S3: kubełki, obiekty, uprawnienia

Poziom: podstawy **Dotyczy: Amazon S3, AWS CLI v2** **Czas: ~30 min**

S3 wygląda jak dysk sieciowy z folderami, ale nim nie jest — ta strona tłumaczy, czym naprawdę jest kubełek i obiekt, zanim przejdziemy do komend, którymi się nimi zarządza.

Komendy wykonujemy z CLI bash ze skonfigurowanym profilem AWS CLI, opisanym na stronie AWS CLI: instalacja, profile, poświadczenia.

Skróty użyte na tej stronie (10)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
AWS	Amazon Web Services: chmura Amazonu
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
IAM	Identity and Access Management: zarządzanie tożsamościami i uprawnieniami (w AWS: użytkownicy, role, polityki)
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami
KMS	Key Management Service: usługa chmurowa przechowująca i zarządzająca kluczami szyfrującymi
PDF	Portable Document Format: format dokumentu, który wygląda tak samo na każdym komputerze, np. zakres akredytacji jednostki na stronie PCA
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
S3	Simple Storage Service: magazyn plików w chmurze AWS
URL	Uniform Resource Locator: adres strony lub zasobu

Fiszka

TYLKO ODCZYT

Wylistowanie wszystkich kubełków na koncie: aws s3 ls

```
aws s3 ls
```

Bez podanej ścieżki `aws s3 ls` zwraca listę wszystkich kubełków widocznych na koncie, z datą utworzenia każdego z nich.

Przykład:

```
2024-03-11 09:22:15 przyklad-backup
2025-01-08 14:03:47 przyklad-logi-aplikacji
2025-06-02 11:47:03 przyklad-statyczne-pliki
```

aws = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

Wylistowanie zawartości kubełka: aws s3 ls

```
aws s3 ls s3://NazwaKubelka/
```

Zwraca obiekty i „foldery" (w rzeczywistości wspólne prefiksy) leżące bezpośrednio pod wskazanym kluczem — jeden poziom w dół, nie rekurencyjnie.

Przykład:

```
                           PRE faktury/
2026-01-15 08:12:33     481293 raport-styczen.csv
```

aws = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

Rekurencyjne wylistowanie całej zawartości: aws s3 ls --recursive

```
aws s3 ls s3://NazwaKubelka/ --recursive
```

Dodanie `--recursive` schodzi we wszystkie prefiksy naraz i zwraca pełną, płaską listę kluczy — przydatne, gdy chcemy zobaczyć rzeczywistą liczbę i rozmiar obiektów, a nie tylko

strukturę pierwszego poziomu.

Przykład:

```
2026-01-10 07:00:01     102934 faktury/2026/styczen/f-001.pdf
2026-01-11 07:00:02      98211 faktury/2026/styczen/f-002.pdf
2026-01-15 08:12:33     481293 raport-styczen.csv
```

aws = AWS CLI: narzędzie chmury Amazonu

ZMIENIA SYSTEM

Skopiowanie pojedynczego pliku do kubełka: aws s3 cp

```
aws s3 cp SciezkaDoPliku s3://NazwaKubelka/raport-styczen.csv
```

Kopiujemy jeden plik lokalny pod wskazany klucz w kubełku — działa symetrycznie w drugą stronę (`aws s3 cp s3://NazwaKubelka/raport-styczen.csv SciezkaDoPliku` pobiera obiekt lokalnie).

Przykład:

```
upload: ./raport-styczen.csv to s3://przyklad-backup/raport-styczen.csv
```

aws = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

Synchronizacja katalogu lokalnego z kubełkiem: aws s3 sync

```
aws s3 sync SciezkaDoPliku s3://NazwaKubelka/faktury/
```

> **Pojęcie: `sync`.** Kopiuje tylko te pliki, których nie ma jeszcze w celu albo które zmieniły rozmiar lub datę modyfikacji — w odróżnieniu od `cp`, które kopiuje zawsze, `sync` porównuje źródło i cel przed każdym transferem. Domyślnie `sync` nigdy nie usuwa plików po stronie docelowej, nawet jeśli zniknęły ze źródła.

Przykład:

```
upload: faktury/f-003.pdf to s3://przyklad-backup/faktury/f-003.pdf
upload: faktury/f-004.pdf to s3://przyklad-backup/faktury/f-004.pdf
```

aws = AWS CLI: narzędzie chmury Amazonu

OSTROŻNIE · NIEODWRACALNE

Synchronizacja z usunięciem plików osieroconych: sync --delete

```
aws s3 sync SciezkaDoPliku s3://NazwaKubelka/faktury/ --delete
```

Flaga `--delete` dodatkowo usuwa z kubełka te obiekty, których nie ma już w źródle — dokładne lustrzane odbicie katalogu lokalnego, przydatne przy publikowaniu statycznej strony, gdzie nie chcemy zostawiać nieaktualnych plików.

