Developer Runbook for Application Continuity

Carol Colrain
Ian Cookson
Troy Anthony

Oracle Database Development
Safe Harbor

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.

Statements in this presentation relating to Oracle’s future plans, expectations, beliefs, intentions and prospects are “forward-looking statements” and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle’s Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading “Risk Factors.” These filings are available on the SEC's website or on Oracle’s website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.
Program Agenda

What is Continuous Availability?
Maintenance Without Impact
Transparent Application Continuity
Customer Examples
What is Continuous Availability?

Continuous Availability focuses on the User Experience

• Planned maintenance, timeouts, unplanned outages hidden
• Inflight work continues
• Performance is predictable

It is easy to achieve Continuous Availability today
Is continuous availability achievable today?

Absolutely, with best practices.

Well behaved application

Supported connection stack

Properly configured Oracle DB

Stars align when our tips are followed
What are the pre-requisites to achieve this?

1. A well behaved application – we will explain how TAC solves this
2. Enabled Client Stack (driver, pool, or app server)
   Oracle or enabled 3rd party stack
   Properly configured Net connection
3. Properly configured database and Grid Infrastructure
   Configuration necessary to cover likely outages
4. Business SLAs can be met during outages
   Recovery time performance targets
   Maintenance window

Many deployments do not currently meet these requirements!
Baseline – Understand where you are

Is the workload divided into services?

Is the application using a proper connection string?

If using a connection pool are connections returned regularly?

Does your application have built-in failover?

Do you have head room to failover to other nodes?

Are application level timeouts set?

Are special features used: XA, external files, side-effects such as file transfers, OJVM

Not essential, but a head start if you do
New Concept: Request Boundaries
Delineate the Unit of Work

Oracle Pools 12c+, Java Standard (JDK9+), Transparent Application Continuity (TAC)

PoolDataSource pds = GetPoolDataSource();
Connection conn = pds.getConnection();
PreparedStatement pstmt = …

... SQL, PL/SQL ...

... How do we use Requests?
- Draining
- Load Balancing
- Recovery of in-flight work
- Planned Failover
- Reset session state
- Performance metrics
- Protection level

conn.commit();
conn.close();

Begin Request
Request Body often ends with COMMIT
End Request
Maintenance with No Impact

Preventable situations:

- Service is unavailable
- Application owners unable to agree maintenance windows
- Long running jobs see errors
- DBA’s and engineers work off hours
- Application and middleware components need to be restarted

Family Holidays website is down
"Our online transaction services are currently unavailable. Our server may be temporarily down or we may be performing routine maintenance functions scheduled every Sunday from 12 a.m. to 5 a.m. (Eastern Standard Time). We apologize for any inconvenience."
Drain Work before Maintenance

Services are relocated
Automatically, user work drains

Many places to drain:
- Return to connection pool
- Connection tests
- Log-Off
- Planned Failover
Keep it simple – Drain Batch then Online

Use BATCH services

Use OLTP services

Sample Runtime Seconds vs Number of Jobs Completed for ERP Customers
Stagger Draining – Batch then Online Services

Use at least two application services – online and batch/backend
Start draining batch early
Online drains in seconds

First Instance

Online

Batch

Shut-down

Online

Online

Last Instance

Online

Batch

Shut-down

Online

Online

Quiesce Batch for X hours...
Quiesce OLTP for Y seconds

Complete Maintenance
# Tip: Use Oracle Pool, Return to Pool

## BEST SOLUTION – FAN Drains and Rebalances

### Applications use

Oracle Pools: WebLogic Active GridLink, UCP, ODP.NET managed and unmanaged, OCI Session Pool, Tuxedo

3rd party App Servers using UCP: IBM WebSphere and Liberty, Apache Tomcat, NEC WebOTX, Red Hat WildFly (JBoss), Spring

### DBA uses

`srvctl [relocate|stop] service -drain_timeout`

### Sessions Drain Automatically

- Immediately new work is redirected
- Idle sessions drain gradually
- Active sessions are released when returned to pools
All Oracle uses ONS

