AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

GPS402

Design patterns to consider when moving to Amazon Aurora

Jatin Singh (he/him)
DB Specialist Partner Solutions Architect
AWS

Kiran Singh (she/her)
DB Specialist Partner Solutions Architect
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Overview of Amazon Aurora

Design patterns

Call to action

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Overview

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora...

Enterprise database at open-source price

Delivered as a **managed** service



Speed and **availability** of high-end commercial databases

Simplicity and **cost-effectiveness** of open-source databases

Drop-in **compatibility** with MySQL and PostgreSQL

Simple **pay-as-you-go** pricing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 1:
High availability (HA)

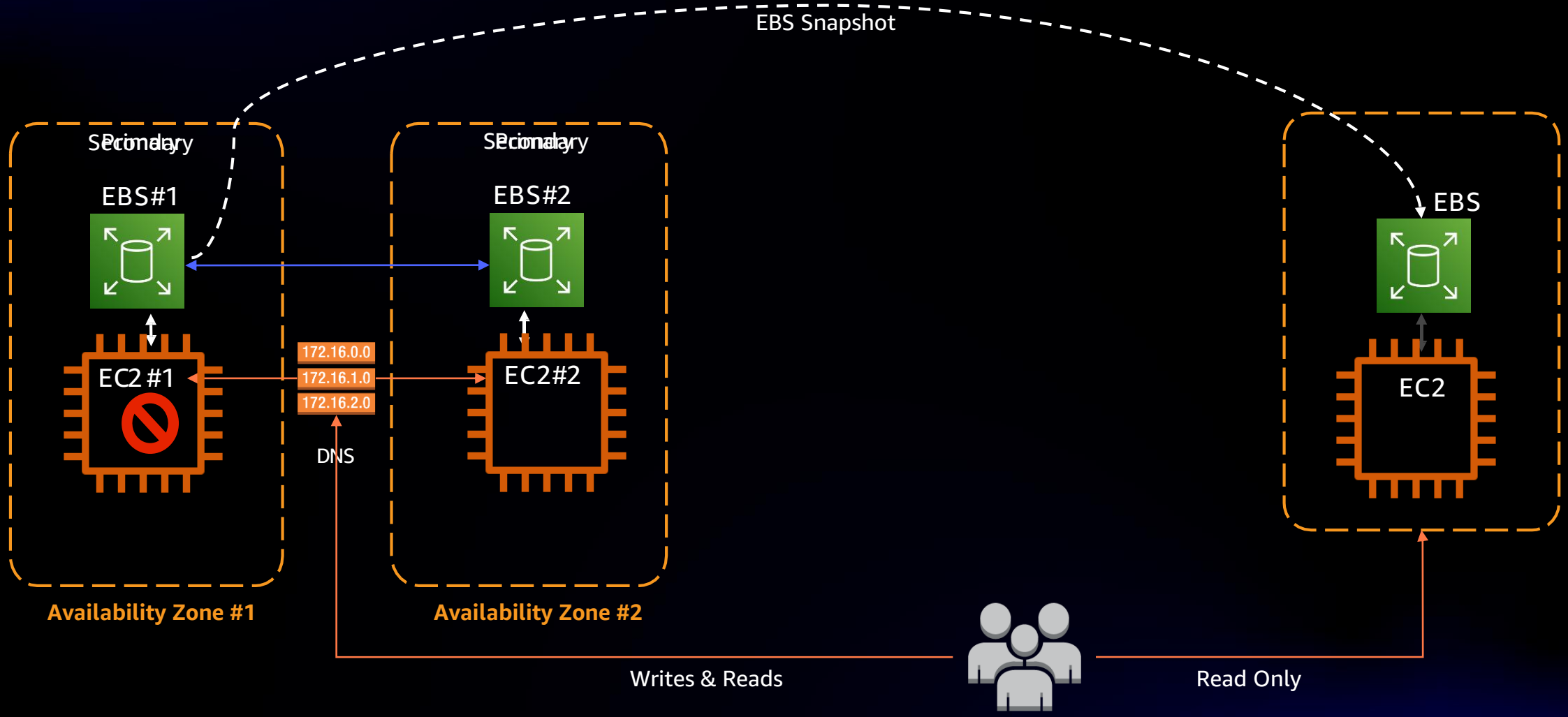
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

- Application behavior with respect to database outage
- Read replica vs. failover target
- Planned vs. unplanned downtime

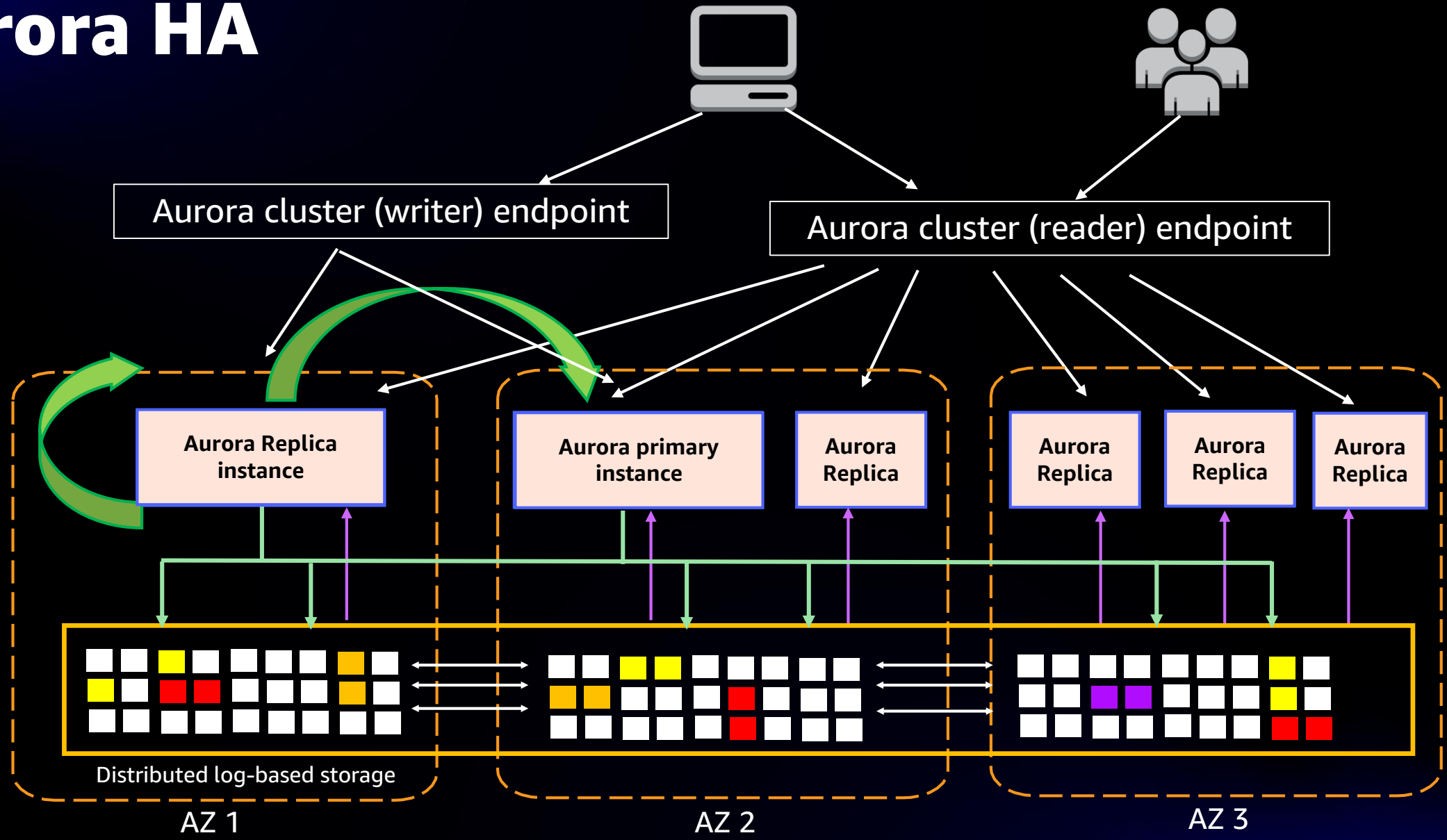
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS HA



