Deep Dive on Amazon RDS PostgreSQL

Jignesh Shah, Amazon RDS

PostgreSQL

Robust feature sets and extensions

Multi-Version Concurrency Control (MVCC), point in time recovery, granular access controls, tablespaces, asynchronous replication, nested transactions, online/hot backups, a refined query planner/optimizer, and write ahead logging

Supports international character sets, multi-byte character encodings, Unicode, and it is locale-aware for sorting, case-sensitivity, and formatting

Reliable

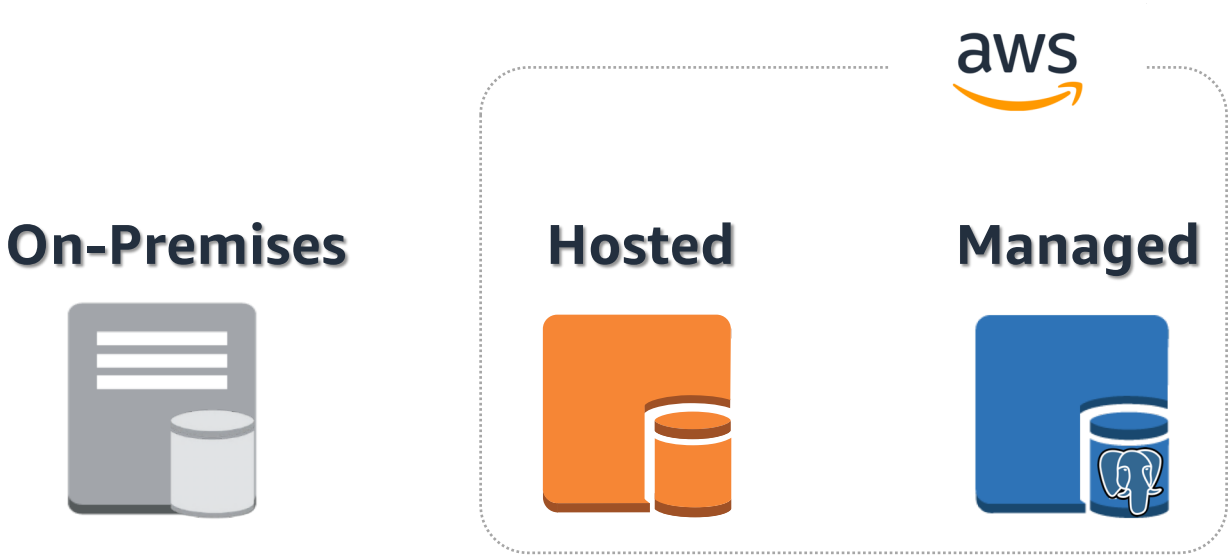
High fault tolerance, ACID compliance, and full support for foreign keys, joins, views, triggers, and stored procedures

Standards-compliant

Includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP. Supports storage of binary large objects, including pictures, sounds, or video

aws

PostgreSQL Deployment Options



aws

Running PostgreSQL on EC2

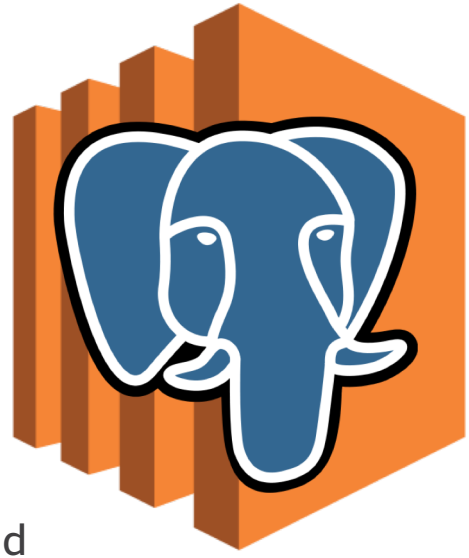
Full control over parameters of server, OS, and database

Remote access to host via ssh

Customer can install 3rd party applications and extensions

Self Managed PostgreSQL

- Customer has full responsibility for upgrades and backup
- Customer has major responsibility for security
- High Availability and replication are expensive, complex, and require a lot of engineering



Amazon Relational Database Service

Multi-engine support: Amazon Aurora, MySQ
MariaDB, PostgreSQL, Oracle, SQL Server

