KubeCon | CloudNativeCon

North America 2023

Is Kubernetes suitable to run Very Large Postgres Databases?

You can now restore a VLDB 300 times faster!

Disaster Recovery with Very Large Postgres Databases

Michelle Au, Google

Gabriele Bartolini, EDB

About Us

Michelle Au

Software Engineer at Google

Kubernetes sig-storage TL

Kubernetes contributor since 2017

Gabriele Bartolini

VP/CTO of Cloud Native at EDB

PostgreSQL user since ~2000

PostgreSQL Community member since 2006

DoK Ambassador

DevOps evangelist

Open source contributor

- Barman (2011)
- CloudNativePG (2022)

Outline

1. Postgres Disaster Recovery
2. Volume snapshot backup & recovery with CloudNativePG
3. Volume Snapshot API & CRDs
4. Demo
5. Conclusions

Postgres Disaster Recovery: an intro

Business continuity goals

- Recovery Point Objective (**RPO**)
  - Amount of data we can afford to lose
    - Measured in time or bytes
  - Primarily for Disaster Recovery
- Recovery Time Objective (**RTO**)
  - How long the service can be restored after a failure
    - Measured in time
  - Primarily for High Availability

Postgres is a Rock Solid Database

PostgreSQL ★ ROCK solid database
ACiD
Atomicity - Consistency - Isolation - Durability

