Upgrading the Mammoth

Crafting the Best Path to Modernize your Database

Presented by Brian Fehrle

INTRODUCTION

Speaker: Brian Fehrle

PostgreSQL DBA and Data Architect since 2008

Started working on 8.3, quickly upgrading clients to 8.4

In 2020 - I witnessed mammoths being born

COMMAND PROMPT, INC.

AGENDA

- Introduction and Agenda
- What is a Mammoth?
- Preparing for the Big Jump
- Asking the Right Questions
- *"**Yeah, but…**"*
- Let's Get Clever!
- Real World Examples
- Q&A

COMMAND PROMPT, INC.

WHAT IS A MAMMOTH?

- How you got there
- End of life status
- Security issues that won’t be fixed
- Missing out on new features and performance improvements

COMMAND PROMPT, INC.

End of Life Schedule

https://www.postgresql.org/support/versioning/

Releases

Version	Current minor	Supported	First Release	Final Release
17	17.4	Yes	September 26, 2024	November 8, 2029
16	16.8	Yes	September 14, 2023	November 9, 2028
15	15.12	Yes	October 13, 2022	November 11, 2027
14	14.17	Yes	September 30, 2021	November 12, 2026
13	13.20	Yes	September 24, 2020	November 13, 2025
12	12.22	No	October 3, 2019	November 21, 2024
11	11.22	No	October 18, 2018	November 9, 2023
10	10.23	No	October 5, 2017	November 10, 2022
9.6	9.6.24	No	September 29, 2016	November 11, 2021
9.5	9.5.25	No	January 7, 2016	February 11, 2021
9.4	9.4.26	No	December 18, 2014	February 13, 2020

COMMAND PROMPT, INC.

ASKING THE RIGHT QUESTIONS

Before your upgrade, know what questions to ask:

- How big is the database?
- How complex is it?
  - What are the access patterns across the data.
- What is its overall environment? (replicas, backups, clones, etc)
- What ***downtime*** is allowed, and when?
- What ***rollback*** requirements do you have?
- What is your infrastructure?
  - Will you be CHANGING your infrastructure?

COMMAND PROMPT, INC.

Preparing for the BIG JUMP

PREPARING FOR THE BIG JUMP

- What to look out for on such a big jump:
  - FEATURE MATRIX
  - Backup options may have changed
  - Replication may have changed
  - Table Partitioning is different
  - CTE’s are different with version 12
  - Learn JIT!
- Test Test Test Test!!!!

COMMAND PROMPT, INC.

FEATURE MATRIX

https://www.postgresql.org/about/featurematrix/

Partitioning & Inheritance

	17	9.6
Accelerated partition pruning	Yes	No
Declarative table partitioning	Yes	No
Default Partition	Yes	No
Foreign Key references for partitioned tables	Yes	No
Foreign table inheritance	Yes	Yes
Partitioning by a hash key	Yes	No
Partition pruning during query execution	Yes	No
Support for PRIMARY KEY, FOREIGN KEY, indexes, and triggers on partitioned tables	Yes	No
Table Partitioning	Yes	Yes
UPDATE on a partition key	Yes	No

COMMAND PROMPT, INC.

BACKUP OPTIONS MAY HAVE CHANGED

- PGBackrest is a solid enterprise level backup option
- pg_start_backup() replaced by pg_backup_start() in pg15
  - This **will** break old backup scripts / processes

COMMAND PROMPT, INC.

REPLICATION MAY HAVE CHANGED

- Logical replication added in PG 10
- Improved every version afterwards

COMMAND PROMPT, INC.

TABLE PARTITIONING IS DIFFERENT

- Native partitioning added in PG 10
  - Improved every version afterwards

COMMAND PROMPT, INC.