Automated provisioning, patching, scaling,
replicas, backup/restore, failover

High availability with RDS Multi-AZ and Ama
Aurora

Amazon Aurora
PostgreSQL
MySQL
MariaDB
ORACLE
Microsoft SQL Server

aws

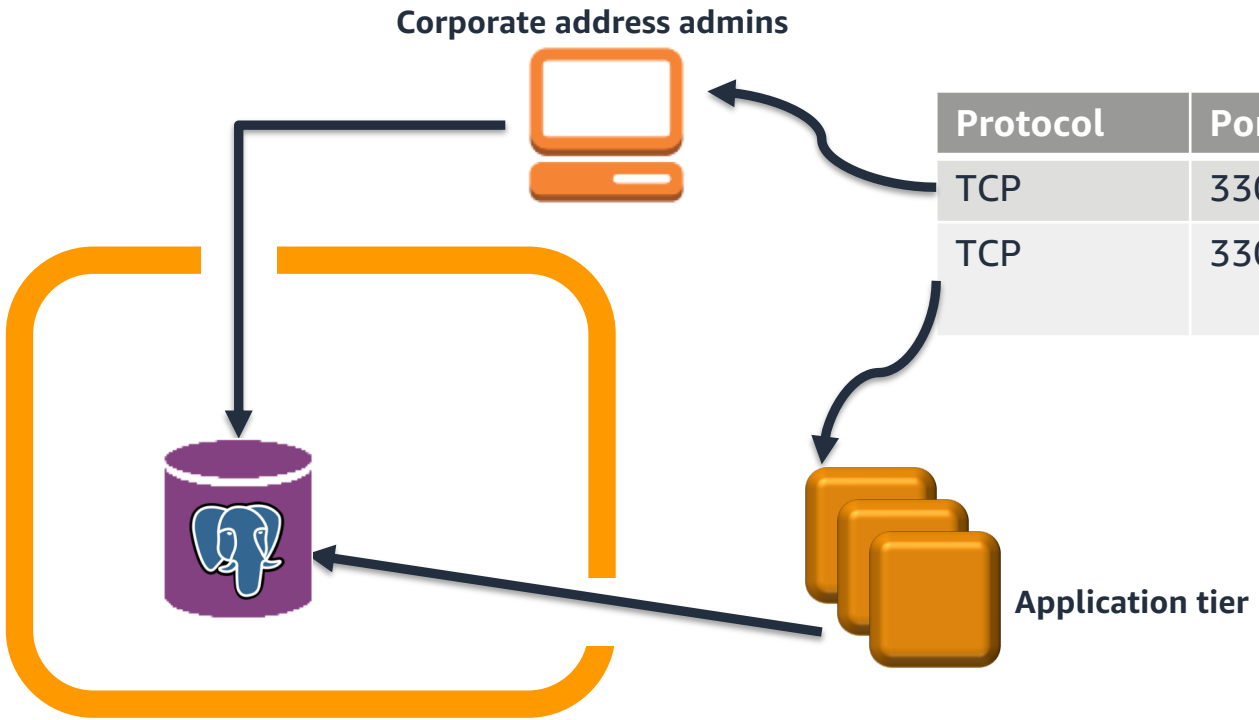
Security and Compliance

Cloud security at AWS is the highest priority

Compliance is important to meet industry and local regulations

Security Groups

Database IP firewall protection

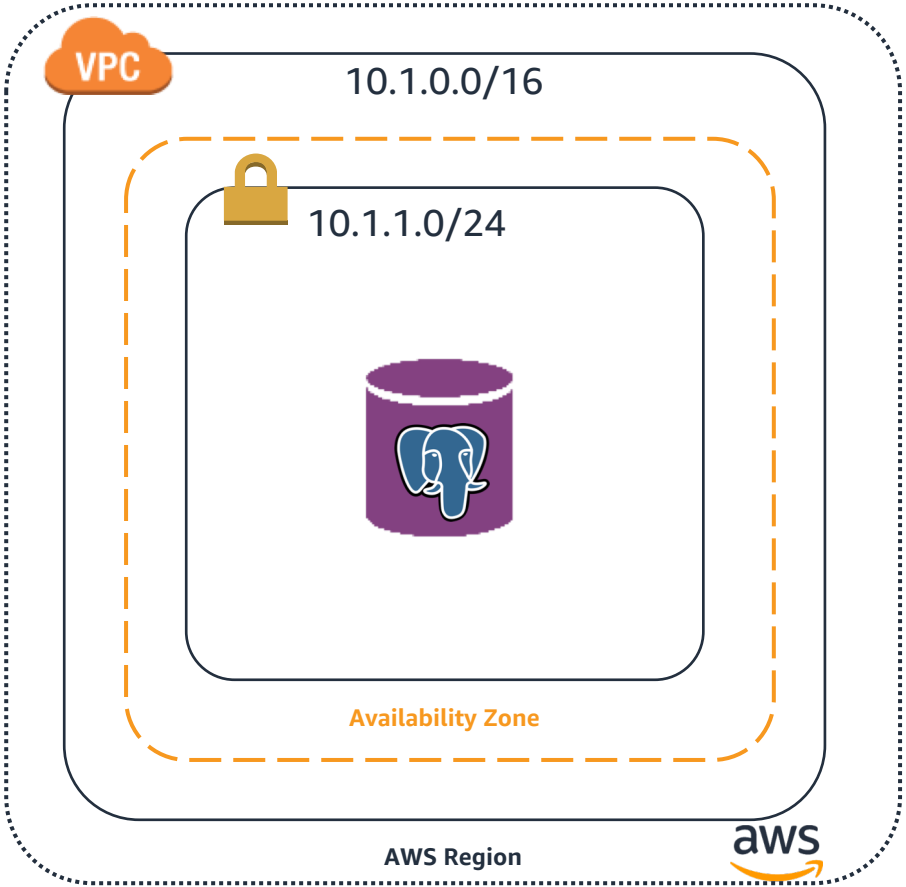


Protocol	Port Range	Source
TCP	3306	172.31.0.0/16
TCP	3306	“Application security group”

Amazon Virtual Private Cloud (Amazon VPC)

Private Network

Security Group for fine grained control



At Rest Encryption

Leveraging AWS Key Management Service (KMS)

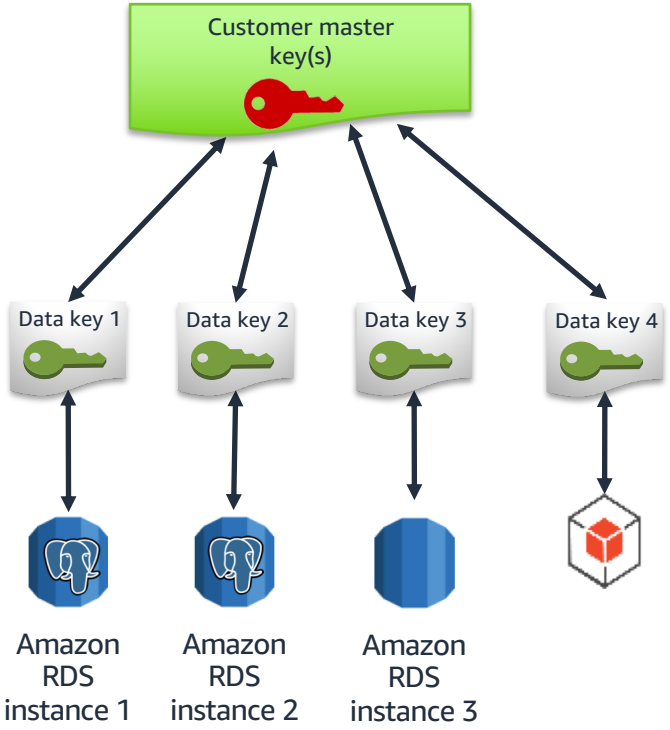
Default key available for encryption

- Cannot share outside account

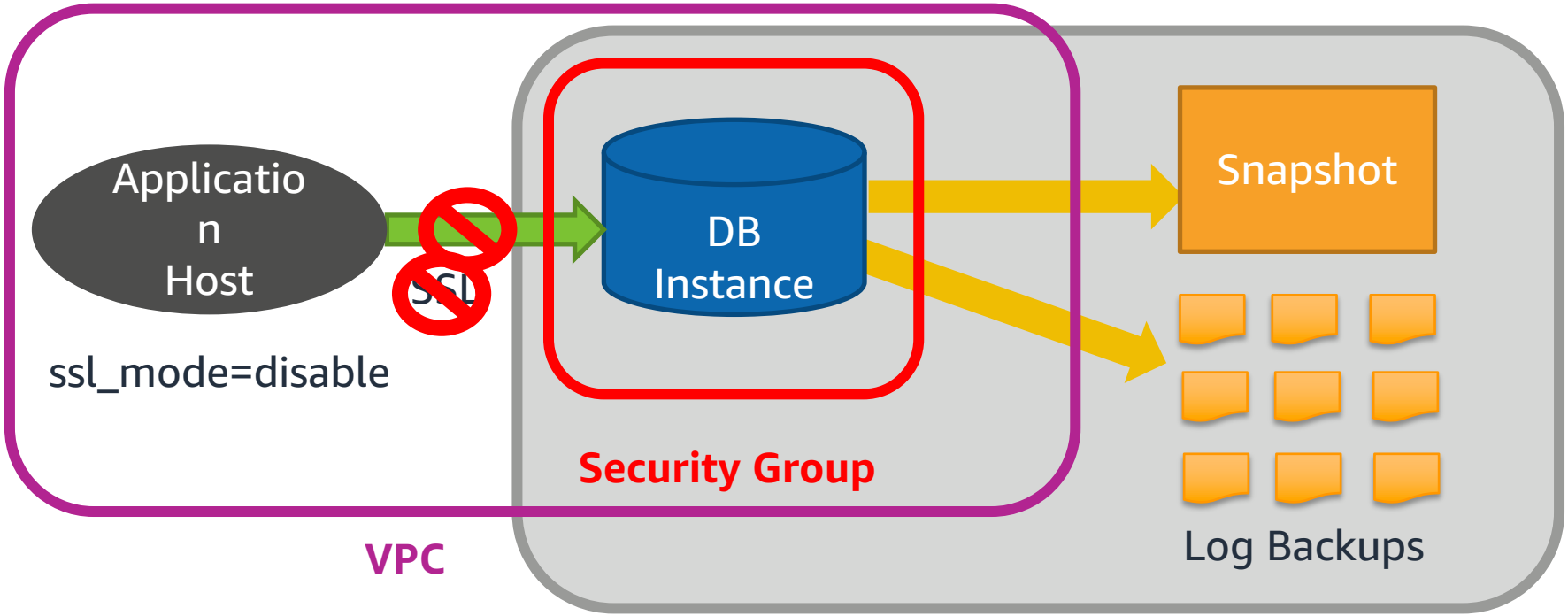
Recommended to security use separate keys for each instances

Benefits:

- Ability to share encrypted snapshots across accounts
- Limited risks of a compromised key



Forcing SSL on all connections



rds.force_ssl=1 (default 0)

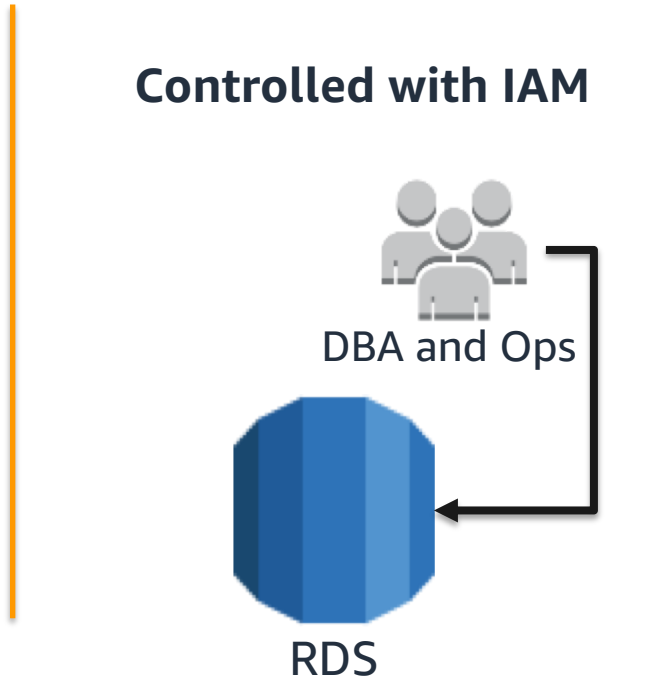
Encryption at Rest

IAM Governed Access

Use AWS Identity and Access Management (IAM) to control who can perform actions on RDS using web console or “aws” command line interface

