

**How often do you patch
your Oracle Databases?**





Upgrade.Patch.Repeat.

Martin Berger



SPEAKER

Martin Berger

martin.x.berger@accenture.com

Database Platforms

Data Eng, Mgmt & Governance Mgr


accenture

- LOCATION Kestenholz - Jurasüdfuss - [Switzerland](#)
- EXPERIENCE Oracle Database Platforms
- FOCUS Multicloud - Engineered Systems
- COMMUNITY [Oracle ACE Pro](#)  - Speaker
- BEYOND WORK Firefighter (2nd in Command) - E-Biker - Runner
- FAMILY 2 Junior-DBAs at home (12 & 14)



www.martinberger.swiss

Accenture's Oracle Expertise and Partnership

Proven leader, focused on customer success

Modern Data Platforms

WARUM? Wir sind der Game Changer für die Datenplattform-Projekte unserer Kunden

WIE? Maximale Automatisierung, maximale Effizienz, maximale Qualität!

WAS? Wir entwickeln innovative Datenplattformen basierend auf unseren Blueprints, Assets und Tools.



3 Hauptvorteile

1 Architektur-Expertise aus diversen Projekten

2 Lieferung massgeschneiderter Datenplattformen

3 Integrierte Teams–Wie eine Rudermannschaft, perfekte Abstimmung und Interaktion.



Tools und Blueprints

Schlüsselfaktor für die Implementierung moderner Datenplattformen mit hoher Geschwindigkeit und Qualität.

Kontinuierliche Optimierung

Tools und Blueprints werden kontinuierlich an die Kunden- und Projektbedürfnisse angepasst.

Expertise & Light Towers

Expertengruppe für moderne Datenplattformen – von der technischen Umsetzung bis zum Projektmanagement und zur Organisation



MERCI SOUG FÜR DIE COOLE AKTION ❤️



Gemeinsam Familien mit kranken Kindern unterstützen

Wenn ein Kind schwer erkrankt und im Spital behandelt werden muss, steht für die ganze Familie plötzlich die Welt still.

Das **Ronald McDonald Haus Bern** bietet Familien ein Zuhause auf Zeit – einen Ort zum Schlafen, Durchatmen und Kraftschöpfen, ganz in der Nähe ihres Kindes.

Auch in unserer **SOUG Community** gibt es eine Familie, die bereits auf diese **wertvolle Unterstützung** angewiesen war. Das zeigt uns, wie wichtig das Haus für betroffene Familien ist.

Möchtest du uns dabei unterstützen?

Scanne den QR-Code und spende direkt an das Ronald McDonald Haus Bern.

Jeder Beitrag hilft. Die Spende geht direkt an das Haus; die SOUG nimmt keine Spenden entgegen.

Jetzt mit TWINT spenden!

QR-Code mit der TWINT App scannen

Betrag und Spende bestätigen



Herzlichen Dank für deine Unterstützung!

Swiss Oracle User Group (SOUG)

