

MIGRATING 200+ PDBS FROM ON-PREMISES EXADATA TO EXACC USING ZDM

Learning Session

Presented by Maxime Dumont and Simon Tardif

October 2022

Direction des technologies
de l'information



Oracle Zero Downtime Migration (ZDM) - How it works



Downloadable
Binary

Light
(800 MB)



Migration
Orchestrator

ZDM
Automates & Coordinates
all actions



Customizable
Workflows

User and case
specific customization via
Response File

Oracle Zero Downtime Migration (ZDM) - Benefits



Simple



End-to-End
Automation



Leverages Oracle
MAA Best Practices



Free

Oracle Zero Downtime Migration (ZDM) - Overview

Simple

Single-button approach



Comprehensive

MAA Compliant

Resumable

Dry-run option

Customizable User Actions

Automated Encryption

Fleet Scale

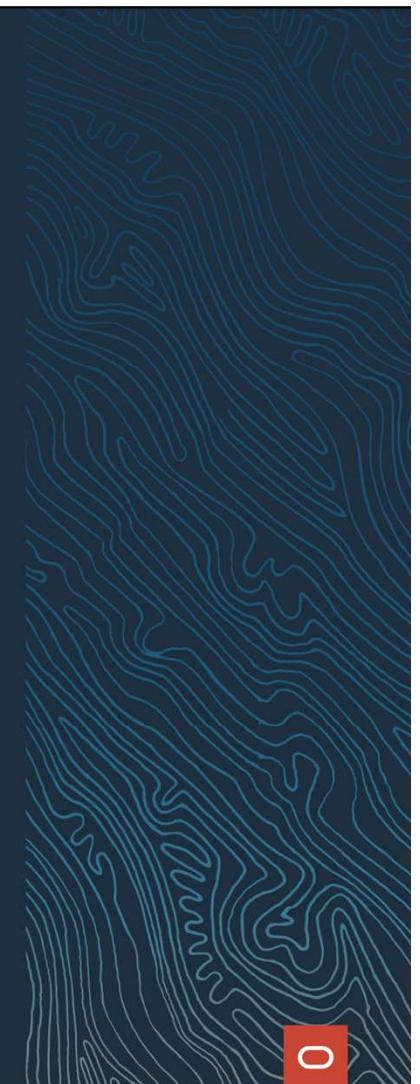
Centralized

Scheduled Operations

Audit Trail

Migrations in Parallel

Jobs Framework



Oracle Zero Downtime Migration (ZDM) Support

Source Databases



SE & EE
On-Premises &
3rd Party Clouds



11.2.0.4, 12.1, 12.2, 18c, 19c, 21c

Source Platforms



ORACLE
Solaris

AIX

Supported Targets



Oracle Base
Database Service



Oracle Autonomous
Database



ExaDB-Cloud@Customer,
ExaDB-D,
Exadata On-Premises



Oracle Zero Downtime Migration – Physical Migration

Physical Offline Migration Workflow



Physical Online Migration Workflow



Oracle Zero Downtime Migration – Physical Migration

Source
Versions



Target
Versions



✓
Non-CDB
to
PDB Ok

Target
Platforms



Oracle Base
Database Service

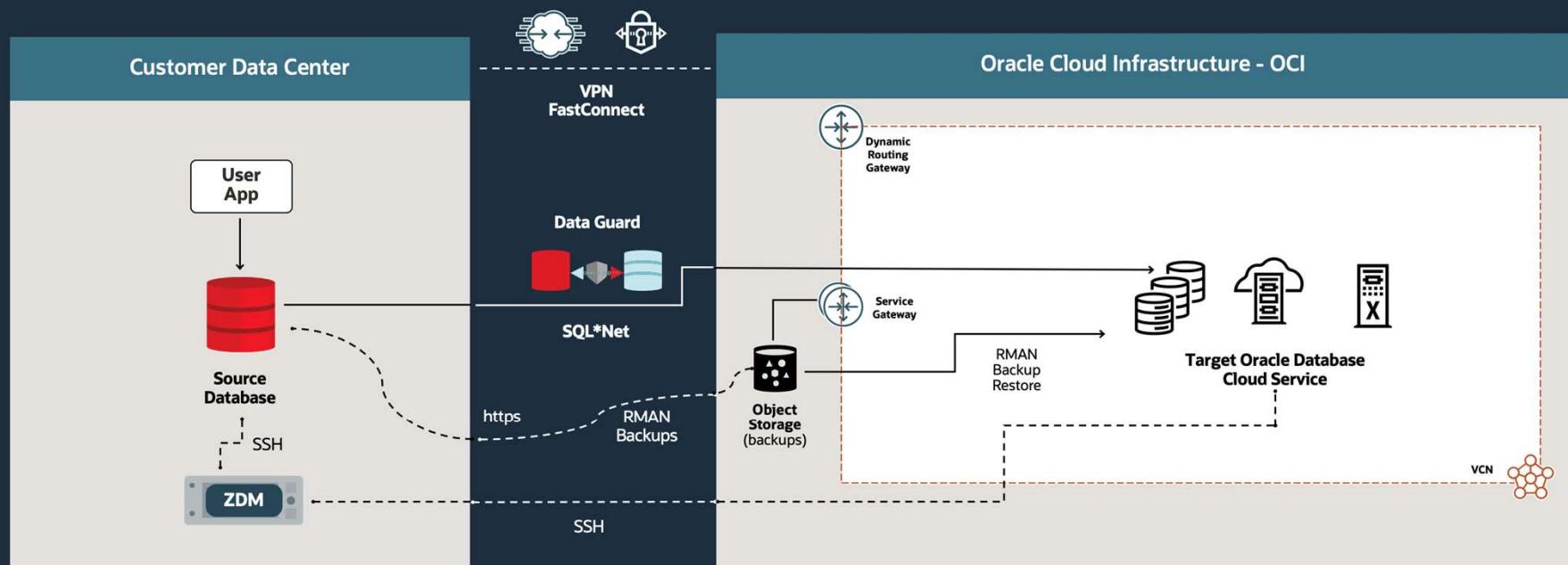


ExaDB-Cloud@Customer,
ExaDB-D,
Exadata On-Premises



Technical Architecture

ZDM Physical Online Database Migration



Oracle Zero Downtime Migration – Logical Migration

Logical Offline
Migration Workflow



Logical Online
Migration Workflow



Oracle Zero Downtime Migration – Logical Migration

Source
Versions



Target
Versions



Target
Platforms



Oracle Base
Database Service



ExaDB-Cloud@Customer,
ExaDB-D,
Exadata On-Premises

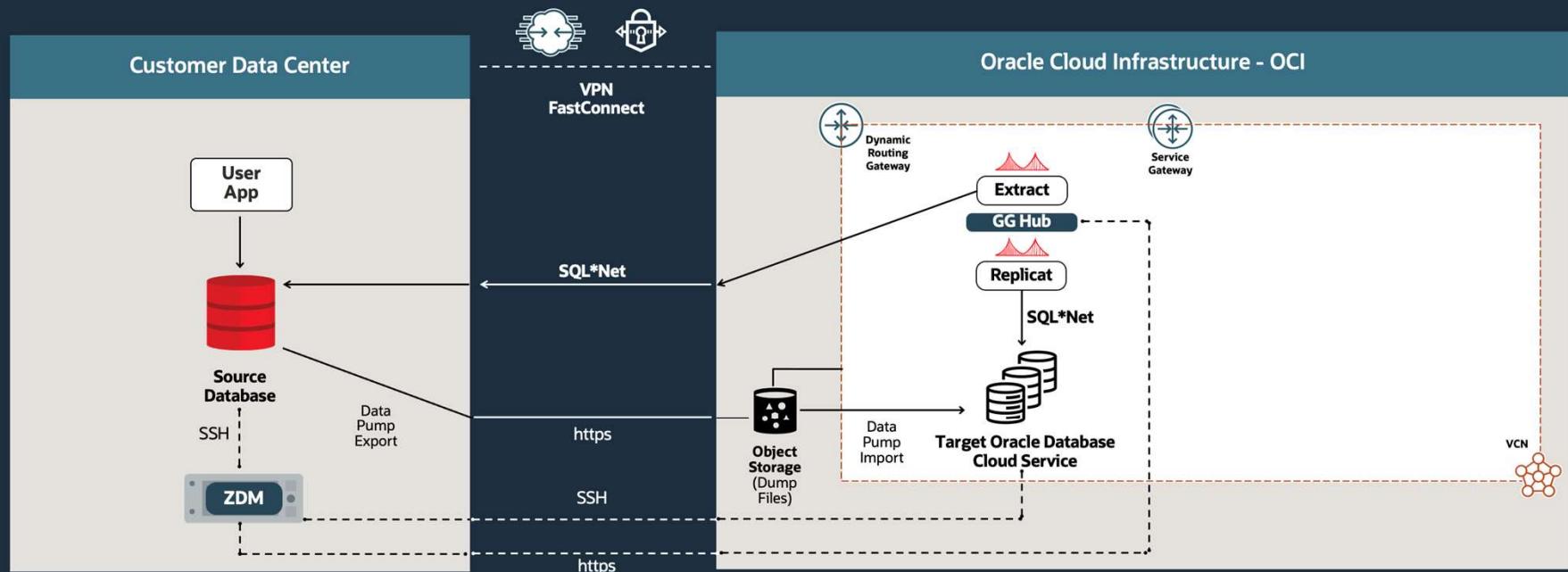


Oracle
Autonomous Database



Technical Architecture

ZDM Logical Online Database Migration



Summary

ZDM is the Best-in-Class Solution for Moving your Databases to Oracle Cloud Infrastructure



