
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

ARC312

Using cell-based architectures for resilient design on AWS

Anand Gaitonde
(he/him)
Principal Solutions Architect
AWS

Anuj Gupta
(he/him)
Principal Solutions Architect
AWS

Agenda

01 Why cell-based architecture?

02 Key concepts

03 Cellular design considerations

04 Q&A/discussion

Why cell-based architecture (CBA)?

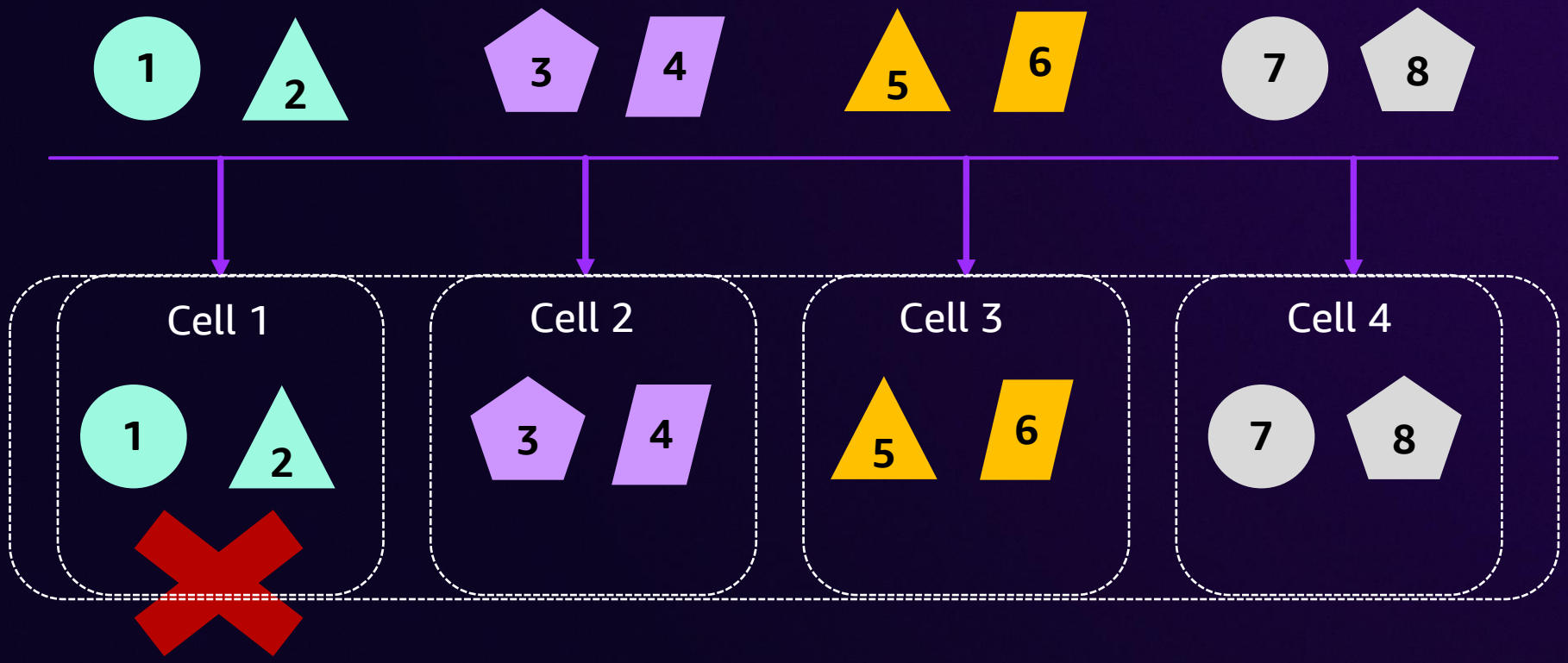
Everything fails, all the time.

Dr. Werner Vogels

VP and CTO at Amazon.com

Customer impact

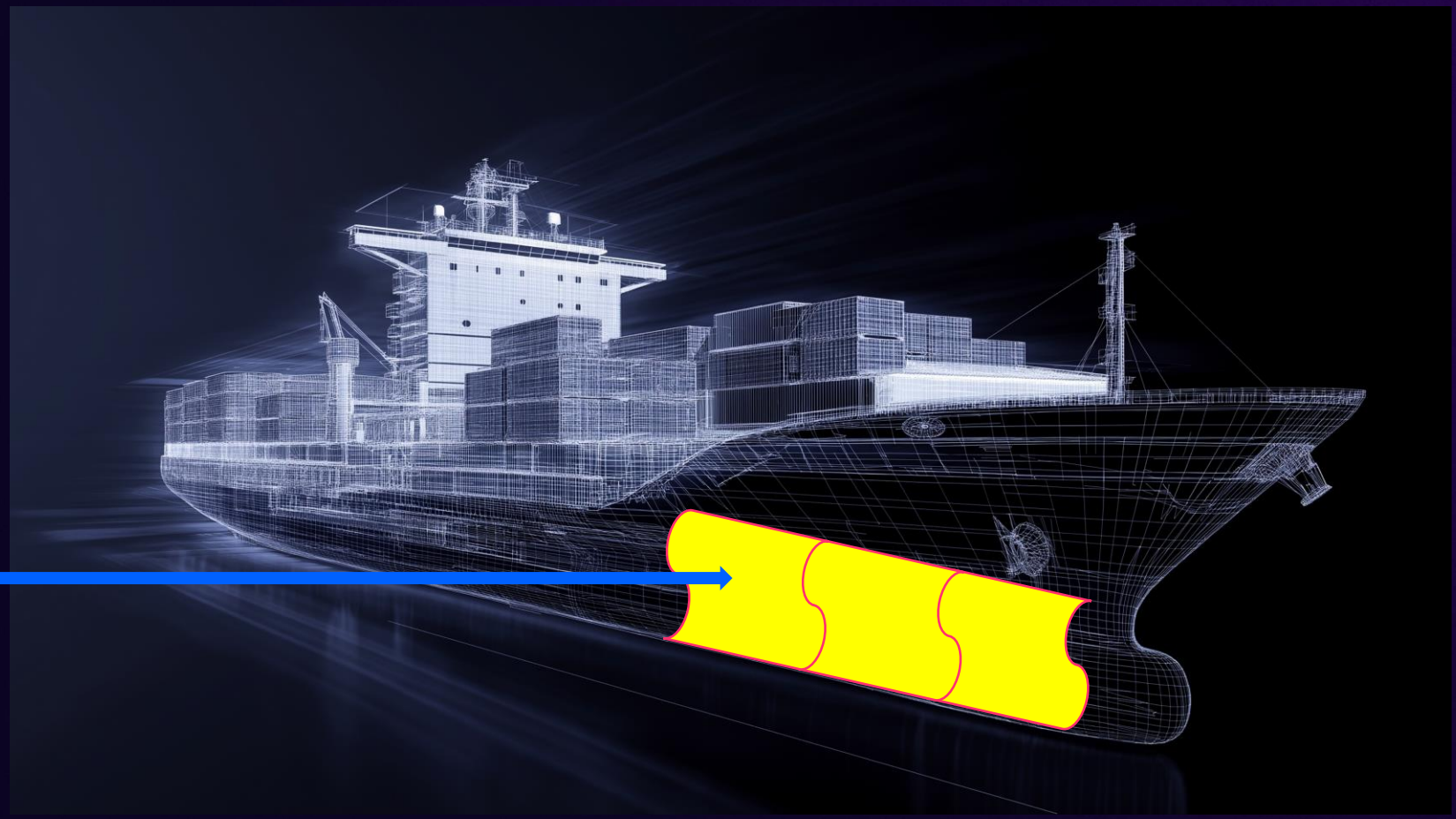
FAILURE ISOLATION



$$\text{Blast radius} = \frac{|\text{customers in each cell}|}{|\text{customers}|} = 2/8 = 25\%$$

Compartmentation

DIVIDING SPACES INTO SEPARATE SECTIONS

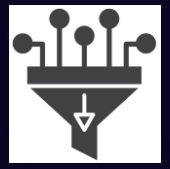


Minimize impact

Code deployments and configuration
(e.g., bad deployment, cred expiration)

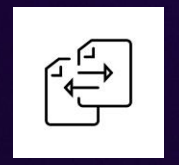
Dependencies
(e.g., external APIs)

