Tips and Tricks with Amazon RDS for PostgreSQL

Jignesh Shah

Amazon RDS for PostgreSQL

PGConf RU 2019

Amazon Relational Database Service

Managed relational database service with a choice of popular database engines

Easy to administer

No need to provision infrastructure, install, and maintain DB software

Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover

Highly scalable

Scale DB compute and storage with a few clicks; minimal downtime for your application

Fast & secure

SSD storage and guaranteed provisioned I/O; data encryption at rest and in transit

aws

Amazon RDS for PostgreSQL

Supports

- Multiple major versions of PostgreSQL
    - 10, 9.6, 9.5 and 9.4
- Up to 96 vCPU, 768 GiB RAM,
- Up to 32TiB Storage, 40K IOPS
- Multi-AZ for High Availability
- Encryption at rest
- Start and Stop of PostgreSQL Instances
- In-region and Cross region Read Replicas

aws

Amazon RDS for PostgreSQL

Supports:

- 60+ PostgreSQL Extensions and decoder plugins
- In-built Native Logical replication
- Replication to Amazon Aurora PostgreSQL
- RDS Performance Insights for PostgreSQL

Amazon RDS Database Preview environment

- PostgreSQL 11 available

aws

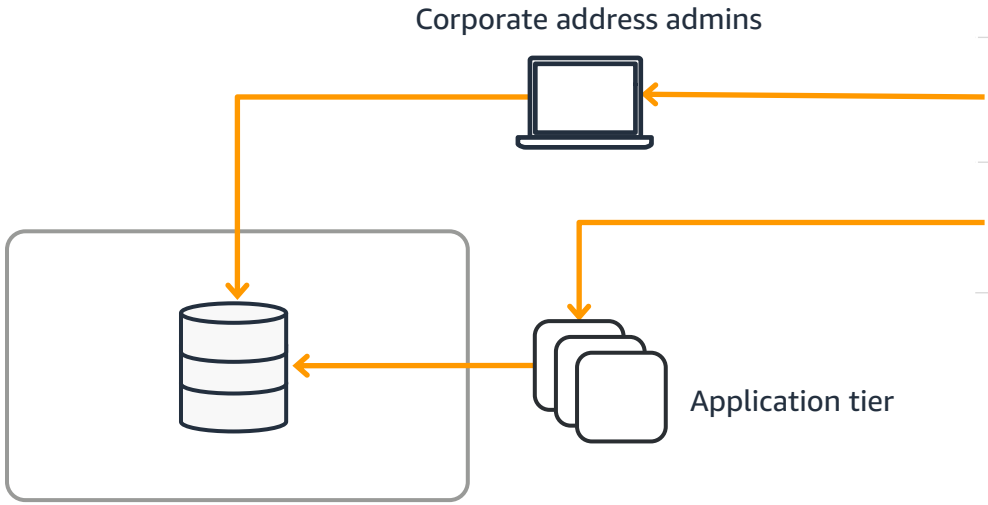
New PostgreSQL Extensions Supported

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

Tips for Amazon RDS

Tip #1: Secure network access with Amazon VPC

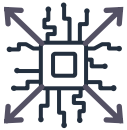
Security groups provide a network firewall for your database server



Protocol	Port range	Source
TCP	5432	172.31.0.0/16
TCP	5432	“Application security group”

Tip #2: Scale your compute and storage

Right Scale your instance size and storage before your peaks



Scale compute up to handle increased load or down to save costs

- As little as 1vCPU / 1 GiB of RAM
- Up to 64 vCPU and 488 GiB of RAM
- Minimal disruption for compute scaling