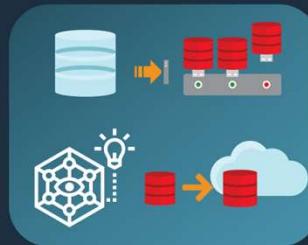
Variety of
Migration
Sources



Variety of
Migration
Targets



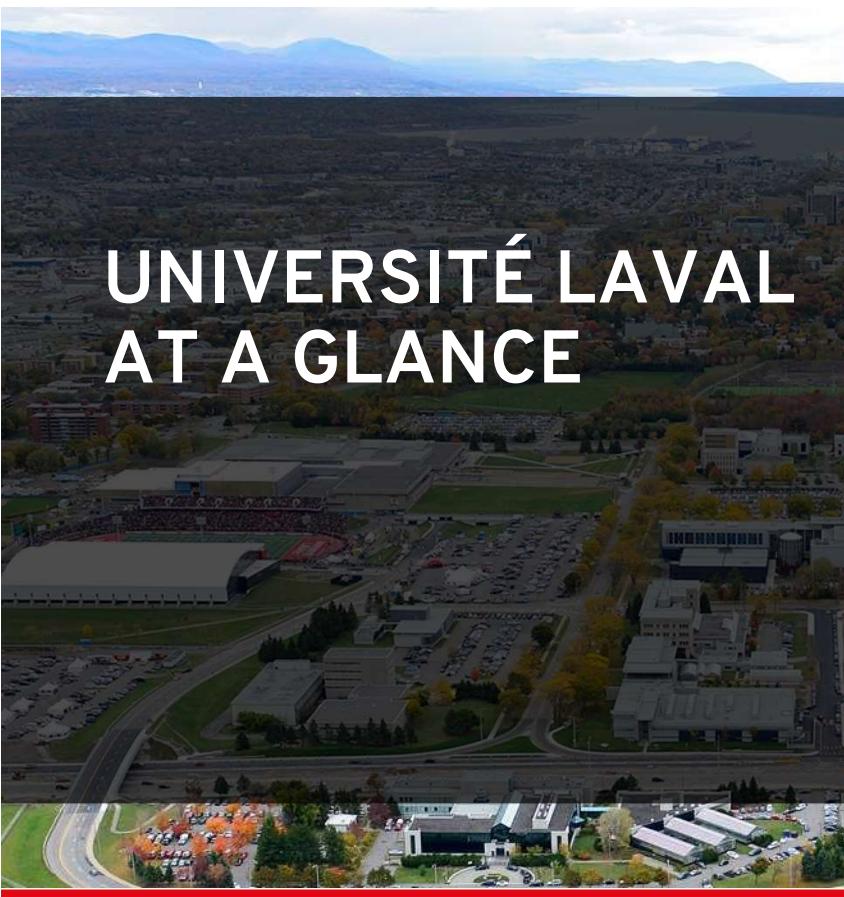
Multiple
Migration
Workflows



Best-in-Class
Features &
Functionality

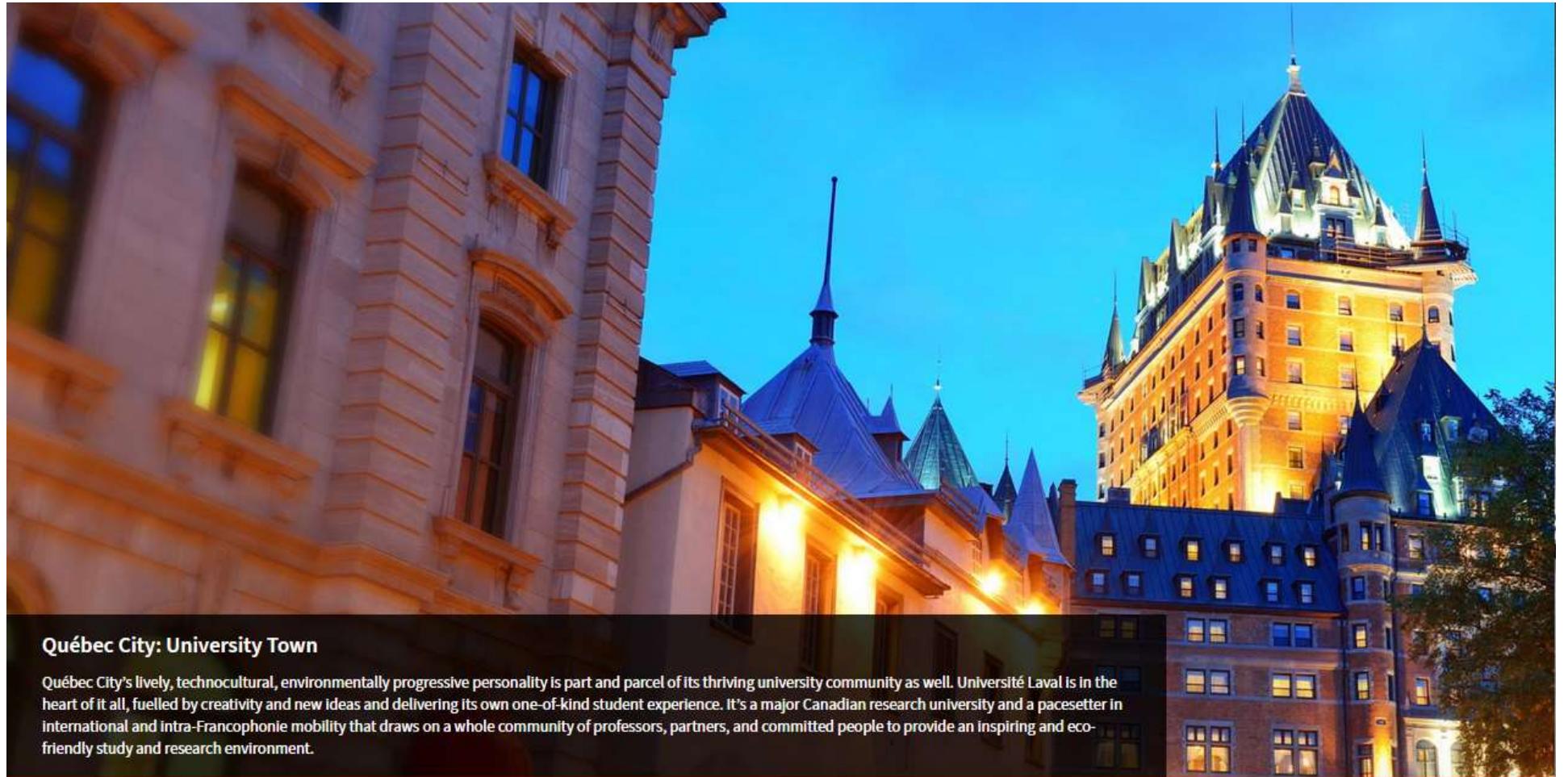


Free of
Additional
Charge



UNIVERSITÉ LAVAL AT A GLANCE





Québec City: University Town

Québec City's lively, technocultural, environmentally progressive personality is part and parcel of its thriving university community as well. Université Laval is in the heart of it all, fuelled by creativity and new ideas and delivering its own one-of-kind student experience. It's a major Canadian research university and a pacesetter in international and intra-Francophonie mobility that draws on a whole community of professors, partners, and committed people to provide an inspiring and eco-friendly study and research environment.

A modern university

With centuries of experience

Université Laval was the very first French-language university in North America.

In 1663 Monsignor François de Montmorency-Laval, the first bishop of New France, founded the colony's first educational institution: Séminaire de Québec.