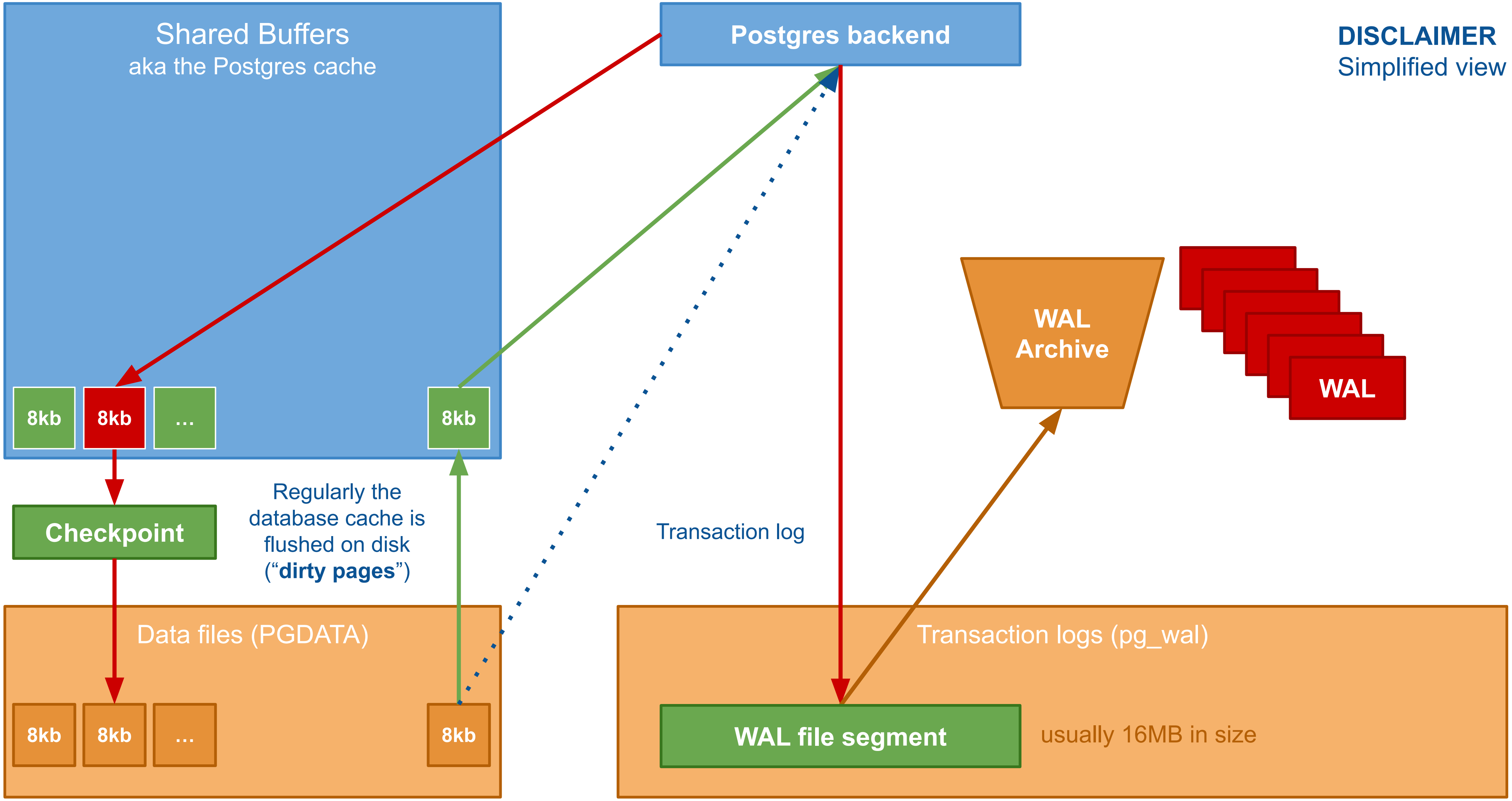
since 1995

Business continuity in Postgres 101

- Crash recovery with Write-Ahead Log, aka WAL (version 7.1, 2001)
- Continuous backup & Point in Time Recovery (8.0, 2005)
  - **Physical Hot Base Backups and WAL archiving for Disaster Recovery (DR)**
- Continuous recovery through WAL shipping (8.2, 2006)
  - Warm standby replicas for High Availability (HA)
- Streaming replication with Hot Standby replicas (9.0, 2010)
  - Synchronous replication at transaction level (9.1, 2011)
- Physical Hot Base Backups from a Hot Standby replica (9.6, 2016)
- NOTE: pg_dump takes logical backups (not for business continuity)

The Write-Ahead Log (WAL)

DISCLAIMER
Simplified view



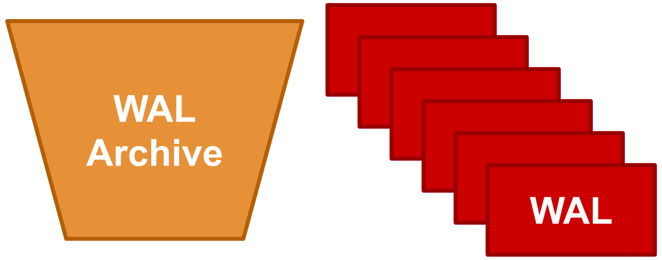
Regularly the database cache is flushed on disk (“**dirty pages**”)