Scale storage for larger data sets

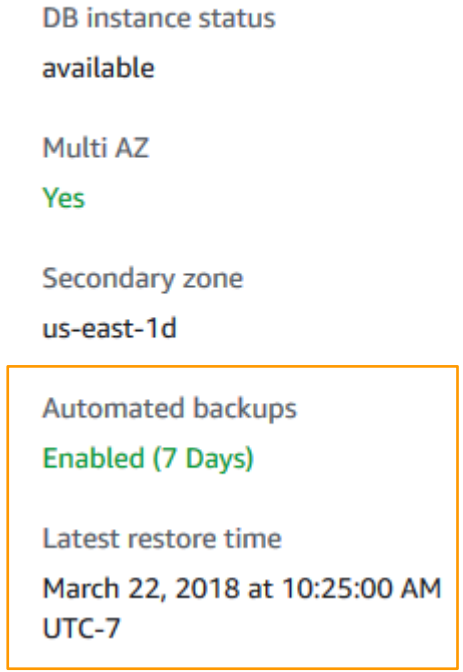
- Scalable Amazon EBS storage up to 16 TiB
- General Purpose SSD or Provisioned IOPS SSD storage
- No downtime for storage scaling
- Amazon Aurora automatically scales to 64 TiB

aws

Tip #3: Turn on automated backups

Point-in-time recovery for your DB instance

- Scheduled daily volume backup of entire instance
- Archive database change logs
- 35–day maximum retention
- Negligible impact on database performance
- Taken from standby when running Multi-AZ



