

■ ZOPDEV RESEARCH · JUNE 2026

The State of Cloud 2026

An Emerging Markets view of global cloud, drawn as engineering plates. The system, its boundaries, and its failure domains, annotated.

Sheet index & survey notes.

This excerpt plots the three most **diagrammable** chapters of the full report — the footprint, the waste, and the architecture — as numbered plates. Read the drawings first; the prose is annotation.

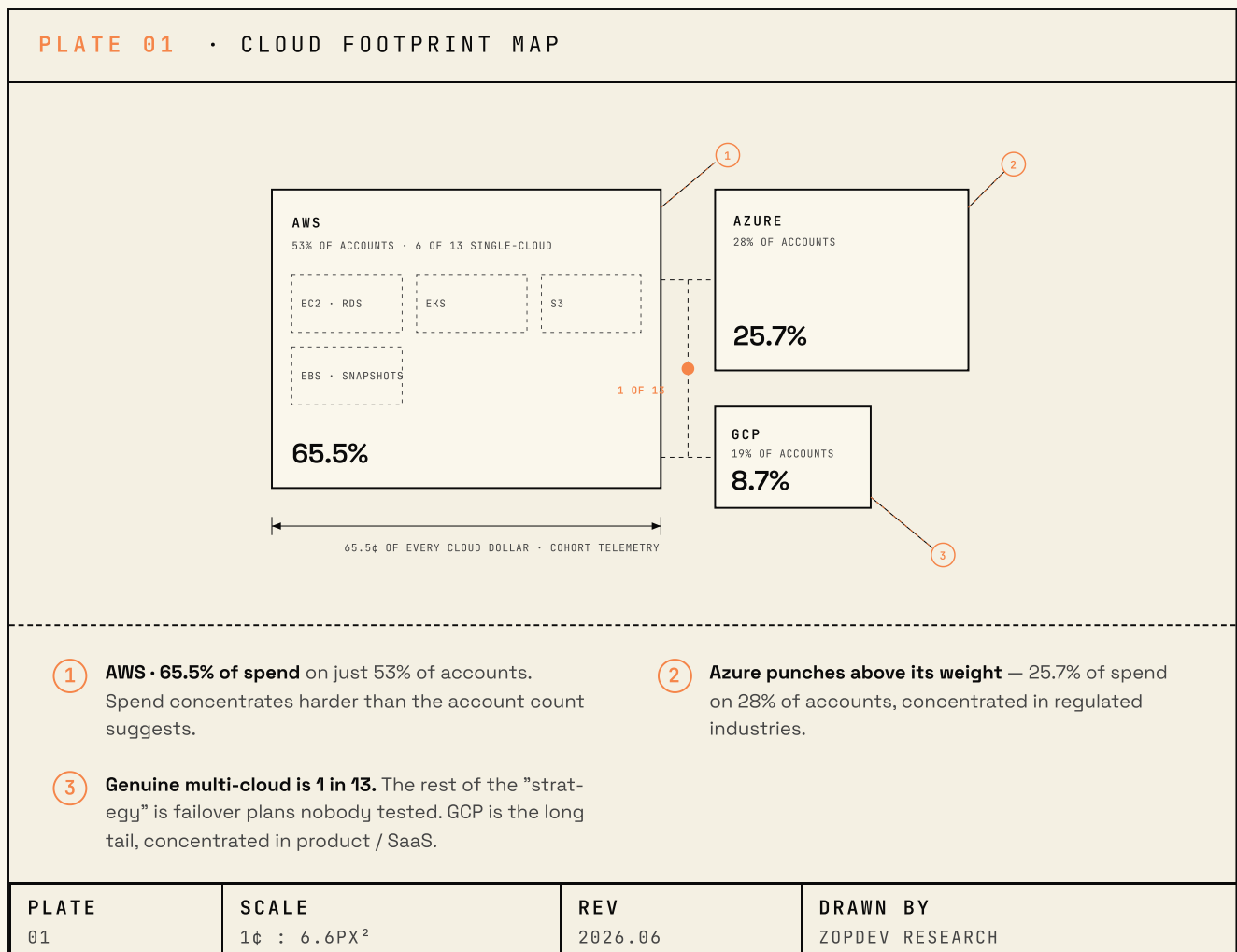
NO	CHAPTER	STATUS
ES	Executive Summary — cloud stopped being new; the bill is still a mystery	FULL EDITION
01	Hypothesis Scoreboard — 10/10 pre-registered hypotheses confirmed	FULL EDITION
02	The Defensible Number — 16.9% provable; true figure likely 30-40%	FULL EDITION
03	The State of Enterprise Cloud — spend, footprint, trajectory	PLATES 01-02
04	Where the Waste Lives — 87% forgotten, not strategic	PLATE 03
05	The Detection-to-Action Gap — 30× faster to find than to fix	FULL EDITION
06	The Architectural Picture — sprawl is the new technical debt	PLATES 04-05
07	An Emerging Markets View — currency, sovereignty, labor economics	FULL EDITION
08	Predictions for 2027 — falsifiable, dated, gradeable	FULL EDITION
09	The GPU Shift — the same anti-patterns, three more zeros	FULL EDITION
CL	Closing — three trends, three counter-trends, one bet	FULL EDITION