Uważaj: `--delete` jest nieodwracalne bez włączonego wersjonowania kubełka — usunięty obiekt bez wersjonowania znika bezpowrotnie.

aws = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

Synchronizacja z pominięciem wybranych plików: sync --exclude

```
aws s3 sync SciezkaDoPliku s3://NazwaKubelka/faktury/ --exclude "*.tmp"
```

Wzorzec podany w `--exclude` pomija pasujące pliki podczas synchronizacji — działa na ścieżce względem katalogu źródłowego, nie na samej nazwie pliku.

aws = AWS CLI: narzędzie chmury Amazonu

TYLKO ODCZYT

Synchronizacja z wyjątkiem od wykluczenia: sync --exclude i --include razem

```
aws s3 sync SciezkaDoPliku s3://NazwaKubelka/faktury/ --exclude "*" --include "*.pdf"
```

> **Pojęcie: kolejność `--exclude` / `--include`.** AWS CLI przetwarza filtry w kolejności podania w komendzie, każdy kolejny nadpisując decyzję poprzedniego dla pasujących plików. `--exclude "*"` odrzuca wszystko, a następujący po nim `--include "*.pdf"` przywraca do synchronizacji tylko pliki PDF — odwrócenie kolejności tych dwóch flag da inny, zwykle niezamierzony wynik.

AWS = Amazon Web Services: chmura Amazonu · **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **PDF** = Portable Document Format: format dokumentu, który wygląda tak samo na każdym komputerze, np. zakres akredytacji jednostki na stronie PCA · **aws** = AWS CLI: narzędzie chmury Amazonu

ZMIENIA SYSTEM

Przeniesienie obiektu w obrębie S3: aws s3 mv

```
aws s3 mv s3://NazwaKubelka/raport-styczen.csv s3://NazwaKubelka/archiwum/raport-styczen.csv
```

Wewnętrznie wykonuje kopię pod nowym kluczem i usunięcie starego — nie jest to operacja atomowa jednego zasobu, tylko dwie operacje wykonane jedna po drugiej.

Przykład:

```
move: s3://przyklad-backup/raport-styczen.csv to s3://przyklad-backup/archiwum/raport-styczen.csv
```

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

OSTROŻNIE · NIEODWRACALNE

Usunięcie pojedynczego obiektu: aws s3 rm

```
aws s3 rm s3://NazwaKubelka/archiwum/raport-styczen.csv
```

Usuwa wskazany obiekt. Bez włączonego wersjonowania usunięcie jest natychmiastowe i ostateczne.

Przykład:

```
delete: s3://przyklad-backup/archiwum/raport-styczen.csv
```

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

□□ OSTROŻNIE · NIEODWRACALNE

Usunięcie całego prefiksu: aws s3 rm --recursive

```
aws s3 rm s3://NazwaKubelka/archiwum/ --recursive
```

Bez `--recursive` `aws s3 rm` na kluczu kończącym się ukośnikiem nie usuwa niczego poza ewentualnym pustym obiektem-znacznikiem — usunięcie całej zawartości „folderu" wymaga jawnego rekurencyjnego przejścia po wszystkich pasujących kluczach.

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

□□ TYLKO ODCZYT

Wygenerowanie tymczasowego linku do obiektu: aws s3 presign

```
aws s3 presign s3://NazwaKubelka/faktury/2026/styczen/f-001.pdf --expires-in 3600
```

> **Pojęcie: presigned URL.** Adres URL zawierający w parametrach zapytania podpis kryptograficzny wygenerowany naszymi poświadczeniami — pozwala dowolnej osobie z tym linkiem pobrać (albo, przy odpowiednim wywołaniu, wgrać) konkretny obiekt bez posiadania własnych poświadczeń AWS, przez ograniczony czas określony flagą `--expires-in` (w sekundach). Przydatne, gdy chcemy udostępnić jeden plik z prywatnego kubełka bez zmiany jego uprawnień na trwałe.

Przykład:

```
https://przyklad-backup.s3.eu-central-1.amazonaws.com/faktury/2026/styczen/f-001.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20260125%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20260125T090000Z&X-Amz-Expires=3600&X-Amz-Signature=8f7b2a...
```

Uważaj: presigned URL działa dla dowolnej osoby, która go otrzyma, aż do wygaśnięcia — traktujemy go jak hasło jednorazowe, nie wklejamy do kanałów, które ktoś niepowołany może przeczytać.