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

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

aws

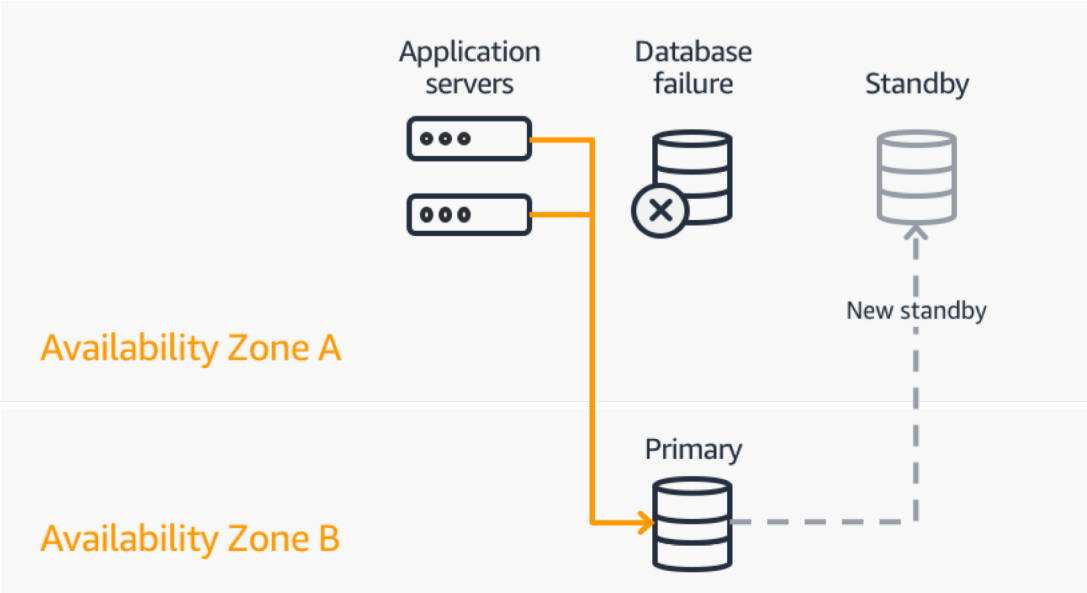
Tip #4: Enable Multi-AZ configuration

Enterprise-grade high availability

Fault tolerance across multiple data centers

- Automatic failover
- Synchronous replication
- Enabled with a few clicks

Redirection to the new primary instance is provided through DNS

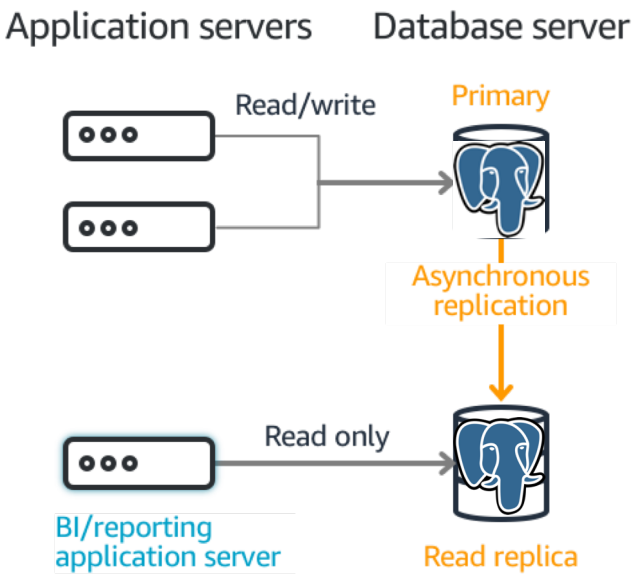


aws

Tip #5: Create read replicas

Read scaling and disaster recovery

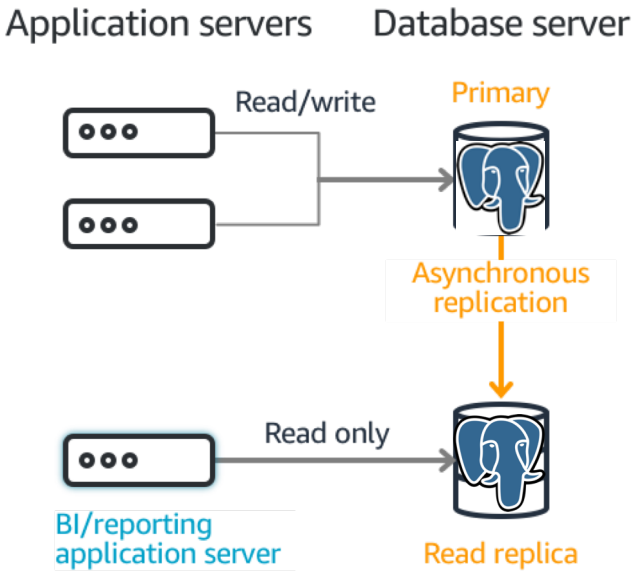
- Relieve pressure on your master node with additional read capacity
- Bring data close to your applications in different regions
- Promote a read replica to a master for faster recovery in the event of disaster
- Low latency replicas for Amazon Aurora can be used for both read scaling and failover targets



Tip #5: Read Replicas (Contd)

Read scaling and disaster recovery

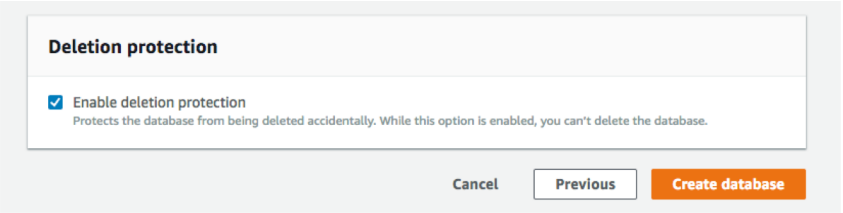
- Outbound access (FDW access) from read replica also available
- Faster DR with Multi-AZ Read Replicas



Tip #6: Delete Protection for Production

Prevent accidental deletes

- Prevent accidental deletes from CLI, Console, CloudFormation, etc.



Tip #7: Automated Minor Version Upgrade

Keep up with new minor releases

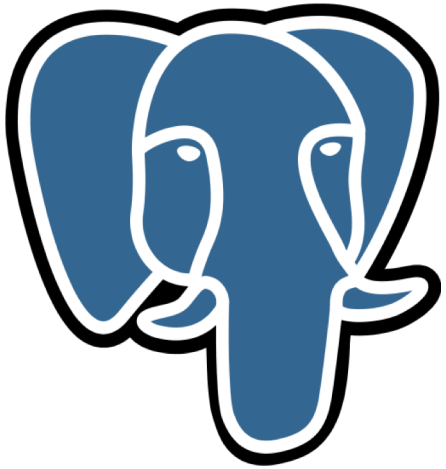
- Minor versions are upgraded in maintenance window when the option is selected