Data and state
(e.g., data corruption)



Service quotas

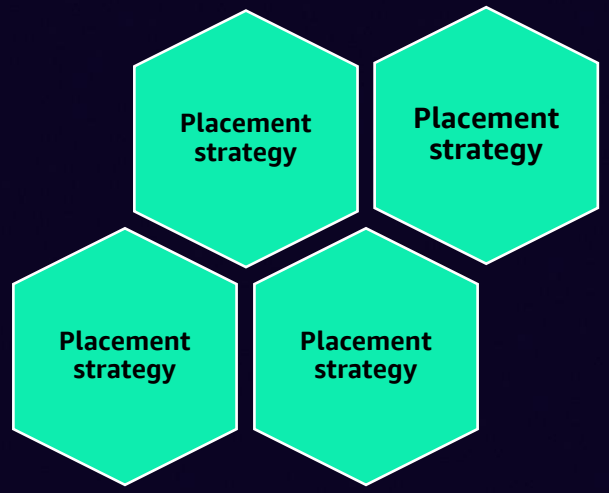
Poison pill



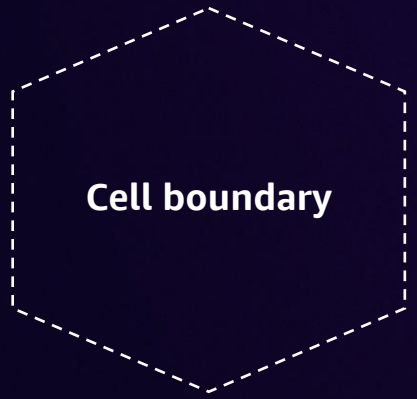
Noisy neighbor

Cell-based design components

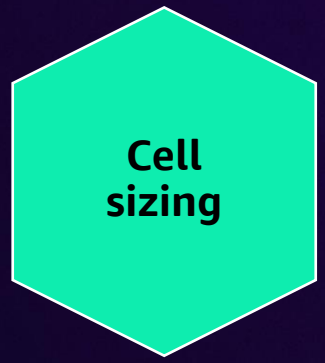
CONCEPTS



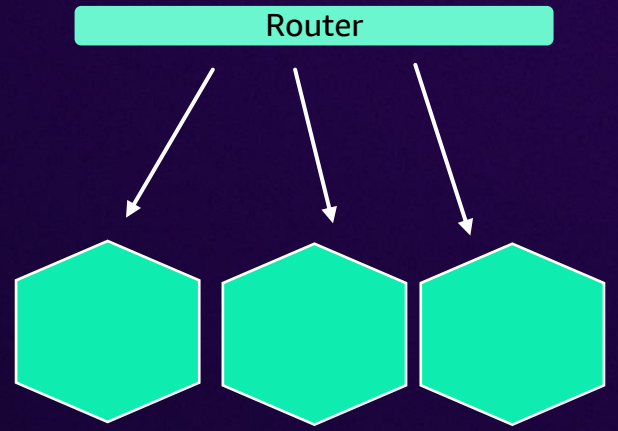
Hashing, range, geography



Region, AZ, VPC, AWS account

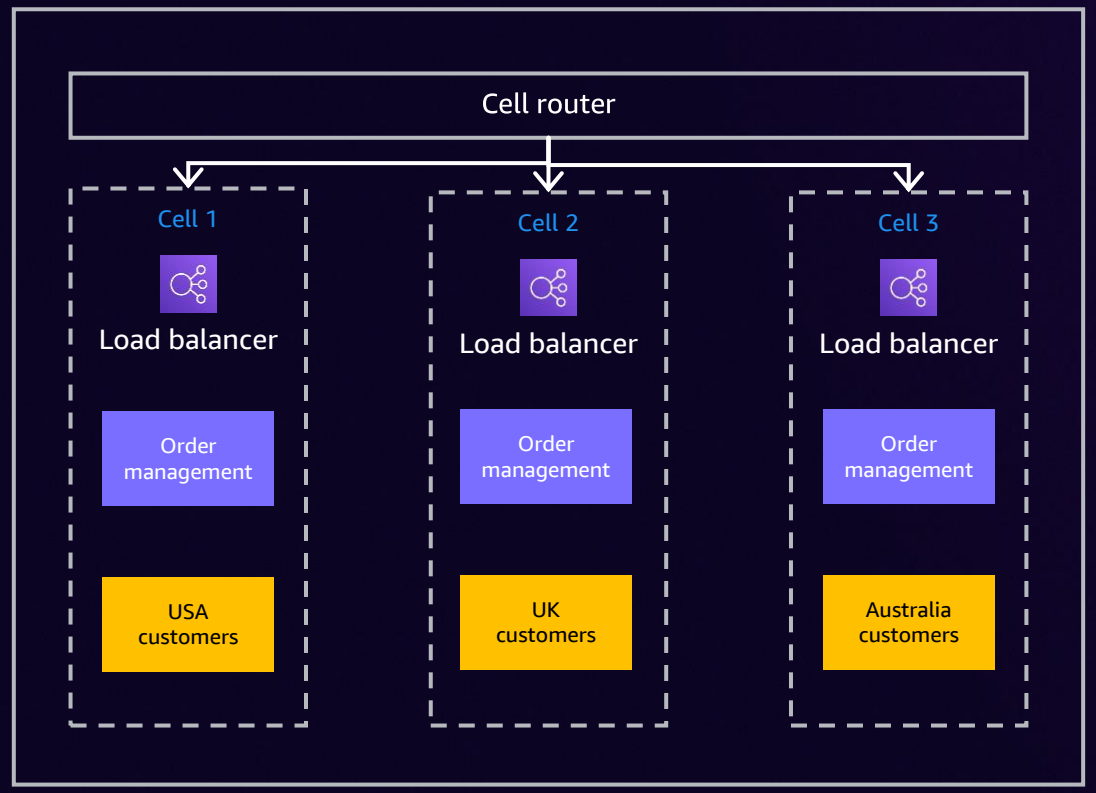


