



11 months with Oracle Autonomous Transaction Processing

Oracle OpenWorld San Francisco. September 16–19, 2019.
Session CAS3039.

<https://indico.cern.ch/event/839681/>

Eric Grancher, head of database services group, CERN IT department

Outline

Introduction to CERN

Introduction, ATP seen from the user perspective

Creation, Web console/CLI/API

Connecting to ATP

SQL and PL/SQL

Managing

11 months experience

Open Days, Oracle Cloud

Conclusion

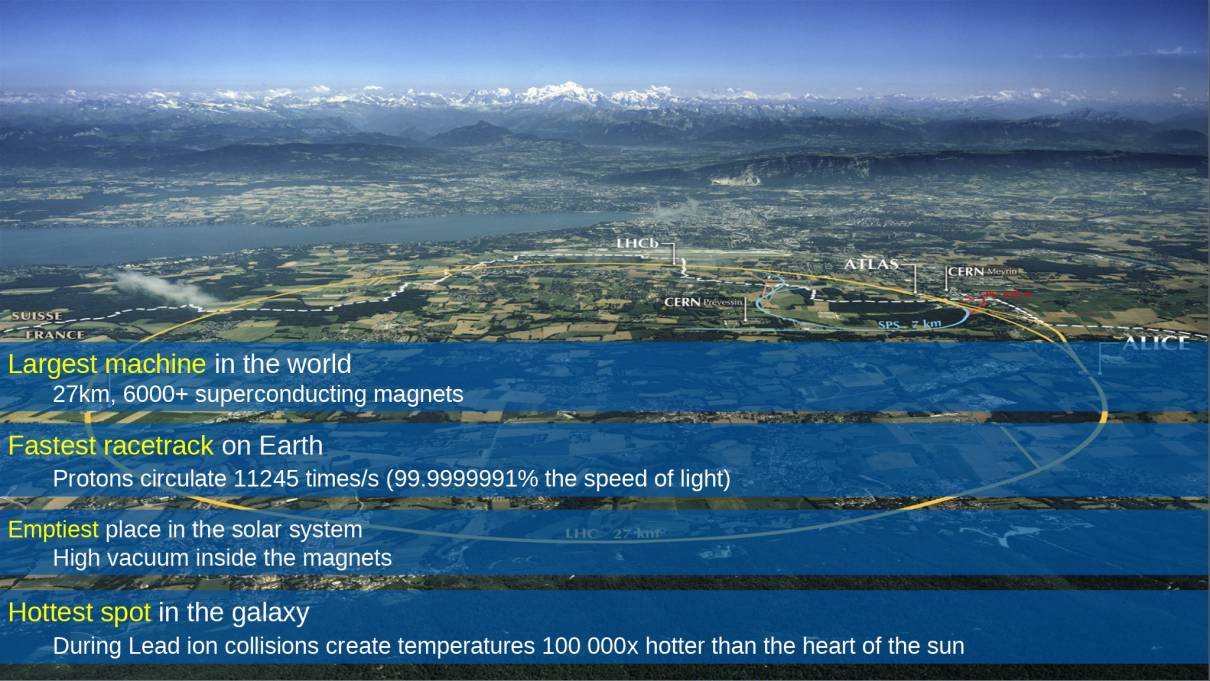
CERN

- CERN - European Council for Nuclear Research
- Founded in 1954 by 12 countries for fundamental physics research in the post-war Europe
- Today 23 members states (see <https://home.cern/about/member-states>) and world-wide collaborations, 2 300 CERN personnel.
- More information at <https://home.cern/about>
- Can be visited, see <https://visit.cern/> (and 14-15 September 2019 CERN Open Days).

Fundamental Research

- What is 95% of the Universe made of?
- Why do particles have mass?
- Why is there no antimatter left in the Universe?
- What was the Universe like, just after "Big Bang"?

$$\begin{aligned}\mathcal{L} = & -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} \\ & + i\bar{\psi}\not{D}\psi + h.c. \\ & + \chi_i y_{ij} \chi_j \phi + h.c. \\ & + |D_\mu \phi|^2 - V(\phi)\end{aligned}$$



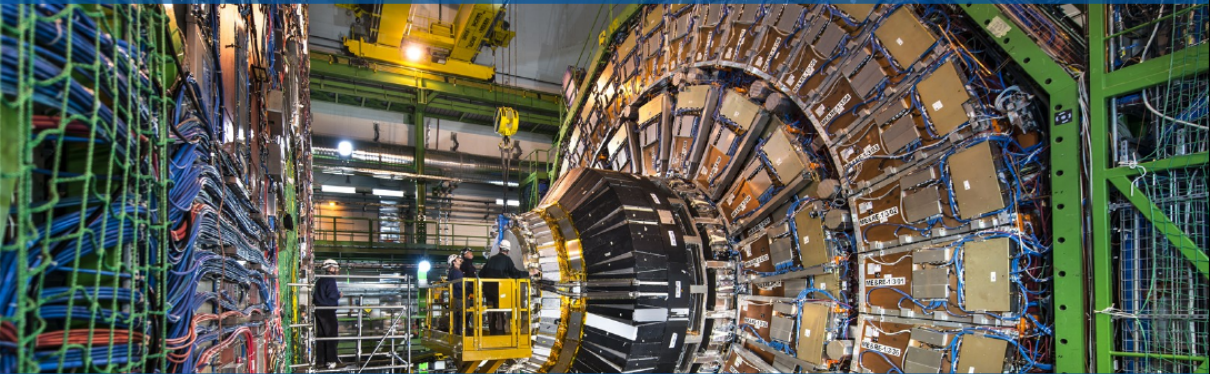
Largest machine in the world
27km, 6000+ superconducting magnets

