
PostgreSQL Migration/Upgrade with a Seamless Downtime

- by Avinash Vallarapu
OpenSCG

Agenda

- Major Concerns in an Infrastructure
- PostgreSQL Upgrade from 8.2.13 to 9.5.5 (Case Study)
- Results
- Learnings
- Questions ?

Major concerns in an Infrastructure

- Legacy Software
- Downtime during Software Upgrades
- Scaling Performance by adding more Servers
- Business Logic inside the Database.

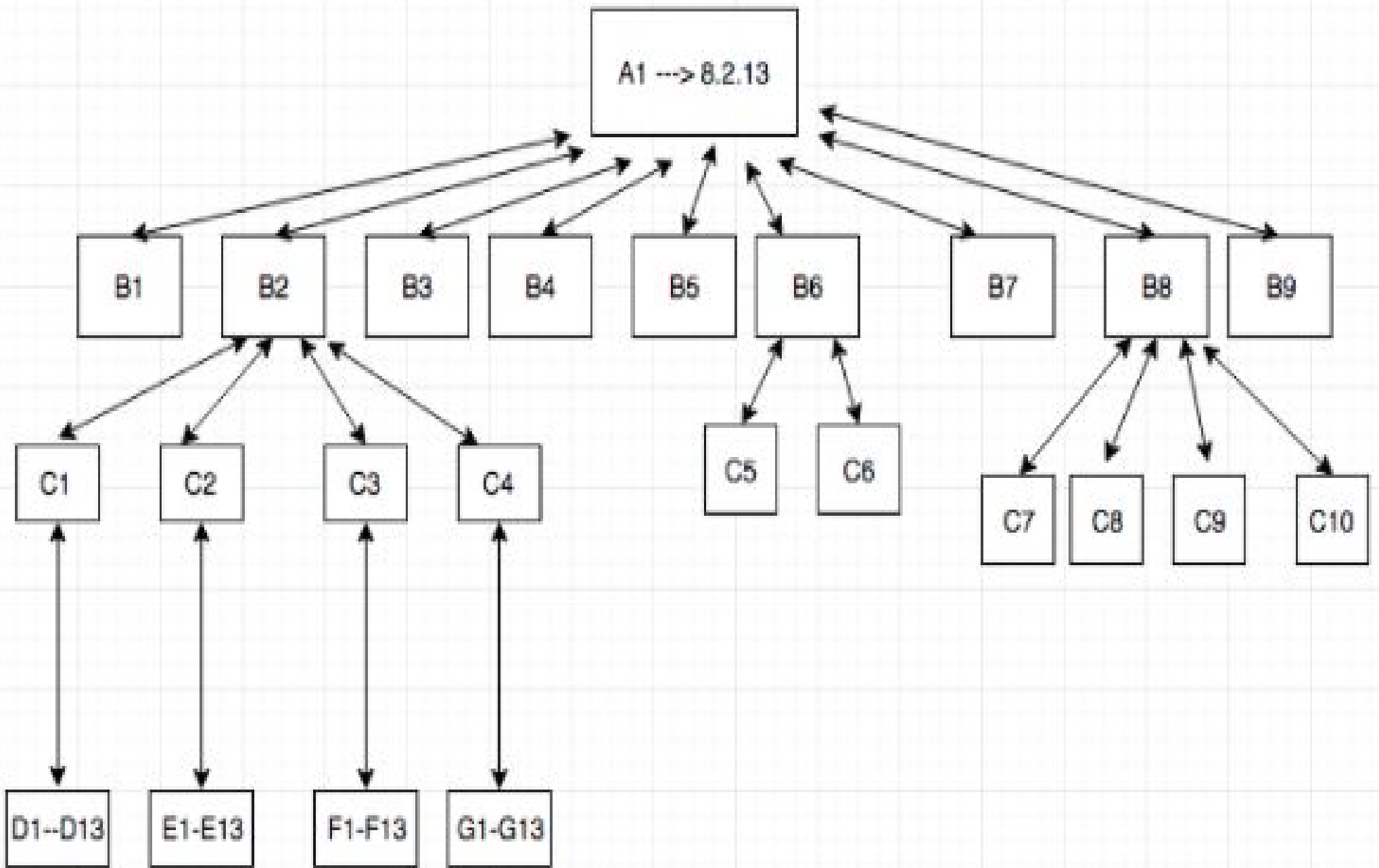
PostgreSQL Upgrade from 8.2.13 to 9.5.5 (Case Study)

