house of brick

THE DEFINITIVE GUIDE FOR DATABASE ADMINISTRATORS & IT MANAGERS

The AWS Database Sprawl Playbook

Why visibility beats guesswork — a practical guide to identifying, measuring, and controlling AWS database sprawl before it creates licensing risk and compliance exposure.

What's Inside

This playbook gives database administrators and IT managers a clear framework for understanding, measuring, and eliminating AWS database sprawl — before Oracle or Microsoft shows up for an audit.

300+
Oracle audits successfully defended

$4.2B
In potential Oracle claims avoided

25+
Years of Oracle & SQL Server expertise

About House of Brick: For over 25 years, House of Brick has helped organizations manage Oracle and SQL Server licensing complexity. House of Brick is part of Opscompass, a Software Asset Intelligence platform providing continuous visibility into database environments across on-premises, cloud, and hybrid infrastructure.

SECTION 1 FOUNDATION

What Is AWS Database Sprawl?

Database sprawl occurs when database instances multiply across your AWS environment faster than your ability to track them. It happens gradually: a developer spins up an RDS instance for testing, a team clones a production database for a migration proof-of-concept, someone provisions Aurora for a new microservice. Each instance makes sense individually. Collectively, they create a management and compliance problem.

Where Sprawl Lives in AWS

- **Amazon RDS instances** running Oracle, SQL Server, MySQL, or PostgreSQL
- **Self-managed databases on EC2** that bypass RDS entirely
- **Aurora clusters** that started as experiments and became semi-production
- **Multi-AZ deployments** where the standby counts as a second licensable instance
- **Read replicas** created for reporting that require their own licenses
- **Dev/test environments** mirroring production without governance

> **The core problem:** Spreadsheet-based tracking fails because databases change faster than anyone can update a spreadsheet. By the time you reconcile your inventory, new instances have appeared, old ones have changed, and your documentation is already outdated.

SECTION 2 VISIBILITY GAPS

Why AWS-Only Tools Create Blind Spots

AWS provides visibility into infrastructure. AWS Config can tell you that an RDS instance exists, what instance class it runs, and when it was created. But AWS tools see instances, not databases. They cannot tell you what is happening inside those databases.

What AWS Tools Cannot See

For Oracle and SQL Server workloads, the licensing risk lives *inside* the database:

- **Oracle option usage** — Advanced Compression, Diagnostics Pack, Tuning Pack, Partitioning, Advanced Security
- **SQL Server feature activation** — Always On AG, online indexing, Enterprise Edition features
- **Historical usage patterns** — whether a feature was ever used, even briefly
- **Configuration drift** — parameter changes that affect licensing status

The EC2 Visibility Gap

Self-managed Oracle and SQL Server on EC2 exist entirely outside AWS's database management plane. AWS inventory tools track the EC2 instance but cannot see what database software is installed, what edition is running, or what features are active.

The Hybrid Reality

Most organizations do not run everything in AWS. Databases on-premises, in VMware, on Nutanix, and in other clouds mean multiple incomplete inventories that cannot connect to each other.

SECTION 3 FINANCIAL EXPOSURE

The Licensing Risk Hidden in Sprawl

Database sprawl creates direct financial exposure through licensing. Oracle and Microsoft both audit their customers, and audit findings in cloud environments can be substantial. The risk is not theoretical.

The Oracle vCPU Trap

Oracle's cloud licensing policy counts **2 vCPUs as 1 processor** license when hyperthreading is enabled. The Oracle Processor Core Factor Table does not apply in AWS.

> **Critical rule:** Every 2 vCPUs on AWS requires 1 Oracle processor license, regardless of underlying processor type. Multi-AZ doubles your requirement.

RDS Instance Class	vCPUs	OEE Licenses	Initial License Cost + 1st Year of Support
db.r5.large	2	1	$57,950
db.r5.xlarge	4	2	$115,900
db.r5.2xlarge	8	4	$231,800
db.r5.4xlarge	16	8	$463,600

Source: Oracle Technology Global Price List (March 2025). OEE = $47,500/processor + 22% annual support.

Multi-AZ Doubles Your License Requirement

AWS is explicit: BYOL deployments require licenses for **both the primary and the standby** DB instance in a Multi-AZ deployment.

Oracle Options: Hidden Cost Multipliers

Option	$/Processor	Common Trigger
Diagnostics Pack	$7,500	Running an AWR report
Tuning Pack	$5,000	SQL Tuning Advisor
Partitioning	$11,500	Creating a partitioned table
Advanced Compression	$11,500	Enabling table compression
Advanced Security	$15,000	Enabling TDE encryption

Each option also requires 22% annual support.

