OpenSearchCon INDIA

I've been paged at 3 AM.

Have you?

Building HA & DR for OpenSearch : the hard way, and the right way.

Presented by

Ruchikka Chaudhary
Solution Architect II,
AWS

Rishav Kumar
Software Development Engineer,
Amazon

Agenda

What we'll cover today

OpenSearchCon INDIA

01

Why HA & DR Matter

Search on the critical path

02

Cluster Fundamentals

Building blocks of resilience

03

Multi-AZ Patterns

In-region high availability

04

Shards & Replicas

Allocation and sizing

05

Snapshots & Backup

Point-in-time recovery

06

Cross-Cluster Replication

Active replication across clusters

07

DR Architecture Patterns

Cross-region strategies

08

Automated Failover

Reducing manual intervention

09

Monitoring Cluster Health

Detecting issues early

10

Real-World Lessons

Best practices from production

Why High Availability & Disaster Recovery Matter for Search

OpenSearchCon INDIA

Search is no longer a 'nice to have' — it's on the critical path

Search powers customer-facing experiences: e-commerce, logs, support, security.
An OpenSearch outage means blind SOCs, broken dashboards, and lost revenue.
Compliance and SLAs increasingly mandate measurable RTO/RPO targets.

$5,600

Per Minute

Average enterprise downtime cost (Gartner)

15min

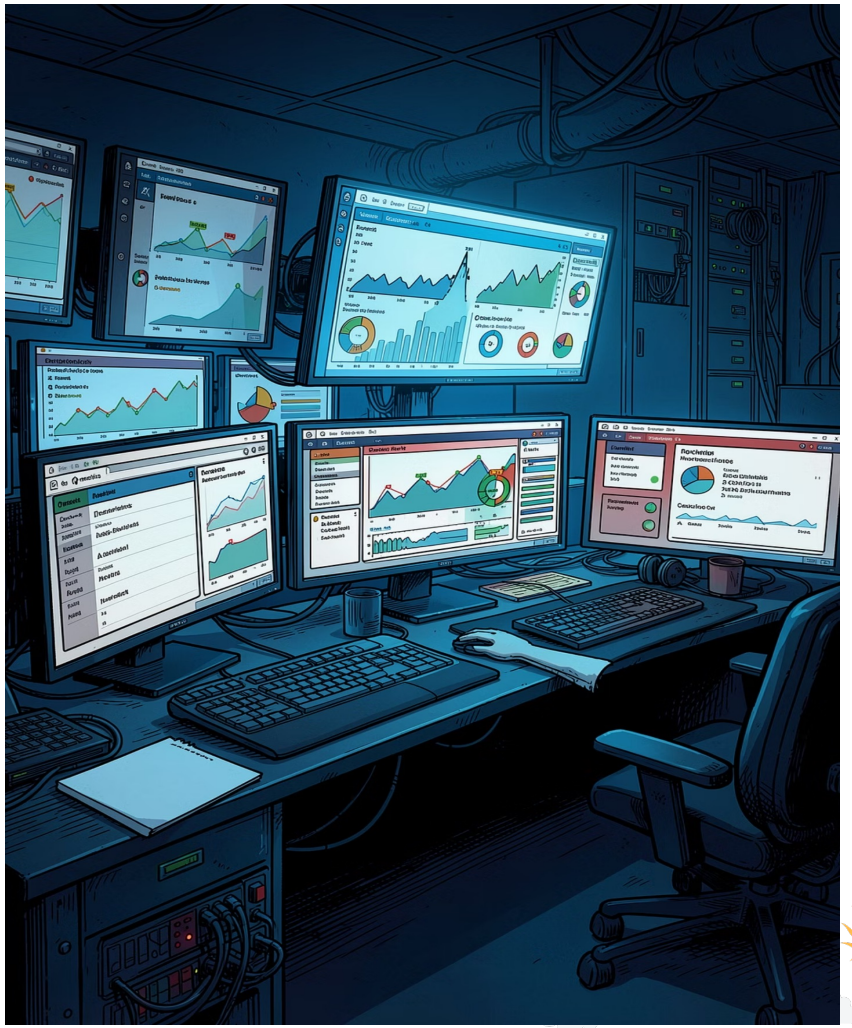
RTO Target

Recovery Time Objective for Tier-1 search workloads

5min

RPO Target

Recovery Point Objective for Tier-1 search workloads



HA vs DR — Two Different Problems

Solved by different mechanisms, often confused

OpenSearchCon INDIA

HIGH AVAILABILITY

Survives node, AZ, or rack failure

- Same region, same logical cluster
- Replicas + dedicated Cluster Managers + AZ awareness
- Recovery is automatic — seconds to minutes
- Goal: zero customer-visible downtime