□□ **URL** = Uniform Resource Locator: adres strony lub zasobu · **AWS** = Amazon Web Services: chmura Amazonu · **aws** = AWS CLI: narzędzie chmury Amazonu

Z życia wzięte

Z ŻYCIA WZIĘTE

O 14:20 ktoś w zespole próbuje „zmienić nazwę folderu" faktury w konsoli AWS i po dziesięciu minutach pyta Tomka, dlaczego operacja na milionie plików wciąż się kręci, a licznik kosztów rośnie. Tomek tłumaczy, że w S3 nie ma prawdziwych katalogów, więc zmiana nazwy prefiksu to w rzeczywistości skopiowanie i usunięcie każdego z tych plików osobno, nie jedna operacja na strukturze. Zespół przerywa zadanie, dzieli je na mniejsze partie uruchamiane w tle i od tej pory planuje duże przenosiny prefiksów poza godziny szczytu.

Czym jest kubełek i obiekt

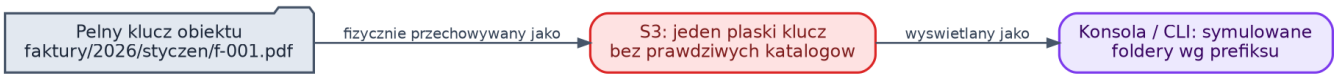
Pojęcie: kubełek (bucket). Kontener najwyższego poziomu w S3 — jednostka, do której przypisujemy region, uprawnienia, ustawienia wersjonowania i szyfrowania. Nazwa kubełka jest globalnie unikatowa w obrębie całego AWS, nie tylko naszego konta — raz zajętej nazwy nie da się użyć ponownie, dopóki poprzedni właściciel jej nie zwolni.

Pojęcie: obiekt (object). Pojedynczy plik przechowywany w kubełku, wraz z metadanymi (typ zawartości, znaczniki, nagłówki cache). Obiekt jest identyfikowany przez klucz — pełną „ścieżkę" w obrębie kubełka, np. `faktury/2026/styczen/f-001.pdf`.

Pojęcie: klucz (key). Ciąg znaków jednoznacznie identyfikujący obiekt wewnątrz kubełka. Klucz może zawierać ukośniki i konsola AWS wyświetla je jak strukturę katalogów, ale to złudzenie interfejsu — S3 nie ma pojęcia katalogu. `faktury/2026/styczen/f-001.pdf` to dla S3 jeden płaski klucz, nie trzy zagnieżdżone foldery.

Dlaczego S3 nie jest systemem plików

Mimo ukośników w kluczach i mimo że konsola i CLI potrafią symulować przechodzenie po „folderach", S3 to magazyn typu obiekt-klucz-wartość (object storage), nie system plików:



Klucz obiektu jest fizycznie jednym płaskim ciągiem znaków, a foldery widziane w konsoli czy CLI to tylko symulacja oparta na prefiksie (kliknięcie otwiera pełny rozmiar)

- Nie ma operacji „zmień nazwę katalogu" ani „przenieś podkatalog" jako operacji atomowej — zmiana klucza wymaga skopiowania obiektu pod nowym kluczem i usunięcia starego, co przy dużej liczbie obiektów pod wspólnym prefiksem oznacza tyle operacji, ile obiektów.
- Nie ma blokad plikowych ani jednoczesnego zapisu przyrostowego — obiekt zapisuje się w całości albo wcale, nie da się dopisać bajtów na końcu istniejącego obiektu tak jak do otwartego pliku.
- Pusty „folder" w konsoli AWS to w rzeczywistości osobny, zerobajtowy obiekt z kluczem kończącym się ukośnikiem — usunięcie go nie usuwa niczego poza tym jednym obiektem, w szczególności nie usuwa obiektów, których klucze zaczynają się tym samym prefiksem.

Konsekwencja praktyczna: kiedy myślimy o „katalogu" w S3, myślimy naprawdę o prefiksie klucza — wspólnym początku wielu kluczy, po którym filtrujemy i grupujemy, a nie o realnym obiekcie hierarchii.

Klasy magazynowania

> **Pojęcie: klasa magazynowania (storage class).** Wariant przechowywania tego samego obiektu, różniący się ceną za gigabajt, czasem dostępu do danych i minimalnym okresem przechowywania rozliczanym niezależnie od tego, czy obiekt faktycznie tak długo leżał. Wybór klasy jest kompromisem między kosztem przechowywania a kosztem i czasem odczytu.