<table>
<thead>
<tr>
<th>All Oracle uses ONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC Universal Connection Pool</td>
</tr>
<tr>
<td>OCI/OCCI driver</td>
</tr>
<tr>
<td>ODP.NET Unmanaged Provider (OCI)</td>
</tr>
<tr>
<td>ODP.NET Managed Provider (C#)</td>
</tr>
<tr>
<td>OCI Session Pool</td>
</tr>
<tr>
<td>WebLogic Active GridLink</td>
</tr>
<tr>
<td>Tuxedo</td>
</tr>
<tr>
<td>JDBC Thin Driver (new 12.2)</td>
</tr>
<tr>
<td>CMAN in Traffic Manager mode (new 18c)</td>
</tr>
</tbody>
</table>

Auto-Configured

DESCRIPTION =
(CONNECT_TIMEOUT=90)
(RETRY_COUNT=20)(RETRY_DELAY=3)
(TRANSPORT_CONNECT_TIMEOUT=3)

(ADDRESS_LIST =
(LOAD_BALANCE =
( ADDRESS = (PROTOCOL = TCP)
(HOST=primary-scan) (PORT=1521)))

(ADDRESS_LIST =
(LOAD_BALANCE =
( ADDRESS = (PROTOCOL = TCP)
(HOST=second-scan)(PORT=1521)))

(CONNECT_DATA=(SERVICE_NAME=gold)))

ONS Node Set 1

ONS Node Set 2
FAN for JDBC

12c or later JDBC clients and database

1. Use recommended URL for auto-ons
2. ons.jar (plus Wallet jars, osdt_cert.jar, osdt_core.jar, oraclepki.jar) on the CLASSPATH
3. UCP pool property fastConnectionFailoverEnabled=true
4. Third party pools, UCP is recommended
5. Open port 6200 for ONS

Before 12c clients

Set oracle.ons.nodes =mysun05:6200,mysun06:6200, mysun07:6200,mysun08:6200
Tip: 3rd Party Java Applications

A very simple jar replacement

- IBM WebSphere
- IBM Liberty
- Apache Tomcat
- NEC WebOTX
- Red Hat WildFly (JBoss)
- Hibernate
- Spring
- custom

Simple jar replacement
- Replay driver for Application Continuity
- UCP jar for pool
- ONS jar for FAN
Tip: Enable Connection Tests

18c Database and 18c Drivers

Application “tests” the connection
Database/driver responds connection is bad
New work continues on another connection

Tip: View in DBA_CONNECTION_TESTS
Add more tests with DBMS_APP_CONT_ADMIN
Tip: Enable Connection Tests

Client-Side

UCP
setValidateConnectionOnBorrow(), isValid is local

OCI
OCI_ATTR_SERVER_STATUS in server handle

ODP.NET
CheckConStatus is on by default.

Database-side

<table>
<thead>
<tr>
<th>CONNECTION_TEST</th>
<th>SQL_CONNECTION_TEST</th>
<th>ENABLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_TEST</td>
<td>SELECT 1 FROM DUAL</td>
<td>Y</td>
</tr>
<tr>
<td>SQL_TEST</td>
<td>SELECT COUNT(*) FROM DUAL</td>
<td>Y</td>
</tr>
<tr>
<td>SQL_TEST</td>
<td>SELECT 1</td>
<td>Y</td>
</tr>
<tr>
<td>SQL_TEST</td>
<td>BEGIN NULL;END</td>
<td>Y</td>
</tr>
<tr>
<td>PING_TEST</td>
<td></td>
<td>NA</td>
</tr>
<tr>
<td>ENDREQUEST_TEST</td>
<td></td>
<td>NA</td>
</tr>
</tbody>
</table>
Tip: Enable Connection Tests for Application Servers

