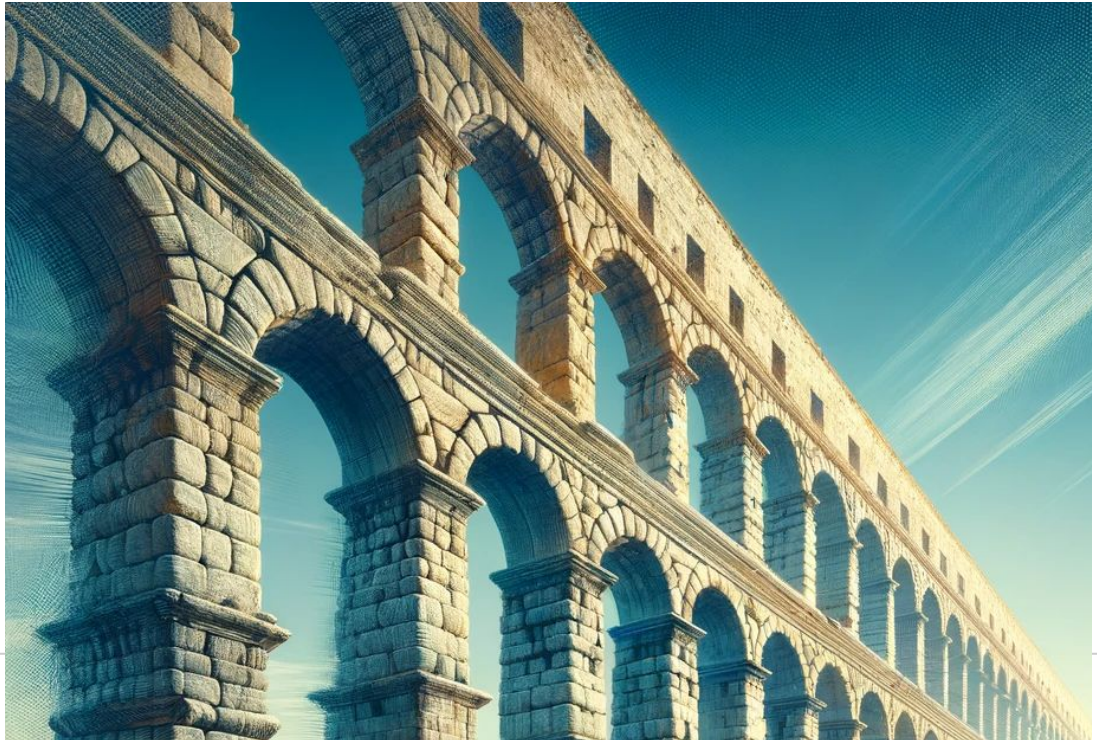
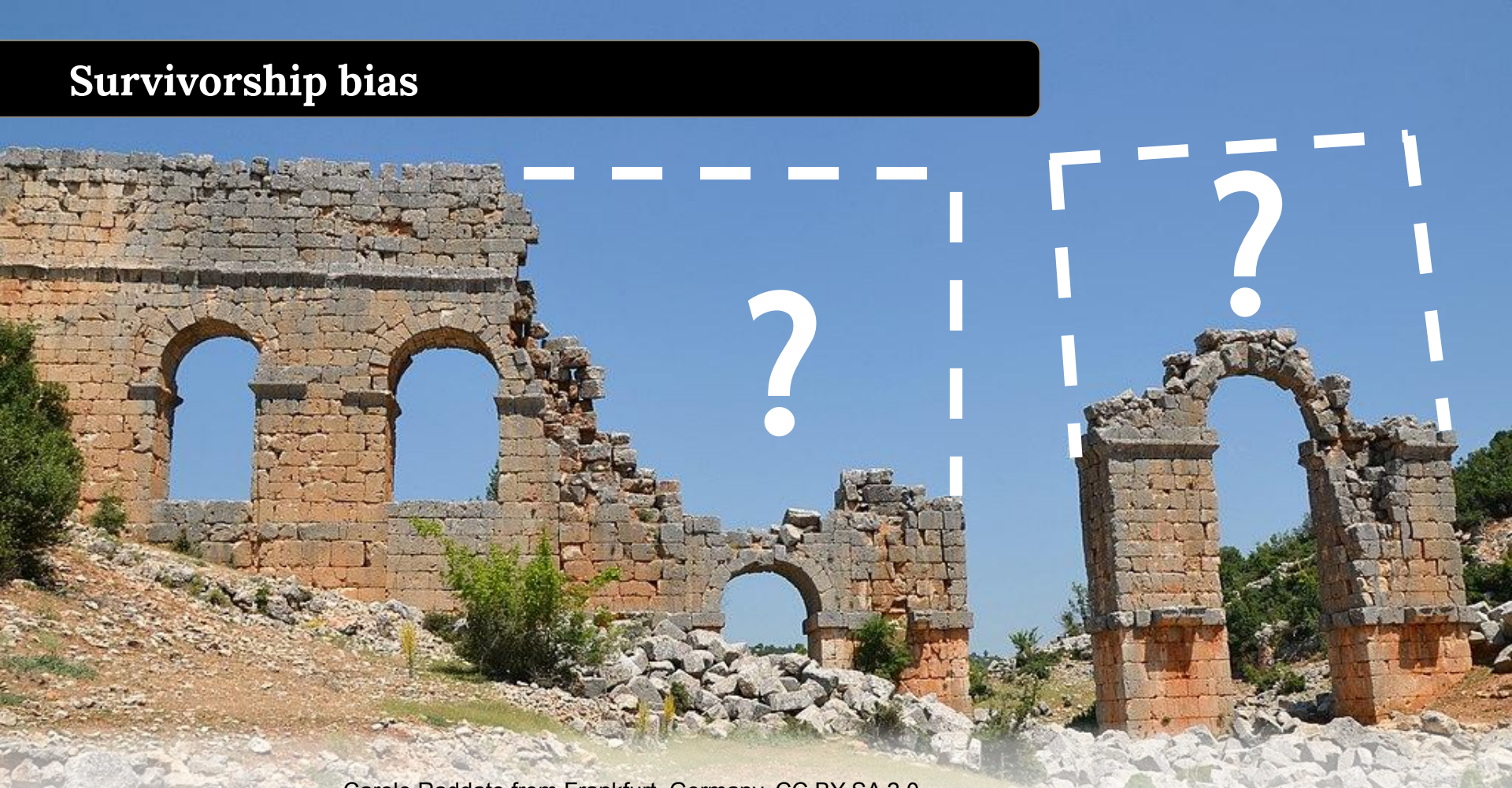
Testing your PostgreSQL backups (a practical guide)

Nick Meyer
Academia.edu
SCaLE 21x

“We just don’t build things like we used to”



Survivorship bias



Academia.edu

- https://www.academia.edu/about
- **Our goals**
  1. Ensure that every paper ever written is:
     - ✓ on the internet
     - ✓ available for free
  2. Accelerate the world's research
- **Some stats**
  1. 47 million papers uploaded
  2. 20 million paper recommendations per day

Academia.edu - postgres + engineering stats

- **Data: ~100TB across ~15 “clusters”**
  - Entirely on AWS
  - Some Aurora, some self-managed (EC2)
- **HA setup + high read workloads**
  - Tons of read-only replicas
- **50 (+/-) engineers**

A bit about me (Nick Meyer)

- https://github.com/aristocrates
- **Team lead of Platform Engineering**
- **Areas of focus**
  - Developer experience
  - Interface: application and infra
  - **Data layer**
  - **Postgres**

A Our old postgres backup solution

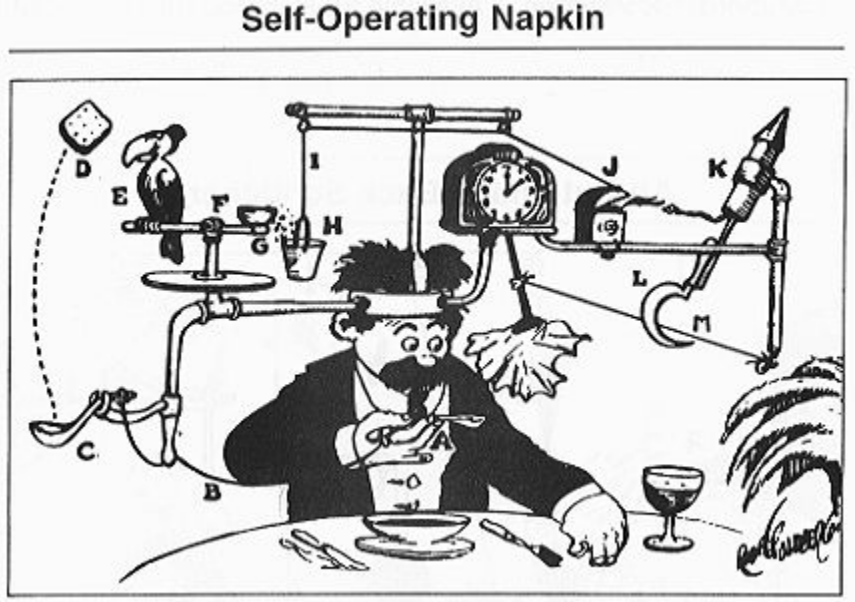
- Ruby script
- A great way to learn about backups...
- ... but a bad idea otherwise

A

“A practical guide”

Backup testing

No time for this



A

“Experience is the best teacher”

The experience of others: “cost-effective teacher”

A Roadmap

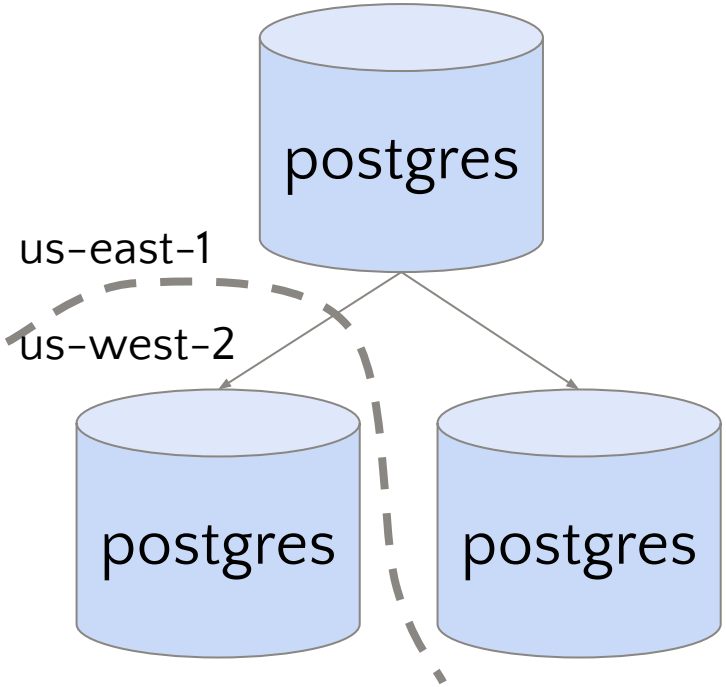
1. Why do we have backups: What could go wrong?
2. How to test backups
3. Measurable goals
4. Monitoring: how backups work
5. Monitoring: how to monitor

What could go wrong?

A

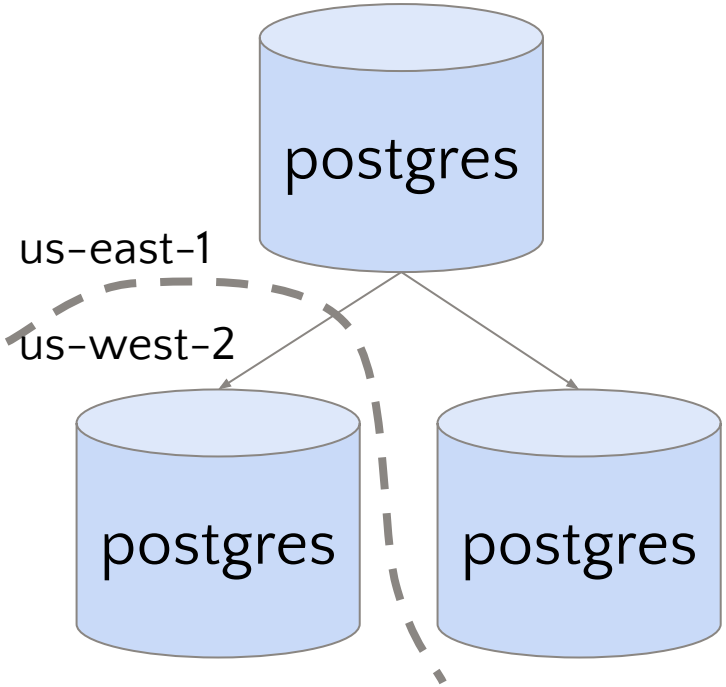
A What could go wrong: Several nodes

- What if all nodes go down?
- Some nodes go down: all good?