Scannen & direkt spenden





00 INTRO

01 ORACLE_HOME

03 UPGRADE

04 TEAM

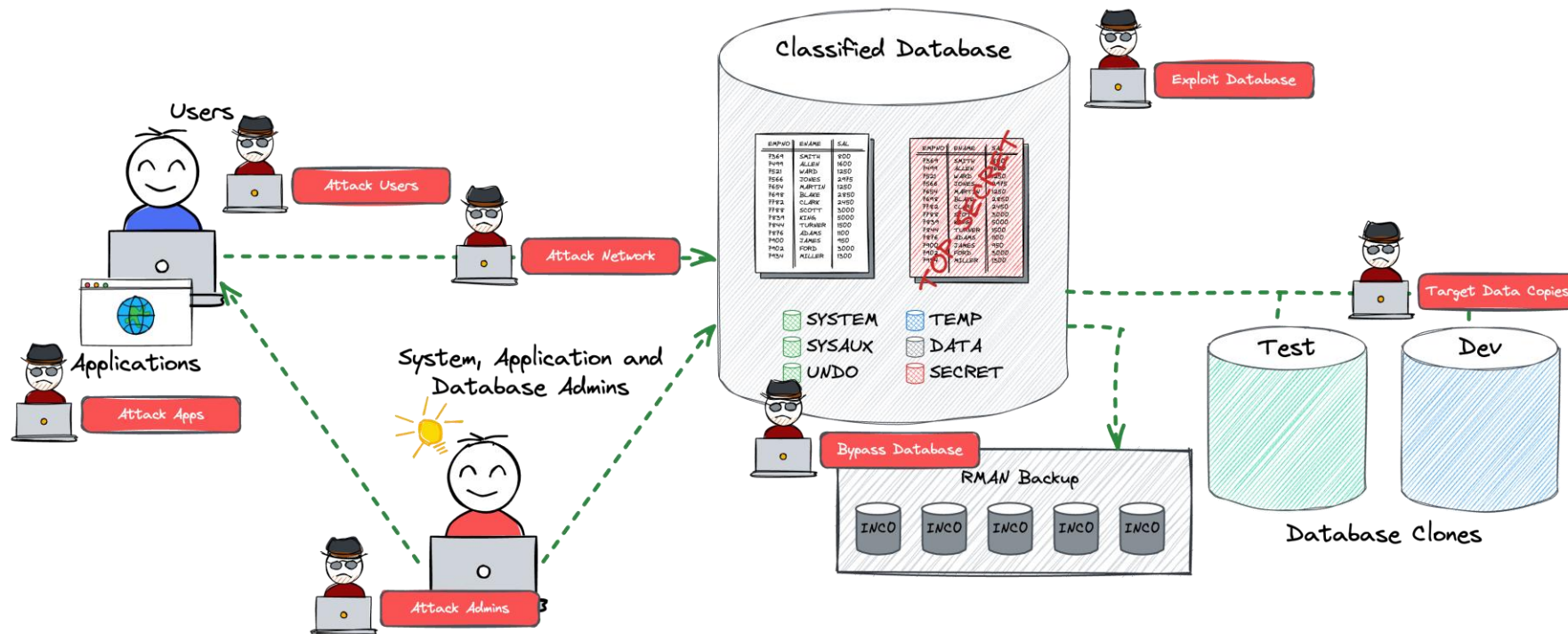
05 DEMO TIME

00

INTRO

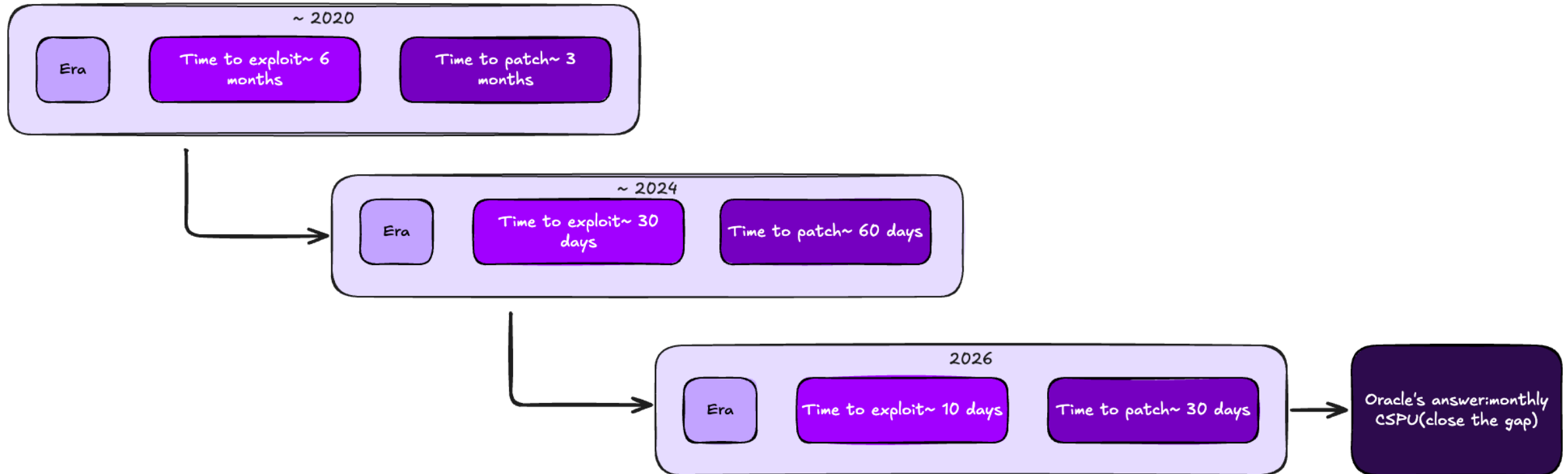
Where attackers come at the Oracle DB

- Listener / network surface TLS stack, naming services, anonymous probes
- Privilege paths PUBLIC, ANY-system privs, granted roles
- SQL injection via views, procs, the app tier
- Data exfiltration DB links, DataPump, UTL_FILE/UTL_HTTP



The pace is getting higher

- 2020: Exploit ~6 months / Patch ~3 months
- 2024: Exploit ~30 days / Patch ~60 days (gap closing)
- 2026: Exploit ~10 days / Patch ~30 days Oracle's answer: monthly cadence
- On Linux 19c/26ai delivered as MRP, on other platforms as CSPU



