ORACLE
DatabaseWorld
at CloudWorld

Application Continuity
with Oracle Database 23ai

Markus Michalewicz (Oracle)
Sebastian Solbach (Oracle)
Harsh Gupta (Deutsche Bank)

September 11, 2024

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

The materials in this presentation pertain to Oracle Health, Oracle, Oracle Cerner, and Cerner Enviza which are all wholly owned subsidiaries of Oracle Corporation. Nothing in this presentation should be taken as indicating that any decisions regarding the integration of any EMEA Cerner and/or Enviza entities have been made where an integration has not already occurred.

Click to edit Title

Markus Michalewicz

Vice President
Mission Critical Database
Product Management
Oracle

Sebastian Solbach

Distinguished Product Manager
Mission Critical Database
Product Management
Oracle

Harsh Gupta

Deputy Head Global Database Services
Head Of Oracle Cloud Engineering
Deutsche Bank

The Problem:
Applications are Impacted by Database Outages

- Waits
- Receive an Error
- Need to reconnect
- Need to restart
- Unsure where left off

The Goal:
Business Continuity

Eliminate downtime for users

Masking of recoverable errors

Maintenance is hidden

Automatically reconnect

In-flight work is preserved

The Solution:
Transparent Application Continuity 23ai

Better performance

Simpler for developers

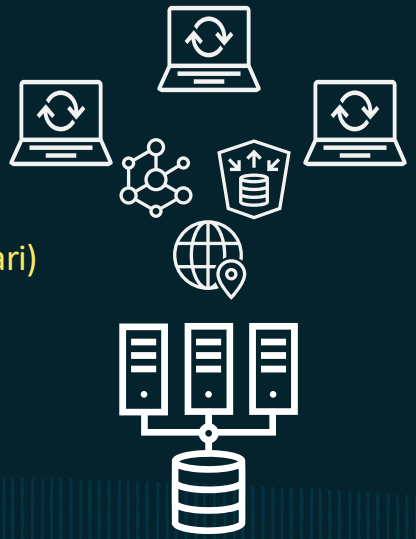
Detailed statistics

Built on Oracle Maximum Availability Architecture

Application Protection – One Solution – Two Flavors

Application Continuity (AC)

- For planned maintenance and unplanned outages
- Available with Oracle RAC and Active Data Guard
- Oracle and 3rd party connection pools that are JDK-compliant (incl. JBoss, Hikari)
- Supports customizable "side effects" (e.g. UTL_HTTP)



Transparent Application Continuity (TAC)

- For planned maintenance and unplanned outages
- Available with Oracle RAC and Active Data Guard
- Same as AC and discovered TAC boundaries
- Disables "side effects", customizable (23ai)
- Default on Oracle Autonomous Database

Application Continuity 23ai Performance

New In 23ai

>40%

Faster failover for selects with Transparent Application Continuity repositioning cursors at failover time.

Up to 50%

Lower database CPU cost compared to 19c running SPEC-J using a reduced Application Continuity code path.

Up to 55%

Lower client CPU cost with OCI driver using OCI_THREADED_V2.

Even faster with Native Transaction Guard*

Up to 95%

Lower database CPU cost

using native XID when possible

*Transaction Guard guarantees the commit outcome when an error code is returned after an error or outage. It is the basis for Application Continuity and Transparent Application Continuity.

Simpler for Developers

Allow developers to focus on functionality

Database-initiated session migration during draining

New In 23ai

Let the database determine when to relocate sessions during planned maintenance and failback

Draining

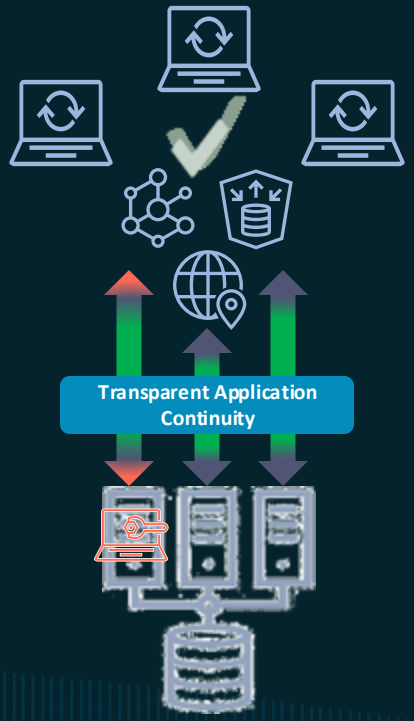
- Allows applications to complete their work before performing maintenance or any form of draining. With Oracle Database 23ai, the drain timeout can be set per database service or session.