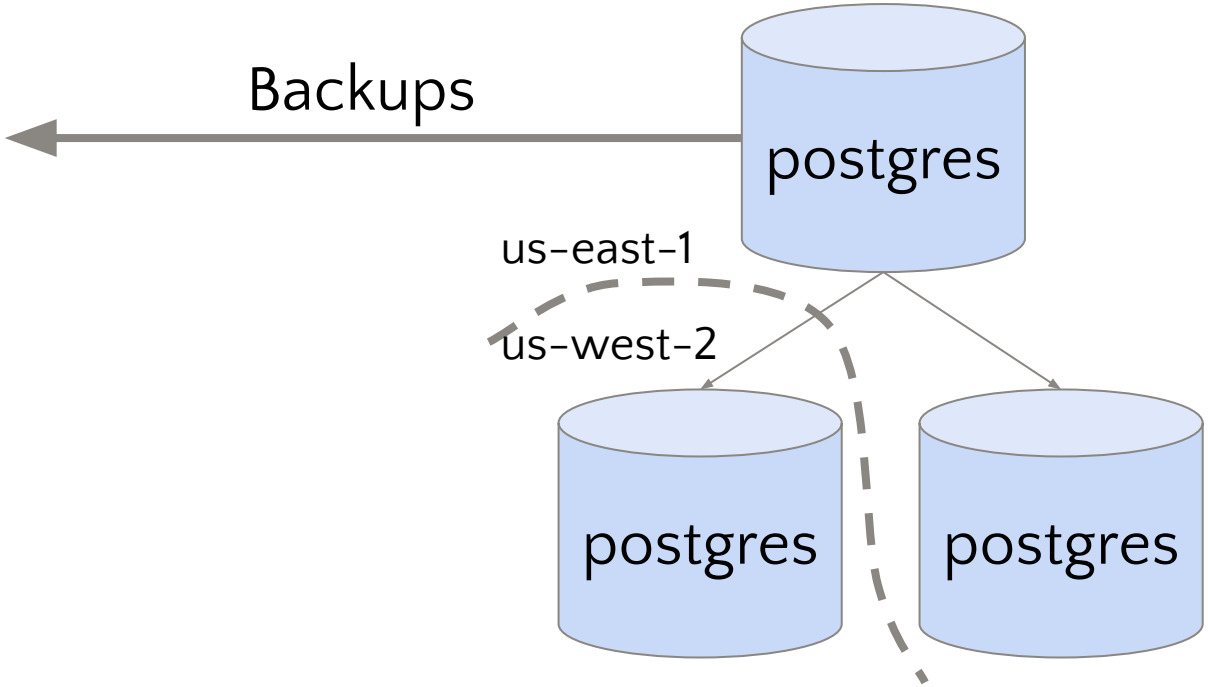


A What could go wrong: Several nodes

- What if all nodes go down?
- Some nodes go down: all good?
- `DELETE FROM users;`
- `DROP TABLE users;`



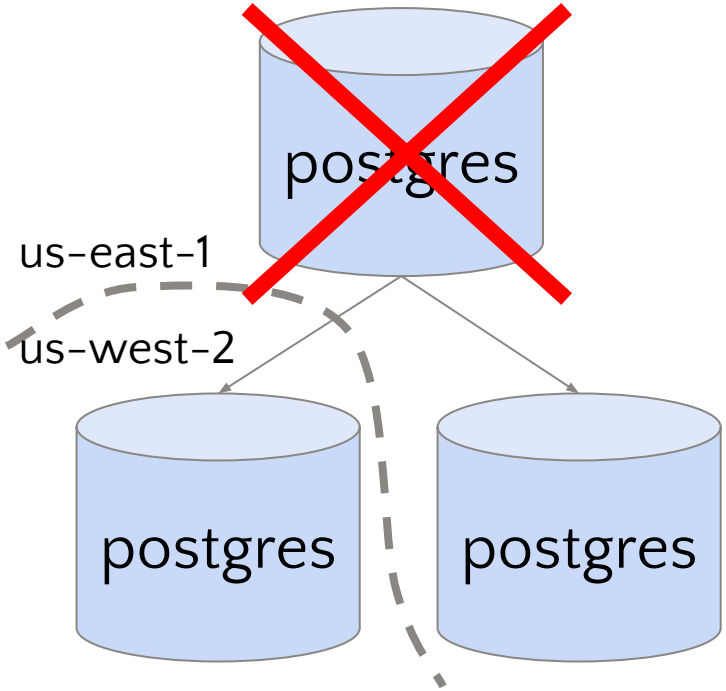
A What could go wrong: Several nodes and backups



What could go wrong: Several nodes and backups

S3 bucket

What now?



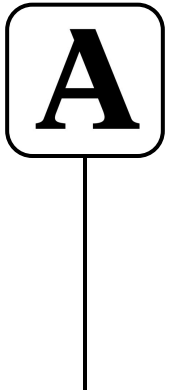
We do not care about backups

We care about restores

A What do we want with backups?

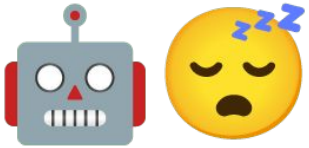
- **Allow us to “restore” the data layer to how it was**
  - Even if everything running postgres disappears
  - Restore what?
    - Everything that was ever written*
      - *(or as much as possible)
    - The recent past (e.g. the past 30-60 days)
- **Restores need to be fast enough to be useful**
  - Need to replace that node within hours, not days

Schrödinger's Backup:
"The condition of any backup is unknown until a restore is attempted."
—Spotlight on IT series #212, Spiceworks 2013



A Backup failures that I have witnessed in prod

1. Backups just weren’t happening

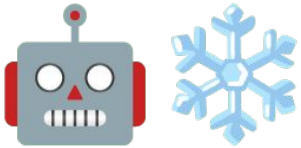


A Backup failures that I have witnessed in prod

1. Backups just weren't happening
2. "Successful" backups in s3 that are just an empty file

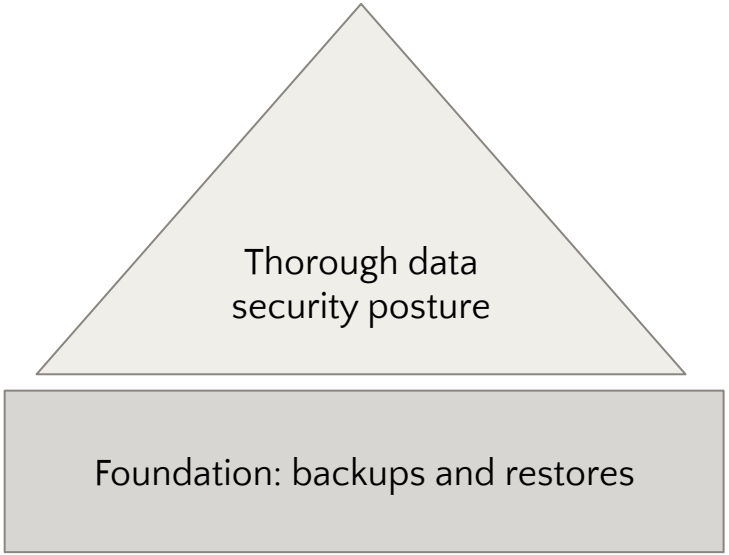
A Backup failures that I have witnessed in prod

1. Backups just weren't happening
2. "Successful" backups in s3 that are just an empty file
3. 😱 Looked good, but postgres never finished starting...



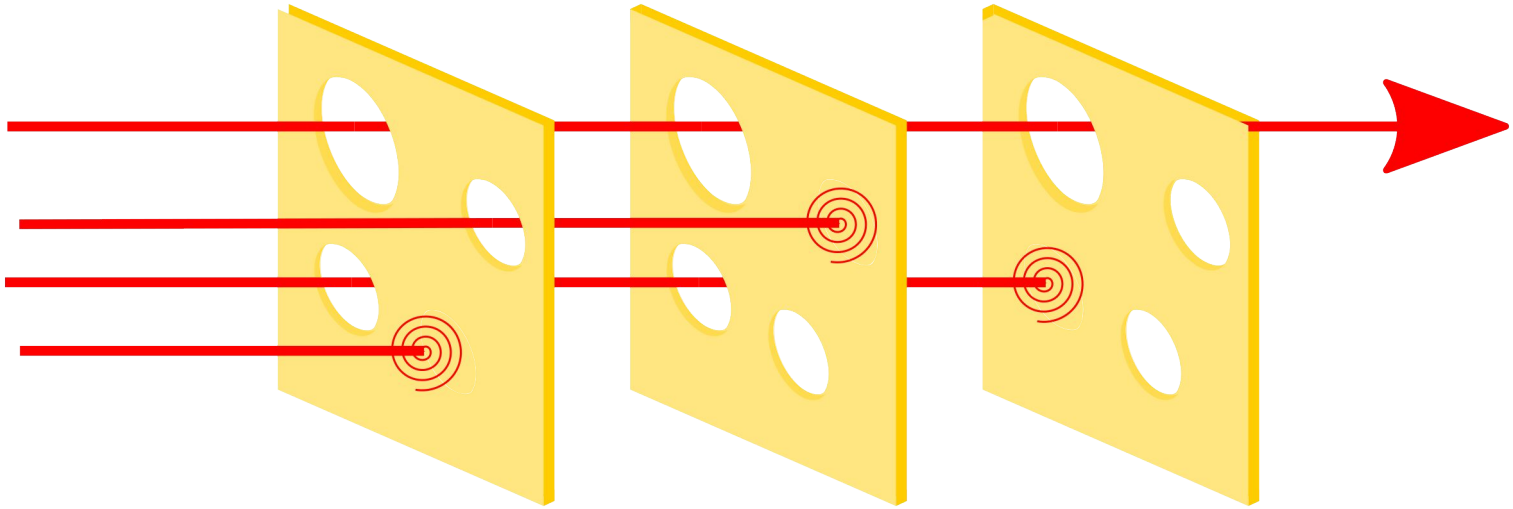
A What is out of scope for today?

- Ransomware
- Data corruption
- Insider threats



How do we test restores?

A “Swiss Cheese” model



A 🚨 Our strategy

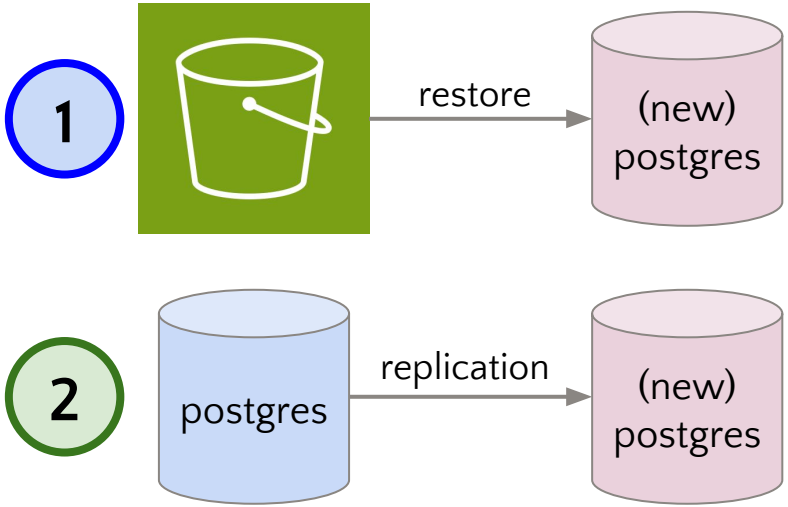
Business need: lots of read capacity

1. **Every time you need a new replica, use your backups**

➢ **When backups break:**

✓ You will notice

✓ Fixing will be a priority



A 🚨 Our strategy

Business need: prod-realistic data when testing

2. Bring up a copy of prod in a staging environment

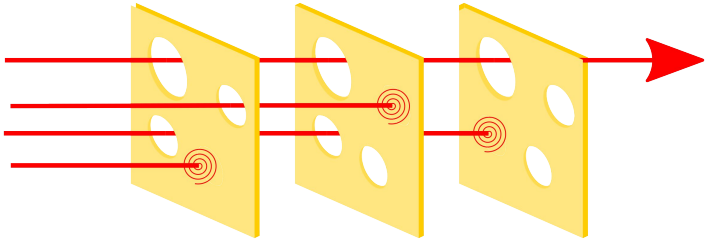
➢ Confirm that you can restore from nothing

A 🚨 Our strategy

1. Every time you need a new replica, use your backups
2. When you need to test, bring up a copy in staging

Everything else is an optimization*

*(optimizations are important too)



A How often?