Fastest racetrack on Earth
Protons circulate 11245 times/s (99.9999991% the speed of light)

Emptiest place in the solar system
High vacuum inside the magnets

Hottest spot in the galaxy
During Lead ion collisions create temperatures 100 000x hotter than the heart of the sun

CMS Detector

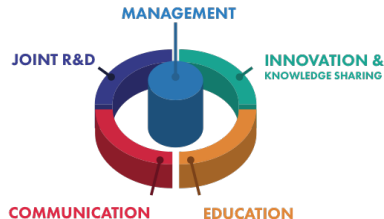


150 Million of sensor
Control and detection sensors

Massive 3D camera
Capturing 40+ million collisions per second
Data rate TB per second

CERN openlab

- Public-private partnership, through which CERN collaborates with leading ICT companies and other research organizations.
- Evaluate and improve state-of-the-art technologies in a challenging environment
- Train the next generation of engineers/researchers.
- Promote education and cultural exchanges.
- Communicate results and reach new audiences.
- Oracle is a member since 2003.



ATP - Introduction 1/3

- A cloud service for OLTP workload (pay as you need, flexible, -automated-scalability)
- All database administration tasks handled by Oracle/automation, no need to intervene (for patches, backup, etc.)
- Implements from the beginning many "good practices" and automation to a very high level
- Implements many security mechanisms, regular fixes, implies some limitations (no OS access for example), online patching
- With specifics in the implementation (automatic scaling, etc.)
- Shares "autonomous database" concepts with ADW (and the two services complement each other for different workloads), see strategy document [1].
This presentation actually uses both ATP and ADW

ATP - Introduction 2/3

- Two ATP services: "shared serverless" (focus of this presentation, 18c) and "dedicated" (available since June 2019)
- Leverages many features of Oracle database (Multitenant, RAC, online patching, resource manager, etc.)
- Presentation and view from the user point of view, cannot indicate how it is implemented (feature), see presentations from Oracle employees on the "how"
- The presentation also gives indications on how to perform some "advanced" operations which many of the ATP users probably do not need (possible if needed!)
- **⚠** ATP is changing fast, see "What's New for ATP" [2] (ex: "Upgrading Always Free Autonomous Databases to Paid Instances")
- Integrated with other Oracle cloud services, for example OMC [3] April 2019

ATP - Introduction 3/3

- Guarantee in term of availability (see ATP-D with DataGuard)
- Implemented on the Exadata platform
- Makes use of the Multitenant (CDB / PDB) architecture where each ATP database is a distinct PDB
- Licensing with "Pay as you go" or "Bring Your Own License" Oracle DB license. (1-16: "For each supported Processor license of Oracle Database Enterprise Edition plus Options: Multitenant, You may activate up to 2 OCPUs of the BYOL Cloud Service.", bigger requires RAC)
- "Autonomous Database subscription includes many management, testing, and security capabilities that previously had to be licensed separately, including: Data Encryption, Diagnostics Pack, Tuning Pack, ..."

Via Web console

- One selects the "compartment" and key characteristics
- OCPU and memory are linked
- Storage
- Licensing decision of the Oracle database, can be changed anytime later
- Shared versus dedicated
- Administration account (then connect to admin@dbname_tp)

Oracle Cloud Infrastructure x +

← → ↻ console.eu-frankfurt-1.oraclecloud.com/db/atp/create ☆ Incognito

ORACLE Cloud eu-zurich-1

Create Autonomous Database [Help](#) [Close](#)

Provide basic information for the Autonomous Database

Choose a compartment

cernprod1 (root)

Display name

DB 201909062326

Database name

DB201909062326

The name must contain only letters and numbers, starting with a letter. Maximum of 14 characters.

Choose a workload type

Data Warehouse

[Terms of Use and Privacy](#) Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Demo1: creation an ATP using the Web console (2min45s)

Via CLI or programmatically 1/3

- Set the environment and copy keys to "API Keys (Add Public Key)"

```
bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-  
cli/master/scripts/install/install.sh)"  
oci setup config  
cat ~/.oci/oci_api_key_public.pem
```

-  Federated accounts, using keys

Via CLI or programmatically 2/3

```
$ oci db autonomous-database create --admin-password $P --compartment  
  ↪ -id $C --cpu-core-count 1 --data-storage-size-in-tbs 1 --db-  
  ↪ name T1 --db-workload OLTP --is-auto-scaling-enabled false --  
  ↪ license-model BRING_YOUR_OWN_LICENSE
```

- SDKs (Java, Python [4], Ruby and Go), CLI [5] and REST APIs
- Integration with Eclipse, Terraform, Ansible, Jenkins, ... see [6]
- Syntax is common for the two autonomous database flavours
- (as always) Check the defaults (e.g. license-model)
- The command line tool "oci"[4]

Via CLI or programmatically 3/3

- ⚠ Using "wait-for-state" is very useful
- "If timeout is reached, a return code of 2 is returned. For any other error, a return code of 1 is returned."

```
$ oci ... --max-wait-seconds 1200  
    ↪ --wait-for-state AVAILABLE  
$ echo $?  
0
```

`--max-wait-seconds [integer]`

The maximum time to wait for the resource to reach the lifecycle state defined by `--wait-for-state`. Defaults to 1200 seconds.

