RDS for Oracle Hands-On Lab

Dallas Willett, Senior Database Engineer

Agenda

- **Create an Instance**
- Modify an Instance
- Upgrade an Instance
- Backup and Restore
- RDS PL/SQL Packages
- Best Practices

Create an RDS for Oracle Instance

- Login to the AWS Console
  - https://console.aws.amazon.com/
- Pick a region - US West 2 (Oregon)
- Search for RDS
- Launch an Oracle Enterprise Edition (EE) instance
  - Click: Create Database
  - Select: Oracle + Oracle Enterprise Edition
  - Click: Next
  - Select: Dev/Test
  - Click: Next

Create an RDS for Oracle Instance

- Specify DB Details
  - DB Engine Version: 12.1.0.2.v14 (not latest version so we can upgrade)
  - Instance Class: r5.xlarge
  - Multi-AZ deployment: No
  - Estimated Monthly Cost: ~$342
  - Estimated Hourly Costs: ~$0.48
- Settings - Customize RDS instance identifier and RDS ”master” user
  - DB Instance Identifier: <user defined> e.g. my-oracle-12c-db
  - Master Username: <user defined> e.g. admin
  - Master Password: <user defined>
- Click: Next

Create an RDS for Oracle Instance

- Network and Security
  - Public accessibility: Yes
- Database Options
  - Optionally choose “Database name” (ORACLE_SID) and Port
  - Create will be slightly quicker with default Database name (no rename involved)
- Backups
  - Disable backups by changing “Backup Retention Period”: 0 days
- Create Database
  - Click: Create Database
  - Click: View DB instance details
  - “DB instance status” will say “creating” and eventually switch to “Available”

Create an RDS for Oracle Instance

- AWS CLI
  - https://docs.aws.amazon.com/cli/latest/userguide/installing.html
- Example Command:

```
aws rds create-db-instance \
  --db-instance-identifier $USER-ee-test-12102v14 \
  --db-name ORCL \
  --allocated-storage 20 \
  --storage-type gp2 \
  --db-instance-class db.r5.xlarge \
  --engine oracle-ee \
  --port 1521 \
  --backup-retention-period 0 \
  --license-model=byol \
  --master-user-password fjie87bna09bfe3 \
  --master-username admin \
  --engine-version 12.1.0.2.v14
```

AWS Global Infrastructure (2019)

AWS Global Infrastructure (2017)

SQL Developer - Check Security Group Rule

Determine your IP address

- https://checkip.amazonaws.com
- NOTE: may not give correct IP if behind a firewall

Find your instance on the AWS Console

- Click on Link: VPC Security Groups (e.g. rds-launch-wizard-... takes you to EC2 service)
- Select Tab: Inbound
- Type: Oracle-RDS
- Protocol: TCP
- Port Range: 1521
- Source: My IP (e.g. 123.45.67.89) - should have automatically been added - add if necessary
- Avoid 0.0.0.0/0 (open to the world)

SQL Developer – Test Connection

amazon web services

Find your instance on the AWS Console

- Copy: Endpoint

Start SQL Developer

- New Connection
- Paste: Endpoint -> Hostname
- Port: 1521
- SID: ORCL [or your ORACLE_SID if you changed it when creating database]
- Username: [your master username]
- Password: [your master password]

- Test Connection in SQL Developer

SQL Developer – Example Queries

```
select instance_name, host_name, version, startup_time, archiver
from v$instance;

INSTANCE_NAME    HOST_NAME          VERSION    STARTUP_TIME ARCHIVER
---------------- ------------------ ---------- ------------ ----------
ORCL             ip-123-45-67-89    12.1.0.2.0 13-FEB-19    STOPPED
```

```
show parameter cpu_count

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------
cpu_count                            integer     4
```

Agenda

- Create an Instance
- **Modify an Instance**
  - **Scale Compute**
  - **Scale Storage**
  - **Convert to HA (Multi-AZ)**
  - **Parameter Groups**
  - **Option Groups**
- Upgrade an Instance
- Backup and Restore
- RDS PL/SQL Packages
- Best Practices