SQL Server Complexity

Edition mismatch: Enterprise features on a Standard license

Always On AG: Secondary replicas require SA or additional licenses

Core counting: Different calculation from Oracle's vCPU rule

Annual true-ups: Unlicensed deployments surface at renewal

The Audit Evidence Problem

Auditors ask what you have run over the past several years — not just today. While the burden of proof is on the vendor to demonstrate historical usage, without historical tracking you lack justification for discounting any such evidence.

SECTION 4 MIGRATION RISK

How Sprawl Derails Cloud Migrations

The Pre-Migration Discovery Gap

Migration planning requires knowing exactly what you have. When inventory is incomplete:

- Project timelines slip as "discovered" databases get added mid-project
- Budget estimates expand as hidden databases surface
- Undocumented dependencies emerge late in the project
- Licensing calculations prove wrong because they were based on incomplete data

The Sizing Problem

Without visibility into actual utilization, teams over-provision (wasting money) or under-provision (causing performance problems). For Oracle specifically, over-provisioning means over-licensing — choosing a larger instance class than necessary directly increases your license cost.

Post-Migration Sprawl

Migrations often create sprawl rather than reduce it. Parallel test environments, source databases running longer than planned, and validation replicas become permanent fixtures without governance.

SECTION 5 THE SOLUTION

The Visibility Solution: What You Actually Need to See

Cross-Platform Inventory

You need a single view that spans all environments:

- RDS instances (Oracle, SQL Server, MySQL, PostgreSQL)
- Self-managed databases on EC2
- On-premises databases
- Databases in other clouds (Azure, OCI, GCP)
- Databases on VMware, Nutanix, and Hyper-V

Database-Level Visibility

Beyond instance inventory, you need to see inside each database:

- Feature and option usage (especially Oracle)
- Configuration parameters affecting licensing
- Patch levels and version information
- Storage utilization and growth trends
- Performance metrics for rightsizing decisions

Configuration Drift Detection

Databases change constantly. You need to detect:

- New instances appearing in your environment
- Feature activation that triggers new licensing
- Instance class changes that alter license counts
- Multi-AZ enablement that doubles requirements
- Security configuration changes

Historical Evidence for Audit Defense

Time-series data must demonstrate:

- When features were or were not in use
- Configuration state at any point in time
- Changes and attribution
- Compliance status across the entire audit period

SECTION 6 REFERENCE

The Database Sprawl Consequence Map

Database sprawl rarely announces itself. It shows up as an unexpected audit finding, a cloud bill that doubled after migration, or a security request you can't answer. This map connects seven of those consequences to what Opscompass surfaces and the decision each one makes possible.

SPRAWL CONSEQUENCE	WHAT THE DBA EXPERIENCES	WHAT OPSCOMPASS SHOWS THEM	DECISION IT ENABLES
Developers use Oracle features without realizing the licensing implications.	Developers use Oracle features (Partitioning, Diagnostics, Tuning) without telling anyone. DBA finds out when the auditor does.	Oracle option and feature usage active across all databases. Flag when new features are detected. Side-by-side view of licensed entitlements vs. actual usage.	Proactively monitor for unlicensed feature usage before audit. Build a defensible audit response with evidence already on hand.
Cloud costs spike after a migration.	Databases were sized like on-prem workloads. Nobody re-evaluated instance size after cutover. Cloud bill arrives and nobody can explain it.	CPU and IOPS usage per database instance over time. Over-provisioned instances flagged with rightsizing guidance. Cost exposure broken down by database and environment.	Rightsize cloud instances with actual usage data. Prioritize cost reduction work across the full database footprint.
Shadow databases running unlicensed or untracked.	Dev teams spin up databases in AWS or Azure without notifying DBAs. Old dev/test/stage environments linger for months. No single place to see everything that exists.	Full inventory across on-prem, AWS, Azure, GCP, Oracle Cloud. New database instances detected and flagged automatically. Databases with no owner, no tags, or no activity highlighted.	Decommission abandoned databases before they create liability. Enforce tagging and ownership standards at scale.
Infrastructure change causes a database outage or compliance failure.	Infrastructure team makes a config change without telling DBAs. DBA doesn't know until something breaks or an audit flags it. No trail connecting the infrastructure change to the database impact.	Configuration drift detected across infrastructure and database layers. Change history tied to specific databases. Alerts when changes occur in infrastructure that affect licensed or monitored databases.	Identify the root cause of issues faster. Hold infrastructure and DB teams accountable to the same change record.
VMware or Nutanix expansion triggers a massive Oracle licensing event.	A new host is added to a VMware cluster. Virtual machines with Oracle software could be moved to the new host causing additional license requirements. Licensing exposure jumps without anyone making a deliberate choice.	Virtualization topology mapped against Oracle database placement. Cluster-level host counts and changes tracked over time. License exposure modeled when infrastructure changes occur.	Catch virtualization changes that impact licensing. Model licensing scenarios before expanding infrastructure.
Cloud migration proposal from Oracle, AWS, or Azure doesn't reflect actual cost.	Vendor proposal is based on assumptions, not actual workload data. DBAs don't have the usage evidence to push back. Migration estimate turns out to be 2-3x what was projected.	Actual CPU, IOPS, and memory usage per database. Oracle feature usage that would carry into the cloud. Side-by-side cost modeling for different migration paths.	Negotiate cloud contracts from a position of real data. Avoid licensing blowups by planning migrations with full visibility.
Security team can't confirm which databases are encrypted or patched.	Security asks for a database inventory report. DBA cobbles one together from spreadsheets and tribal knowledge. Answer is out of date by the time it's delivered.	Encryption and patch status across all databases in real time. Configuration checks against CIS, NIST, and FedRAMP benchmarks. Insecure configurations flagged with remediation context.	Respond to security audits with a current, automated report. Prioritize patching based on actual exposure, not guesswork.