0	->	once
once	->	yearly
yearly	->	monthly
		(etc)

A AWS Aurora/RDS

- **“One size does not fit all”**
- **We trust Amazon RDS to know what they’re doing***
  - *provided configuration is correct

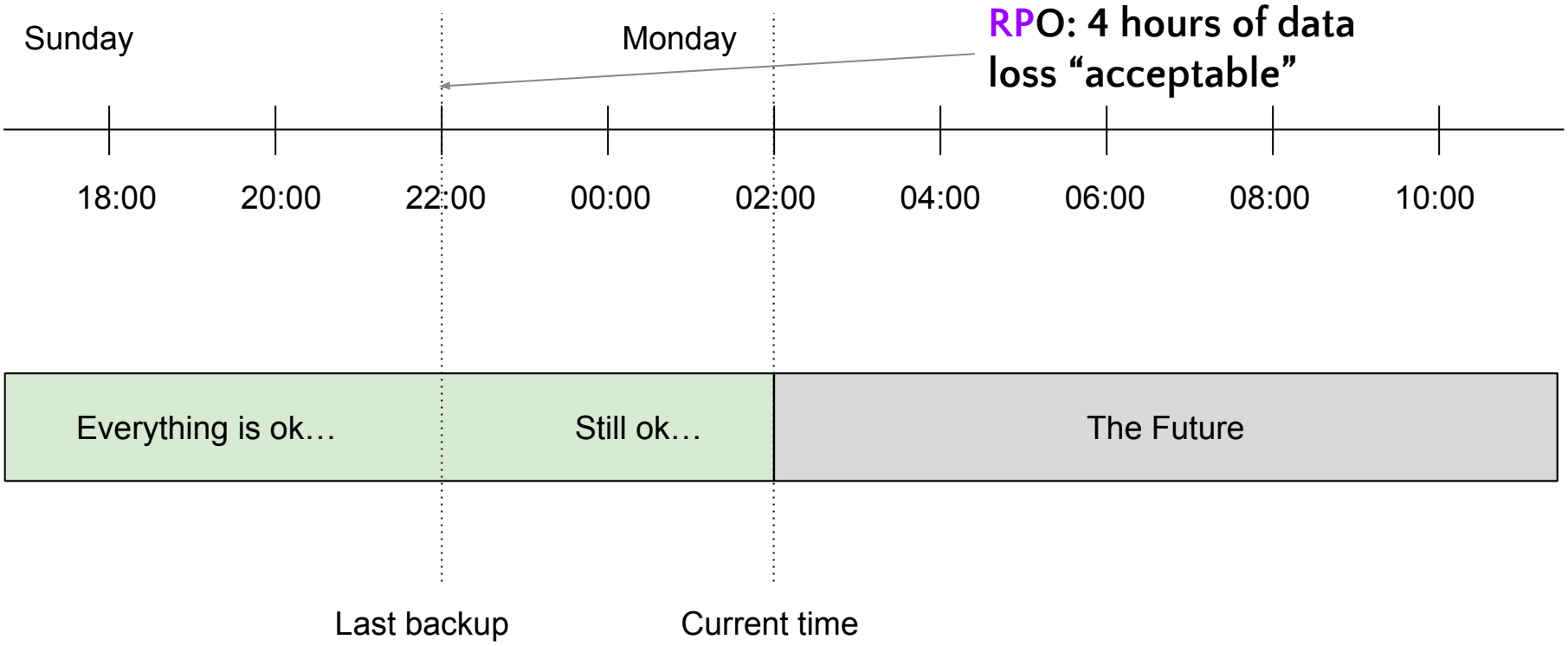
What goals should we set?

A

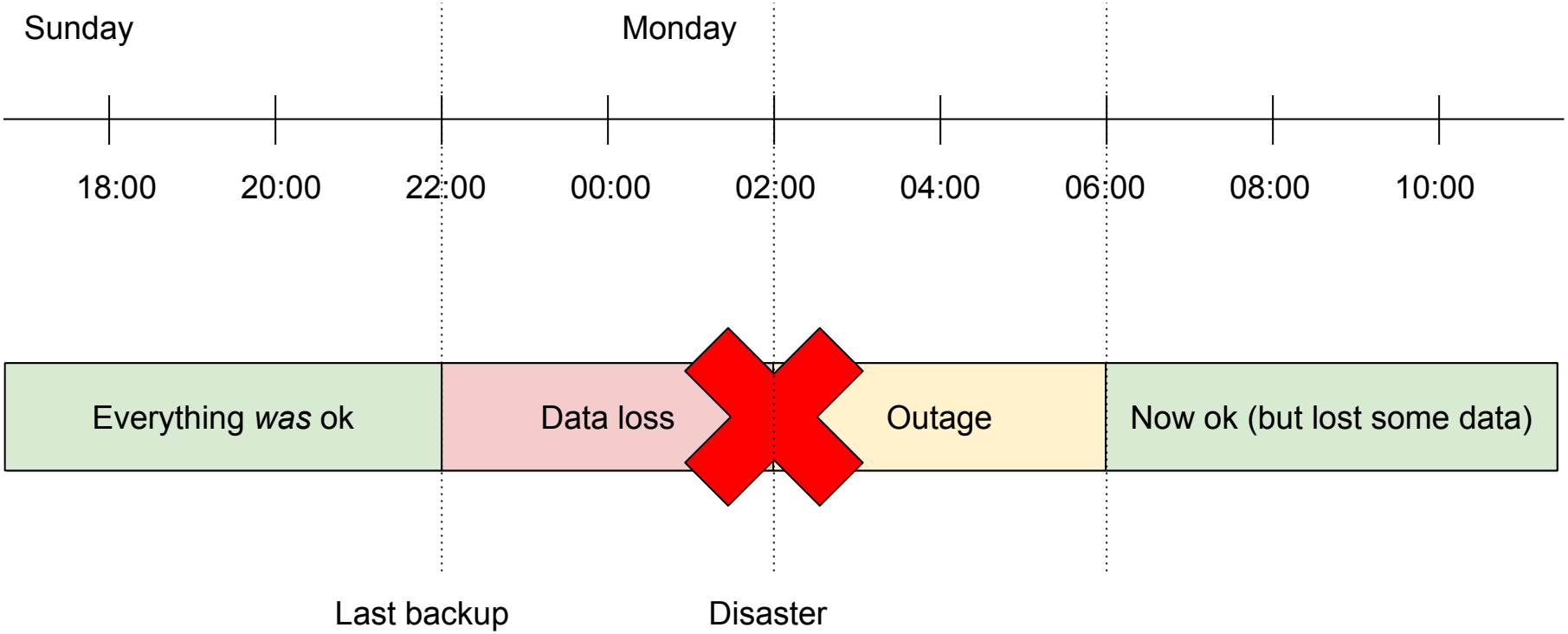
A RPO and RTO

- **How much data loss?**
  - Recovery Point Objective (RPO)
- **How long until we're back?**
  - Recovery Time Objective (RTO)

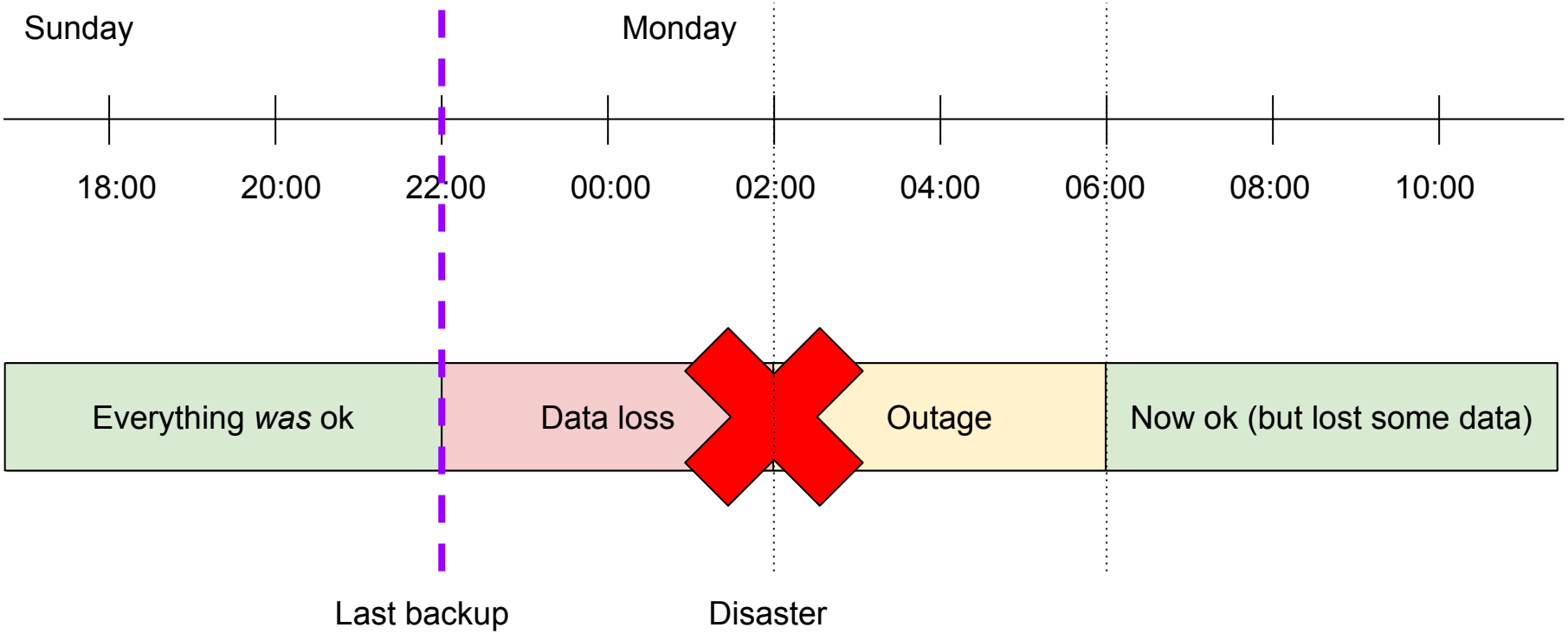
A Recovery Point Objective (RPO)



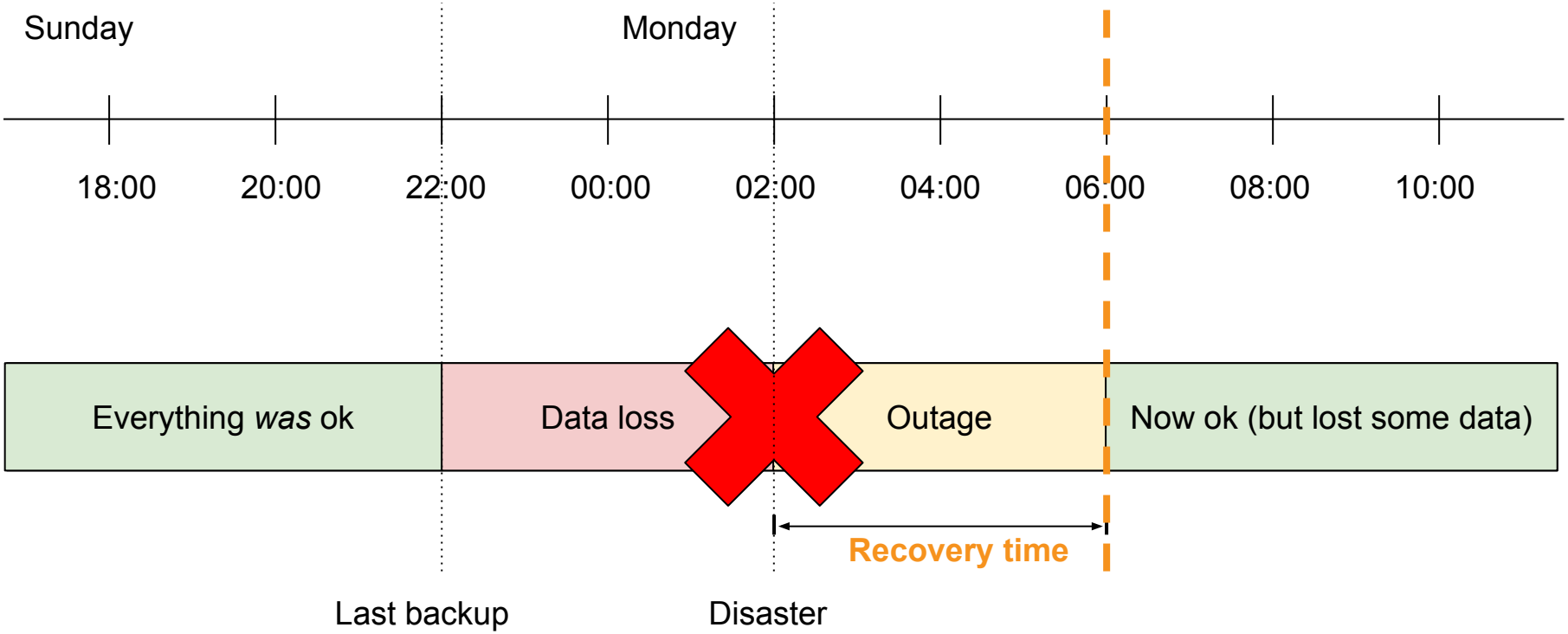
A Recovery Point Objective (RPO)