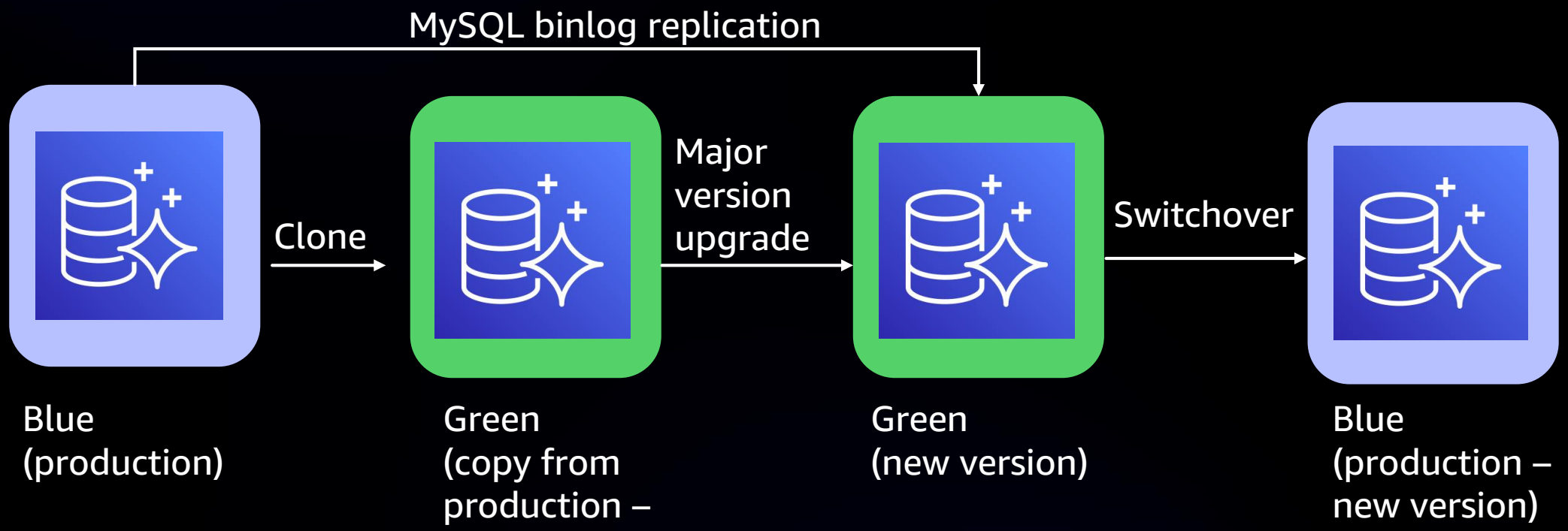
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora HA



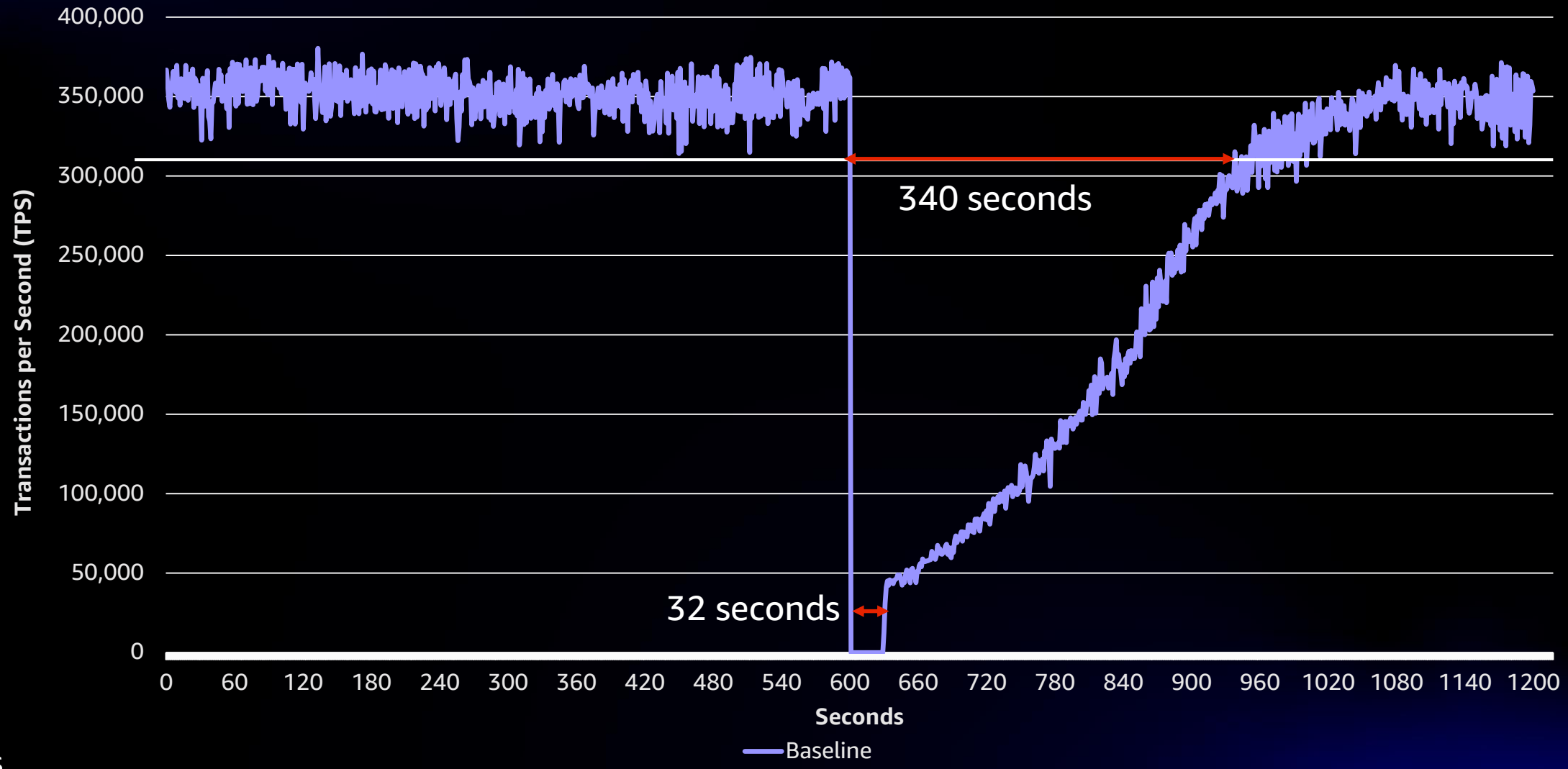
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Planned downtime: Blue/green deployment