Nearly 200 years later, in 1852, this institution became a university and the first building block for all French-language higher education in Québec, Canada, and North America.





More than...

56,000 ~~47,000~~
students

9,500
full and part-time employees

Education

17 Faculties

Over 60 departments, schools, and institutes

500 programs, in every field of knowledge

More than...

125 programs and 1,000 courses offered entirely online

7,000 course websites available online using Oracle Database

Université Laval ranks second for student satisfaction out of Canada's top 15 research universities according to Maclean's magazine





Research

Seventh-ranked research university in Canada

Over \$500 million in research funding

Over 300 research centres, chairs, institutes, and other groups

Many affiliated research institutions, including the CHU de Québec–Université Laval Research Centre, North America's premier francophone health research centre.





Source:
On-Premises
Exadata X6-2

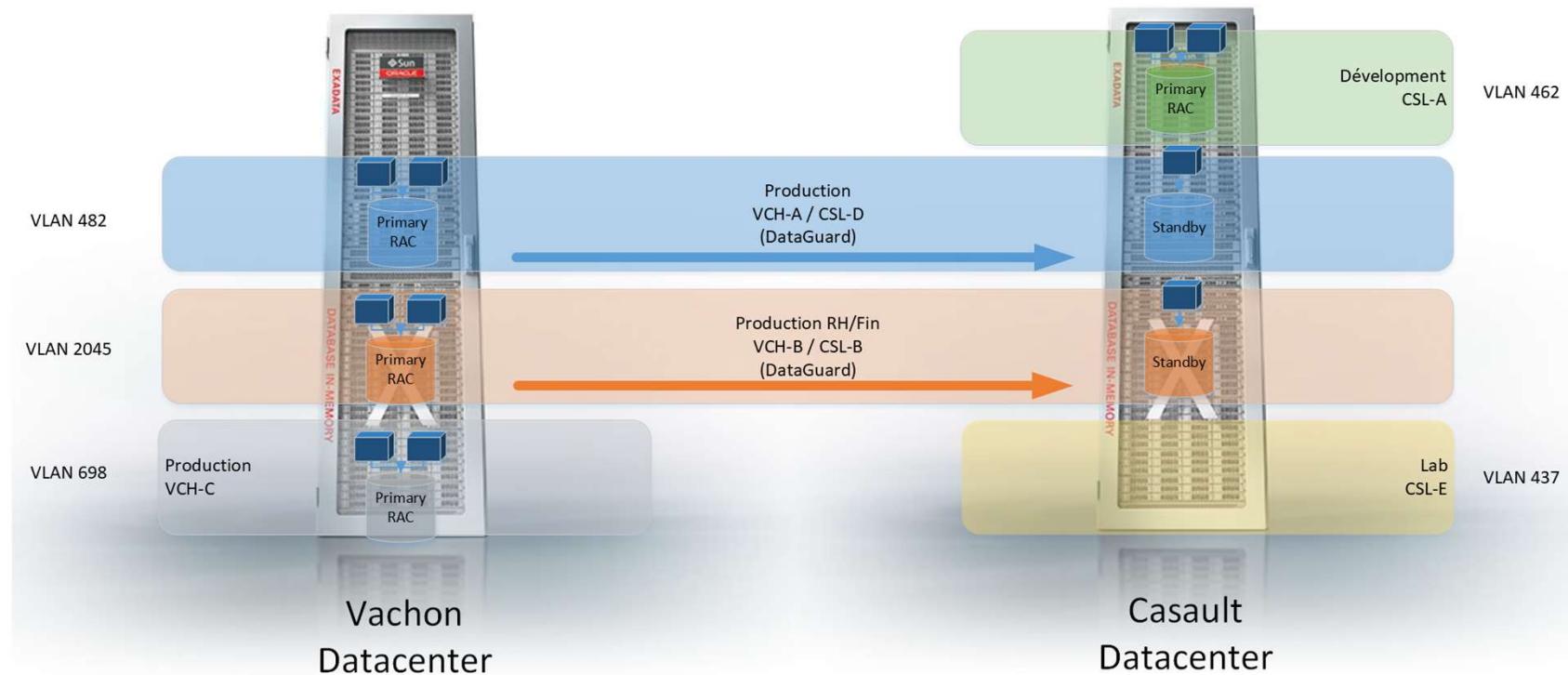


Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Source architecture Exadata X6-2 On premises



Source architecture Details

- 35 Container Databases with over 200 PDBS
 - Our largest CDB has 24 PDBs
- All in multitenant in version 12.2 and 19c
- About 56 Tb of data for all the CDB



Target:
Exadata Cloud at
Customer Gen 2

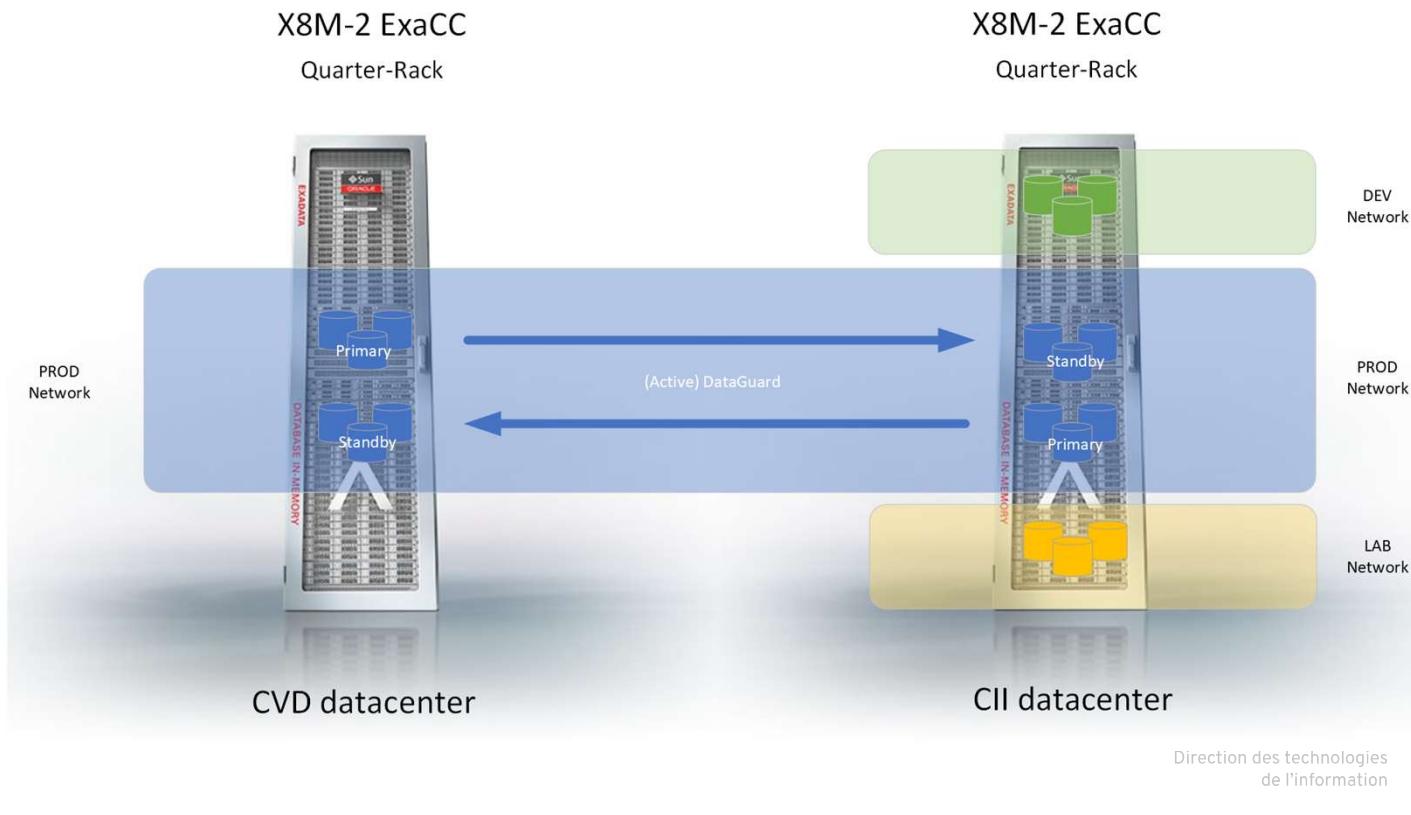


Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Target architecture schema



Key differences with on-premises architecture

- Simplified architecture with less vmcluster
- All licenced options are included (Partitionning, Active Dataguard, etc.)
- We are now responsible for vmcluster patching - domU (OS, Clusterware, Databases)
- We can use OCI console to manage a lot of parts of the architecture
 - A lot of operations are accessible by OCI API
- We pay for each OCPU allocated to the vmclusters
 - We use dynamic scaling plugin to save\$\$\$\$\$

Key differences with on-premises architecture ...