DISASTER RECOVERY

Survives region-wide outage or data corruption

- Cross-region or cross-cluster
- Snapshots + Cross-Cluster Replication
- Recovery is intentional — minutes to hours
- Goal: bounded data loss, predictable RTO

OpenSearch Cluster Fundamentals

The building blocks every resilient design depends on

OpenSearchCon INDIA

Cluster Manager

Coordinates cluster state — never serves data. Always run dedicated, odd-numbered (3) to avoid split-brain.

Data Nodes

Hold shards, run search and indexing. Scale independently for throughput.

Coordinating Nodes

Route requests, aggregate results — act as search load balancers for read-heavy workloads.

Ingest Nodes

Pre-process documents before indexing. Isolate heavy parsing workloads from data nodes.

An OpenSearch cluster without a manager node is like Hogwarts without a Headmaster.

Multi-AZ Cluster Topology

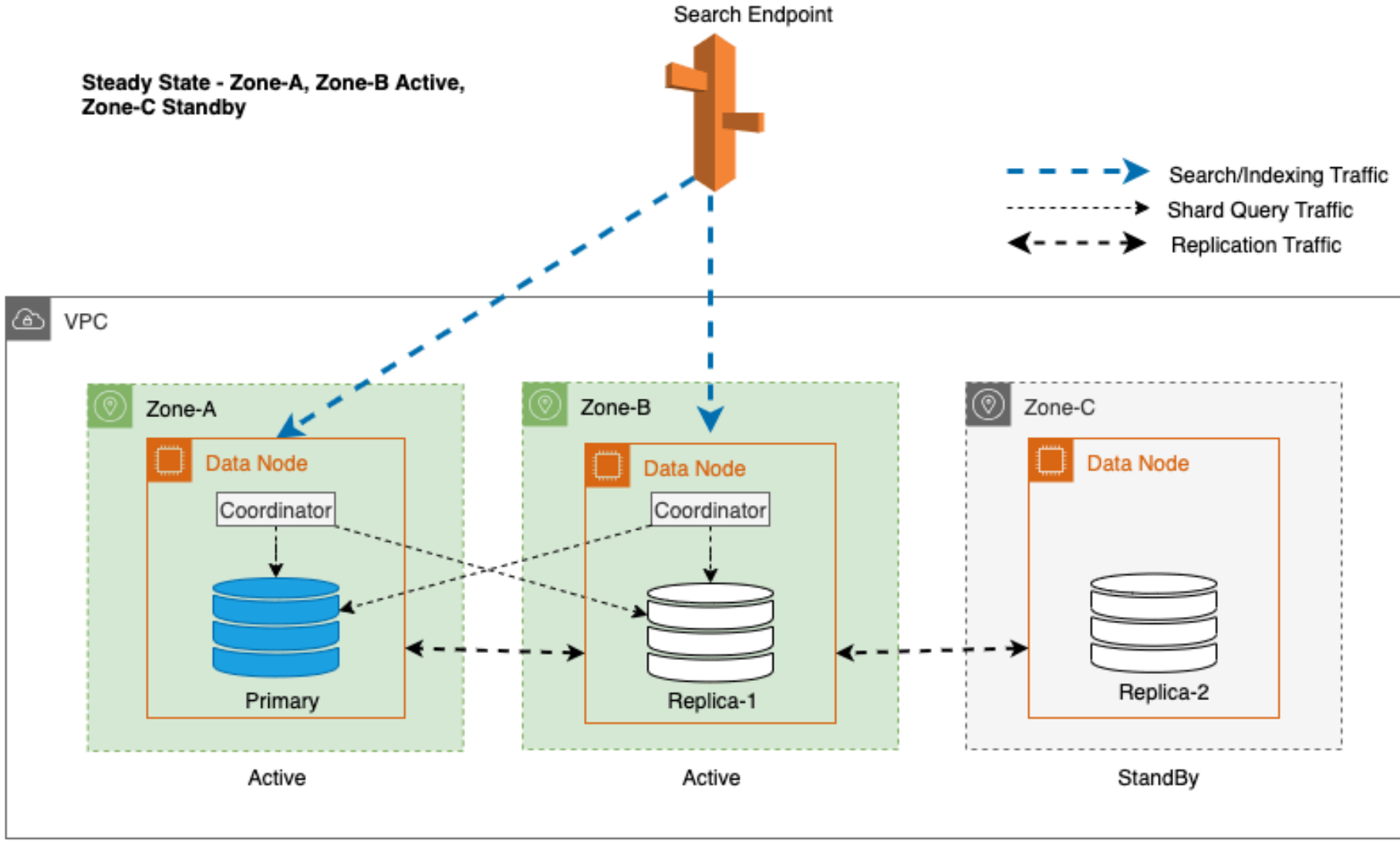
The reference architecture for in-region HA