Examples:

1. Deploy, Modify, delete instances
2. Create, delete, promote read replicas
3. Reboot, Start/Stop instance

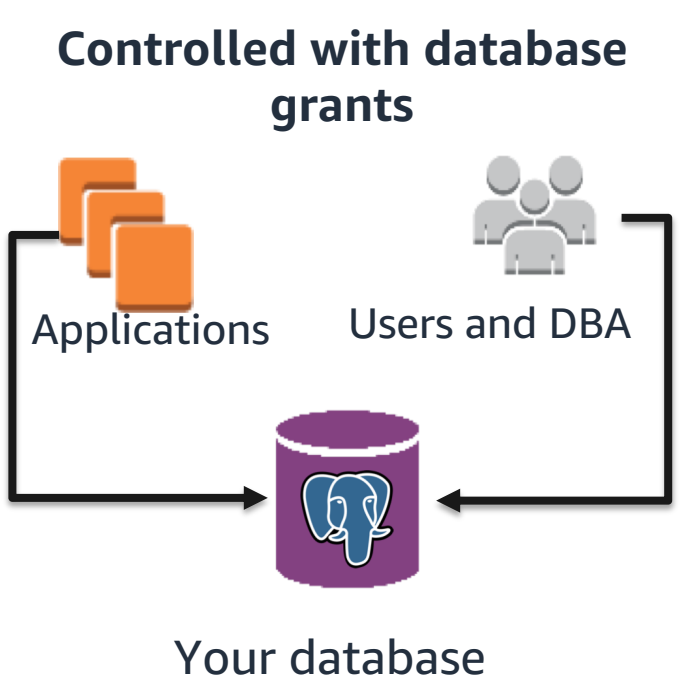


Database Users Access

Connects to the database instance using PostgreSQL Clients

Examples:

```
CREATE USER foobar
ALTER DATABASE testdb RENAME to proddb
```



AWS Identity and Access Management (IAM) authentication

AWS Identity and Access Management

PostgreSQL authentication is managed externally using IAM

- Available for Amazon RDS PostgreSQL and Aurora PostgreSQL

Authentication tokens are used to validate the user

- Tokens have a lifetime of 15 minutes
- Generated using AWS Signature Version 4

New role rds_iam available when IAM integration is enabled

Restrict Password changes on PostgreSQL instances

AWS Identity and Access Management

Simplifies integration of home grown or 3rd party password management tools

New database parameter to restrict password changes

- **rds.restrict_password_commands = on/off**

Flexibility to assign a role to allow certain users to

- GRANT rds_password TO tom;

aws

Compliance

EC2

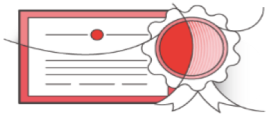
SOC 1, 2, 3
ISO 20001/9001
ISO 27107/27018
PCI
FedRAMP
HIPAA BAA
UK Gov. Programs
Singapore MTCS

RDS PostgreSQL

SOC 1, 2, 3
ISO 20001/9001
ISO 27107/27018
PCI
FedRAMP
HIPAA BAA
UK Gov. Programs
Singapore MTCS

Aurora

SOC 1, 2, 3
ISO 20001/9001
ISO 27107/27018
PCI
HIPAA BAA

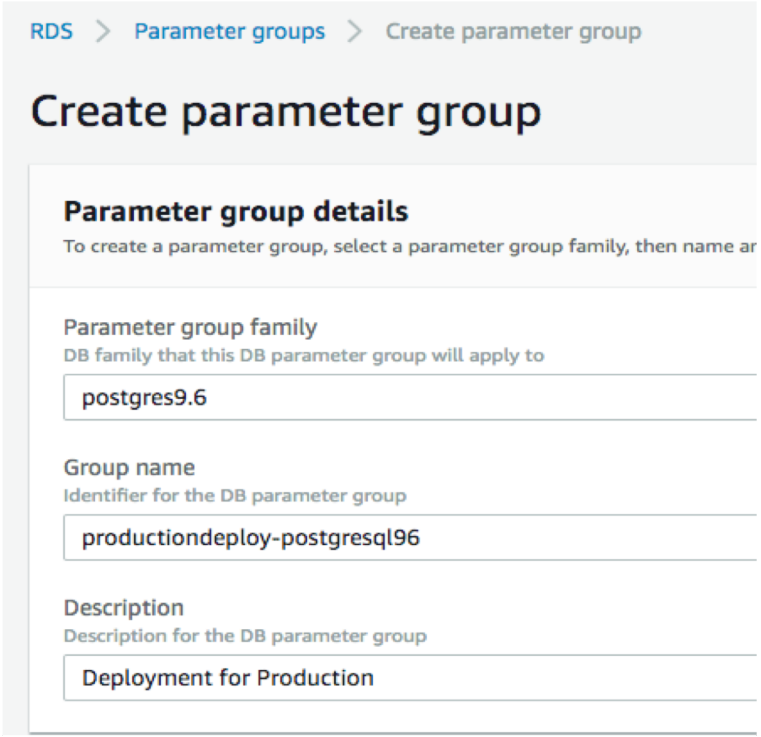


Singapore MTCS

27001/9001
27017/27018

Details: https://aws.amazon.com/compliance/services-in-scope/

Database Parameter Groups



Create a standard group based on your needs (IAM protected)

```
rds.force_ssl=true
shared_preload_libraries=pgaudit,
auto_explain,pg_stat_statements,pg_re
pack
pgaudit.role = rds_pgaudit
huge_pages = on
```

Use your standard group or copy to instance specific parameter group

- Avoid typos while editing parameters

Database server instance types

General purpose (T2/T3)

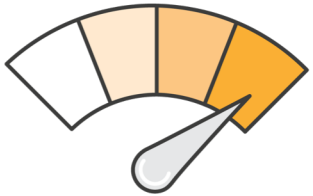
- 1 vCPU / 1 GB RAM to 8 vCPU 32 GB RAM
- Moderate networking performance
- Good for smaller or variable workloads
- T2.micro is eligible for free tier

General purpose (M4/M5)

- 2 vCPU / 8 GiB RAM to 96 vCPU 384 GiB RAM
- High performance networking
- Good for running CPU intensive workloads (e.g. WordPress)

Memory optimized (R4/R5)

- 2 vCPU / 16 GiB RAM to 96 vCPU 768 GiB RAM
- High performance networking
- Good for query intensive workloads or high connection counts



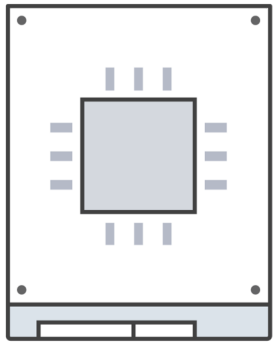
High performance database storage

General purpose (GP2)

- SSD storage
- Maximum of 32 TB
- Latency in milliseconds
- IOPS determined by volume size
- Bursts to 3,000 IOPS (applicable below 1.3 TB)
- Affordable performance

Provisioned IOPS (IO1)

- SSD storage
- Maximum of 32 TB
- Single digit millisecond latencies
- Maximum of 40 K IOPS
- Delivers within 10% of the IOPS performance 99.9% of the time
- High performance and consistency



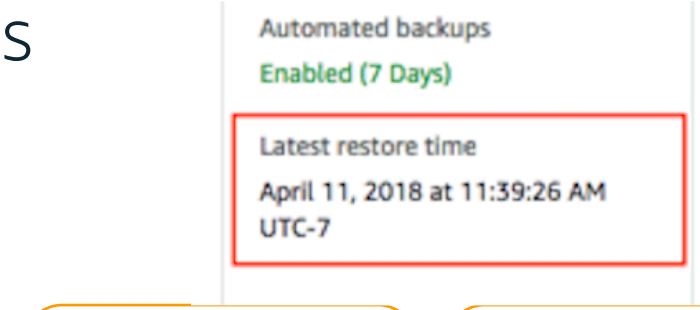
Automated Backup Snapshots

RDS PostgreSQL Snapshots

- Scheduled daily volume backup of entire instance
- Archive database change logs (WAL)
- 35-day retention
- Multiple copies in each AZ when running multi-AZ
- Taken from standby when running multi-AZ