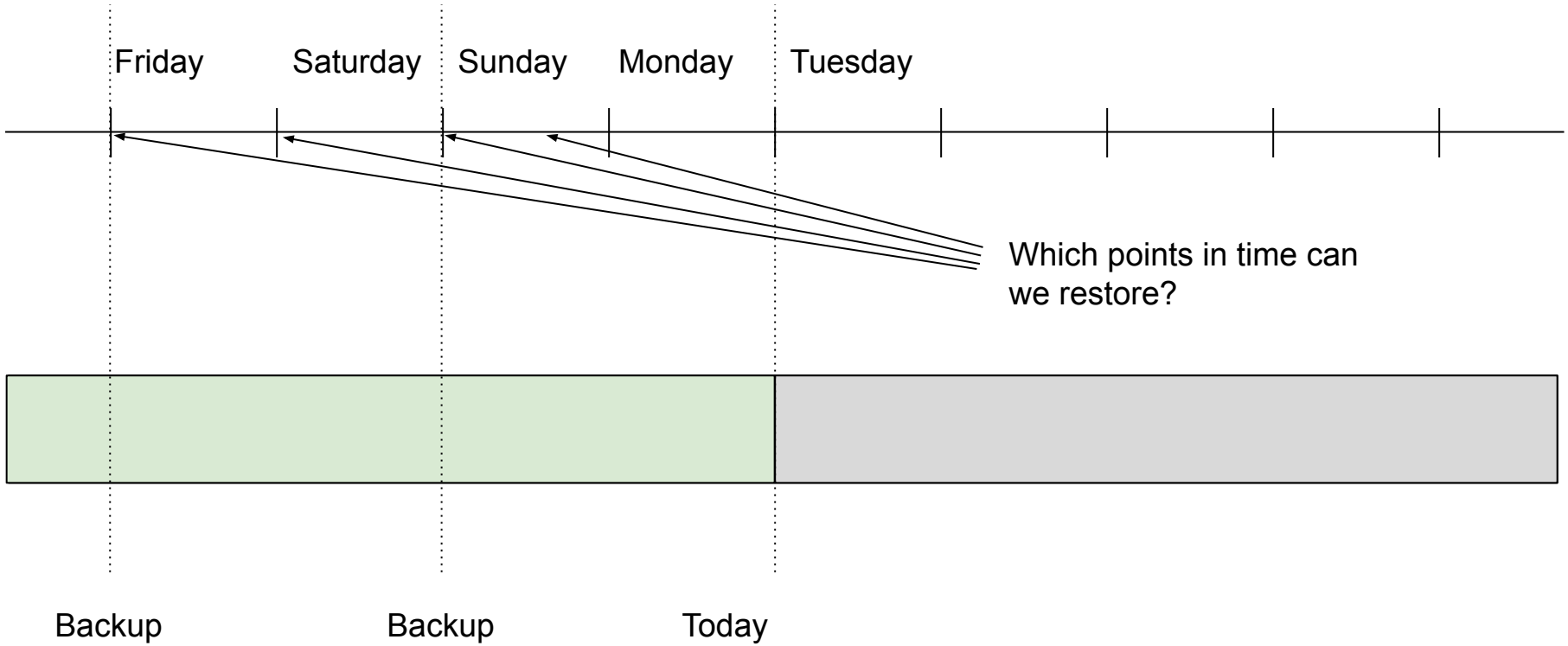
A Recovery Point Objective (RPO)



A Recovery Time Objective (RTO)



A Point-In-Time Recovery (PITR)



A Some of our specific numbers

- **For a 15 TB DB @ Academia:**

Objective	Target
Recovery Point	Everything*
Recovery Time	6 hours**
Point-in-time	1 month

* Allowance for several seconds to several minutes
** Multiply by 3 in full disaster (restore from nothing)

How do backups work?

A

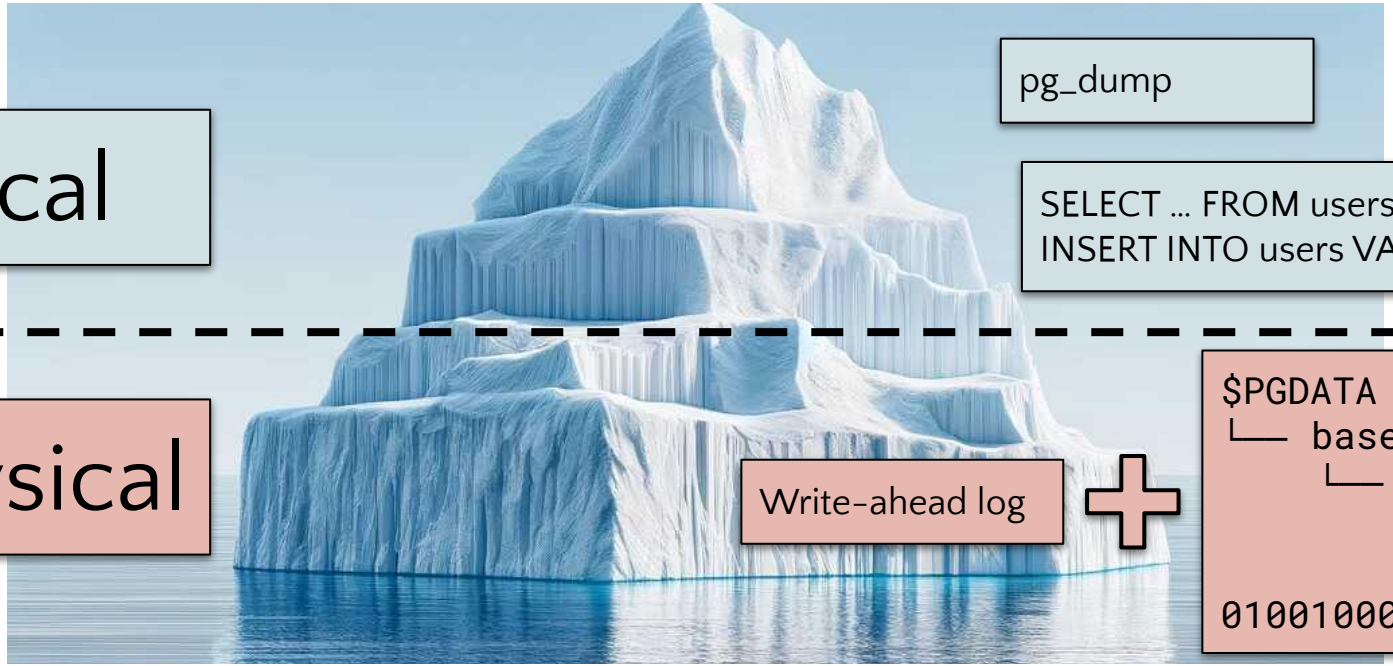
A

🚨 WARNING: Do not roll your own backup system

- It is very tricky to do it all yourself
- We will not go in depth enough to replicate:
  - pgBackRest
  - Barman
  - wal-g
  - etc

💡 Goal: Know how to test and monitor restores

A Physical vs Logical



logical

pg_dump

SELECT … FROM users WHERE …
INSERT INTO users VALUES …

physical

Write-ahead log

```
$PGDATA
└── base
    └── 16388
        └── 1247

0100100001101001…
```

A Write-Ahead Log (WAL)

INSERT INTO users(id, name)
VALUES (1, “Nick”);

A Write-Ahead Log (WAL)

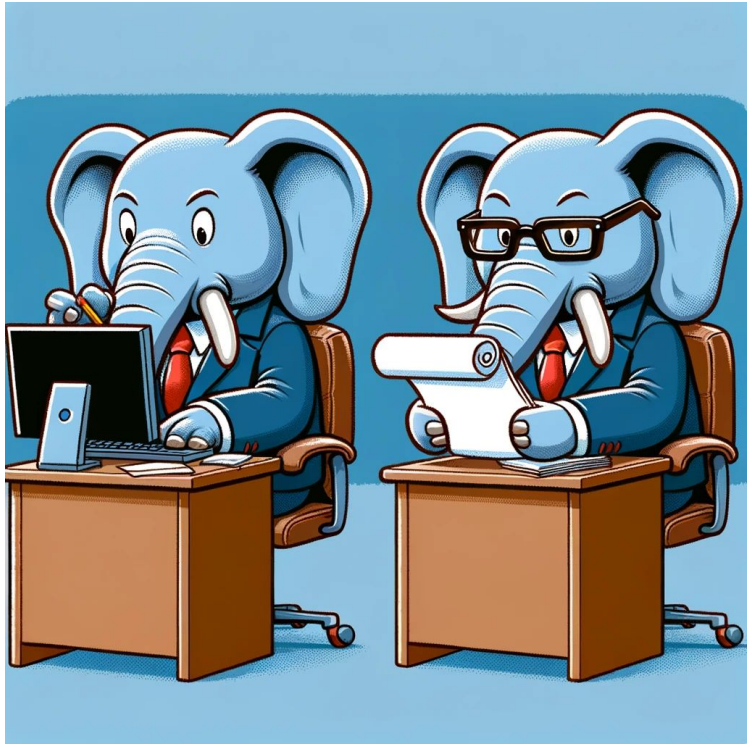
1 shared memory (shared_buffers)