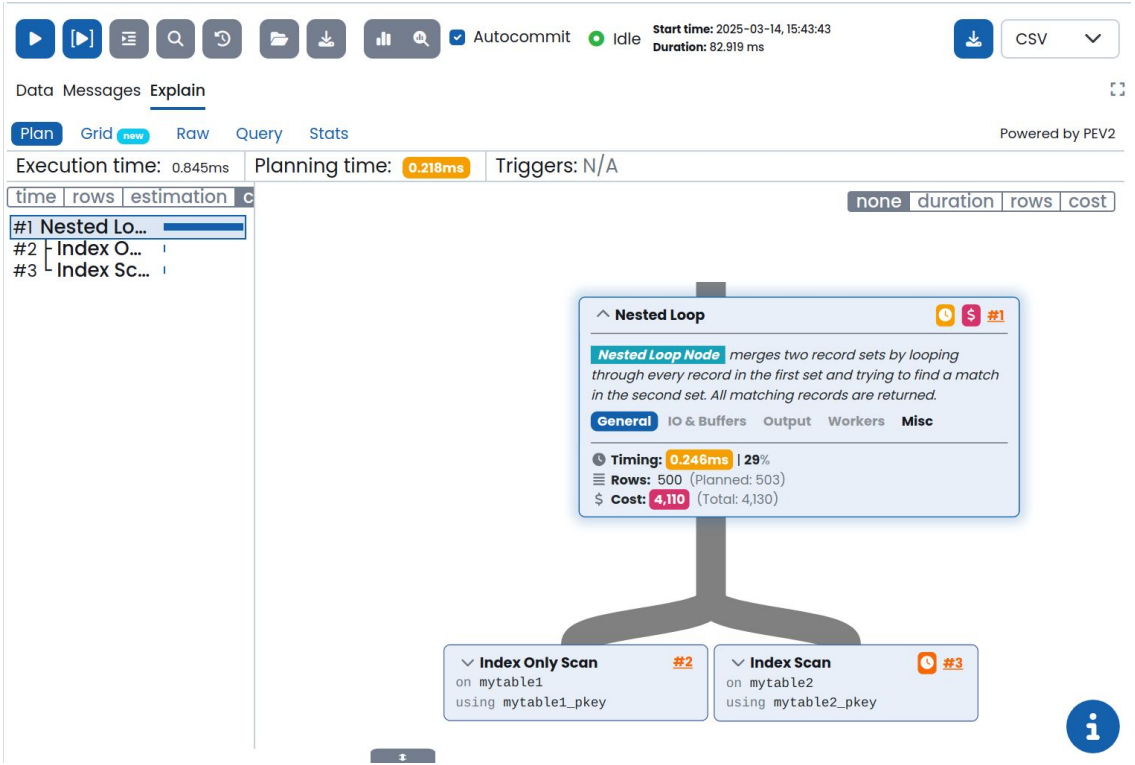
CTE'S ARE DIFFERENT WITH VERSION 12

- Common Table Expressions are interpreted by the query planer differently
- Before PG 12, CTE's were always MATERIALIZED
- After PG 12, CTE's are not MATERIALIZED unless explicitly specified

```
1   WITH
2     mycount AS (
3       SELECT
4         mytable1_sid
5       FROM
6         mytable1
7       WHERE
8         mytable1_sid BETWEEN 1 AND 500
9     )
10  SELECT
11    *
12  FROM
13    mytable2,
14    mycount
15  WHERE
16    mycount.mytable1_sid = mytable2.mytable2_sid;
```