`--wait-for-state [text]`

This operation creates, modifies or deletes a resource that has a defined lifecycle state. Specify this option to perform the action and then wait until the resource reaches a given lifecycle state. If timeout is reached, a return code of 2 is returned. For any other error, a return code of 1 is returned.

Accepted values are:

AVAILABLE, AVAILABLE_NEEDS_ATTENTION, BACKUP_IN_PROGRESS, MAINTENANCE_IN_PROGRESS, PROVISIONING, RESTORE_IN_PROGRESS

`--wait-interval-seconds [integer]`

Check every `--wait-interval-seconds` to see whether the resource has reached the lifecycle state defined by `--wait-for-state`. Defaults to 30 seconds.

```
2. [screen 2: bash] opc@egocivm1:~ (ssh_from_elcapit)
Projector (ssh_from_elcapit) %1 Default (bash) %2

-- Python version 3.6.8 okay.

==> In what directory would you like to place the install? (leave blank to use '/home/opc/lib/oracle-cli'):
-- Creating directory '/home/opc/lib/oracle-cli'.
-- We will install at '/home/opc/lib/oracle-cli'.

==> In what directory would you like to place the 'oci' executable? (leave blank to use '/home/opc/bin'):
-- Creating directory '/home/opc/bin'.
-- The executable will be in '/home/opc/bin'.

==> In what directory would you like to place the OCI scripts? (leave blank to use '/home/opc/bin/oci-cli-scripts'):
-- Creating directory '/home/opc/bin/oci-cli-scripts'.
-- The scripts will be in '/home/opc/bin/oci-cli-scripts'.

==> Currently supported optional packages are: ['db (will install cx_Oracle)']
What optional CLI packages would you like to be installed (comma separated names; press enter if you don't need any optional packages)??:
-- The optional packages installed will be ''.
-- Downloading virtualenv package from https://github.com/pypa/virtualenv/archive/15.0.0.tar.gz.
-- Downloaded virtualenv package to /tmp/tmpfgr_q1ss/15.0.0.tar.gz.
-- Checksum of /tmp/tmpfgr_q1ss/15.0.0.tar.gz OK.
-- Extracting '/tmp/tmpfgr_q1ss/15.0.0.tar.gz' to '/tmp/tmpfgr_q1ss'.
-- Executing: ['/usr/bin/python3', 'virtualenv.py', '--python', '/usr/bin/python3', '/home/opc/lib/oracle-cli']
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/opc/lib/oracle-cli/bin/python3
Also creating executable in /home/opc/lib/oracle-cli/bin/python
Installing setuptools, pip, wheel...
```

Demo2: CLI installation and configuration, creation of ATP from command line

Connecting - Wallet

One has to download a wallet, either from the Web console or using the OCI CLI

```
$ oci db autonomous-database generate-wallet --autonomous-database-id  
  ↳ $D --password 1uD8jadfhf --file wN2.zip  
Downloading file [#####] 100%  
$ unzip -t wN2.zip  
Archive: wN2.zip  
  testing: cwallet.sso OK  
  testing: tnsnames.ora OK ...  
No errors detected in compressed data of wN2.zip.
```

sqlcl / SQL Developer / others

- SQLcl and SQL Developer can take the Wallet zip file directly as input
- **⚠** Even if taken from a given ATP, services from all databases in the tenancy are listed
- For other clients, one needs to unzip the client credentials (wallet files) and edit the sqlnet.ora file to reflect the path

```
set cloudconfig ../../Wallet_N2.zip
```

Client versions

- **⚠** Relatively recent client versions should be used to connect
- Works for example with 12.1, 18.3, 19.3
- Below example with the OTN 11.2.0.4 Linux x86-64 client

```
$ ./instantclient_11_2/sqlplus /nolog
SQL*Plus: Release 11.2.0.4.0 Production on Sun Sep 8 18:18:48 2019
Copyright (c) 1982, 2013, Oracle. All rights reserved.
SQL> connect a/a@n2_tp
ERROR:
ORA-28864: SSL connection closed gracefully
```