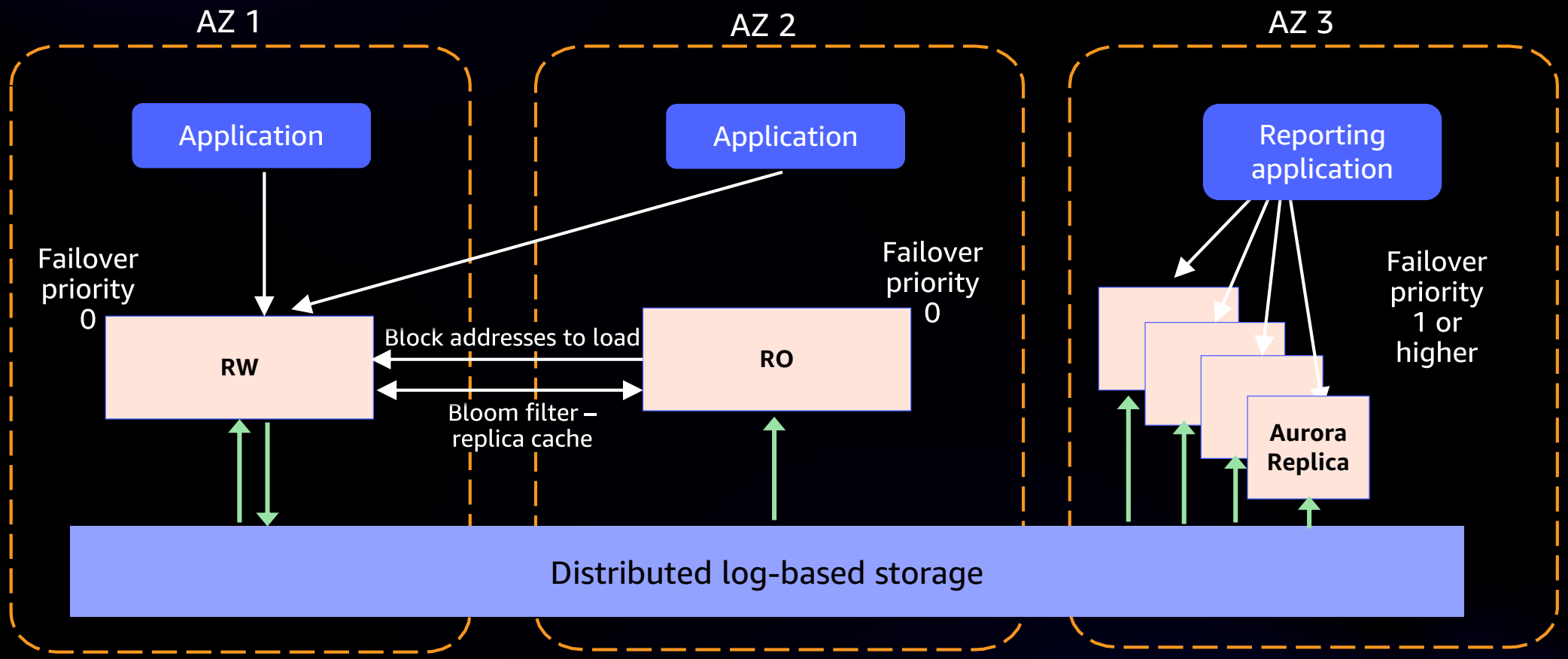
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Failover – How applications perceive it



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

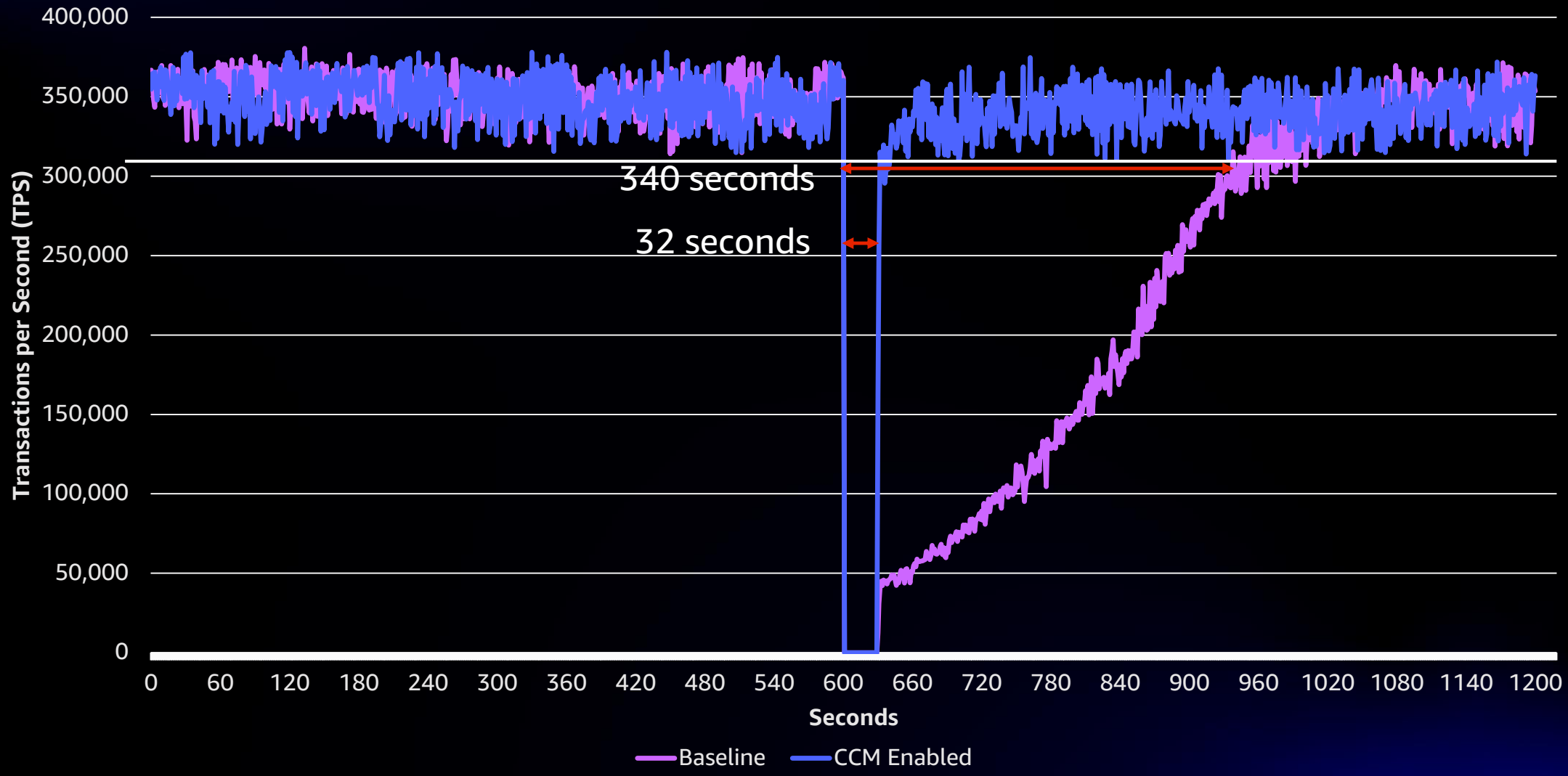
Unplanned downtime: Cluster cache management



apg_ccm_enabled=on

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Failover behavior with cluster cache management



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unplanned outages

TCP keepalives

Java DNS caching timeout

JDBC timeout variables in JDBC Connection String

Using read & write cluster endpoints
(not the instance endpoints)

Use AWS JDBC Driver for PostgreSQL (preview)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 2: Scalability

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

Problem 1

- How does aurora scale the storage as the data grows?