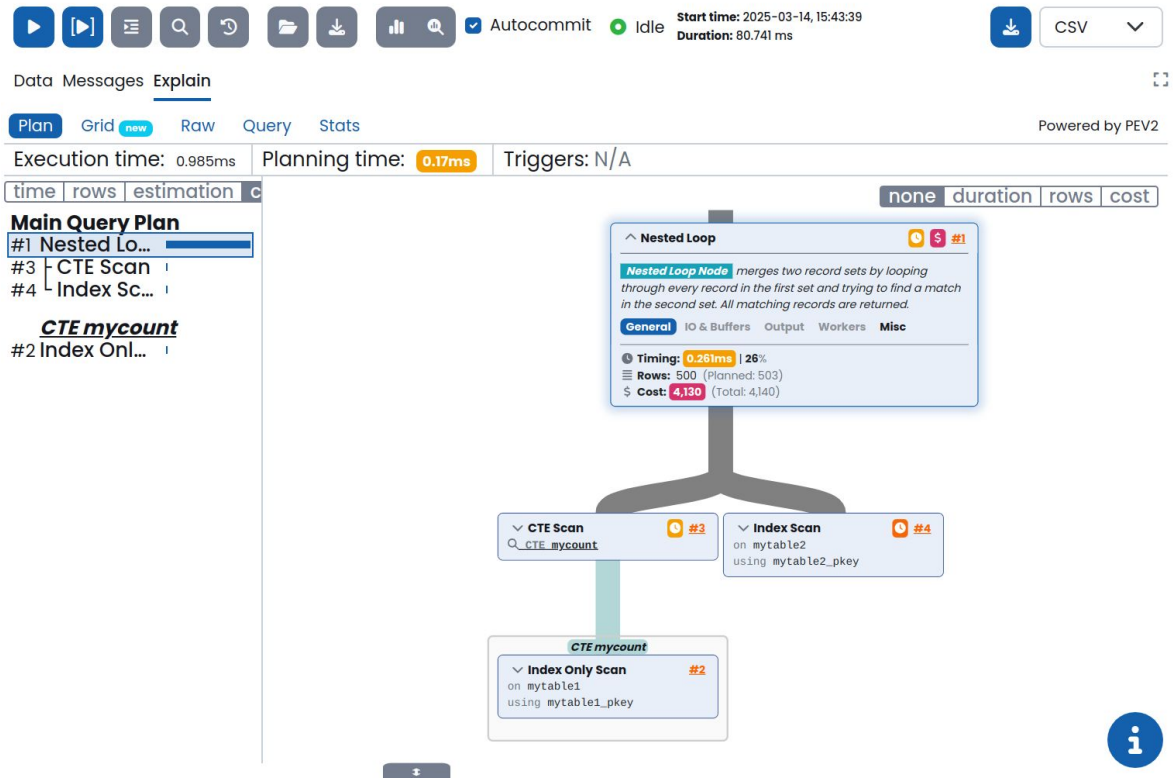
COMMAND PROMPT, INC.

CTE'S ARE DIFFERENT WITH VERSION 12



COMMAND PROMPT, INC.

CTE'S ARE DIFFERENT WITH VERSION 12



COMMAND PROMPT, INC.

CTE'S ARE DIFFERENT WITH VERSION 12

```
1  WITH
2    mycount AS (
3      SELECT
4        mytable1_sid
5      FROM
6        mytable1
7      WHERE
8        mytable1_sid BETWEEN 1 AND 500
9    )
10 SELECT
11   *
12 FROM
13   mytable2,
14   mycount
15 WHERE
16   mycount.mytable1_sid = mytable2.mytable2_sid;
```

```
1  WITH
2    mycount AS MATERIALIZED (
3      SELECT
4        mytable1_sid
5      FROM
6        mytable1
7      WHERE
8        mytable1_sid BETWEEN 1 AND 500
9    )
10 SELECT
11   *
12 FROM
13   mytable2,
14   mycount
15 WHERE
16   mycount.mytable1_sid = mytable2.mytable2_sid;
```

COMMAND PROMPT, INC.

LEARN ABOUT JIT

- Just In Time Compilation introduced in PG 11 with **DEFAULT OFF**
- Changed in PG 12 to **DEFAULT ON**
- Can cause queries that ran quickly before to run slower...
- Can cause queries that ran slowly before to run quicker!

COMMAND PROMPT, INC.

TEST TEST TEST TEST

- Test functionality
- Test performance
- Test every piece of functionality
- Test *everything*

COMMAND PROMPT, INC.

Crafting the UPGRADE

COMMAND PROMPT, INC.

“YEAH, BUT...”