Aurora PostgreSQL Snapshots

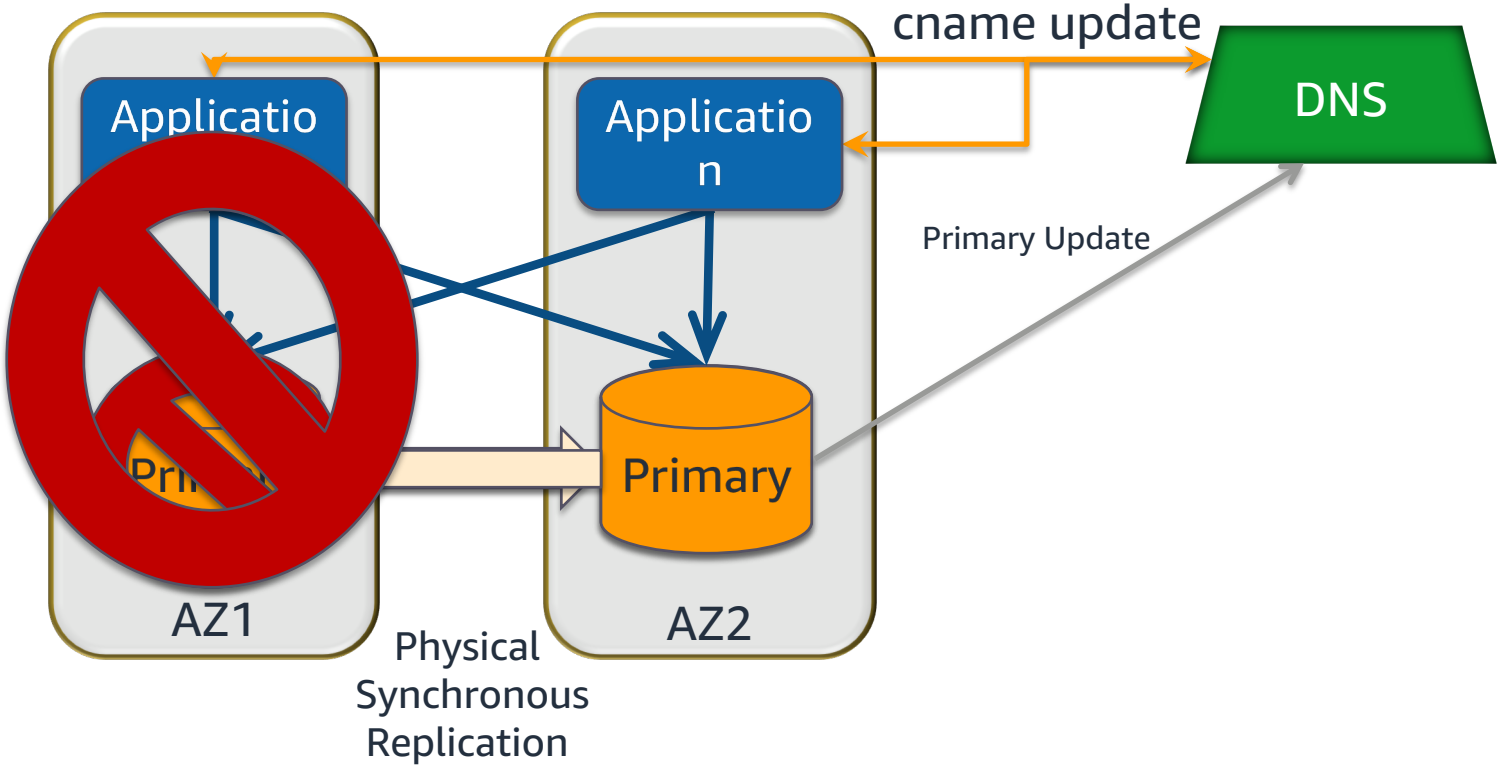
- Automatic, continuous, incremental backups
- No impact on database performance
- 35-day retention



Every day during your backup window, RDS creates a storage volume snapshot of your database

Every five minutes, RDS backs up the transaction logs of your database

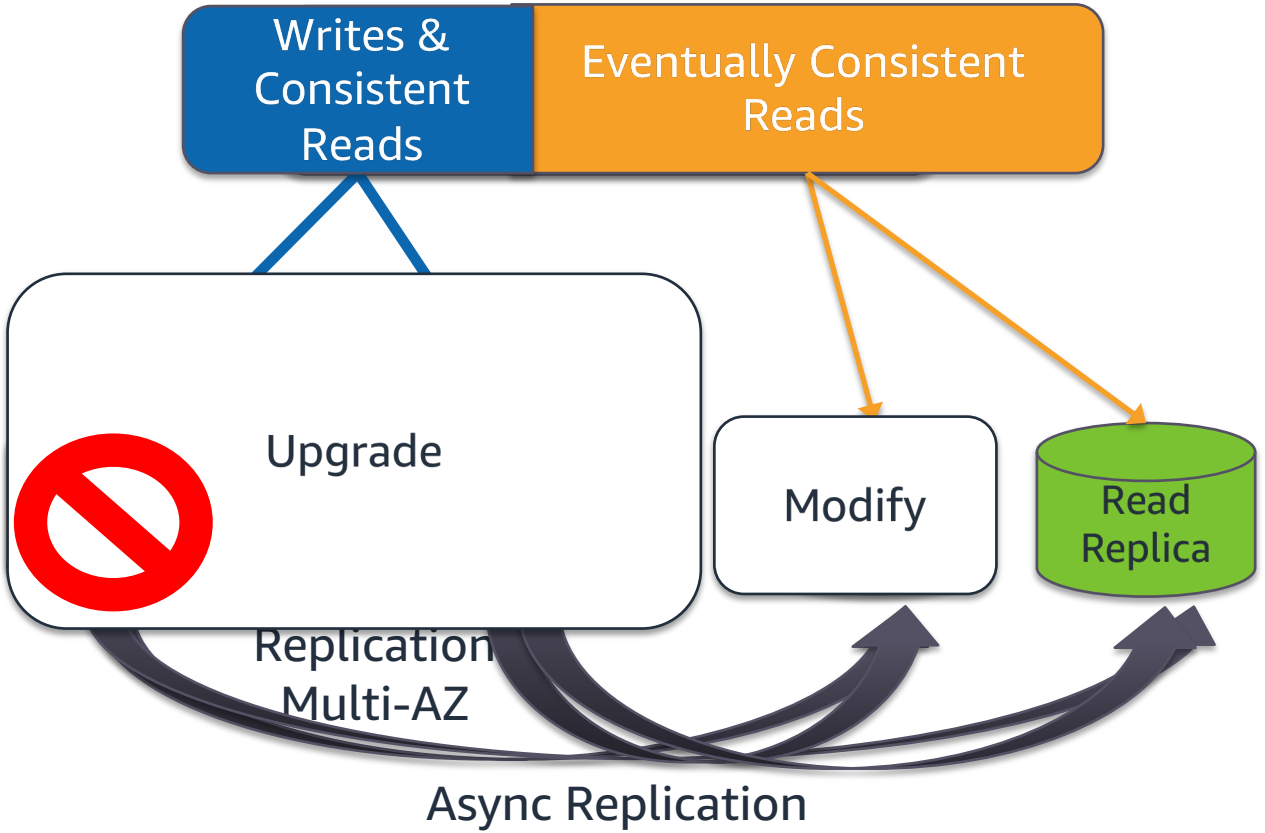
Availability – Read and Write – Multi-AZ