Tip #8: Upgrade from PostgreSQL 9.3 databases

First released by community in 09/2013

End of support in Amazon RDS in 11/2018

Preferred to be upgraded to PostgreSQL 10 or 9.6



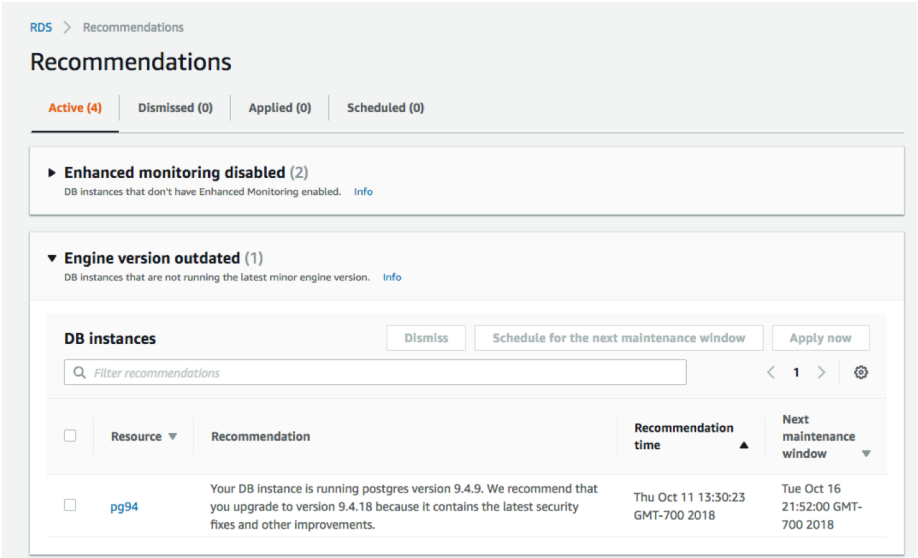
PostgreSQL 9.3

aws

Tip #9: RDS Recommendations

Most common things to do

- Provide best practice guidance for customers by analyzing configuration and usage metrics
- Initial launch includes 7 configuration-based recommendations
- Results are presented in the AWS Console so they can be applied immediately (if relevant)



Tip #10: Aurora Read Replica or Snapshot Import

Planning to upgrade to Amazon Aurora PostgreSQL?

Supports Snapshot import (encrypted snapshot is also available)

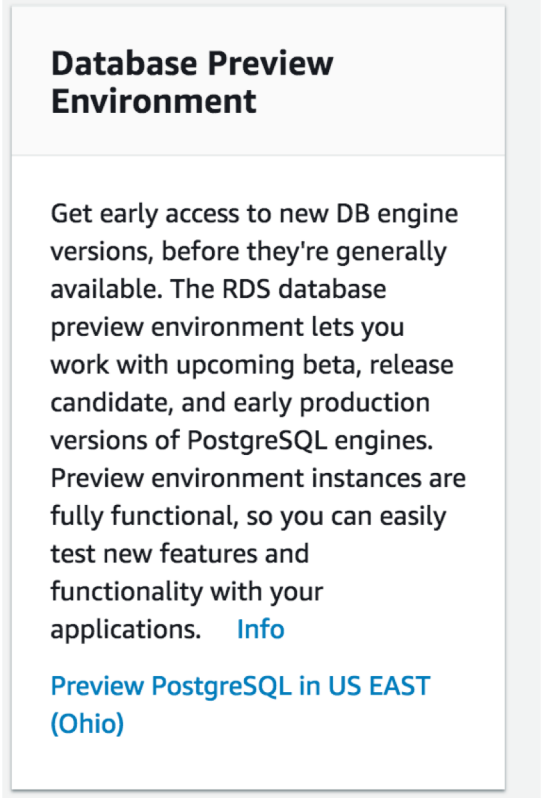
Create Aurora Read Replica for RDS PostgreSQL Promote Aurora Read Replica to Aurora PostgreSQL Cluster

