

DATABASES

ADVANCED

3 DAYS

PostgreSQL Performance Tuning, High Availability and Replication

An advanced course for teams already running PostgreSQL who need it faster and more resilient. Three days covering workload baselining, optimiser and index tuning, vacuum and write path optimisation, partitioning, lock contention, streaming and logical replication, automated failover with Patroni, and disaster recovery testing. Assumes working PostgreSQL administration knowledge and runs on clusters large enough for plan choices to matter.

Why Advanced PostgreSQL Tuning and HA Training?

Measured, not guessed

Every tuning change is baselined, applied and re-measured, so teams learn to prove improvement rather than assume it.

Targets real availability numbers

Architecture choices are tied to stated recovery point and recovery time objectives rather than discussed in the abstract.

Failover rehearsed under pressure

Participants promote replicas, handle split-brain risk and recover a broken cluster against the clock.

Self-managed and cloud

Covers Patroni-managed clusters alongside what changes on RDS, Aurora and Cloud SQL.

What you'll be able to do

- ✓ Baseline a workload and identify the queries and resources worth tuning
- ✓ Tune memory, connection and planner configuration for a specific workload
- ✓ Interpret execution plans and diagnose optimiser misestimation
- ✓ Design index strategies including partial, covering and expression indexes
- ✓ Tune vacuum, checkpoint and write path behaviour under sustained load
- ✓ Partition very large tables and verify pruning actually occurs
- ✓ Diagnose and resolve lock contention and blocking chains
- ✓ Configure streaming replication including synchronous and quorum commit
- ✓ Implement automated failover with Patroni and prevent split-brain
- ✓ Use logical replication for scaling out and near-zero-downtime migration
- ✓ Design, document and test a disaster recovery plan against defined objectives

Who this is for

- Database administrators operating production PostgreSQL
- Platform, DevOps and site reliability engineers

- Data engineers running high volume PostgreSQL workloads
- Solution architects designing highly available data tiers
- Teams preparing for growth, audit or availability commitments

Curriculum

01 Establishing a Performance Baseline

- Defining what good looks like before changing anything
- `pg_stat_statements` for total impact ranking rather than single query time
- Wait event analysis to separate CPU, I/O, lock and network bound problems
- Benchmarking correctly with `pgbench` and realistic workloads
- Capturing before and after evidence for every change
- **Hands-on:** baseline a supplied workload and rank the tuning candidates

02 Configuration and Resource Tuning

- Shared buffers, effective cache size and operating system cache interaction
- `work_mem` sizing and diagnosing spills to disk
- Parallel query configuration and when parallelism hurts
- Planner cost parameters and tuning for SSD storage
- Connection limits, pooling modes and the cost of high connection counts
- **Hands-on:** tune an instance for a defined workload and quantify the gain

03 The Optimiser and Execution Plans

- Plan structure, operators and reading bottom up
- Estimated versus actual rows as the primary diagnostic
- Statistics, extended statistics and correlated columns
- Join algorithm selection and join order search
- Plan regressions and controlling planner behaviour
- **Hands-on:** diagnose and repair a set of misestimated plans

04 Index Strategy and Maintenance

- Choosing between B-tree, GIN, GiST, BRIN and hash for a workload
- Composite ordering, covering indexes and index-only scans
- Partial and expression indexes for skewed data
- B-tree skip scan behaviour and its effect on composite index design
- Index bloat detection and rebuilding concurrently
- **Hands-on:** redesign an index set and prove the write cost is acceptable

05 Vacuum, Bloat and the Write Path

- Autovacuum tuning per table for high churn workloads
- Bloat measurement and reclamation without downtime
- HOT updates, fill factor and reducing index write amplification
- Checkpoint tuning, full page writes and WAL volume reduction
- Long transactions, replication slots and blocked cleanup
- **Hands-on:** tune a heavy write workload and halve its WAL generation

06 Partitioning and Very Large Tables

- Range, list and hash partitioning design decisions
- Partition pruning at plan time and execution time
- Partition-wise joins and aggregation
- Automated partition maintenance and retention with pg_partman
- Migrating an existing large table to a partitioned structure online
- **Hands-on:** partition a live table with minimal locking and verify pruning

07 Locking, Concurrency and Contention

- Lock modes, lock queues and how one blocked session stalls many
- Diagnosing blocking chains and deadlocks from the logs
- DDL locking and safe online schema changes
- Contention hotspots: sequences, counters and hot rows
- Lock timeouts, statement timeouts and defensive session settings
- **Hands-on:** reproduce a production style blocking incident and resolve it

08 Streaming Replication in Depth

- Physical replication mechanics and WAL shipping
- Asynchronous, synchronous, remote_apply and quorum commit trade-offs
- Replication slots, retention and the disk risk of inactive slots
- Cascading replicas and delayed standbys as a protection layer
- Hot standby query conflicts and feedback settings
- **Hands-on:** build a replication topology and measure the latency cost of each commit mode

09 Automated Failover and Cluster Management

- Patroni architecture, distributed consensus and the role of etcd or Consul
- Automatic failover, switchover and controlled maintenance
- Fencing, watchdog and split-brain prevention
- Connection routing with HAProxy, PgBouncer or service discovery
- repmgr and pg_auto_failover as alternatives
- **Hands-on:** kill the primary and recover the cluster with no data loss

10 Logical Replication and Scaling Out

- Publications, subscriptions and selective replication design
- Cross-version replication for near-zero-downtime upgrades
- Read scaling, workload separation and routing read traffic
- Conflict handling, sequence management and DDL limitations
- Sharding options and when Citus becomes the right answer
- **Hands-on:** migrate a database across major versions with seconds of downtime

11 Disaster Recovery and Operational Assurance

- Recovery point and recovery time objectives as architecture inputs
- pgBackRest for parallel backup, retention and restore verification
- Cross-region and cross-cloud recovery topologies
- Restore testing as a scheduled, automated activity
- Writing runbooks the on-call engineer can follow at three in the morning
- Managed service differences on RDS, Aurora and Cloud SQL
- **Hands-on:** capstone disaster scenario recovered against a stated objective