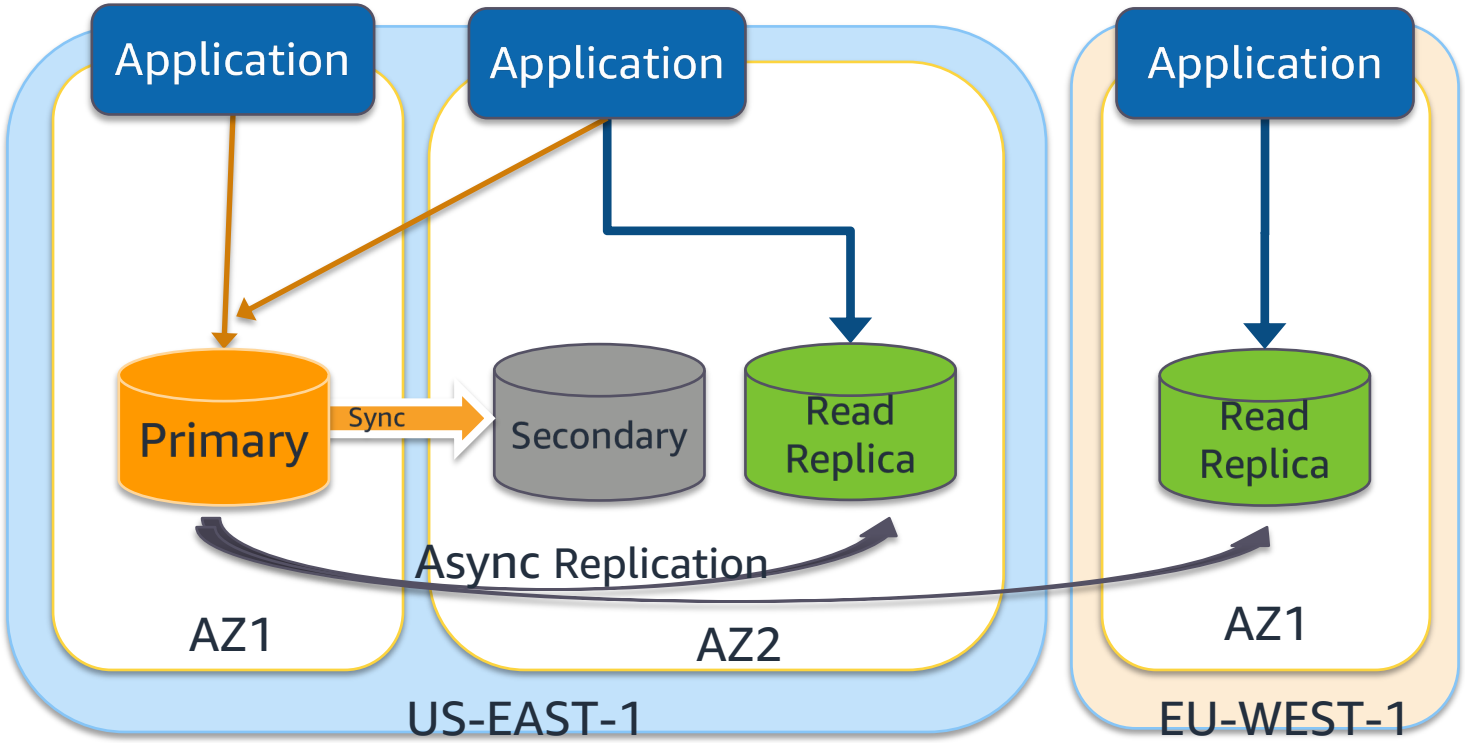
Read Replicas

- Relieve pressure on your ma
  with additional read capacit
- Bring data close to your app
  in different regions
- Promote a Read Replica to a
  for faster recovery in the ev
  disaster