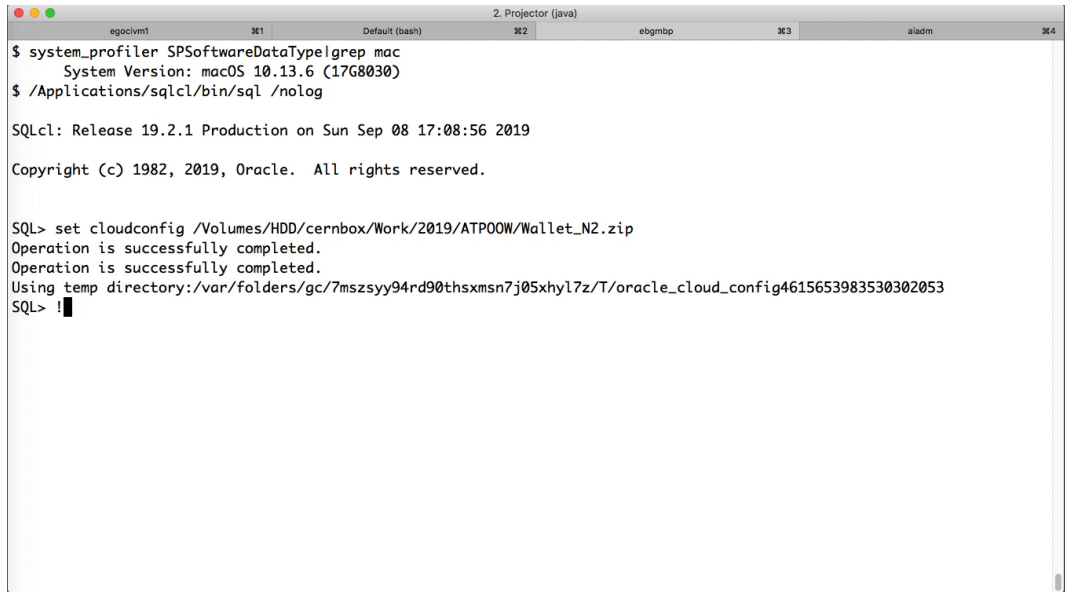

Network

- One should restrict the IPs from which one can connect. It is described at [7]
- One can list individual IP addresses or classless prefixes (RFC 4632, see [8]) like 172.16.0.0/16)

```
$ cat MyIPs.json  
[ "172.16.0.0/16" ]  
$ oci db autonomous-database update --autonomous-database-id $D --  
  ↪ whitelisted-ips file://MyIPs.json --max-wait-seconds 1200 --  
  ↪ wait-for-state AVAILABLE --wait-interval-seconds 2
```

Services

- Several services per ATP are defined
- For the applications, ATPName_TP should be used as a default
- **⚠** ATPName_MEDIUM and ATPName_HIGH should not be used for transactional applications as they run by default all operations in parallel. See Franck Pachot ([9]) or Connor McDonald ([10])



A terminal window titled "2. Projector (java)" with tabs for "egocivm1", "Default (bash)", "ebgmbp", and "aladm". The terminal shows the following commands and output:

```
$ system_profiler SPSoftwareDataType | grep mac
  System Version: macOS 10.13.6 (17G8030)
$ /Applications/sqlcl/bin/sql /nolog

SQLcl: Release 19.2.1 Production on Sun Sep 08 17:08:56 2019

Copyright (c) 1982, 2019, Oracle. All rights reserved.

SQL> set cloudconfig /Volumes/HDD/cernbox/Work/2019/ATP00W/Wallet_N2.zip
Operation is successfully completed.
Operation is successfully completed.
Using temp directory:/var/folders/gc/7mszsy94rd90thxmsn7j05xhyl7z/T/oracle_cloud_config4615653983530302053
SQL> !
```

Demo3: connection with SQLcl, modification of network ACL and validation, change of CPU count, parallel DMLs and service tp vs high

SQL and PL/SQL as entry points

- SQL Developer provides many tools. (was announced that EM will become available for ATP)
- Independence of the version when calling packages rather than using tools (expdb, impdb), scripts (awrrpt.sql, etc.)
- Call from SQL*Plus, SQLcl, a program, DBMS_SCHEDULER, etc.
- DBMS_CLOUD package provides procedures to copy files from / to object storage as well as manage files on the DATA_PUMP_DIR directory
- DBMS_CLOUD_ADMIN provides procedures to manage database links and quotas

```
SELECT object_name,bytes FROM DBMS_CLOUD.LIST_FILES('DATA_PUMP_DIR');
```

Example AWR

```
-- DBID, Instance, snap_id (beginning), snap_id (end)
set serveroutput on
set verify off
set lines 32767
set pages 50000
spool &5
SELECT OUTPUT FROM TABLE(DBMS_WORKLOAD_REPOSITORY.awr_report_html(&1,
    ↪ &2, &3, &4));
spool off
```

Example expdb / impdb (1/4)

DECLARE

```
h1 NUMBER; -- Data Pump job handle
job_state VARCHAR2(30); -- To keep track of job state
sts ku$_Status; -- The status object returned by get_status
file VARCHAR2(30); -- file prefix for the export dmp and log file
ostorage VARCHAR2(130); -- object storage url (Ending with a /)
```

BEGIN

```
file := 'ATPCAN2_' || to_char(sysdate, 'YYYYMMDD-HH24MISS');
ostorage := 'https://objectstorage.eu-frankfurt-1.oraclecloud.com/p
    ↪ /xxx/n/yyy/b/ATPstorage/o/';
```