25	13	69	32	10/10
PRODUCTION ACCOUNTS	ENTERPRISES · 4+ REGIONS	DAYS OF TELEMETRY	CLOUD REGIONS COVERED	HYPOTHESES CONFIRMED

Survey instruments: native AWS / Azure / GCP resource-discovery and cost APIs, plus a 315+ rule recommendation engine, applied uniformly. Window: March 20 – May 27, 2026. Published June 2026.

Three views: spend, footprint, trajectory.

Spend by provider tells a different story than account count. **AWS still takes two-thirds of every cloud dollar** — and account count and spend are not the same shape.



Footprint itself is a choice — **most enterprises have not made it consciously.** Two of the four patterns below reflect deliberate architecture. Two reflect accident.

TABLE 3.3 • FOOTPRINT PATTERNS • ANONYMIZED

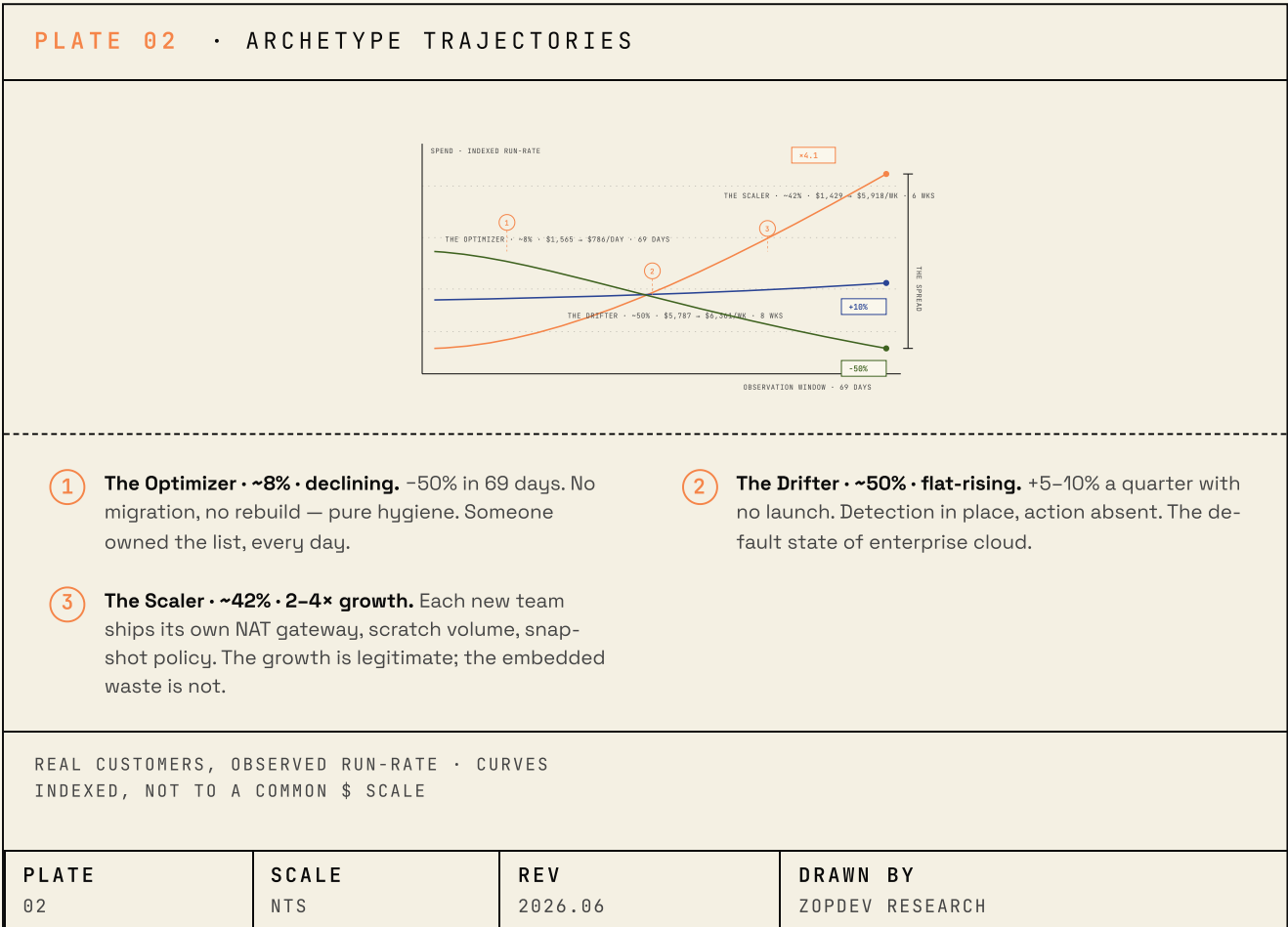
FOOTPRINT PATTERN	EXAMPLE	REGIONS	\$ / REGION / MO
Global sprawl, one tenant	Global sporting goods · Azure	32	~\$60
Concentrated production	Global CPG · Azure	3	~\$5,900
Single-region monolith	Consumer internet · AWS	2	~\$12,500
Spread without intent	E-commerce · AWS	19	~\$560

