# Zimne archiwum w chmurze

Azure Blob Storage i Amazon S3 Glacier z niezmiennością.


- Zimne archiwum w chmurze: Azure Blob Storage i Amazon S3 Glacier

# Zimne archiwum w chmurze: Azure Blob Storage i Amazon S3 Glacier

**◫◫ Poziom: średnio**

**◫◫ Dotyczy: Azure Blob Storage (warstwy Cool, Cold i Archive), Amazon S3 (Glacier Instant, Flexible Retrieval i Deep Archive), Azure CLI, AWS CLI v2, restic 0.19.1, rclone**

**▯ Czas: ~90 min**

*Archiwum w chmurze to miejsce na miesięczne, kwartalne i roczne paczki, które mają przetrwać utratę obu lokalizacji. Poziomy retencji (180 dni, 365 dni, 5 lat i 10 lat) pochodzą ze strony Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje, a tu pokazujemy, jak ustawić je w Azure i S3. Opisujemy tu równorzędnie Azure Blob Storage i Amazon S3: warstwy, czasy odtworzenia, niezmienność, wysyłkę narzędziami restic i rclone oraz konta, które mogą tylko zapisywać. Wracamy tu przy zakładaniu archiwum, przy zmianie retencji i wtedy, gdy trzeba coś z archiwum wyciągnąć.*

**◫◫ ZAŁOŻENIA SCENARIUSZA**

w tym ćwiczeniu obie lokalizacje wysyłają do chmury przez własne łącza internetowe, a nie przez tunel: `bkp01` przez `fw01` (adres publiczny 203.0.113.10), a `bkp02` przez `rtr-mag01` (203.0.113.40). Regiony są przykładowe: Azure `polandcentral` (tak jak na stronie Azure: podstawy operacyjne) i AWS `eu-central-1`, a ostateczny wybór regionów należy do pana Zbigniewa. Konto Azure leży w osobnej subskrypcji i osobnym tenancie Entra ID, a konto AWS w osobnym koncie, niepowiązanym z kontami, w których firma trzyma maszyny wirtualne i pocztę. W innych księgach te dostawcy mogą występować w innych rolach, na przykład Azure jako Cloud Witness klastra.

### ◫◫ Skróty użyte na tej stronie (20)

| Skrót | Co znaczy |
|---|---|
| **API** | Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą |

| Skrót | Co znaczy |
|---|---|
| **AWS** | Amazon Web Services: chmura Amazonu |
| **CLI** | Command Line Interface: praca przez wpisywanie komend w terminalu |
| **DNS** | Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP |
| **FAQ** | Frequently Asked Questions: zbiór najczęściej zadawanych pytań z odpowiedziami |
| **GB** | Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej |
| **GiB** | gibibajt: 1024 MiB, czyli około 1,07 GB; tej jednostki używa dokumentacja MailCow przy wymaganiach pamięci |
| **GRS** | Geo-Redundant Storage: nadmiarowość w Azure, która dodatkowo kopiuje dane do drugiego, odległego regionu |
| **GZRS** | Geo-Zone-Redundant Storage: nadmiarowość w Azure łącząca strefy dostępności w regionie głównym z kopią w drugim regionie |
| **IAM** | Identity and Access Management: zarządzanie tożsamościami i uprawnieniami (w AWS: użytkownicy, role, polityki) |
| **IP** | Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection) |
| **LRS** | Locally Redundant Storage: nadmiarowość w Azure, która trzyma trzy kopie danych w jednym centrum danych |
| **NFS** | Network File System: udostępnianie katalogów po sieci w świecie Linuksa |
| **RA-GRS** | Read-Access Geo-Redundant Storage: nadmiarowość GRS z możliwością odczytu danych z regionu dodatkowego |
| **RA-GZRS** | Read-Access Geo-Zone-Redundant Storage: nadmiarowość GZRS z możliwością odczytu danych z regionu dodatkowego |
| **S3** | Simple Storage Service: magazyn plików w chmurze AWS |
| **SAS** | Serial Attached SCSI: złącze i magistrala dyskowa w serwerach, następca równoległego SCSI, pozwala podłączyć wiele dysków i macierzy przez jeden kontroler z większą przepustowością niż SATA; w Azure także Shared Access Signature: podpisany adres z tokenem, który daje czasowy dostęp do danych w koncie magazynu |
| **SFTP** | SSH File Transfer Protocol: bezpieczne przesyłanie plików przez SSH |
| **VPN** | Virtual Private Network: szyfrowany tunel do sieci firmowej |
| **ZRS** | Zone-Redundant Storage: nadmiarowość w Azure, która rozkłada kopie danych na kilka stref dostępności w jednym regionie |

# Fiszka

Komendy `az` wykonujemy na `bkp01` albo na stanowisku administratora po `az login` (patrz Azure: podstawy operacyjne), komendy `aws` z profilem AWS CLI ze strony AWS CLI: instalacja, profile, poświadczenia, a `restic` i `rclone` na `bkp01` albo `bkp02` jako użytkownik `restic`. Nazwy `NazwaKonta`, `NazwaGrupyZasobow`, `NazwaKontenera`, `NazwaKubelka`, `NazwaPliku`, `LiczbaDni` zastępujemy własnymi. Konto magazynu Azure i kubełek S3 mają nazwy globalnie unikatowe. Okres niezmienności jest ustawieniem kontenera (Azure) albo domyślną retencją kubełka (S3), więc zakładamy osobny kontener i osobny kubełek na każdy poziom: kontenery `arch-RRRR-180`, `arch-RRRR-365`, `arch-RRRR-1825` i `arch-RRRR-3650` oraz kubełki `NazwaKubelka-180`, `NazwaKubelka-365`, `NazwaKubelka-1825` i `NazwaKubelka-3650`. W przykładach niżej pokazujemy poziom 365 dni, a poziom 10 lat (3650 dni) osobno tam, gdzie różni się ustawieniami.

**◫◫ ZMIENIA SYSTEM**

## Utworzenie konta magazynu Azure: az storage account create

```
az storage account create --name NazwaKonta --resource-group NazwaGrupyZasobow --location polandcentral --sku Standard_LRS --kind StorageV2 --access-tier Cool --min-tls-version TLS1_2 --allow-blob-public-access false
```

Tworzy konto ogólnego przeznaczenia v2 z nadmiarowością lokalną i bez publicznego dostępu do blobów. Warstwa Archive działa tylko z nadmiarowością LRS, GRS i RA-GRS, a nie ze ZRS, GZRS ani RA-GZRS [1], dlatego do archiwum nie wybieramy nadmiarowości strefowej. Wybór między LRS a GRS (kopia w drugim regionie) jest decyzją o koszcie.