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Test Name</th>
<th>Connection Test to DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic – Generic and Multi data sources</td>
<td>TestConnectionsOnReserve</td>
<td>isUsable() SQL – SELECT 1 FROM DUAL</td>
</tr>
<tr>
<td></td>
<td>TestConnectionsOnCreate</td>
<td></td>
</tr>
<tr>
<td>Oracle WebLogic Active GridLink</td>
<td>Embedded</td>
<td>isUsable()</td>
</tr>
<tr>
<td>IBM WebSphere</td>
<td>PreTest Connections</td>
<td>SQL – SELECT 1 FROM DUAL</td>
</tr>
<tr>
<td>RedHat WildFly (JBoss)</td>
<td>Check-valid-connection-sql</td>
<td>SQL – SELECT COUNT(*) FROM DUAL</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>TestOnBorrow</td>
<td>SQL – SELECT 1 FROM DUAL</td>
</tr>
<tr>
<td></td>
<td>TestOnRelease</td>
<td></td>
</tr>
<tr>
<td>ODP.Net Unmanaged</td>
<td>Connection.status()</td>
<td>OCI_ATTR_SERVER_STATUS</td>
</tr>
</tbody>
</table>
Tip: Enable Connection Tests for Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Condition</th>
<th>Connection Test to DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBusiness Suite</td>
<td>Connection borrowed from WebLogic</td>
<td>TestConnectionsOnReserve with “BEGIN NULL;END;”</td>
</tr>
<tr>
<td>Fusion Applications</td>
<td>Connection returned to WebLogic and C++ pools and checked</td>
<td>TestConnectionsOnReserve with isValid()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCI_ATTR_SERVER_STATUS</td>
</tr>
<tr>
<td>Siebel</td>
<td>Connection requested</td>
<td>OCI_ATTR_SERVER_STATUS</td>
</tr>
<tr>
<td>Peoplesoft</td>
<td>Connection requested</td>
<td>OCI_ATTR_SERVER_STATUS</td>
</tr>
<tr>
<td>Customer example</td>
<td>Custom pool with Metadata table Checks status every 60 seconds</td>
<td>OCI_ATTR_SERVER_STATUS</td>
</tr>
</tbody>
</table>
Align Application Timeouts

TIP: Nothing to do when RAC or RAC One

TIP: Set Application Timeout > Drain + Switch to DG

Stop or Relocate Service to Drain Work

RAC Primary

RAC Standby

Drain Work

Switchover
Preset services to drain

\texttt{drain\textunderscore timeout} 600 sec.

\texttt{stopoption} immediate

Group operations pdb, instance, node, database

\textbf{FAN} full and in-band
Drain Sessions Before Maintenance

Demonstration
Drain... Connect... Failover

1. Drain Node 1
2. Connect to Node 2
3. Terminate in-flight requests
4. Replay in-flight requests

Copyright © 2019 Oracle and/or its affiliates.
In-flight work continues:

- Replays in-flight work on recoverable errors
- Masks hardware, software, network, storage, session errors and timeouts
- 12.1 JDBC-Thin, UCP, WebLogic Server, 3rd Party Java application servers
- 12.2 OCI, ODP.NET unmanaged, JDBC Thin on XA data source
- Transparent Application Continuity
- TAC on by default on ADB Dedicated
Standardize on TAC to hide impact of outages
### Application Continuity Explained

#### Normal Operation

Client marks requests: explicit and **implicit**.

Server **tracks session state**, decides which calls to replay, disables side effects.

Directed, client holds original calls, their inputs, and **validation data**.

#### Failover Phase 1: Reconnect

- Checks replay is enabled
- Verifies timeliness
- Creates a new connection
- Checks target database is legal for replay
- Uses Transaction Guard to guarantee commit outcome

#### Failover Phase 2: Replay

- Restores and verifies the session state
- Replays held calls, restores mutables automatically
- Ensures results, states, messages match original.
- On success, returns control to the application
Exclusions
When replay is not enabled

**Application Level**
Default database or
default PDB service

Deprecated, JDBC
classes before 18c

**Remainder of Request**
Alter system, database,
session (subset)

Best effort streaming
lobs

XA after promote

OCI – old
OCIStmtPrepare, misc
apis

DB links ADG to primary

**Target Database**
Different Database

DBMS_ROLLING*

Golden Gate

3rd Party Replication
Using Transparent/Application Continuity

<table>
<thead>
<tr>
<th>1. Request Boundaries</th>
<th>Use Oracle pools and Return to Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Session State</td>
<td>Use FAILOVER_RESTORE</td>
</tr>
<tr>
<td>3. Side Effects</td>
<td>TAC - not replayed, AC - customizable</td>
</tr>
<tr>
<td>4. SYSDATE, Sequences,...</td>
<td>Do nothing for SQL, Grant for PL/SQL</td>
</tr>
<tr>
<td>Reset State</td>
<td>Reset session state between requests</td>
</tr>
<tr>
<td>Coverage</td>
<td>Always know your protection level</td>
</tr>
</tbody>
</table>
Request Boundaries Are Automatic

