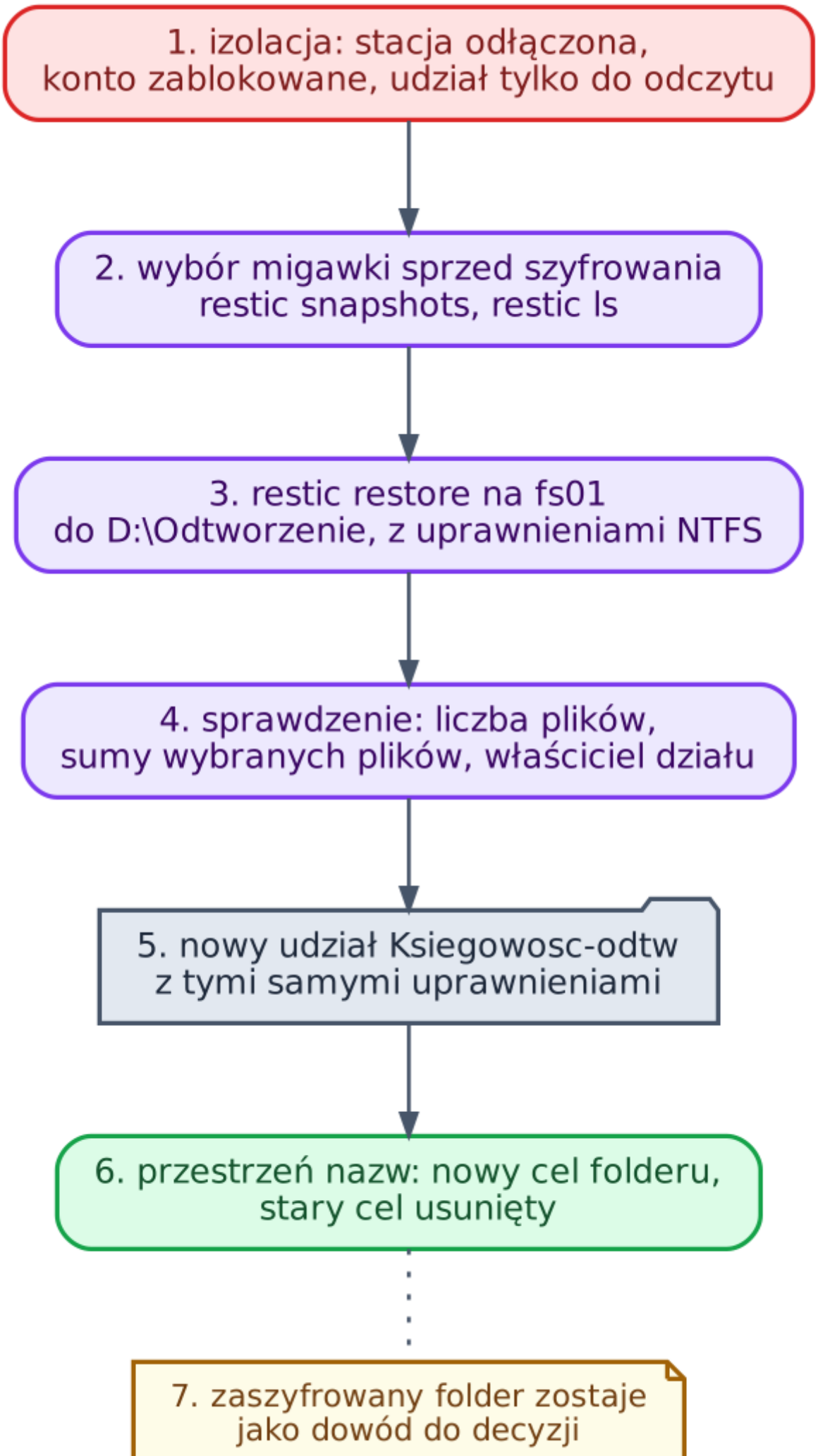
Test pliku i test udziału

Test	Gdzie	Co sprawdza	Kiedy
Plik (folder)	`bkp01` albo `bkp02`, skrypt `test-plik.sh`	hasło, spójność danych, czas na gigabajt	co miesiąc według kalendarza testów plików (pierwszy czwartek), na zmianę z `bkp01` i `bkp02`
Cały udział	`hvtest01`, restic dla Windowsa	czas odtworzenia, liczba plików, uprawnienia NTFS, sumy wybranych plików	co kwartał we wtorkowych terminach testów kwartalnych (2.02, 4.05, 3.08 i 5.10.2027), w lutym obok testu maszyny `fs01`
Z archiwum	`bkp01` po rehydratacji	paczka kwartalna `arch-RRRR-365/pliki`	raz w roku, według strony o testach

Test udziału nie dotyka produkcji: odtwarzamy na `hvtest01`, a `fs01` tylko czyta przy porównaniu. Mimo to jest zmianą normalną we wtorkowym oknie 17:00 do 20:00, bo obciąża `bkp01` i sieć, tak jak testy maszyn na stronie o testach. Wybór udziału rotujemy: luty `Ksiegowosc`, maj `Wysylki`, sierpień `Handel`, październik `Magazyn`.

Odtworzenie udziału po ataku

Prawdziwe odtworzenie po zaszyfrowaniu udziału to nie test, tylko incydent. Kolejność kroków jest ważniejsza niż szybkość.


1. izolacja: stacja odłączona,
konto zablokowane, udział tylko do odczytu
2. wybór migawki sprzed szyfrowania
restic snapshots, restic ls
3. restic restore na fs01
do D:\Odtworzenie, z uprawnieniami NTFS
4. sprawdzenie: liczba plików,
sumy wybranych plików, właściciel działu
5. nowy udział Ksiegowosc-odtw
z tymi samymi uprawnieniami
6. przestrzeń nazw: nowy cel folderu,