Four terms, one model

RU	MRP	CSPU	CPU
Release Update Quarterly cumulative baseline - contains all prior MRPs/CSPUs of the quarter. Always apply first. 4× per year (Jan / Apr / Jul / Oct).	Monthly Recommended Patch Linux 19c and 26ai only. Security + regression + third-party fixes. Up to 6 per RU window. Apply on top of the current RU.	Critical Security Patch Update Non-Linux platforms (Windows, Solaris, AIX). Security fixes only a subset of MRP. Same monthly schedule. Only option on non-Linux.	Critical Patch Update Historical umbrella term - no longer a specific DB patch bundle. Used generically across Oracle products. Not a deliverable you download.
Example 19.31, 23.26.2	Example MRP May-26, MRP Jun-26	Example Windows/Solaris/AIX 19c	Example Legacy marketing term

n-1 was a reasonable heuristic - the math changed

N-1 STRATEGY (BEFORE)

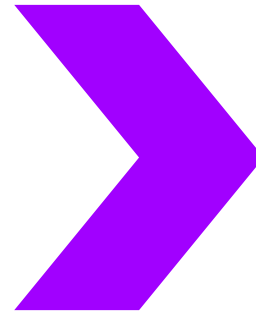
Take the previous RU

Apply RU n-1 (e.g. 19.31 in July 2026 instead of 19.32)

Rationale: seasoned patches, known issues documented, community tested.

Timeline reality: requests one-offs (3 wks) → test/dev rollout → app team sign-off → production.

Best case: 4–6 weeks total.



**AI CHANGED
THE MATH**

CURRENT REALITY (TODAY)

515 security fixes missing

Comparing 19.31 + MRP3 (Jul 2026) vs 19.32 (same release date):

- 656 total fixes missing
- 515 of those are security fixes

RUs are the **PRIMARY** vehicle for security content.
MRPs/CSPUs deliver the highest-score subset only.

n-1 means accepting a half-year security deficit every cycle.

<https://mikedietrichde.com/2026/09/21/patching-recommendations-which-route-is-the-best-rus-mrps-n-1/>



What do i miss? oradiff.oracle.com know it!

Oracle Release Analyzer Diff Utility

MOS Account

martin.x.berger@accenture.com

Home

Parameters

Users, Roles, etc

Privileges

Included Fixes

List of Fixes

Total Fixes

Search Fixes

List of CVEs

CPU Explore

Fix Control

Oracle Database Home

Objects

Fixed Objects

Audit

Version Selector

Source Release
19c

Source Patch Level
19.31.0 MRP 5 (260915)

Target Release
19c

Target Patch Level
19.32.0 MRP 3 (260915)

List Mode

Fixes

Added

Removed

Q

Search

Go

Actions

Bug Id	Patch Id	Bug Description
18792392	39472050	Fix for Bug 18792392
19247024	39472050	Fix for Bug 19247024
27707980	39472050	Fix for Bug 27707980
28080178	39472050	Fix for Bug 28080178
28672713	39472050	Fix for Bug 28672713
29531191	39472050	77619 DRG-10900 ERROR ON INDEX_STATS API WITH SPECIAL CHARACTER IDENTIFIERS



The Case

To keep the patching pace and your security score high, you need a strategy and a toolset to achieve. The manual way is no way.

Manual means risky

- Dozens of SQL scripts in the wrong order
- Patch sets, pre-checks, post-checks
- Each instance configured slightly differently

Doesn't scale

- 10 databases = 10× effort & 10× risk
- Weekends, pizza
- Downgrade? Have fun searching.

Out-of-Place Upgrade

For patches in Oracle AI Database 26ai, Oracle recommends using out-of-place patching with gold images to keep up with the patch cycle.

In-Place - the classic

- Patches land directly in the running ORACLE_HOME
- Rollback ≈ restore from backup
- Long downtime, high anxiety

Out-of-Place - the pro move

- New home installed & patched in parallel
- Rollback = switch back to the old home
- Minimal downtime, peaceful sleep

Oracle AutoUpgrade

The game changer - from pre-check to post-processing, fully automated.

What it can do

- A single JAR file, runs with Java runtime
- Hundreds of instances in **one** config file
- Pre-checks, fixups, upgrade & post-tasks in one pass
- Restartable after errors - no duplicate work

What it replaces

- **DBUA** was marked as deprecated with 23ai and is removed in 26ai
- Manual `dbupgrade` or `catctl.pl` calls
- Hand-crafted pre/post-scripts

01

ORACLE_HOME

Building a patched ORACLE_HOME in 3 Steps

Create MOS Keystore

1

AutoUpgrade
-load_password

Keystore
auto-login

Output
keystore ready to use

Download Patches

2

AutoUpgrade
-mode download

MOS API
File download

STAGE DIR
Zips validated

Output
Ready for setup

Create up-to-date home

3

AutoUpgrade
-mode create_home

Installer
Binary install

OPatch
Apply

root.sh

Output
ORACLE_HOME

Autoupgrade Pro-Feature: New ORACLE_HOMEs

Goal: The latest Oracle RDBMS release is downloaded and installed.