3.1

THE BIMODAL ECONOMY

Three archetypes, three trajectories.

The “average enterprise cloud bill” is a meaningless number. In the same quarter, **one customer halved spend while another quadrupled it**. The story is in the spread.



Which trajectory are we on — and is it the one we chose?

THE QUESTION EVERY CTO SHOULD ASK • CH 03

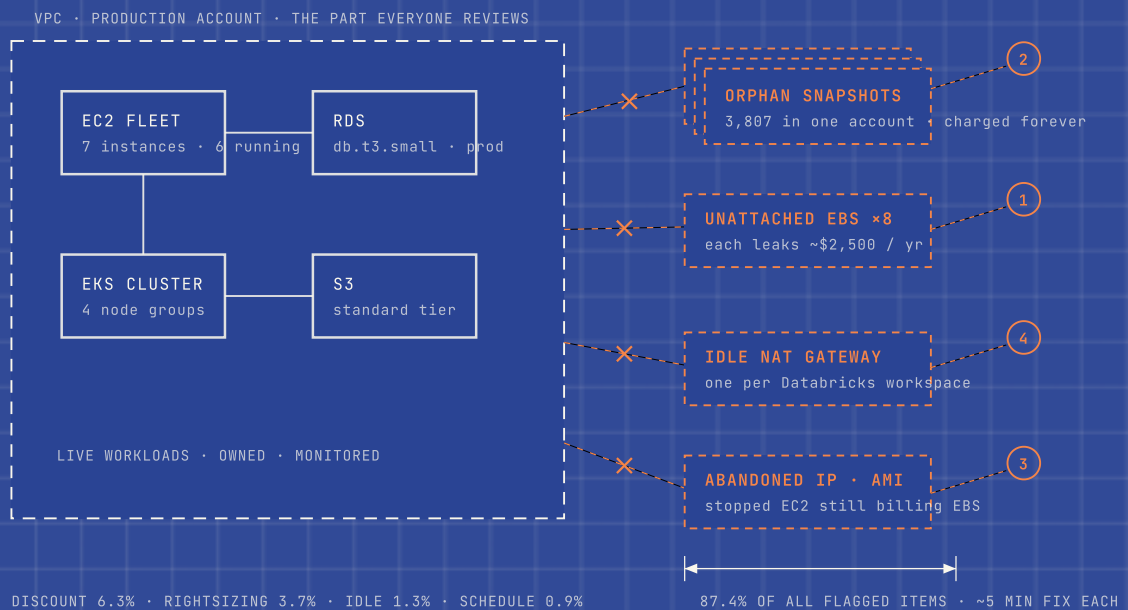
04

WHERE THE WASTE LIVES

87% of cloud waste isn't strategic. It's forgotten.