Oracle Database

- Detects sessions that will not drain and can failover
- Proactively chooses failover sessions based on rules
  - **Bounds maintenance windows**
  - **Bounds moving sessions**
- Most requests reach failover conditions quickly

Selected sessions continue with **(Transparent) Application Continuity**

- Minimizes ungraceful termination of sessions unable to relocate before timeout
- Reduces replay time on failover



RESET_STATE for all

New In 23ai

Prevents applications from leaking state – the most essential developer feature in 23ai

Problem

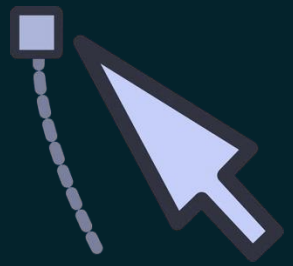
- When an application returns a connection to the pool, cursors in FETCH and session state remains on that session, unless an action is taken to clear them. Application session state and cursors leak to later reuses.

With Oracle Database 23ai

- RESET_STATE restores the session state to login and cancels cursors in FETCH at the end of each request
- Oracle Database guarantees cleaning the state which is otherwise left to developers
- Independent of (Transparent) Application Continuity

Benefits

- Application security holes are prevented, code savings, lower DB CPU usage
- Broader protection with TAC – e.g. Microservices, APEX, ORDs, Fusion Apps, SwingBench
- Highly demanded customer feature – e.g. requested by banks, retail, government



Resumable Cursors

New In 23ai

Reduces TAC discovery of request boundaries to “not in transaction and snapshot-able session state”

- On capture:
  - Select statements are held across transactions
- On replay:
  - Execution state restored (if not the same)

- Eliminates the need to close cursors in fetch
- >40% faster at failover by repositioning cursors
- Vastly increases TAC coverage

```
SQL> SELECT order_id FROM orders
WHERE order_date > sysdate – 7
order by order_id;

  while (!end_of_fetch)
  {
        FETCH order_id INTO :order_num;
        < Perform transactional work for
        an order>
  commit;
  }
```


Ensure Replay uses the same session state

New In 23ai

Failover_Restore with Database Templates

- Database Template = set of session states maintained by the database
- Eliminates the need for custom callbacks
- Session state is automatically assigned for
  - `FAILOVER_RESTORE = LEVEL2 / AUTO`
  - Same network security as advised for database links

```
$ sqlplus / as syskm

SQL> ALTER DATABASE DICTIONARY ENCRYPT
CREDENTIALS CONTAINER = ALL;

SQL> select * from
DICTIONARY_CREDENTIALS_ENCRYPT;

ENFORCEM
--------
ENABLED
```


Customizable Side Effects, Database Links, and More

New In 23ai

- Database requests such as external calls can cause side effects
  - Not under transaction control of the database; E.g. UTL_HTTP, UTL_FILE, UTL_TCP
- Customizing allows for choosing
  - side-effects, database links and Autonomous Transactions to Replay
- Set on a per-service level or use the PL/SQL API for finer control
  - AC default is allow “replay all”; TAC disallows “replay all” by default
- Example: make database links replayable for service ‘myservice’

```
SQL> execute dbms_app_cont_admin.set_replay_rules
(service_name => ‘myservice’, replayable => true,
dbms_app_cont_admin.database_links);
```


Consolidated Data Source

New In 23ai

One package for everything – set once and for all

- Included in ojdbcXX.jar
- Only one Data Source needs to be included on client-side
  - Developers no longer need to switch Data Source to enable Application Continuity
  - Client automatically enables Application Continuity based on server side service attribute setting

```
oracle.ucp.jdbc.PoolDataSource pds =

oracle.ucp.jdbc.PoolDataSourceFactory.getPoolDataSource();
    pds.setConnectionFactoryClassName(
      "oracle.jdbc.datasource.impl.OracleDataSource");
```

https://docs.oracle.com/en/database//oracle/oracle-database/23/jajdb/oracle/jdbc/datasource/impl/package-summary.html