Blast radius, resource utilization

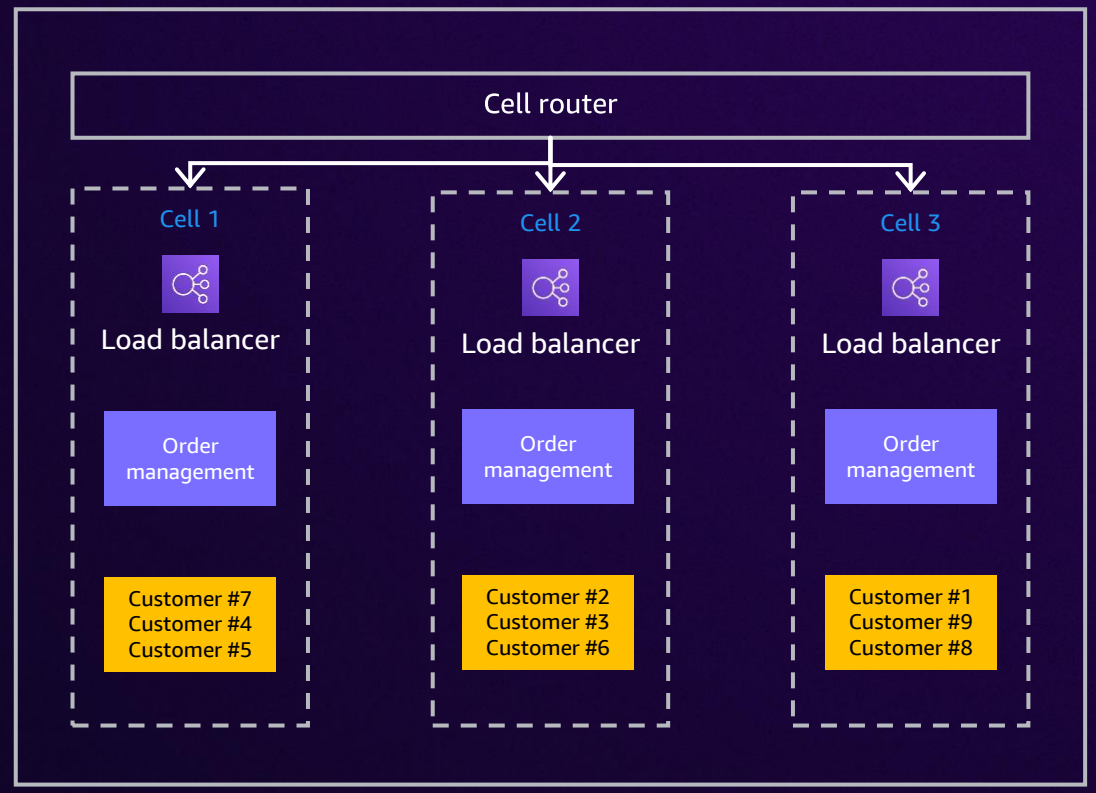


Discovery mechanism, proxy service

Cell partitioning strategy



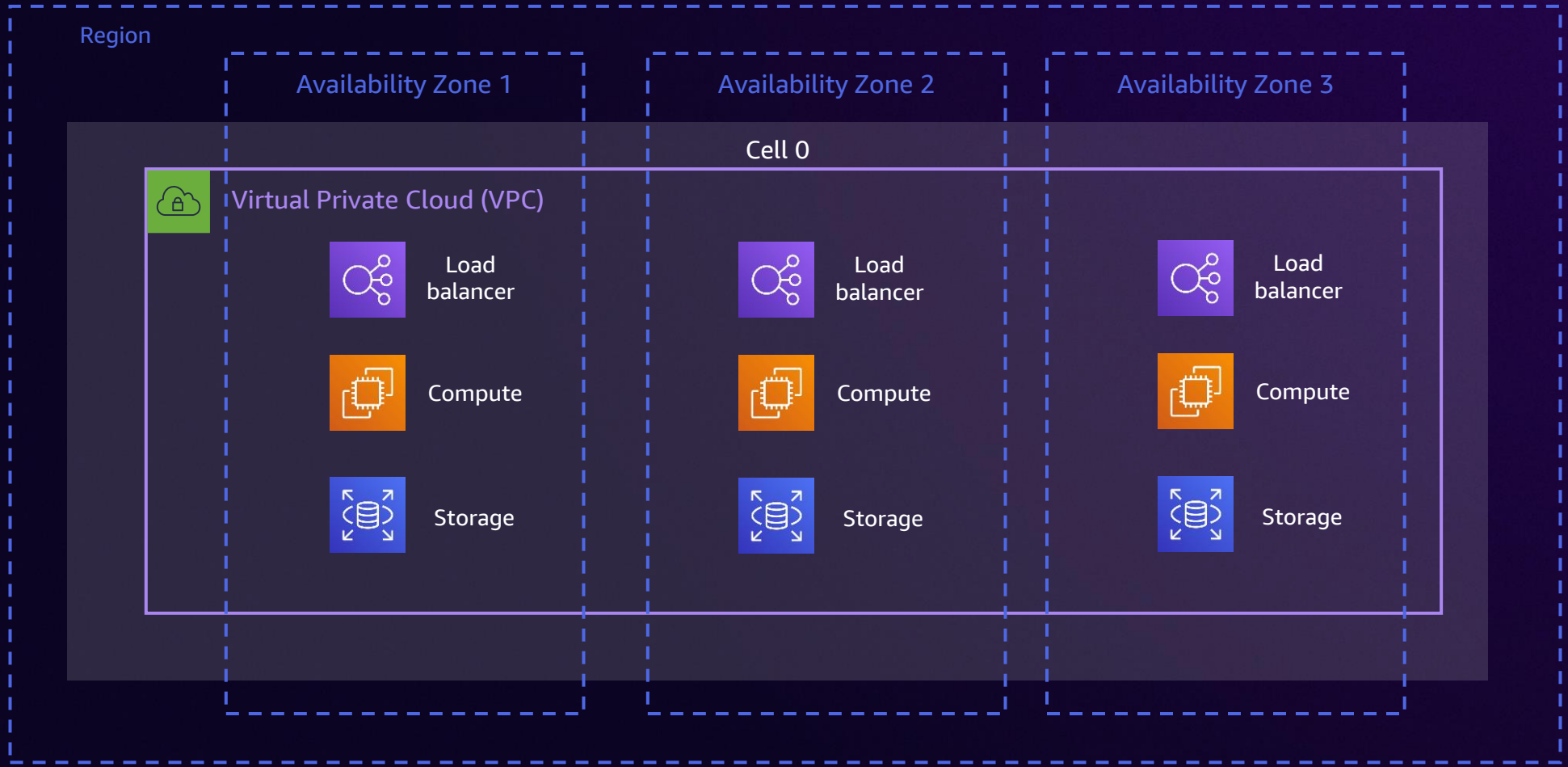
Geo-based cells



Customer-based cells

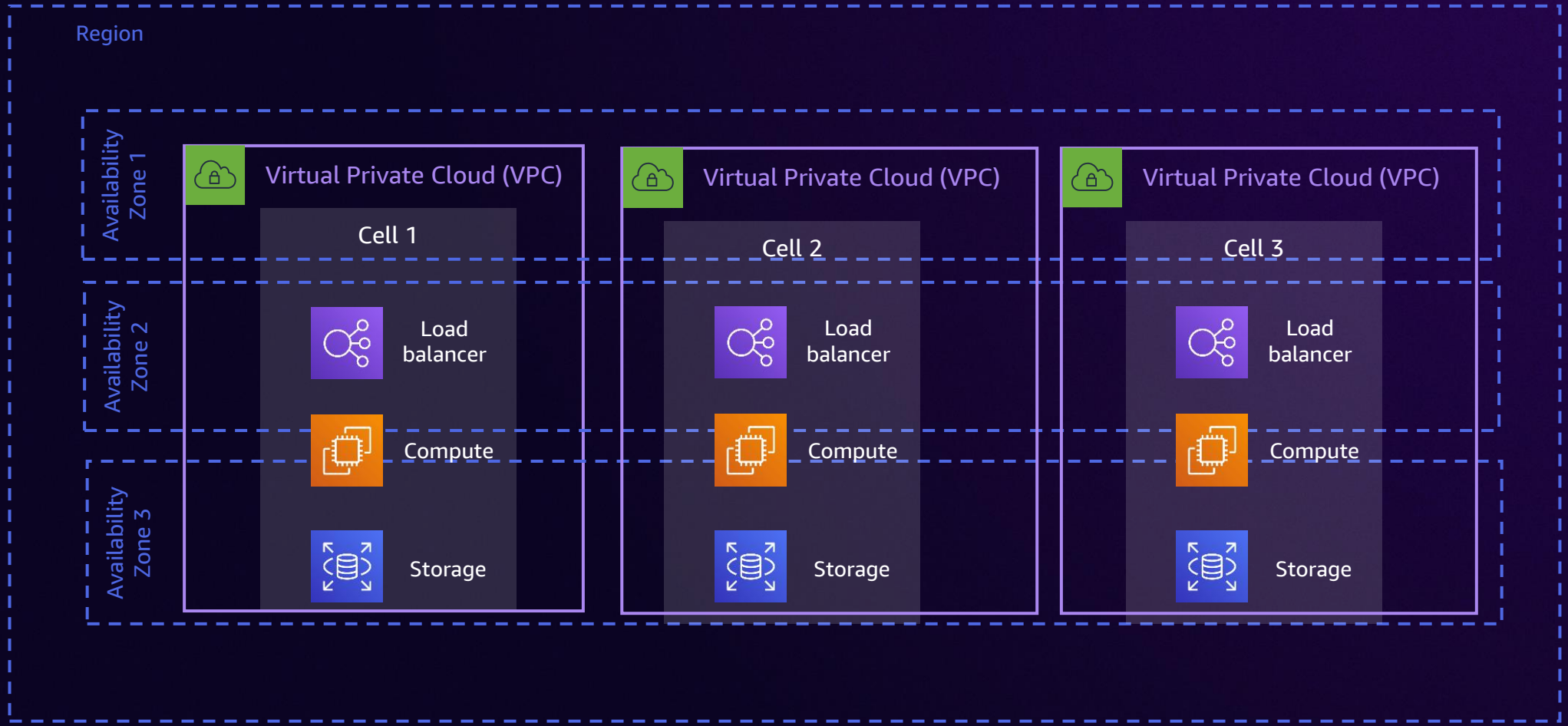
Cell boundary

BUILDING CELLS WITH HA



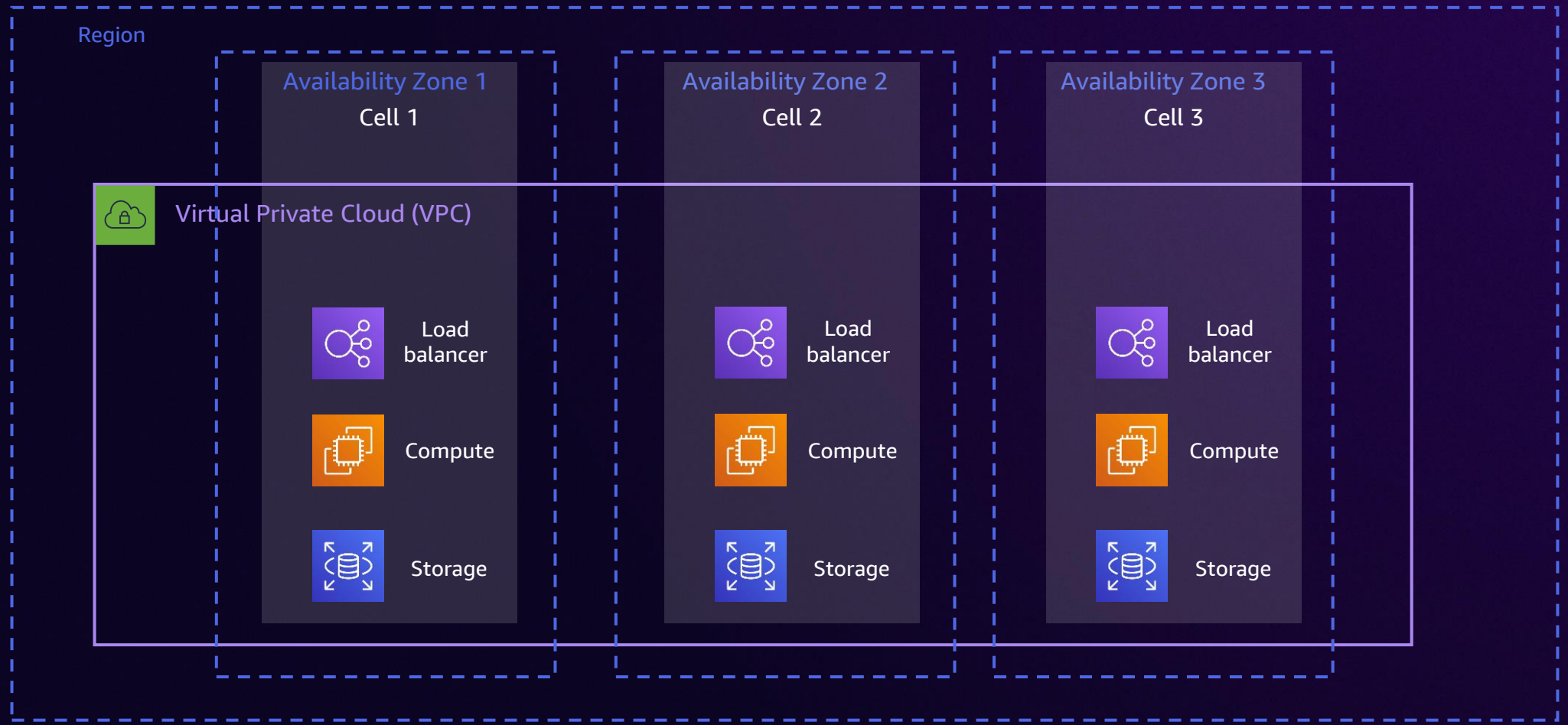
Cell Boundary