SECTION 7 COMMON QUESTIONS

Frequently Asked Questions

How does Oracle count processor licenses for RDS instances?

Oracle counts 2 vCPUs as 1 processor license when hyperthreading is enabled (the default on AWS). For example, a db.r5.xlarge with 4 vCPUs requires 2 Oracle processor licenses. The Oracle Processor Core Factor Table does not apply in AWS environments.

What is the difference between BYOL and License Included on RDS?

With BYOL (Bring Your Own License), you are responsible for providing the Oracle licenses and maintaining compliance. License Included bundles the Oracle license into the AWS hourly fee, but is only available for Oracle Standard Edition 2 and has a 16 vCPU limit. Oracle Enterprise Edition always requires BYOL on RDS.

Can Oracle audit my AWS environment?

Yes. Your Oracle license agreement grants Oracle audit rights regardless of where the software runs. Running Oracle on AWS does not exempt you from audit provisions in your contract. Auditors have full rights to review cloud deployments.

What happens if an Oracle audit finds unlicensed usage?

Oracle typically requires you to purchase licenses for unlicensed usage, often with back-support fees. Audit findings can reach into the millions of dollars, especially when Enterprise Edition options were used without licenses. House of Brick has defended over 300 Oracle audits and helped clients avoid more than $4.2 billion in potential Oracle licensing claims.

How long does it take to get visibility into database sprawl?

Initial inventory discovery can happen within days of deploying appropriate tooling. Building historical evidence takes longer — you can only capture data from the point you start monitoring forward, which is why starting immediately is critical to audit readiness.

Key Takeaways

Sprawl is invisible by default. Without purpose-built tooling, most organizations undercount their Oracle and SQL Server exposure by a significant margin.

AWS tools see instances, not databases. What AWS Config and Cost Explorer miss is precisely where licensing risk lives.

Multi-AZ and options are silent cost multipliers. Both can double or triple licensing costs without triggering any AWS-level alert.

Historical evidence is non-negotiable. Audit defense requires being able to counter inaccurate assertions with historical data. You cannot rely on the vendor to have yourbest interests (or your budget) in mind.

Migrations amplify the problem. Discovery gaps identified mid-migration are more expensive to fix than gaps found before the project starts.

Start monitoring now. You can only build historical evidence from the moment you begin. Every day without monitoring is a day you cannot defend in an audit.

Why Most Database Sprawl Efforts Fail

Most teams try to control database sprawl with periodic reviews, ad-hoc scripts, or spreadsheets. That approach breaks down quickly in hybrid and multi-cloud environments.

To actually control sprawl, you need four things working together:

Continuous tracking
New databases are created constantly across AWS, Azure, VMware, and on-prem environments. If tracking isn't continuous, your inventory is already outdated.

Versioned history
Audits don't ask what exists today. They ask what existed months ago. Use the historical tracking to build a defensible audit response with evidence.

Normalized inventory
Database license calculations vary across RDS, EC2, Azure SQL, or VMware. Without normalization, you can't compare, track, or make licensing decisions across environments.

Drift detection
Configuration and feature changes happen silently. One toggle can trigger licensing exposure. With drift tracking, you can be be aware of the changes and their licensing impact.

This is the gap most tools don't solve. It's also where Opscompass is built to operate.

Opscompass continuously tracks database assets, normalizes them across environments, tracks historical changes, and flags drift that affects licensing. This gives DBAs and infrastructure teams a single, accurate view of what exists, what changed, and what it means for cost and compliance.

READY TO ELIMINATE GUESSWORK?

See What You Are Missing

Schedule a demo to see how continuous database visibility helps you identify sprawl, reduce licensing risk, and face any Oracle or Microsoft audit with confidence.

300+	$4.2B	25+
Oracle audits defended	In avoided Oracle claims	Years of expertise

Control Your Database Sprawl

Automatically map every RDS and EC2 database in your AWS account. Catch unauthorized spin-ups and configuration changes regularly — no more stale spreadsheets.

Schedule a full Opscompass demo at