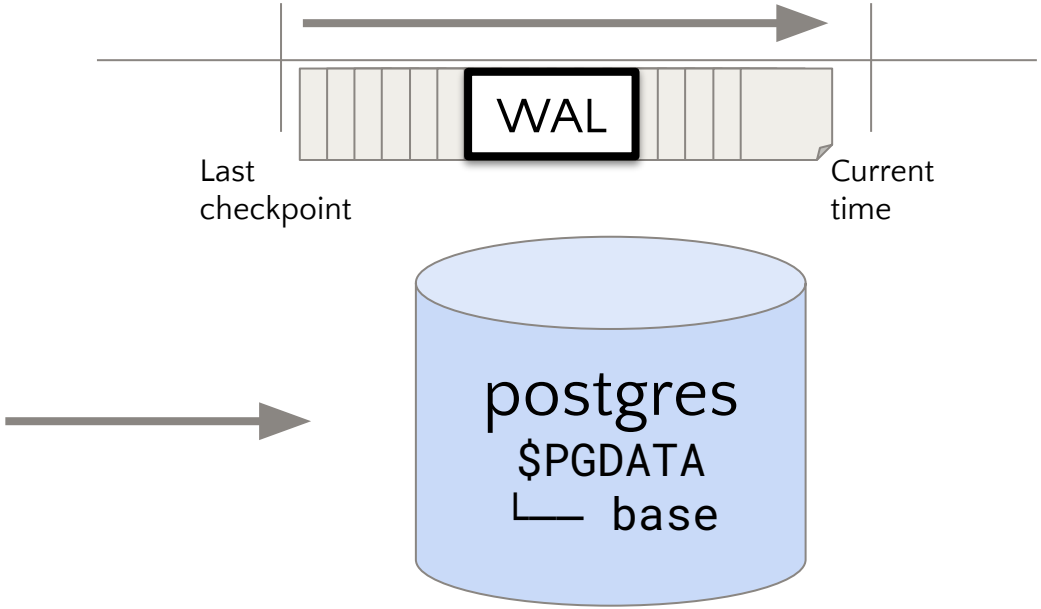
2 WAL file

3 To: client

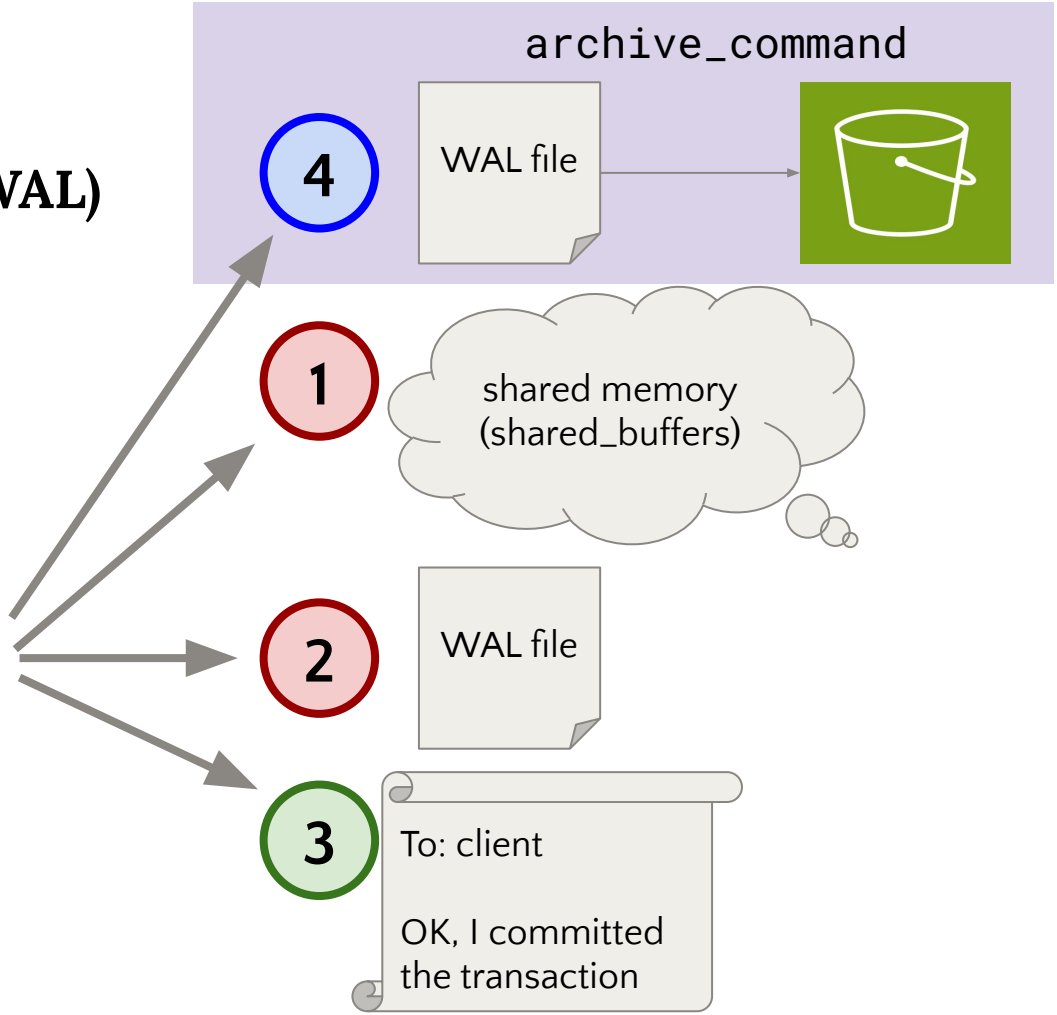
OK, I committed the transaction

A Write-Ahead Log (WAL) -> “Checkpointing”

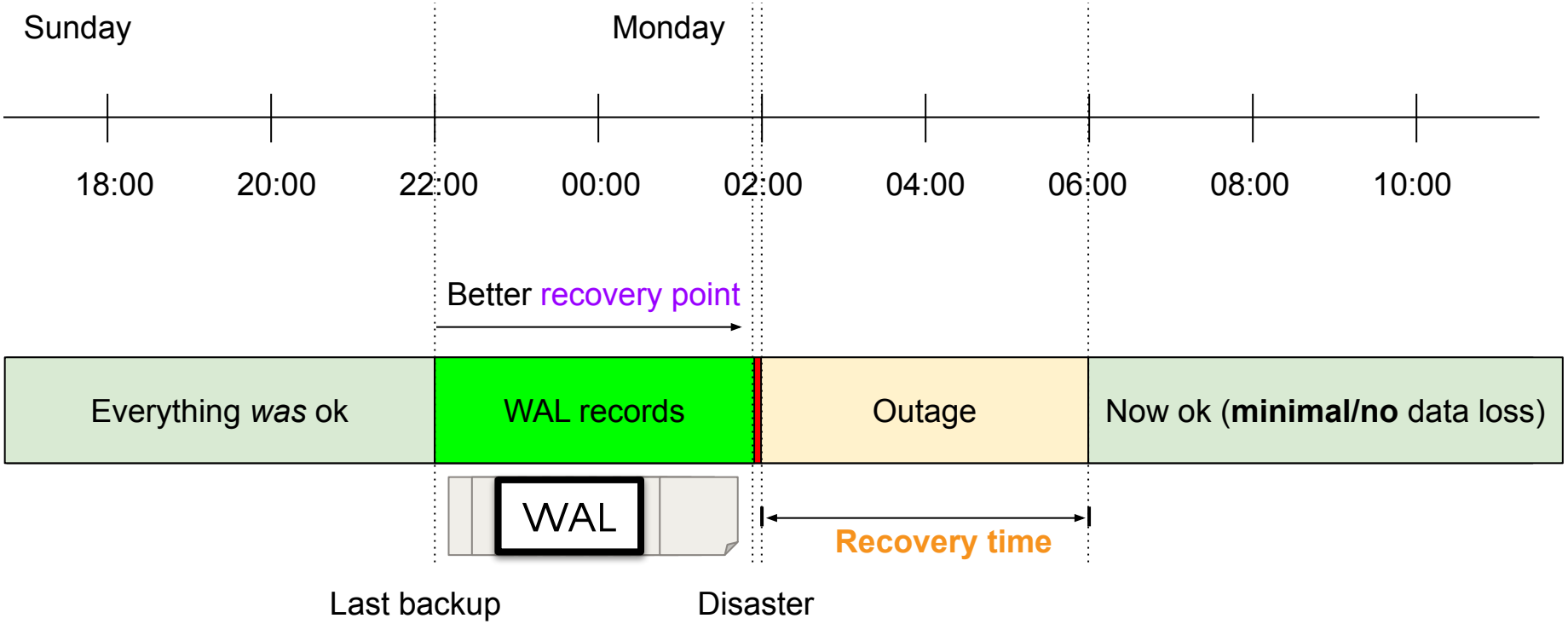




A Write-Ahead Log (WAL)

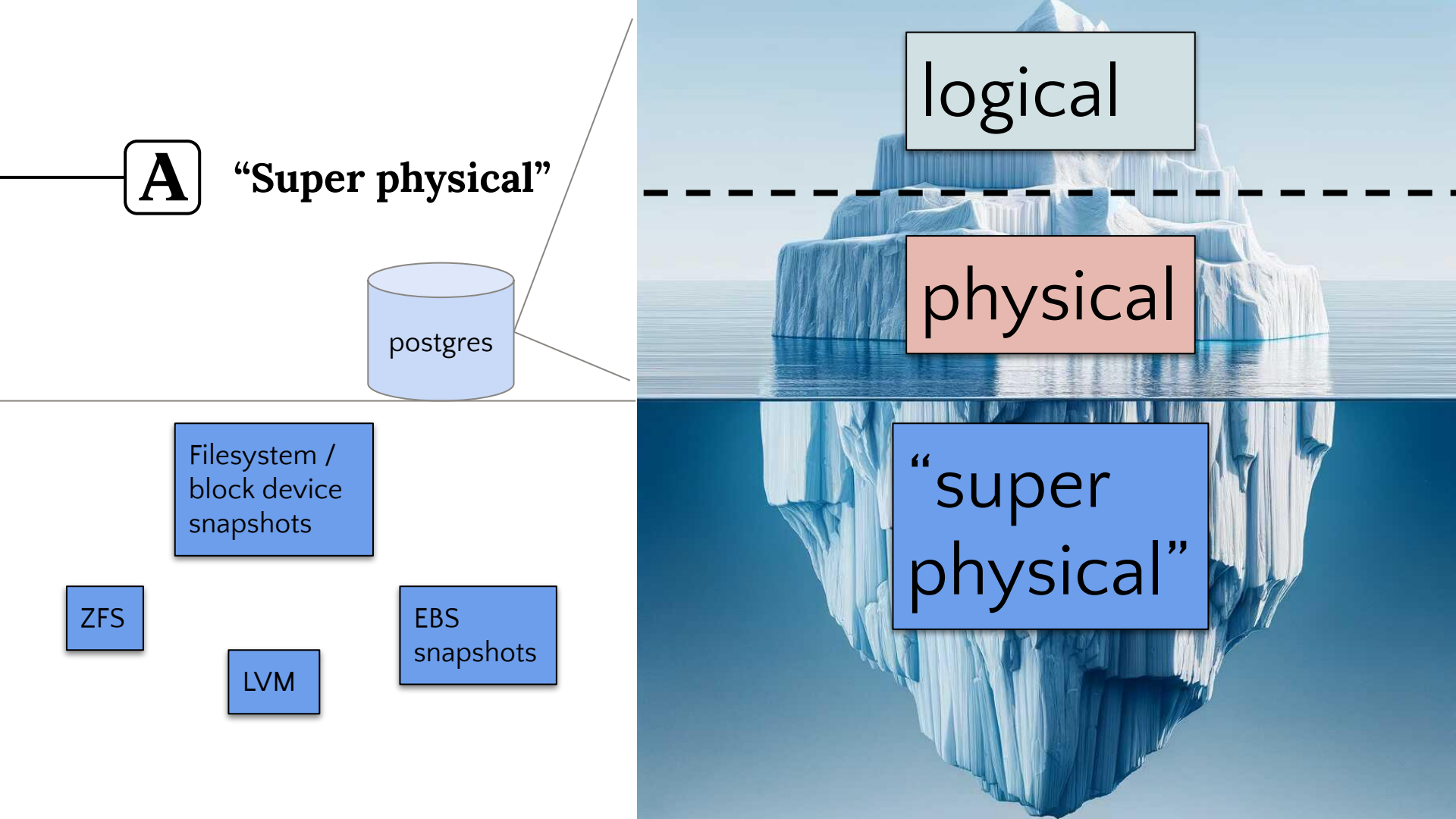


A Recovery Point Objective (RPO) with the WAL



A Physical vs logical

- Backups are faster, more frequent
- WAL => better RPO, continuous PITR
- Restores are faster => better RTO


A
“Super physical”
postgres
Filesystem / block device snapshots
ZFS
LVM
EBS snapshots
logical
physical
“super physical”

A Physical vs “super physical”

- “Super physical”: can use with e.g. MySql too
- Physical: Less fragile
  - `CREATE TABLESPACE …`
- Better postgres tooling for physical

Monitoring

A Good monitoring

Loud when it needs to be

A Great monitoring

Quiet the rest of the time

A How to notice when restores are failing?

- Alerts?
- Dashboards?

A Slack bot

Monday, January 29th

incoming-webhook APP 10:41 AM

Setup: i-0123456789abcdef0 being set up as Postgres::News in qa in us-east-1b

incoming-webhook APP 10:50 AM

Setup: Couldn't setup postgres-news-20240129-innocent-sam qa
i-0123456789abcdef0 Postgres::News

incoming-webhook APP 11:19 AM

Setup: i-fedcba987654321ff being set up as Postgres::News in qa in us-east-1b

incoming-webhook APP 11:29 AM

Setup: i-fedcba987654321ff now available as postgres-news-20240129-huffy-estate

A Slack bot

Monday, January 29th

incoming-webhook APP 10:41 AM
Setup: i-0123456789abcdef0 being set up as Postgres::News in qa in us-east-1b

incoming-webhook APP 10:50 AM
Setup: Couldn't setup postgres-news-20240129-innocent-sam qa
i-0123456789abcdef0 Postgres::News

incoming-webhook APP 11:19 AM
Setup: i-fedcba987654321ff being set up as Postgres::News in qa in us-east-1b

incoming-webhook APP 11:29 AM
Setup: i-fedcba987654321ff now available as postgres-news-20240129-huffy-estate

~10 minutes

A Slack bot (a prod example)