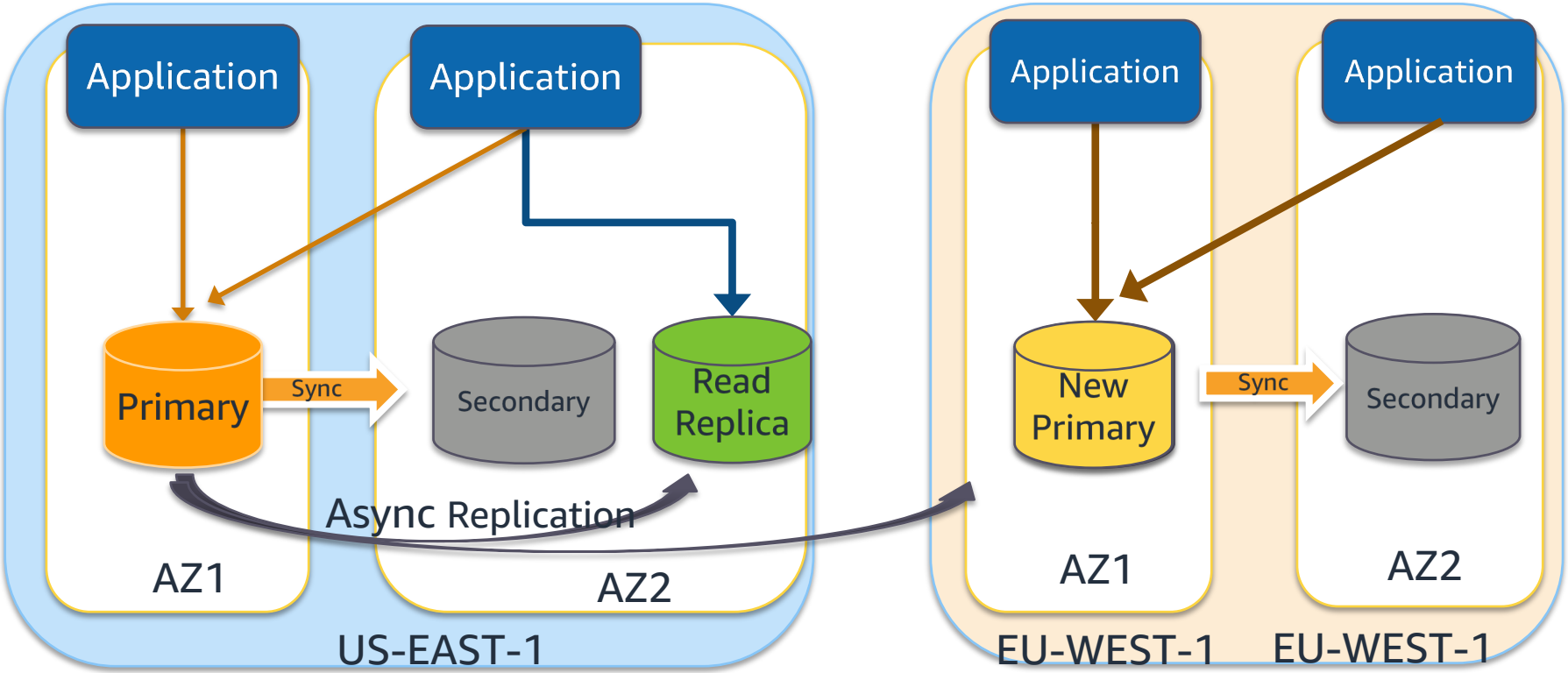
Read Replicas = Availability



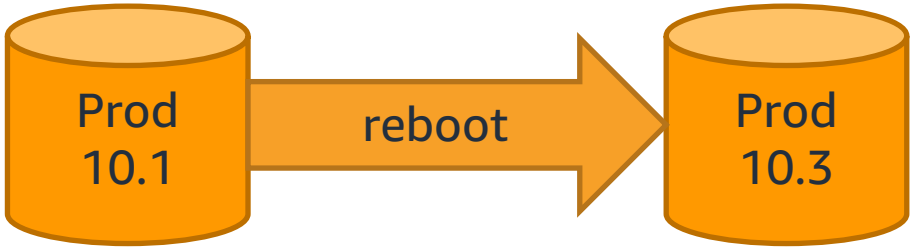
Cross Region Replicas – Reduce Latency



Cross Region Replicas – DR & Moves



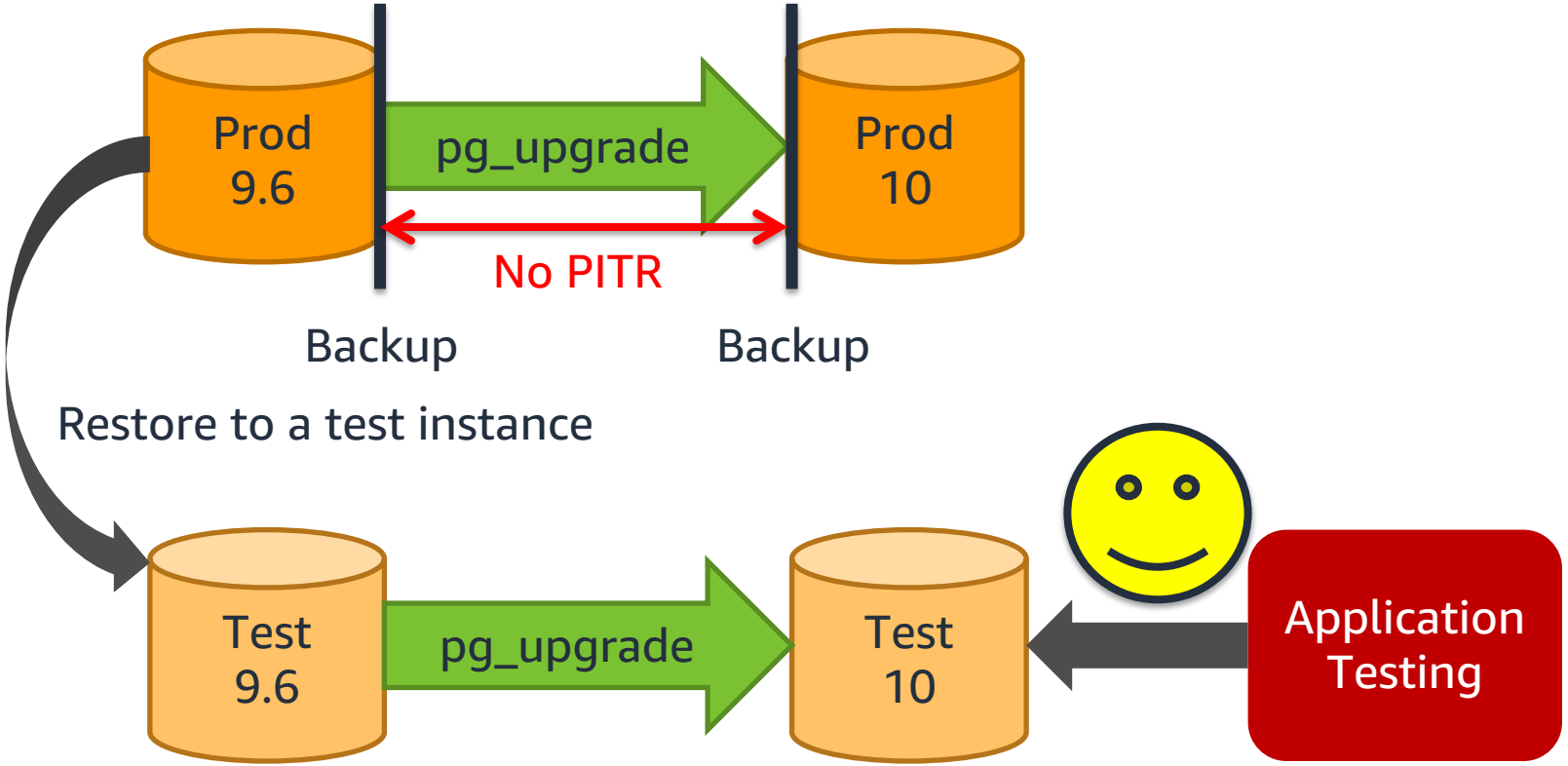
Minor version upgrade



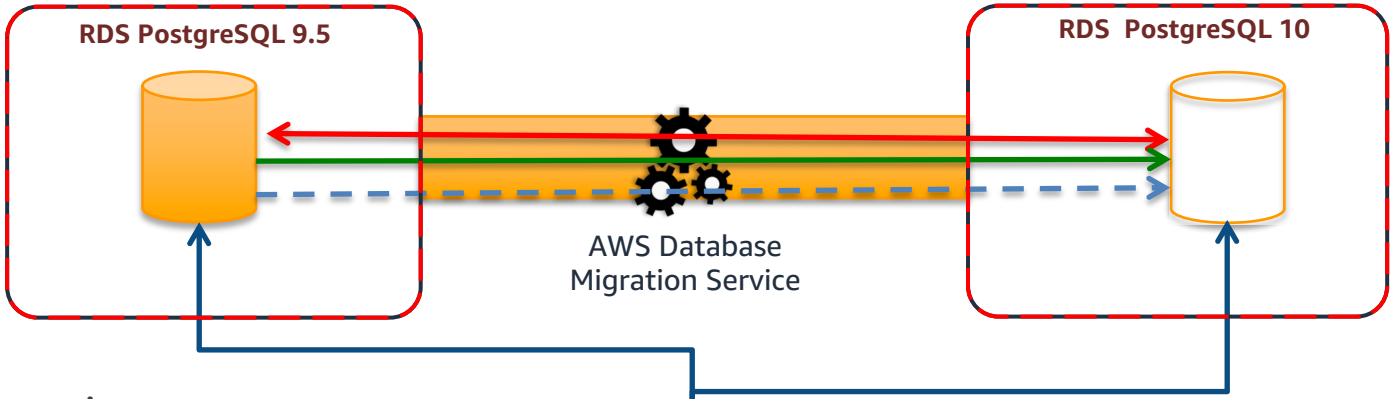
Minor Versions upgrade on managed services

- Shutdown instance
- Replace version binaries
- Start instance

Major version upgrade



Upgrade with minimum downtime using DMS



Create new target instance

Create Schema on Target (SCT)

Start a replication instance

Connect to source and target databases

Select tables, schemas, or databases

Application Users

Let the AWS Database Migration Service truncate tables and load data

Uses change data capture to keep them in sync

Switch applications over to the target at your convenience

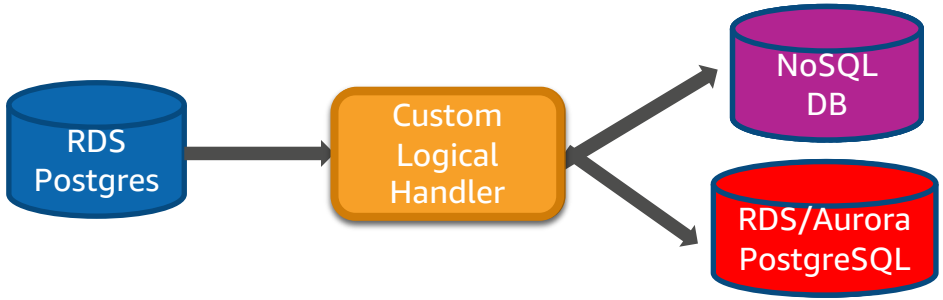
Logical Replication Support

- Set rds.logical_replication parameter to 1
- As user who has rds_replication & rds_superuser role

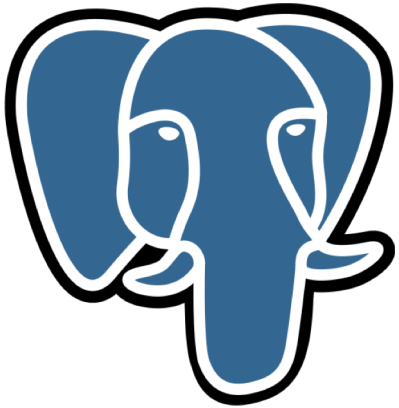
```
SELECT * FROM pg_create_logical_replication_slot('test_slot',
'test_decoding');
```

```
pg_recvlogical -d postgres --slot test_slot -U master --host $rds_hostname -f - --
start
```

- Support for Event Triggers
- Now support
  - Native Logical replication
  - Pglogical
  - wal2json
  - decoder_raw