◫◫ **LRS** = Locally Redundant Storage: nadmiarowość w Azure, która trzyma trzy kopie danych w jednym centrum danych · **GRS** = Geo-Redundant Storage: nadmiarowość w Azure, która dodatkowo kopiuje dane do drugiego, odległego regionu · **RA-GRS** = Read-Access Geo-Redundant Storage: nadmiarowość GRS z możliwością odczytu danych z regionu dodatkowego ·**ZRS** = Zone-Redundant Storage: nadmiarowość w Azure, która rozkłada kopie danych na kilka stref dostępności w jednym regionie · **GZRS** = Geo-Zone-Redundant Storage: nadmiarowość w Azure łącząca strefy dostępności w regionie głównym z kopią w drugim regionie · **RA-GZRS** = Read-Access Geo-Zone-Redundant Storage: nadmiarowość GZRS z możliwością odczytu danych z regionu dodatkowego ·**az** = Azure CLI: narzędzie chmury Microsoftu

**◫◫ ZMIENIA SYSTEM**

## Dodanie adresu biura do zapory konta: az storage account network-rule add

```
az storage account network-rule add --resource-group NazwaGrupyZasobow --account-name NazwaKonta --ip-address 203.0.113.10
```

Reguła IP dotyczy publicznych adresów i publicznego punktu końcowego konta [2]. Powtarzamy ją dla 203.0.113.40 (Swarzędz). Dopiero po dodaniu adresów przełączamy domyślną akcję na odmowę.

**IP** = Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection) · **az** = Azure CLI: narzędzie chmury Microsoftu

**ZMIENIA SYSTEM**

## Zablokowanie dostępu spoza listy adresów: az storage account update --default-action Deny

```
az storage account update --resource-group NazwaGrupyZasobow --name NazwaKonta --default-action Deny
```

Od tej chwili konto odpowiada tylko adresom z reguł. Wykonanie przed dodaniem własnego adresu zamyka dostęp także nam, więc kolejność z dwóch ostatnich fiszek jest istotna.

**az** = Azure CLI: narzędzie chmury Microsoftu

**ZMIENIA SYSTEM**

## Utworzenie kontenera na paczki: az storage container create

```
az storage container create --name arch-2027-365 --account-name NazwaKonta --auth-mode login
```

Jeden kontener na rok i poziom, bo polityka niezmienności działa na poziomie kontenera. Tak samo zakładamy `arch-2027-180`, `arch-2027-1825` i `arch-2027-3650`, a ich politykę ustawiamy z okresem równym poziomowi.

**az** = Azure CLI: narzędzie chmury Microsoftu

**ZMIENIA SYSTEM**

## Ustawienie niezablokowanej polityki czasowej: az storage container immutability-policy create

```
az storage container immutability-policy create --account-name NazwaKonta --container-name arch-2027-365 --period 365 --resource-group NazwaGrupyZasobow
```

> (gdzie `365` to okres w dniach, równy poziomowi kontenera; dozwolony zakres to od 1 do 146 000 dni [3])

Polityka niezablokowana jest do prób: można ją skrócić, wydłużyć i usunąć. Zostawiamy ją tak przez pierwszy tydzień testów.

**az** = Azure CLI: narzędzie chmury Microsoftu

**ZMIENIA SYSTEM**

## Ustawienie polityki czasowej dla paczek rocznych na 10 lat: az storage container immutability-policy create

```
az storage container create --name arch-2027-3650 --account-name NazwaKonta --auth-mode login
az storage container immutability-policy create --account-name NazwaKonta --container-name arch-2027-3650 --period 3650 --resource-group NazwaGrupyZasobow
```

> (gdzie `3650` to dziesięć lat w dniach, a dla poziomu 5 lat kontener `arch-2027-1825` dostaje `--period 1825`)

Dziesięć lat mieści się w dozwolonym zakresie od 1 do 146 000 dni [3]. Po zablokowaniu polityki wydłużyć ją można najwyżej pięć razy w kontenerze [3], więc okresu nie ustawiamy na zapas: o 10 latach decyduje zatwierdzona polityka (patrz sekcja o okresach z przepisów na stronie o strategii), a nie przypuszczenie, że dłuższy znaczy bezpieczniejszy.

**az** = Azure CLI: narzędzie chmury Microsoftu

**OSTROŻNIE · NIEODWRACALNE**

## Zablokowanie polityki czasowej: az storage container immutability-policy lock

```
az storage container immutability-policy lock --account-name NazwaKonta --container-name arch-2027-365 --if-match "EtagPolityki" --resource-group NazwaGrupyZasobow
```

> 

> (gdzie `EtagPolityki` to wartość z `az storage container immutability-policy show`, a bez niej polecenie się nie wykona)

Po zablokowaniu polityki nie da się jej usunąć ani skrócić, można ją tylko wydłużyć, w kontenerze najwyżej pięć razy [3]. Microsoft zaleca zablokowanie w ciągu 24 godzin od testów. Wykonujemy to dopiero po próbie odtworzenia, dwie osoby przy jednym stanowisku.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

**□□ ZMIENIA SYSTEM**

## Założenie blokady prawnej: az storage container legal-hold set

```
az storage container legal-hold set --tags NazwaSprawy --container-name arch-2027-365 --account-name NazwaKonta --resource-group NazwaGrupyZasobow
```

> (gdzie `NazwaSprawy` to znacznik z 3 do 23 znaków alfanumerycznych, na przykład numer zgłoszenia)

Blokada prawna trzyma dane, dopóki jej nie zdejmiemy poleceniem `az storage container legal-hold clear` z tym samym znacznikiem. Używamy jej, gdy retencja jeszcze się nie skończyła, a prawnik prosi o zamrożenie danych.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

**□□ ZMIENIA SYSTEM**

## Wysłanie pliku do warstwy Archive: az storage blob upload --tier Archive

```
az storage blob upload --file NazwaPliku --container-name arch-2027-365 --name NazwaPliku --account-name NazwaKonta --tier Archive --auth-mode login
```

Dokumentacja polecenia wymienia przy `--tier` warstwy Hot, Cool i Archive, więc warstwę Cold ustawiamy osobno przez `az storage blob set-tier` (do potwierdzenia dla naszej wersji CLI).

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

## Odczyt stanu blobu i rehydratacji: az storage blob show

```
az storage blob show --account-name NazwaKonta --container-name arch-2027-365 --name NazwaPliku --auth-mode login
```

W wyniku szukamy warstwy blobu oraz pól `archiveStatus` i `rehydrationStatus`, które pojawiają się, gdy blob jest w Archive albo trwa rehydratacja [4].

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

## Rehydratacja przez skopiowanie do kontenera roboczego: az storage blob copy start

```
az storage blob copy start --account-name NazwaKonta --source-container arch-2027-365 --source-blob NazwaPliku --destination-container odtwarzanie --destination-blob NazwaPliku --tier Cool --rehydrate-priority Standard --auth-mode login
```

Kopiowanie zostawia źródło w Archive, nie wywołuje opłaty za wcześniejsze usunięcie i nie wraca pod regułę cyklu życia, dlatego dokumentacja poleca je przed zmianą warstwy [4]. Kontener `odtwarzanie` nie ma polityki niezmienności, więc po teście można go posprzątać. Priorytet `High` wybieramy tylko w awarii.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

## Rehydratacja przez zmianę warstwy: az storage blob set-tier

```
az storage blob set-tier --tier Cool --rehydrate-priority Standard --account-name NazwaKonta --container-name arch-2027-365 --name NazwaPliku --auth-mode login
```

Zmiana warstwy nie ma odwrotu po rozpoczęciu i przenosi sam blob, więc używamy jej tylko wtedy, gdy kopii nie chcemy trzymać obok. Blob pozostaje w warstwie Archive do końca rehydratacji [4].