Transaction log

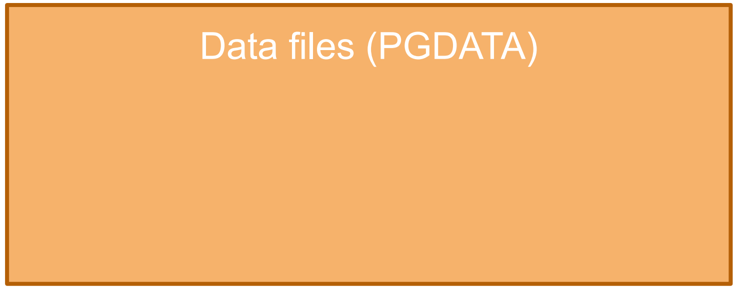
What needs to be backed up

DISCLAIMER
Simplified view

WAL archive is key for any recovery (crash, full, point-in-time) and replication



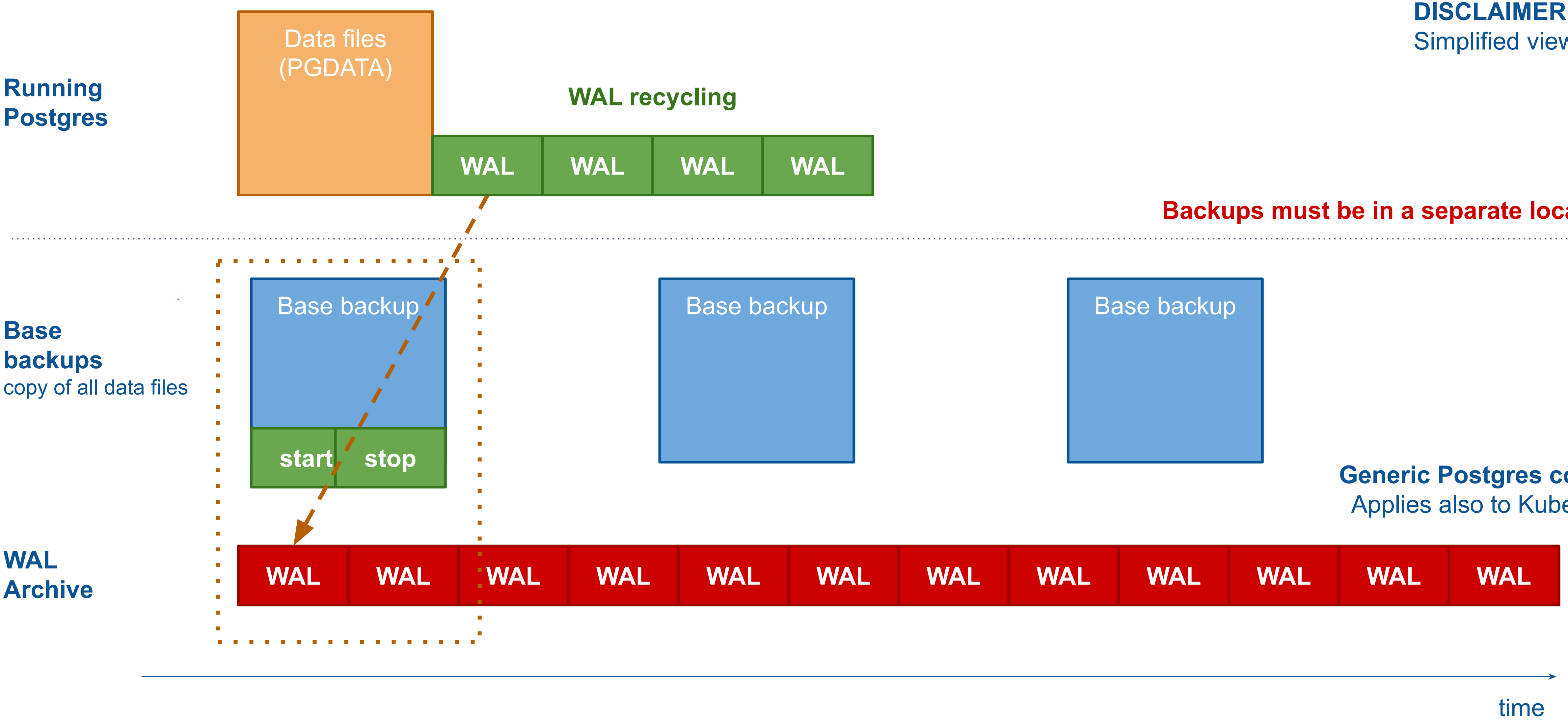
PostgreSQL data files



Generic Postgres concept
Applies also to Kubernetes

Continuous backup 101

DISCLAIMER
Simplified view

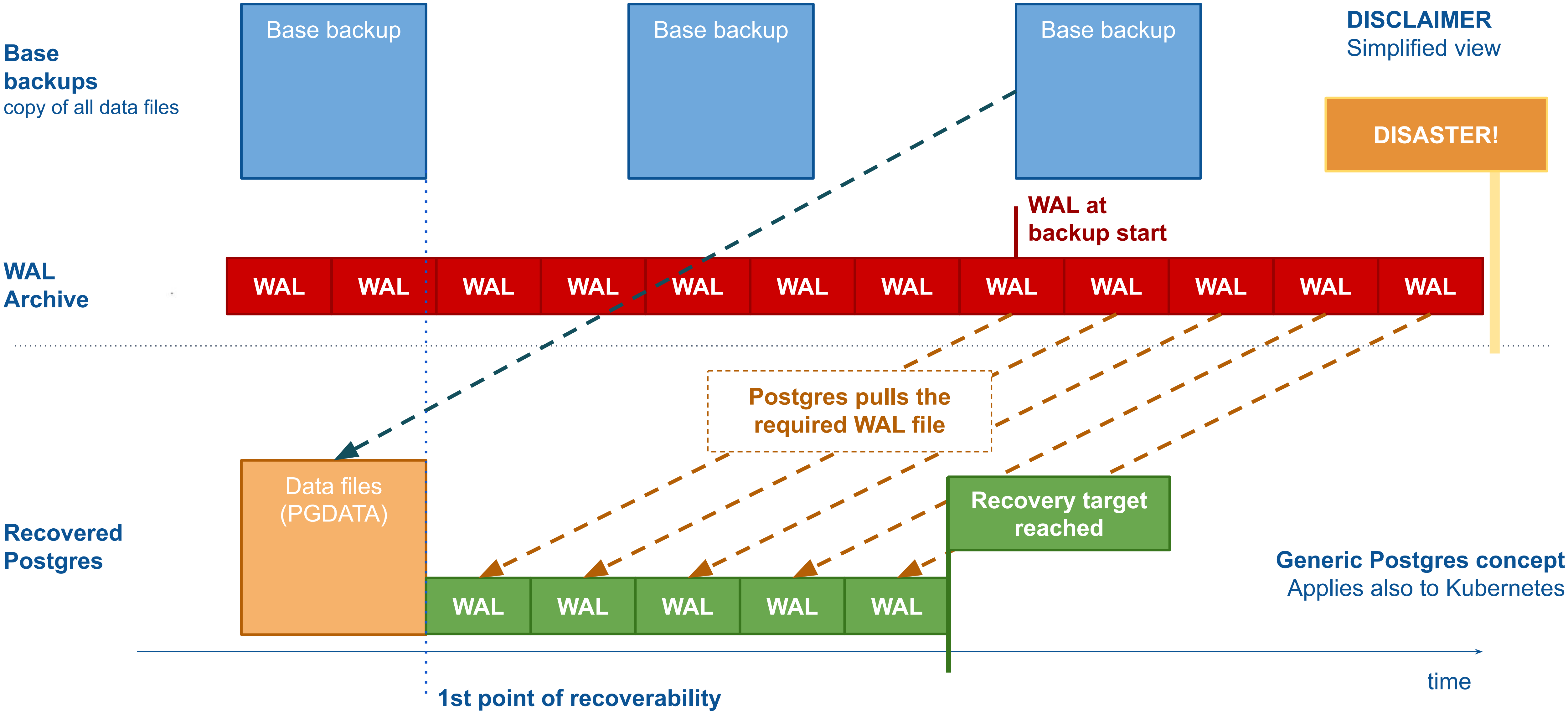


Backups must be in a separate location

Generic Postgres concept
Applies also to Kubernetes

Point In Time Recovery 101

DISCLAIMER
Simplified view



Generic Postgres concept
Applies also to Kubernetes

Recap for Disaster Recovery

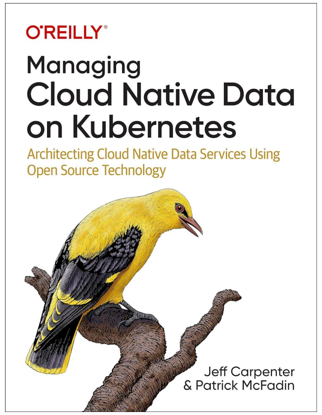
- Take regular base backups of your Postgres database
  - Hourly, daily, weekly
- Ensure continuous WAL archiving is in place
- Safely store both base backups and WAL archive
  - In proximity of the original database (for fast RTO)
  - In different locations, including regions (for Disaster Recovery)
- You can recover at any time
  - From the end of the 1st available backup to the latest archived transaction
- Practices adopted in production by many organizations for 10+ years

Volume snapshot backup & recovery with CloudNativePG

KubeCon | CloudNativeCon
North America 2023

CloudNativePG

- Kubernetes native database for Postgres workloads (Carpenter & McFadin)
  - Maximum leverage of the Kubernetes API
  - Automated, declarative management via operators
  - Observable through standard APIs
  - Secure by default
- Production ready operator and operand images for Postgres
  - Extends Kubernetes to manage the full lifecycle of a Postgres database
  - Directly manages persistent volume claims (no statefulsets)
- Open source, openly governed, vendor-neutral: cloudnative-pg.io
- Used to run Postgres in Kubernetes for this presentation



O'REILLY®

Managing
Cloud Native Data
on Kubernetes

Architecting Cloud Native Data Services Using
Open Source Technology

Jeff Carpenter
& Patrick McFadin

Disaster Recovery with CloudNativePG

- **WAL archive** is on Object storage
    - By default, WAL files are archived every 5 minutes maximum (RPO)
- **Physical base backups** can be taken on:
    - Object storage
    - **Volume Snapshots** via the standard Kubernetes API
        - Introduced in CloudNativePG 1.21 (October 2023)
- **Volume snapshot backup & recovery is the focus of this presentation**