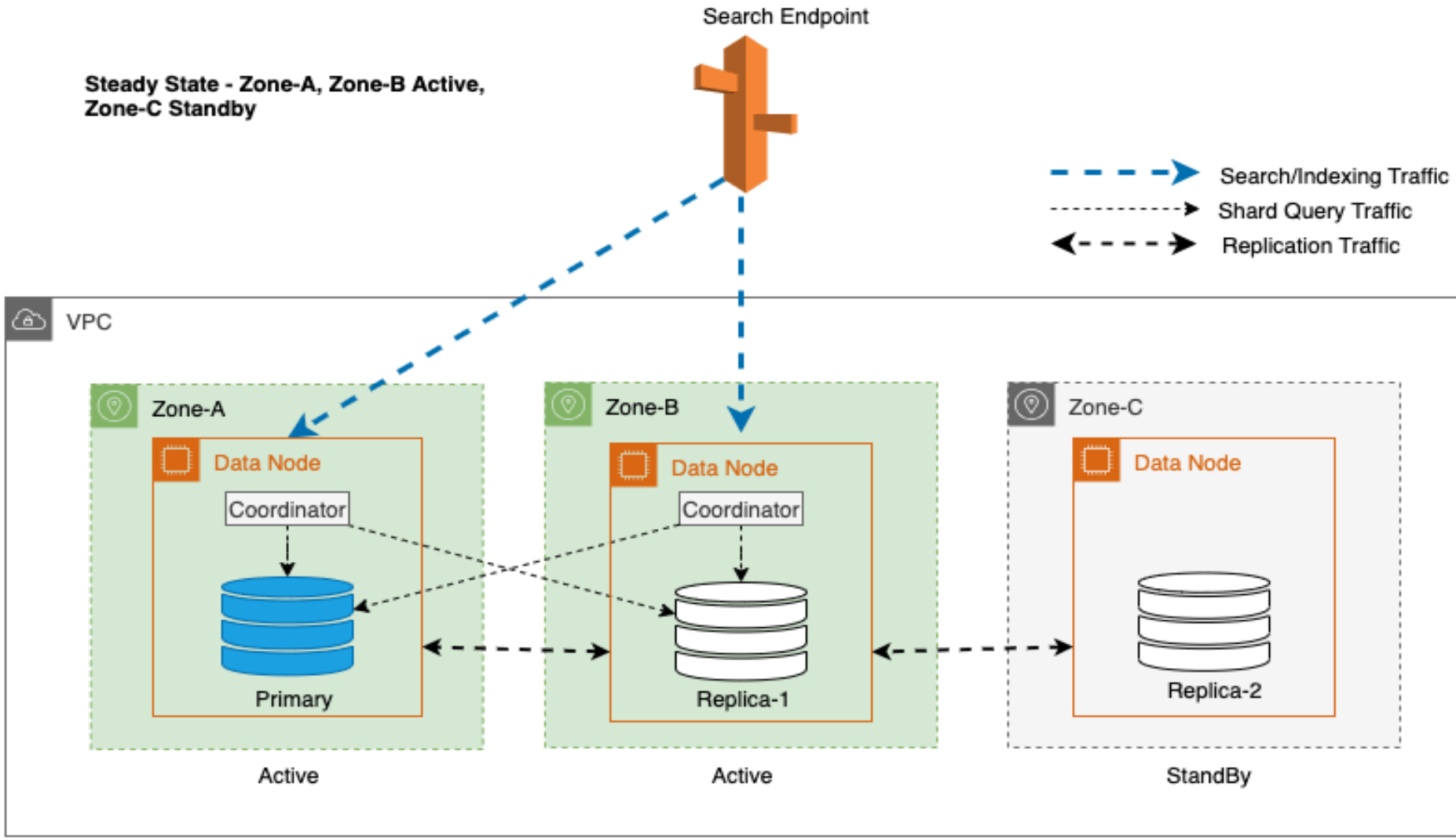
OpenSearchCon INDIA

AZ-a	AZ-b	AZ-c
Cluster Manager	Cluster Manager	Cluster Manager
Data node 1	Data node 1	Data node 1
Data node 2	Data node 2	Data node 2
Coordinator	Coordinator	Coordinator