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

## Utworzenie kubełka S3 z włączonym Object Lock: aws s3api create-bucket

```
aws s3api create-bucket --bucket NazwaKubelka --region eu-central-1 --create-bucket-configuration LocationConstraint=eu-central-1 --object-lock-enabled-for-bucket
```

Object Lock wymaga wersjonowania i po włączeniu nie da się go wyłączyć ani zawiesić wersjonowania kubełka [5][6]. Regiony poza us-east-1 wymagają `LocationConstraint` [7]. Polecenie powtarzamy dla każdego poziomu, z nazwą `NazwaKubelka-180`, `NazwaKubelka-365`, `NazwaKubelka-1825` i `NazwaKubelka-3650`, bo domyślna retencja jest ustawieniem kubełka, a nie prefiksu.

□□ **S3** = Simple Storage Service: magazyn plików w chmurze AWS · **aws** = AWS CLI: narzędzie chmury Amazonu

□□ ZMIENIA SYSTEM

## Zablokowanie dostępu publicznego do kubełka: aws s3api put-public-access-block

```
aws s3api put-public-access-block --bucket NazwaKubelka --public-access-block-configuration BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true
```

□□ **aws** = AWS CLI: narzędzie chmury Amazonu

□□ ZMIENIA SYSTEM

## Ustawienie domyślnej retencji Object Lock: aws s3api put-object-lock-configuration

```
aws s3api put-object-lock-configuration --bucket NazwaKubelka-365 --object-lock-configuration '{"ObjectLockEnabled":"Enabled","Rule":{"DefaultRetention":{"Mode":"GOVERNANCE","Days":365}}}'
```

Tryb `GOVERNANCE` pozwala użytkownikowi z uprawnieniem `s3:BypassGovernanceRetention` skrócić albo usunąć retencję, więc nadaje się do prób [5]. Po próbie zmieniamy tryb na `COMPLIANCE` tym samym poleceniem. W trybie zgodności nikt, także użytkownik główny, nie skróci retencji ani nie usunie wersji przed terminem, a jedyną drogą jest usunięcie konta AWS [5].

Dla poziomu 10 lat ten sam kubełek `NazwaKubelka-3650` dostaje liczbę dni 3650, a poziom 5 lat (`NazwaKubelka-1825`) liczbę 1825:

```
aws s3api put-object-lock-configuration --bucket NazwaKubelka-3650 --object-lock-configuration '{"ObjectLockEnabled":"Enabled","Rule":{"DefaultRetention":{"Mode":"GOVERNANCE","Days":3650}}}'
```

Dokumentacja API pozwala podać okres w dniach (`Days`) albo w latach (`Years`), ale nie obu naraz [24]. Górnego limitu dla 10 lat na stronie opisu pól nie znaleźliśmy, więc przed wdrożeniem sprawdzamy go w dokumentacji (do potwierdzenia). Tryb `COMPLIANCE` na 10 lat ustawiamy dopiero po próbie na kubełku testowym z krótką retencją i po zatwierdzeniu okresu przez księgową.

**AWS** = Amazon Web Services: chmura Amazonu · **API** = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **aws** = AWS CLI: narzędzie chmury Amazonu

**TYLKO ODCZYT**

## Sprawdzenie retencji obiektu: aws s3api get-object-retention

```
aws s3api get-object-retention --bucket NazwaKubelka-365 --key NazwaPrefiksu/NazwaPliku
```

Pokazuje tryb i datę końca retencji. Te same dane zwraca `aws s3api head-object` razem ze stanem odtwarzania, jeśli konto ma uprawnienie `s3:GetObjectRetention` [8].

**aws** = AWS CLI: narzędzie chmury Amazonu

**ZMIENIA SYSTEM**

## Założenie blokady prawnej na obiekcie: aws s3api put-object-legal-hold

```
aws s3api put-object-legal-hold --bucket NazwaKubelka --key NazwaPrefiksu/NazwaPliku --legal-hold Status=ON
```

Blokada prawna nie ma terminu i działa niezależnie od retencji. Zdejmujemy ją tym samym poleceniem ze `Status=OFF`.

**aws** = AWS CLI: narzędzie chmury Amazonu

**ZMIENIA SYSTEM**

## Wysłanie pliku do Glacier Deep Archive: aws s3 cp --storage-class

```
aws s3 cp NazwaPliku s3://NazwaKubelka/NazwaPrefiksu/ --storage-class DEEP_ARCHIVE
```

Obiekt trafia od razu do wybranej klasy, bez opłaty za przejście z klasy Standard. Zamiast `DEEP_ARCHIVE` możemy podać `GLACIER` (Flexible Retrieval) albo `GLACIER_IR` (Instant Retrieval).

**aws** = AWS CLI: narzędzie chmury Amazonu

**ZMIENIA SYSTEM**

## Ustawienie reguły cyklu życia kubełka: aws s3api put-bucket-lifecycle-configuration

```
aws s3api put-bucket-lifecycle-configuration --bucket NazwaKubelka --lifecycle-configuration file://cykl-zycia.json
```

Plik z regułą pokazujemy niżej w sekcji o lifecycle. Reguła obejmuje obiekty wysłane narzędziem, które nie umie ustawić klasy przy zapisie.

**aws** = AWS CLI: narzędzie chmury Amazonu

**ZMIENIA SYSTEM**

## Zlecenie odtworzenia obiektu z Glacier: aws s3api restore-object

```
aws s3api restore-object --bucket NazwaKubelka --key NazwaPrefiksu/NazwaPliku --restore-request '{"Days":7,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Tworzy tymczasową kopię obiektu na `Days` dni, a oryginał zostaje w archiwum [9]. Dla Deep Archive `Standard` oznacza do 12 godzin, a `Bulk` do 48 godzin, a poziom `Expedited` nie jest tam dostępny [10].

**aws** = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

## Sprawdzenie, czy odtworzenie się skończyło: aws s3api head-object

```
aws s3api head-object --bucket NazwaKubelka --key NazwaPrefiksu/NazwaPliku
```

Dopóki odtworzenie trwa, pole `Restore` ma wartość `ongoing-request="true"` [8].

**aws** = AWS CLI: narzędzie chmury Amazonu

ZMIENIA SYSTEM

## Utworzenie repozytorium restic w S3: restic init