BUILDING CELLS WITH VPC ISOLATION



Cell Boundary

BUILDING CELLS WITH AZ AFFINITY

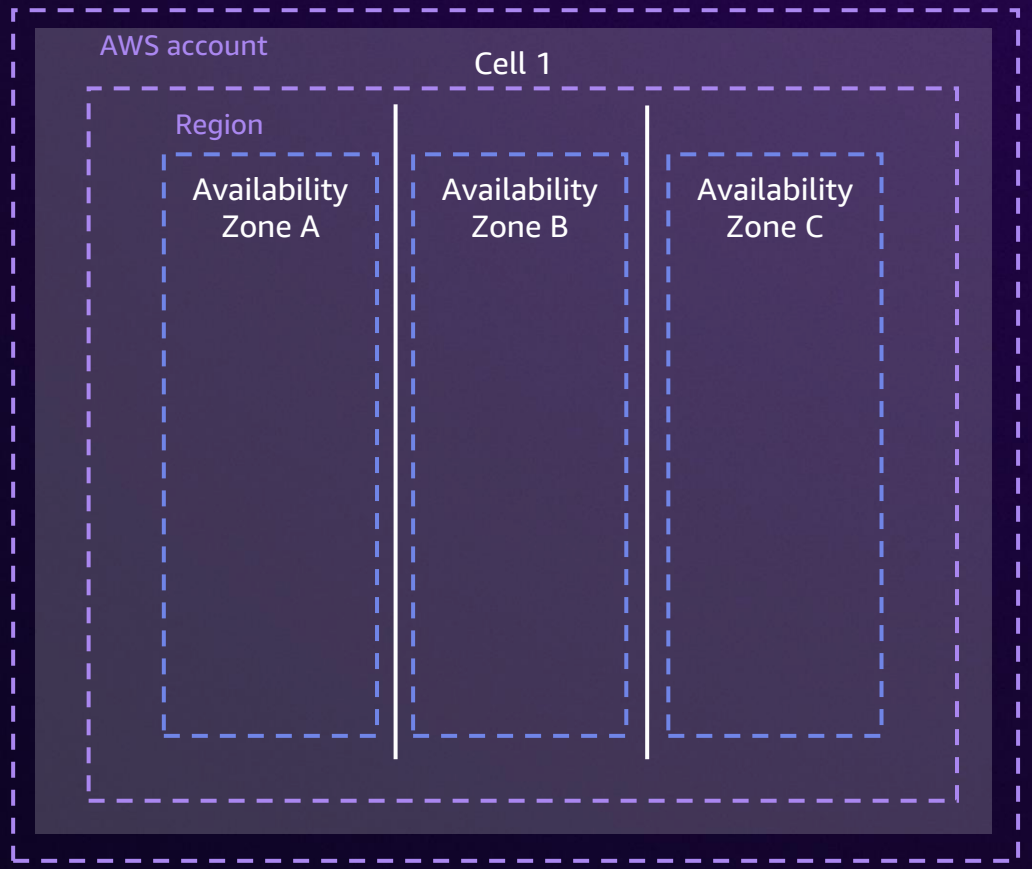


Cell boundary

BUILDING CELLS WITH AWS ACCOUNT ISOLATION

AWS Organizations

AWS Control Tower

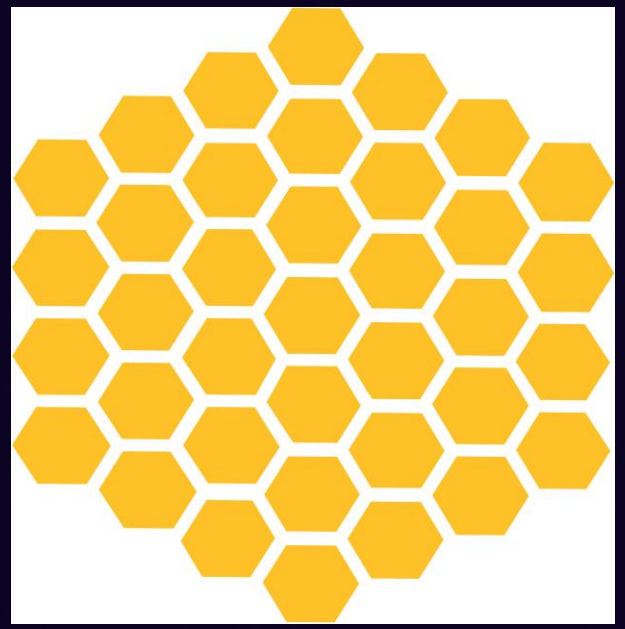


Cell size

FOR PRE-DETERMINED SCALE

Many small cells

- Smaller blast radius
- Easier to test
- Easier to operate (cell)