Scale Compute

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Click: Modify
- Change DB Instance Class: m5.large
  - Estimated Monthly Cost: ~$125
  - Estimated Hourly Costs: ~$0.17
- Click: Continue
- Select: Apply Immediately
- Click: Modify DB Instance

- “Info” will change from “Available” to “Modifying” and back to “Available”

Scale Compute – Instance Types

R5 – Memory Optimized

- 1 vCPU : 8 GiB memory
- starting at ~$170 / month

Model	vCPU	Mem (GiB)	Storage (GiB)	Dedicated EBS Bandwidth (Mbps)	Networking Performance (Gbps)
r5.large	2	16	EBS-Only	up to 3,500	Up to 10
r5.xlarge	4	32	EBS-Only	up to 3,500	Up to 10
r5.2xlarge	8	64	EBS-Only	up to 3,500	Up to 10
r5.4xlarge	16	128	EBS-Only	3,500	Up to 10
r5.12xlarge	48	384	EBS-Only	7,000	10
r5.24xlarge	96	768	EBS-Only	14,000	25

M5 - General Purpose

- 1 vCPU : 4 GiB memory
- starting at ~$125 / month

Model	vCPU*	Mem (GiB)	Storage (GiB)	Dedicated EBS Bandwidth (Mbps)	Network Performance (Gbps)
m5.large	2	8	EBS-only	Up to 3,500	Up to 10
m5.xlarge	4	16	EBS-only	Up to 3,500	Up to 10
m5.2xlarge	8	32	EBS-only	Up to 3,500	Up to 10
m5.4xlarge	16	64	EBS-only	3,500	Up to 10
m5.12xlarge	48	192	EBS-only	7,000	10
m5.24xlarge	96	384	EBS-only	14,000	25

T3 – Burstable

- 1 vCPU : 0.5 - 4 GiB memory
- Free for 1 year
- starting at $15 / month

Scale Storage

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Click: Modify
- Allocated Storage: 40
- Click: Continue
- Select: Apply Immediately
- Click: Modify DB Instance

- “Info” will change from “Available” to “Modifying” to “Storage-optimization” and eventually back to “Available”

- NOTE: Max size today is 32 TB and cannot scale back down.

Scale Storage

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Select: Configuration tab
- Under section: Backup
- “Storage” should now say “40 GiB”

Monitoring Storage

- Select: Monitoring tab
- Should see a graph for “Free Storage Space” with a jump from 20 GiB to 40 GiB.
- Can see more data in “Enhanced Monitoring” graphs

Recommendation - setup automated alerts in CloudWatch to email/page before you run out of space