```
export AWS_ACCESS_KEY_ID=KluczDostepu
read -r -s -p "Sekret AWS: " AWS_SECRET_ACCESS_KEY; echo; export AWS_SECRET_ACCESS_KEY
export RESTIC_PASSWORD_FILE=/etc/restic/haslo-NazwaRepozytorium
restic -r s3:s3.eu-central-1.amazonaws.com/NazwaKubelka init
```

> (gdzie `/etc/restic/haslo-NazwaRepozytorium` to plik z hasłem repozytorium o uprawnieniach 600, a adres musi wskazywać punkt końcowy właściwego regionu [11])

Ten wariant służy do repozytorium „chłodnego" w warstwie dostępnej od ręki (Standard, Standard-IA albo Glacier Instant Retrieval), nie do Archive ani Deep Archive. Restic nie obsługuje natywnie przechowywania na zimno: odtworzenia z takiego repozytorium są wolne, a `prune` powoduje opłaty za wcześniejsze usunięcie [12][13].

**S3** = Simple Storage Service: magazyn plików w chmurze AWS · **export** = ustaw zmienną środowiskową widoczną dla uruchamianych programów · **read** = bash: wczytaj wiersz z klawiatury do zmiennej, a z opcją -s bez wypisywania wpisanego tekstu na ekran, na przykład dla tokenu API · **AWS** = Amazon Web Services: chmura Amazonu · **restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

ZMIENIA SYSTEM

## Utworzenie repozytorium restic w Azure Blob Storage: restic init

```
export AZURE_ACCOUNT_NAME=NazwaKonta
read -r -s -p "Klucz konta Azure: " AZURE_ACCOUNT_KEY; echo; export AZURE_ACCOUNT_KEY
export RESTIC_PASSWORD_FILE=/etc/restic/haslo-NazwaRepozytorium
restic -o azure.access-tier=Cool -r azure:NazwaKontenera:/NazwaRepozytorium init
```

> (gdzie `azure.access-tier` przyjmuje według dokumentacji restic wartości Hot, Cool albo Cold [11], a zamiast klucza konta możemy ustawić `AZURE_ACCOUNT_SAS`)

Nie wybieramy tu Archive, z tych samych powodów co przy S3.

**S3** = Simple Storage Service: magazyn plików w chmurze AWS · **export** = ustaw zmienną środowiskową widoczną dla uruchamianych programów · **read** = bash: wczytaj wiersz z klawiatury do zmiennej, a z opcją -s bez wypisywania wpisanego tekstu na ekran, na przykład dla tokenu API · **restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

**ZMIENIA SYSTEM**

## Zasilenie repozytorium archiwalnego z kopii dziennej: restic init --copy-chunker-params

```
restic -r /backup/archiwum/arch-2027-365/NazwaRepozytorium init --from-repo
/backup/rest/NazwaRepozytorium --from-password-file /etc/restic/haslo-NazwaRepozytorium
--password-file /etc/restic/haslo-archiwum --copy-chunker-params
```

Inicjalizacja z tymi samymi parametrami podziału na bloki, bo inaczej skopiowane pliki mogą zajmować do dwa razy więcej miejsca w repozytorium docelowym [14]. Parametrów istniejącego repozytorium nie da się zmienić.

**restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

**ZMIENIA SYSTEM**

## Skopiowanie migawki do repozytorium archiwalnego: restic copy

```
restic -r /backup/archiwum/arch-2027-365/NazwaRepozytorium --password-file
/etc/restic/haslo-archiwum copy --from-repo /backup/rest/NazwaRepozytorium --from-
```

```
password-file /etc/restic/haslo-NazwaRepozytorium IdMigawki
```

> (gdzie `IdMigawki` to identyfikator migawki z `restic snapshots`, na przykład ostatniej z danego miesiąca)

W repozytorium archiwalnym nigdy nie uruchamiamy `forget` ani `prune`. Rośnie przez rok, jest tylko dopisywane, a po upływie retencji ostatniego pliku kasujemy je w całości. Dla każdego poziomu jest osobne repozytorium (`arch-RRRR-180`, `arch-RRRR-365` i dalej), bo każdy poziom ma inny okres niezmienności.

**restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

**ZMIENIA SYSTEM**

## Wysłanie repozytorium archiwalnego do Azure: rclone copy

```
rclone copy /backup/archiwum/arch-2027-365 azure-zielona:arch-2027-365 --azureblob-access-tier archive --ignore-existing --exclude "locks/**"
```

Kopiuje tylko pliki, których jeszcze nie ma, więc istniejące bloby w Archive nie są dotykane. Opcja `--azureblob-access-tier` przyjmuje według dokumentacji rclone także `archive` [15]. Pomijamy katalog `locks`, bo to krótkotrwałe pliki restic.

**rclone** = rsync for cloud storage: synchronizacja z chmurami

**ZMIENIA SYSTEM**

## Wysłanie repozytorium archiwalnego do S3: rclone copy

```
rclone copy /backup/archiwum/arch-2027-365 s3-zielona:NazwaKubelka-365/arch-2027 --s3-storage-class DEEP_ARCHIVE --ignore-existing --exclude "locks/**"
```

Opcja `--s3-storage-class` przyjmuje między innymi `GLACIER`, `DEEP_ARCHIVE` i `GLACIER_IR` [16].

**S3** = Simple Storage Service: magazyn plików w chmurze AWS · **rclone** = rsync for cloud storage: synchronizacja z chmurami

**ZMIENIA SYSTEM**

## Zlecenie rehydratacji katalogu w Azure: rclone settier

```
rclone settier Cool azure-zielona:arch-2027-365/NazwaRepozytorium
```

Zmienia warstwę wszystkich blobów w katalogu, czyli rozpoczyna rehydratację [17]. Nie ma w niej wyboru priorytetu, więc przy pilnej potrzebie używamy `az storage blob copy start` z `--rehydrate-priority High`.

**rclone** = rsync for cloud storage: synchronizacja z chmurami

**ZMIENIA SYSTEM**

## Zlecenie odtworzenia katalogu w S3: rclone backend restore

```
rclone backend restore s3-zielona:NazwaKubelka-365/arch-2027/NazwaRepozytorium -o priority=Standard -o lifetime=7
```

Rclone sam nie odczyta obiektu z Glacier, dopóki ten nie zostanie odtworzony [16]. Stan sprawdzamy poleceniem `rclone backend restore-status` z tą samą ścieżką.

**S3** = Simple Storage Service: magazyn plików w chmurze AWS · **rclone** = rsync for cloud storage: synchronizacja z chmurami

**TYLKO ODCZYT**

## Sprawdzenie odtworzonego repozytorium: restic check

```
restic -r /backup/odtworzenie/arch-2027-365/NazwaRepozytorium --password-file /etc/restic/haslo-archiwum check --read-data-subset=10%
```

Po pobraniu z chmury sprawdzamy 10 procent paczek, a przy teście rocznym całość opcją `--read-data` [14].

**restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

**PLIK KONFIGURACYJNY**

## Plik konfiguracyjny rclone dla obu dostawców: rclone.conf

```
[azure-zielona]
type = azureblob
account = NazwaKonta
sas_url = AdresSasKontenera