Base Backup Comparisons

Features	**Object Storage**	**Volume Snapshots**
WAL archiving	Required	Recommended
Backup type	Hot backup	Hot and cold backup
Backup size	Full backup	**Incrementals and differentials**
Point in Time recovery	Yes	With WAL archiving
*Geographic availability**	Cross multi-region	Multi-region
*Optimizations**		**Copy on write**

* Depends on storage type

Benchmarks

Database size	PGDATA volume size	WAL volume size	Snapshot full backup time	Object store full backup time	Snapshot recovery time	Object store recovery time
4.5 GB	8 GB	1 GB	1m 50s	9m 15s	31s	3m 29s
44 GB	80 GB	10 GB	20m 38s	1h 6m	27s	31m 59s
438 GB	800 GB	100 GB	2h 42m	9h 53m	48s	59m 51s
4381 GB	8000 GB	200 GB	3h 54m 6s	95h 12m 20s	**2m 2s**	**10h 6m 17s**

x **24.40** faster

x **298.17** faster

* Benchmarked using AWS EBS gp3 disks
* The test considers base backup recovery only, without WAL file recovery

Volume snapshot API & CRDs

Kubernetes Volume Snapshots

- GA since K8s 1.20
- Standard and portable API across storage providers
- Supported by major cloud providers and on-prem storage providers
- Operations:
  - Create a snapshot of a PVC
  - Delete a snapshot
  - Create a PVC from a snapshot

Kubernetes Volume Snapshots

User

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
 name: my-snapshot
spec:
 volumeSnapshotClassName: my-snapshot-class
 source:
   persistentVolumeClaimName: my-pvc
```

Admin

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
 name: my-snapshot-class
driver: my-driver
deletionPolicy: Delete
parameters:
  driver-option1: foo
```

Kubernetes Volume Restore

User

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: restore-pvc
spec:
 dataSourceRef:
   name: my-snapshot
   kind: VolumeSnapshot
   apiGroup: snapshot.storage.k8s.io
 accessModes:
   - ReadWriteOnce
 resources:
   requests:
     storage: 10Gi
```

CloudNativePG API - Backups

```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: my-cluster
spec:
  ...
  backup:
    volumeSnapshot:
      className: my-snapshotclass
    barmanObjectStore: # For WAL archive
      destinationPath: <obj storage path>
    retentionPolicy: '7d'
```

```yaml
apiVersion: postgresql.cnpg.io/v1
kind: ScheduledBackup
metadata:
  name: my-cluster-backup
spec:
  schedule: '0 0 0 * * *'
  backupOwnerReference: self
  cluster:
    name: my-cluster
  immediate: true
  method: volumeSnapshot
```

On demand:

```
$ kubectl cnpg backup -m volumeSnapshot my-cluster
```

CloudNativePG API - Restore