Spread shards across all 3 AZs | Replicas guarantee an AZ-loss survivable copy

Multi-AZ Architecture Patterns




Steady State - Zone-A, Zone-B Active, Zone-C Standby
Search Endpoint
Search/Indexing Traffic
Shard Query Traffic
Replication Traffic
VPC
Zone-A
Data Node
Coordinator
Primary
Active
Zone-B
Data Node
Coordinator
Replica-1
Active
Zone-C
Data Node
Replica-2
StandBy

Shard Allocation Awareness

Telling OpenSearch about your failure domains

Configuration Steps

- Tag every node with its zone attribute (e.g., node.attr.zone=az-a)
- Set cluster.routing.allocation.awareness.attributes: zone
- Forced awareness prevents shards from collapsing into one zone if others fail
- Combine with shard_balance_threshold to prevent hotspotting
- Validate with _cat/shards — primary and replica should never share a zone

opensearch.yml

```
node.attr.zone: az-a

cluster.routing.allocation.awareness
.attributes: zone

cluster.routing.allocation.awareness
.force.zone.values:
 az-a,az-b,az-c
```

Primaries, Replicas, and Sizing

How many shards? How many copies?

OpenSearchCon INDIA

- 1 primary + 1 replica = minimum for HA (2 copies)
- 1 primary + 2 replicas = strong reads, faster recovery, 3x storage
- Target shard size: 10–50 GB for most workloads
- Avoid oversharding — each shard adds Lucene overhead
- Use the index template to enforce defaults across new indices
- number_of_replicas can be tuned online; number_of_primaries cannot

Quick math

1 TB index

→ 30 GB shards = 34 primaries

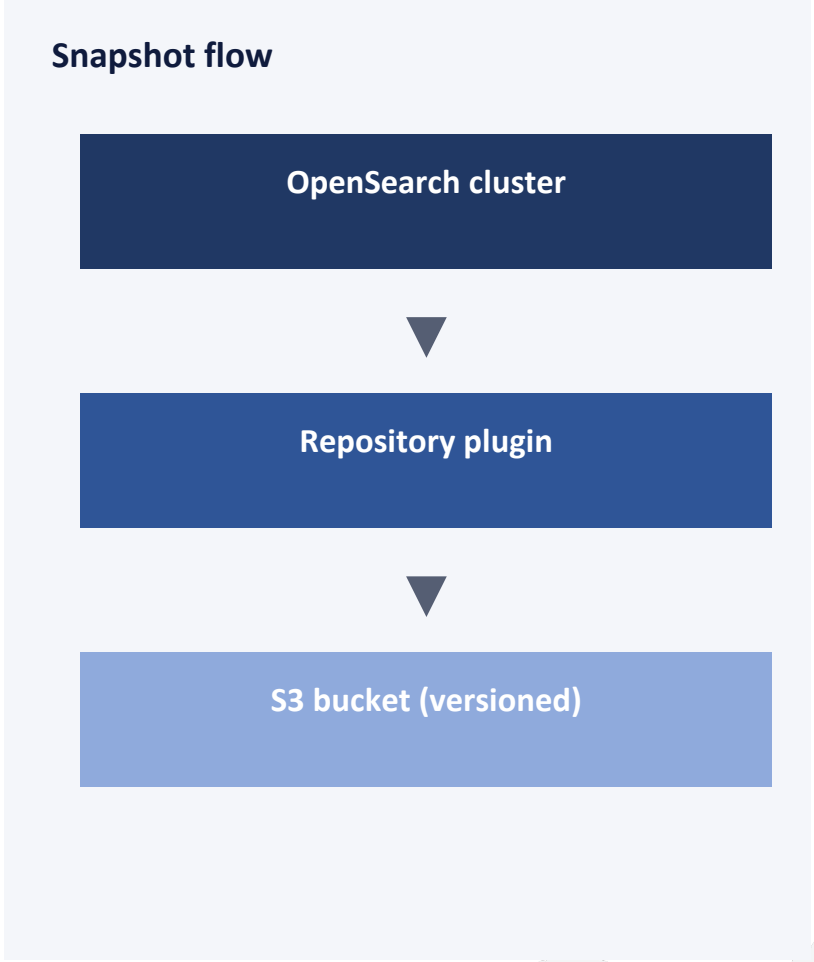
→ 1 replica = 2 TB on disk

→ Spread across ≥ 3 data nodes

Snapshots — Your Last Line of Defense

Incremental, point-in-time backups to durable object storage

- Snapshots are incremental and stored in a snapshot repository
- Common backends: S3, Azure Blob, GCS, shared FS, HDFS
- Use a separate bucket/prefix per cluster — never share writers
- Snapshot Management (SM) policy automates schedule + retention
- Always test restores — an untested backup is a hope, not a strategy