Vendor narratives focus on Reserved Instances, Savings Plans, Spot — only 6.3% of what we flagged. **The real waste is things nobody remembers creating.**

PLATE 03 • WASTE TOPOLOGY



- 1 Orphans dominate.** Provisioning friction is near zero; de-provisioning friction is enormous. Nobody owns the lifecycle.
- 3 You didn't buy the wrong thing.** You forgot to delete the right thing. Strategic discount findings are only 6.3% of the total.

- 2 Snapshots compound silently.** Every dev snapshots before risky changes; almost no one deletes. Snapshots can exceed live volumes by 14x.
- 4 Sprawl ships waste by default.** Each new workspace spins up its own NAT gateway and Premium SSD scratch volumes.

SHARE OF FLAGGED ITEMS, COHORT-WIDE • 315+
DETECTION RULES APPLIED UNIFORMLY

PLATE
03

SCALE
NTS

REV
2026.06

DRAWN BY
ZOPDEV RESEARCH

4.1

THE HALL OF SHAME

Fix ten things. Recover 80% of the waste.

None of these are strategic decisions to revisit. They are hygiene to install. If your platform team cannot recite this list from memory, you have a 16.9% problem.

TABLE 4.2 • ANTI-PATTERN SCHEDULE • UNIVERSALITY & ANNUAL STAKE

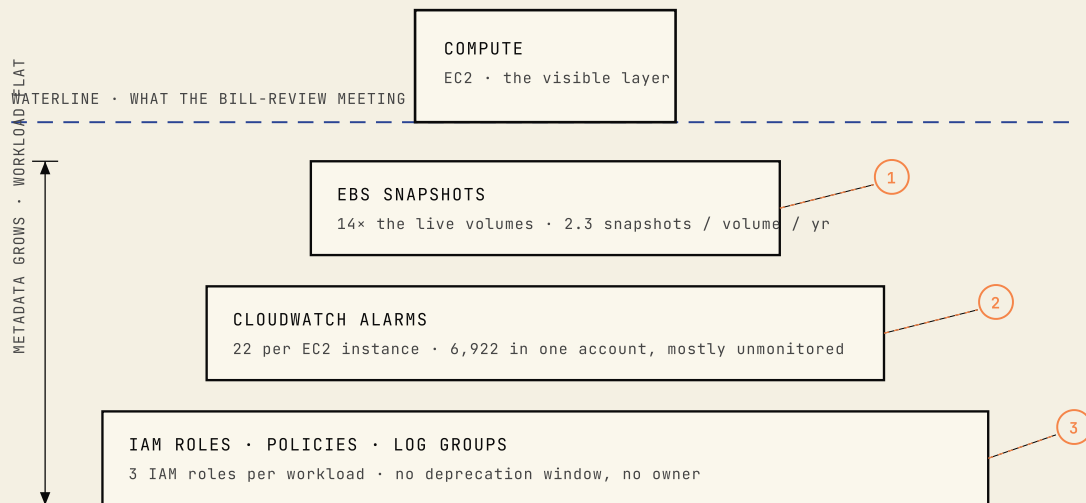
#	ANTI-PATTERN	UNIVERSALITY	\$ / YR AT STAKE
01	Orphan EBS snapshots — no source volume, charged forever	100% of AWS users	\$80K+
02	Unattached EBS volumes — single volumes leak \$2,500/yr	100% of AWS users	\$25K+
03	Premium SSD in non-prod Databricks (Azure)	100% of Databricks on Azure	\$66K+
04	EC2 not covered by Savings Plans	~90% of prod fleets	\$50K+
05	Single-AZ production RDS — not cost, risk	Every AWS enterprise	Resilience risk
06	Dev/test workloads with no schedule	100% of dev environments	\$40K+
07	Windows VMs not using Azure Hybrid Benefit	Common in regulated industries	40% on license
08	On-demand for stateless / batch workloads	Common in cloud-native shops	\$30K+
09	x86 where Graviton / ARM would work	80% of compute fleets	\$25K+
10	Stopped instances still incurring EBS charges	Found in every account	\$5K+

10 fixes → 80%
OF RECOVERABLE WASTE • BEFORE ANY ARCHITECTURE CHANGE

Resource sprawl is the new technical debt.