- We have no access to dom 0
- All problems with the hardware are under Oracle responsibility
- Infrastructure patching (dom0, storage cells and switches) now occurs 4 times per year (**maintenance windows**)
- On our side, we patched the OS, clusterware (GI) and DB Homes 2 times per year



The migration strategy and timeframes



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

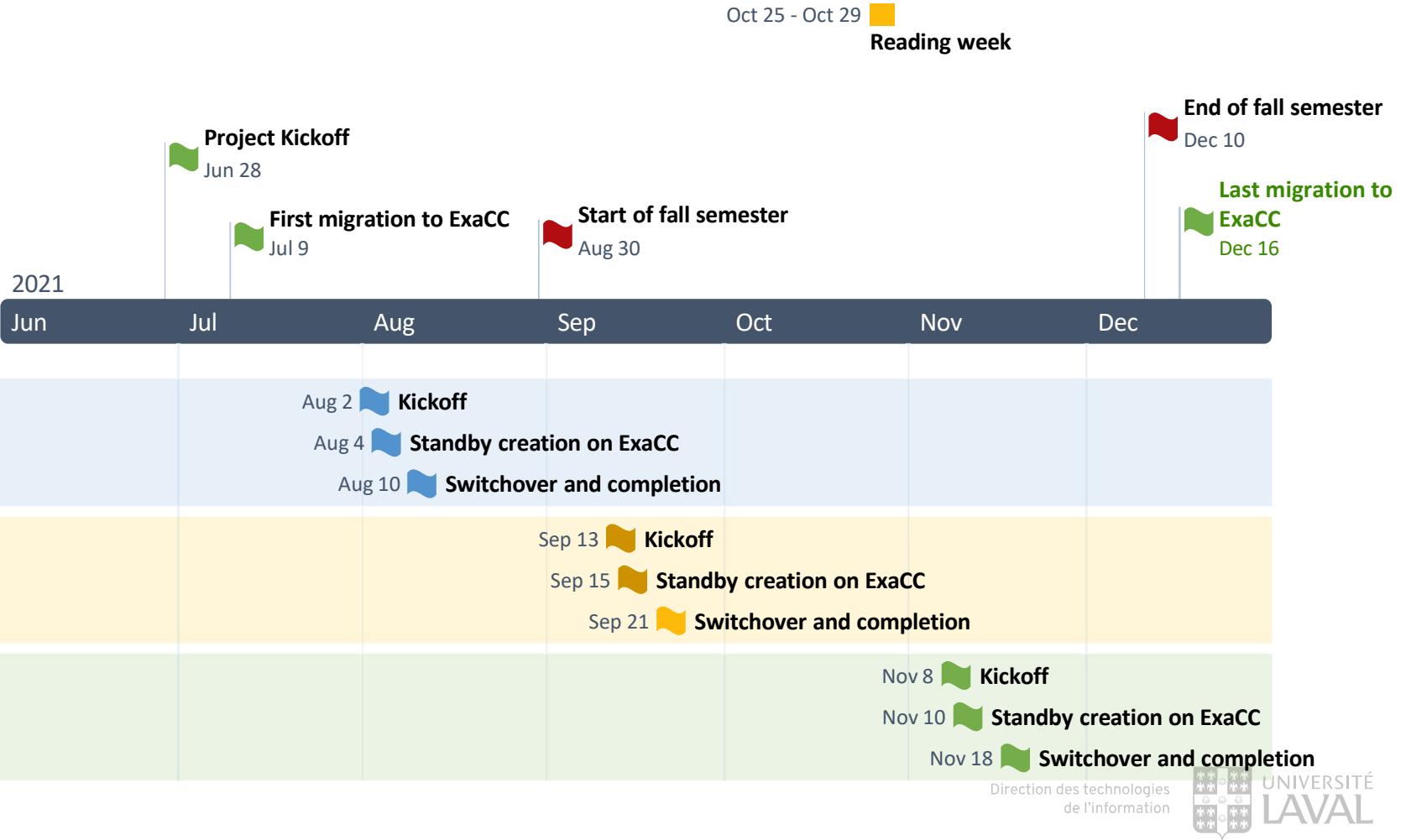
Migration strategy

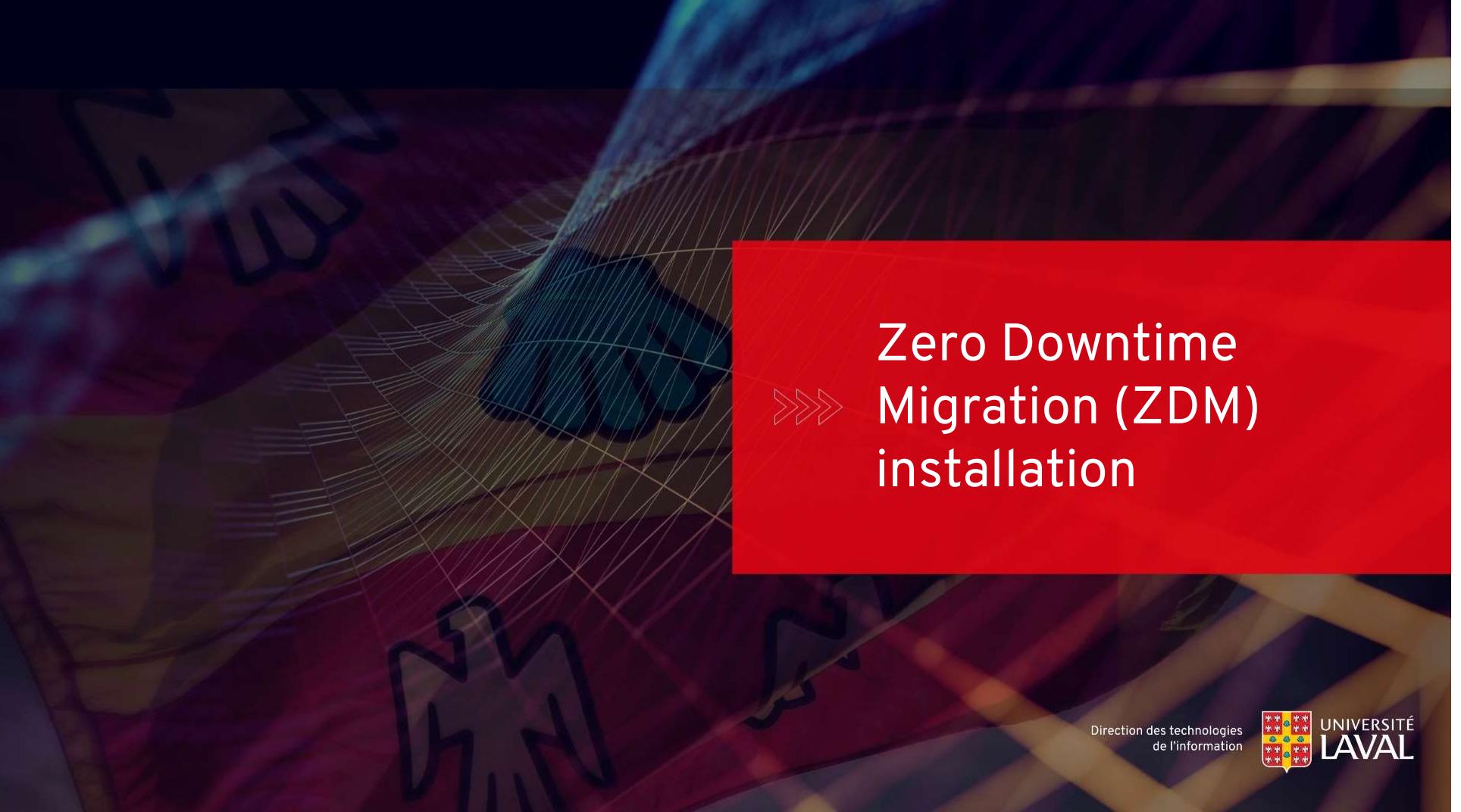
- The goal was to migrate by **line of business** instead of lifecycle/environment
- It provides a greater flexibility for the customers
- For each CDB:
 - We had a kickoff with the customers involved at the beginning
 - We specify the switchover date/hour to finalize the migration
 - **We keep the customer involved**
- We paused the CDB/PDB migration from 12.2 to 19c
- We paused the patching process for DB (release update)

Migration timeframe

About 6 months

- Should take in concern the HOT periods:
 - Start of semester
 - Exam weeks
 - ExaCC Infrastructure patches (occurs every quarter)
 - Payroll production
 - Admission
 - Course registration
 - Etc.