Cross-Cluster Replication

Continuous, low-latency replication between clusters

- Leader-follower model: writes go to leader, replicate to follower(s)
- Follower indices are read-only until failover is invoked
- Replicates per-index — pick what's worth the bandwidth
- RPO measured in seconds, not hours like snapshots
- Pairs perfectly with a warm DR region or a read-scaled secondary
- Stop replication on the follower to make it writable for failover

Active-Passive vs Active-Active

Different DR postures, different complexity bills

OpenSearchCon INDIA

Active-Passive

- One region serves all traffic
- Standby is read-only via CCR
- Simple, well-supported
- Failover is a deliberate decision
- RTO: minutes • RPO: seconds

Active-Active (read)

- Both regions serve reads
- Writes still go to one leader
- Lower latency for global users
- Requires routing logic
- RTO: seconds • RPO: seconds

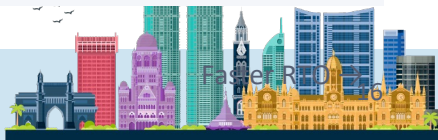
DR Architecture Patterns — Cost vs RTO

Pilot light, warm standby, hot standby

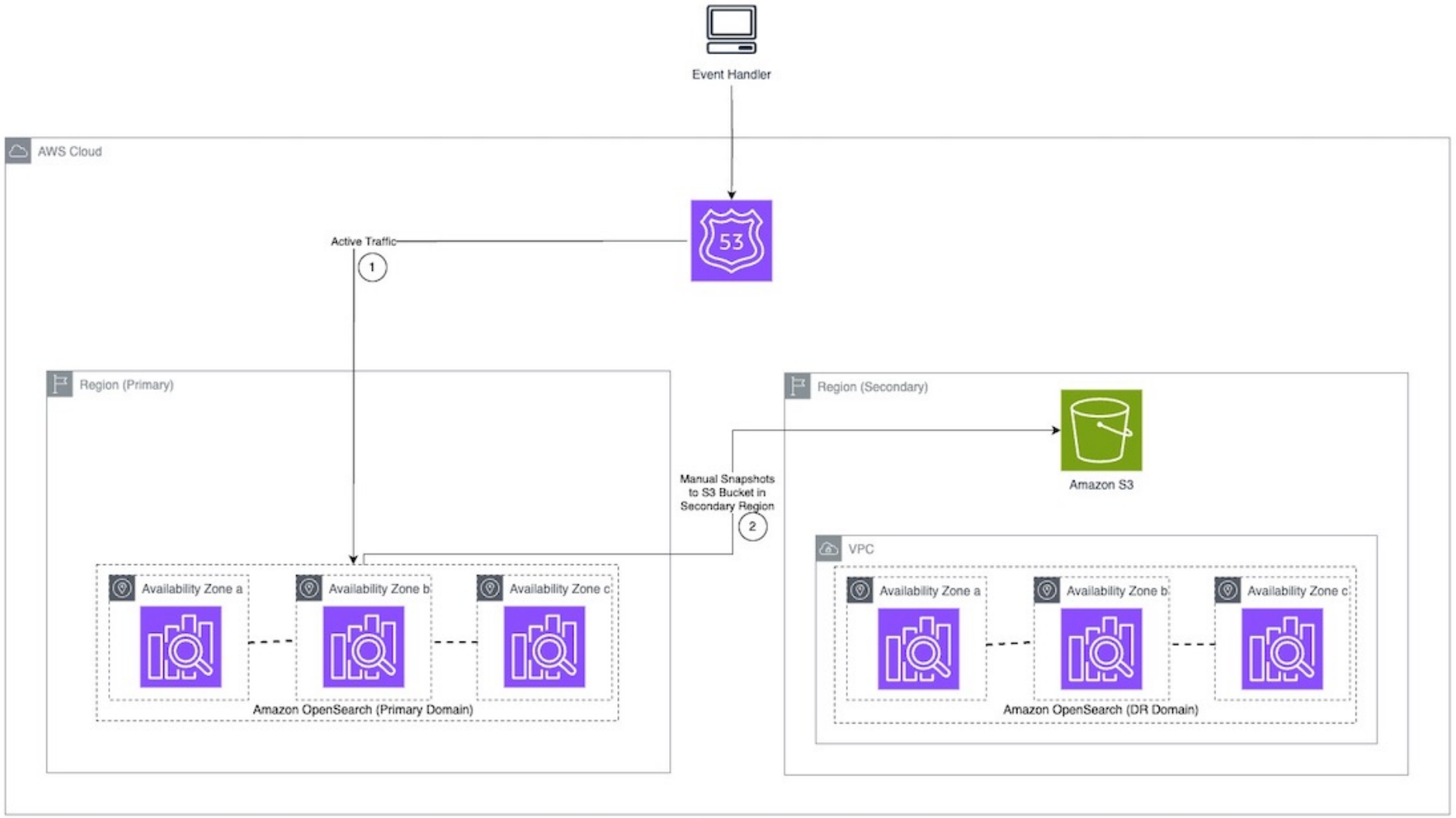
OpenSearchCon INDIA

Backup & Restore	Pilot Light	Warm Standby	Hot / Active-Active
RTO: hours **RPO: minutes-hours**	**RTO: < 1 hour** **RPO: minutes**	**RTO: < 15 min** **RPO: seconds**	**RTO: seconds** **RPO: seconds**
• Snapshots only • No standby cluster running • Provision + restore on demand • Cheapest • Acceptable for tier-3	• Tiny standby cluster idle • Snapshots replicated cross-region • Scale up + restore on failover • Low steady-state cost • Common for tier-2	• Right-sized standby running • CCR keeps follower fresh • Promote + scale up traffic • Moderate cost • Tier-1 default	• Both regions serve traffic • Symmetric capacity • DNS or service mesh routing • Highest cost • Tier-0 / global apps