Case Study

Purpose : PostgreSQL Upgrade from 8.2.13 to 9.5.5

Challenges :

- 72 Database Servers in Replication using Slony 1.2 (1 Master, 68 Read and 3 Writable Slony Replicas)
- 150 App Servers
- 4 Applications (Transactional, Back-Office, Reporting, Audits)
- 1200 Stored Procs
- **DOWNTIME**
- pg_upgrade does not work between 8.x and 9.x.
- Slony cannot replicate between 8.2 and 9.x

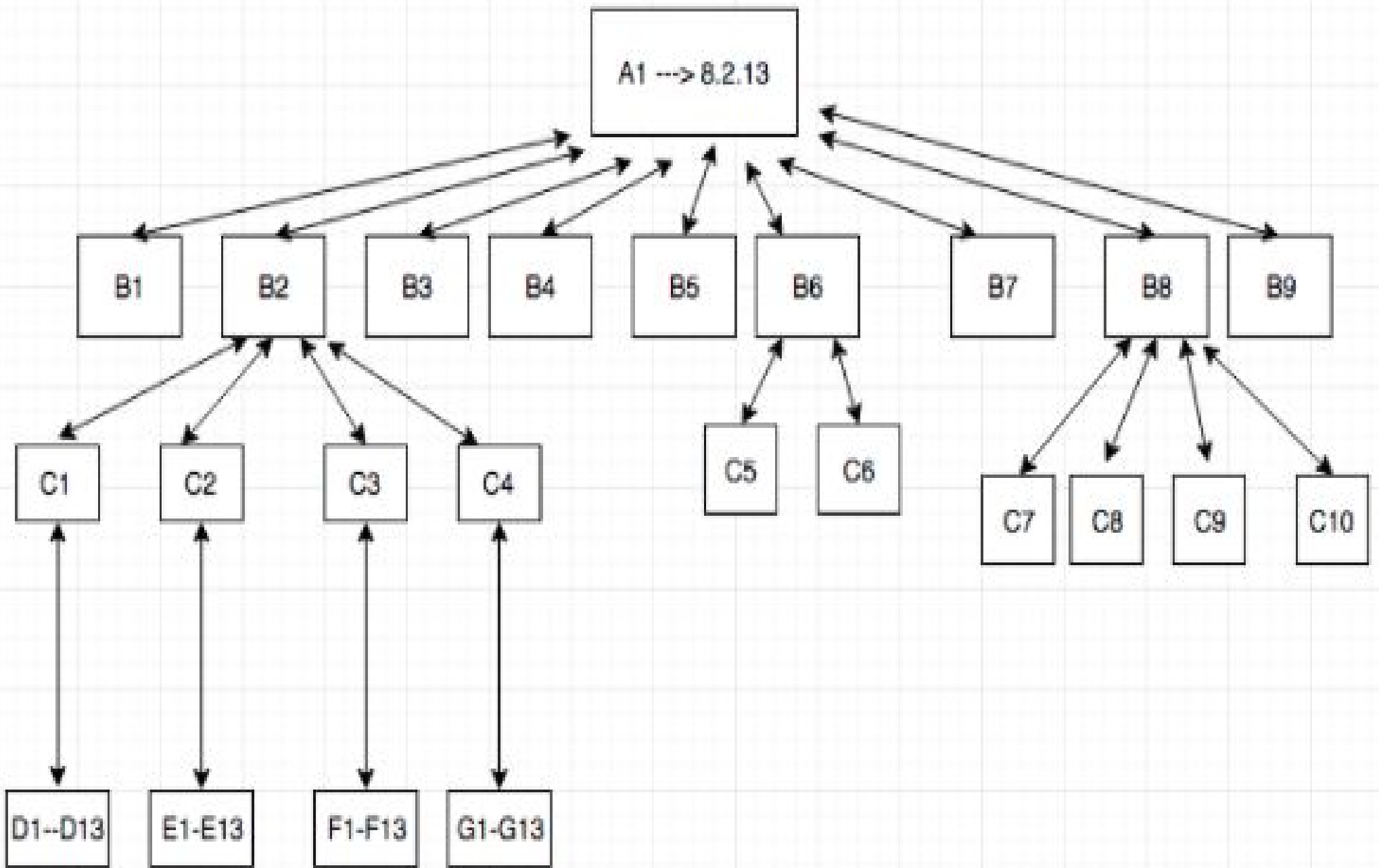


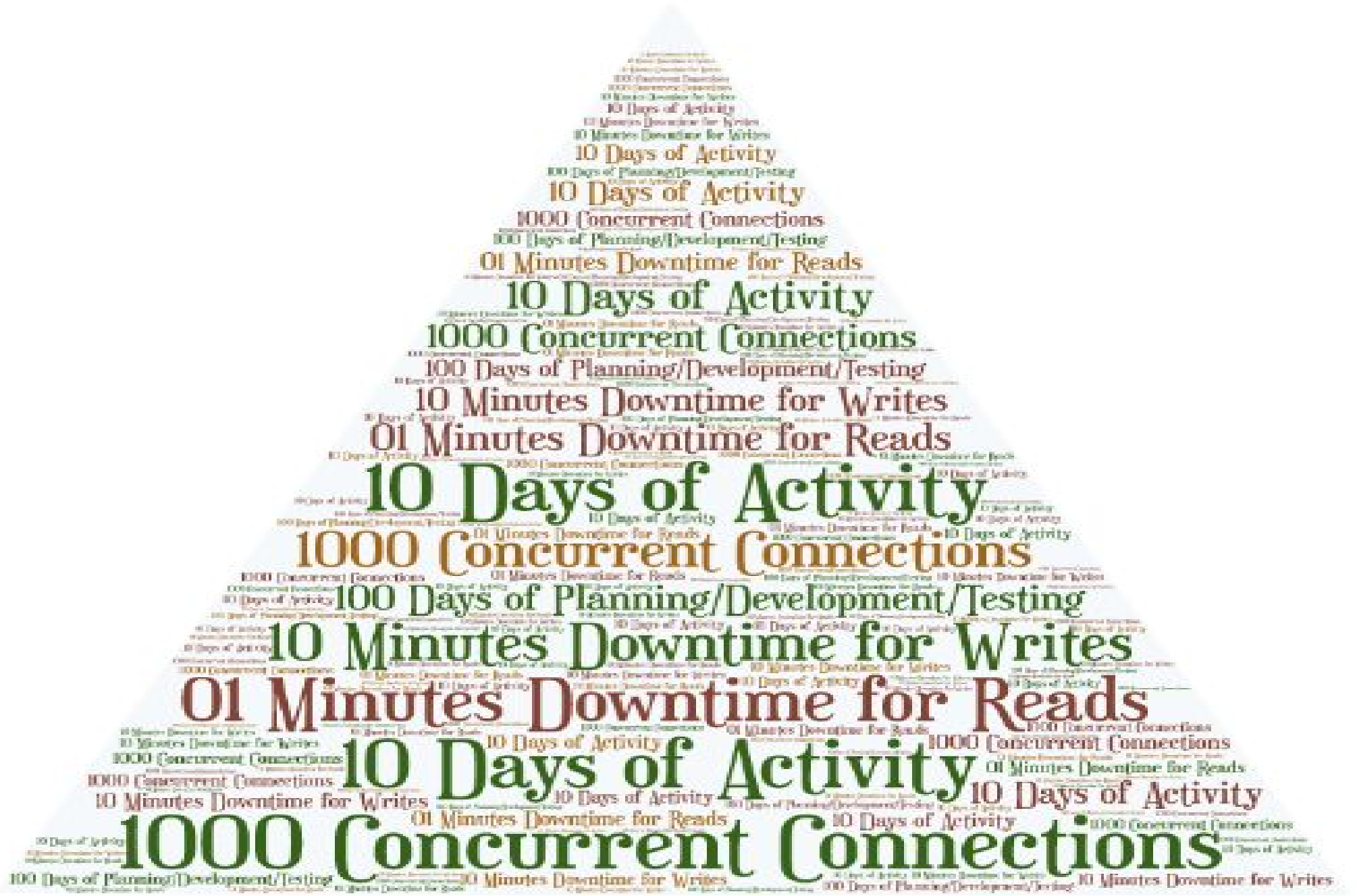
Why should we Upgrade?