Transparent Application Continuity

- Request boundaries advanced automatically
- Capture re-enables, if disabled
- Smaller capture set means faster recovery
- Pooled, non-pooled and long running

For highest protection
- Return connections to pool
- Return cursors to driver statement cache
- Reset states (option)
Explicit Request Boundaries, Standard in JDK9
Required for AC, Recommended with TAC

Oracle
Return connections
UCP
WebLogic GridLink
ODP.NET unmanaged
OCI Session Pool
SQL*Plus

Use UCP or Java Standard
Return connections
IBM WebSphere
IBM Liberty
Apache Tomcat
NEC WebOTX
RedHat Wildfly (JBoss)
Spring
custom

Custom Java
Return connections
Add Request boundaries
beginRequest
endRequest
Session State Restored for Replay

• Restore session states before replaying
  • `FAILOVER_RESTORE` on your service

• Full parameter restore
  • Configure wallets

• Customization
  • Labels, TAF callback, Logon triggers
Set FAILOVER_RESTORE on your service and use wallets
Side Effects Not Replayed
TAC – stops capture automatically until next enable point

TAC decides if any requests should not be replayed, e.g.
- UTL_HTTP
- UTL_URL
- DBMS_FILE
- DBMS_FILE_TRANSFER
- UTL_SMTP
- UTL_TCP
- UTL_MAIL
- EXTPROC

Customized? - use AC
Restore SYSDATE, SYSTIMESTAMP, Sequences ...

During replay the same values are restored for SYSDATE, SYSTIMESTAMP, and SEQUENCES

- Automatically for SQL
- Grant keep for PL/SQL

For owned sequences:

```sql
ALTER SEQUENCE.. [sequence]  [KEEP|NOKEEP]
CREATE SEQUENCE.. [sequence]  [KEEP|NOKEEP]
```

Grant and Revoke for other users:

```sql
GRANT [KEEP DATE TIME | KEEP SYSGUID][to USER]
REVOKE [KEEP DATE TIME | KEEP SYSGUID] [from USER]
GRANT KEEP SEQUENCE on [sequence][to USER]
REVOKE KEEP SEQUENCE on [sequence][from USER]
```
Reset Session State Between Requests

Applications use state in requests
  Temporary tables
  PL/SQL globals
  Cursors in fetch ...

Database resets state at end of request
Next request starts with clean state
Service attribute (RESET_STATE)
Always Know Your Protection Level

• AWR, system, session, service statistics
• Your application is fully protected when
  \( \text{cumulative user calls in request} = \text{cumulative user calls protected} \)

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Total</th>
<th>per Second</th>
<th>per Trans</th>
</tr>
</thead>
<tbody>
<tr>
<td>cumulative begin requests</td>
<td>1,500,000</td>
<td>14,192.9</td>
<td>2.4</td>
</tr>
<tr>
<td>cumulative end requests</td>
<td>1,500,000</td>
<td>14,192.9</td>
<td>2.4</td>
</tr>
<tr>
<td>cumulative user calls in request</td>
<td>6,672,566</td>
<td>63,135.2</td>
<td>10.8</td>
</tr>
<tr>
<td>cumulative user calls protected</td>
<td>6,672,566</td>
<td>63,135.2</td>
<td>10.8</td>
</tr>
</tbody>
</table>
### Coverage checks