Enable Application Continuity

New In 23ai

AC/TAC self service in private clouds and rapid deployments

- Application Continuity can be enabled/disabled per-service with srvctl
  - For any subsequent connection to the service
    - Enable PDB-admins to create an AC/TAC service and benefit from RESET_STATE, set_draining, etc.
- Starting with Oracle Database 23ai DBMS_APP_CONT_ADMIN can be used

```
dbms_app_cont_admin.enable_ac('<svc>', 'LEVEL1', 600);
dbms_app_cont_admin.enable_tac('<svc>', 'AUTO', 600, 'AUTO');
```


Proven in many configurations

Springboot & Oracle Universal Connection Pool (UCP)

https://blogs.oracle.com/developers/post/hikaricp-best-practices-for-oracle-database-and-spring-boot

Tomcat & Oracle Universal Connection Pool (UCP)

https://www.oracle.com/docs/tech/database/planned-unplanned-rlb-ucp-tomcat.pdf

JBoss EAP & native Request Boundaries

Configuration in downloadable hidden slides

AC/TAC Planned Downtime

Jboss EAP: Management Console or CLI to update Datasource definition

```
<datasource jndi-name="java:jboss/datasources/ExampleDS"
    pool-name="ExampleDS” enabled="true" use-java-context="true">
<driver>oracle</driver>
<datasource-class>oracle.jdbc.datasource.impl.OracleDataSource</datasource-class>
<connection-property name="URL"> jdbc:oracle:thin:@......... </connection-property>
<pool>.........</pool>
<security>
    <user-name>dbuser</user-name><password>dbpasswd</password>
</security>
<validation>
    <validate-on-match>true</validate-on-match>
    <background-validation>false</background-validation>
        .........</validation>
</datasource>
<drivers>
    <driver name="oracle" module="com.oracle.jdbc"></driver>
</drivers>
```


AC/TAC Planned Downtime

Jboss EAP: Management Console or CLI to update Datasource definition

```
<connection-property name="URL">
  jdbc:oracle:thin:@CURRENT_URL?TNS_ADMIN=/absolute_path/ojdbc.properties
</connection-property>

In /absolute_path/ojdbc.properties, include the connection properties and
values:
    prop1=value1
    prop2=value2
    ...
```


Proven in Many Configurations

Programmatically with Python

- Call init_oracle_client() to enable Thick Mode

```
import oracledb
oracledb.init_oracle_client(lib_dir=="/absolute_path/instantclient_23_4")
```

- Use pooled connection (with DRCP)
  - https://github.com/oracle/python-oracledb/blob/main/samples/connection_pool.py
  - www.oracle.com/docs/tech/drcp-technical-brief.pdf

```
pool = oracledb.create_pool(user="cj", password=pw, dsn=cs,
       min=pool_min, max=pool_max, increment=pool_inc,session_callback=init_session)
```


Detailed Statistics

See what’s going on at any step of the way

AWR report for Application Continuity

Pre-analysis if Application Continuity is the right solution for your application

- Connection Pool correctly configured?
  - Begin/End Request
- Application Continuity enabled?
  - Protected calls = 0
  - Small number = Low protection
  - High number = Good Protection
    Run ACCHK for further analysis

Instance Activity Stats			
cumulative DB time in requests	577,879,345	1,554,086.64	12,505.77
cumulative DB time protected in requests	491,764,077	1,322,497.48	10,642.17
cumulative begin requests	415,508	1,117.42	8.99
cumulative end requests	415,533	1,117.49	8.99
cumulative time in requests	505,808,371	1,360,266.70	10,946.10
cumulative user calls in requests	518,067	1,393.23	11.21
cumulative user calls protected by Application Continuity	518,067	1,393.23	11.21

Determine the protection provided with ACCHK

Instance & session statistics

More advice:

- Replay statistics
- PLSQL session state
- Side effects
- External and CDB-level reports
- Report by Time Interval
- PDBADMIN role
- Backported to19RU19
- canned sql using history

New In 23ai

Table based

```
EXTERNAL CON_ID  : 0
CON_UID : 1
-----------------------------------------------------------------------------------------------------------------------------------
                          Fail Protected Protected                        Avg Avg Protected            Avg   Avg Protected            Max   Max Protected
                   Service  over   calls %    time %     Requests calls/request calls/request time/request ms time/request ms   calls/request   calls/request
------------------------- ----- --------- --------- ------------ ------------- ------------- --------------- --------------- --------------- ---------------
acdyn1.cdbtest.regress.rd TRANS      52.3      83.5           26         5.077         2.654           5.164           4.314              83              21
sqlplus_ac1.cdbtest.regre TRANS      65.3      61.5            1            75            49          15.377           9.461              75              49
sqlplus_tac1.cdbtest.regr  AUTO       100      99.6            8         9.375         9.375           3.777           3.763              43              43
tac_1.cdbtest.regress.rdb  AUTO      99.2      99.8           28         4.714         4.679           4.363           4.356              41              41
```