Klasa	Czas dostępu	Typowe zastosowanie
`STANDARD`	natychmiastowy	dane używane regularnie
`STANDARD_IA` (Infrequent Access)	natychmiastowy, wyższa opłata za odczyt	dane rzadko czytane, ale wymagające natychmiastowego dostępu przy potrzebie
`INTELLIGENT_TIERING`	natychmiastowy	dane o nieprzewidywalnym wzorcu dostępu — S3 samo przenosi obiekt między warstwami
`GLACIER`	od minut do godzin (w zależności od trybu odzyskania)	archiwum, rzadkie odtworzenie

Klasa	Czas dostępu	Typowe zastosowanie
`GLACIER_DEEP_ARCHIVE`	do 12 godzin	archiwum długoterminowe, np. wymogi regulacyjne

Reguły cyklu życia

> **Pojęcie: reguła cyklu życia (lifecycle rule).** Zasada zapisana na poziomie kubełka, automatycznie przenosząca obiekty między klasami magazynowania albo usuwająca je po upływie zadanego czasu od utworzenia, bez ręcznej interwencji. Typowa reguła: obiekty w prefiksie `logi/` starsze niż 30 dni przenosimy do `STANDARD_IA`, starsze niż 90 dni do `GLACIER`, starsze niż 365 dni usuwamy. Reguły cyklu życia są podstawowym sposobem trzymania kosztów magazynu pod kontrolą bez ręcznego przeglądania zawartości kubełka.

Kiedy `aws s3` nie wystarcza — `aws s3api`

> **Pojęcie: `aws s3` kontra `aws s3api`.** `aws s3` to zestaw poleceń wysokopoziomowych, zaprojektowanych pod kątem wygody (kopiowanie, synchronizacja) i ukrywających część szczegółów API. `aws s3api` to bezpośrednie odwzorowanie każdego wywołania REST API S3 jeden do jednego — mniej wygodne w codziennej pracy, ale jedyna droga do operacji, których `aws s3` nie udostępnia: zarządzania politykami, wersjonowaniem, ACL-ami i metadanymi na poziomie pojedynczych wersji obiektów.

Odczytanie polityki kubełka: aws s3api get-bucket-policy

```
aws s3api get-bucket-policy --bucket NazwaKubelka --query Policy --output text
```

Zwraca aktualnie obowiązującą politykę dostępu przypisaną do kubełka w postaci dokumentu JSON — `--query Policy --output text` wyciąga sam dokument polityki bez otaczającej go koperty odpowiedzi API.

Przykład:

```
{"Version":"2012-10-17","Statement":[{"Sid":"AllowReadOnly","Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:user/nazwa_usera"},"Action":["s3:GetObject","s3:ListBucket"],"Resource":["arn:aws:s3:::przyklad-backup","arn:aws:s3:::przyklad-backup/*"]}]}
```

Włączenie wersjonowania kubełka: aws s3api put-bucket-versioning

```
aws s3api put-bucket-versioning --bucket NazwaKubelka --versioning-configuration Status=Enabled
```

> **Pojęcie: wersjonowanie (versioning).** Mechanizm, dzięki któremu nadpisanie albo usunięcie obiektu nie kasuje poprzedniej zawartości, tylko tworzy nową wersję pod tym samym kluczem, zachowując poprzednie wersje dostępne do odczytu i przywrócenia. Usunięcie obiektu w kubełku z włączonym wersjonowaniem nie usuwa danych — dodaje znacznik usunięcia (delete marker) jako najnowszą wersję, a starsze wersje wciąż leżą w kubełku, dopóki nie usunie się ich jawnie po identyfikatorze wersji.

Włączamy wersjonowanie przede wszystkim z dwóch powodów: ochrony przed przypadkowym nadpisaniem albo usunięciem (błąd operatora, błąd w skrypcie z `--delete`) oraz jako wymogu zgodności w regulacjach wymagających zachowania historii zmian dokumentu. Kosztem jest rosnące zużycie magazynu — każda nadpisana wersja nadal zajmuje miejsce, dopóki nie usunie jej reguła cyklu życia dedykowana starym wersjom.

Przykład:

```
{
    "ResponseMetadata": {
        "HTTPStatusCode": 200
    }
}
```

Wylistowanie wszystkich wersji obiektów: aws s3api list-object-versions

```
aws s3api list-object-versions --bucket NazwaKubelka --prefix faktury/2026/styczen/
```

Zwraca pełną historię wersji i znaczników usunięcia dla obiektów pod wskazanym prefiksem — jedyny sposób, żeby zobaczyć wersje inne niż najnowsza, bo `aws s3 ls` pokazuje zawsze tylko bieżący stan.