stary cel usunięty
7. zaszyfrowany folder zostaje
jako dowód do decyzji

Rysunek 2. Odtworzenie udziału po ataku: izolacja, wybór migawki sprzed szyfrowania, restic restore na fs01 do D:\Odtworzenie z uprawnieniami, sprawdzenie z właścicielem działu, nowy udział z tymi samymi uprawnieniami, podmiana celu w przestrzeni nazw, a zaszyfrowany folder zostaje jako dowód (kliknięcie otwiera pełny rozmiar)

1. Izolacja: zainfekowana stacja odłączona, konto zablokowane (pułapka FSRM zrobiła to na udziałach, my dokładamy blokadę w AD), udział tymczasowo tylko do odczytu.
2. Wybór migawki: `restic snapshots` pokazuje noce, a `restic ls` z numerem migawki pozwala sprawdzić, czy pliki mają jeszcze stare nazwy. Bierzemy ostatnią migawkę sprzed pierwszego zaszyfrowanego pliku (czas z alarmu pułapki i z Grayloga).
3. Odtworzenie na `fs01` do `D:\Odtworzenie`, jeżeli na `D:` jest miejsce; jeżeli nie, na nowy dysk VHDX po sprawdzeniu puli S2D.
4. Sprawdzenie: liczba plików, sumy kilku plików, a potem właściciel działu otwiera trzy dokumenty, które sam wskaże.
5. Nowy udział `NazwaUdzialuOdtw` na odtworzonym folderze, z tymi samymi uprawnieniami udziału (`Get-SmbShareAccess` na starym, `New-SmbShare` na nowym), bo uprawnienia NTFS wróciły z kopii.
6. Podmiana celu folderu w przestrzeni nazw (fiszka wyżej): ludzie wracają do pracy bez zmiany ścieżek.
7. Zaszyfrowany folder zostaje nietknięty do decyzji o zabezpieczeniu dowodów; usuwamy go dopiero po zamknięciu incydentu.