• • •

```
EXTERNAL CON_ID  : 0
CON_UID : 1
------------------------------------------------------ Event details per service ------------------------------------------------------

Service name  : acdyn1.cdbtest.regress.rdbms.dev.us.oracle.com
Failover type : TRANSACTION
-----------------------------------------------------------------------------------------------------------------------------------
          Event Type Error Code             Program              Module              Action        SQL_ID            Call      Total
---------------- ---------- -------------------- -------------------- -------------------- ------------- --------------- ----------
             DISABLE      41409     JDBC Thin Client     JDBC Thin Client                                       COMMIT          1
      NOT_REENABLING      41459     JDBC Thin Client     JDBC Thin Client                                       COMMIT          1
       NEVER_ENABLED      41462     JDBC Thin Client                                                                            1

Service name  : sqlplus_ac1.cdbtest.regress.rdbms.dev.us.oracle.com
Failover type : TRANSACTION
-----------------------------------------------------------------------------------------------------------------------------------
          Event Type Error Code             Program              Module              Action        SQL_ID            Call      Total
---------------- ---------- -------------------- -------------------- -------------------- ------------- --------------- ----------
             DISABLE      41409 sqlplus@slc15dnd (TN            SQL*Plus                                       COMMIT          1
      NOT_REENABLING      41459 sqlplus@slc15dnd (TN            SQL*Plus                                       COMMIT          1
```

Granular filtering of ACCHK

New In 23ai

Gather only relevant data for a focused analysis

- DBMS_APP_CONT_ADMIN filters data to be collected based on
  - Service
  - Module
  - Program
  - (Source) Machine name
  - ACCHK_SET_FILTER, ACCHK_SHOW_FILTERS, ACCHK_CLEAR_FILTER

```
SQL> EXEC dbms_app_cont_admin.acchk_set_filter
(DBMS_APP_CONT_ADMIN.SERVICE_FILTER, 'ORACLE.Service1');

SQL> dbms_app_cont_admin.acchk_set(true,300);
```


Built on
Oracle Maximum Availability Architecture

Documented Application High Availability Levels

https://docs.oracle.com/en/database/oracle/oracle-database/23/haovw/continuous-availability-applications.html

30 Configuring Continuous Availability for Applications

Ensure that your applications are configured to quickly and automatically shift workload to available Oracle RAC instances or standby databases during planned maintenance and unplanned outages.

Application up time is maximized by following these recommendations when there are outages.

The primary audience for this document is application developers and application owners. Operational examples are included for database administrators and PDB administrators.

Topics:

- About Application High Availability Levels
- Configuring Level 1: Basic Application High Availability
- Configuring Level 2: Prepare Applications for Planned Maintenance
- Configuring Level 3: Mask Unplanned and Planned Failovers from Applications
- Reference

Application Continuity DBMS_ROLLING Support

New In 23ai

Minimizes application impact throughout the entire database upgrade process

(Transparent) Application Continuity

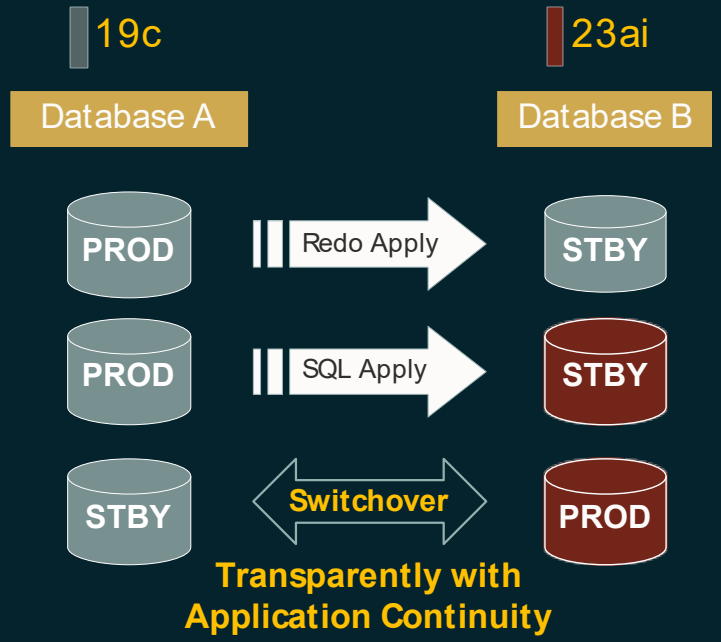
- Hides database downtime from your users