Today

incoming-webhook APP 3:06 PM

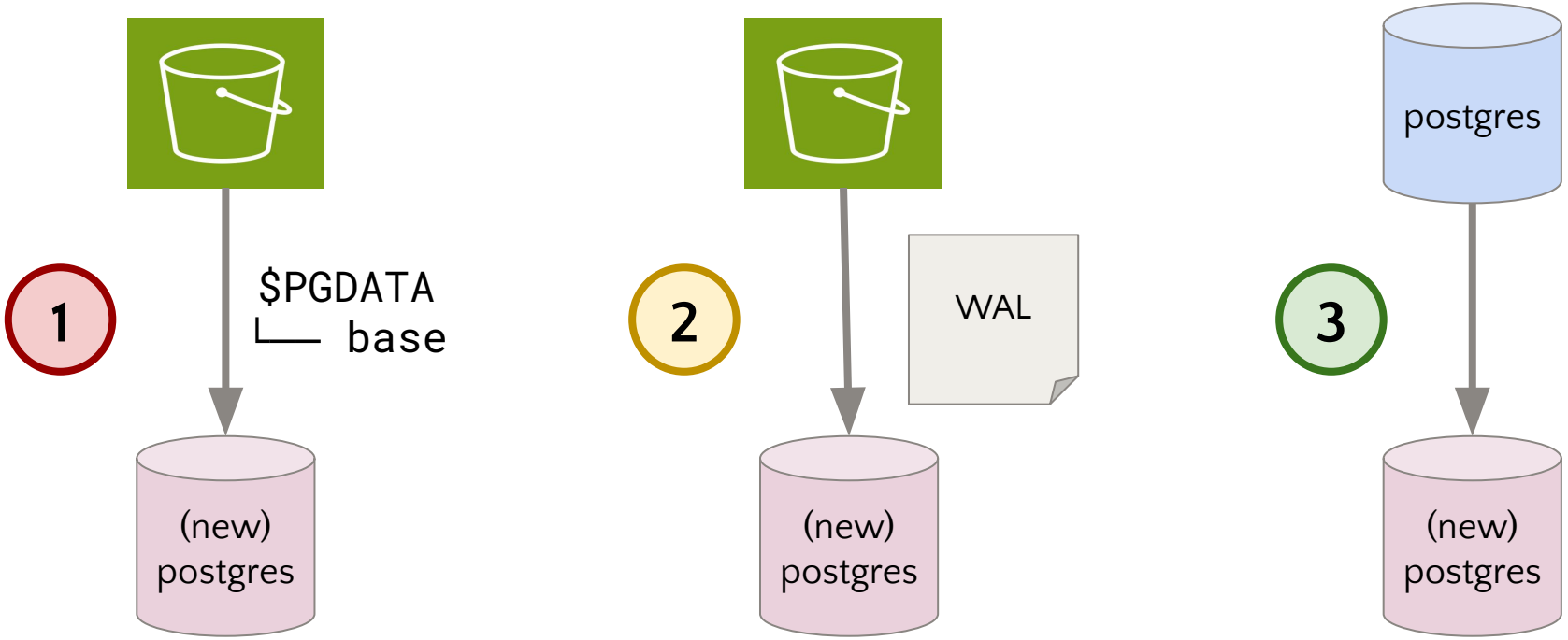
Setup: i-0123456789abcdef2 being set up as Postgres::News in production in us-east-1d

incoming-webhook APP 9:17 PM

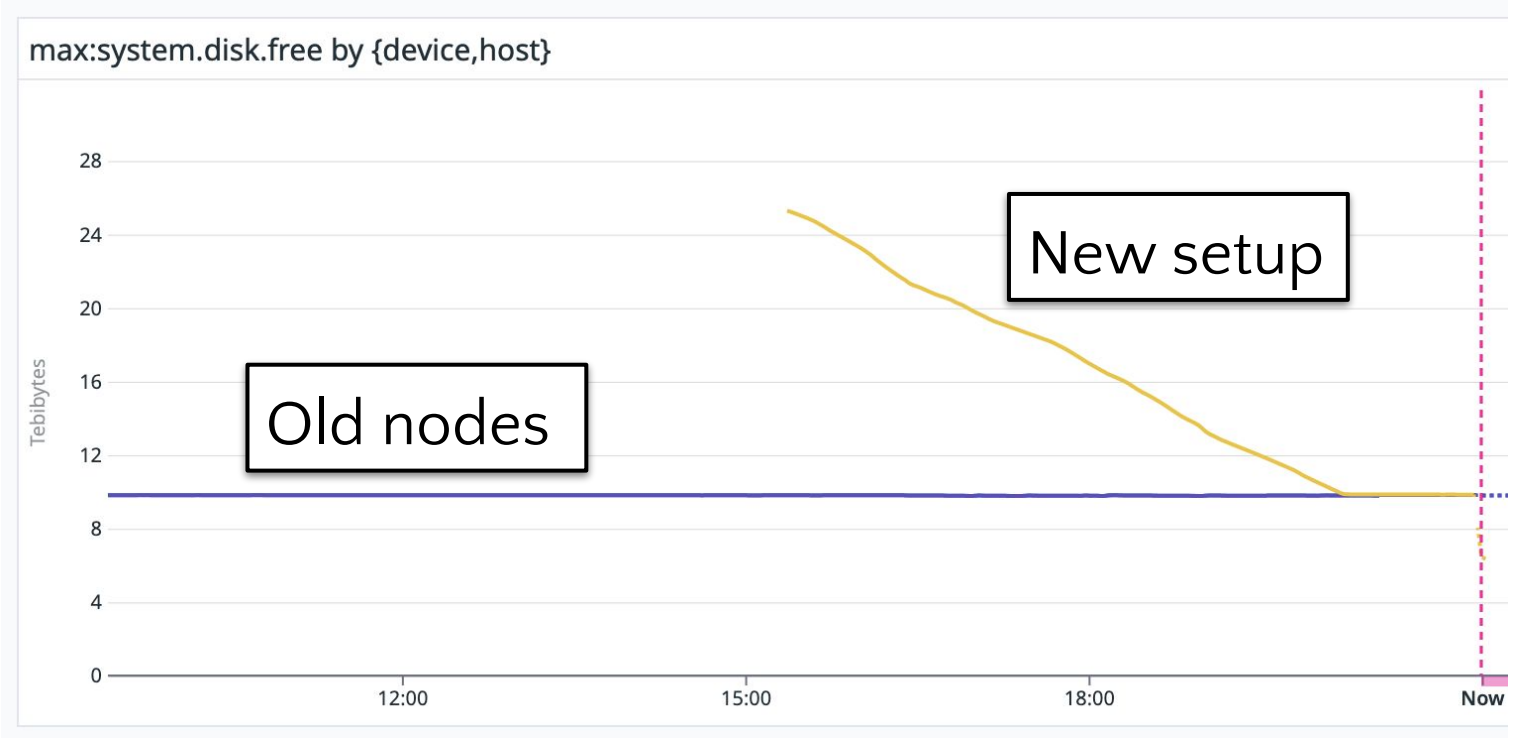
Setup: i-0123456789abcdef2 now available as postgres-news-20240314-status-absence

~6 hours

A Physical restore steps

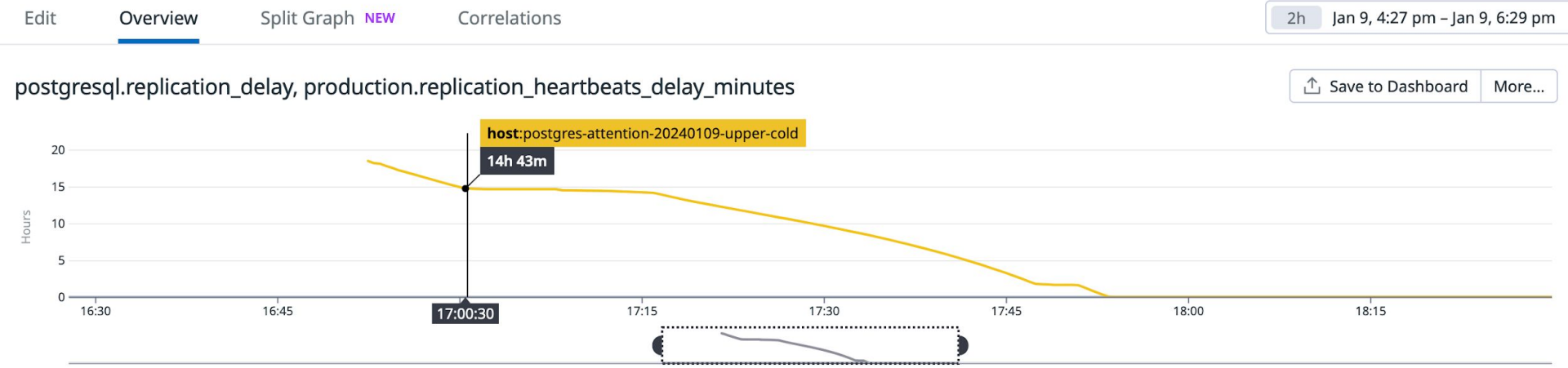


A Disk usage


max:system.disk.free by {device,host}
Tebibytes
28
24
20
16
12
8
4
0
12:00
15:00
18:00
Now
New setup
Old nodes

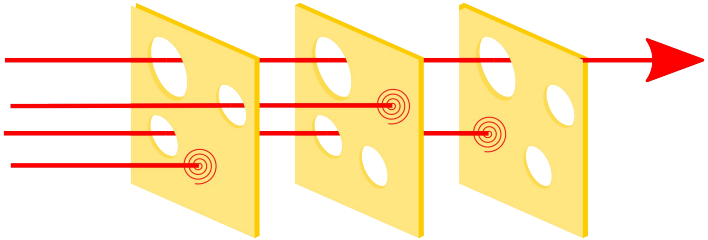
A Replication delay graph

- Look for the “catch up” slopes



A The search for leading indicators

- Restores are what we care about
- Broken restores = lagging indicator of broken backups
- Are there any **leading indicators** to monitor?



A pgbackrest info command: pipe to head

```
postgres@host $ pgbackrest info | head
stanza: news
   status: ok
   cipher: [value]

[...]
```

A pgbackrest info command: pipe to tail

```
postgres@host $ pgbackrest info | tail
[...]
        full backup: 20240309-181002F
            timestamp start/stop: 2024-03-09 18:10:02 /
2024-03-09 18:10:45
            wal start/stop: 00000002000003D1000000BB /
00000002000003D1000000BB
            database size: 2.9GB, database backup size: 2.9GB
            repo1: backup set size: 696.8MB, backup size: 696.8MB
```

A Check S3: is anything there?

Amazon S3 > Buckets > [bucket name]-us-east-1 > pgbackrest/ > news-15/ > backup/ > news/

news/

Objects | Properties

Objects (17) Info

Copy S3 URI | Copy URL | Download | Open | Delete | Actions

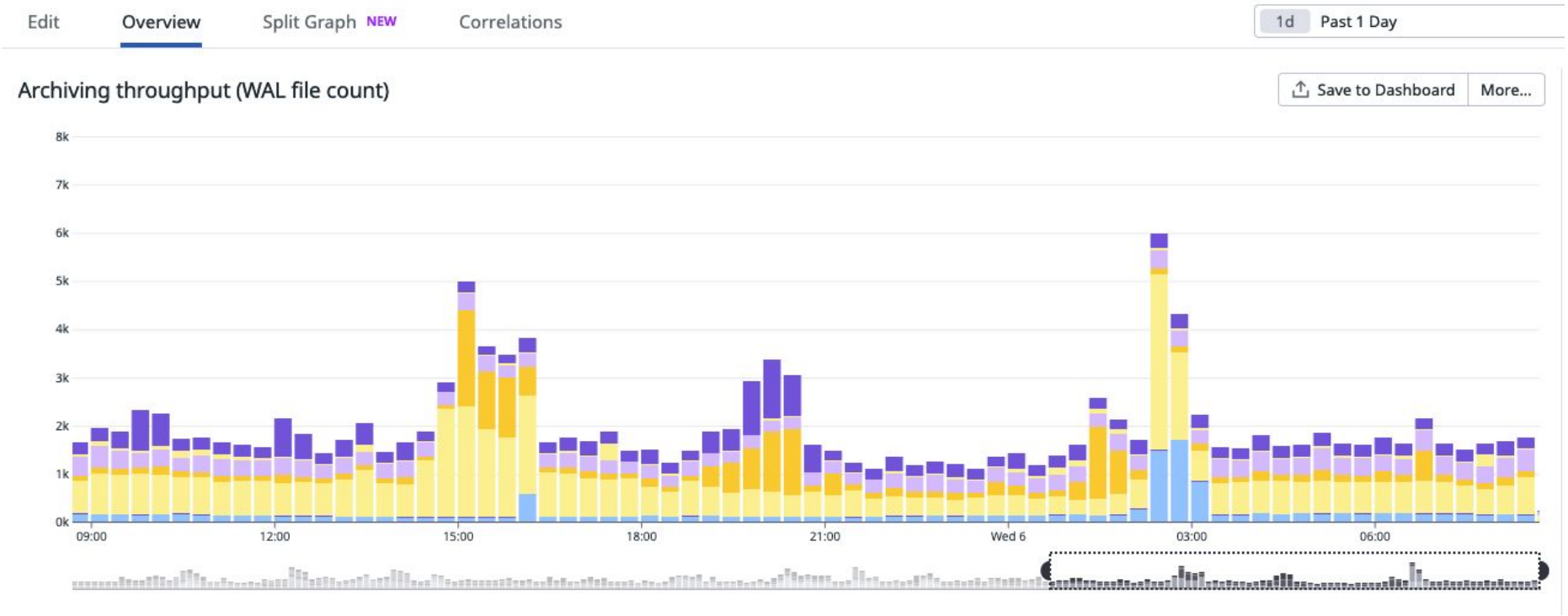
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissio

Find objects by prefix

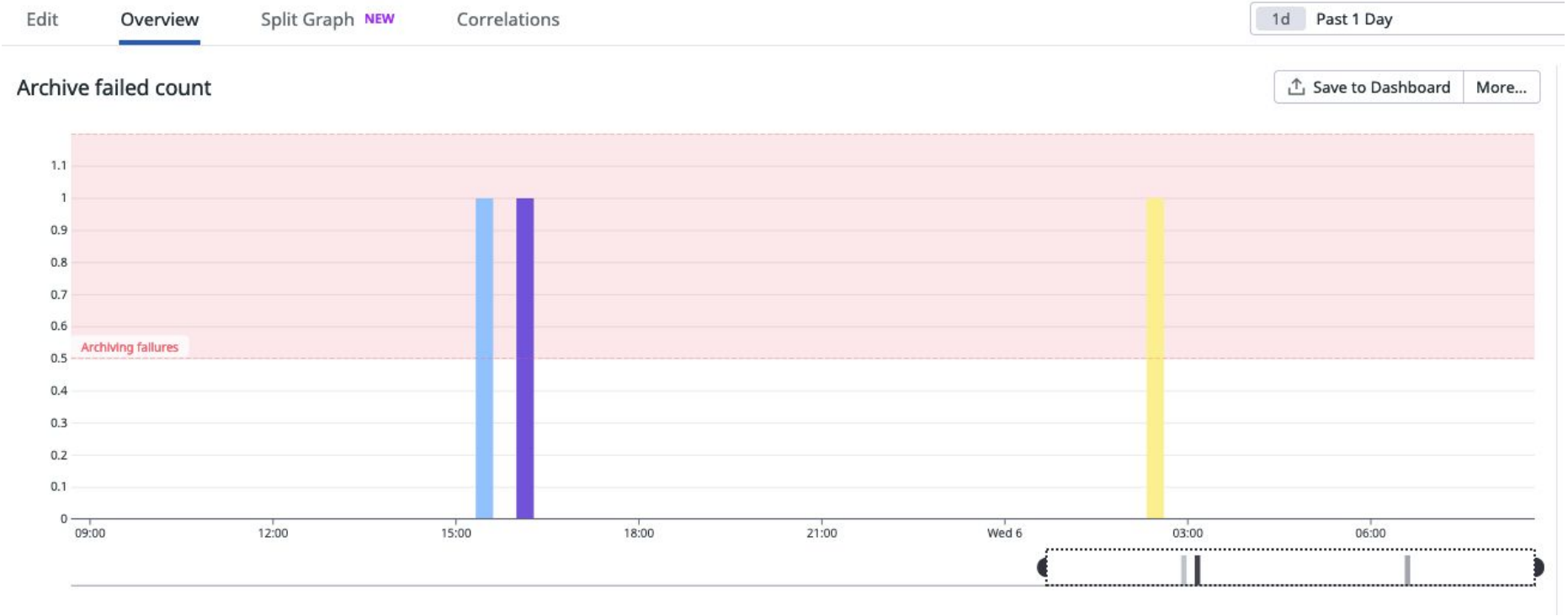
Show versions

Name	Type	Last modified	Size
20240203-181003F_20240206-181003I/	Folder	-	-
20240203-181003F_20240208-181002I/	Folder	-	-
20240203-181003F/	Folder	-	-

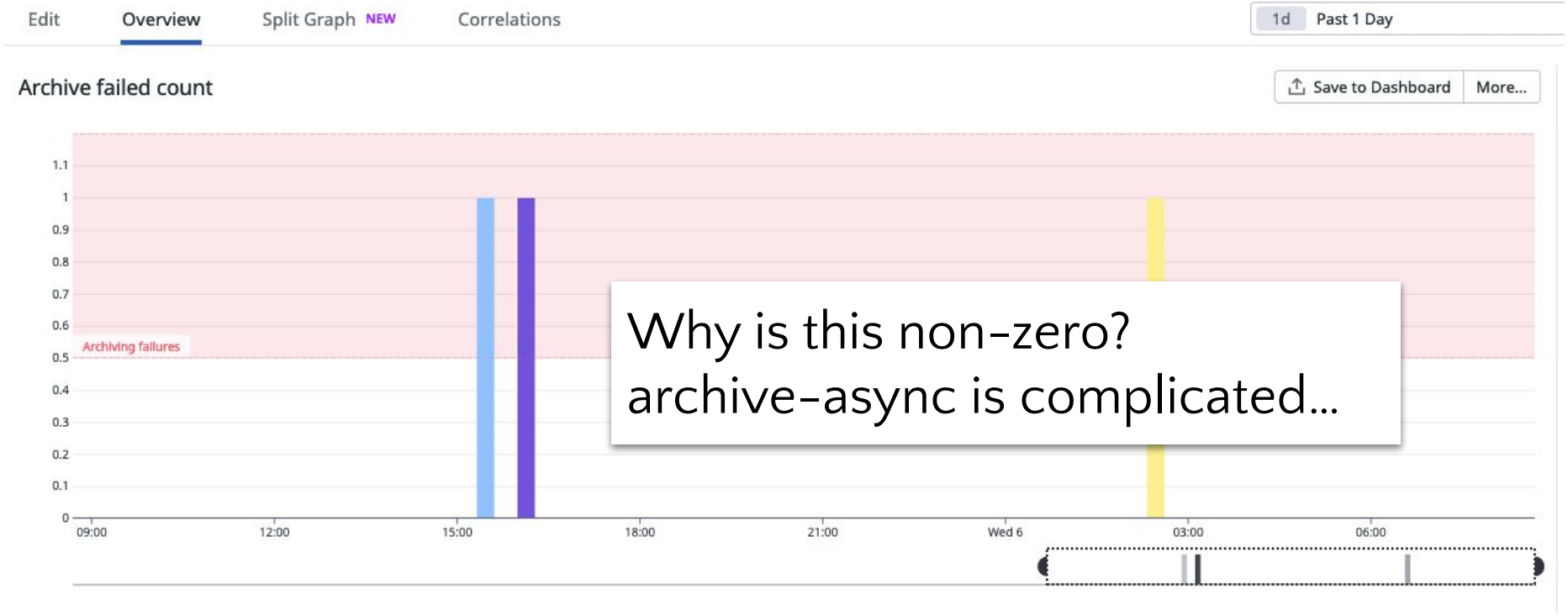
A WAL archiving stats: throughput, failures



A WAL archiving stats: throughput, failures



A WAL archiving stats: throughput, failures

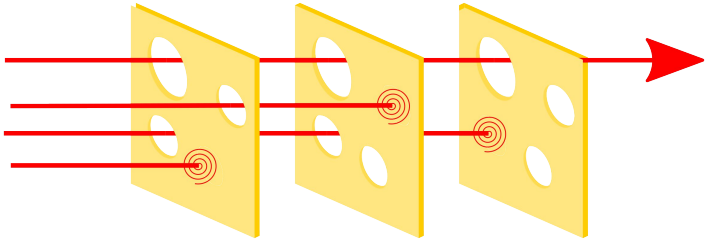


Why is this non-zero?
archive-async is complicated...

Recap

A Summary

- Every time you need a replica, use your backups
- Periodically test a cold-restore in QA/staging
- Visualize the restore process
- Make sure your monitoring pulls its weight



A Acknowledgements

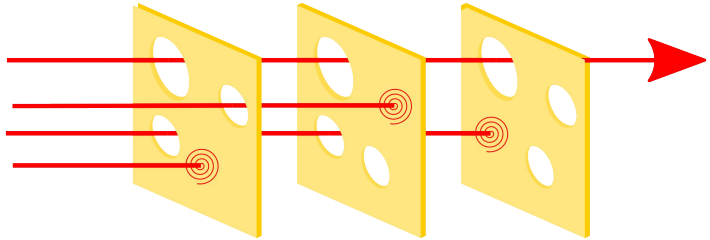
- Academia.edu
- My team
- Michael, Founder of pgMustard

A Summary

- Every time you need a replica, use your backups
- Periodically test a cold-restore in QA/staging
- Visualize the restore process
- Make sure your monitoring pulls its weight

Questions?

https://github.com/aristocrates



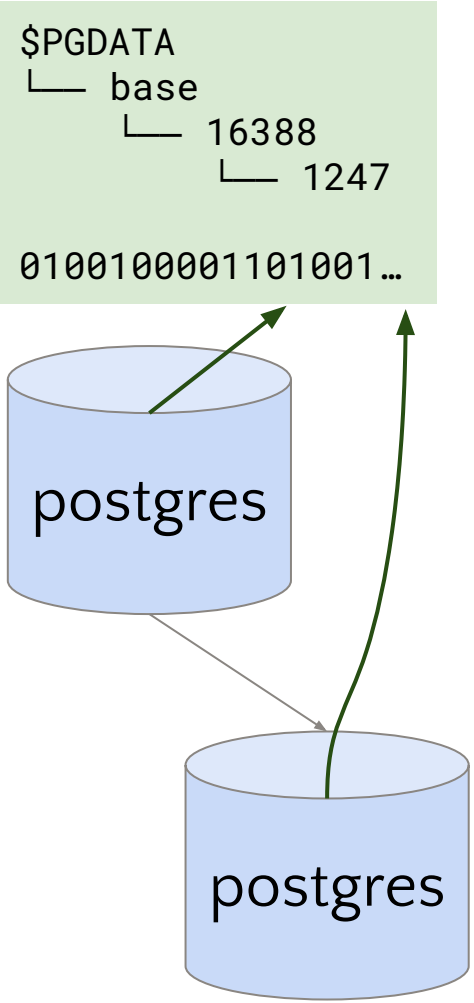
Appendix

A

(There's definitely no time for this, but if you're reading this after the conference, enjoy!)

A Sidenote: streaming replication

- **This talk assumes some familiarity with:**
  - Streaming replication in postgres
    - "Binary compatibility"
    - Read-only replicas, HA replicas
  - The Write Ahead Log (WAL)
    - (at a high level)
- **Some resources:**
  - pgBackRest User Guide
  - Dude, where's my byte? | SCaLE 17x
    - (recording, youtube)



“Replication heartbeats”

A `replication_heartbeats`

- **Sometimes the built in Datadog metric has issues**
  - (Not always recognized until the first time a replica catches up)
- **So we have a secondary system to fill in the gaps**

A replication_heartbeats

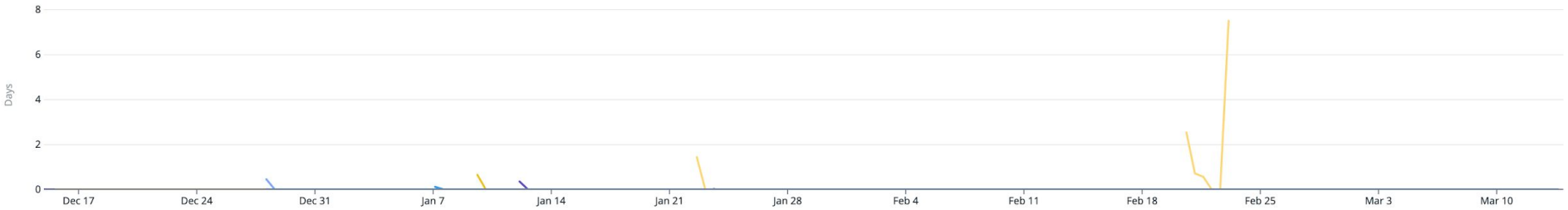
```
CREATE TABLE public.replication_heartbeats (
  created_at TIMESTAMP WITHOUT TIME ZONE PRIMARY KEY DEFAULT now()
);
```

- **Cron job to insert the current time**
- **Metric: diff against replica system time**
- **Sloppiness aside…**
  - time zones
  - NTP point of failure
- **… it works pretty well in practice**

A replication_heartbeats

Edit | Overview | Split Graph NEW | Correlations

3mo Past 3 Months



1 Select your visualization

Timeseries | Query Value | Table | Heatmap | Scatter Plot | Distribution | Top List | Host Map | Change | Geomap | Tree Map | Pie Chart

2 Graph your data

Edit | JSON | Share

Graphing help

a Metrics postgresql.replication_delay from server_type:postgres , $stage , $db , $postgres_cluster max by host ×

b Metrics production.replication_heartbeats_delay_minutes from server_type:postgres , $stage , $db , $postgres_cluster max by host ×

A replication_heartbeats

Edit | Overview | Split Graph NEW | Correlations

3mo Past 3 Months

1 Select your visualization

Timeseries | Query Value | Table | Heatmap | Scatter Plot | Distribution | Top List | Host Map | Change | Geomap | Tree Map | Pie Chart

2 Graph your data

Edit | JSON | Share

Graphing help

a Metrics postgresql.replication_delay from server_type:postgres , $stage , $db , $postgres_cluster max by host ×

b Metrics production.replication_heartbeats_delay_minutes from server_type:postgres , $stage , $db , $postgres_cluster max by host ×