<table>
<thead>
<tr>
<th>Outage Type</th>
<th>Status</th>
<th>Message</th>
</tr>
</thead>
</table>
| Coverage checks | | TotalRequest = 1088  
PASS = 1084  
WARNING = 1  
FAIL = 3 |
| FAIL | Trace file name = db1_ora_30467.trc Line number of Request start = 1409 Request number = 6  
SERVICE NAME = (srv_auto_pdb1) MODULE NAME = (SQL*Plus) ACTION NAME = () CLIENT ID = ()  
Coverage(%) = 12 Protected Calls = 1 Unprotected Calls = 7  
Row number of the last call before DISABLED : 1422, Disable reason : ORA-41429 |
| WARNING | Trace file name = CDB12_ora_321597.trc Line number of Request start = 653 Request number = 1  
SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (JDBC Thin Client) ACTION NAME = () CLIENT ID = ()  
Coverage(%) = 25 Protected Calls = 1 Unprotected Calls = 3  
Row number of the last call before DISABLED : 668, Disable reason : ORA-41409 |
| FAIL | Trace file name = CDB12_ora_292714.trc Line number of Request start = 1598 Request number = 7  
SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (SQL*Plus) ACTION NAME = () CLIENT ID = ()  
Coverage(%) = 16 Protected Calls = 1 Unprotected Calls = 5  
Row number of the last call before DISABLED : 1622, Disable reason : ORA-41429 |
| FAIL | Trace file name = CDB12_ora_112022.trc Line number of Request start = 1167 Request number = 3  
SERVICE NAME = (PDB1_tp.cloud.com) MODULE NAME = (JDBC Thin Client) ACTION NAME = () CLIENT ID = ()  
Coverage(%) = 0 Protected Calls = 0 Unprotected Calls = 1  
Row number of the last call before DISABLED : 1024, Disable reason : ORA-41406 |
| PASS | Report containing checks that passed: /scratch/nfs/acchk/orachk_dbj14_091119_10491/reports/acchk_scorecard_pass.html |
Configuration for Clients

JDBC Thin

Use replay data source

replay datasource=oracle.jdbc.replay.OracleDataSourceImpl

Use JDBC driver statement Cache

Use acchk to check for concrete classes pre-18c driver

OCI, ODP.Net unmanaged, SQL*Plus (19c), open source

On when enabled on the service

Use OCI driver statement Cache
Configuration for Services

DRAIN_TIMEOUT = < long time, in seconds >
STOP_OPTION = IMMEDIATE
FAILOVER_TYPE = AUTO or TRANSACTION
FAILOVER_RESTORE = AUTO or LEVEL1
COMMIT_OUTCOME = TRUE
AQ_HA_NOTIFICATIONS=True for FAN OCI
REPLAY_INITIATION_TIMEOUT = 300 (seconds before canceled)
RESET_STATE = NONE or LEVEL1 (for stateless apps)
Align Application Timeout

- Application Timeout
  - FSFO + Crash Recovery
  - 2 x MISSCOUNT + FSMT (60s)

- FAST START FAILOVER (FSFO)

- Application Timeout (SLA)

- e.g. - MISSCOUNT (15)
  - EXA Only (2)
  - FAST START MTTR TARGET (30)
  - AC REPLAY TIMEOUT (600)

- Crash Recovery (FSMT + Open)
- AC REPLAY TIMEOUT
<table>
<thead>
<tr>
<th>My Application uses..</th>
<th>TAC</th>
<th>AC</th>
<th>Draining</th>
</tr>
</thead>
<tbody>
<tr>
<td>I don’t know</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Transactions</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Oracle state (temp lobs, PL/SQL, temp tables, aq..)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>No connection pool</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Initial state on the session</td>
<td>Yes and custom</td>
<td>Yes and custom</td>
<td>yes</td>
</tr>
<tr>
<td>Side effects (utl_mail, dbms_file_transfer..)</td>
<td>Yes, not replayed</td>
<td>Yes, custom</td>
<td>yes</td>
</tr>
</tbody>
</table>
Customer Stories – Unplanned Outages
Chicago Mercantile Exchange
CME Group Overview

CME Group is the world’s leading and most diverse derivatives marketplace bringing together those who need to manage risk or those that want to profit by accepting it.