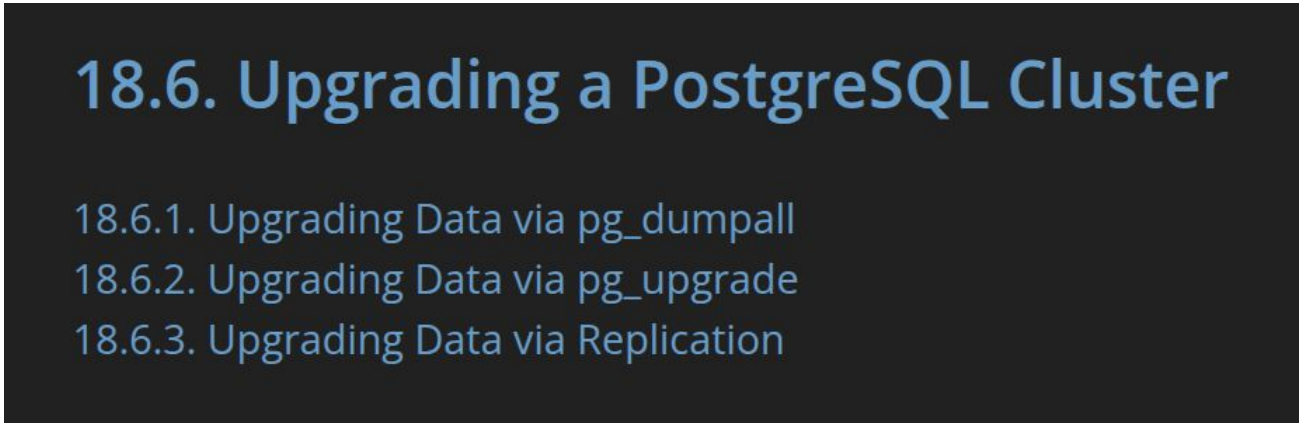
- Upgrades are easy, until they aren't
- Some common options (and their ‘Yeah Buts!”)
  - pg_upgrade
  - In-place pg_upgrade
  - Dump and Restore
  - Logical Replication
  - Pglogical replication
- It’s all about the tradeoff
- It’s time to get clever!

COMMAND PROMPT, INC.

Let's Get CLEVER

COMMAND PROMPT, INC.

DOCUMENTATION HAS LIMITS



COMMAND PROMPT, INC.

DOCUMENTATION HAS LIMITS

18.6.3. Upgrading Data via Replication

It is also possible to use logical replication methods to create a standby server with the updated version of PostgreSQL. This is possible because logical replication supports replication between different major versions of PostgreSQL. The standby can be on the same computer or a different computer. Once it has synced up with the primary server (running the older version of PostgreSQL), you can switch primaries and make the standby the primary and shut down the older database instance. Such a switch-over results in only several seconds of downtime for an upgrade.

This method of upgrading can be performed using the built-in logical replication facilities as well as using external logical replication systems such as pglogical, Slony, Londiste, and Bucardo.

COMMAND PROMPT, INC.

PG_UPGRADE

Standard

- Pros:
  - Manages nearly everything for you.
  - Fairly quick (unless very large databases)
- Cons:
  - Could go badly!
  - Can't roll back **with changes**

In Place

- Pros:
  - Less disk space needed
  - MUCH FASTER
- Cons:
  - MUCH DANGEROUS
  - Can't roll back **with changes**

***Getting Clever**: *Upgrade a replica, your original is a fallback.*

COMMAND PROMPT, INC.

DUMP / RESTORE

- Pros:
  - High Compatibility
  - Generally Scriptable
  - Target DB is a fresh initialization
  - Customizable
- Cons:
  - Large DB = Long Downtime
  - Disk space for source, backup, and target likely required
  - Can’t rollback **with changes**

COMMAND PROMPT, INC.

LOGICAL REPLICATION

- Pros:
  - Minimal downtime
  - Great for major version upgrades
  - Great for data center transfers / service changes
  - Rollback **with data** possible but requires setup
- Cons:
  - More complex
  - Requires additional hardware / storage for some time
  - Longer setup time
  - Relations require primary keys or replica identities
  - **Schema must be compatible!**

COMMAND PROMPT, INC.

“PGLOGICAL” REPLICATION

- Pros:
  - Same as logical replication
  - Works back to version 9.4 (logical only goes back to 10)
- Cons:
  - Same as logical replication
  - Not as feature rich as logical replication
  - **Schema must be compatible!**

COMMAND PROMPT, INC.