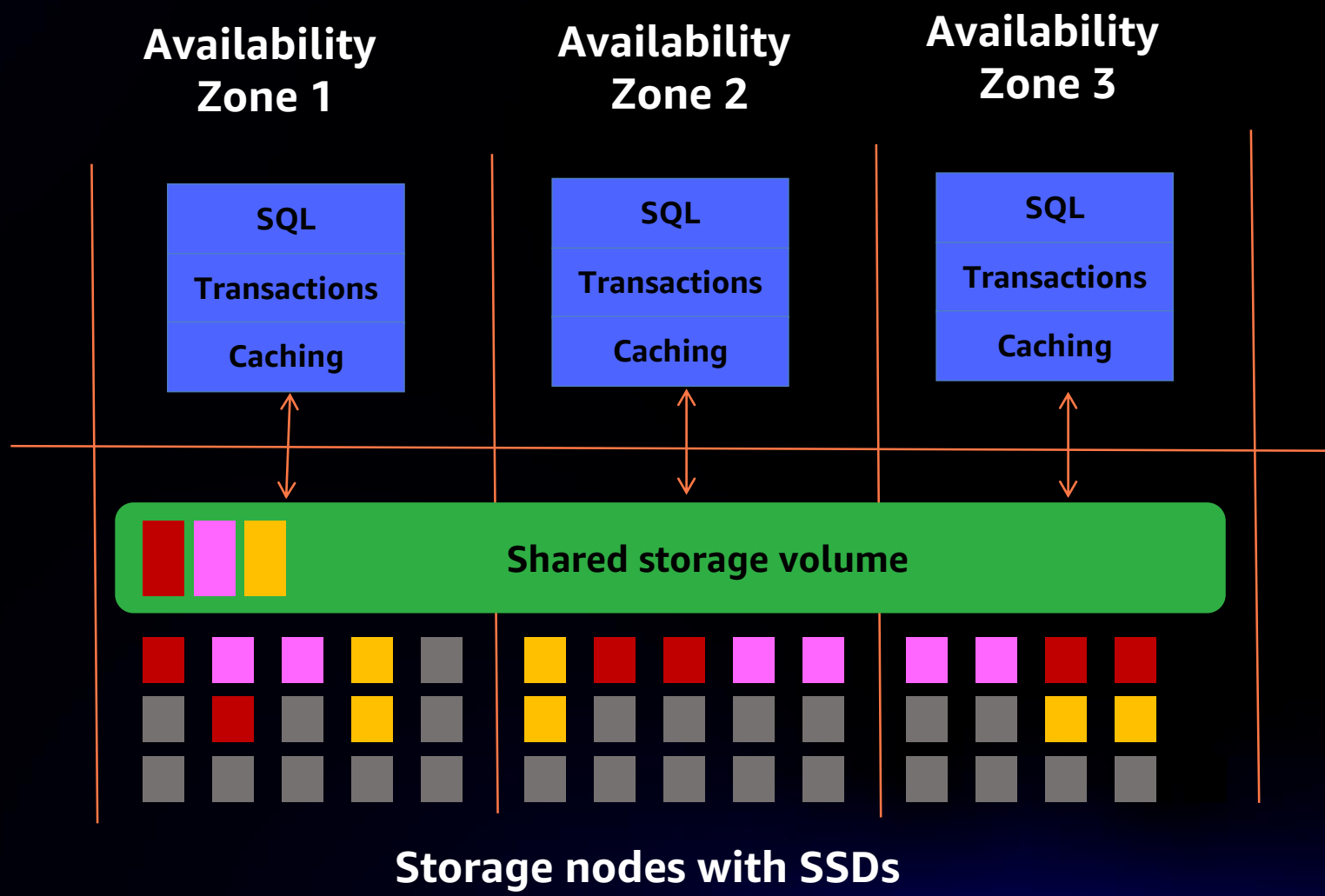
Problem 2

- How can I elastically scale out beyond the capacity of a single instance for read heavy workloads?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

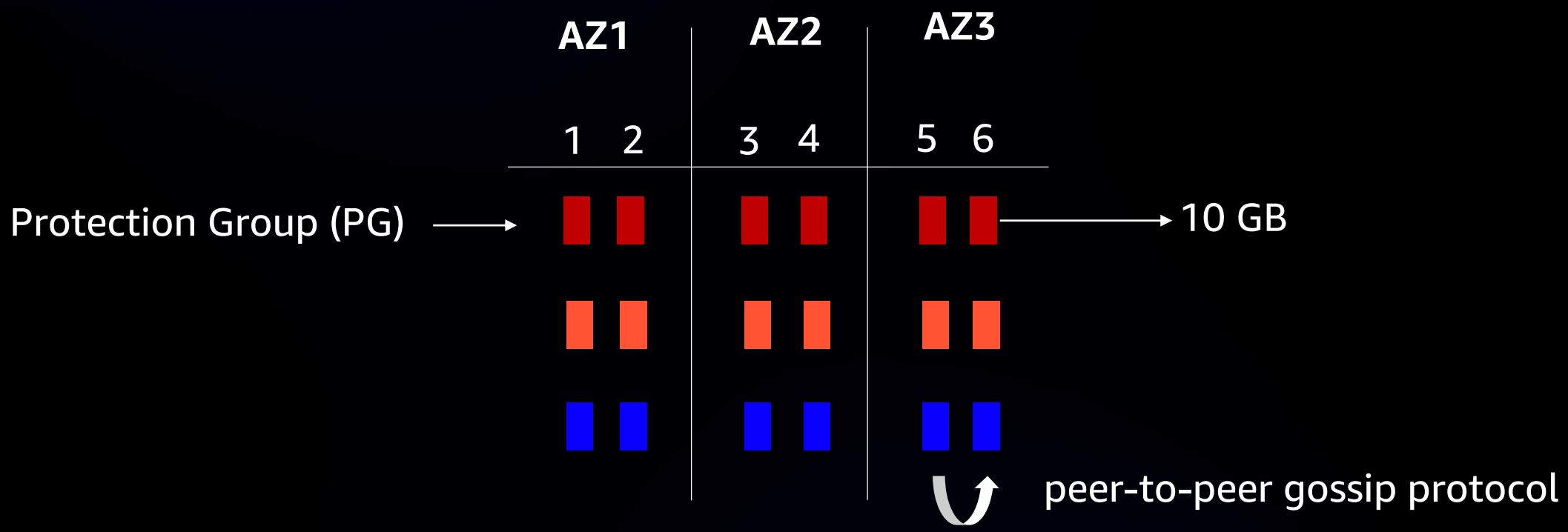
Aurora storage architecture

- Storage volume is striped across hundreds of storage nodes distributed over 3 different Availability Zones
- Six copies of data, two copies in each Availability Zone to protect against failures
- 10 GB of segment size
- Primary and replicas (up to 15) all point to the same storage