Costs

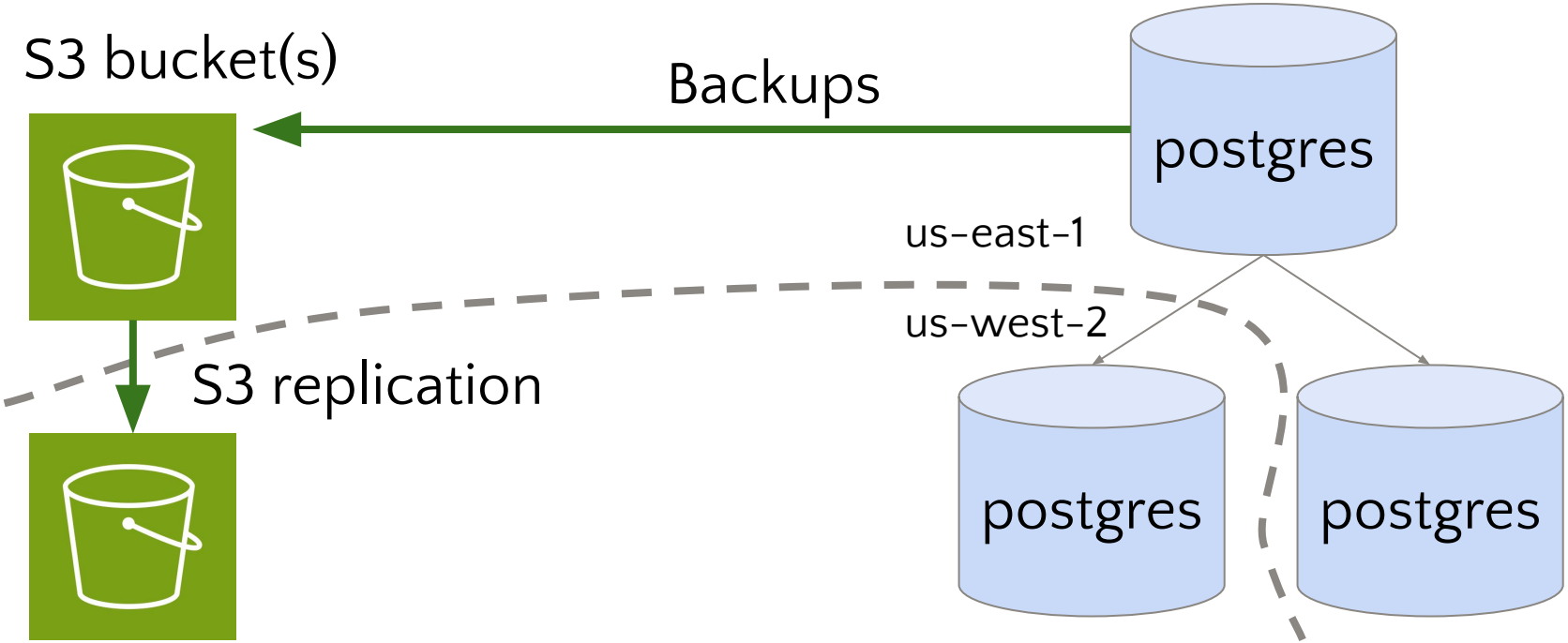
A Amazon EC2 + pgBackRest + Amazon S3

- **We have different business divisions**
  - Each with own data $ budgets
- **Our tagging scheme:**
  - postgres_cluster
  - business_area
  - function (e.g. paper_recommendations)
  - cost_owner (e.g. a team)

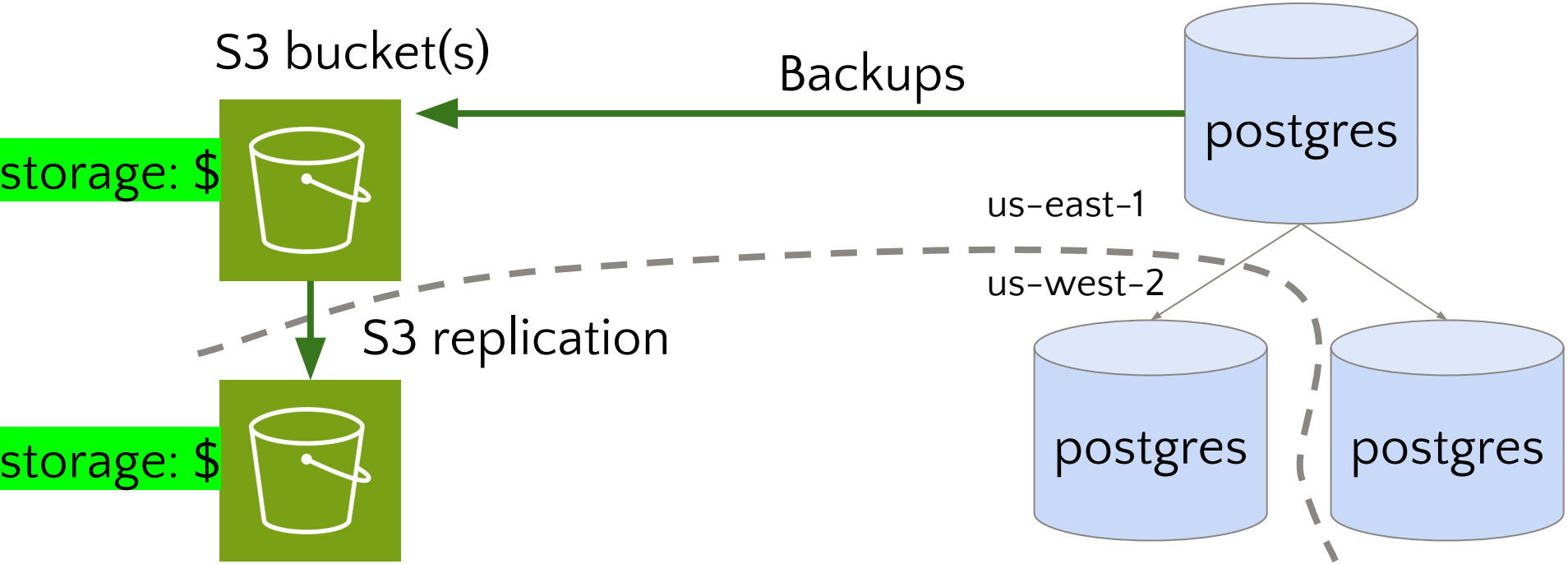
A What costs money in a backup system? (AWS and similar)

- Storage
- Network (sometimes)
- Other

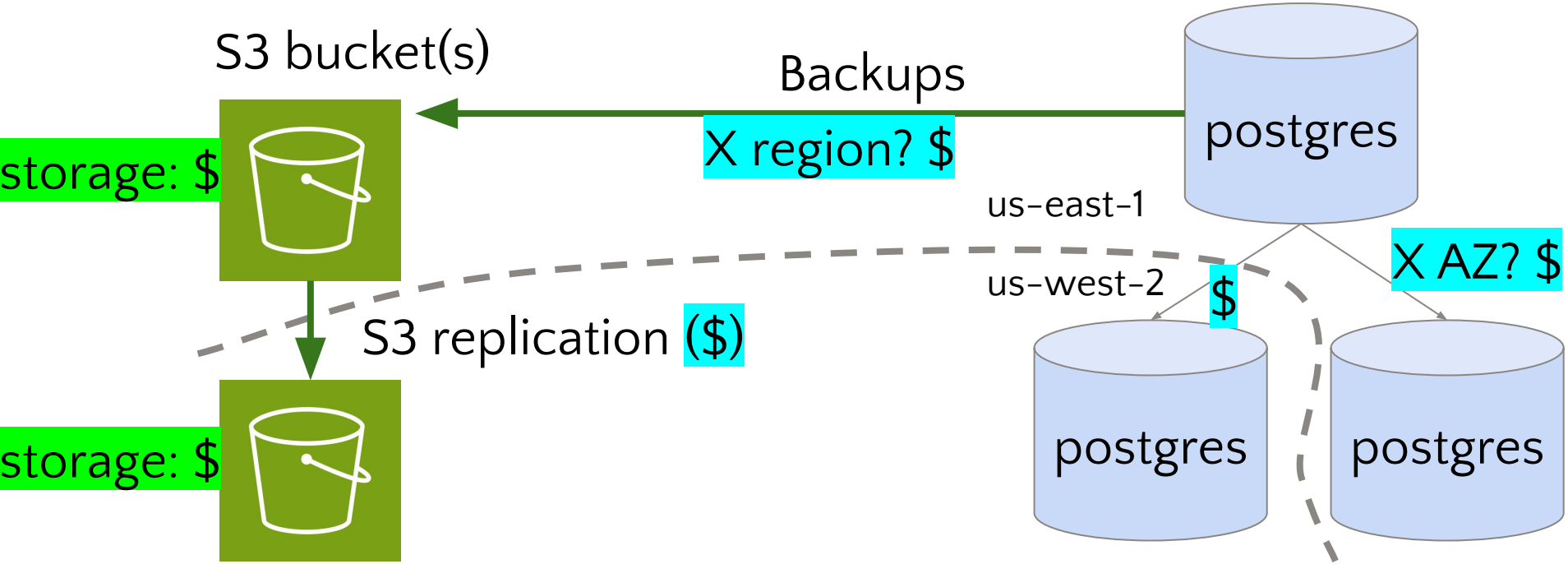
A Where are the costs?



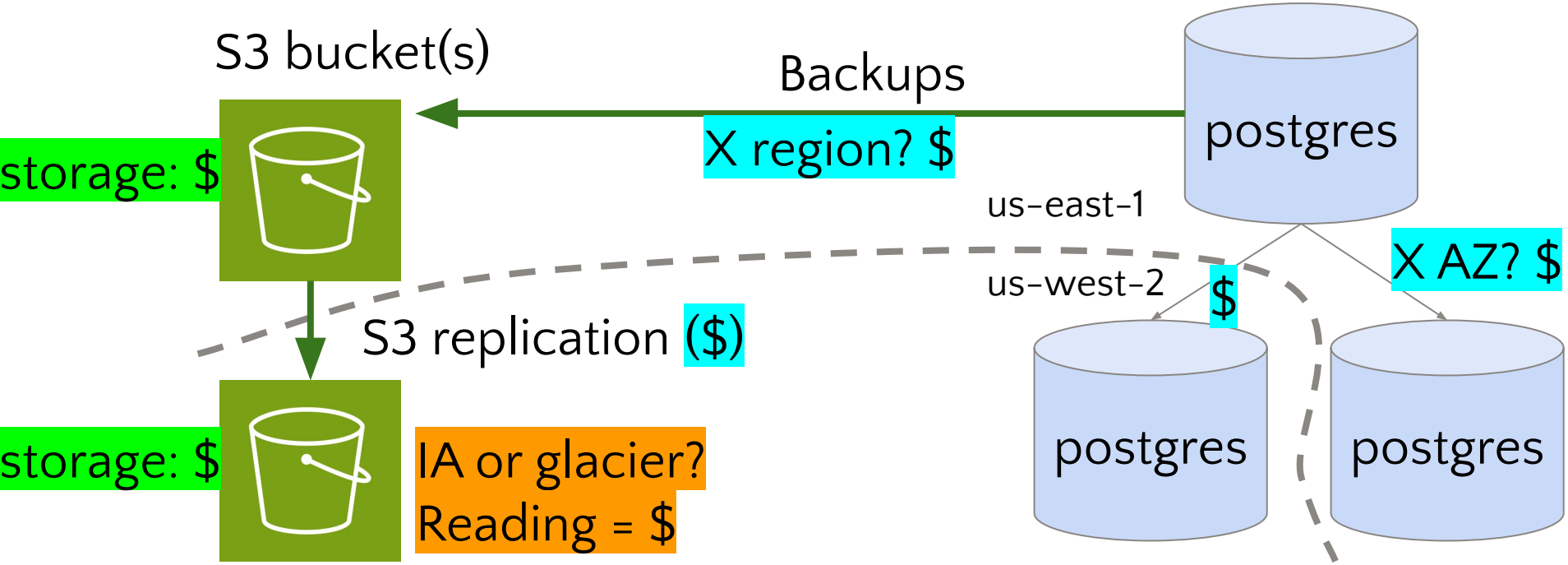
A Where are the costs?



A Where are the costs?



A Where are the costs?



pgBackRest: info command

```
stanza: news
      status: ok
      cipher: [value]

      db (current)
      wal archive min/max (15): 000000020000036F00000050/00000002000003C800000026

      full backup: 20240203-181003F
            timestamp start/stop: 2024-02-03 18:10:03 / 2024-02-03 18:10:48
            wal start/stop: 000000020000036F00000050 / 000000020000036F00000050
            database size: 2.9GB, database backup size: 2.9GB
            repo1: backup set size: 696.8MB, backup size: 696.8MB
```

A S3: calculate storage used

news/

Objects | Properties

Objects (17) Info

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly

Find objects by prefix

Show versions

Name	Type	Last modified	Size
20240203-181003F_20240206-181003I/	Folder	-	
20240203-181003F_20240208-181002I/	Folder	-	
20240203-181003F/	Folder	-	

Actions:
- Download as
- Share with a presigned URL
- Calculate total size
- Copy
- Move
- Initiate restore
- Query with S3 Select
- **Edit actions**
  - Rename object
  - Edit storage class

A S3: calculate storage used

Calculate total size Info

The information below will no longer be available after you navigate away from this page.

Summary

Source	Total number of objects	Total size
s3://[bucket name]/pgbackrest/news-15/backup/news/	1,309	697.2 MB

Specified objects

Find objects by name

Name	Type	Last modified	Size	Total number of objects
20240203-181003F/	Folder	-	697.2 MB	1309