DBMS_ROLLING (available with Active Data Guard)

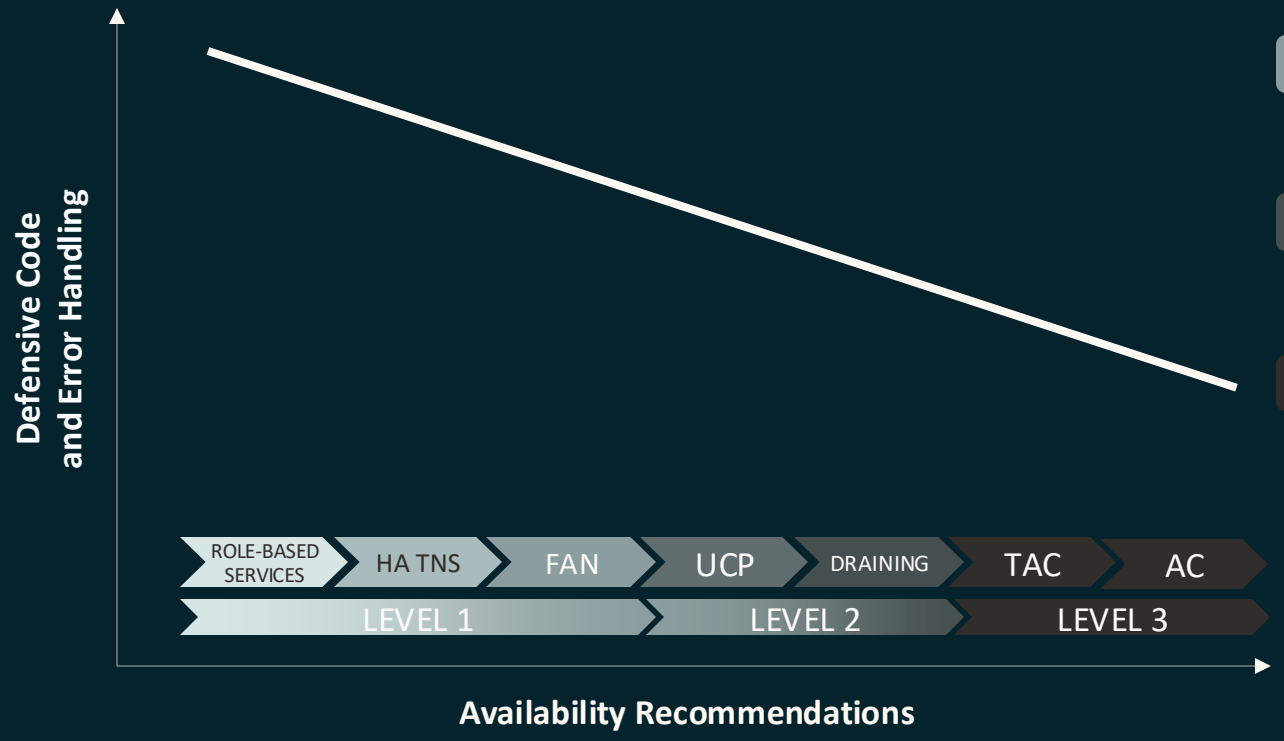
- Enables the automated rolling application of version-changing upgrades and patch sets

Together

- They hide the final switchover needed at the end of the fully automated DBMS_ROLLING process.
  - 23ai to 23ai available now / 19c to 23ai planned for future 19c RU



Higher Availability and less defensive programming



LEVEL 1

Basic HA

LEVEL 2

Prepare for Planned Maintenance

LEVEL 3

Mask Outages

Level 1: Services

User-defined database services for location independence

```
$ srvctl add service -d <DATABASE NAME> -s <SERVICE NAME> -pdb <PDB NAME>
```