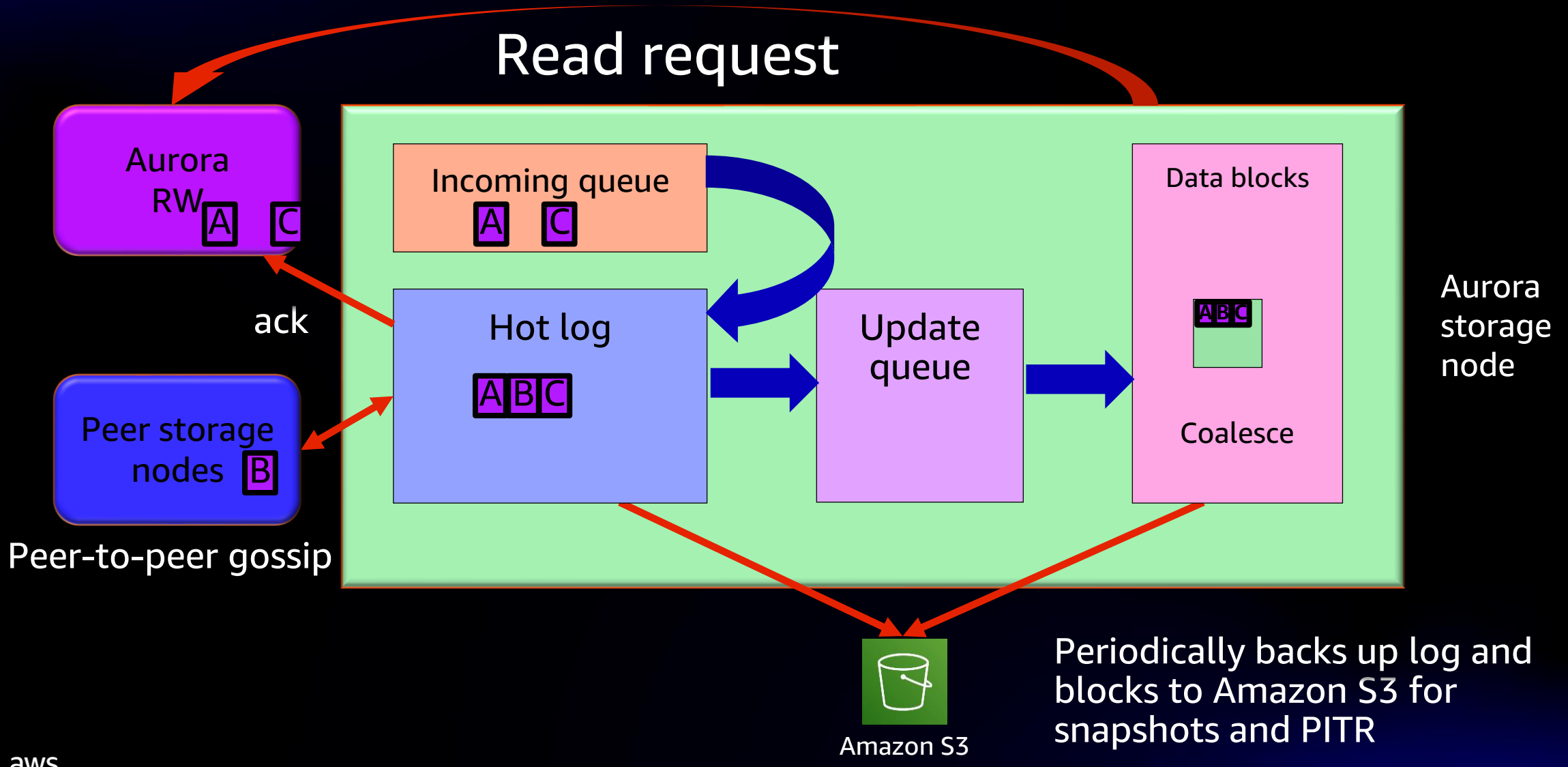
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora storage architecture



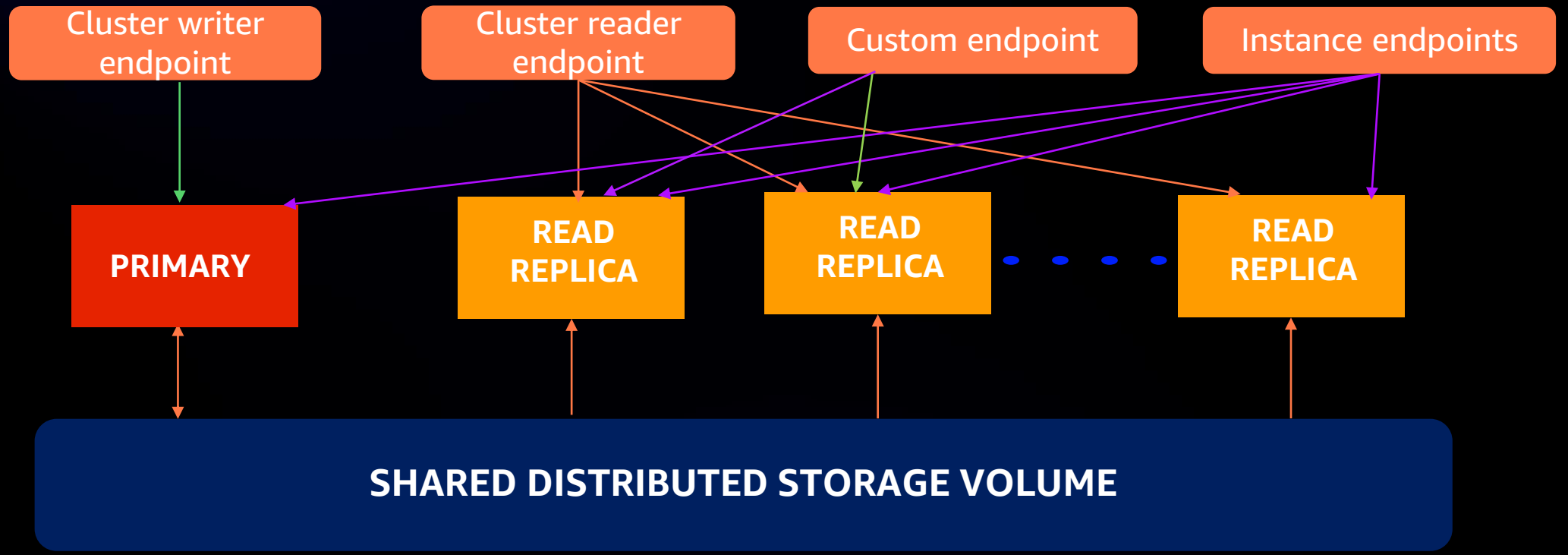
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

I/O flow in Amazon Aurora storage mode



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Read replicas, custom reader endpoints



Up to 15 promotable read replicas across multiple Availability Zones

Custom reader endpoint with configurable failover order

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Read Replica Auto Scaling

- Enables Aurora DB cluster to handle sudden increases in connectivity or workload by scaling out the number of Read Replicas

- When the connectivity or workload decreases, Aurora Auto Scaling automatically removes unnecessary Read Replicas

- Enable Aurora Auto Scaling by defining a scaling policy
  - Specify minimum and maximum number of Aurora Replicas
  - Use predefined metrics or create custom metrics to define the scaling policy