Zero Downtime Migration (ZDM) installation



Direction des technologies
de l'information



Vm specification for ZDM installation

- Oracle Linux 7
- 4 vcpu
- 8 G Ram
- 100 GB HD space
- Access to the targets (ExaCC Gen2) and Exadata on-premises

Package installation

```
sudo su -  
  
yum -y install \  
    gcc \  
    kernel-devel \  
    kernel-headers \  
    dkms \  
    make \  
    bzip2 \  
    perl \  
    glibc-devel \  
    expect \  
    zip \  
    unzip \  
    kernel-uek-devel-$(uname -r)
```

Group, user creation and /u01

```
groupadd zdm
useradd -g zdm zdmuser

# Change ZDM user password
passwd zdmuser
```

```
mkdir /u01
chown zdmuser:zdm /u01
```

Ssh keys generation and copy of public key

```
su - zdmuser

# Generate a rsa key for the source host
# "/home/zdmuser/.ssh/srghost" for the key name
# No pass phrase
ssh-keygen -t rsa

# Generate a rsa key for the target host
# "/home/zdmuser/.ssh/tghost" for the key name
# No pass phrase
ssh-keygen -t rsa
```

Next we copy `~/.ssh/srghost.pub` in `~/.ssh/authorized_keys` of oracle user on the source host(s) (Old Exadata on-premises)

Next we copy `~/.ssh/tghost.pub` in `~/.ssh/authorized_keys` of oracle user on the ExaCC VMCluster

Create ZDM tree structure before installation

```
su - zdmuser

echo "INVENTORY_LOCATION=/u01/app/oraInventory; export INVENTORY_LOCATION" >> ~/.bashrc
echo "ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE" >> ~/.bashrc
echo "ZDM_BASE=$ORACLE_BASE; export ZDM_BASE" >> ~/.bashrc
echo "ZDM_HOME=$ZDM_BASE/zdm21; export ZDM_HOME" >> ~/.bashrc
echo "ZDM_INSTALL_LOC=/u01/zdm21-inst; export ZDM_INSTALL_LOC" >> ~/.bashrc
source ~/.bashrc

mkdir -p $ORACLE_BASE $ZDM_BASE $ZDM_HOME $ZDM_INSTALL_LOC
```

Start ZDM installation

```
#Create staging directory and copy ZDM zipfile on the VM
mkdir /u01/staging
```

```
# Extract zipfile in $ZDM_INSTALL_LOC
unzip zdm21-3.zip -d $ZDM_INSTALL_LOC

#Start installation as zdmuser
su - zdmuser

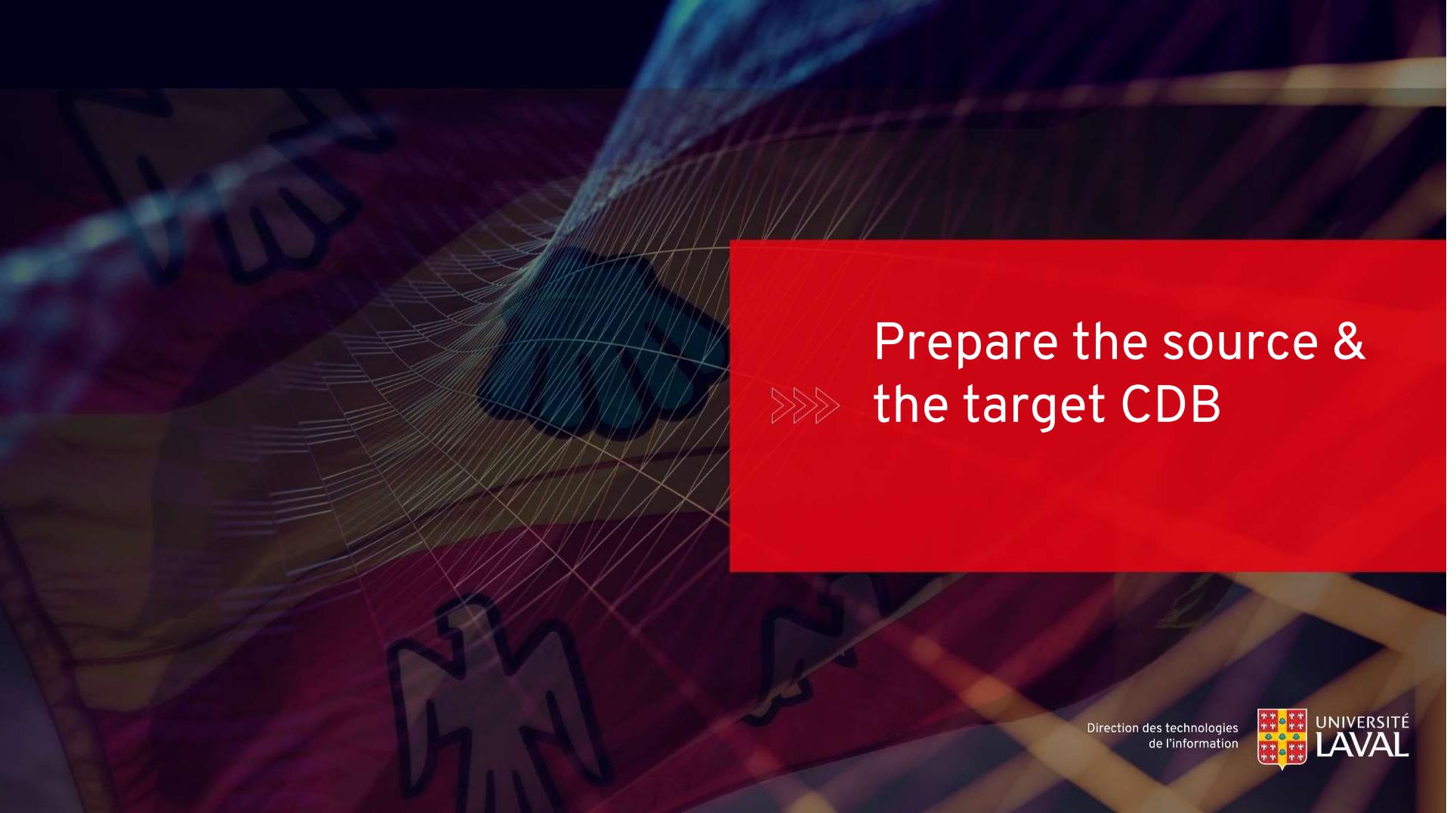
$ZDM_INSTALL_LOC/zdm21.3/zdminstall.sh setup \
oraclehome=$ZDM_HOME \
oraclebase=$ZDM_BASE \
ziploc=$ZDM_INSTALL_LOC/zdm21.3/zdm_home.zip -zdm
```

ZDM installation logs

```
-----  
Generating nogi.enabled file  
-----  
nogi.enabled file generated successfully  
-----  
Generating standalone_config.properties file  
-----  
Setting base folder permissions  
-----  
Copying service script to bin folder in Oracle Home  
ZDM service setup finished successfully...
```

ZDM service start

```
-----  
su - zdmuser  
$ZDM_HOME/bin/zdmService start  
  
# Return code is 0  
# Server started successfully.
```



Prepare the source &
the target CDB



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Prepare the source CDB

- Make oracle account suoder without password
oracle ALL=(root) NOPASSWD: ALL
- Add the destination SCAN address in file /etc/hosts
lab config
X.X.X.X scan-address.domain scan-address
- Change the SYS password to respect the password complexity requirement for ExaCC
- Plan a CDB reboot to enable the TDE wallet (we were not using TDE on our ON-PREM Exa)
- Be sure that connectivity is open on port 1521 between source and target nodes

Prepare the source CDB

- Copy the password file from ASM to \$ORACLE_HOME/dbs
To find the path
`srvctl config database -db c90csle | grep -i password`
`pwccopy +DATA/C90CSLE/PASSWORD/pwdc90csle.516.1063728837`
`/u01/app/oracle/product/19.4.0.0/dbhome_at_e20/dbs`

Prepare the source CDB

Enabling the TDE Wallet/Keystore

- Each CDB will have its own wallet and TNS files
- Create a directory that will contains the wallet files
`mkdir -p /u01/app/oracle/admin/c90csle/wallet`
- Copy the shared sqlNet.ora and Idap.ora to the new directory
`cp -p $ORACLE_HOME/network/admin/*.ora /u01/app/oracle/admin/c90csle`
- In sqlNet.ora file, add the parameter ENCRYPTION_WALLET_LOCATION pointing to the new wallet location
`ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/c90csle/wallet/)))`