Tip #11: Amazon RDS Database Preview

New upcoming PostgreSQL major versions available

Perfect for development and testing

PostgreSQL 11 beta 3 available



aws

Tricks with Amazon RDS for PostgreSQL

Trick #1: Optimizing Replica Lag

Read Replicas use asynchronous replication

Increase `wal_keep_segments` to allow replicas to catch up after interruption

Use `wal_compression = on`

Use higher `checkpoint_timeout`

Use `hot_standby_feedback = on` on read replicas

Use similar sized Instance size

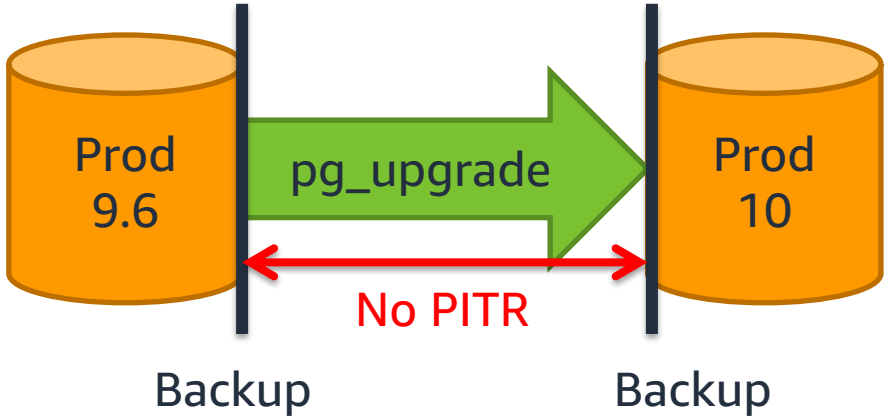
Lag for replicas vs long running queries on replicas

- `max_standby_archive_delay`
- `max_standby_streaming_delay`

Trick #2: Major Version Upgrade

A single version is not life long

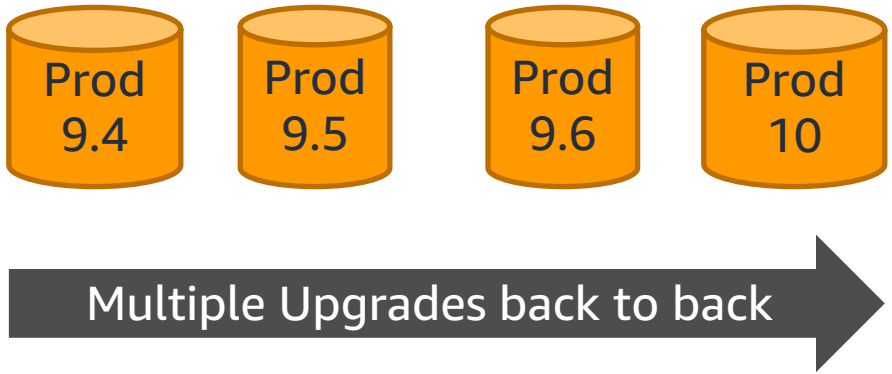
- Automated minor version upgrade already takes care of your minor versions
- Recommended to do major version upgrade to the current major version once 2-3 years
- Highly recommended to test applications with new versions



Trick #2: Major Version Upgrade (Contd)

Multiple versions helps to save another downtime next year

- Create with a read replica of your production without Multi-AZ
- Stop workload on production server
- Promote Read replica with 0 days backup
- Use major version upgrade multiple times to get to latest preferred major version
- Enable Automated Backup (and Multi-AZ protection)
- Spin up more read replicas (Use aws CLI to spin multiple at the same time