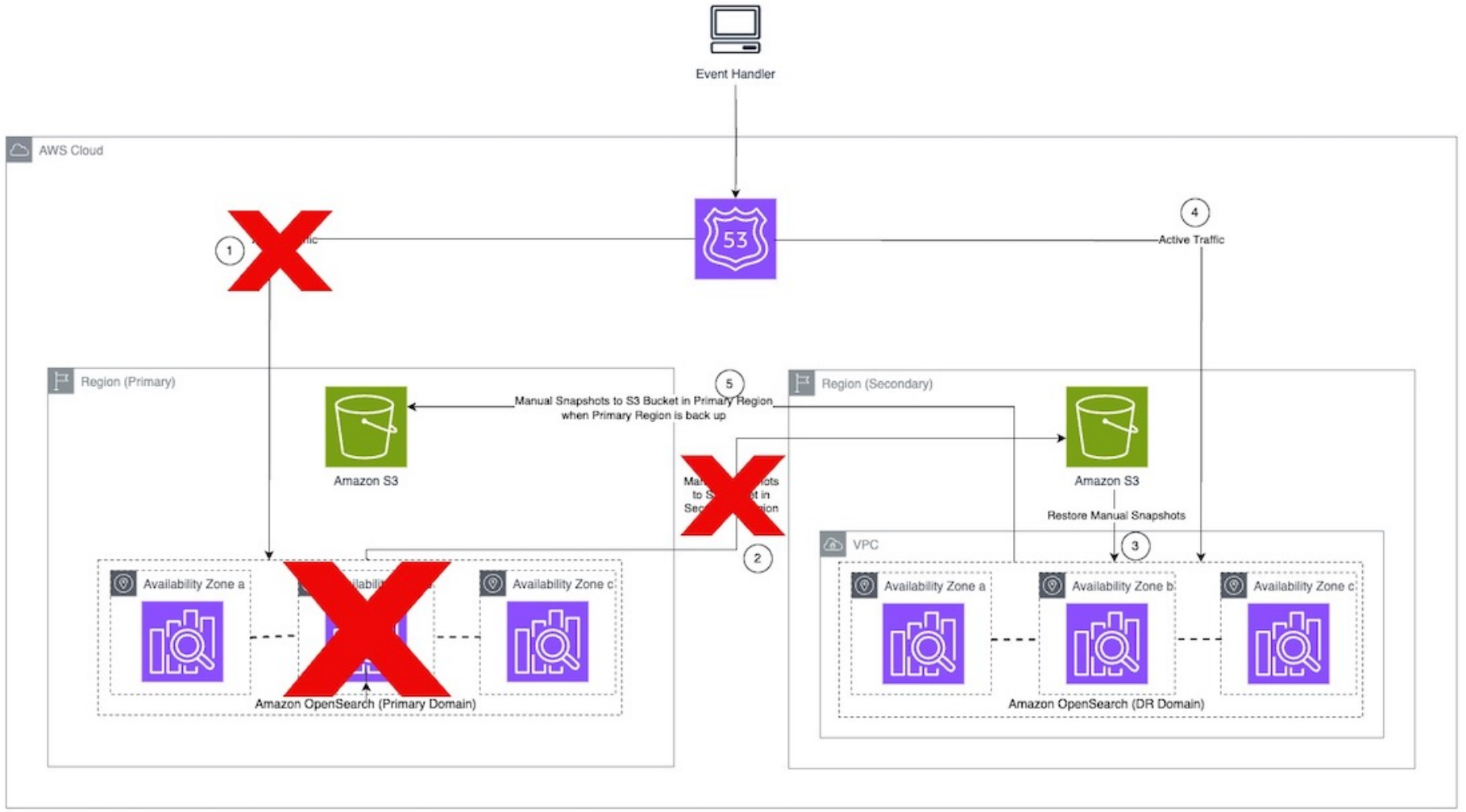
← Lower cost