[s3-zielona]
type = s3
provider = AWS
access_key_id = KluczDostepu
secret_access_key = SekretDostepu
region = eu-central-1
```

Plik leży w `/etc/rclone/rclone.conf` z uprawnieniem 600, właściciel `restic`. Dla Azure rclone akceptuje klucz konta, adres SAS albo uwierzytelnianie z środowiska [15]. Adres SAS ze strony kontenera zawęża operacje do jednego kontenera.

**SAS** = Serial Attached SCSI: złącze i magistrala dyskowa w serwerach, następca równoległego SCSI, pozwala podłączyć wiele dysków i macierzy przez jeden kontroler z większą przepustowością niż SATA; w Azure także Shared Access Signature: podpisany adres z tokenem, który daje czasowy dostęp do danych w koncie magazynu · **type** = pokaż, czym jest dane słowo dla powłoki: poleceniem, aliasem czy funkcją · **AWS** = Amazon Web Services: chmura Amazonu

**PLIK KONFIGURACYJNY**

## Reguła cyklu życia dla kubełka S3 poziomu 365 dni: cykl-zycia.json

```
{
  "Rules": [
    {
      "ID": "poziom-365-do-deep-archive-i-wygasniecie",
      "Filter": { "Prefix": "arch-2027/" },
      "Status": "Enabled",
      "Transitions": [
        { "Days": 1, "StorageClass": "DEEP_ARCHIVE" }
      ],
      "Expiration": { "Days": 731 },
      "NoncurrentVersionExpiration": { "NoncurrentDays": 1 }
    }
  ]
}
```

Przy zapisie z podaną klasą (`rclone`, `aws s3 cp`) przejście do Deep Archive w tej regule nie jest potrzebne. Obiekty mniejsze niż 128 KB reguła domyślnie pomija [18], więc małe pliki repozytorium (migawki i część indeksów) wysłane bez klasy zostałyby w klasie Standard. Wygaśnięcie ustawiamy na poziom plus 366 dni (365 + 366 = 731), a nie na sam poziom, i powód jest w sekcji „Poziomy retencji w chmurze". Wersja zablokowana przez Object Lock nie zniknie, dopóki trwa retencja; zachowanie reguły wobec zablokowanych wersji i znaczników usunięcia sprawdzamy na kubełku testowym (do potwierdzenia).

**S3** = Simple Storage Service: magazyn plików w chmurze AWS

**PLIK KONFIGURACYJNY**

## Reguła cyklu życia dla kontenera Azure poziomu 365 dni: zasada-cyklu-zycia.json