aws

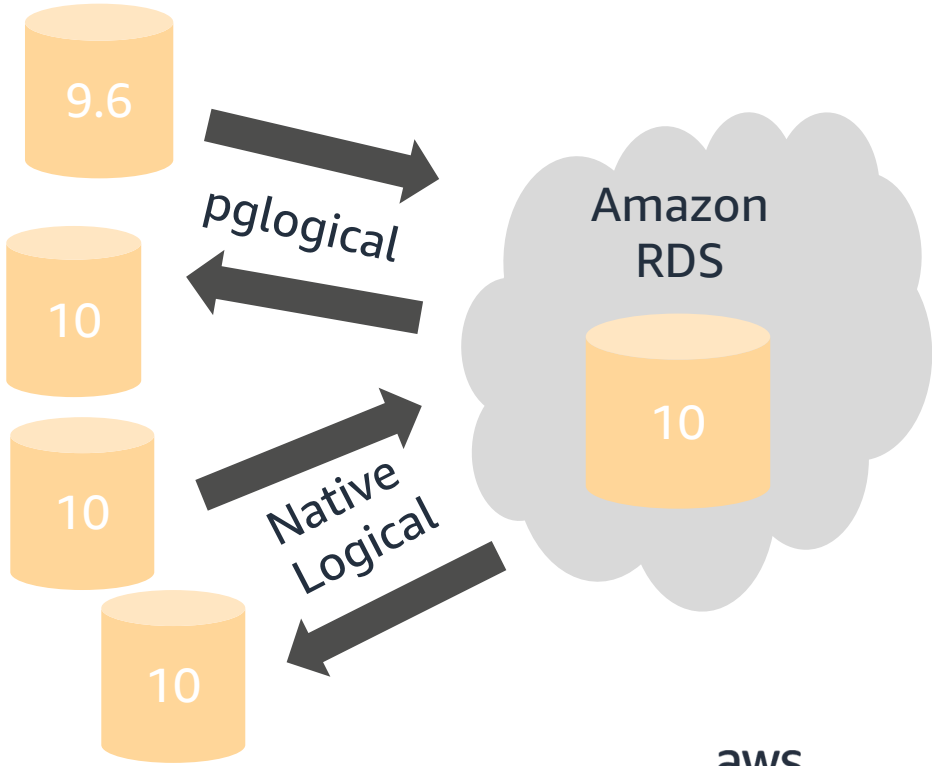
Trick #3: On-Premise to Amazon RDS replication

Introducing inbuilt logical replication and pglogical extension

- PostgreSQL 10 supports inbuilt logical replication using publication/subscription
- Pglogical Extension available in Amazon RDS for PostgreSQL 9.6 and 10

For source in Amazon RDS

- Requires setting rds.logical_replication to “on” on source
- Set role rds_replication to the user



Trick #4: IAM Authentication Support for PostgreSQL

Access and Password Management

- Use IAM to control who can perform actions on Amazon RDS resources
- Do not use AWS root credentials to manage Amazon RDS resources—you should create an IAM user for everyone, including yourself
- Can use AWS multi-factor authentication (MFA) to provide extra level of protection
- Can use IAM to control access to the database


```
{
"Version": "2012-10-17",
"Statement": [
 { "Effect": "Allow", "Action": [
     "rds-db:connect"
    ],
    "Resource": [
      "arn:aws:rds-db:us-east-
2:1234567890:dbuser:db-
12ABC34DEFG5HIJ6KLMNOP78QR/jane_doe"
    ]
  }
  ]
}
```

Trick #5: Restrict Password changes on PostgreSQL

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

Trick #6: Performance Tuning

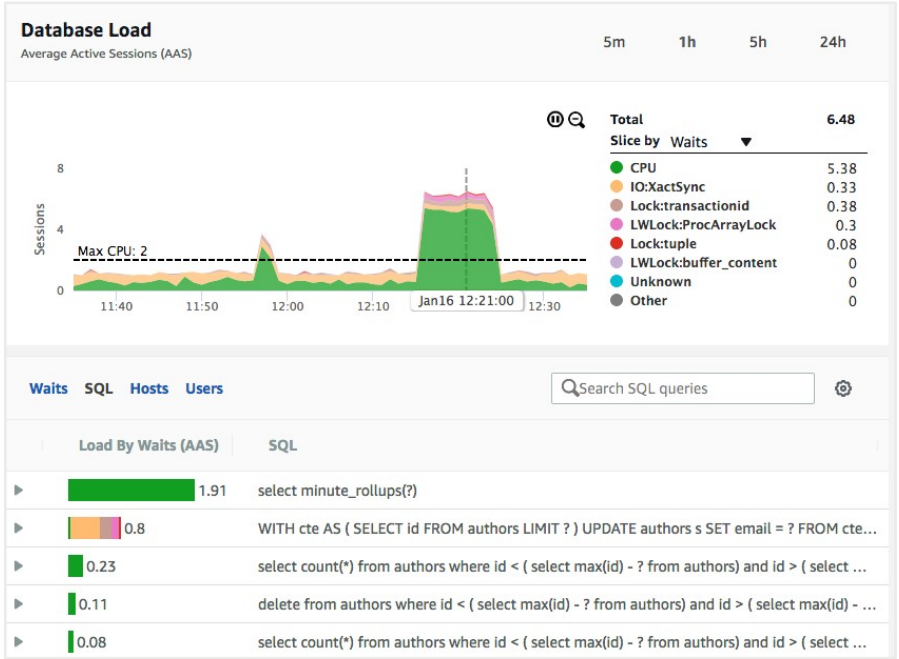
Tuning results in lower resources and helps to save costs