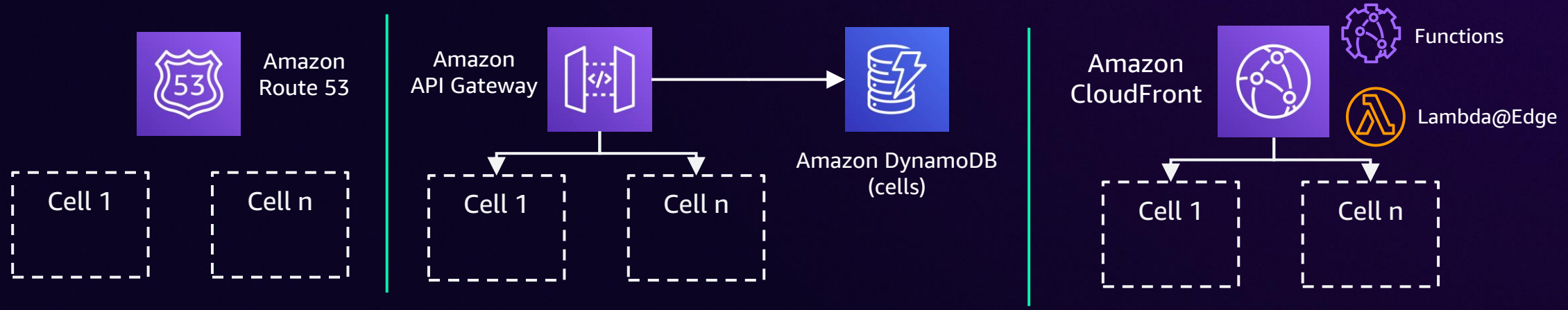
Few large cells

- Better capacity utilization
- Easier to accommodate “whales”
- Easier to operate (system)

Cell router

THINNEST POSSIBLE LAYER

- **Discovery** mechanism for clients
- **Proxy service** – transparent to clients, retry mechanism
- Fast and **reliable** (handles failures)



Cell-based architecture

CHALLENGES

- Increased operational complexity
- Difficult to maintain database isolation
- Challenges with data consistency during cell failovers
- Cell rebalancing

Key takeaways

- Start with defining the need for cell-based design
- Consider long-term growth (>5 years) to define cell boundaries and cell size
- Operational processes, health, and tools for cell-level visibility
- Cell scaling and rebalancing based on telemetry and operational reviews
- Availability requirements for cell router are higher than those for cells
- Independent databases for cells to avoid shared dependencies, when possible

Resources

White paper: Reducing the Scope of Impact with Cell-Based Architecture

AWS Well-Architected Framework: Reliability Pillar

AWS Resilience Hub

Blog: Using Containers and Cell-based design for higher resiliency and efficiency

Thank you!

Please complete the session survey in the mobile app

Anand Gaitonde

https://www.linkedin.com/in/anandprasannag

Anuj Gupta

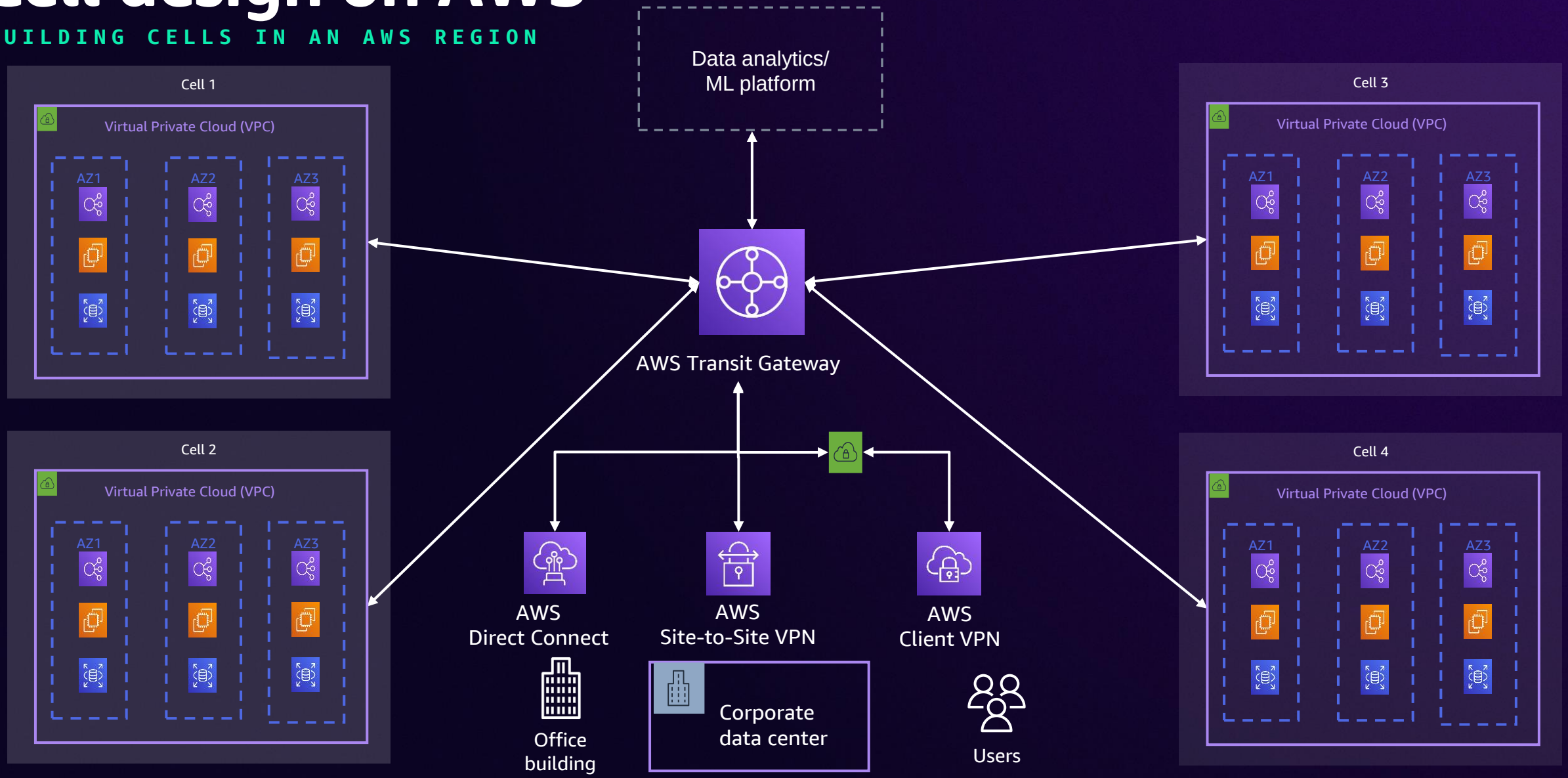
https://www.linkedin.com/in/anuj-gupta-aws/