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Global Database

Faster disaster recovery and enhanced data locality

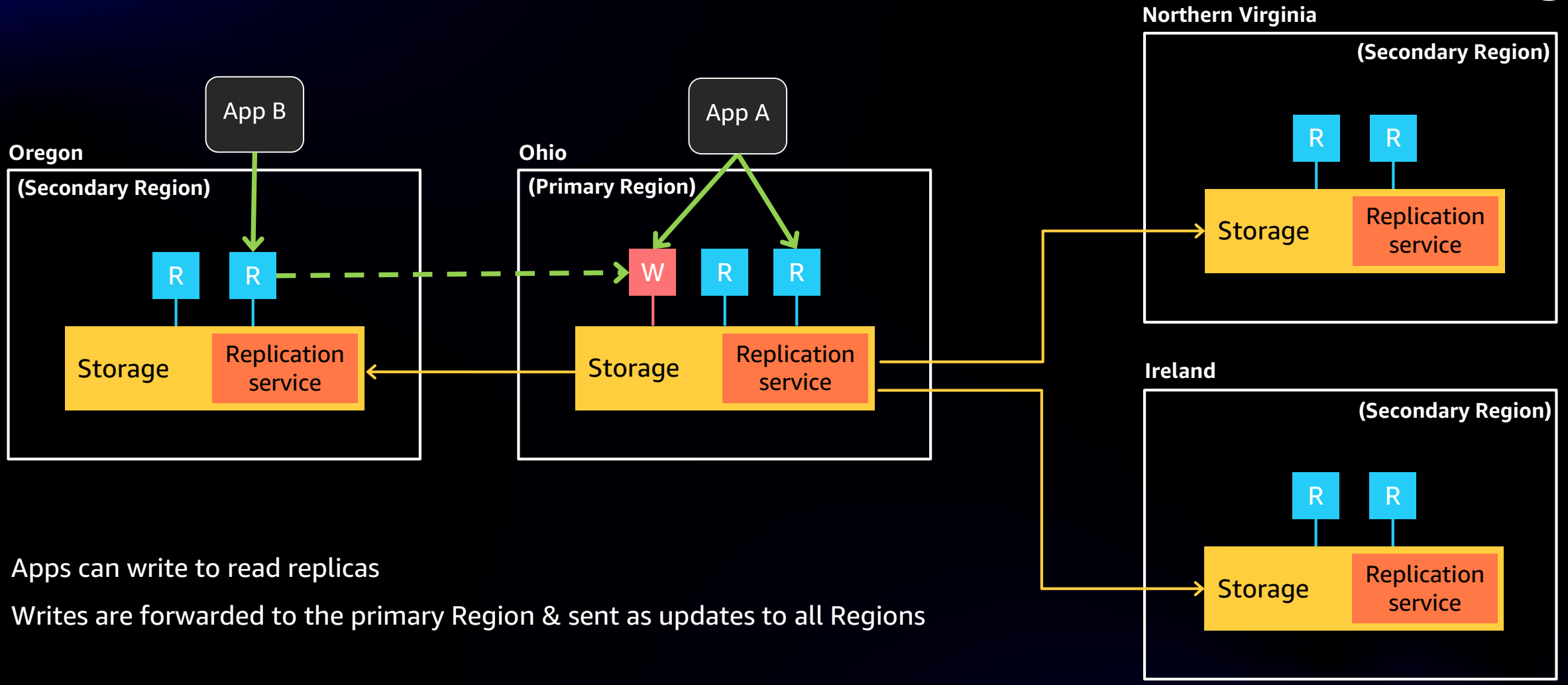


Architecture:

- Physical, log-based asynchronous replication
- Up to five secondary Regions

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Global Database: Write forwarding



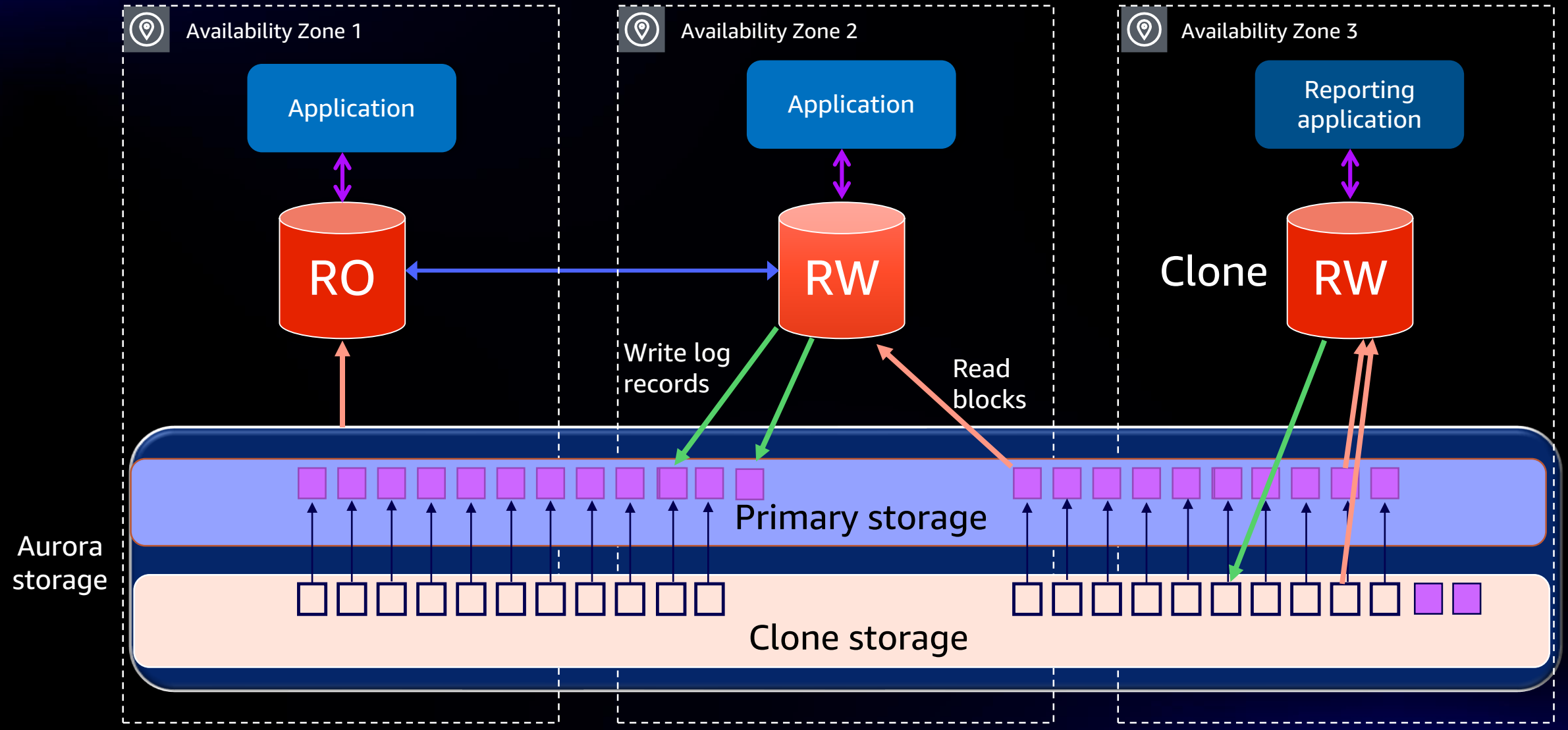
Apps can write to read replicas

Writes are forwarded to the primary Region & sent as updates to all Regions

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Write scalability: Clones



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 3: Security

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

Problem 1

- How to secure Aurora on AWS?

Problem 2

- What are the security features available to manage user access to database?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kerberos authentication

- Network authentication protocol
- Uses tickets and symmetric-key cryptography
- Eliminates the need to transmit passwords over the network