Example expdb / impdb (2/4)

```
h1 := DBMS_DATAPUMP.OPEN(operation=>'EXPORT',job_mode=>'SCHEMA');
DBMS_DATAPUMP.ADD_FILE(handle=>h1,filename=>file||'.log',directory
    ↳ =>'DATA_PUMP_DIR',filetype => dbms_datapump.
    ↳ ku$_file_type_log_file,reusefile=>1);
DBMS_DATAPUMP.ADD_FILE(handle=>h1,filename=>file||'.dmp',directory
    ↳ =>'DATA_PUMP_DIR',filesize=>'4G',filetype => dbms_datapump.
    ↳ ku$_file_type_dump_file,reusefile=>1);
-- A metadata filter is used to specify the schema that will be
    ↳ exported.
DBMS_DATAPUMP.METADATA_FILTER(h1,'SCHEMA_EXPR','IN (('CANUSER'))');
```

Example expdb / impdb (3/4)

```
DBMS_DATAPUMP.SET_PARAMETER(h1,'COMPRESSION','ALL');
DBMS_DATAPUMP.START_JOB(h1);
job_state := 'UNDEFINED';
while (job_state != 'COMPLETED') and (job_state != 'STOPPED') loop
    dbms_datapump.get_status(h1,
        dbms_datapump.ku$_status_job_error +
        dbms_datapump.ku$_status_job_status +
        dbms_datapump.ku$_status_wip,-1,job_state,sts);
end loop;
```


Example expdb / impdb (4/4)

```
dbms_datapump.detach(h1);  
DBMS_CLOUD.PUT_OBJECT('DEF_CRED_NAME_EG',ostorage||file||'.dmp', '  
    ↪ DATA_PUMP_DIR',file||'.dmp');  
DBMS_CLOUD.PUT_OBJECT('DEF_CRED_NAME_EG',ostorage||file||'.log', '  
    ↪ DATA_PUMP_DIR',file||'.log');  
END;  
/
```

Thanks to the documentation with examples [11]

Organisation - lessons learnt

- Can use multiple "regions", deploying to any data center has the same price
- Federated identity very useful: control who has access, avoid multiple accounts, SSO. Federated accounts with local rights, local OCI users for service accounts
- **⚠** Security of the accounts able to spend a lot of credit, alerts
- Organisation with compartments is essential
- In many cases, applying quotas is interesting. (not deployed for us)
- **⚠** Given current implementation: use compartment names which makes immediately sense. For example /ProjectOD-Prod and not /ProjectOD/Prod
- Define policy as "high" as possible with ACL at compartment level
- Accounting history is limited to 6 months, we extract a copy for longer term analysis

Credit: Arash Khodabandeh

Backup and restore

- Export a copy outside the database is probably not needed but can anyway be interesting (common mode risk, organisation data, ...) just like on-premises multiple data copy protection
- A copy to Oracle "Archive Storage": 0.0026 US\$ per GB per month. (example: 10TB compressed, monthly export kept for one year, 312 US\$ per year, much better -cost, common mode- than leaving on DATA_PUMP_DIR)
- Ability to restore to a copy/clone would be really useful (announced)
- ⚠ "The restore operation also restores the DATA_PUMP_DIR directory and user defined directories to the timestamp you specified for the restore; files that were created after that timestamp would be lost"
- ⚠ Backups (incl. "manual backups") cannot be restored beyond 60 days

Notification

Via email, Slack, PagerDuty and/or HTTPS (custom)

Create Rule [Help](#)

Event Matching

Rule Conditions
Limit the events that trigger actions by defining conditions based on event types, attributes and filter tags. [Learn more](#)

SERVICE NAME	EVENT TYPE
Event Type ▾	Database service ▾
	Autonomous Container Database - Maintenance Begin ✕
	Autonomous Container Database - Restore Begin ✕
	+ Add Condition

Event Matching Logic
MATCH event WHERE (
event.type EQUALS ANY OF (
com.oraclecloud.databaseservice.autono...
com.oraclecloud.databaseservice.autono...
)
)

Actions

Actions trigger for the specified event conditions. [Learn more](#)

ACTION TYPE	NOTIFICATIONS COMPARTMENT	TOPIC
Notifications ▾	ATPOOWEG ▾	testNotificationATPCAN2 ▾
		+ Add Action

[Advanced options](#)

[Test Rule](#) [Create Rule](#) [Cancel](#)

Using ATP 1/2

- Use as a "standard" Oracle database at the SQL and PL/SQL level, a number of options are ignored which makes it easier for compatibility
- With some restrictions: Java, no OS access, etc.
- The data tablespace is DATA. The database character set is Unicode AL32UTF8
- Timezone is UTC by default
- All described in Appendix B "ATP for Experienced Oracle Database Users" [12]
- (looking forward to 19c Automatic Indexing feature)

Using ATP 2/2

- HCC compression can be set if one needs
- ⚠ One has to let the system organise, for example when importing data

```
impdp ... TRANSFORM=TABLE_COMPRESSION_CLAUSE:NONE
```

SQL 1/2

```
SQL> create tablespace data2;  
create tablespace data2  
*  
ERROR at line 1:  
ORA-01031: insufficient privileges
```

```
SQL> create user atpcan4 default tablespace sysaux;
```

User created.

```
SQL> select default_tablespace from dba_users where username='ATPCAN4';
```

```
DEFAULT_TABLESPACE
```

```
-----
```

```
DATA
```

```
SQL> alter user atpcan4 quota 20g on data;
```

User altered.