Appendix

Cell-based design patterns

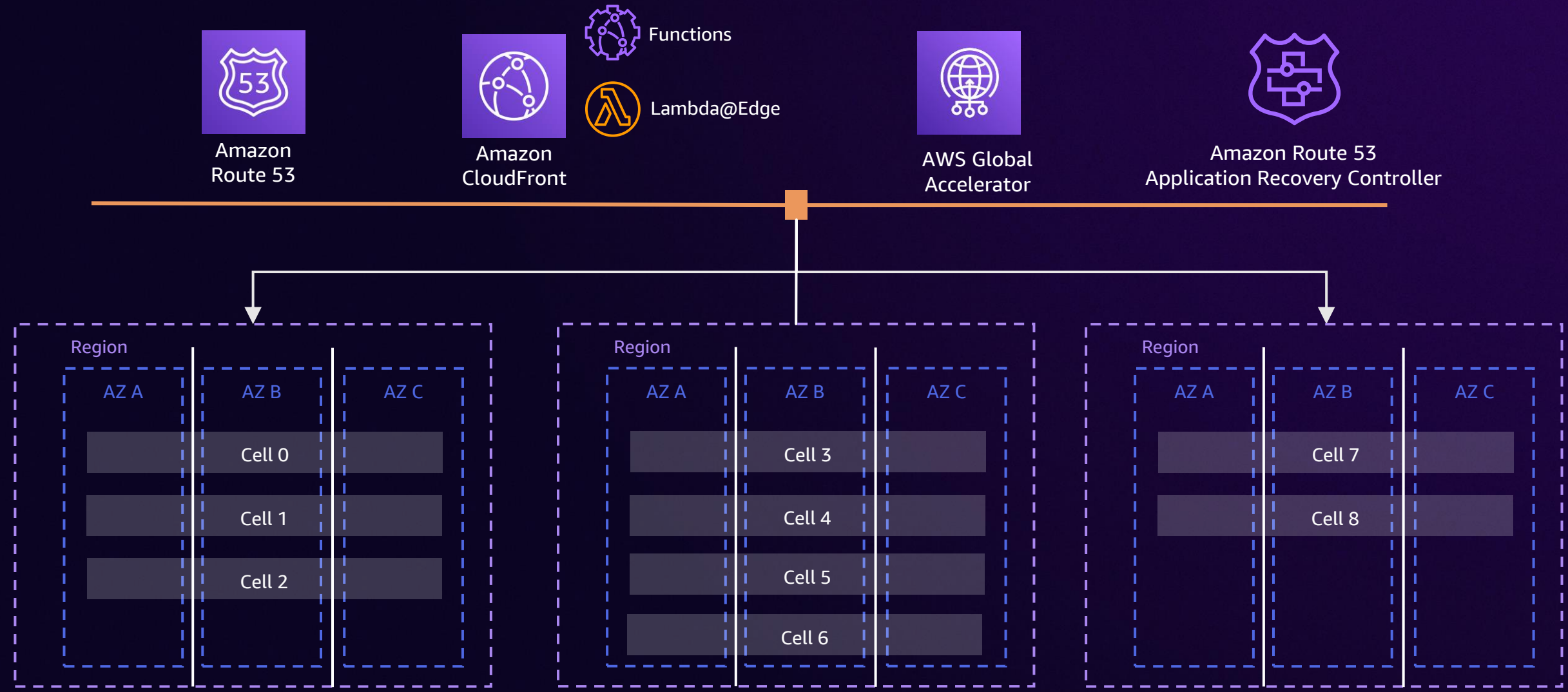
Cell design on AWS

BUILDING CELLS IN AN AWS REGION



Cell design on AWS

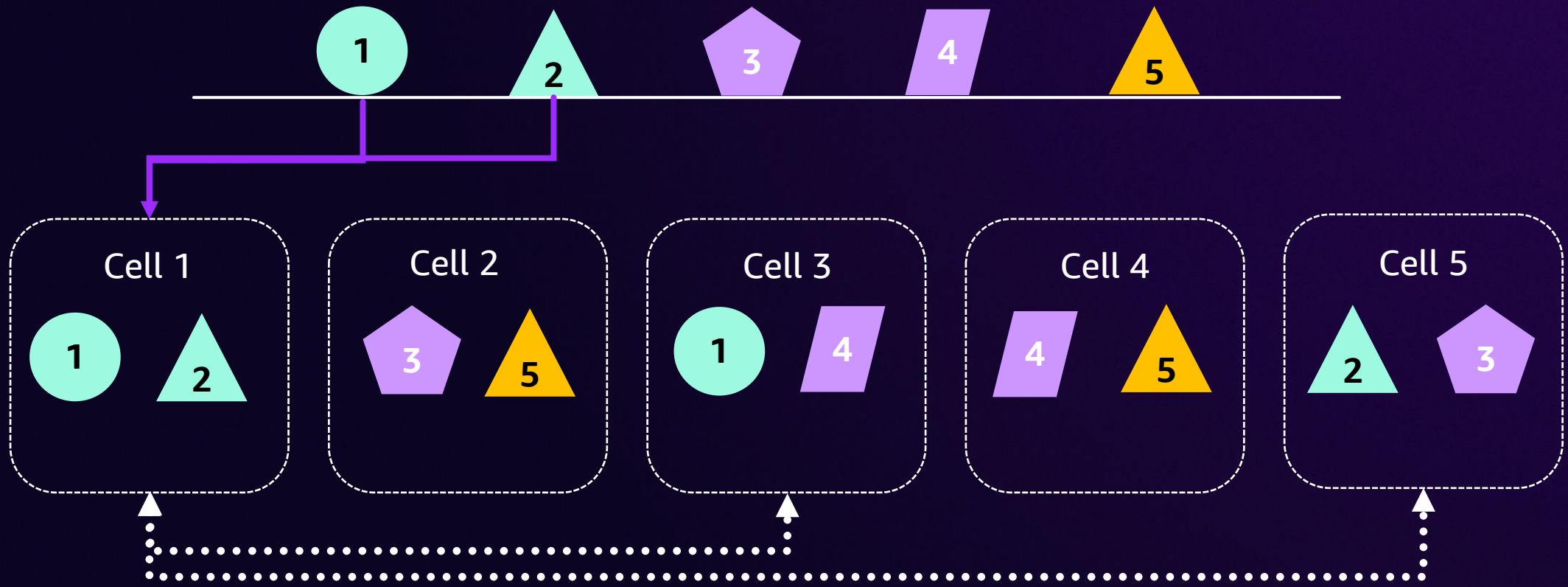
BUILDING MULTI-REGION CELLS



Architecture scenarios

Shuffle sharding

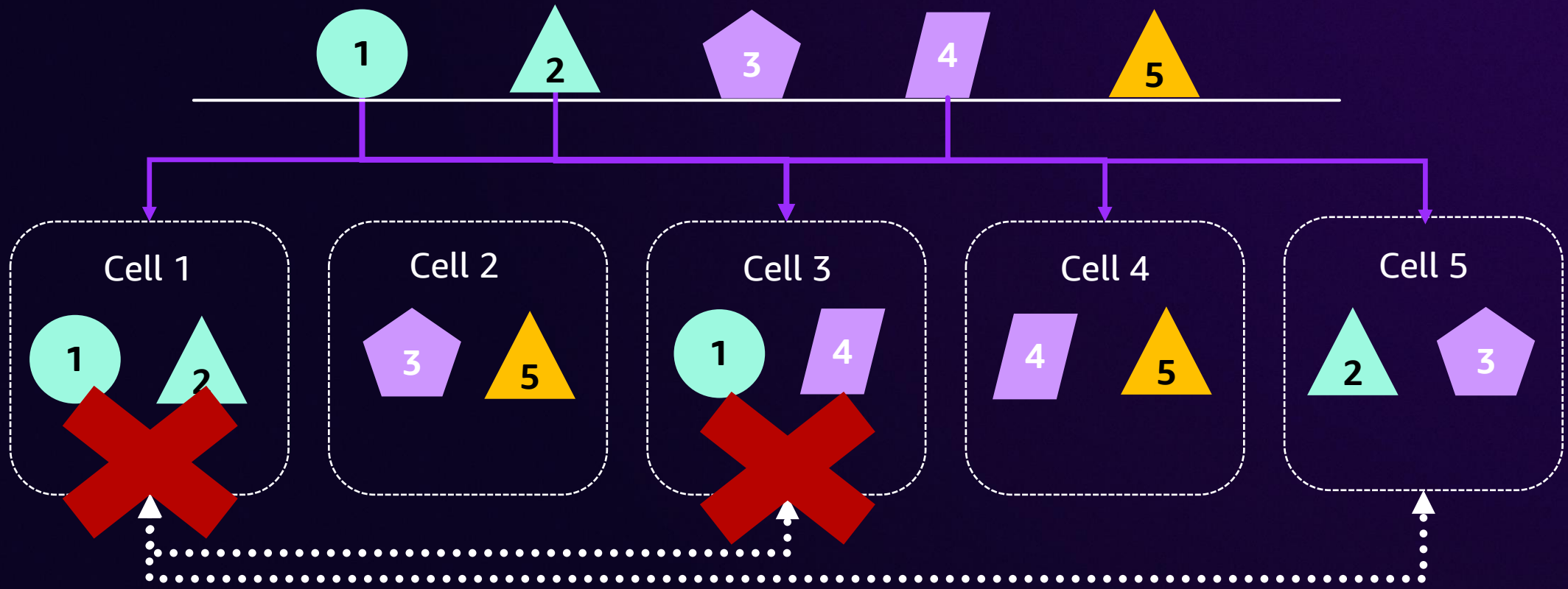
CUSTOMER-BASED CELLS



Customers = 5, Shards = 5, Overlap =1

Shuffle sharding

CUSTOMER-BASED CELLS



Customers = 5, Shards = 5, Overlap =1, Blast radius = 20%