Przykład:

```
{
    "Versions": [
        {
            "Key": "faktury/2026/styczen/f-001.pdf",
            "VersionId": "3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo",
            "IsLatest": true,
            "LastModified": "2026-01-15T08:12:33.000Z",
            "Size": 102934
        },
        {
            "Key": "faktury/2026/styczen/f-001.pdf",
            "VersionId": "QUpfdndhfd8438MNFDN93jdnFkRFi9gsdzZFrN8gTsvBQ1KyMKp",
            "IsLatest": false,
            "LastModified": "2026-01-10T07:00:01.000Z",
            "Size": 101822
        }
    ]
}
```

Szyfrowanie po stronie serwera

> **Pojęcie: szyfrowanie po stronie serwera (Server-Side Encryption, SSE).** Szyfrowanie danych w spoczynku wykonywane przez S3 automatycznie przy zapisie i odszyfrowywane automatycznie przy odczycie autoryzowanym żądaniem — klient nie musi implementować własnej kryptografii, żeby dane na dysku dostawcy nie leżały jawnym tekstem. S3 udostępnia trzy warianty: `SSE-S3` (klucze zarządzane w całości przez AWS, najprostszy, domyślny dla nowych kubełków), `SSE-KMS` (klucze zarządzane przez usługę KMS, z możliwością audytu użycia klucza i rotacji), oraz `SSE-C` (klucz dostarcza klient przy każdym żądaniu, AWS go nie przechowuje). Wybór wariantu jest kompromisem między prostotą (`SSE-S3`) a kontrolą i audytowalnością (`SSE-KMS`).

Szyfrowanie po stronie serwera chroni dane leżące na dysku dostawcy przed dostępem do surowego nośnika — nie zastępuje kontroli dostępu opisanej politykami IAM i politykami kubełka na

stronie Konto i uprawnienia do pobierania plików z S3, bo autoryzowane żądanie zawsze otrzymuje dane odszyfrowane.

Częste błędy

Traktowanie prefiksu jak prawdziwego katalogu przy dużej liczbie obiektów. Zmiana nazwy „folderu" zawierającego milion kluczy oznacza milion operacji kopiowania i usuwania, nie jedną operację zmiany nazwy — czas i koszt tej operacji rośnie liniowo z liczbą obiektów, nie jest stały.

Użycie `sync --delete` bez wcześniejszego `--dryrun`. `--dryrun` pokazuje, co komenda by zrobiła, bez faktycznego wykonania transferu — pominięcie tego kroku przy pierwszym uruchomieniu `sync --delete` na nowo skonfigurowanym źródle bywa przyczyną nieodwracalnej utraty danych w kubełku bez wersjonowania.

Zakładanie, że usunięcie obiektu w kubełku z wersjonowaniem faktycznie zwalnia miejsce. Dopóki nie usunie się jawnie starych wersji (ręcznie albo regułą cyklu życia), obiekt „usunięty" nadal zajmuje magazyn pod poprzednimi identyfikatorami wersji.

□□ Ćwiczenia

> Zadania wykonujemy na testowym kubełku, najlepiej pustym i przeznaczonym wyłącznie do ćwiczeń.

1. Wyjaśniamy, dlaczego S3 nie jest systemem plików, mimo że konsola AWS pokazuje jego zawartość jako drzewo folderów.
2. Tworzymy lokalny katalog z kilkoma plikami, synchronizujemy go do testowego kubełka komendą `sync`, usuwamy jeden plik lokalnie i sprawdzamy różnicę między ponownym `sync` bez i z flagą `--delete`.
3. Włączamy wersjonowanie na testowym kubełku, nadpisujemy ten sam obiekt dwa razy i komendą `list-object-versions` sprawdzamy, ile wersji faktycznie tam leży.
4. Generujemy presigned URL ważny 60 sekund, otwieramy go w przeglądarce od razu i ponownie po minucie — opisujemy różnicę w zachowaniu.

Powiązane: AWS CLI: instalacja, profile, poświadczenia · Konto i uprawnienia do pobierania plików z S3 · Pobieranie plików z S3 przez PowerShell · Backup bazy do S3: agent backint i jego ograniczenia

Źródła: dokumentacja AWS CLI, sekcja `s3` i `s3api` · dokumentacja Amazon S3, rozdziały o klasach magazynowania, wersjonowaniu i szyfrowaniu po stronie serwera

Revision #38
Created 2026-09-25 08:26:20 UTC by Tomasz Konieczny
Updated 2026-10-26 14:16:23 UTC by Tomasz Konieczny