Securing Your Database in Amazon RDS

Sarah Conway

September 17, 2015
Agenda

- Intro
  - What is Amazon RDS?
  - Features
  - Pricing
  - Setting Up & Connecting Tutorial
  - Data Ownership
  - PCI Compliance Certifications

- User & Database Management
  - Best User Management Practices
  - Database Management

- Network Management
  - Encryption
  - VPC Security Groups
Agenda cont.

- Auditing & Monitoring
  - Auditing
  - Analyzing Logs
  - Monitoring
- Backups & Recovery
  - Automated Backups
  - DB Snapshots
- End
  - Questions
What is Amazon RDS?

- Amazon Relational Database Service
- Managed service
- Good for DBA’s, DevOps, Sysadmin, etc
- Versions 9.3.1 - 9.4.4 available
Features

- PostGIS
- Language Extensions (PL/V8, PL/Perl, PL/Python)
- Full Text Search Dictionaries
- HStore, JSON Data Types
- pg_stat_statements
- postgres_fdw
- auto-minor-version-upgrade
- Multi-AZ
Cons

- Limited control
- Can’t force SSL connections
- Not a solution for a company with extensive privacy needs
Pricing

- Varies depending on additional services selected
- Priced by usage
- Online cost calculator
  
  http://calculator.s3.amazonaws.com/index.html
- Billing alerts and notifications
  
  http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/monitor-charges.html
Setting up a RDS Instance

Select Engine

To get started, choose a DB Engine below and click Select.

MySQL

PostgreSQL

Oracle

SQL Server

Select

Cancel
Setting up a RDS Instance

Do you plan to use this database for production purposes?

- Yes, use Multi-AZ Deployment and Provisioned IOPS Storage as defaults while creating this instance.
- No, this instance is intended for use outside of production or under the RDS Free Usage Tier.

For databases used in production or pre-production we recommend:
- Multi-AZ Deployment for high availability (99.95% monthly up time SLA)
- Provisioned IOPS Storage for fast, consistent performance

Billing is based upon the RDS pricing table.
Setting up a RDS Instance


Sarah Conway  Postgres Open 2015
Setting up a RDS Instance

![Diagram of RDS setup interface]

- **VPC**: Default VPC
- **Subnet Group**: default
- **Publicly Accessible**: Yes
- **Availability Zone**: No Preference
- **VPC Security Group(s)**: Create new Security Group
  - Default VPC Group

**Database Options**

- **Database Name**: postgres
- **Database Port**: 5432
- **DB Parameter Group**: default_postgres
- **Option Group**: default_postgres
- **Copy Tags To Snapshots**: No
- **Enable Encryption**: No

**Backup**

- **Backup Retention Period**: 35 days
- **Backup Window**: No Preference

**Maintenance**

- **Auto Minor Version Upgrade**: Yes
- **Maintenance Window**: No Preference

*Select the number of days, between 1 and 35, that Amazon RDS should retain automatic backups of this DB instance. The backup retention period determines the period for which you can perform a point-in-time recovery. Select 0 to disable backups. Learn More.*
Setting up a RDS Instance

VPC Dashboard
Filter by VPC:
None

Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Peering Connections

Security
Network ACLs
Security Groups

VPN Connections
Customer Gateways
Virtual Private Gateways
VPN Connections

sg-c47d02a1

Cancel Save

Type Protocol Port Range Source Remove
ALL TCP TCP (6) ALL sg-c47d02a1
SSH (22) TCP (6) 22

Add another rule
Setting up a RDS Instance

```
[root@ip-192-168-1-233 ~]# psql -h middle-earth.rds.amazonaws.com -U saruman -d isengard
Password for user saruman:
psql (9.2.13, server 9.4.4)
Some psql features might not work.
SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)
Type 'help' for help.

isengard=>
```
Setting up a RDS Instance

A screenshot showing the configuration options for an RDS instance, including inbound and outbound rules.

This image is a part of a presentation slide from Postgres Open 2015.
Setting up a RDS Instance

```
$coben@rapture ~/Documents $ psql -h middle-earth. rds.amazonaws.com -p 5432 -U saruman -d isengard
Password for user saruman:
psql (9.3.5, server 9.4.4)
Some psql features might not work.
SSL connection (cipher: ECDHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

isengard=>
```
8.1 Your Content. As between you and us, you or your licensors own all right, title, and interest in and to Your Content.

3.2 Data Privacy. We will not move Your Content from your selected AWS regions without notifying you, unless required to comply with the law or requests of governmental entities.

https://aws.amazon.com/agreement
Certified PCI Service Provider

http://d0.awsstatic.com/whitepapers/compliance/AWS_Anitian_Wookbook_PCI_Cloud_Compliance.pdf

Create an IAM Role

Create Role

Step 1: Set Role Name
Step 2: Select Role Type
Step 3: Establish Trust
Step 4: Attach Policy
Step 5: Review

Set Role Name

Enter a role name. You cannot edit the role name after the role is created.