Amazon RDS for PostgreSQL



Support for latest minor releases

- 10.6, 9.6.11, 9.5.15, 9.4.20

60+ extensions supported

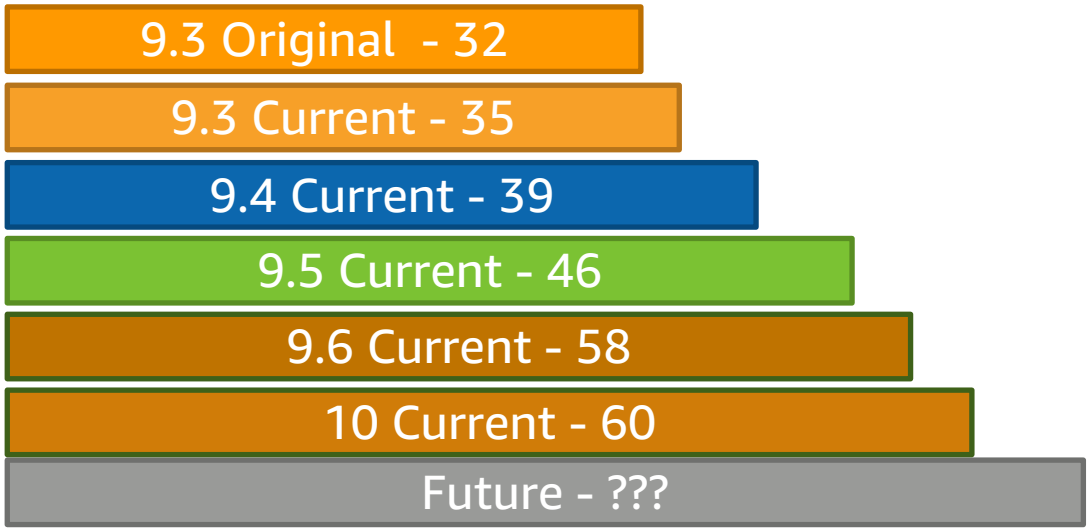
- pg_similarity, orafce, pageinspect, amcheck

PostgreSQL Version 11 available in preview

https://aws.amazon.com/rds/databasepreview/

aws

PostgreSQL Extensions/Modules Supported

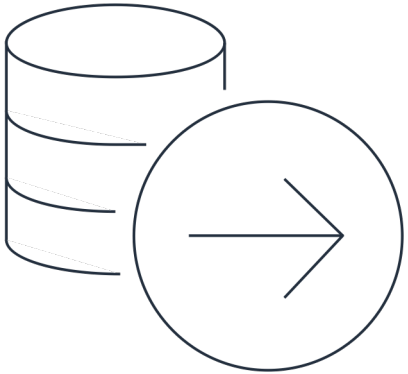


Email: rds-postgres-extensions-request@amazon.com

New PostgreSQL Extensions Supported

Extensions	Description
pglogical	Support logical replication –for PostgreSQL 9.6 and PostgreSQL 10
pg_similarity	Extension for supporting similar text queries
pageinspect	Allows to inspect the contents of database pages at a low level
protobuf	Enable Map Box Vector Tiles support in PostGIS
amcheck	Allows verify the logical consistency of the structure of indexes
orafce	Implements commonly used functions to ease migration from Oracle
prefix	Makes it easy to match prefix using @> operator

Replication in Amazon RDS PostgreSQL



Logical - SQL

- Statement based
- Trigger Based

Logical - Engine

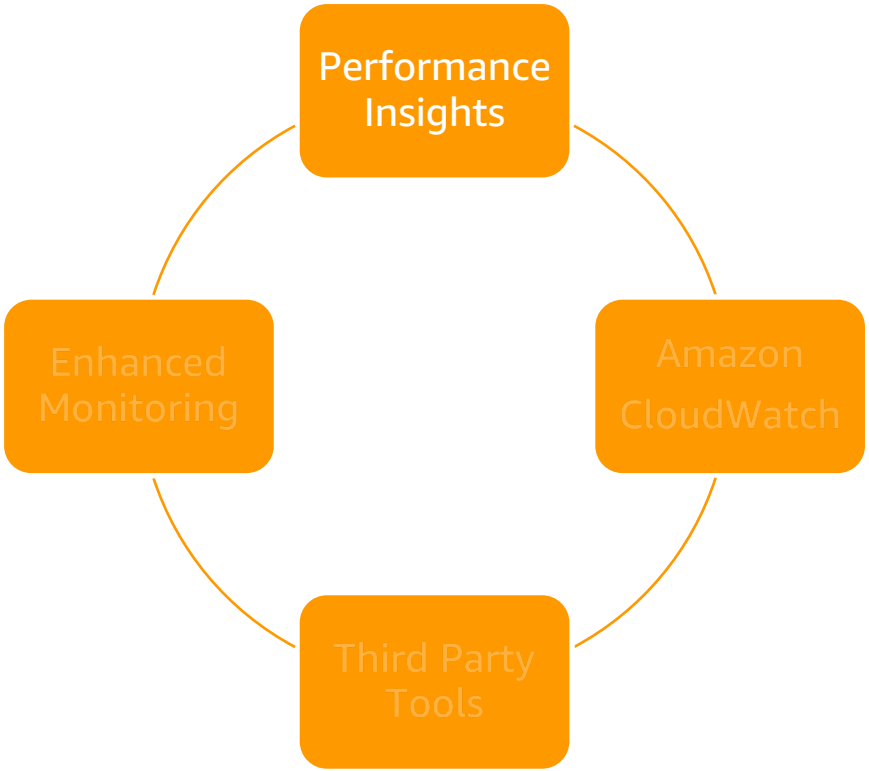
- Standard PostgreSQL
- Extension “pglogical”
- AWS DMS
- Third-party

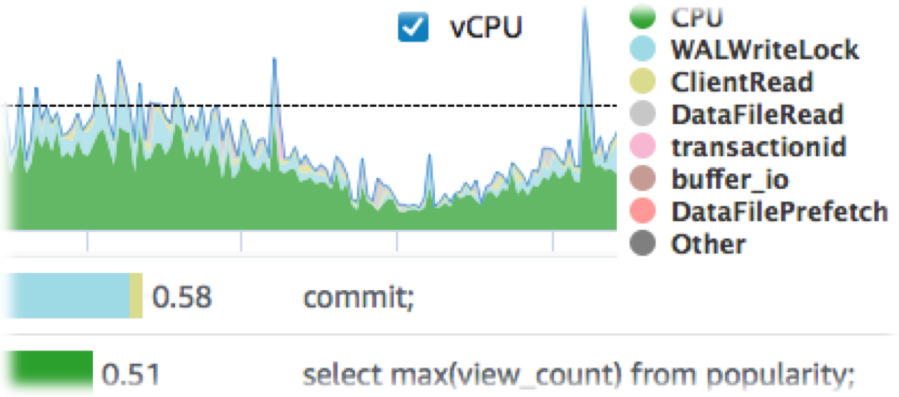
Physical - Engine

- Read replicas
- Multi-AZ

aws

Monitoring Amazon RDS

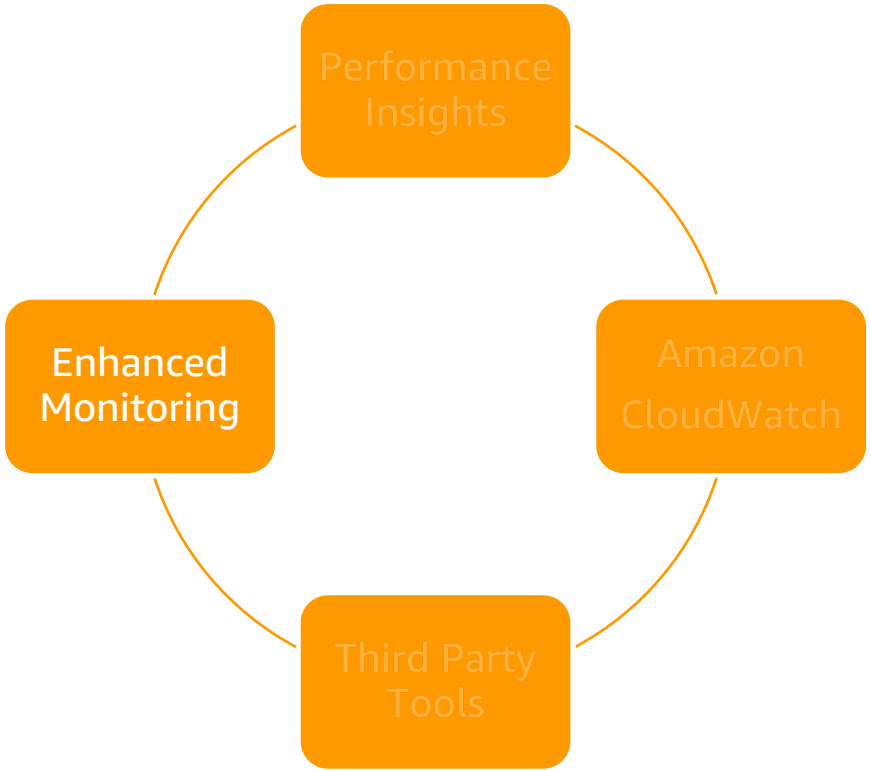




- Measures database load to help you identify bottlenecks
  - Top SQL/most intensive queries
  - Adjustable timeframe: hour, day, week, longer
- Compliments other key tools
  - query execution plans
  - pg_stat_statements