SQL 2/2

```
SQL> create table atpcan4.at as select * from all_tables;
```

Table created.

```
SQL> create table atpcan4.athbc (a1 number) COLUMN STORE COMPRESS FOR QUERY HIGH;
```

Table created.

```
SQL> select table_name,tablespace_name,compress_for from dba_tables where owner='ATPCAN4';
```

TABLE_NAME	TABLESPACE_NAME	COMPRESS_FOR
ATHBC	DATA	QUERY HIGH
AT	DATA	

Patches 1/2

- Querying DBA_REGISTRY_SQLPATCH, one can see (some) of the patches being deployed.

Patches 2/2

```
SQL> select to_char(min(capture_date), 'YYYYMMDD-HH24MISS') min_capture, description
from ATPCAN2.LOG_PATCH group by description order by 1;
```

MIN_CAPTURE	DESCRIPTION
20190504-080730	ADW Bundle Patch : 18.4.0.0.1 (29463850)
20190511-093002	ADW Bundle Patch : 18.4.0.0.1 (29612455)
20190511-093002	ADW Bundle Patch : 18.4.0.0.1 (29672289)
20190527-171003	ADW Bundle Patch : 18.4.0.0.1 (29798022)
20190608-150002	ADW Bundle Patch : 18.4.0.0.1 (29851765)
20190622-175001	ADW Bundle Patch : 18.4.0.0.1 (29908908)
20190707-003002	ADW Bundle Patch : 18.4.0.0.1 (29968420)
20190721-091001	ADW Bundle Patch : 18.4.0.0.1 (30047276)
20190804-155002	ADW Bundle Patch : 18.4.0.0.1 (30104464)
20190819-055002	ADW Bundle Patch : 18.4.0.0.1 (30167754)
20190901-042002	ADW Bundle Patch : 18.4.0.0.1 (30221697)

11 rows selected.

Change of node 1/2

- Querying V\$INSTANCE, one can observe on which instance the workload is running.

```
INSTANCE_NUMBER, INSTANCE_NAME, VERSION, STARTUP_TIME FROM  
  ↪ V$INSTANCE
```

Change of node 2/2

```
SQL> select to_char(d,'Day') DoW, to_char(d,'YYYYMMDD') d,instance_name from (select d,instance_name,
FIRST_VALUE(instance_name) OVER (ORDER BY d,instance_name ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) p_instance_name
from (select distinct trunc(dbserver_date) d,instance_name from ATPCAN2.log_up order by 1,2))
where p_instance_name<>instance_name;
```

DOW	D	INSTANCE_NAME
Saturday	20190511	e7e1pod3
Saturday	20190511	e7e1pod8
Saturday	20190525	e7e1pod1
Saturday	20190525	e7e1pod8
Sunday	20190526	e7e1pod1
Friday	20190531	e7e1pod2
Friday	20190531	e7e1pod3
Saturday	20190608	e7e1pod4
Saturday	20190622	e7e1pod5
Saturday	20190706	e7e1pod6
Saturday	20190706	e7e1pod7
Saturday	20190706	e7e1pod8
Saturday	20190720	e7e1pod1
Saturday	20190720	e7e1pod8
Sunday	20190721	e7e1pod1
Sunday	20190804	e7e1pod2
Sunday	20190804	e7e1pod3
Saturday	20190817	e7e1pod4
Saturday	20190831	e7e1pod5

Investigating availability

- Flow of connections to the dbname_tp
- Each connection loops and insert rows in a table
- Identifying any gaps
- Making the sum of the time of the gaps

CERN Open Days 2019 - introduction

- Open Days 2019 is a week-end free and open for the public to visit CERN
- 89.5k persons maximum registered.
<https://opendays.cern/>
- Multiple events and activities each with limited capacity
- Past Open Days indicated good reservation system is essential
- We developed an application capable of handling functional and potential high-load requirements