- Older OS and DB Software Versions lacks several new features.
- Increase in Transactions, needing more Servers.
- No possibility of Performance Improvements.
- Not too many extensions/tools for PostgreSQL 8.x
- Maintenance of Database Objects is a hurdle.
- No Support for Older Versions(below PostgreSQL 9.2)

Pre-Upgrade Procedure

- Plan your Hardware Specifications
- Application to DB Connectivity.
- High Availability.
- Performance Testing.
- Backup Strategy.
- Plan your postgresql.conf parameters.
- Install all Softwares and Extensions in advance.





OpenSCG

 **BIGSQL**

How did we Upgrade?

Step 1 : Build the Infrastructure & Install Software

Build the Infrastructure & Install Software

- Configured 2 Dedicated Machines for Master and Failover Slave.
- Configured 71 Slave Virtual Machines on Cloud.
- Pushed BigSQL Sandbox using Ansible to all the Servers.

What did the BigSQL Sandbox include?

BigSQL Sandbox includes a variety of Packaged/Compiled Softwares & Extensions.

Following are the Softwares and Extensions we have installed using BigSQL.

- PostgreSQL 9.5.5
- PlProfiler
- PgBadger
- pg_stat_statements
- pgBouncer
- pg_repack
- pg_buffercache
- Pldebugger

Step 2 : Replication between 8.2 and 9.5

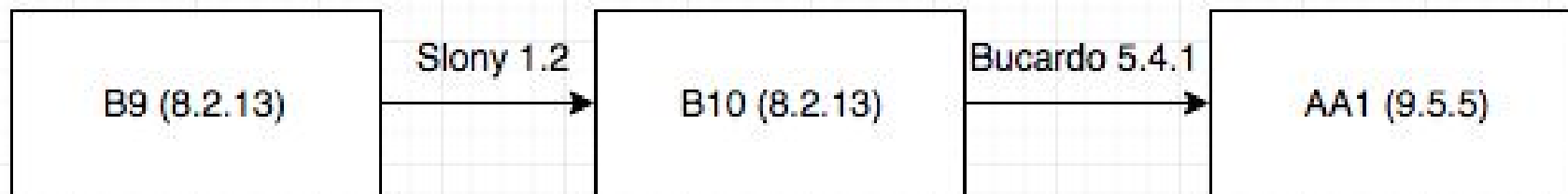
Setup Replication between an 8.2 Instance and a New 9.5 PostgreSQL Instance.

Does Streaming Replication Work ? → No

Can Slony work between 8.2 & 9.5? → No

Does Londiste Work → No

What about Bucardo → Yes



Step 3 : Streaming Replication between New 9.5 Instance and Slaves/Cascaded Slaves

Streaming Replication between New 9.5 Instance and Slaves/Cascaded Slaves

- Installed Rep Manager on all the Servers.
- Set Up Streaming Replication between Master and all the Slaves and Cascaded Slaves.