1

Get latest autoupgrade

Pull the latest version directly from Oracle. MOS is not required.

2

Create MOS keystore

Store credentials in the auto-login keystore.

3

Get Release Update

Download of the Release Update, OPatch, DBPB, JDK etc.

4

ORACLE_HOME

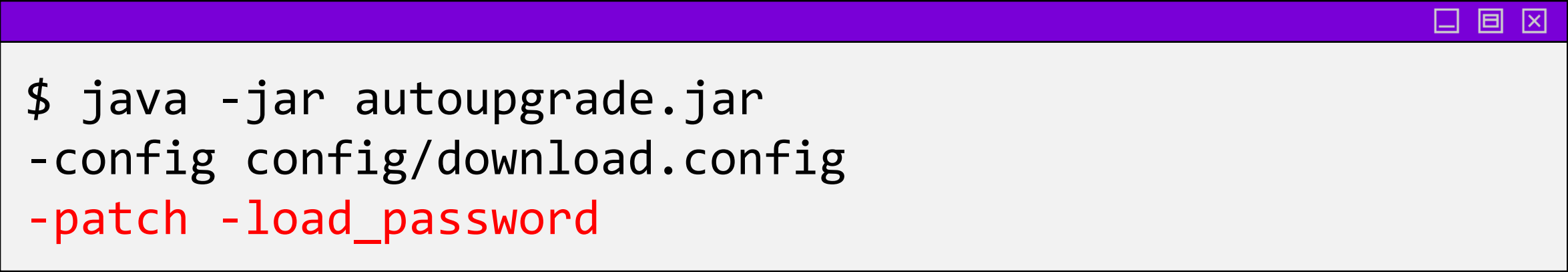
Create new ORACLE_HOME



*Since May, only the latest release is automatically downloaded – currently **19.32** or **26.0.3**, optionally with **Gold Image**.*

Step 2 - Create MOS Keystore

AutoUpgrade needs My Oracle Support credentials to pull patches. Instead of passwords in plain text: an auto-login keystore.

A terminal window with a blue title bar and standard window controls. The command is entered in a monospaced font.

```
$ java -jar autoupgrade.jar  
-config config/download.config  
-patch -load_password
```

What happens

- `-load_password` starts the interactive MOS prompt
- Keystore is stored as **auto-login** - no password needed anymore

Step 3 – Pull Release Update

With the keystore in hand, AutoUpgrade pulls the RU in download mode directly from MOS - fully automated.

```
$ java -jar autoupgrade.jar  
-config config/get-release.config  
-patch -mode download
```

What happens

- AutoUpgrade authenticates against MOS via keystore
- ZIPs land under `patch1.folder`
`target_home=/tmp/dummy_home`

get-release.config

```
patch1.patch=RECOMMENDED  
patch1.target_version=26  
patch1.platform=LINUX.X64  
patch1.folder =/u01/app/stage/patch
```