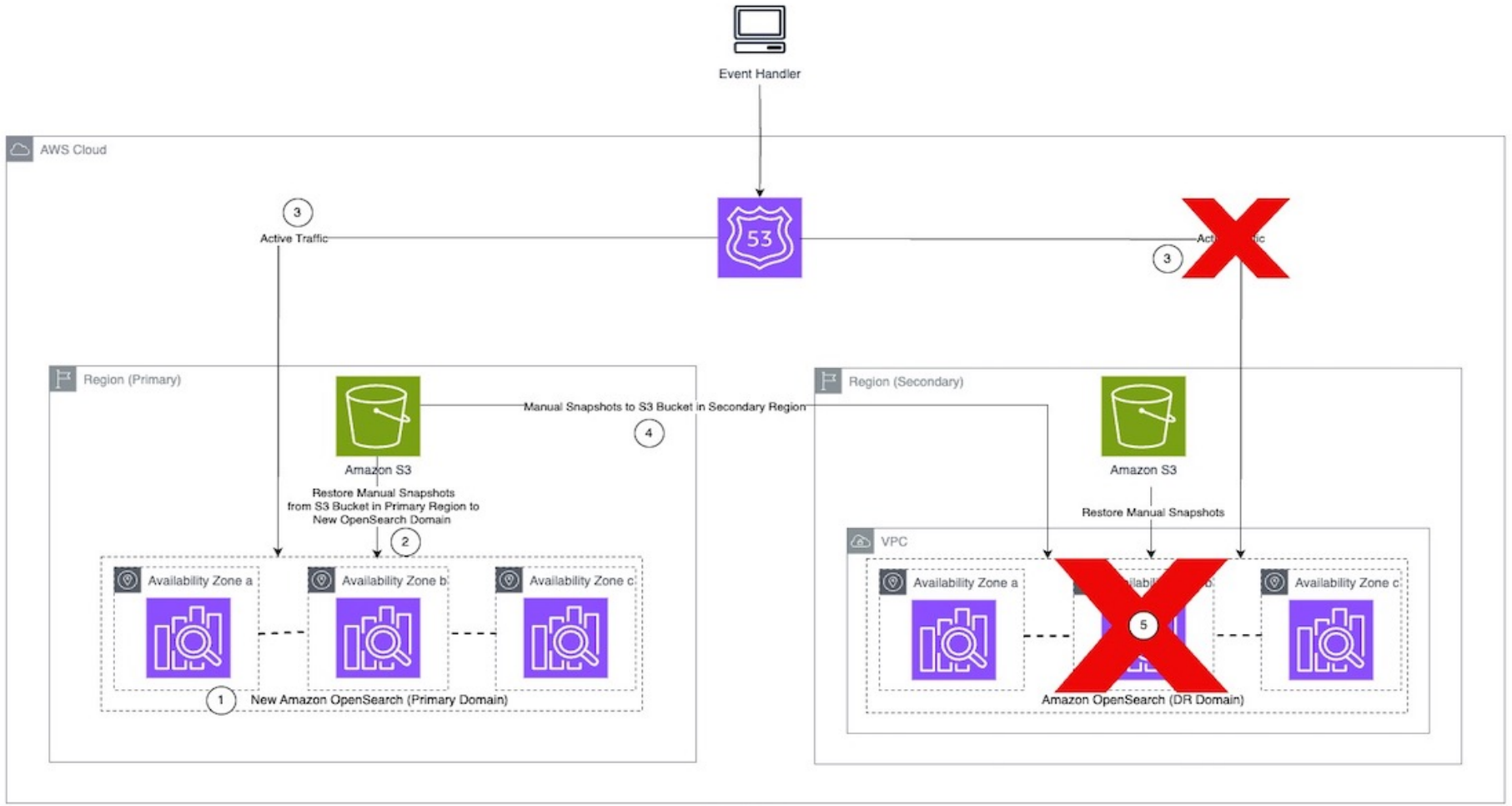
Snapshot based recovery : Reference Architecture