Shuffle sharding

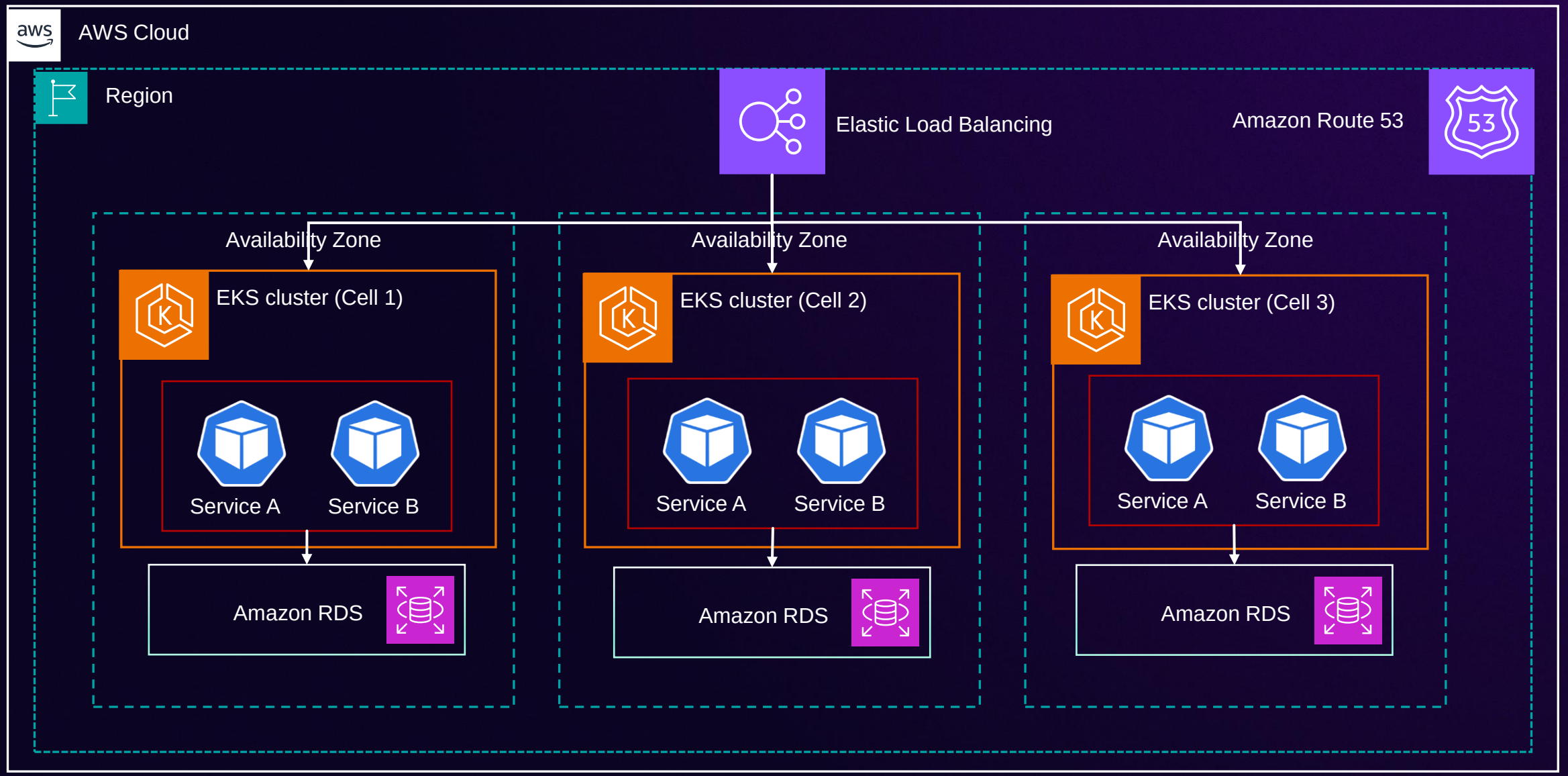
CUSTOMER IMPACT

Customers:100
Customer assignment per shard: 20

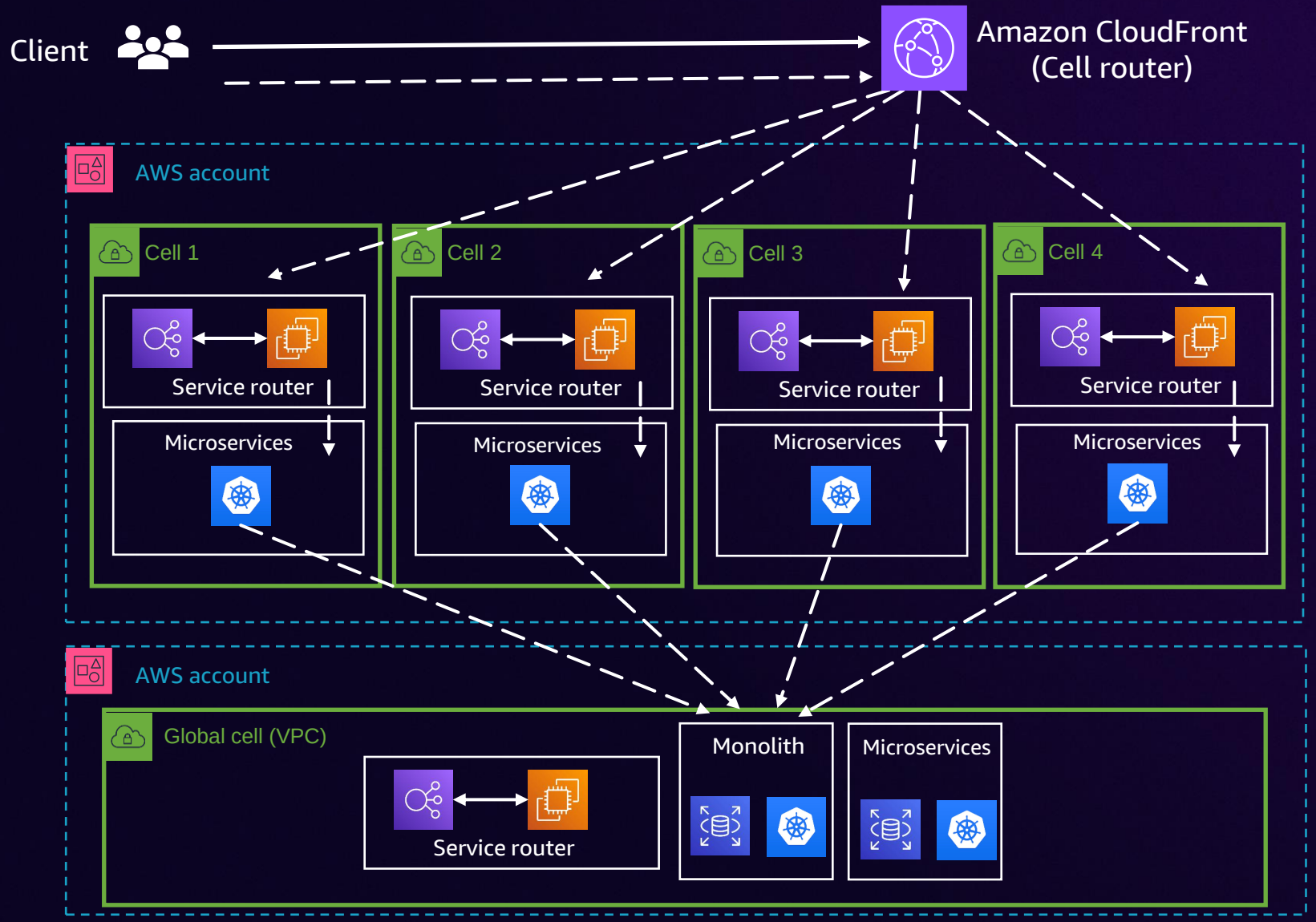
Overlap	Blast radius of shuffle shard
1	4.3252089920389238386%
2	12.5919390816617102757 4%
3	21.5861812828486421267 4%
4	24.3687749638408597263 6%
5	19.1950965869023306709 1%
6	10.9063048789217802436 7%
7	4.5578587553702956114 4%
8	1.4159524442234929786 4%