- Operating Multiple Exchanges – CME, CBOT, Nymex and COMEX
- Trade hundreds of products across the globe on a single platform
- Average daily volume of 15.6 million contracts

- CME Clearing – matches and settles all trades and guarantees the creditworthiness of every transaction
- Cleared more than 3.9 billion contracts with a value exceeding $1 quadrillion
- Highest Volume Day – 44.5 million contracts after the election
CME HIGH AVAILABILITY OVERVIEW

Requirements

• Critical DB’s – 10 second to SLA
• Component Failure Cannot cause DR Event
• 24X7 Application up time
  • Including Planned Maintenance
• RPO – 30 seconds (Disaster Only)
• RTO – 2 hours (Disaster Only)

Solution

• Exadata
  • Addresses Performance
  • Allows Consolidation
  • Reduces recovery time (component failure)
• Active Data Guard
• Application Continuity – Planned/unplanned
**WHY CME IS ADOPTING APPLICATION CONTINUITY**

- Database Outages cause in-flight work to be lost
- A Database Outage can effect many applications concurrently due to schema consolidation
- Critical Applications are becoming 24x7 – These are referential applications
- Database planned downtime on behalf of patching is exceedingly harder to schedule due to shrinking maintenance windows.
- Avoid dedicating maintenance windows to the database group
- Applications work together as a system. It can take several hours to start and normalize
## Failure Proofing Applications Is Hard

<table>
<thead>
<tr>
<th>What’s Hard</th>
<th>What’s Today’s Solution</th>
</tr>
</thead>
</table>
| Hanging on TCP/IP Timeouts – Application is not aware of an issue because there has been no ack for the last operation | • FAN – Fast Application Notification  
• FCF – Fast Connection Failover  
• ONS – Notification Services  
These features work together to overcome TCP hangs |
| Reconnecting to surviving nodes or standby database after failure          | Application Continuity automatically performs connection retries all configurable in the connection string |
| Assuring any in-flight transactions were committed to the database.        | Application Continuity features handle this transparently. Transactions are crosschecked and replayed safely |
| Confidence leaving applications live during planned Database Maintenance   | AC has proven to be resilient at CME.                                                  |
Normal Operation

- All OLTP services configured as 1 active, rest available
- Over 400 services across environment
- Over 100 applications
- Node capacity actively managed

- Most Application Servers “Lie in Wait”
- Critical Applications are connected in a RO mode
Planned Maintenance

- Exadata Full Stack Patching takes 4 hours at best
- CME does not do rolling patches (duration too long)
- AC allows apps to stay up and undergo updates while patching happens.
Planned Maintenance

- DR is always patched first
- Applications in DR are taken offline
- Normal change window applies
- Application changes in PROD coincide with DR patching
Planned Maintenance

- Local Standby databases are patched after DR
- Patching the local standby database does not impact running application
- Patched during normal maintenance window
- Application changes and testing can continue

PROD

EXA 1

App Cluster 1 (WL Tomcat or Stand-Alone java)

App Cluster 2 (WL Tomcat or Stand-Alone java)

DR

EXA 1

App Cluster 1 (WL Tomcat or Stand-Alone java)

App Cluster 2 (WL Tomcat or Stand-Alone java)
Planned Maintenance – Database As A Service

- AC compliant applications stay running and available
- Non compliant applications are stopped and restarted (Transition period)
- A database switchover is performed
- An LDAP job modifies connection strings for non compliant apps

- Non compliant apps are restarted
- Changes and testing continues during maintenance window
- Process repeated for fail back
UNPLANNED OUTAGES

- Node 1 fails
- All services fail to available instance (2 illustrated)
- Application connections follow service location using Application Continuity
UNPLANNED OUTAGES

• What if the whole Exadata Fails?
• At CME – this is not allowed to cause a DR event
UNPLANNED OUTAGES

- Catastrophic Data Center Failure
  - Uncontrolled network outage (All HA FAILS)
  - Physical Damage to building
  - EXA 1 and EXA 2 fail in same week

- Critical Apps Up for customer RO access
- Databases are converted – Apps convert to RW
- All apps started - < 2 hours
- All automated

© 2017 CME Group. All rights reserved.
CME Best Practices

- Good test environment that mirrors production
- Credible HA and DR testing methodology on a mandatory interval commensurate with your change rate
- Application simulation for testing that is realistic
- Capacity Planning – keep utilization of servers <50%
- Client Interrupted using FAN and FCF
- Time Based Failover, supported by Application Continuity
- No Single Points of Failure
- Strong Change Control
The preceding 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.

Statements in this presentation relating to Oracle’s future plans, expectations, beliefs, intentions and prospects are “forward-looking statements” and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle’s Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading “Risk Factors.” These filings are available on the SEC’s website or on Oracle’s website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.