Credit: <https://www.instagram.com/p/B2eHHgJI9mB/>

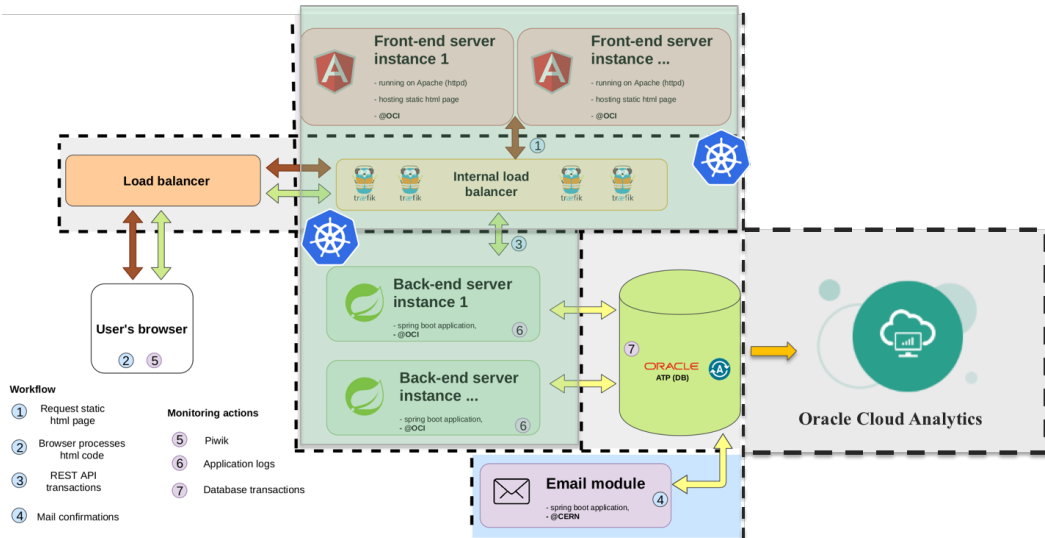
CERN Open Days 2019 - Cloud

Decided to deploy on Oracle Cloud PaaS services

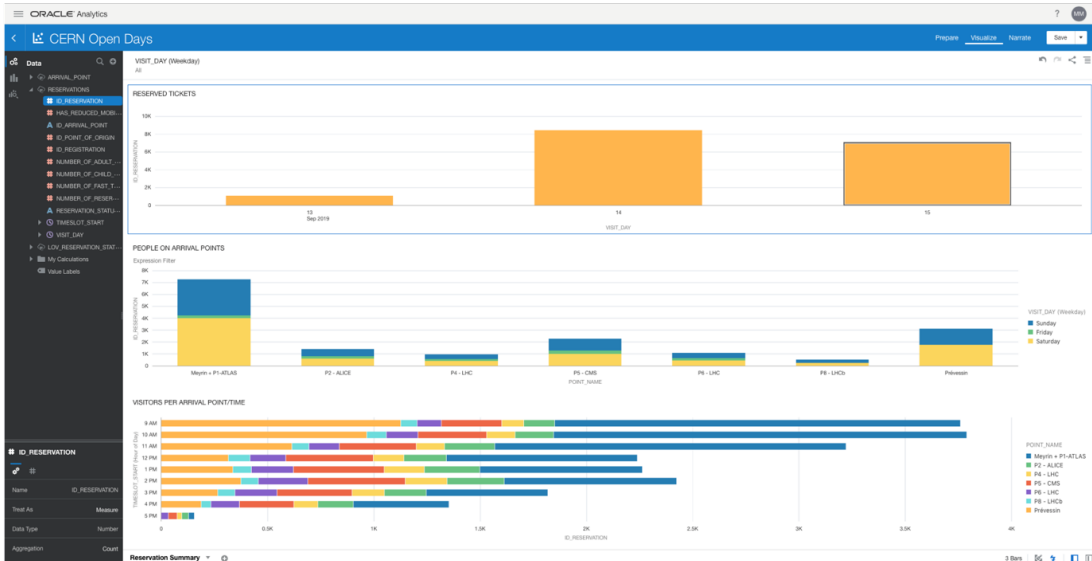
- to be able to deploy at the needed scale
- to save time and benefit from automation using cloud services when possible
- Load balancer aaS, OCI Container Engine for Kubernetes, Autonomous Transaction Processing services, Oracle Management Cloud, Oracle Analytics Cloud
- to ease integration with other interfaces (on site organization) with Oracle database as a common repository
- to gain time and avoid having to deploy new services for the event

CERN Open Days 2019 - experience

- Deployment architecture designed using Oracle cloud managed PaaS services
- This has helped to focus on development, UI, scalability, and architecture
- Easy to scale resources (up/down)
- Stable and highly available
- Simple and easy configurable
- Updates applied (User/Developer/DevOps) transparently without noticeable downtime
- Support of rapid and agile development
- Scalability tests have helped to discover a number of configuration issues, which have been solved before opening to production, no scalability problem observed on the production environment