```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: my-cluster
spec:
  ...
  bootstrap:
    recovery:
      volumeSnapshots:
        storage:
          name: volume-snap-1
          kind: VolumeSnapshot
          apiGroup: snapshot.storage.k8s.io
        walStorage:
          name: wal-snap-1
          kind: VolumeSnapshot
          apiGroup: snapshot.storage.k8s.io
```

Demo

Demo

Backup and restore of 3 node CNPG cluster on GKE

Conclusions

Future

Kubernetes enhancements

K8s 1.27: Volume group snapshots (alpha)

Container Object Storage Interface (alpha)

CloudNativePG enhancements

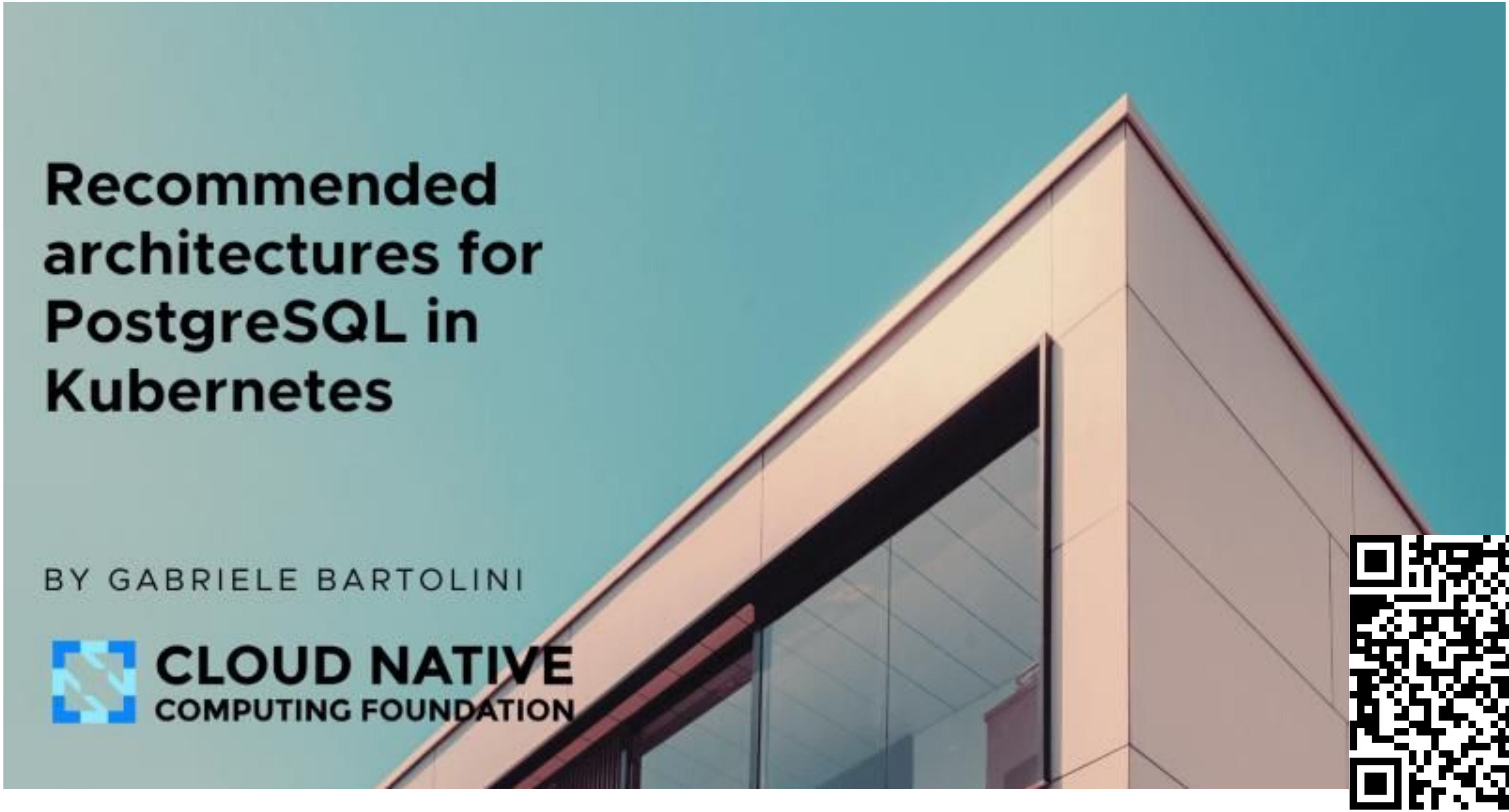
CloudNativePG 1.22: Tablespaces

PVC cloning for scale up and in-place upgrades

Takeaways

- Kubernetes + PostgreSQL + CloudNativePG is a full open source stack
  - Vendor lock-in risk mitigation
- Main benefits of using volume snapshots
  - Better RPO and RTO
  - Suitable for all major cloud service providers
    - For on-premise deployments make sure you check the storage capabilities
  - Unleashes Postgres VLDB in Kubernetes
    - Incremental/differential backup & recovery

Suggested reading



Recommended architectures for PostgreSQL in Kubernetes

BY GABRIELE BARTOLINI

CLOUD NATIVE
COMPUTING FOUNDATION

Suggested reading

PostgreSQL Disaster Recovery with Kubernetes’ Volume Snapshots

References

CloudNativePG backups: https://cloudnative-pg.io/documentation/1.21/backup/

Kubernetes Volume Snapshots: https://kubernetes.io/docs/concepts/storage/volume-snapshots/

Demo configs and scripts: https://github.com/gbartolini/postgres-kubernetes-playground/tree/main/gke

Questions?

Please scan the QR Code above
to leave feedback on this session