AutoUpgrade fetches a complete patch set

19c – RECOMMENDED

db_goldimage_19_32_20260929105920.zip

19c Gold Image inklusive RU + MRP

p39222882

OJVM RELEASE UPDATE 19.32.0.0.0

p39657094

DATAPUMP BUNDLE PATCH 19.32.0.0.0

p6880880

OPatch 12.2.0.1.53

autoupgrade.jar

Latest version AutoUpgrade

26ai – RECOMMENDED

p39581612

23.26.3 Gold Image inklusive RU + MRP

p39593097

DATAPUMP BUNDLE PATCH 23.26.3.0.0

p6880880

OPatch 12.2.0.1.53

autoupgrade.jar

Latest version AutoUpgrade

Parameter patch.RECOMMENDED = Gold Image, OJVM (19c), DPBP,MRP + AU

patch1.patch=OEM:24.1

*Additional parameter values to download Oracle Enterprise Manager 13.5/24.1 patchsets and many more. Latest AutoUpgrade version **26.5.260807** required!*

```
-----
Downloading files to /u01/app/oracle/autoupgrade/patch
-----
Oracle Enterprise Manager 24ai Release 1 Update 13 (24.1.0.13) for Oracle Management Service
  File: p39815035_241000_Generic.zip - VALIDATED

Oracle Enterprise Manager 24ai Release 1 Holistic Patch for Oracle Management Service - September 2026
  File: p39984254_241000_Generic.zip - 100%
VALIDATED

Oracle Enterprise Manager 24ai Release 1 Update 13 (24.1.0.13) for Oracle Management Agent
  File: p39815042_241000_Generic.zip - VALIDATED

OMSPatcher patch of version 13.9.24.17.0 for Enterprise Manager 24.1.0.0.0
  File: p19999993_241000_Generic.zip - VALIDATED

AgentPatcher patch of version 13.9.25.3.0 for Enterprise Manager 24.1.0.0.0
  File: p33355570_241000_Generic.zip - VALIDATED
-----
```



Build ORACLE_HOME

A single call: AutoUpgrade installs the binaries and applies the RU or additional patches at the same time. The finished home is ready.

```
$ java -jar autoupgrade.jar  
-config config/create-home.config  
-patch -mode create_home
```

What happens

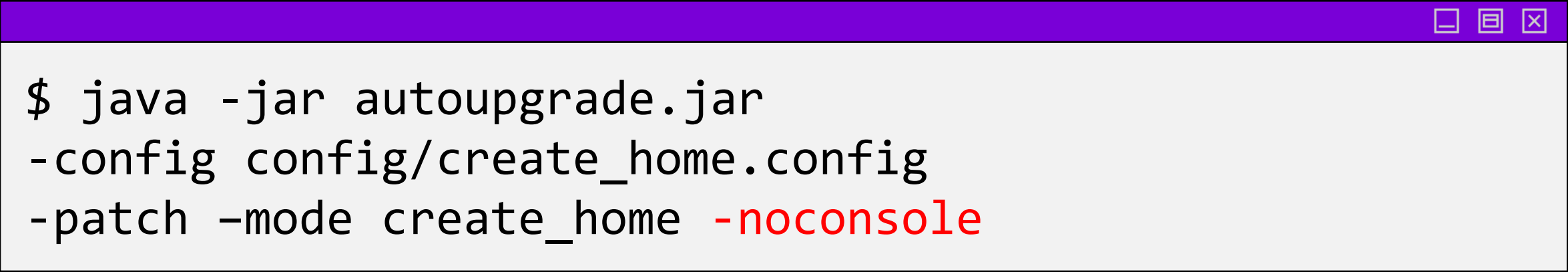
- Install + Patch components in one step
- Afterwards: `orainstRoot.sh` & `root.sh` as root

create-home.config

```
install1.target_home=/u01/app/oracle/product/...  
install1.home_settings.oracle_base=/u01/app/oracle  
install1.home_settings.edition=EE
```

-noconsole - the automation switch

*By default, AutoUpgrade prompts interactively. In a script or Ansible run, this is fatal. **-noconsole** disables all prompts.*



```
$ java -jar autoupgrade.jar  
-config config/create_home.config  
-patch -mode create_home -noconsole
```

What happens

- No user prompts
- No interactive console, no **deploy>** input line
- Output goes directly into the log file

Offline? Yes!

*Target host has no MOS access? Pull the patches **externally** once, provide them via **NFS**, and point to the NFS path in the `create_home` configuration file.*

1

Download externally

Client with internet pulls the package from MOS.

2

Push to NFS

Copy ZIPs to an **NFS** share - e.g.
/srv/oracle-patches/26.0.0/.

3

Create Home

Mount NFS on the target host, set `install1.folder` to the mount path
install1.download=no.



If the files are available locally, the parameter `install1.download=no` must be set.

02

UPGRADE

The 3 Modes of AutoUpgrade

One run - three clean stages. Each callable individually, all automatable together.

1

Analyze

Finds everything that will blow up later without changing anything.

2

Fixups

Proactively repairs the findings from Phase 1. Statistics, parameters, etc.

3

Deploy

The actual upgrade. With parallel workers in the new ORACLE_HOME. With restore point, postchecks, postfixups.

Analyze

Inspects the database without touching it. Delivers a detailed report of all stumbling blocks.

```
$ java -jar autoupgrade.jar  
-config config/upgrade.config  
-patch -mode analyze
```

What happens

- Reads `DBA_*` views & init parameters
- Checks TIMEZONE, COMPATIBLE, APEX, OLAP, Spatial
- Writes HTML & JSON reports

upgrade.config

```
global.autoupg_log_dir=/u01/app/oracle/autoupgrade/logs  
upg1.source_home=/u01/app/oracle/product/19.0.0/dbhome_1  
upg1.target_home=/u01/app/oracle/product/23.0.0/dbhome_1  
upg1.sid=CH01
```