aws

Monitoring Amazon RDS



Operating system process list

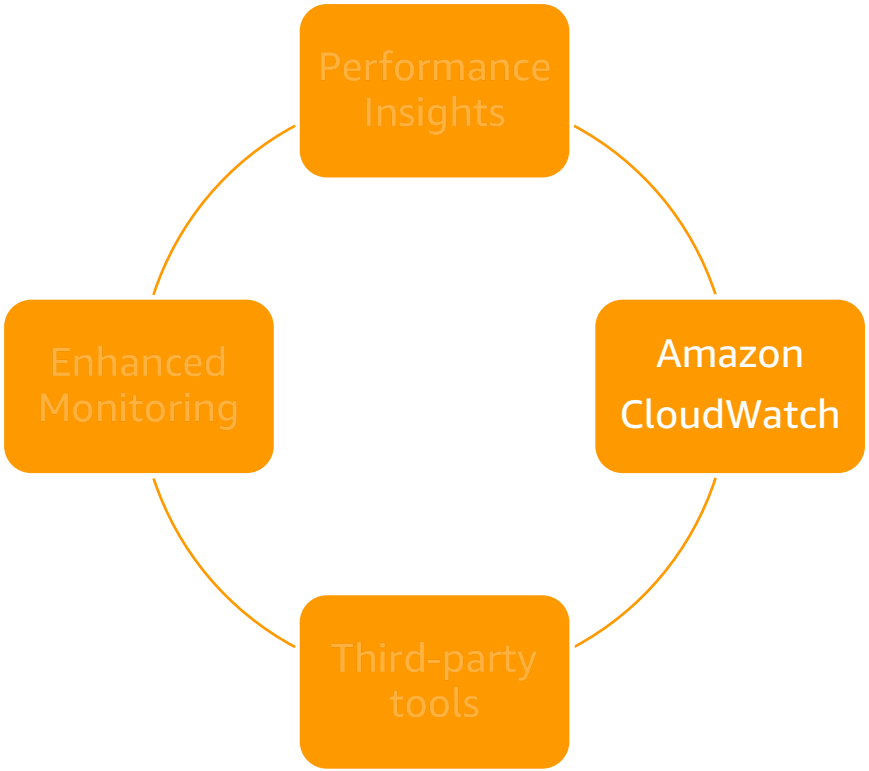
NAME	RES	CPU%
postgres [6329]	10.31 MB	64
postgres: writer process [6333]	6.44 MB	0
postgres: wal writer process [6334]	6.45 MB	0.5
postgres: wal sender process root 172.31.56.121(60... [31812]	11.3 MB	6.5
postgres: stats collector process [6337]	6.34 MB	0
postgres: root db1 172.31.63.69(42876) idle in tra... [20846]	12.59 MB	9.5

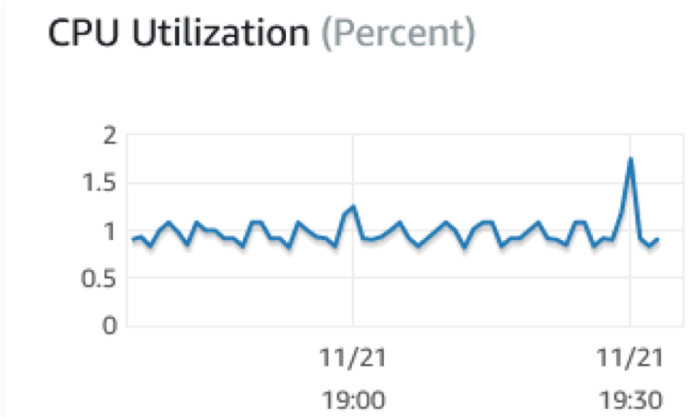
Enhanced Monitoring for Amazon RDS

- Access to over 50 CPU, memory, file system, and disk I/O metrics
- Access to top processes
- As low as 1 second intervals

aws

Monitoring Amazon RDS





Amazon CloudWatch metrics

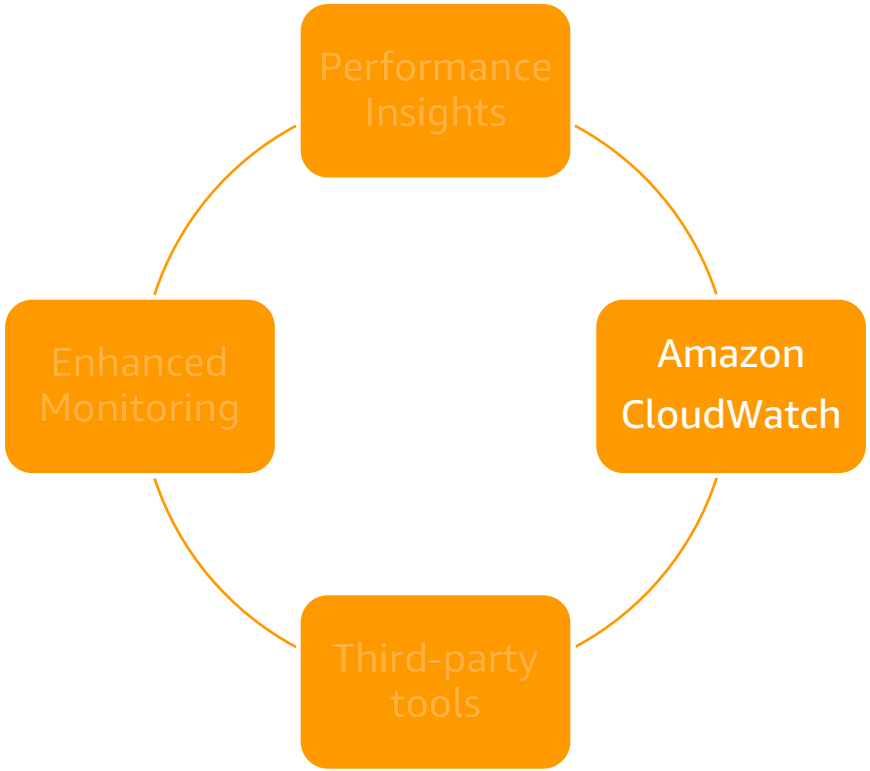
- Displayed in the Amazon RDS Console or personalized CloudWatch dashboards
- As low as one minute intervals

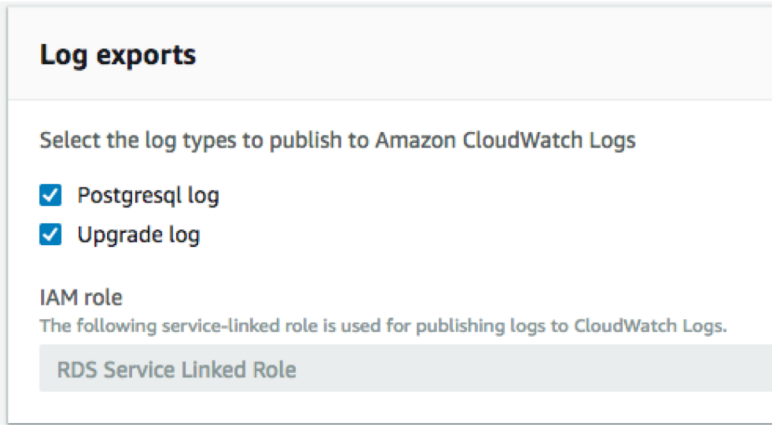
Amazon CloudWatch alarms

- Trigger actions based on a metric value relative to a threshold you set

aws

Upload PostgreSQL Logs to Amazon Cloudwatch





Amazon CloudWatch

- Upload PostgreSQL logs to CloudWatch
- Export logs to S3 from CloudWatch

aws

AWS Forums

EC2

- https://forums.aws.amazon.com/forum.jspa?forumID=30

Amazon RDS

- https://forums.aws.amazon.com/forum.jspa?forumID=60

Aurora with PostgreSQL compatibility

- https://forums.aws.amazon.com/forum.jspa?forumID=227

Thank you!
aws.amazon.com/rds/postgresql