Prepare the source CDB

- Modify the parameters SQLNET.ENCRYPTION_TYPES_SERVER and SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER to fit those on ExaCC if your server were using old encryption and checksum methods
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER= (SHA1,MD5)
SQLNET.ENCRYPTION_TYPES_SERVER= (AES256,AES192,AES128)
- Using SRVCTL set the environment variable TNS_ADMIN to the new location and bounce the instances

```
srvctl setenv database -db c90csle -env "TNS_ADMIN=/u01/app/oracle/admin/c90csle"  
srvctl getenv database -db c90csle -envs TNS_ADMIN
```

```
srvctl stop instance -db c90csle -instance c90csle1 -force  
srvctl start instance -db c90csle -instance c90csle1  
srvctl stop instance -db c90csle -instance c90csle2 -force  
srvctl start instance -db c90csle -instance c90csle2
```

Prepare the source CDB

Create the TDE Keystore in CDB and PDBs

- All PDBs must be in OPEN state
- Create the keystore in CDB\$ROOT as SYS
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/u01/app/oracle/admin/c90csle/wallet/' identified by <keystore_pass>;
- Open the keystore for all PDBs
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY <keystore_pass> container = ALL;
- Create and activate the master key for the CDB and all its PDBs
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY <keystore_pass> with backup container = ALL;
- Verify the views V\$ENCRYPTION_KEYS and v\$encryption_wallet;
SELECT * FROM v\$encryption_wallet;
SELECT * FROM v\$encryption_keys;

WRL_TYPE	WRL_PARAMETER	STATUS	WALLET_TYPE	WALLET_ORDER	KEYSTORE_MODE	FULLY_BACKED_UP	CON_ID
► FILE	/u01/app/oracle/admin/c19csle/wallet/	OPEN	PASSWORD	SINGLE	NONE	NO	1
FILE		OPEN	PASSWORD	SINGLE	UNITED	NO	2
FILE		OPEN	PASSWORD	SINGLE	UNITED	NO	3

Prepare the source CDB

Create the TDE Keystore in CDB and PDBs

- Make the TDE wallet autologin

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE  
'/u01/app/oracle/admin/c90csle/wallet/' IDENTIFIED BY <keystore_pass>;
```

- Close the wallet and verify that it is now opened in AUTOLOGIN mode

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY <keystore_pass>;  
SELECT * FROM v$encryption_wallet;
```

WRL_TYPE	WRL_PARAMETER	STATUS	WALLET_TYPE	WALLET_ORDER	KEYSTORE_M...	FULLY_BACKED_UP	CON_ID
▶ FILE	/u01/app/oracle/admin/c90csle/wallet/	OPEN	AUTologin	SINGLE	NONE	NO	1
FILE		OPEN	AUTologin	SINGLE	UNITED	NO	2
FILE		OPEN	AUTologin	SINGLE	UNITED	NO	3

- Copy the wallet files on the second node of RAC cluster

```
scp -p /u01/app/oracle/admin/c90csle/wallet/ew* vm02.domain :/u01/app/oracle/admin/c90csle/wallet
```

- scp -p /u01/app/oracle/admin/c90csle/wallet/cw* vm02.domain :/u01/app/oracle/admin/c90csle/wallet

Prepare the target CDB

Create the place holder on ExaCC (empty CDB reused by ZDM)

- DB Name must be the same as the source
- DB UNIQUE NAME is different like always in a dataguard setup
- SYS password is the same as the source
- DB HOME is the same version or above
- Choose the same character set
- Same compatible parameter

Prepare the target CDB

Test the SSH connectivity from ZDM server to target nodes

```
ssh -o ServerAliveInterval=300 -i /home/zdmuser/.ssh/tgthost opc@vm01.domain
```

```
ssh -o ServerAliveInterval=300 -i /home/zdmuser/.ssh/tgthost opc@vm02.domain
```

Prepare the target CDB

Change CDB\$ROOT parameters because of the place holder on ExaCC

The CDB parameters from source do not follow with the ZDM migration

```
alter system set nls_language='CANADIAN FRENCH' scope=spfile sid='*';
alter system set nls_territory='CANADA' scope=spfile sid='*';
alter system set nls_length_semantics='CHAR' sid='*';

alter system set audit_sys_operations=FALSE scope=spfile sid='*';
alter system set audit_trail='DB','EXTENDED' scope=spfile sid='*';
alter system set global_names=FALSE sid='*' scope=both;
alter system set cpu_count=2;
alter system set job_queue_processes=4;
alter system set resource_manager_plan='DEFAULT_CDB_PLAN';

alter system set processes=<to_precise> scope=spfile sid='*';
alter system set db_files=<to_precise> scope=spfile sid='*';
alter system set sga_target=<to_precise> scope=spfile sid='*';
alter system set pga_aggregate_target=<to_precise> scope=spfile sid='*';
alter system set pga_aggregate_limit=<to_precise> scope=spfile sid='*';
alter system set db_flashback_retention_target=<to_precise> sid='*';
alter system set db_recovery_file_dest_size=<to_precise> sid='*';

alter system set temp_undo_enabled=true sid='*';
...
```

Bounce the CDB

Prepare the target CDB

Deploy `ldap.ora` or edit `sqlnet.ora` if needed

- 1) We use Oracle Internet Directory (OID) as a name resolution for our TNS entries

We have to deploy a `ldap.ora` file on both servers in the `TNS_ADMIN` directory

- 2) On some PDBs, we have old password hashes (10g) and we have to specify this parameter in `sqlnet.ora` :

`SQLNET.ALLOWED_LOGON_VERSION_SERVER=11`

We know, this is a bad workaround...

Prepare the target CDB

Test the sqlnet connectivity between source and target

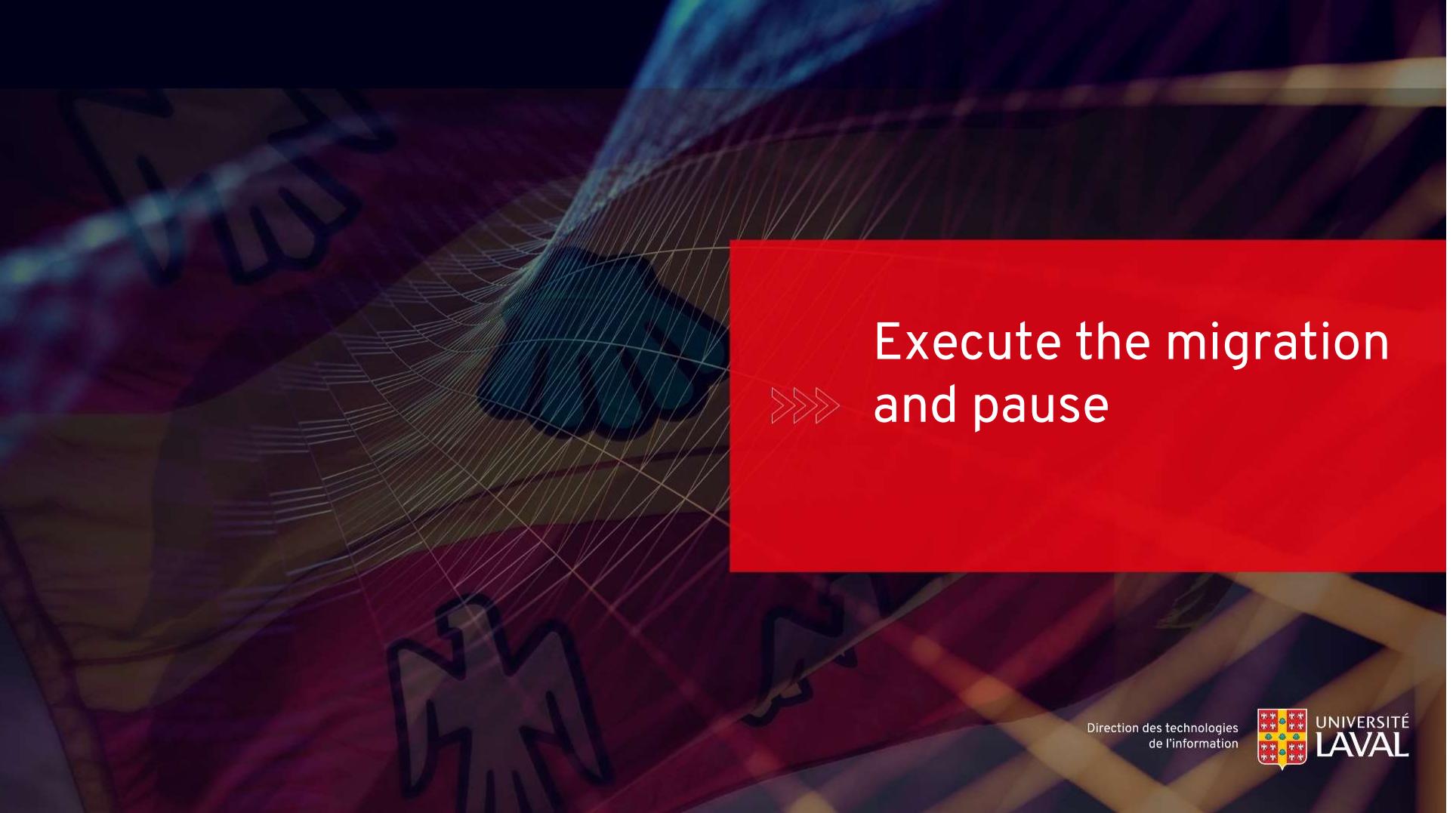
- From source to target

sqlplus system/[password]@scan_address.domain/c90csle_pdb1.paas.oracle.com

- From target to source

sqlplus system/[password]@scan_address.domain/c90csle.ul.ca

That's all! We are ready



Execute the migration
and pause



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Copy and edit the template file for ZDM