Fixups

AutoUpgrade automatically fixes what the Analyze phase found - proactively, before the actual upgrade.

```
$ java -jar autoupgrade.jar  
-config config/upgrade.config  
-patch -mode fixups
```

What happens

- Re-gather statistics on Dictionary & Fixed-Objects
- Empty the recycle bin
- Remove deprecated init.ora parameters

upgrade.config

```
global.autoupg_log_dir=/u01/app/oracle/autoupgrade/logs  
upg1.source_home=/u01/app/oracle/product/19.0.0/dbhome_1  
upg1.target_home=/u01/app/oracle/product/23.0.0/dbhome_1  
upg1.sid=CH01
```

Deploy

The upgrade job. Now, the database is moved to the new ORACLE_HOME.

```
$ java -jar autoupgrade.jar  
-config config/upgrade.config  
-patch -mode deploy
```

What happens

- Database shutdown in the old home
- Startup UPGRADE in the new home
- Dictionary upgrade of all PDBs

upgrade.config

```
global.autoupg_log_dir=/u01/app/oracle/autoupgrade/logs  
upg1.source_home=/u01/app/oracle/product/19.0.0/dbhome_1  
upg1.target_home=/u01/app/oracle/product/23.0.0/dbhome_1  
upg1.sid=CH01
```

Phase 3 – Example

```
$ cat upg_summary.log

Oracle Database Release 23 Post-Upgrade Status Tool    06-12-2026 11:39:1
Container Database: CH01
[CON_ID: 1 => CDB$ROOT]

Component                               Current      Full      Elapsed Time
Name                                   Status      Version   HH:MM:SS

Oracle Server                           UPGRADED    23.26.2.0.0 00:08:45
JServer JAVA Virtual Machine            UPGRADED    23.26.2.0.0 00:02:04
Oracle XDK                               UPGRADED    23.26.2.0.0 00:00:51
Oracle Database Java Packages            UPGRADED    23.26.2.0.0 00:00:15
OLAP Analytic Workspace                  UPGRADED    23.26.2.0.0 00:00:10
Oracle Label Security                    UPGRADED    23.26.2.0.0 00:00:09
Oracle Database Vault                    UPGRADED    23.26.2.0.0 00:00:33
Oracle Text                              UPGRADED    23.26.2.0.0 00:00:21
Oracle Workspace Manager                  UPGRADED    23.26.2.0.0 00:00:30
Oracle Real Application Clusters          UPGRADED    23.26.2.0.0 00:00:00
Oracle XML Database                      UPGRADED    23.26.2.0.0 00:00:33
Spatial                                  UPGRADED    23.26.2.0.0 00:01:52
Oracle OLAP API                          UPGRADED    23.26.2.0.0 00:00:06
Datapatch                                00:00:14
Final Actions                            00:00:23
Post Upgrade                             00:00:09
```



Remaining tasks 1/2

TIMEZONE VERIFICATION - upg1.timezone_upg=no

```
SQL> select version from v$timezone_file;
```

```
VERSION
-----
      45
```

GATHER DICTIONARY STATISTICS - dictionary_stats_after=no

```
SQL> EXECUTE DBMS_STATS.GATHER_DICTIONARY_STATS;
```

COMPATIBLE PARAMETER - change when 100% no downgrade

```
SQL> show parameter compatible
```

NAME	TYPE	VALUE
compatible	string	19.0.0
noncdb_compatible	boolean	FALSE

Remaining tasks 2/2

DROP RESTORE POINT

SQL> SELECT name, guarantee_flashback_database, time, scn
2 FROM v\$restore_point WHERE guarantee_flashback_database = 'YES';

NAME	GUA	TIME	SCN
AUTOUPGRADE_9212_DBOE_SITE11932000	YES	01-OCT-26 11.41.05.000000000 PM	3183534

Data Guard: Standby-First Patching

The RU goes to the standby first; the old primary follows after a controlled switchover. Downtime: just the switchover seconds.

```
$ java -jar autoupgrade.jar -config config/upgrade.config -mode deploy
```

Why Standby-First at all?

- Primary serves users until the switchover occurs
- Risk isolation: test on standby first, then go live
- Real rollback - old primary is still running
- Oracle allows RU **mixed-binary operation**

What AutoUpgrade does for this

- `-mode deploy` with Data Guard awareness
- Sets Redo Apply off/on
- Restart in the new home, cross-check of binary versions
- Datapatch runs **only once** - at the end on the current primary

03

THE TEAM

Ansible - the buddy who has your back

AutoUpgrade takes care of the database. Ansible handles everything else. IaC at it's best.