For CTOs and cloud architects. The visible compute layer of any production account is dwarfed by its metadata — **and the metadata, unlike the compute, has no lifecycle.**

PLATE 04 • THE METADATA ICEBERG — SECTION VIEW



① **The compute is the tip.** Snapshots outnumber live volumes 14x, compounding at 2.3 per volume per year.

② **The observability tax.** 6,922 alarms in one production account — no SNS destination, no on-call listening. Noise generators paying their own bill.

③ **Treat monitoring like production:** lifecycle policies, deprecation windows, ownership tags.

SECTION THROUGH A TYPICAL PRODUCTION AWS ACCOUNT • STRATA WIDTHS $\propto$ ITEM COUNT

PLATE
04

SCALE
NTS

REV
2026.06

DRAWN BY
ZOPDEV RESEARCH

6.1

THE FRAGMENTATION PROBLEM (AZURE)

Consolidators vs Fragmenters.

One customer ran 33 distinct Azure subscriptions in a single tenant. Another ran 3 for comparable spend. The first is governance debt waiting to compound. The second is governance done right.

FIG 6.2A • CONSOLIDATOR

Consolidators

3-5 LARGE SUBSCRIPTIONS

Cost / subscription / mo	~\$15-20K
Governance	Tractable
Cleanup	Feasible

FIG 6.2B • FRAGMENTER

Fragmenters

20-35 SMALL SUBSCRIPTIONS

Cost / subscription / mo	~\$500-1,500
Governance	Scales with team count
Cleanup	Hopeless

The most visible symptom of fragmentation: **Databricks workspace proliferation** — each workspace spinning up its own NAT gateway and Premium SSD scratch volumes. Confirmed across every Databricks-on-Azure enterprise in the cohort.

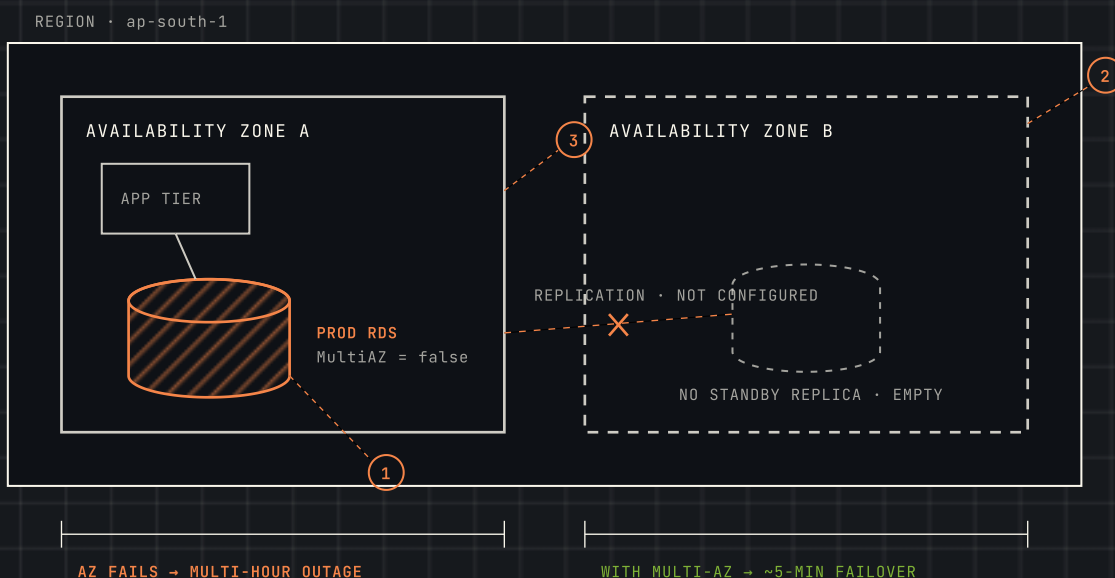
6.2

RESILIENCE THEATER

Every AWS enterprise had at least one Single-AZ production database.