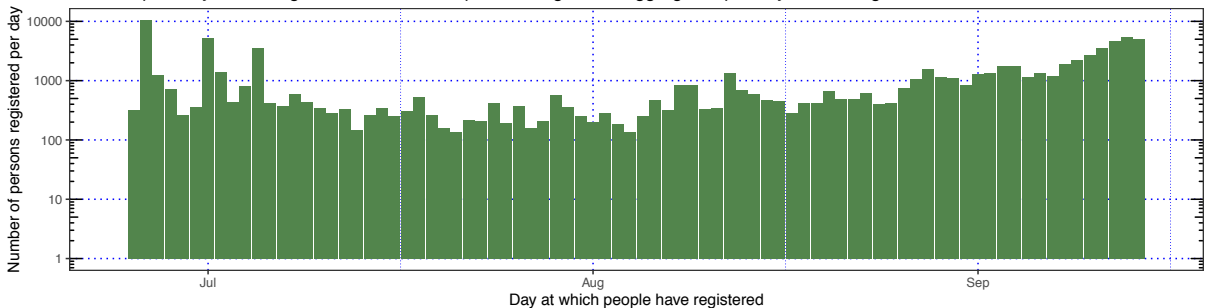


Credit: Viktor Kozlovsky

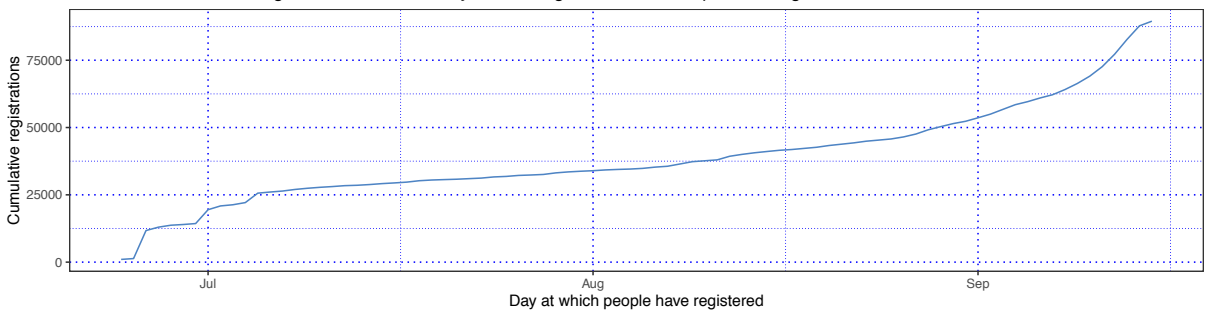


Credit: Manuel Martín Márquez

CERN Open Days 2019 registration: number of persons registered aggregated per day of their registration



Cumulative number of registrants versus the day of their registration: 89496 persons registered



Conclusion

Some takeaways 1/2

- Managed PaaS cloud services present a significant change / opportunity in the way to do operations. Will help to focus on what really matters, in our case research. Fast and flexible deployments
- Automation frees time from some of the routine tasks (no prior expertise, no constant check needed on some of the parts)
- Can help to run cloud-native new services or move "off-premises" services
- Networking is a key element which requires careful planning
- "Serverless / shared" ATP proven to be stable, evolving fast
- Open Days registration: no platform downtime even if Oracle has of course updated the platform during this period of time (mid July security patches, notification received for Cloud Engine interventions on the back-plane)

Some takeaways 2/2

- ATP, even if it is a "recent" solution, has proven very stable in the past 11 months for us (builds on components available or developing for years).
Control plane with diverse interfaces
- ATP is changing fast, integration with Oracle cloud (notification, etc.) is important
- Looking forward to 19c "shared/serverless", including migration path implementations
- Have not tested / deployed "dedicated", appears as the strong solution for networking separation, higher control (patches for example) and "not sharing"
- Enterprise Manager and data management

Thank you!

Questions, suggestions most welcome, now or via email.

Thanks to my CERN colleagues.

Thanks for the support of many at Oracle (Cristobal Pedregal-Martin, Pauline Mahrer, Vincent Leocorbo, Sebastian Solbach and others).

Do not hesitate to contact me at eric.grancher@cern.ch.

Slides available at: <https://indico.cern.ch/event/839681/>

References (I)

- [1] Oracle. White paper: Oracle Autonomous Database Strategy.
<https://www.oracle.com/a/ocom/docs/database/oracle-autonomous-database-strategy-wp.pdf>
- [2] ———. What's New for Autonomous Transaction Processing.
<https://docs.oracle.com/en/cloud/paas/atp-cloud/atpwn/index.html>
- [3] ———. What's New in Oracle Management Cloud.
<https://docs.oracle.com/en/cloud/paas/management-cloud/omcwn/index.html>
- [4] ———. Oracle Cloud Infrastructure Python SDK. <https://oracle-cloud-infrastructure-python-sdk.readthedocs.io/en/latest/index.html>
- [5] ———. Oracle Cloud Infrastructure CLI Command Reference.
https://docs.cloud.oracle.com/iaas/tools/oci-cli/latest/oci_cli_docs/

References (II)

- [6] ——. Developer Tools, Other Tools and Plug-ins.
<https://docs.cloud.oracle.com/iaas/Content/API/Concepts/devopstools.htm>
- [7] K. Laker. How To Make Your Autonomous Database Even More Secure Using Network ACLs. <https://blogs.oracle.com/datawarehousing/how-to-make-your-autonomous-database-even-more-secure-using-network-acls-v>
- [8] V. Fuller and T. Li. (2006, Aug.) Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632.
<https://rfc-editor.org/rfc/rfc4632.txt>
- [9] F. Pachot. Oracle ATP: MEDIUM and HIGH services are not for OLTP.
<https://medium.com/@FranckPachot/oracle-atp-medium-and-high-services-are-not-for-oltp-5a9281b68d72>

References (III)

- [10] C. McDonald. Autonomous Transaction Processing – your slice of the pie. <https://connor-mcdonald.com/2019/08/07/autonomous-transaction-processing-your-slice-of-the-pie/>
- [11] Oracle. Oracle Data Pump API. https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/using-oracle_datapump-api.html
- [12] ——. Appendix B: Autonomous Transaction Processing for Experienced Oracle Database Users. <https://docs.oracle.com/en/cloud/paas/atp-cloud/atpug/experienced-database-users.html>