Zmiany między ostatnią kopią a atakiem (do 24 godzin) próbujemy jeszcze odzyskać z poprzednich wersji, o ile napastnik nie skasował migawek. Przy zaszyfrowaniu całego `fs01` kolejność odtwarzania udziałów ustalamy z właścicielami zawczasu, bo przy prędkości z testu 2 TB nie wróci w cztery godziny (ćwiczenie 1 na stronie o testach).

Odpowiednik bez Microsoftu

Element	Bez Microsoftu	Uwagi
Migawka na czas kopii	migawka ZFS zbioru danych działu, restic czyta z katalogu `.zfs/snapshot/NazwaMigawki`	spójność taka jak przy VSS, migawkę usuwa skrypt po kopii
Kopia na serwer kopii	restic z Rocky Linux na ten sam rest-server, konto `pliki`	skrypt i timer systemd jak na stronie o kopii `docker01`
Uprawnienia w kopii	restic zapisuje właściciela, grupę i tryb; listy ACL i rozszerzone atrybuty (do sprawdzenia w dokumentacji naszej wersji)	test odtworzenia sprawdza, czy uprawnienia wracają
Replikacja całego zbioru	`zfs send` i `zfs receive` na drugi serwer z ZFS	wymaga ZFS po obu stronach; `bkp01` ma mdadm, więc to wariant dla nowego sprzętu

Element	Bez Microsoftu	Uwagi
Nextcloud	kopia katalogu danych i zrzut bazy w trybie konserwacji	według konwencji Dockera, jak inne aplikacje na `docker01`

Częste błędy

Objaw	Przyczyna	Co robimy
Kod 3 co noc	restic nie mógł odczytać części plików, zwykle bez migawki albo przez brak uprawnień	sprawdzamy w logu, czy migawka powstała, i czy zadanie działa jako SYSTEM
`VSS error` albo przekroczony czas migawki	wolny dysk, trwająca inna migawka, pełne miejsce na różnice	`-o vss.timeout`, `vssadmin list shadowstorage`, przesunięcie harmonogramu migawek dla użytkowników
`unable to open repository`	ścieżka repozytorium inna niż nazwa konta REST albo certyfikat spoza zaufanych	adres kończy się `/pliki`, konto `pliki`; certyfikat urzędu w magazynie komputera
Błąd 403 w logu kopii	ktoś dopisał do skryptu `forget` albo `prune`	usuwamy; retencja jest na `bkp01`
`--include` niczego nie odtwarza	ścieżka w zapisie Windowsa (`D:\Dane\...`) zamiast zapisu z migawki	ścieżkę przepisujemy z `restic ls` (`/D/Dane/...`)
Odtworzone pliki mają uprawnienia dziedziczone z celu	odtwarzanie bez praw administratora albo na Linuksie	odtwarzamy na Windowsie jako administrator, sprawdzamy `icacls`
Alarm wieku kopii `pliki` w Zabbixie, a zadanie na `fs01` zielone	kopia przeszła, ale `restic-status.sh` nie zna repozytorium	wpis w `/etc/restic/repozytoria.lista`

□□ Ćwiczenia

1. Pani Halina skasowała folder `Faktury\2026\09` we wtorek o 14:00 i zgłasza to w czwartek. Z której warstwy i której migawki odtwarzamy?

Odpowiedź

Najszybciej z poprzednich wersji (migawka z wtorku 12:00), o ile jeszcze są. Jeżeli nie, z repozytorium `pliki`: migawka restic z nocy z poniedziałku na wtorek (wtorek 3:15) zawiera folder, bo skasowano go po niej. Odtwarzamy `restic restore` z numerem tej migawki do

`D:\Odtworzenie` i kopiujemy folder na miejsce.

1. Dlaczego restic na `fs01` działa jako SYSTEM, a nie jako zwykłe konto usługi z prawem odczytu do `D:\Dane`?

Odpowiedź

Bo do migawki VSS i do odczytu pełnych deskryptorów zabezpieczeń potrzebne są uprawnienia administratora albo `SeBackupPrivilege`. Konto z samym odczytem zapisałoby tylko część uprawnień i nie zrobiłoby migawki. Konto gMSA z prawem „Kopie zapasowe plików i katalogów” jest możliwą alternatywą (do sprawdzenia, czy wystarcza dla VSS).

