

RDS User Guide

- [Introduction](#)
- [Requesting a Database](#)
- [Standard Configuration](#)
- [Accounts and Passwords](#)
- [Backup](#)
- [Maintenance](#)
- [References](#)

Introduction

CDL's primary database platform is MySQL 5.7 running on Amazon's Relational Database Service (RDS). As future versions of MySQL are available, we will upgrade on a periodic basis. Incremental upgrades are performed only on request (for bug fixes or enhancements) or when Amazon drops support for an older version. Consult with IAS about other managed database options if this infrastructure does not meet your needs.

Requesting a Database

Use the database request form on IntraCDL (http://intractl.cdlib.org/groups/infrastructure_applications/database_request.php)

Standard Configuration

- A separate database instance for each service/environment
- Naming convention: `rds-program-service-env` (`rds-d2d-htmm-dev`)
- MySQL 5.7 community edition running on port 3306
- SSD storage, with provisioned IOPS as required for the workload
- The CDL default for Amazon RDS instances is Multi-AZ deployment in stage and production, single AZ in development. Multi-AZ instances assure high availability: Failover to the shadow instance typically takes less than 60 seconds. Application failover testing is advised to be sure your application can reconnect to the database after failover/DNS change.
- The default timezone for Amazon RDS is UTC. CDL sets the default timezone on each database to 'America/Los_Angeles'. RDS configurations (windows for backup or maintenance) remain UTC (and do not adjust to daylight savings time).
- AWS security groups restrict access by IP address
 - Default access for dev/stg is from CDL desktops, UCOP VPNs, and the CDL AWS environment
 - Default access for prd is from AWS production only
 - Custom security groups can be configured as needed for each RDS instance
- We use reserved instances (reserved for a period of one-year) where possible. Good performance testing is advised to understand your workload and requirements.

Accounts and Passwords

Each database instance has three standard accounts by default: a dba account with all privileges on the database, a read-write user, and a read-only user. Accounts are not host-specific (`user@%`): We rely on security groups to restrict access by source address.

Application owners are encouraged to change the default password(s) to strong password(s) unique to each user and known only to the application. Every MySQL user can change their own password: <https://dev.mysql.com/doc/refman/5.7/en/set-password.html>

Avoid entering passwords on the command line or placing them in scripts.

For command line access: `$ mysql -hHOST -uUSER -p` (and enter the password at the prompt)

For script access: Place your credentials in a `.my.cnf` file in your home directory (or the home directory of the role account) with permissions set to 400 (readable only by that user)

Example .my.cnf

Permissions on this file must be 400 (-r-----)

```
[client]
user="USER"
password="PASSWORD"
host="HOST"
database="DATABASE"
```

See *Application Credential Management* (<https://confluence.ucop.edu/display/CDLTC/Shared+Documents>) for other tips to secure credentials.

Backup

Each database instance must define a daily backup window at least 30 minutes for Amazon RDS backups. This is set when the instance is created and can be modified later. The CDL default retention period is 7 days for development, 14 days for stage, 35 days for production. This is the period of time in which a point-in-time restore may be requested. For Multi-AZ deployments, backups are done on the replica and you will see no impact on performance.

Restoring a database from the daily snapshot takes about 20 minutes, point-in-time restores vary depending on the amount of database activity.

For MySQL, RDS requires all database tables to be InnoDB for integrity of backup and restore.

IAS copies snapshots of production databases to alternate regions for disaster recovery.

If for application purposes you make use of `mysqldump`, be aware that by default it records information that you will not have permission to restore (specifically, "global transaction ids"). This was apparently added somewhere between 5.7.20-5.7.30. To achieve a dump that you will be able to successfully restore, add this flag to `mysqldump`:

```
--set-gtid-purged=OFF
```

Similarly, the db user's privileges might not include tablespaces; if you get an error "when trying to dump tablespaces" then add this to `mysqldump`:

```
--no-tablespaces
```

Maintenance

Amazon manages the RDS infrastructure and requires a weekly maintenance window at least 30 minutes for any system maintenance. For Multi-AZ deployments, the database is failed over during maintenance and the actual outage is typically short. For CDL databases, minor database upgrades are applied automatically by default.

Note: Maintenance is a periodic event and does not occur every week.

RDS is integrated with SNS to provide notifications to application owners about any database exceptions. Developers may provide email address(es) to be notified (or any endpoint supported by SNS).

Most of CDL's RDS instances are marked to receive automatic updates when new minor versions of database software are released. In addition, rare updates to the underlying operating system or changes to the underlying hardware are required. These changes are performed automatically during the specified maintenance window if possible.

To determine the maintenance windows for the RDS instances of a given program, go to your program's "aws2-ops" EC2 instance (e.g. d2d-aws2-ops, uc3-aws2-ops, pub-aws2-ops, etc.) and run the "`list-rds-maint-windows.sh`" script, specifying your program's 3-letter code. This script assumes you have the IAM permissions necessary to list and describe RDS instances and maintenance windows in your AWS Account (all the "aws2-ops" EC2 instances have these permissions).

If you provide a program name from the following list as an argument, this script will list only the instances tagged as belonging to that program, along with their maintenance windows and date/time (all converted to Pacific) after which scheduled maintenance will be automatically applied (in the first maintenance window after this date/time). For example:

```

d2d-aws2-ops,~,d2daws>whoami
d2daws
d2d-aws2-ops,~,d2daws>/apps/local/bin/list-rds-maint-windows.sh d2d
DB Instance | Maintenance Window (Pacific Time) | Apply Date (Pacific Time)
rds-d2d-htmm-stg | Mon:21:30-Mon:22:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-htmm-prd | Tue:21:30-Tue:22:00 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-pid-prd | Tue:17:00-Tue:17:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-papr-prd | Wed:00:00-Wed:01:00 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-htmm-dev | Sun:21:30-Sun:22:00 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-pid-stg | Mon:17:00-Mon:17:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-papr-dev | Sat:00:00-Sat:01:00 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-jreport-stg | Fri:23:00-Fri:23:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-jreport-prd | Sat:23:00-Sat:23:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-pid-dev | Sun:17:00-Sun:17:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-ipservice-stg | Mon:00:00-Mon:00:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-papr-stg | Tue:00:00-Tue:01:00 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-ipservice-prd | Tue:00:00-Tue:00:30 | Sun Jan 30 16:00:00 PST 2022
rds-d2d-ipservice-dev | Mon:00:00-Mon:00:30 | Sun Jan 30 16:00:00 PST 2022
d2d-aws2-ops,~,d2daws>

```

The first line of this output tells you that patching will commence on the instance 'rds-d2d-htmm-stg' on the first Monday after 4PM January 30, 2022, between 9:30 PM and 10:30 PM Pacific Time. No guarantees are made about when the patching will finish.

If you provide no arguments to the command, it will loop through the list of programs and list the information for all programs' RDS instances in the main account (see list below). There are about 60 instances or so at the time of this writing, and unless you're IAS, this is probably a lot more information than you need or want, so I suggest you use the example above for your own program tag.

Programs:

- d2d
- dsc
- ias
- ids
- mdg
- pub
- uc3
- web

If you provide an argument or some combination of arguments that is anything other than one and only one of the programs in this list, the script will probably give you an error, and may very well fail in spectacular and unexpected ways.

References

- RDS Documentation: <https://aws.amazon.com/rds/mysql/>
- MySQL Documentation: <https://dev.mysql.com/doc/refman/5.7/en/>