OTHER / COMBINATION

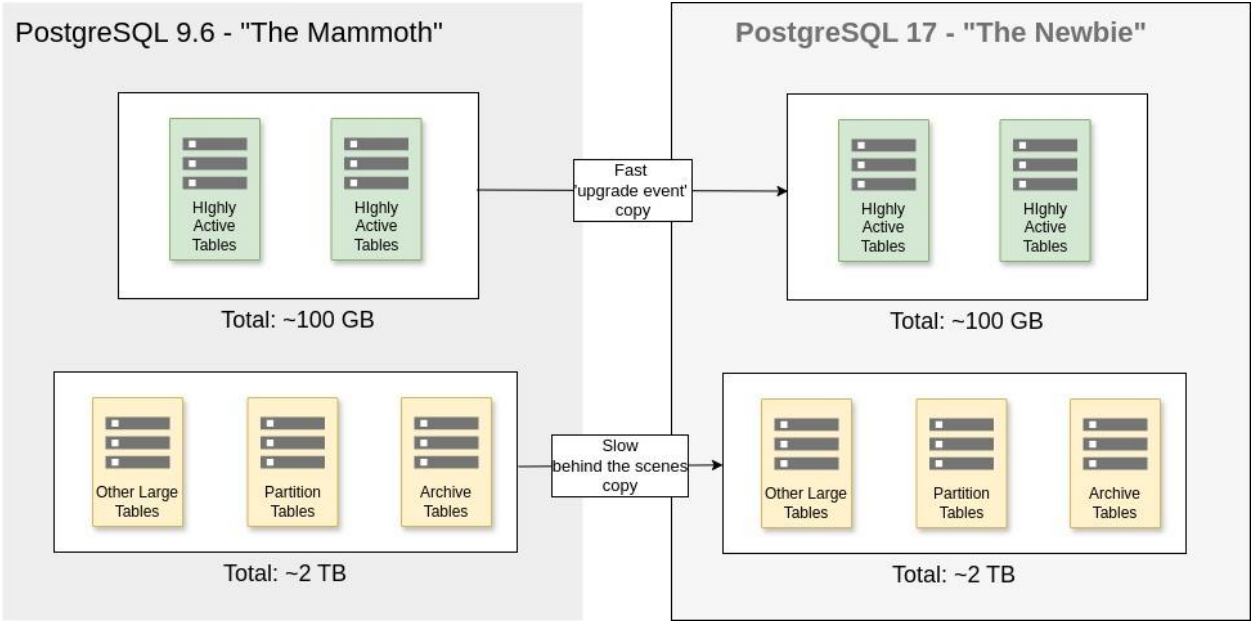
- Slony?
- Bucardo?
- Triggers and foreign data wrappers? (Gasp)
- Combination approaches
  - Logical Replication + Dump / Restore

COMMAND PROMPT, INC.

Real World
EXAMPLES

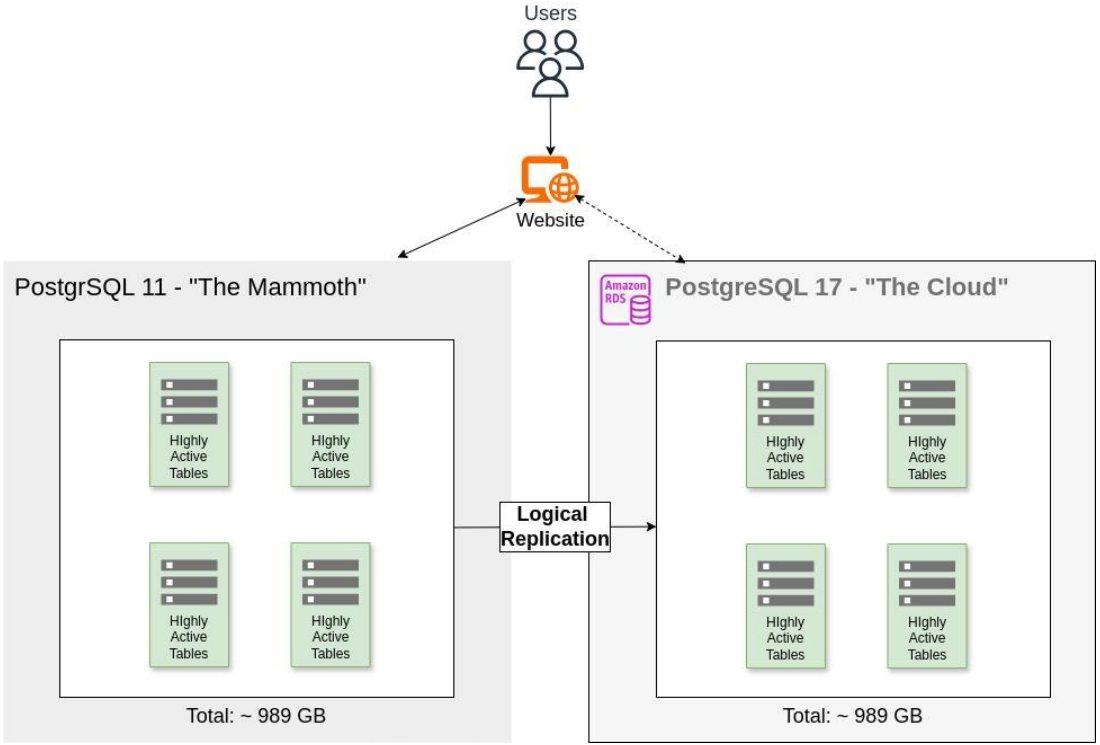
COMMAND PROMPT, INC.