Role Name: S3-Admin

Maximum 64 characters. Use alphanumeric and '-', '@', '_'. characters
Create an IAM Role

Select Role Type

- AWS Service Roles
  - Amazon EC2
    - Allows EC2 instances to call AWS services on your behalf.
  - AWS Directory Service
    - Allows AWS Directory Service to manage access for existing directory users and groups to AWS services.
  - AWS Lambda
    - Allows Lambda Function to call AWS services on your behalf.
  - AWS Config
    - Allows AWS Config to call AWS services and collect resource configurations on your behalf.
  - AWS SWF
    - Allows SWF workflows to invoke Lambda functions on your behalf.
- Role for Cross-Account Access
- Role for Identity Provider Access
Create an IAM Role

Create Role
Step 1: Set Role Name
Step 2: Select Role Type
Step 3: Establish Trust
Step 4: Attach Policy
Step 5: Review

Review
Review the following role information. To edit the role, click an edit link, or click Create Role to finish.

<table>
<thead>
<tr>
<th>Role Name</th>
<th>Role ARN</th>
<th>Trusted Entities</th>
<th>Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDS-Admin</td>
<td>arn:aws:iam::22042727352:role/RDS-Admin</td>
<td>ec2.amazonaws.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sarah Conway
Postgres Open 2015
User Management - Best Practices

- Create and use unprivileged users rather than master user
- Grant least privilege
- Make use of the policy generator
- Enable AWS CloudTrail to get logs of API calls
- Configure a strong password policy
  - Multi-Factor Authentication
  - Password expiration/rotation/reuse
- Remove unused security credentials that aren’t needed

# postgres=> create role testuser with password 'testuser' login;
# CREATE ROLE
# postgres=> grant rds_superuser to testuser;
# GRANT ROLE
# postgres=>
Password Policy

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to Managing Passwords in Using IAM.

Currently, this AWS account does not have a password policy. Specify a password policy below.

Minimum password length: [ ]

- [ ] Require at least one uppercase letter
- [ ] Require at least one lowercase letter
- [ ] Require at least one number
- [ ] Require at least one non-alphanumeric character

Allow users to change their own password: [ ]
Enable password expiration: [ ]

Password expiration period (in days): [ ]

Prevent password reuse: [ ]
Number of passwords to remember: [ ]

Password expiration requires administrator reset: [ ]

Apply password policy [ ] Delete password policy [ ]
Managing PostgreSQL Object Privileges

# postgres=> revoke all on database <database name> from public;
# REVOKE
# test=> grant connect on database test to mytestuser;
# GRANT
Parameter Groups

- Equivalent to postgresql.conf
- Change values with ALTER DATABASE, ALTER ROLE, SET
- Need to create new parameter group else default settings are used
- Use safe practices when altering database parameters

Viewing Parameter Settings

```
# select name, setting, boot_val, reset_val, unit
# from pg_settings
# order by name;
```

sslinfo

```sql
# postgres=> create extension sslinfo;
# CREATE EXTENSION
# postgres=> select ssl_is_used();
# ssl_is_used
# -------------
# t
# (1 row)
# postgres => show ssl;
# ssl
# ----
# on
# (1 row)
```
Accessing Instances with SSL

- Import certificate
- Append sslmode=require to connection string
- Reference public key using sslrootcert parameter (sslrootcert=rds-ssl-ca-cert.pem)
- Verify endpoints using sslmode=verify-full
- Note: Use "-" instead of "." in bucket names for SSL
# Password for user master:
# psql (9.3.1)
# SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
# Type "help" for help.
#
# postgres=>
AWS KMS

- Managed encryption service
- Create keys with a unique alias and description
- Define which IAM users and roles can manage keys
- Define which IAM users and roles can use keys to encrypt and decrypt data
- Choose to have AWS KMS automatically rotate keys annually
- Temporarily disable or reenable keys
- Audit use of keys through CloudTrail

VPC Security Groups

- Virtual firewall
- Permissive only rules
- Changes automatically applied
- Rules modifiable at any time
Default Groups

- Named default
- Allows inbound traffic only from other instances associated with default group
- Allows all outbound traffic from instance
- Can be modified but not deleted
VPC Best Practices

- Have proper naming conventions
  - ”AWS Region + Environment Code + OS Type + Tier + Application Code”
  - NA-D-LWB424
- Don’t make names self-explanatory! (UbuntuWebCRMProd)
- Detect security group names for information revealing names
- Enable alerts for security groups
- Take advantage of CloudTrail
- Do not create least restrictive security groups like 0.0.0.0/0
- Don’t have SSH port set to public (for EC2 instances)
- Don’t use default security groups
CloudTrail

- Provides full history of API calls
- Access control to log files using IAM
- Alerts for log files
- Compliant with internal policies and regulatory standards (PCI DSS v2.0, FedRAMP, etc)
- Important: Create IAM group for CloudTrail

pgbadger http://sourceforge.net/projects/pgbadger/
- Written in Perl
- Autodetects log file format (syslog, stderr or csvlog)
- Can parse huge log files inc. gzip
http://dalibo.github.io/pgbadger/
https://github.com/sportnginx/rds-pgbadger
- Not to be confused with the honey badger
CloudWatch

- Free
- Can set up alerts
- Subscribe to Amazon RDS events
- Detailed monitoring
- https://github.com/Netflix/security_monkey
Automated Backups

- On by default
- PITR
- Database & transaction logs stored
- Retention period can be configured up to 35 days

DB Snapshots

- User initiated backups
- Stored until manually deleted
- Create instance from snapshot at any time
- Can copy across regions
Thank You!