In the event of a disaster



Primary region becomes available



Reference Architecture — Multi-Region Warm Standby

End-to-end view of a tier-1 OpenSearch DR setup

OpenSearchCon INDIA

Clients → Global DNS / Health-checked endpoint (Route 53 / CloudFlare / NS1)

REGION A — Primary (active)

- **Ingest pipelines**
- **Application reads/writes**
- **OpenSearch cluster (3 AZs)**
  - 3 CM • 6 data • 3 coord
  - Allocation awareness ON • 1+ replica per index
- **Snapshot repo: s3://prod-snapshots-region-a**
- **Metrics → Prometheus / OpenSearch Dashboards**

CCR ⇒

S3 CRR ⇒

REGION B — DR (warm standby)

- **Ingest (parallel)**
- **Read-only (until failover)**
- **Follower cluster (3 AZs)**
  - Right-sized • CCR follower indices
  - Auto-scaling group ready to scale on promote
- **Snapshot repo: s3://prod-snapshots-region-b**
- **Same observability stack — pre-wired**

Monitoring Cluster Health

What to watch, before something goes wrong

OpenSearchCon INDIA

Cluster status

GREEN / YELLOW / RED — page on RED, escalate on YELLOW > 5m

Search latency p99

Watch tail latency, not averages

JVM heap

< 75% used; sustained > 85% = imminent GC pressure

Disk watermark

Stay under low watermark (85%); high (90%) blocks writes

Pending tasks

> 50 sustained = cluster manager bottleneck

Snapshot success rate

100% expected; one failure = page

Indexing latency p99

Track per index; correlate with merge & refresh activity

Replication lag

Per-index follower lag in seconds; alert > 60s sustained

Lessons Learnt

OpenSearchCon INDIA

The Silent Snapshot Failure : Real story: 6 months of 'green' backups, none restorable

What happened

- IAM role rotated; snapshot writes silently 403'd
- Repository status check missing from monitoring
- Discovered during a real incident requiring a restore
- Restored from a 7-month-old snapshot — multi-day data loss

Fix

- Page on snapshot failure events explicitly
- Quarterly restore drill — no exceptions
- Repository health check in synthetic monitoring
- Object Lock + bucket policies that survive role rotation
- Snapshot age SLO: 'newest snapshot < N minutes old'

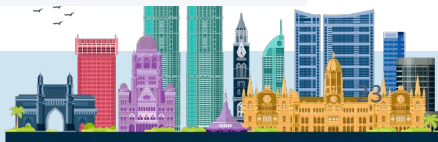
The Single-AZ Cluster That Wasn’t Multi-AZ in name only

What happened

- Cluster ran across 3 AZs — but no allocation awareness
- All replicas of one hot index ended up in AZ-a
- AZ-a outage took the index offline despite multi-AZ topology
- 30 minutes of search errors before traffic shifted

Fix

- Set node.attr.zone on every node
- Enable forced awareness with all 3 zone values
- Validate post-deploy: _cat/shards | grep <index>
- Add a synthetic 'zone diversity' test to CI
- Game-day: simulate AZ loss and watch behavior



The Failover That Failed Open : Auto-promoted, but with stale data

What happened

- Network partition between regions tripped auto-failover
- CCR follower was 4 minutes behind
- Promoted follower; primary recovered moments later
- Result: split-brain writes, manual reconciliation

Fix

- Region failover requires human approval — always
- Replication-lag gate before promotion
- Fence the old leader before flipping DNS
- Idempotent ingest pipelines so replays are safe
- Quarterly fail-back drill, not just fail-over

OpenSearchCon INDIA

Thank you!

Questions, war stories, and follow-ups welcome.


OpenSearchCon
INDIA