```
{
  "rules": [
    {
      "name": "poziom-365-wygasniecie",
      "type": "Lifecycle",
      "definition": {
        "filters": {
          "blobTypes": ["blockBlob"],
          "prefixMatch": ["arch-2027-365/"]
        },
        "actions": {
          "baseBlob": {
            "delete": {
              "daysAfterModificationGreaterThan": 731
            }
          }
        }
      }
    }
  ]
}
```

Wykonujemy `az storage account management-policy create --account-name NazwaKonta --resource-group NazwaGrupyZasobow --policy @zasada-cyklu-zycia.json`. Prefiks zaczyna się od nazwy kontenera (zapis do potwierdzenia na koncie testowym), a bloby wysłane od razu do

warstwy Archive nie potrzebują reguły przejścia. Dokumentacja Azure mówi, że działanie `delete` nie działa na blobach w kontenerze z niezmiennością [25]. Czy po upływie okresu niezmienności reguła zaczyna działać, sprawdzamy na koncie testowym (do potwierdzenia). Jeśli nie, repozytorium kasujemy skryptem po upływie okresu ostatniego pliku.

# Z życia wzięte

**□□ Z ŻYCIA WZIĘTE**

Środa, 8:40. Ola prosi o stan jednego repozytorium w Gitea sprzed pięciu miesięcy, bo ktoś z zespołu klienta twierdzi, że dostał od nas inny kod. Tomek znajduje właściwą paczkę miesięczną w kontenerze `arch-2026-180`, zleca rehydratację i dostaje komunikat, że blob jest w trakcie przenoszenia. Odpowiedź przychodzi dopiero po południu, Ola dostaje plik na drugi dzień rano, a Tomek dopisuje do protokołu zmierzony czas: kilkanaście godzin, nie minuty.

# Jak to działa

**□□ Rozwiń: jak to działa pod spodem**

## Dlaczego dwie warstwy w chmurze

Restic nie obsługuje natywnie przechowywania na zimno, bo do pracy czyta z repozytorium indeksy i paczki, a `prune` przepisuje paczki [12][13]. Dokumentacja i maintainerzy radzą trzymać repozytorium w warstwie dostępnej od ręki, a Glacier traktować co najwyżej jako jednorazowe, dopisywane kopie, których nie planujemy odtwarzać [13]. Dlatego w Zielonej rozdzielamy dwie rzeczy.

1. Chmurowe repozytorium restic, jeśli w ogóle je mamy, leży w warstwie online (Azure Cool albo S3 Standard-IA) i służy do małych, ważnych zbiorów, które trzeba odtworzyć szybko po utracie obu lokalizacji (na przykład Vaultwarden i konfiguracje).
2. Zimne archiwum to osobne repozytoria restic na dysku serwera kopii, po jednym na rok i poziom (`arch-RRRR-180`, `arch-RRRR-365`, `arch-RRRR-1825`, `arch-RRRR-3650`), zasilane `restic copy` miesięcznie, kwartalnie albo raz w roku i nigdy niesprzątane. Do chmury wysyłamy je jako zwykłe pliki narzędziem rclone, wprost do warstwy Archive albo Deep Archive.

Odtworzenie z archiwum składa się z trzech kroków: rehydratacja (godziny), pobranie całego repozytorium (albo potrzebnych paczek) na `bkp01` i dopiero `restic restore` lokalnie. Pobieramy całość, bo bez indeksów i migawek restic nie wie, które paczki są potrzebne.




*Rysunek 1. Droga paczki archiwalnej: repozytorium dzienne, repozytorium archiwalne na dysku, rclone do Azure i S3, a przy odtworzeniu rehydratacja, pobranie na bkp01 i restic restore (kliknięcie otwiera pełny rozmiar)*

# Azure Blob Storage

> **Pojęcie: warstwa dostępu.** Ustawienie blobu, które decyduje o cenie przechowywania, cenie odczytu i czasie dostępu. Im zimniejsza warstwa, tym taniej leży i drożej oraz dłużej się odczytuje.

| Warstwa | Dostęp | Minimalny czas przechowywania | Uwagi |
|---|---|---|---|
| Hot | od ręki | brak | codzienne dane |
| Cool | od ręki | 30 dni | rzadko czytane dane |
| Cold | od ręki | 90 dni | jeszcze rzadziej czytane |
| Archive | offline, godziny | 180 dni | wymaga rehydratacji; tylko LRS, GRS i RA-GRS |

Dane według [1]. Cold może być domyślną warstwą konta, Archive nie.

Opłata za wcześniejsze usunięcie jest proporcjonalna do brakujących dni minimum i dotyczy bloba usuniętego, nadpisanego albo przeniesionego do innej warstwy przed upływem minimum: blob w Archive usunięty po 45 dniach jest obciążony za 135 dni, czyli 180 minus 45 [1]. Dla naszych poziomów od 180 dni w górę niezmienność i tak blokuje usunięcie przed terminem, więc opłata groziłaby tylko po pomyłce na koncie bez niezmienności albo przy przenoszeniu bloba między warstwami.

Rehydratacja przenosi blob z Archive do warstwy online. Priorytet standardowy trwa do 15 godzin dla obiektów poniżej 10 GB, wysoki zwykle krócej niż godzinę dla obiektów poniżej 10 GB. Przy priorytecie wysokim obowiązuje limit 10 GiB na godzinę na konto, a czasy dla wielu blobów naraz nie skalują się liniowo [4]. Dla naszych paczek (pojedyncze pliki po kilkanaście megabajtów, ale tysiące) planujemy zatem dni, nie godziny (to szacunek, który potwierdzamy testem rocznym).

Niezmienność w Azure działa w dwóch trybach [3]:

| Mechanizm | Co robi | Uwagi |
|---|---|---|
| Polityka czasowa (time-based retention) | blob można tworzyć i czytać, a nie zmieniać ani kasować przez ustalony czas | od 1 do 146 000 dni; po zablokowaniu nie da się jej skrócić ani usunąć, wydłużyć najwyżej pięć razy w kontenerze |
| Blokada prawna (legal hold) | to samo, ale bez terminu, do zdjęcia | wymaga znacznika, na przykład numeru sprawy |

Zmiana warstwy jest dozwolona dla blobów chronionych (wszystkie warstwy obsługują niezmienność) [3]. Poziom kontenera nie wymaga wersjonowania, a poziom wersji wymaga. Niezmienność nie jest dostępna na kontach z SFTP, z NFS 3.0, z odtwarzaniem do punktu w czasie i ze śledzeniem ostatniego dostępu [3], więc na koncie archiwalnym tych funkcji nie włączamy.

Dostęp do konta ograniczamy na dwa sposoby. Pierwszy to reguły IP zapory konta: dopuszczają publiczne adresy `fw01` i `rtr-mag01`, działają tylko dla publicznego punktu końcowego, a konto ma ich limit 400 [2]. Drugi to prywatny punkt końcowy (sub-zasób `blob`, strefa DNS `privatelink.blob.core.windows.net`), z którego korzystamy, gdy ruch ma iść tunelem VPN do sieci wirtualnej w Azure, a nie przez internet [19]. Prywatny punkt końcowy nie zamyka sam publicznego, więc zaporę konta ustawiamy osobno [19]. Dla Zielonej zaczynamy od reguł IP, bo prywatny punkt końcowy wymaga sieci wirtualnej, bramy VPN i DNS, czyli trzech kolejnych rzeczy do utrzymania.

# Amazon S3

> **Pojęcie: klasa magazynowania.** Odpowiednik warstwy w Azure: wariant przechowywania obiektu z własną ceną, czasem dostępu i minimalnym okresem rozliczanym niezależnie od tego, jak długo obiekt faktycznie leżał. Opis podstawowy jest na stronie S3: kubełki, obiekty, uprawnienia.

| Klasa | Dostęp | Minimalny czas przechowywania | Czas odtworzenia |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Standard-IA | od ręki | 30 dni | milisekundy |
| Glacier Instant Retrieval | od ręki | 90 dni | milisekundy |
| Glacier Flexible Retrieval | po zleceniu odtworzenia | 90 dni | Expedited 1 do 5 minut, Standard 3 do 5 godzin, Bulk 5 do 12 godzin |
| Glacier Deep Archive | po zleceniu odtworzenia | 180 dni | Standard do 12 godzin, Bulk do 48 godzin, Expedited niedostępny |

Dane według [10][20]. Przy usunięciu, nadpisaniu albo zmianie klasy Glacier przed minimum płacimy za resztę tego okresu [23]. Do obiektów w klasach Glacier Flexible Retrieval i Deep Archive S3 dolicza około 40 KB narzutu na obiekt (32 KB po stawce archiwum i 8 KB po stawce Standard) [18], co ma znaczenie przy tysiącach małych plików.