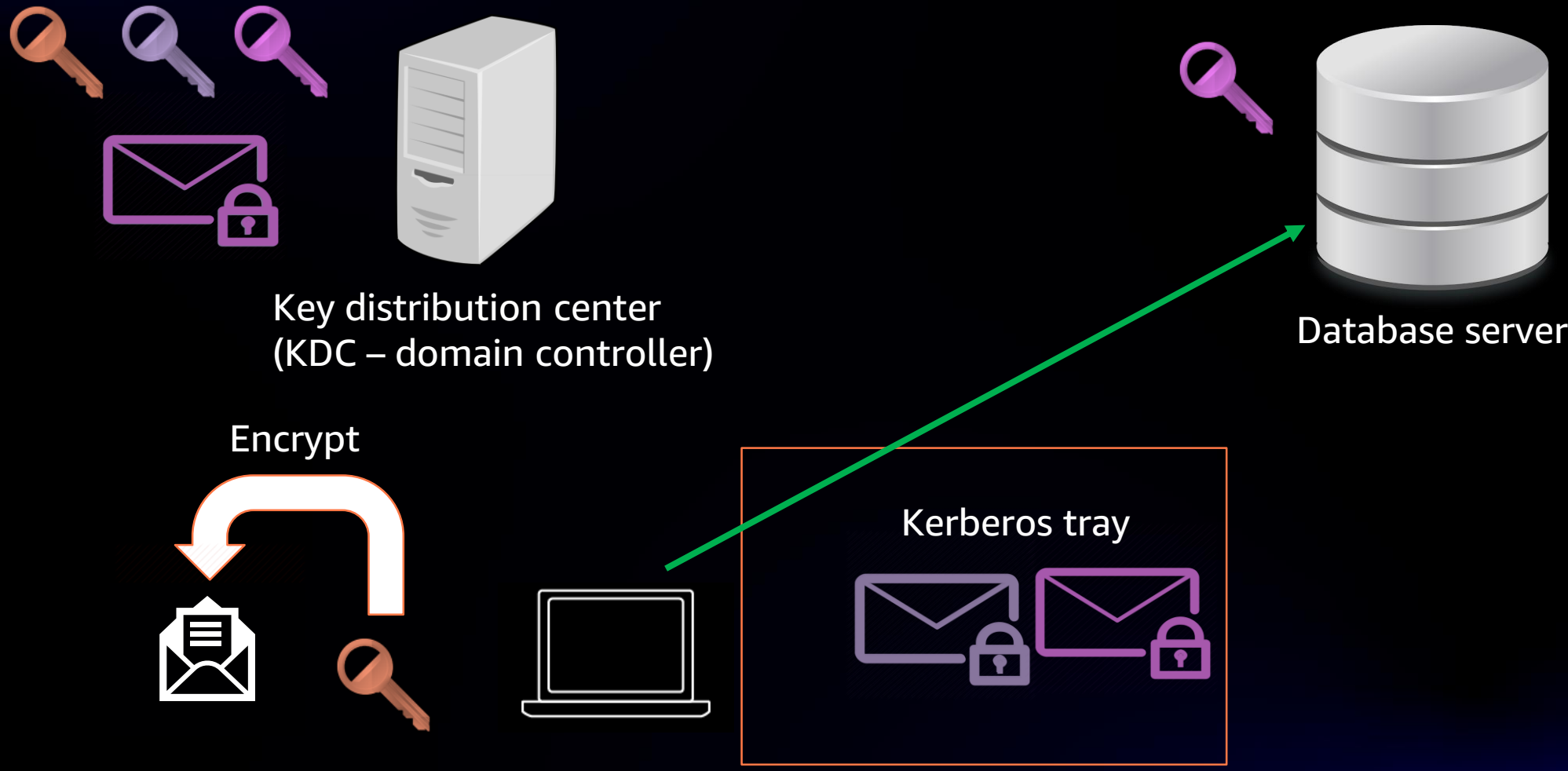
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kerberos authentication: Benefits

- Single sign-on experience
- One place to store credentials, which can be used for multiple RDS database instances (across different RDS engines, VPCs, and AWS accounts), or other services
- Better enforcement of password requirements (password strength, reuse policy, rotation policy)

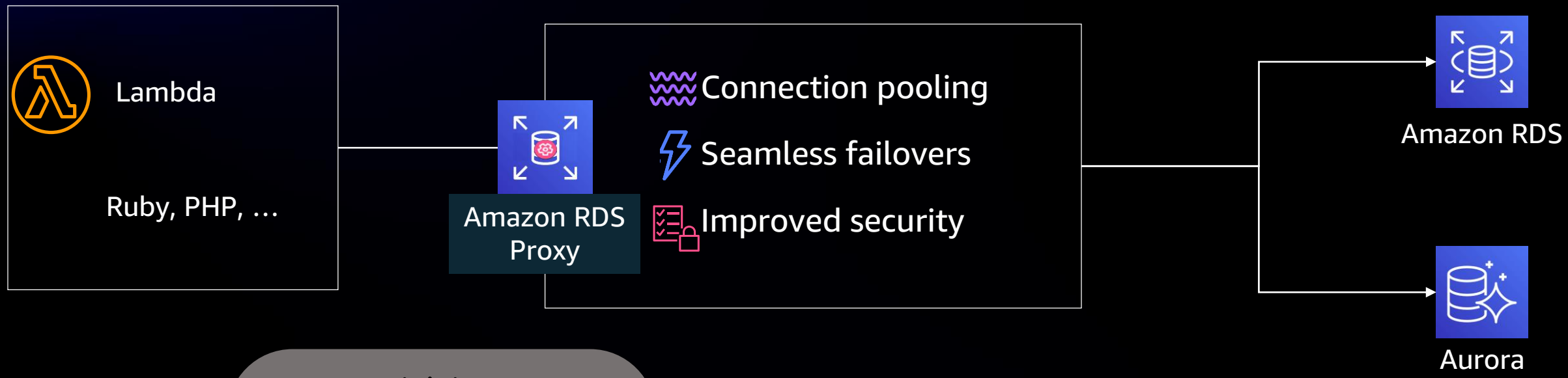
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kerberos deep dive



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS Proxy: Authorization





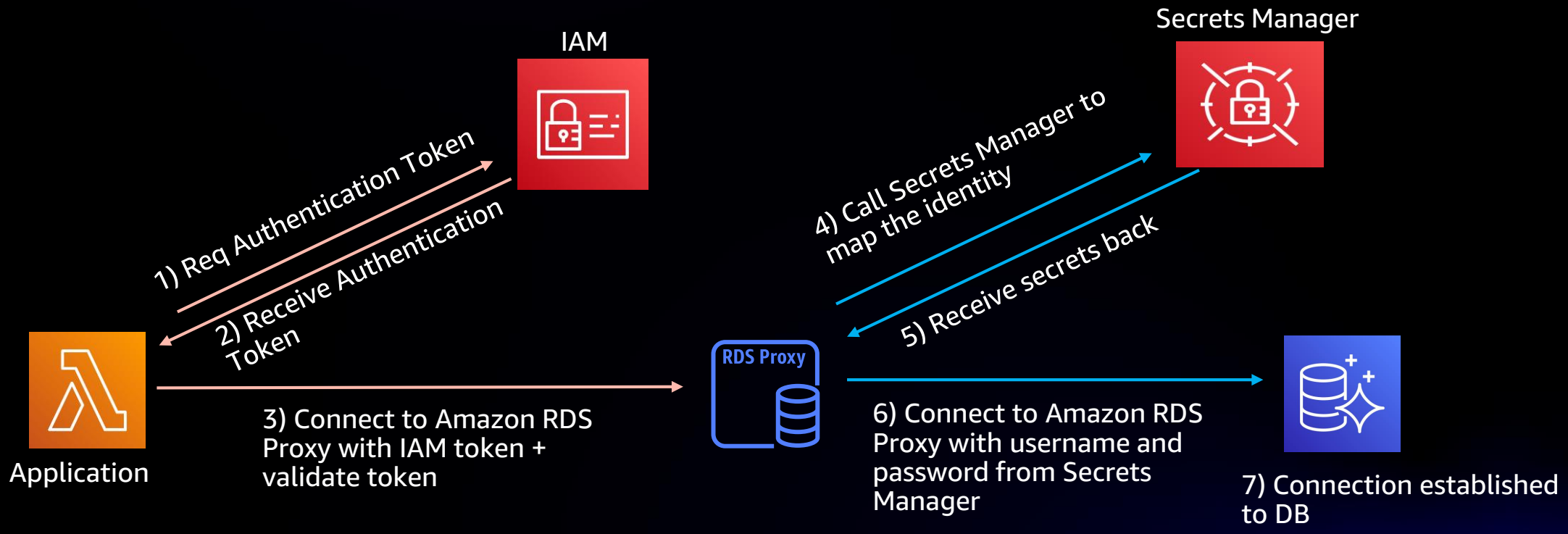


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Improved application security