Load balancing

```
-preferred <INSTANCE NAME, …> -available <INSTANCE NAME, …>
-notification TRUE
```

FAN/ONS for OCI

Application Continuity

```
-commit_outcome TRUE
-failovertype AUTO
-replay_init_time 600
-failover_restore AUTO
```

Workload draining

```
-drain_timeout 300
-stopoption IMMEDIATE
-role PRIMARY/PHYSICAL STANDBY
-resetstate (option)
```

For role-based services when using Data Guard

Level 1: Recommended Connection String

Standard connection string works with Oracle Single Instance, Oracle RAC, and Data Guard

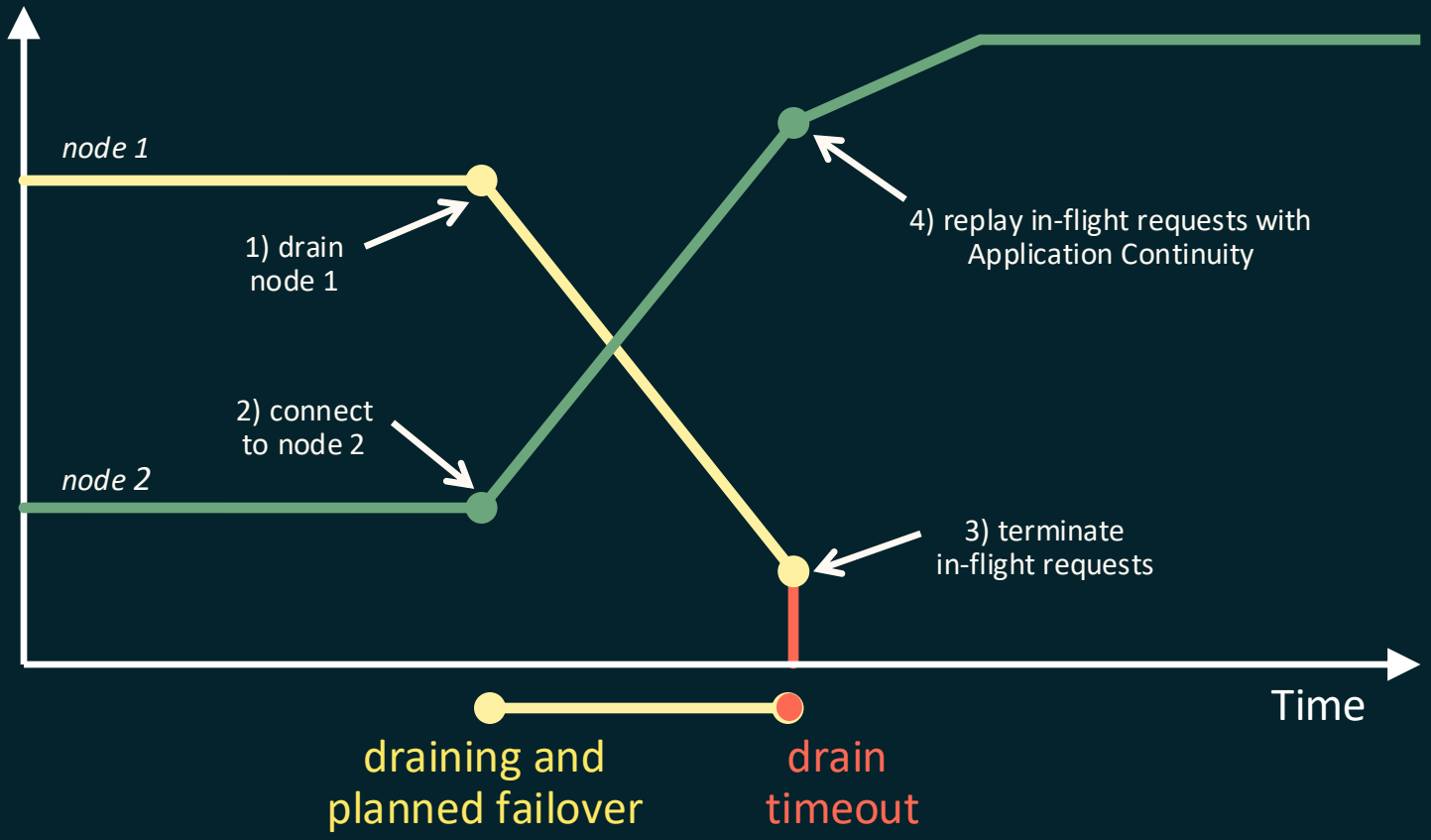
```
SSPDB =
(DESCRIPTION =
  (CONNECT_TIMEOUT=90)(RETRY_COUNT=50)(RETRY_DELAY=250ms)
  (TRANSPORT_CONNECT_TIMEOUT=1000ms)
  (FAILOVER=ON)(LOAD_BALANCE=OFF)
  (ADDRESS_LIST =
    (LOAD_BALANCE=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=1521)))
  (ADDRESS_LIST =
    (LOAD_BALANCE=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST=standby-scan)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME = sspdb.oraclecloud.com)))
```