Enable Amazon RDS Performance Insights and Enhanced Monitoring

- DB load: average active sessions
- Adjustable timeframe
  - Hour, day, week, and longer

Enable `auto_explain, pg_stat_statements extensions.`

Need libraries to shared_preload_libraries. (Requires restart)



Trick #6: Performance Tuning (Contd)

Tuning results in lower resources and helps to save costs

Set following values in db parameters

- `auto_explain.log_min_duration = 5000`
- `auto_explain.log_nested_statements = on`

Trick #6: Performance Tuning (Contd)

Tuning results in lower resources and helps to save costs

```
CREATE EXTENSION pg_stat_statements
SELECT * from pg_stat_statements order by total_time
DESC;
```

```
-[ RECORD 2 ]-------+--------
userid              | 16388
dbid                | 16464
queryid             | 4286627671
query               | UPDATE pgbench_accounts SET abalance = abalance + ? WHERE aid
= ?;
calls               | 165125
total_time          | 5251.54200000001
min_time            | 0.015
max_time            | 5.558
mean_time           | 0.0318034337623008
stddev_time         | 0.0369181019548524
rows                | 165125
```

Trick #7: Connection Poolers

Warm pool is good but can increase memory usage

Connection Pools can be

- Centralized (pgpool, pgbouncer)
- Decentralized (C3P0, inbuilt application ones)

Use intelligent minimums and maximums

Use max_age or equivalent

Helps reclaim work_mem bloat

Use max_idle or equivalent

For large connections, verify hugepages = on

aws

Trick #8: Bloat Management

Lean and Fit results in better performance also

MVCC in PostgreSQL

- DELETE operations mark a row as dead
- UPDATE operations are essentially
  - Marking current row version as dead
  - Row with new version is added
- Eventually, all dead rows need to be garbage collected

VACUUM utility

- Cleans up dead rows but does not release space to OS
- FULL option also reorganizes the table to release space to OS

Trick #8: Bloat Management (Contd)

Lean and Fit results in better performance also

Reorganizes Table with a very short lock

- Must have PRIMARY key or UNIQUE NOT-NULL key

Quick Setup

- `CREATE EXTENSION pg_repack`
- Use pg_repack client utility using rds_superuser privileges with –k option

```
pg_repack -h myproductiondb.XXXXXXXX.us-west-
2.rds.amazonaws.com -U pgadmin -k postgres
```

Trick #9: Preempt Transaction Wrap around

This is not a drill

For 24/7 constant load on Database server

- AUTOVACUUM may not get a chance to finish its job

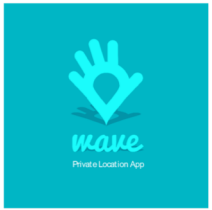
For high number of tables

- Increase `autovacuum_max_workers` from default 3 to higher number
- Increase `autovacuum_vacuum_cost_limit`
- Note: With this change, there may be performance impact

For large tables use

- Decrease `autovacuum_vacuum_scale_factor` from 0.2 (20%) to 0.05 (5%)
- `ALTER TABLE myablename SET autovacuum_scale_factor = 0.02`

Amazon RDS PostgreSQL Customers



Questions or comments?

Thank you!