Enforce IAM authentication with your relational databases

Centrally manage database credentials using AWS Secrets Manager



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 4: Migration

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

- Heterogeneous migration of a critical application
- Low downtime, large volume of changes
- Large number of small tables
- Small number of large tables
- AWS Database Migration Service (AWS DMS) is primary tool

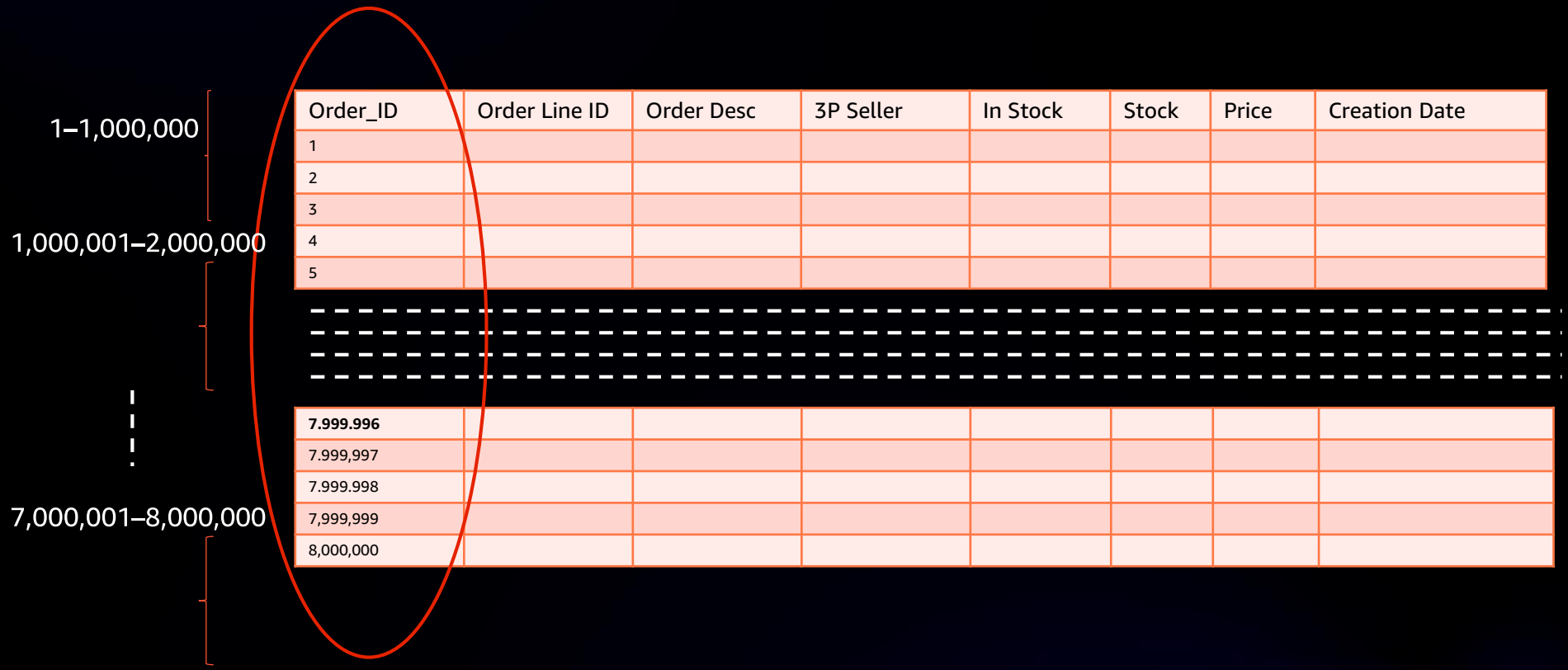
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

General guiding principles

- AWS DMS replication instance sizing
- Source
- Target
- Network
- Logical design of task
  - Full load
  - Change data capture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DMS using row filtering or existing partitions



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DMS

PARALLEL FULL LOAD – PARTITIONED TABLE

```
CREATE TABLE HR.SALES
       (
       SALES_NO NUMBER,
       DEPT_NAME VARCHAR2 (20),
       SALE_AMOUNT NUMBER (10, 2),
       SALE_DATE DATE,
       REGION VARCHAR2 (10)
       )
       PARTITION BY LIST (REGION)
       (
       PARTITION P1 VALUES ('NORTH'),
       PARTITION P2 VALUES ('SOUTH'),
       PARTITION P3 VALUES ('EAST'),
       PARTITION P4 VALUES ('WEST')
       );
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Parallel full load

```
{
  "rule-type": "table-settings",
  "rule-id": "2",
  "rule-name": "2",
  "object-locator":
     { "schema-name": "HR",
       "table-name": "SALES" },
"parallel-load":
     { "type": "partitions-auto" }
}
```

```
[TASK_MANAGER ] I: Start loading segment #1 of 4 of table 'HR'.'SALES' (Id = 1) by subtask 1.
[SOURCE_UNLOAD ] I: Unload finished for segment #1 of segmented table 'HR'.'SALES' (Id = 1). 1 rows sent.
[TARGET_LOAD ] I: Load finished for segment #1 of segmented table 'HR'.'SALES' (Id = 1). 1 rows received. 0 rows skipped.
[TASK_MANAGER ] I: Load finished for segment #1 of table 'HR'.'SALES' (Id = 1) by subtask 1. 1 records transferred.
[TASK_MANAGER ] I: Start loading segment #4 of 4 of table 'HR'.'SALES' (Id = 1) by subtask 1.
[SOURCE_UNLOAD ] I: Unload finished for segment #4 of segmented table 'HR'.'SALES' (Id = 1). 1 rows sent.
[TARGET_LOAD ] I: Load finished for segment #4 of segmented table 'HR'.'SALES' (Id = 1). 1 rows received. 0 rows skipped.
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DMS

PARALLEL FULL LOAD – NONPARTITIONED TABLE

```
{ "rule-type": "table-settings",
  "rule-id": "2",
  "rule-name": "2",
  "object-locator": { "schema-name": "HR", "table-name": "SALES" },
  "parallel-load": { "type": "ranges",
                     "columns": [ "SALES_NO", "REGION" ],
                     "boundaries": [
                        [ "1000", "NORTH" ],
                        [ "3000", "EAST" ]
                                  ]
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern 5: Long-term retention

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement



Time and speed to restore

Outdated software version

Corrupted media, no easy way to test

Large volume of data, no selective way

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Snapshot export to Amazon S3



Time & speed to restore

No software version issues

Durability – no corrupted media, easy to test

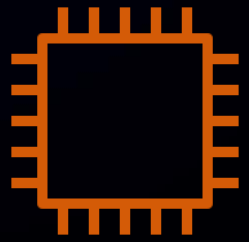
Selective data, highly compressed parquet format

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Restoring data



CSV import/copy



EC2/PostgreSQL

Parquet



Amazon Redshift

Supportability

- Amazon RDS MariaDB
- Amazon RDS MySQL
- Amazon RDS PostgreSQL
- Amazon Aurora with PostgreSQL compatibility

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaways

Objectively define your “nonfunctional” requirements & use solution architecture to address them upfront

Objectively define “modernization” & understand how cloud-based architecture can solve some of the archaic design issues

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Jatin Singh
sinjatin@amazon.com

Kiran Singh
kiransg@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.