The three killer features

- **Agentless** - only SSH & Python on the target
- **Idempotent** - same run = same end state
- **Declarative** - YAML describes the goal, not the path

What Ansible does

- OS setup: LVM, mountpoints, kernel parameters, users
- Software download from My Oracle Support
- ORACLE_HOME installation & patching
- Listener configuration, tnsnames, wallet

Ansible concepts in 60 seconds

Inventory

List of target hosts.

```
[oracle_server]  
db01.internal ansible_host=10.0.0.11
```

Role

Reusable bundle of tasks, templates, variables, and handlers.

Playbook

YAML file with tasks.

```
- hosts: oracle_server  
  
roles:  
  - os  
  - autoupgrade
```

Tags

Selective execution of individual steps -
e.g. only pre-check, only fixups, only
deploy.

```
$ ansible-playbook -i inventory/single playbooks/shared/autoupgrade.yml --tags get_release_update
```

Ansible Playbook Example

```
- name: Run autoupgrade command to create a new oracle home
  ansible.builtin.shell: |
    java -Xms2G -Xmx4G -jar {{ autoupgrade_base_dir }}/autoupgrade.jar \
      -config {{ autoupgrade_base_dir }}/config/create_home.config \
      -patch -mode create_home -noconsole \
      < /dev/null \
      > {{ autoupgrade_base_dir }}/logs/create_home_{{ ansible_date_time.epoch }}.log 2>&1
  become: true
  become_user: "{{ oracle_user }}"
  async: 7200
  poll: 30 # <-- Ansible waits automatically
  register: autoupgrade_result
  args:
    executable: /bin/bash
  tags:
    - online_create_oracle_home
```

Teamwork

Ansible prepares the infrastructure. AutoUpgrade handles the nitty-gritty.

A

Provision

Creates VMs and storage, configures the OS.

> Terraform

B

Prepare

AutoUpgrade creates a new ORACLE_HOME.

> Ansible

C

Upgrade

AutoUpgrade runs the three phases.

> AutoUpgrade

D

Report

Results are stored centrally.

> ORDS + ADB



In this scenario an Oracle 26ai Autonomous Database is used as CMDB.

REST - **RE**presentational **S**tate **T**ransfer

Ansible POSTs server data via HTTP. ORDS (Oracle REST Data Services) writes directly into the Autonomous Database and further actors.

1 Define endpoint

ORDS publishes a REST API directly on a DB table - configured by SQL.

2 Send request

Ansible calls curl: JSON body with hostname, Oracle Home and RU version.

3 Data is stored in ADB

ORDS parses the body and writes it into the table SERVER_INSTANCES.

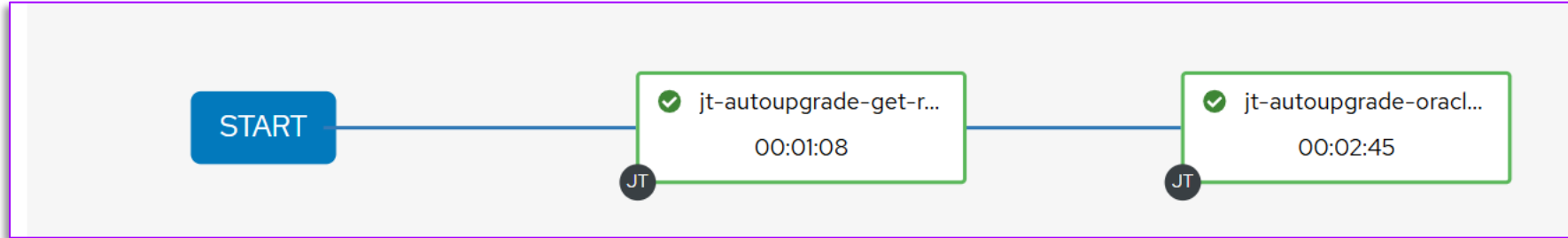
curl example requests

```
--POST

curl -X POST \
  -u "$ORDS_USER:$ORDS_PASS" \
  -H "Content-Type: application/json" \
  -d '{
    "hostname":          "db-server-01",
    "oracle_home":       "/u01/app/oracle/product/19.0.0/dbhome_1",
    "oracle_ru_version": "19.25.0.0.0"
  }' \
  "$ORDS_URL/server_instances/"
```

And two more on the team

OLAM – **Oracle Linux Automation Manager** for automation and the **Oracle Autonomous Database** for tracking. Ok, you can also use PostgreSQL, all you need is **REST** ...