Calculate blast radius script

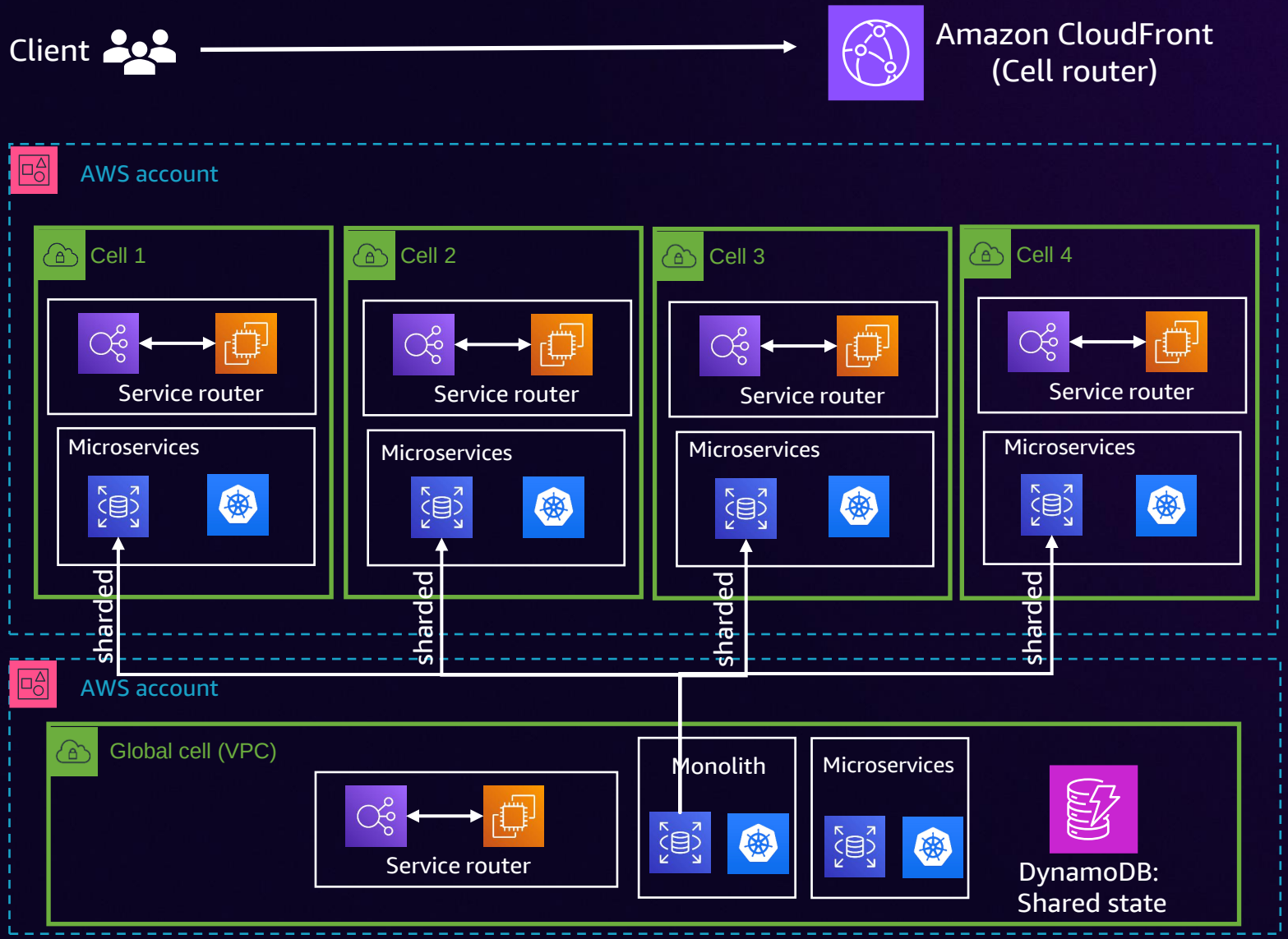
Cells with EKS on AWS



Cell-based architecture in practice

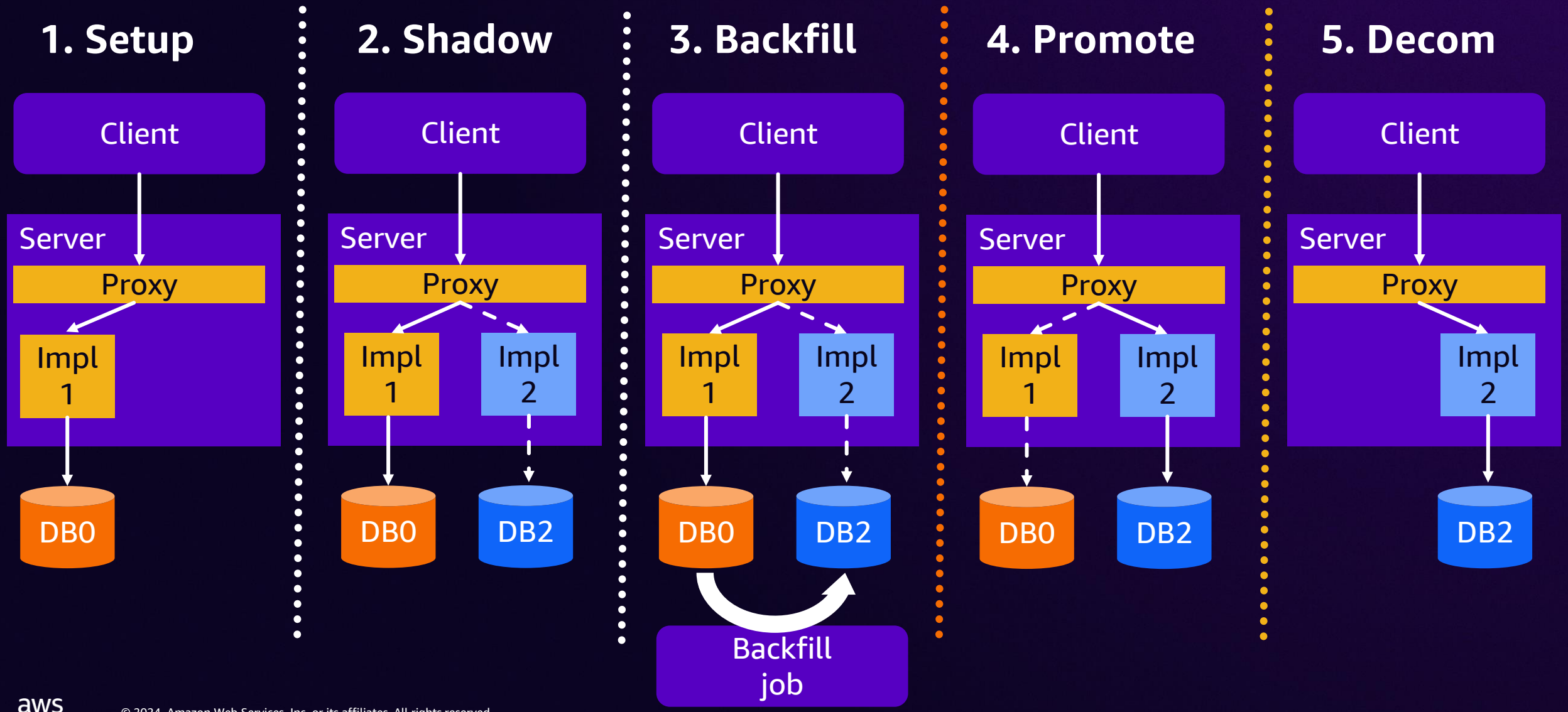


Sharding your database



- Shard db across your cell-partitioning key
- Setup global cluster-cell cluster replication
  - Aurora Postgres logical replication/MySQL binlog
  - AWS Database Migration Service
- Amazon Dynamo DB as global database for shared state

Database migration



Cell failover using ARC