EXAMPLE: CUSTOMIZED TO ACCESS PATTERN



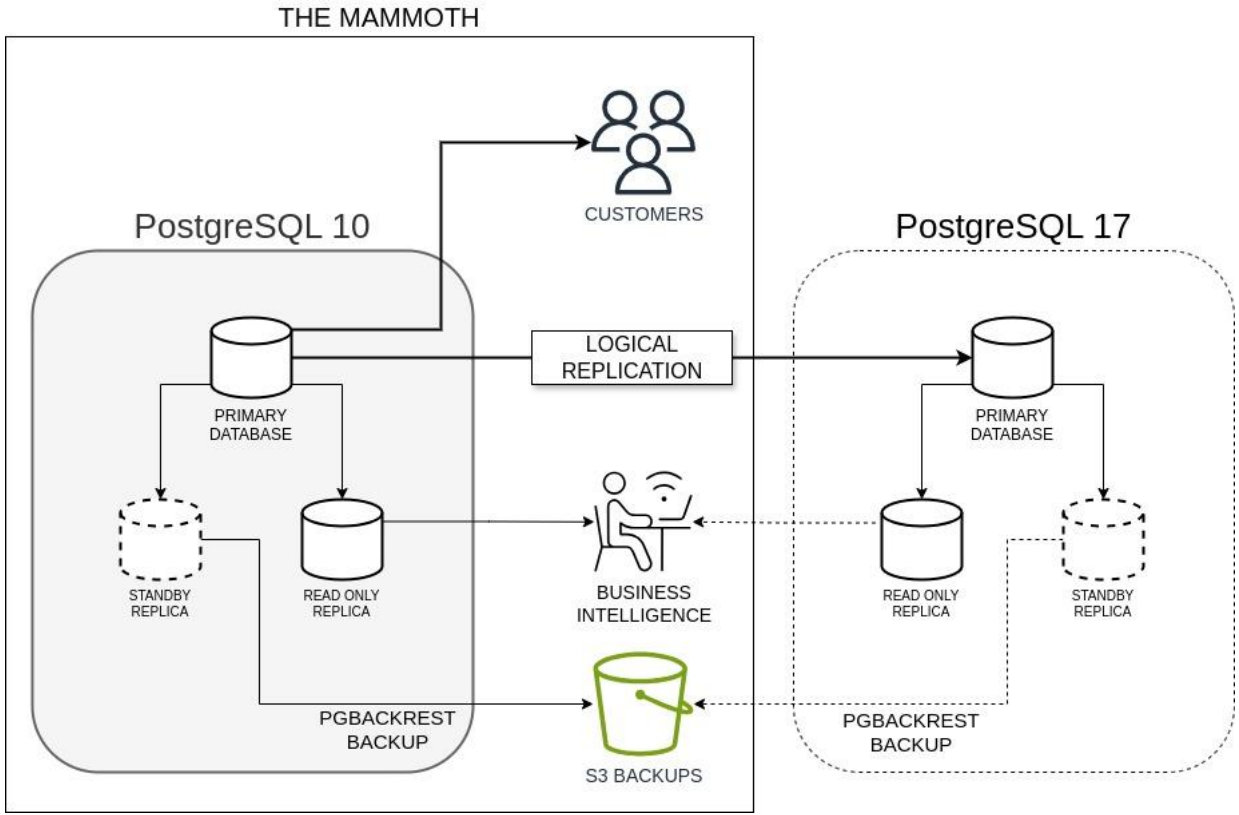
COMMAND PROMPT, INC.