SQL Developer – Example Queries

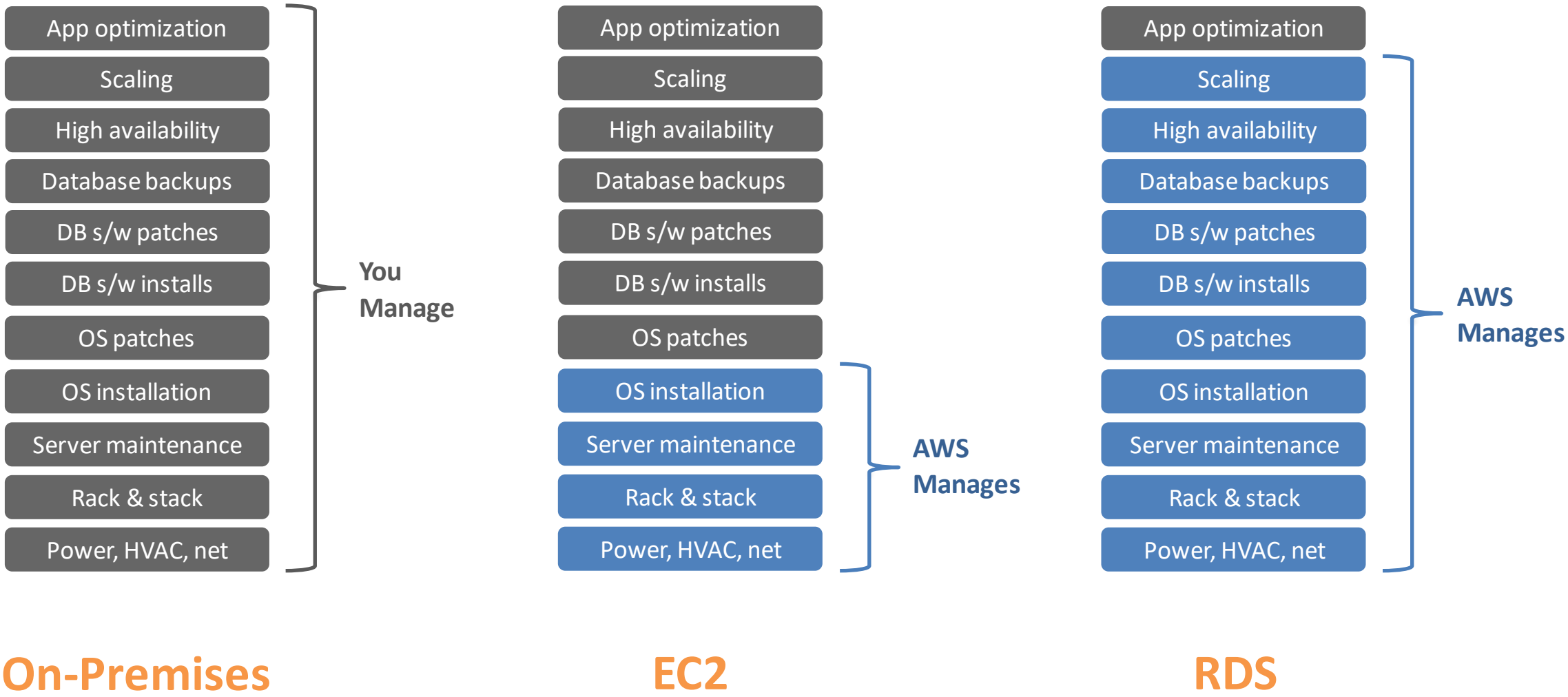
```
select instance_name, host_name, version, startup_time, archiver
from v$instance;

INSTANCE_NAME    HOST_NAME          VERSION    STARTUP_TIME ARCHIVER
---------------- ------------------- ---------- ------------ ----------
ORCL             ip-54-43-32-21     12.1.0.2.0 13-FEB-19    STOPPED
```

```
show parameter cpu_count

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------
cpu_count                            integer     2
```

Enterprise Apps on AWS – RDS Advantages



Convert to Multi-AZ

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Click: Modify
- Select Multi-AZ Deployment: Yes
- Click: Continue
- Select: Apply Immediately
- Click: Modify DB Instance

- “Info” will change from “Available” to “Modifying” and back to “Available”

Convert to Multi-AZ

In the AWS Console

- RDS -> Databases
- Select instance
- Select: Configuration tab
- Under section: Instance
- “Multi-AZ” should now say “Yes”

Recommendation – test a failover with your application

- Select: Actions -> Reboot
- Select: Reboot with Failover

RDS Multi-AZ

- Primary in one AWS Availability Zone (AZ)
- Secondary in a different AZ
- Synchronous replication between Primary and Secondary
- AWS monitors database and will automatically fail-over
- Zero data loss

- Used for all RDS Linux based engines
- Multi-AZ can be added or removed at any time (online – no outage)
- Can be used with Oracle Standard Editions

Create 12.1 Parameter Group

In the AWS Console

- RDS -> Parameter Groups -> Create Parameter Group
- Parameter Group Family: oracle-ee-12.1
- Group Name: test-oracle-ee-12-1-pg
- Description: test
- Click: Create

Select Parameter Group you just created

- Click: Edit Parameters
- job_queue_processes: 99
- Click: Save changes

Create 12.1 Option Group

amazon web services

In the AWS Console

- RDS -> OptionGroups -> Create Option Group
- Name: test-oracle-ee-12-1-og
- Description: test
- Engine: oracle-ee
- Major Engine Version: 12.1
- Click: Create