Object Lock działa tylko w kubełkach z wersjonowaniem [5]. Są dwa tryby i blokada prawna:

| Mechanizm | Kto może skrócić albo usunąć | Uwagi |
|---|---|---|
| Tryb governance | użytkownik z uprawnieniem `s3:BypassGovernanceRetention` i nagłówkiem `x-amz-bypass-governance-retention:true` | na próby i na retencję, od której może być wyjątek |
| Tryb compliance | nikt, także użytkownik główny | retencji nie da się skrócić, a trybu zmienić; usunięcie konta AWS jest jedyną drogą [5] |
| Legal hold | użytkownik z `s3:PutObjectLegalHold` | bez terminu, niezależna od retencji |

Domyślną retencję ustawiamy na kubełku w dniach albo latach, a S3 liczy datę końca od czasu utworzenia wersji obiektu [5]. Object Lock można włączyć także na istniejącym kubełku z wersjonowaniem, co wymaga uprawnienia `s3:PutBucketObjectLockConfiguration`, ale operacja jest nieodwracalna [6].

> **Pojęcie: reguła cyklu życia (lifecycle).** Zasada kubełka, która automatycznie przenosi obiekty między klasami albo je usuwa po zadanym czasie. Jedna reguła nie może przenosić obiektu z klasy, zanim minie jej minimalny czas przechowywania (dokumentacja podaje przykład, w którym przejście do Deep Archive musi nastąpić po co najmniej 94 dniach), więc kolejne kroki układamy z uwzględnieniem tych czasów [18].

# Poziomy retencji w chmurze

Tabela zbiera ustawienia chmury dla poziomów ze strony Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje. Wszystkie cztery poziomy leżą w najzimniejszej warstwie (Archive albo Deep Archive), bo minimum 180 dni zgadza się z najkrótszym poziomem chmurowym.

| Poziom | Co trafia | Azure: kontener i niezmienność | S3: kubełek i Object Lock | Wygaśnięcie reguły (poziom plus 366 dni) |
|---|---|---|---|---|
| 180 dni | paczki miesięczne | `arch-RRRR-180`, 180 dni | `NazwaKubelka-180`, 180 dni | 546 dni |
| 365 dni | paczki kwartalne | `arch-RRRR-365`, 365 dni | `NazwaKubelka-365`, 365 dni | 731 dni |
| 5 lat | paczki roczne | `arch-RRRR-1825`, 1825 dni | `NazwaKubelka-1825`, 1825 dni | 2191 dni |
| 10 lat | paczki roczne | `arch-RRRR-3650`, 3650 dni | `NazwaKubelka-3650`, 3650 dni | 4016 dni |

Dlaczego wygaśnięcie to poziom plus 366 dni. Retencja w S3 liczy się od utworzenia wersji obiektu [5], a w Azure analogicznie od utworzenia bloba w kontenerze z polityką (do potwierdzenia na koncie testowym). Repozytorium `arch-RRRR-POZIOM` rośnie jednak przez cały rok: styczniowe pliki są starsze o jedenaście miesięcy od grudniowych, a grudniowa migawka może korzystać z bloków zapisanych w styczniowych plikach (deduplikacja). Gdyby reguła wygaszała każdy plik dokładnie po poziomie, styczniowe pliki zniknęłyby wtedy, gdy migawki z grudnia jeszcze miałyby żyć, a repozytorium przestałoby się odtwarzać. Dodatek 366 dni gwarantuje, że w chwili wygaśnięcia dowolnego pliku wszystkie migawki, które mogły z niego korzystać, mają już dawno za sobą swój okres. Cena to najwyżej rok dłuższe przechowywanie, więc przy 10 latach dodatek jest tańszy niż ryzyko uszkodzonego archiwum.

Kosztem poziomów 5 i 10 lat nie jest wcześniejsze usunięcie (niezmienność i minimum 180 dni to wykluczają), tylko przechowywanie przez lata, rehydratacja i narzut około 40 KB na obiekt w klasach Glacier [18], więc repozytorium roczne zawiera niewiele dużych plików. Poziom 5 lat nie ma sensu bez uzgodnienia z księgową, a poziom 10 lat w trybie zgodności jest nieodwracalny: pomyłka w nazwie kontenera oznacza dziesięć lat rachunków za zły katalog.

# Porównanie

| Cecha | Azure Blob Storage | Amazon S3 |
|---|---|---|
| Najzimniejsza warstwa | Archive | Glacier Deep Archive |
| Warstwa pośrednia, odczyt od ręki | Cool i Cold | Standard-IA i Glacier Instant Retrieval |

| Cecha | Azure Blob Storage | Amazon S3 |
|---|---|---|
| Czas odtworzenia z najzimniejszej | rehydratacja do 15 godzin (standard, obiekt poniżej 10 GB) | do 12 godzin (standard), do 48 godzin (bulk) |
| Minimalny czas przechowywania najzimniejszej | 180 dni | 180 dni |
| Niezmienność | polityka czasowa (wersja zablokowana albo nie) i blokada prawna | Object Lock: governance, compliance i legal hold |
| Wymóg wersjonowania | tylko dla niezmienności na poziomie wersji | zawsze |
| Ograniczenie dostępu | reguły IP i prywatny punkt końcowy | polityka kubełka z warunkami, punkty końcowe VPC [do potwierdzenia] |
| Struktura kosztu | opłata za gigabajt miesięcznie (zależna od warstwy), za operacje, za rehydratację i odczyt, za wcześniejsze usunięcie, za transfer wychodzący | analogicznie: opłata za gigabajt, za żądania, za odtworzenie, za wcześniejsze usunięcie, narzut na obiekt w klasach Glacier, za transfer wychodzący |

Cen nie podajemy, bo zmieniają się i zależą od regionu. Do rozmowy z panem Zbigniewem liczymy je z kalkulatorów obu dostawców dla naszej objętości i naszej liczby plików.

# Narzędzia wysyłki

Restic i rclone to nasze przykłady, a nie rekomendacja marki. Ta sama klasa potrzeb ma inne narzędzia i warto je znać:

| Klasa | Przykład | Co robi i co sprawdzamy |
|---|---|---|
| Kopia z deduplikacją i szyfrowaniem na własny serwer | restic | opisany na tej i następnej stronie |
| Synchronizacja plików z chmurami | rclone | kopiuje pliki do wielu dostawców, nie robi deduplikacji |
| Serwer kopii maszyn z deduplikacją | Proxmox Backup Server | obsługuje magazyn zgodny z S3 jako backend z lokalnym cache, dokumentacja ostrzega przed kosztami żądań i transferu [21]; zadania synchronizacji z drugim serwerem opisuje strona Proxmox VE w utrzymaniu: kopie PBS, aktualizacje, użytkownicy |

| Klasa | Przykład | Co robi i co sprawdzamy |
|---|---|---|
| Komercyjna kopia maszyn i serwerów | Veeam | repozytorium zabezpieczone (hardened) na Linuksie z niezmiennością [22]; obsługę warstw Archive w Azure i Glacier w S3 sprawdzamy w dokumentacji produktu (do potwierdzenia) |

# Konta i uprawnienia

Najważniejsza zasada brzmi: konto, którym wysyłamy, nie może kasować. Napastnik, który zdobędzie klucz z `bkp01`, powinien móc co najwyżej dopisać śmieci, a nie wyczyścić archiwum.

| Wymaganie | Azure | AWS |
|---|---|---|
| Osobne konto do zapisu | tożsamość z rolą bez prawa kasowania (rola własna lub token SAS z uprawnieniami tylko do utworzenia i zapisu) | użytkownik IAM z `s3:PutObject` i `s3:ListBucket`, bez `s3:DeleteObject*` i bez `s3:BypassGovernanceRetention` |
| Osobne konto do odtwarzania | wejście tylko na czas testu lub awarii | analogicznie, z `s3:GetObject` i `s3:RestoreObject` |
| MFA dla administratora | wymóg dla roli administratora subskrypcji | wymóg dla użytkownika głównego i administratora IAM |
| Oddzielny tenant albo konto | subskrypcja w osobnym tenancie Entra ID | osobne konto AWS, nie w organizacji firmy |
| Awaryjne konto administratora | dwa hasła w kopertach w sejfie | analogicznie |

Przykładowa polityka IAM dla konta zapisu to szkic do sprawdzenia w dokumentacji AWS, a nazwy działań roli własnej w Azure potwierdzamy w dokumentacji ról (do potwierdzenia). Wbudowana rola „Storage Blob Data Contributor" w Azure pozwala kasować, więc nie nadaje się na konto zapisu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject", "s3:ListBucket"],
      "Resource": [
        "arn:aws:s3:::NazwaKubelka",
        "arn:aws:s3:::NazwaKubelka/*"
```

```
        ]
      }
    ]
}
```