EXAMPLE: CHANGING INFRASTRUCTURE



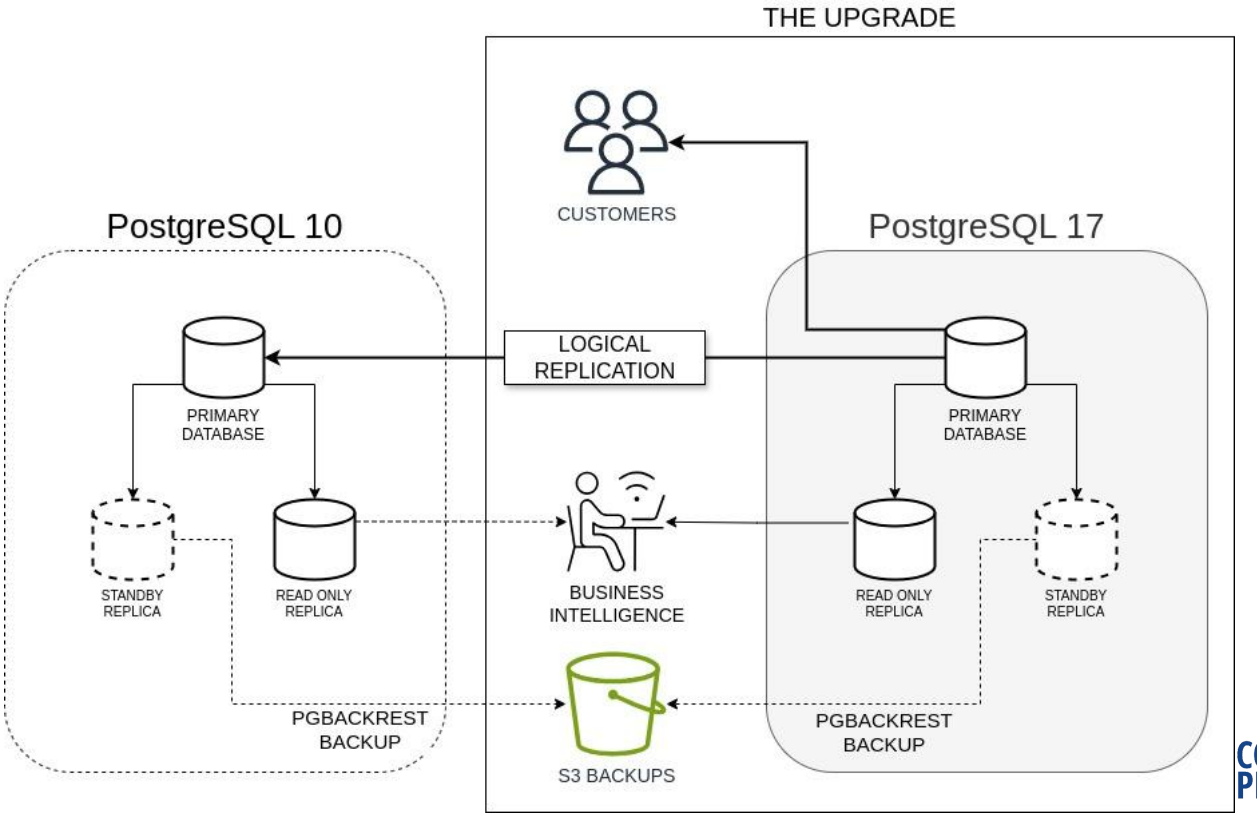
COMMAND PROMPT, INC.

EXAMPLE: RECREATING INFRASTRUCTURE



COMMAND PROMPT, INC.

EXAMPLE: RECREATING INFRASTRUCTURE



COMMAND PROMPT, INC.

SUMMARY

You ***need*** to upgrade

Ask the ***right questions***, get the ***right plan***

Figure out the best path with acceptable ***speed*** and easiest ***path***

Test test test until you're tired of testing

Command Prompt does this ***all the time*** - Get in touch for guidance and support

COMMAND PROMPT, INC.

Questions?

COMMAND PROMPT, INC.

COMMAND PROMPT, INC.

+1 503 667 4564

www.commandprompt.com/contact-us

EXPERTS IN POSTGRES AND OPEN SOURCE INFRASTRUCTURE