Select Option Group you just created

- Add Option
- Option: Timezone
- Time Zone: US/Pacific
- Apply Immediately: Yes
- Click: Add Option

SQL Developer – Example Queries

```
select to_char(sysdate,'YYYY-MM-DD HH24:MI:SS') from dual;

TO_CHAR(SYSDATE,'YY
-------------------
2019-02-13 19:00:06
```

```
show parameter job_queue_processes

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------
job_queue_processes                  integer     50
```

Associate Parameter Group to Instance

In the AWS Console

- RDS -> Databases
- Select instance
- Instance Actions -> Modify
- DB Parameter Group: test-pg-oracle-ee-11-2
- Check: Apply Immediately
- Continue
- Modify DB Instance

Associate Option Group to Instance

In the AWS Console

- RDS -> Databases
- Select instance
- Instance Actions -> Modify
- DB Option Group: test-og-oracle-ee-11-2
- Check: Apply Immediately
- Continue
- Modify DB Instance

Reboot

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Reboot
- If it's a Multi-AZ instance, you have the option to "Reboot With Failover"
- Click: Reboot

SQL Developer – Example Queries

```
select to_char(sysdate,'YYYY-MM-DD HH24:MI:SS') from dual;

TO_CHAR(SYSDATE,'YY
--------------------
2019-02-13 11:02:23
```

```
show parameter job_queue_processes

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
job_queue_processes                  integer     99
```

Agenda

- Create an Instance
- Modify an Instance
- **Upgrade an Instance**
  - **Minor Version Upgrade**
  - **Major Version Upgrade**
- Backup and Restore
- RDS PL/SQL Packages
- Best Practices

Database Upgrade – Minor Version

In the AWS Console

- RDS -> Databases
- Select instance
- Instance Actions -> Modify
- DB Engine Version: 12.1.0.2.v15
- Check: Apply Immediately
- Continue
- Modify DB Instance

- “Info” will change from “Available” to “Upgrading” and back to “Available”

Create 12.2 Parameter Group

In the AWS Console

- RDS -> Parameter Groups -> Create Parameter Group
- Parameter Group Family: oracle-ee-12.2
- Group Name: test-oracle-ee-12-2-pg
- Description: test
- Click: Create

Select Parameter Group you just created

- Click: Edit Parameters
- job_queue_processes: 88
- Click: Save changes

Create 12.2 Option Group

In the AWS Console

- RDS -> OptionGroups -> Create Parameter Group
- Name: test-og-oracle-ee-12-2
- Description: test
- Engine: oracle-ee
- Major Engine Version: 12.2
- Click: Create

Select Option Group you just created

- Add Option
- Option: Timezone
- Time Zone: US/Pacific
- Apply Immediately: Yes
- Click: Add Option

Database Upgrade – Major Version

In the AWS Console

- RDS -> Databases
- Select instance
- Instance Actions -> Modify
- DB Engine Version: 12.2.0.1.ru-2019-01.rur-2019-01.r1
- DB Parameter Group: test-oracle-ee-12-2-pg
- DB Option Group: test-oracle-ee-12-2-og
- Check: Apply Immediately
- Continue
- Modify DB Instance

- “Info” will change from “Available” to “Upgrading” to “Modifying” and eventually back to “Available”

SQL Developer – Example Queries

```
select comp_id, version, status from dba_registry order by comp_id;

COMP_ID                          VERSION                STATUS
-------------------------------- ---------------------- ------------------
CATALOG                          12.2.0.1.0             VALID
CATPROC                          12.2.0.1.0             VALID
CONTEXT                          12.2.0.1.0             VALID
XDB                              12.2.0.1.0             VALID
```

```
show parameter job_queue_processes

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------
job_queue_processes                  integer     88
```

Agenda

- Create an Instance
- Modify an Instance
- Upgrade an Instance
- **Backup and Restore**
  - **Automated Backups**
  - **Manual Backups**
  - **Snapshot Restore**
  - **Point In Time Restore (PITR)**
- RDS PL/SQL Packages
- Best Practices