1. Napastnik przejął `fs01` z uprawnieniami administratora i ma plik z hasłem konta REST `pliki`. Co może zrobić z repozytorium na `bkp01`, a czego nie?

Odpowiedź

Może odczytać repozytorium (ma też hasło repozytorium z tego samego katalogu) i dopisywać nowe migawki. Nie może skasować ani nadpisać istniejących migawek, bo rest-server działa w trybie tylko do dopisywania, a retencja zawiera `--keep-within`, więc fałszywe migawki nie wypchną prawdziwych. Kopia krzyżowa na `bkp02` jest poza jego zasięgiem.

1. Test udziału `Ksiegowosc` (231 GB) trwał 1 h 13 min. Ile mniej więcej trwałoby odtworzenie 1,5 TB i co to znaczy dla RTO 4 h?

Odpowiedź

231 GB w 73 minuty to około 3,2 GB na minutę. 1,5 TB to około 1536 GB, czyli około 480 minut, mniej więcej 8 godzin. RTO 4 h dla całości jest nierealne: ustalamy kolejność udziałów (najpierw księgowość i wysyłki), rozmawiamy o RTO dla reszty albo o szybszym odtwarzaniu (drugi serwer plików, mocniejszy `bkp01`).

Powiązane: Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje · Serwery kopii bkp01 i bkp02: restic, kopia krzyżowa i niezmienność · Testy odtwarzania: plik, maszyna, baza i cała lokalizacja · fs01 od środka: maszyna na hvclu01, wolumeny, udziały działów i przestrzeń nazw DFS · Migawki VSS i poprzednie wersje na fs01: harmonogram, miejsce na różnice i dlaczego to nie jest kopia · FSRM na fs01: limity, blokowanie plików, raporty i folder pułapka na ransomware · Poświadczenia i bezpieczne hasła w skryptach · Zakres i ryzyko w Zielonej: rejestr

aktywów, macierz ryzyka i plan postępowania

Źródła (stan na 26.10.2026):

1. restic.readthedocs.io: Backing up, dokumentacja restic 0.19.1 (`--use-fs-snapshot` i VSS, opcje `vss.timeout` z domyślnymi 120 sekundami, `vss.exclude-all-mount-points`, `vss.exclude-volumes`, `vss.provider`; deskryptory zabezpieczeń z `SeBackupPrivilege` albo jako administrator)
2. restic.readthedocs.io: Scripting, kody wyjścia (0, 1, 3 dla `backup` z nieodczytanymi danymi, 10, 11, 12, 130; nieznany kod traktujemy jak błąd)
3. restic.readthedocs.io: Restoring from backup, dokumentacja 0.19.1 (`--target` z pełną ścieżką pod celem, `--include`, `--dry-run`, `--overwrite`, `--delete` tylko z `--include` albo `--exclude`, deskryptory zabezpieczeń z `SeRestorePrivilege`, `SeSecurityPrivilege` i `SeTakeOwnershipPrivilege`)
4. learn.microsoft.com: DFS Replication overview (kopie danych replikowanych programem wewnątrz maszyny, bez odtwarzania serwera z migawki maszyny, w scenariuszu maszyn Azure)
5. learn.microsoft.com: New-DfsnFolderTarget i Remove-DfsnFolderTarget (moduł DFSN); Register-ScheduledTask, New-ScheduledTaskTrigger, New-ScheduledTaskSettingsSet
6. github.com/restic/restic: strona wydań, archiwum dla Windowsa i plik SHA256SUMS wersji 0.19.1 (nazwa pliku do potwierdzenia przy pobraniu)

Revision #2
Created 2026-10-26 14:05:48 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:14 UTC by Tomasz Konieczny