Level 1: Fast Application Notification (FAN)

- Enables Fast Connection Failover
- Server automatically reports Oracle Notification Server (ONS) configuration to client at authenticate
  - Client automatically subscribes to ONS to receive FAN Events
  - Requirement: Recommended Connection String being used
- Deploy the following JDBC JAR files
  - ojdbcXX.jar, ons.jar, simplefan.jar
- Port 6200 needs to be open on server side
- Example FAN event sent via ONS to client:

```
FAN event type: service_member
Properties: version=1.0 service=OLTP database=SSDB instance=SSDB
host=bumucsvm1 status=up
```


Level 2 Onwards: Drain… Connect… Failover



Monitor Drain timeout

New In 23ai

Make informed decisions about tuning the drain timeout

- Duration of requests is often unknown to DBAs and developers
- Planned maintenance
  - Give applications sufficient time to drain
  - Small enough to proceed with maintenance
- v$service_drain_timeout_advice
  - LAST_DRAIN_TIMEOUT
  - Number of sessions MARKED_TO_DRAIN
  - Number of sessions DRAINED

```
Service      Time Interval      Frequency (in 10's)
-------------------------------------------------------
tac          <= 1 sec           **************
tac          <= 5 sec           *****
ac           <= 5 sec           **************
ac           <= 30 sec          **********
ac           <= 60 sec          *****
```

Deutsche Bank
Identifier

Harsh Gupta

Head of Database Engineering

Oracle Cloud World

#PositiveImpact

Las Vegas, Sept 2024

AC/TAC @Deutsche Bank (DB)

Goal @DB

- Zero Downtime planned maintenance
- Zero Downtime unplanned outages

Challenge @DB

- Every application is unique in itself
- How to analyze 2000+ application
- How to scale project implementation

Oracle @DB

- AC-TAC product team work closely with us
- Client business needs are well understood

Program @DB

- Analyze application design
- Categorize them as Easy, Medium & Complex
    - Decision Matrix
- POC and Sign Off
- Prepare Blue-Print to achieve AC-TAC – Self Service