Cloud

Search resources, services, documentation, and Marketplace

Switzerland North (Zurich)

Oracle AI Database

Autonomous AI Databases

EWRHA9NYWOJFI1YN Available Developer

Database actions Database connection Performance Hub More actions

Autonomous AI Database information Tool configuration Backups Key history Disaster recovery Refreshable clones Guided console tour Security Monitoring Work r

General information

Database name EWRHA9NYWOJFI1YN

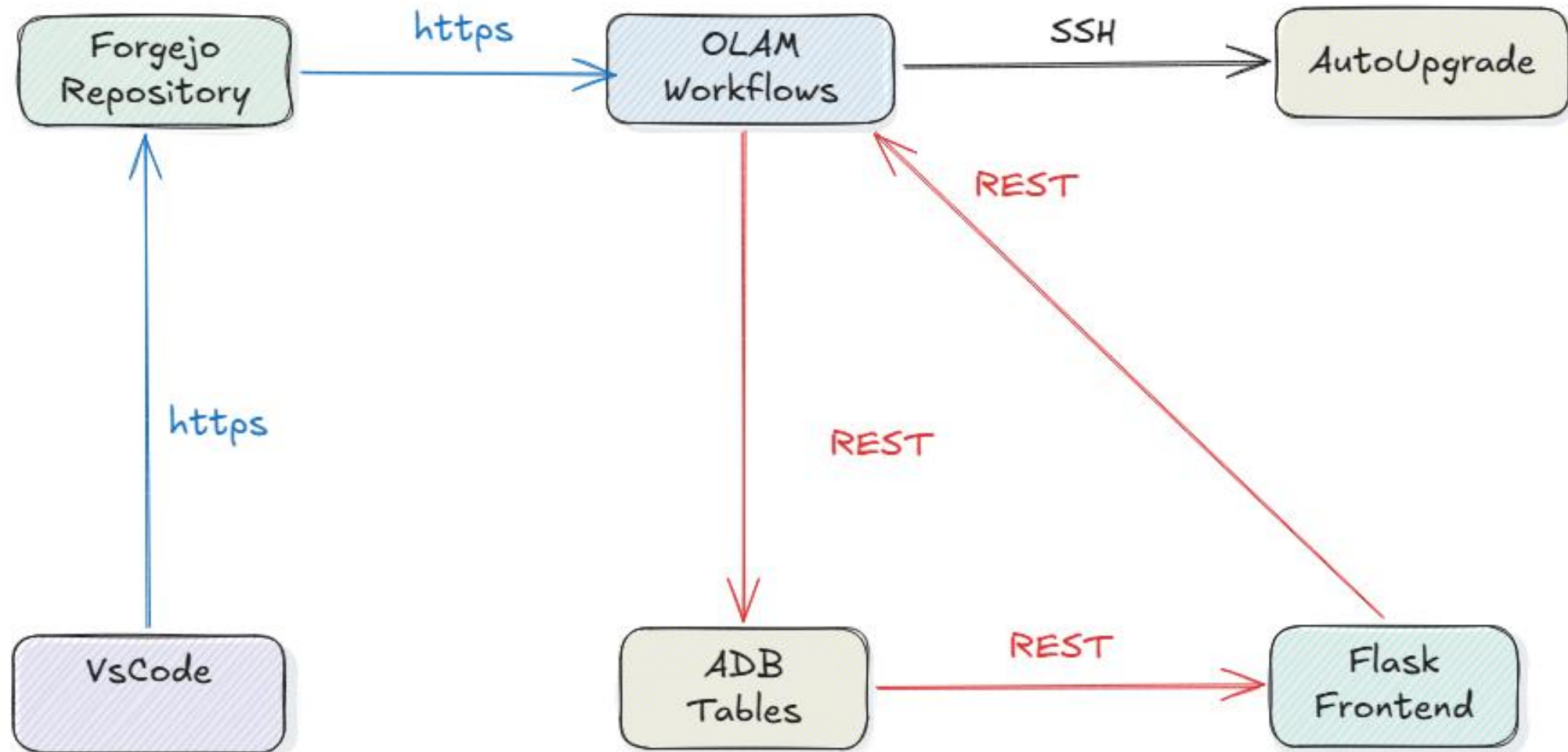
Workload type Transaction

Disaster recovery

Disaster recovery protects your database instance by providing peer databases in a different availability domain or region. [Learn more](#)

Role

Here's how it fits together...



04

DEMO TIME

Key Takeaways

1 **AutoUpgrade is THE standard – DBUA is history**

Officially deprecated as of Oracle 26ai. Switch now, don't wait.

2 **-patch -mode create_home: The underestimated feature**

AutoUpgrade doesn't just upgrade – it builds the new home right away, up to date.

3 **-noconsole is the key to Ansible integration**

No interactive feedback required.

4 **Guarantee Restore Point is your safety net**

Only drop it once it runs stably. Increase COMPATIBLE last.

Questions?