Database Backups – Automated Backups

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Modify
- Backup Retention Period: [select between 1-35 days]
- Check: Apply Immediately
- Continue
- Modify DB Instance

- “Info” will change from “Available” to “Modifying” to “Backing Up” and back to “Available”

Database Backups – Automated Backups

In the AWS Console

- RDS -> Databases
- Select instance
- Select: Maintenance & backups tab
- Under section: Backup
- “Automated backups” should now say “Enabled (X days)”

Enabling Backups allow for two types of restores:

- 1) Snapshot restores - should see one automated snapshot:
  E.g. rds:my-oracle-12c-database-2019-02-13-11-34
- 2) Point-in time restore (PITR) – check “Latest restore time” value. E.g. February 13th, 2019, ...

Can also see backups/snapshots in:

- RDS -> Snapshots

SQL Developer – Example Queries

```
select instance_name, host_name, version, startup_time, archiver
from v$instance;

INSTANCE_NAME    HOST_NAME          VERSION    STARTUP_TIME ARCHIVER
---------------- ------------------ ---------- ------------ ----------
ORCL             ip-54-43-32-21     12.1.0.2.0 13-FEB-19    STARTED
```

```
show parameter archive_lag_target

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------
archive_lag_target                   integer     300
```

Database Backups – Manual Backups

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Take snapshot
- Snapshot Name: [type in a name] e.g. “my-oracle-12c-database-snapshot”
- Take Snapshot

Should take you to the “Snapshots” section of the AWS console

- “Status” of snapshot will change from “creating” to “available”

Database Restores – Snapshot Restore

In the AWS Console

- RDS -> Snapshots
- Select desired snapshot (either manual or automated)
- Actions -> Restore Snapshot
- Under “Settings” section
- Enter new “DB Instance Identifier” - e.g. “my-oracle-12c-database-restore”
- Change anything else that you want for the new instance
- Click: Restore DB Instance

Should take you back to the “Databases” section of the AWS console

- “Status” will change from “Creating” to “Backing-up” to “Available”

NOTE: Snapshot restores or PITR will not affect performance of running instance.

Database Restores – Point in Time (PITR)

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Restore to point in time
- Choose: Latest restorable time or choose a Custom time
- Under “Settings” section
- Enter new “DB Instance Identifier” - e.g. “my-oracle-12c-database-pitr”
- Click: Launch DB Instance

Should take you back to the “Databases” section of the AWS console

- “Info” will change from “Creating” to “Backing-up” to “Available”

Recommendation – create a copy of production and do a test-run of any changes you plan to make (database upgrades, parameter changes, option group changes, etc...)

Agenda

- Create an Instance
- Modify an Instance
- Upgrade an Instance
- Backup and Restore
- **RDS PL/SQL Packages**
- Best Practices

RDS PL/SQL Packages – Kill Session

amazon web services

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.html

```
select sid, serial# from v$session
 where sid = (select distinct sid from v$mystat);
```

In another (or same) SQL Developer Session:

```
exec rdsadmin.rdsadmin_util.kill(sid => [sid], serial => [serial#] );
```

RDS PL/SQL Packages – Granting SYS Privileges

amazon web services

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.html

```
create user USER1 identified by [password];

grant CREATE SESSION, RESOURCE to USER1;
grant DBA to USER1;
grant SELECT_CATALOG_ROLE to USER1;
```

```
begin
rdsadmin.rdsadmin_util.grant_sys_object(
  p_obj_name => 'V_$SESSION',
  p_grantee => 'USER1',
  p_privilege => 'SELECT');
end;
/
```

RDS PL/SQL Packages – Password Verify Function

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.html

```
begin
rdsadmin.rdsadmin_password_verify.create_verify_function(
  p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
  p_min_length => 12,
  p_min_uppercase => 2,
  p_min_digits => 1,
  p_min_special => 1,
  p_disallow_at_sign => true);
end;
/
```