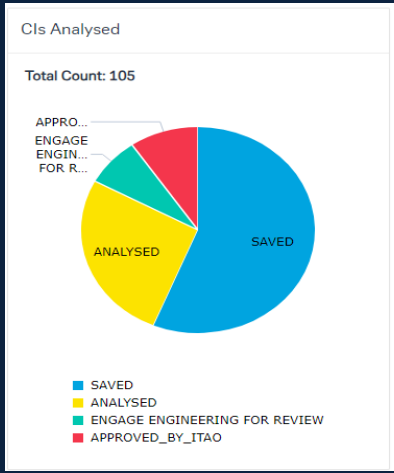
How do we execute

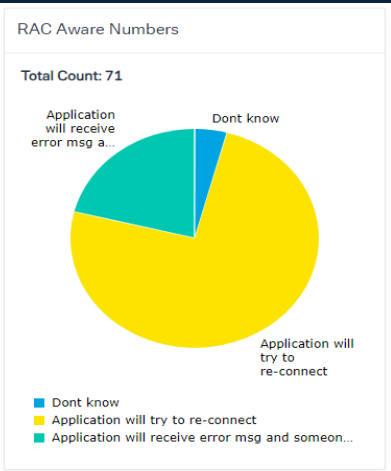
Inventory Management

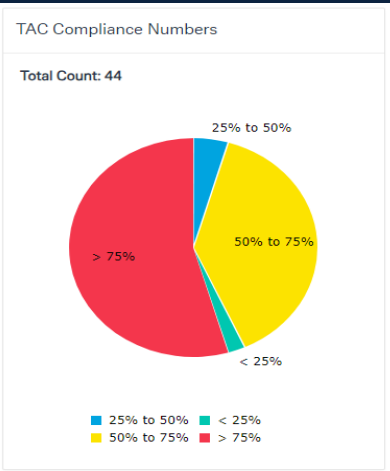
Questionnaire	Action History	Analysis History.	TAC POC	Tac POC Status.	TAC ITAO Signoff	Last Action	Latest TAC Compliance %..
				IN_PRODUCTIO...		ANALYSED	91
				IN_PRODUCTIO...		ANALYSED	100
				IN_PRODUCTIO...		ANALYSED	63
				IN_PRODUCTIO...		SAVED	98
				OUT_OF_SCOPE		SAVED	70
				OUT_OF_SCOPE		SAVED	
				STOPPED		ANALYSED	63
				STOPPED		ANALYSED	76

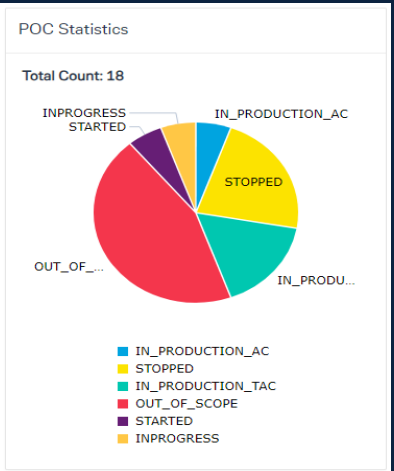
Testing Scenario

Category	Scenario
Outage	Kill Session/application Job Background Process
Outage	Instance Crash inside a cluster
Planned Maintenance	RDBMS Database Patching
Planned Maintenance	Database Switchover - Not valid in UAT testing
Planned Maintenance	Service relocation inside a cluster

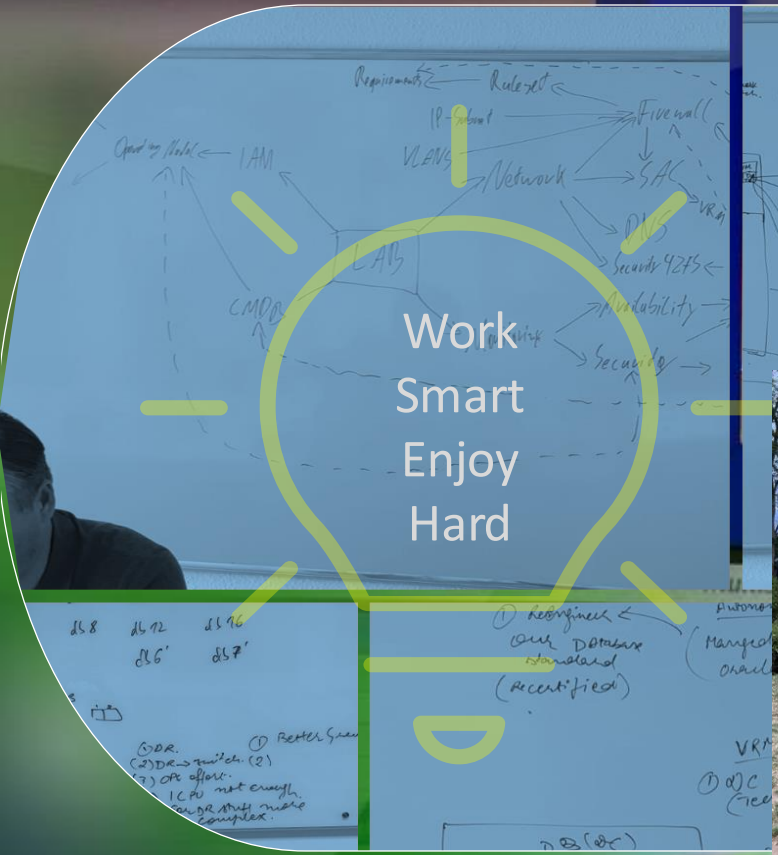








Deutsche Bank
Identifier



Deutsche Bank
Identifier

ExaC@C Platform – Oracle RAC with (Transparent) Application Continuity

Testimonials: How applications can achieve continuous availability on the ExaC@C platform during planned maintenance and outages

ORACLE
DatabaseWorld
at CloudWorld

Thank you!

For questions, please, contact:

Markus.Michalewicz@oracle.com

Sebastian.Solbach@oracle.com