Dobrą praktyką jest też alarm kosztów na koncie archiwalnym, bo niespodziewany transfer wychodzący albo masowe odtwarzanie zobaczymy najpierw na rachunku.

## Test odtworzenia z archiwum

Archiwum jest warte tyle, ile jego ostatni udany test. Raz w roku robimy pełny test według strony Testy odtwarzania: plik, maszyna, baza i cała lokalizacja: zlecamy rehydratację, mierzymy czas do pojawienia się plików, pobieramy repozytorium na `bkp01`, uruchamiamy `restic check` i odtwarzamy jedną maszynę i jedną bazę. Zapisujemy też koszt testu, bo w roku awarii ten rachunek będzie większy.

# Częste błędy

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| `restic init` w Archive kończy się błędami przy kolejnym `backup` | restic czyta indeksy i paczki, a obiekty w Archive są offline | repozytorium chmurowe tylko w warstwie online, archiwum jako osobne repozytorium wysyłane rclone |
| Po zablokowaniu polityki nie da się skrócić retencji | polityka zablokowana jest nieodwracalna, wydłużać wolno najwyżej pięć razy w kontenerze | testy na polityce niezablokowanej i w trybie governance, blokada dopiero po próbie |
| Tryb compliance ustawiony „na próbę" na prawdziwym kubełku | nikt nie skróci retencji, jedyną drogą jest usunięcie konta AWS | próby na osobnym kubełku testowym z krótką retencją |
| Rachunek za wcześniejsze usunięcie | paczka skasowana przed 180 dniami w Archive albo Deep Archive | retencja nie krótsza niż minimalny czas warstwy |
| Archiwum przestaje się odtwarzać po latach, choć niczego nie kasowaliśmy ręcznie | reguła `Expiration` albo `delete` wygasza pliki repozytorium po samym poziomie, a styczniowe pliki są potrzebne grudniowym migawkom | wygaśnięcie nie krótsze niż poziom plus 366 dni albo kasowanie całego repozytorium po okresie ostatniego pliku |
| Rehydratacja wraca do Archive po kilku dniach | reguła cyklu życia znów archiwizuje blob o starej dacie modyfikacji | rehydratacja przez kopiowanie do kontenera roboczego |
| `rclone` zgłasza błąd odczytu archiwalnego obiektu | obiekt jest w Archive albo Glacier i wymaga odtworzenia | najpierw `rclone settier` lub `rclone backend restore`, potem odczyt |

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| Nie możemy się połączyć z kontem Azure po ustawieniu odmowy | domyślna akcja Deny przed dodaniem adresu | dodajemy adres z konta administratora przez portal albo z innego miejsca |
| Włączona niezmienność nie działa na koncie z SFTP | niezmienność nie jest dostępna na kontach z SFTP | osobne konto bez SFTP, NFS 3.0 i odtwarzania do punktu w czasie |

# □□ Ćwiczenia

1. Dlaczego do archiwum w Azure nie wybieramy nadmiarowości ZRS?

**Odpowiedź**

Bo warstwa Archive jest dostępna tylko dla kont z nadmiarowością LRS, GRS i RA-GRS. Wybór ZRS, GZRS albo RA-GZRS wyklucza użycie warstwy Archive.

1. Paczkę z Deep Archive potrzebujemy za sześć godzin. Którego poziomu odtworzenia nie możemy użyć i co zostaje?

**Odpowiedź**

Poziom Expedited nie jest dostępny dla Deep Archive. Standard trwa do 12 godzin, więc sześć godzin nie jest gwarantowane. Jeśli czas jest krytyczny, musimy mieć drugą kopię w warstwie, z której odczyt jest od ręki, na przykład Glacier Instant Retrieval, albo odtwarzamy z kopii krzyżowej.

1. Dlaczego rehydratację w Azure zalecamy robić przez `az storage blob copy start`, a nie `set-tier`?

**Odpowiedź**

Kopiowanie zostawia źródło w Archive, więc nie ma opłaty za wcześniejsze usunięcie z warstwy ani ryzyka, że reguła cyklu życia ponownie zarchiwizuje blob. Dodatkowo kopia trafia do kontenera roboczego bez polityki niezmienności.

1. Kubełek z Object Lock założyliśmy w trybie compliance na 1825 dni i okazało się, że wysłaliśmy zły katalog. Jak go usuniemy?

**Odpowiedź**

Nie usuniemy go przed upływem retencji. W trybie compliance retencji nie skraca nikt, także użytkownik główny, a jedyną drogą jest usunięcie całego konta AWS. Dlatego próby robimy w trybie governance albo na osobnym kubełku testowym z krótkim okresem.

1. Reguła wygaszania repozytorium z poziomem 365 dni ma `Expiration` 365 dni. Co może pójść źle i jaka wartość jest bezpieczna?

**Odpowiedź**

Repozytorium roczne rośnie przez dwanaście miesięcy, a późniejsze migawki korzystają z bloków zapisanych we wcześniejszych plikach. Przy wygaszeniu po 365 dniach styczniowe pliki znikną w styczniu następnego roku, kiedy migawki z końca roku jeszcze mają żyć, więc archiwum przestanie się odtwarzać. Bezpieczna wartość to poziom plus 366 dni, czyli 731 dni.

**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 · S3: kubełki, obiekty, uprawnienia · Konto i uprawnienia do pobierania plików z S3 · Bazy w zimnym archiwum Azure: kopie sql01 i ora01, niezmienność i odtworzenie do punktu w czasie

**Źródła** (stan na 4.10.2026, pozycje 23 do 25 na 10.10.2026):


1. learn.microsoft.com: Access tiers for blob data, warstwy, minimalne czasy, opłata za wcześniejsze usunięcie, Archive tylko LRS, GRS i RA-GRS
2. learn.microsoft.com: Configure Azure Storage firewalls and virtual networks, reguły IP, domyślna akcja Deny, limit 400 reguł
3. learn.microsoft.com: Overview of immutable storage for blob data, polityka czasowa, blokada prawna, zablokowana polityka, ograniczenia kont
4. learn.microsoft.com: Archive rehydration overview, czasy rehydratacji, priorytety, limit 10 GiB na godzinę przy priorytecie wysokim, kopiowanie zamiast zmiany warstwy
5. docs.aws.amazon.com: Locking objects with Object Lock, tryby governance i compliance, legal hold, wymóg wersjonowania
6. docs.aws.amazon.com: Configuring S3 Object Lock, włączenie na istniejącym kubełku, brak możliwości wyłączenia
7. docs.aws.amazon.com: AWS CLI s3api create-bucket, `--object-lock-enabled-for-bucket`, `LocationConstraint`
8. docs.aws.amazon.com: AWS CLI s3api head-object, pola `Restore`, `ObjectLockMode`, `ObjectLockRetainUntilDate`
9. docs.aws.amazon.com: AWS CLI s3api restore-object, składnia `--restore-request`
10. docs.aws.amazon.com: Restoring archived objects, poziomy odtwarzania i czasy dla Flexible Retrieval i Deep Archive

11. restic.readthedocs.io: Preparing a new repository, backend S3 i Azure, opcja `azure.access-tier`
12. restic.readthedocs.io: FAQ, restic nie obsługuje natywnie przechowywania na zimno
13. forum.restic.net: Restic and S3 Glacier Deep Archive, uwagi o `prune` i odtwarzaniu
14. restic.readthedocs.io: Working with repositories, `copy`, `check`, parametry podziału na bloki
15. rclone.org: Microsoft Azure Blob Storage, `--azureblob-access-tier` i uwierzytelnianie
16. rclone.org: Amazon S3, `--s3-storage-class` i `rclone backend restore`
17. rclone.org: rclone settier
18. docs.aws.amazon.com: Lifecycle transition considerations, obiekty poniżej 128 KB, minimalne czasy, narzut 40 KB
19. learn.microsoft.com: Use private endpoints for Azure Storage, podzasób `blob`, strefa DNS, osobna konfiguracja zapory
20. docs.aws.amazon.com: Understanding and managing Amazon S3 storage classes, tabela klas
21. pbs.proxmox.com: Storage, magazyn zgodny z S3 jako backend datastore i zadania synchronizacji
22. helpcenter.veeam.com: Hardened Repository, niezmienność kopii na serwerze z Linuksem
23. docs.aws.amazon.com: Understanding S3 Glacier storage classes, minimalny czas przechowywania i opłata za pozostałą część okresu (stan na 10.10.2026)
24. docs.aws.amazon.com: DefaultRetention (API), pola `Days` i `Years`, brak limitu na stronie opisu pól (stan na 10.10.2026)
25. learn.microsoft.com: Azure Blob Storage lifecycle management overview, działanie `delete` nie działa dla blobów w kontenerze z niezmiennością (stan na 10.10.2026)