RDS PL/SQL Packages – Online Log Files

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.Log.html

```
select group#, bytes/1024/1024 as mb, status from v$log;

-- switch logfile
exec rdsadmin.rdsadmin_util.switch_logfile;

-- add logfile
exec rdsadmin.rdsadmin_util.add_logfile(p_size => '256M');
```

RDS PL/SQL Packages – Archive Log Retention

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.Log.html

```
select name, force_logging from v$database;

-- enable force logging
exec rdsadmin.rdsadmin_util.force_logging;

-- retain archive redo logs for 24 hours on host
begin rdsadmin.rdsadmin_util.set_configuration(
  name => 'archivelog retention hours',
  value => '24');
end;
/
```

RDS PL/SQL Packages – Directories

Login via SQL Developer

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.Misc.html

```
exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'product_descriptions');

select DIRECTORY_PATH from DBA_DIRECTORIES where DIRECTORY_NAME='product_descriptions';

select * from table (rdsadmin.rds_file_util.listdir(p_directory => 'product_descriptions'));

-- read a trace file
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime desc;
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_ORCL.log'));

select message_text from ALERTLOG order by INDX;
```

View / Download Logs in AWS Console

amazon web services

In the AWS Console

- RDS -> Databases
- Select instance
- Select Logs and Events tab
- Scroll down to “Logs” and Filter: alert
- Select: [view | watch | download]

Viewing Enhanced Monitoring

In the AWS Console

- RDS -> Databases
- Select instance and then the “Monitoring” tab
- In the “Monitoring” drop-down, select “Enhanced Monitoring”
- Select “Manage graphs” to choose other graphs to display

Viewing Performance Insights

amazon web services

In the AWS Console

- RDS -> Performance Insights
- Select instance

- Generate some load. E.g.:

```
select count(1) from dba_objects a, dba_objects b;
```

Configuring Enhanced Monitoring

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Modify
- Select Enable or Disable Enhanced Monitoring
- Optionally change Granularity: [1-60 seconds]
- Click: Continue
- Select: Apply Immediately
- Click: Modify DB Instance

Configuring Performance Insights

In the AWS Console

- RDS -> Databases
- Select instance
- Actions -> Modify
- Select Enable or Disable Performance Insights
- Optionally change Retention: [7 days or 2 years]
- Click: Continue
- Select: Apply Immediately
- Click: Modify DB Instance

Delete instances you are not using

amazon web services

Find your instance on the AWS Console

- Select instance
- Actions -> Delete
- Unselect: Create final snapshot
- Unselect: Retain automated backups (new)
- Select: I acknowledge ...
- Confirm deletion. Type in: delete me
- Click: Delete

- “Status” will change from “Available” to “Deleting” before being removed from the AWS console.

Agenda

- Create an Instance
- Modify an Instance
- Upgrade an Instance
- Backup and Restore
- RDS PL/SQL Packages
- **Best Practices**

Test your changes

It's easy and cheap to create a temporary copy of your databases

Test any change first before doing production:

- Engine Version Upgrades
- Parameter Changes
- Option Group Changes
- Scale Compute

Right-size your RDS Instances

- Use Cloud Watch Metrics
- Scale up before peak days
- Scale back down after

Security

amazon web services

- Don’t open to the world – Avoid CIDR rules with “0.0.0.0/0”
- Use IAM Groups to control access
- Encryption at rest – AWS KMS or Oracle TDE
- Encryption in transit – SSL / TLS

Migrations

- Small databases, or databases with generous downtime windows – use Data Pump with DBA Directories
- Larger databases, or near zero downtime requirements – use AWS DMS, Oracle Goldengate, Quest SharePlex, or similar logical replication

Know your usage and check your bill

- AWS Billing – check this to make sure you're not surprised by your bill
- AWS Simple Monthly Calculator
  - https://calculator.s3.amazonaws.com/

Agenda

- Create an Instance
- Modify an Instance
- Upgrade an Instance
- Backup and Restore
- RDS PL/SQL Packages
- Best Practices

RDS for Oracle Hands-On Lab

Questions?