On the ZDM host

```
su - zdmuser
```

```
cp -p $ZDM_HOME/rhp/zdm/template/zdm_template.rsp ~/zdm_c90csle.rsp
```

Edit these parameters according to the CDB name and size
(to change for each CDB):

- TGT_DB_UNIQUE_NAME=c90cslecc
- SRC_RMAN_CHANNELS=2
- TGT_RMAN_CHANNELS=8

Edit the template file for ZDM

We use these values for all our migrations

- PLATFORM_TYPE=EXACC
- MIGRATION_METHOD=**ONLINE_PHYSICAL**
- DATA_TRANSFER_MEDIUM=NFS
- BACKUP_PATH=/mnt/ULEXA → A sub-directory named **db_name** will be created under this directory (example: /mnt/ULEXA/c90csle)
- TGT_SKIP_DATAPATCH=FALSE
- SKIP_FALLBACK=TRUE → Since the target will use TDE, after the switchover, we cannot ship redo logs from encrypted source on ExaCC to former unencrypted CDB

Launch the migration in EVAL mode

Parameters to modify for each migration

- rsp → response file
- sourcedb
- sourcenode
- targetnode
- targethome

*We use the « -ignore PATCH_CHECK » parameter to ignore ONEOFF Patches that are missing on target

Launch the migration in EVAL mode

```
su - zdmuser

$ZDM_HOME/bin/zdmcli migrate database \
-rsp ~/zdm_c90csle.rsp \
-sourcedb c90csle \
-sourcenode [REDACTED] \
-srcauth zdmauth \
-srcarg1 user:oracle \
-srcarg2 identity_file:/home/zdmuser/.ssh/srchost \
-srcarg3 sudo_location:/usr/bin/sudo \
-targetnode [REDACTED] \
-tgtauth zdmauth \
-tgtarg1 user:opc \
-tgtarg2 identity_file:/home/zdmuser/.ssh/tgthost \
-tgtarg3 sudo_location:/usr/bin/sudo \
-targethome /u02/app/oracle/product/19.0.0.0/dbhome_2 \
-ignore PATCH_CHECK \
-eval
```

Launch the migration in EVAL mode

Verify the status

```
[zdmuser@ ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 7
: Audit ID: 23
Job ID: 7
User: zdmuser
Client: 
Job Type: "EVAL"
Scheduled job command: "zdmcli migrate database -rsp /home/zdmuser/zdm_c90csle.rsp -sc
Scheduled job execution start time: 2021-02-10T11:17:28-05. Equivalent local time: 2021-02-10 11:17:28
Current status: SUCCEEDED
Result file path: "/u01/app/oracle/chkbase/scheduled/job-7-2021-02-10-11:17:46.log"
Job execution start time: 2021-02-10 11:17:46
Job execution end time: 2021-02-10 11:20:34
Job execution elapsed time: 2 minutes 48 seconds
ZDM_GET_SRC_INFO ..... PRECHECK_PASSED
ZDM_GET_TGT_INFO ..... PRECHECK_PASSED
ZDM_SETUP_SRC ..... PRECHECK_PASSED
ZDM_SETUP_TGT ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS_TGT ..... PRECHECK_PASSED
ZDM_VALIDATE_SRC ..... PRECHECK_PASSED
ZDM_VALIDATE_TGT ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS_TGT ..... PRECHECK_PASSED
ZDM_CLEANUP_SRC ..... PRECHECK_PASSED
ZDM_CLEANUP_TGT ..... PRECHECK_PASSED
```

Launch the migration in MIGRATE mode

Same command as EVAL but without the switch -EVAL 😊
and **we want to pause the process after the dataguard configuration**

We will use the switch « -pauseafter ZDM_CONFIGURE_DG_SRC »

```
$ZDM_HOME/bin/zdmcli migrate database \
-rsp ~/zdm_c90csle.rsp \
-sourcedb c90csle \
-sourcenode [REDACTED] \
-srcauth zdmauth \
-srcarg1 user:oracle \
-srcarg2 identity_file:/home/zdmuser/.ssh/srchost \
-srcarg3 sudo_location:/usr/bin/sudo \
-targetnode [REDACTED] \
-tgtauth zdmauth \
-tgtarg1 user:opc \
-tgtarg2 identity_file:/home/zdmuser/.ssh/tgthost \
-tgtarg3 sudo_location:/usr/bin/sudo \
-targethome /u02/app/oracle/product/19.0.0.0/dbhome_2 \
-ignore PATCH_CHECK \
-pauseafter ZDM_CONFIGURE_DG_SRC
```

Launch the migration in MIGRATE mode

```
[zdmuser@... ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 8
[...]: Audit ID: 28
Job ID: 8
User: zdmuser
Client: [...]
Job Type: "MIGRATE"
Scheduled job command: "zdmcli migrate database -rsp /home/zdmuser/zdm_c90csle.rsp -sourcedb c90csle -s
Scheduled job execution start time: 2021-02-10T13:43:11-05. Equivalent local time: 2021-02-10 13:43:11
Current status: PAUSED
Current Phase: "ZDM_CONFIGURE_DG_SRC"
Result file path: "/u01/app/oracle/chkbase/scheduled/job-8-2021-02-10-13:43:17.log"
Job execution start time: 2021-02-10 13:43:17
Job execution end time: 2021-02-10 14:01:23
Job execution elapsed time: 18 minutes 5 seconds
ZDM_GET_SRC_INFO ..... COMPLETED
ZDM_GET_TGT_INFO ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_SETUP_TGT ..... COMPLETED
ZDM_PREUSERACTIONS ..... COMPLETED
ZDM_PREUSERACTIONS_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_OBC_INST_SRC ..... COMPLETED
ZDM_OBC_INST_TGT ..... COMPLETED
ZDM_BACKUP_FULL_SRC ..... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC ..... COMPLETED
ZDM_COPYFILES ..... COMPLETED
ZDM_PREPARE_TGT ..... COMPLETED
ZDM_SETUP_TDE_TGT ..... COMPLETED
ZDM_CLONE_TGT ..... COMPLETED
ZDM_FINALIZE_TGT ..... COMPLETED
ZDM_CONFIGURE_DG_SRC ..... COMPLETED
ZDM_SWITCHOVER_SRC ..... PENDING
ZDM_SWITCHOVER_TGT ..... PENDING
ZDM_POST_DATABASE_OPEN_TGT .... PENDING
ZDM_DATAPATCH_TGT ..... PENDING
ZDM_MANIFEST_TO_CLOUD ..... PENDING
ZDM_POSTUSERACTIONS ..... PENDING
ZDM_POSTUSERACTIONS_TGT ..... PENDING
ZDM_CLEANUP_SRC ..... PENDING
ZDM_CLEANUP_TGT ..... PENDING
```

Validate that the archived logs are well replicated

-- On source

```
SELECT host_name, instance_name, db_unique_name, status, database_role, open_mode
FROM v$database, v$instance;
SELECT thread#, max(sequence#) FROM v$archived_log GROUP BY thread#;
```

-- On target

```
SELECT host_name, instance_name, db_unique_name, status, database_role, open_mode
FROM v$database, v$instance;
select process, thread#, sequence#, status from gv$managed_standby;
SELECT thread#, max(sequence#) FROM v$archived_log WHERE applied='YES' GROUP BY thread#;
```

What if a step failed...