The reason is never "we made a deliberate choice." It is always "we ran the migration script in 2021 and never went back."

PLATE 05 · SINGLE-AZ RISK — PLAN VIEW



- ① **Hazard: single point of failure.** 36+ Single-AZ production databases flagged in one customer; the pattern is visible everywhere.
- ③ **Detection rule RC-005:** MultiAZ = false AND identifier matches production pattern. Fired in every AWS enterprise in the cohort.

- ② **The fix is trivial.** A small uptick in storage buys the difference between a 5-minute failover and a multi-hour outage.

HATCHED REGION INDICATES UNMITIGATED
AVAILABILITY HAZARD · RULE RC-005

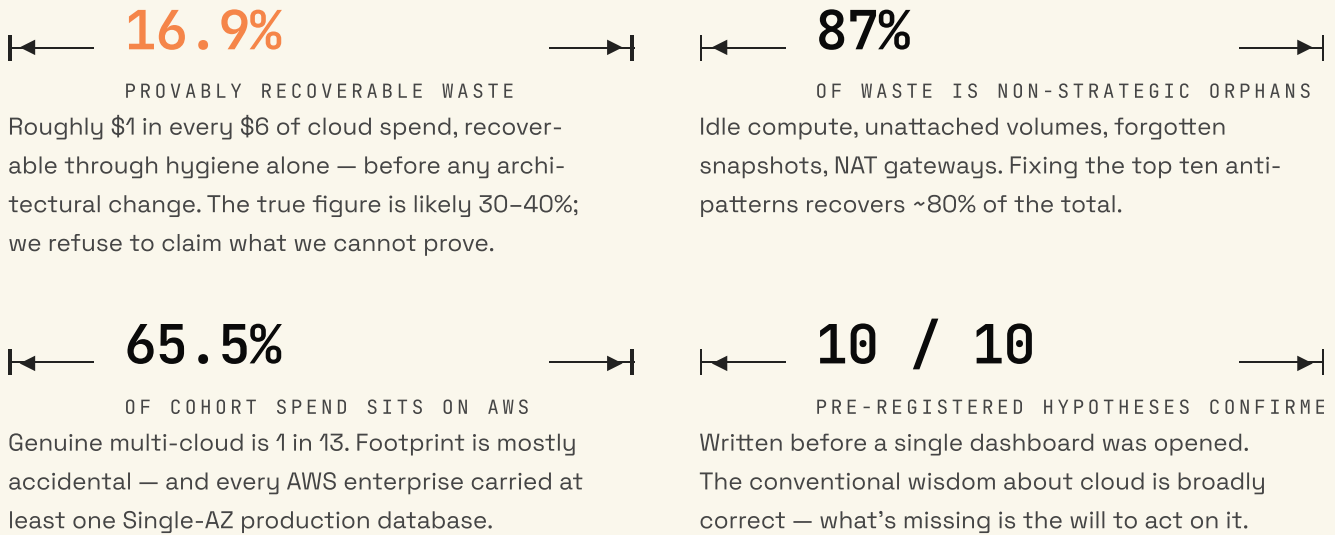
PLATE
05

SCALE
NTS

REV
2026.06

DRAWN BY
ZOPDEV RESEARCH

The dossier, in four dimensions.



The boardroom question is not "can we cut cloud spend?"
It is "why haven't we already?"

■ CLOSING • ISSUED FOR REVIEW

The most important thing you can do with this drawing set is challenge it.

We published the full methodology so you can replicate, contest, and improve it. 13 enterprises is a small sample; when we say 16.9% we mean the central estimate — the true industry-wide value is probably in the 12–22% range. Challenges: contact@zop.dev, 14-day response window.

One bet, willing to be publicly wrong: by the 2027 edition, more than half of this cohort crosses into Optimizer territory — not because tooling improved, but because the next downturn makes cloud cost a board priority.

THE STATE OF CLOUD 2026 • AN EMERGING MARKETS VIEW • EDITION 1 • JUNE 2026

SHEET 12 OF 12	REV 2026.06	STATUS ISSUED FOR REVIEW	DRAWN BY ZOPDEV • INDUSTRY RESEARCH
-------------------	----------------	-----------------------------	--