Step 4 : Re-direct all the Read traffic with a seamless downtime

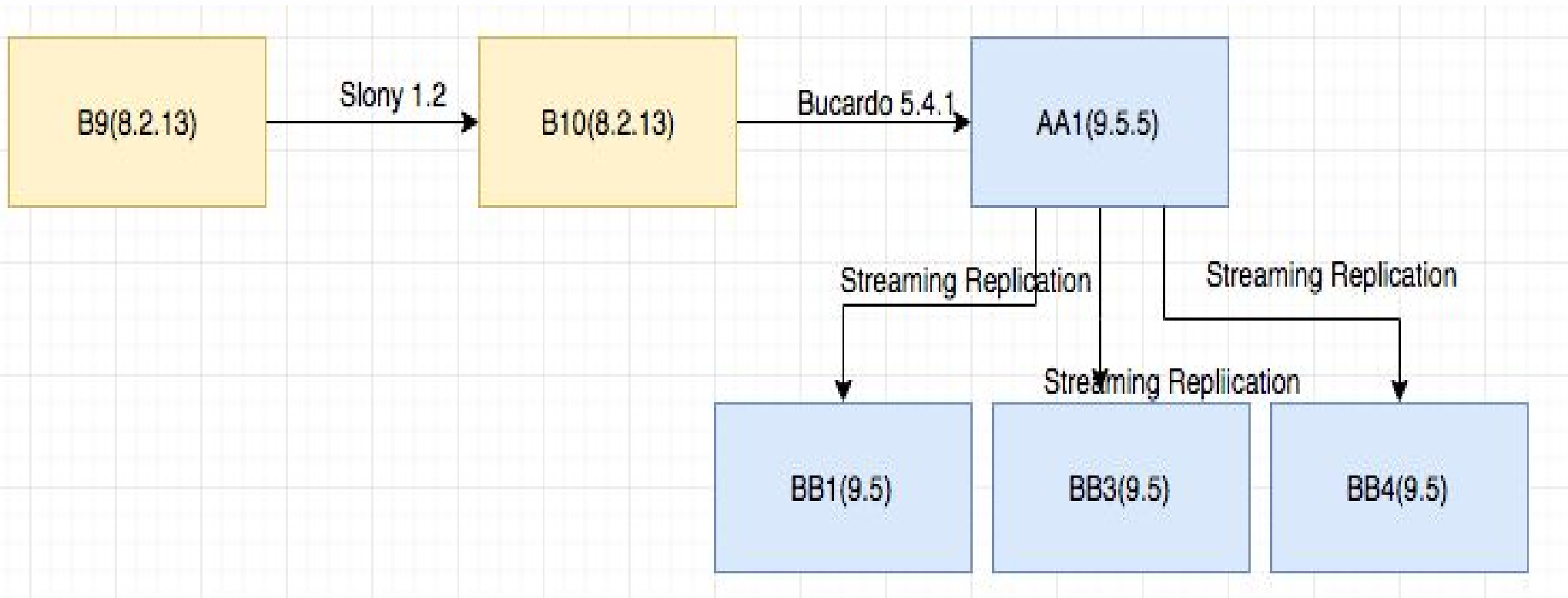
Re-direct all the Read traffic with a seamless downtime

- Make modifications to the already existing HA Proxy configurations with the New Slave Server IP's.
- Reload the HA proxy with the New Server IP's.

Step 5 : Build the 3 Writable Slaves

Build the 3 Writable Slaves

- Set up Streaming Replication between the New 9.5 Master and the 3 future Writable Slaves.
- Stop Bucardo Replication between 8.2 and 9.5.
- Add the 3 Writable Slaves as 3 New Targets to the Bucardo Replication
- Start Bucardo Replication.



Step 6 : Re-direct writes through PgBouncer to
existing 8.2.13 Master

Re-direct writes through PgBouncer to existing 8.2.13 Master

- Configure PgBouncer on a New Server.
- Re-direct all the Application Writes through PgBouncer to the Old 8.2.13 Master.
- This enables us to use a PgBouncer restart to re-direct the Writes to the New 9.5.5 Master.

Step 7 : Re-direct writes to the New 9.5.5 Master.

Re-direct writes to the New 9.5.5 Master

- Stop bucardo Replication between the 8.2.13 and all the 4 → 9.5.5 Instances.
- Setup Bucardo Replication between the New Master and the 3 Writables Slaves.
- Restart PgBouncer with the New Master IP Address for Writes.

Results

Results

- No need to push all the DDL Changes or Code changes to all the Instances manually.
- Improvement in Reads & Writes Performance.
- Max Replication Lag reduced to 0.5 seconds due to Streaming Replication.
- Improved performance of Stored Procs through SQL Profiler.
- Servers able to handle over 3x times the load as earlier.
- Plan to reduce the number of Servers.
- Plan to decommission the 3 Writable Slaves and re-direct the load to Master.
- Automatic-Failover
- Partitioning

Learnings

Learnings

- Replication lag using Bucardo 5.4.1
→ 40 seconds to 120 seconds
- Replication lag using Slony 2.2
→ 8 seconds to 20 seconds.
- Setting `max_standby_streaming_delay` & `max_standby_archive_delay` to -1 may cause a great lag in Streaming Replication.

Questions ?