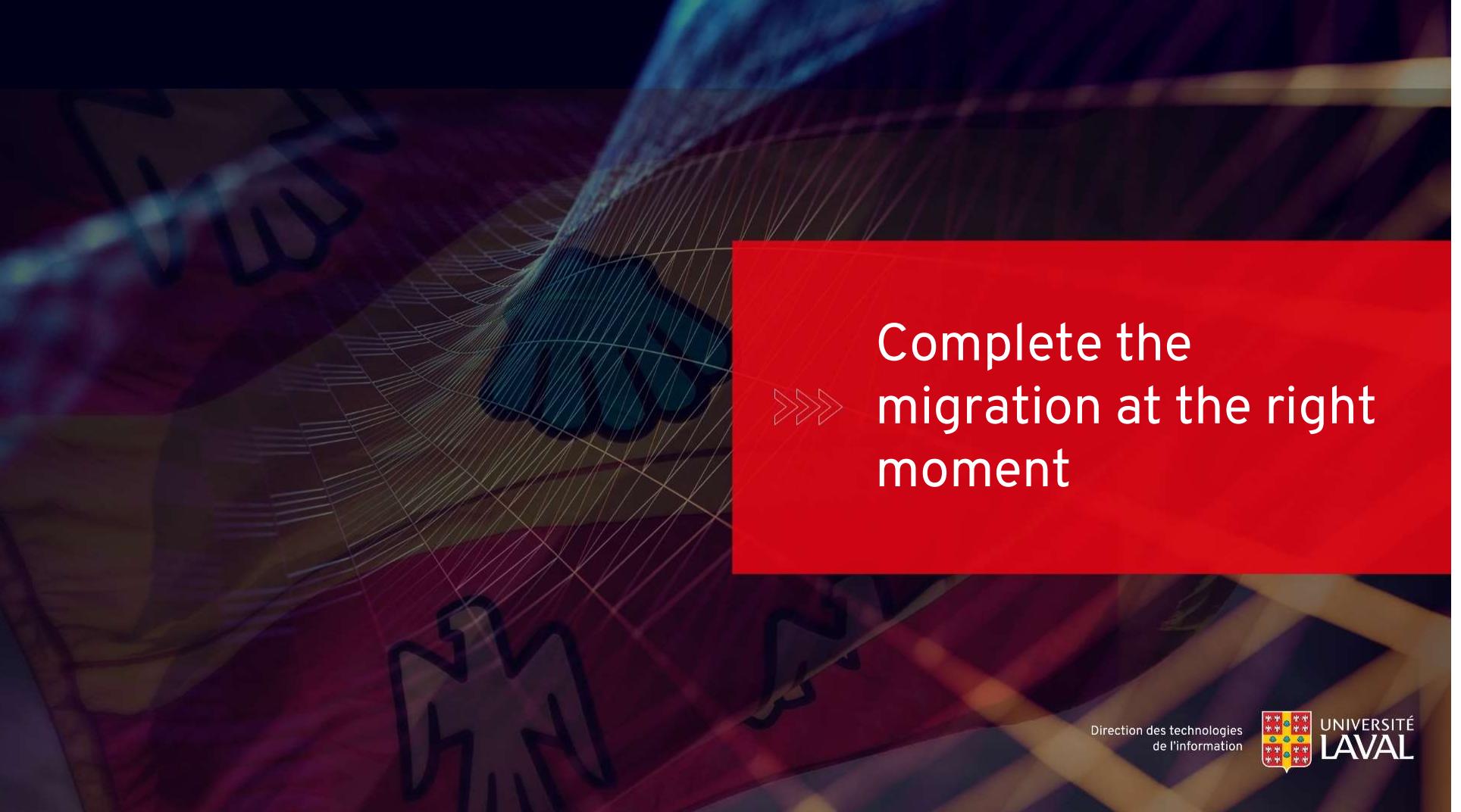
It fails??

No problem!

You look at the ZDM logs (zdm server, source node or target node)

Correct the issue and resume the ZDM job!

```
$ZDM_HOME/bin/zdmcli resume job -jobid <jobid> -pauseafter ZDM_CONFIGURE_DG_SRC
```



Complete the
migration at the right
moment



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Complete the migration at the right moment

With ZDM, we really enjoy the option to choose the right moment to do the switchover and to complete the migration.

Now this is D-Day - 1, we edit our TNS entries in OID (and in our datasource config) to add the second host (as a standard dataguard config) to minimise downtime after the switchover.

```
(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=192.168.1.11) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP)
(HOST=192.168.1.12) (PORT=1521)))
(CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=service_abc_c90csle.ul.ca)))
```

Complete the migration at the right moment

D-DAY

- Put a blackout in OEM on the source CDB and disable backup jobs
- Resume the ZDM Job

```
$ZDM_HOME/bin/zdmcli resume job -jobid 8  
### XXXXXXXXXXXXXX: Audit ID: 29  
  
$ZDM_HOME/bin/zdmcli query job -jobid 8
```

- Took about 12 minutes to complete including datapatch (elapsed time not downtime)

Complete the migration

```
[zdmuser@... ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 8
Audit ID: 36
Job ID: 8
User: zdmuser
Client: ...
Job Type: "MIGRATE"
Scheduled job command: "zdmcli migrate database -rsp /home/zdmuser/zdm_c90csle.rsp -sc
Scheduled job execution start time: 2021-02-10T13:43:11-05. Equivalent local time: 2021-02-10T13:43:11-05
Current status: SUCCEEDED
Result file path: "/u01/app/oracle/chkbase/scheduled/job-8-2021-02-10-13:43:17.log"
Job execution start time: 2021-02-10 14:33:18
Job execution end time: 2021-02-10 14:45:14
Job execution elapsed time: 11 minutes 55 seconds
ZDM_GET_SRC_INFO ..... COMPLETED
ZDM_GET_TGT_INFO ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_SETUP_TGT ..... COMPLETED
ZDM_PREUSERACTIONS ..... COMPLETED
ZDM_PREUSERACTIONS_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_OBC_INST_SRC ..... COMPLETED
ZDM_OBC_INST_TGT ..... COMPLETED
ZDM_BACKUP_FULL_SRC ..... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC ..... COMPLETED
ZDM_COPYFILES ..... COMPLETED
ZDM_PREPARE_TGT ..... COMPLETED
ZDM_SETUP_TDE_TGT ..... COMPLETED
ZDM_CLONE_TGT ..... COMPLETED
ZDM_FINALIZE_TGT ..... COMPLETED
ZDM_CONFIGURE_DG_SRC ..... COMPLETED
ZDM_SWITCHOVER_SRC ..... COMPLETED
ZDM_SWITCHOVER_TGT ..... COMPLETED
ZDM_POST_DATABASE_OPEN_TGT .... COMPLETED
ZDM_DATAPATCH_TGT ..... COMPLETED
ZDM_MANIFEST_TO_CLOUD ..... COMPLETED
ZDM_POSTUSERACTIONS ..... COMPLETED
ZDM_POSTUSERACTIONS_TGT ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
ZDM_CLEANUP_TGT ..... COMPLETED
```

What if datapatch step failed??

You relaunch datapatch manually and resume the ZDM job

```
export NLS_LANG=AMERICAN_AMERICA.US7ASCII  
$ORACLE_HOME/OPatch/datapatch -verbose -pdbs CDB\$/ROOT  
$ORACLE_HOME/OPatch/datapatch -verbose -pdbs PDB\$/SEED  
$ORACLE_HOME/OPatch/datapatch
```

```
select * from dba_registry_sqlpatch;
```

PATCH_ID	PATCH_UID	VERSION	FLAGS	ACTION	STATUS	INSTALL_ID	ACTION_TIME	DESCRIPTION	BL
31219919	23625440	12.2.0.1	NJ	APPLY	SUCCESS	7	2020/12/01 16:40:31,362825	OJVM RELEASE UPDATE 12.2.0.1.200714	
32228578	24008225	12.2.0.1	NB	APPLY	SUCCESS	8	2021/04/15 10:59:10,052439	DATABASE JAN 2021 RELEASE UPDATE 12.2.0.1.210119 DE	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		9	2021/09/02 09:24:25,555626	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		10	2021/09/02 09:25:27,106634	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		11	2021/09/02 09:26:32,755799	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		12	2021/09/02 09:27:50,335545	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		13	2021/09/02 09:28:52,695650	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK BEGIN		14	2021/09/02 09:29:54,304244	OJVM RELEASE UPDATE 12.2.0.1.200714	
31219919	23625440	12.2.0.1	NJ	ROLLBACK SUCCESS		15	2021/09/02 09:32:53,931222	OJVM RELEASE UPDATE 12.2.0.1.200714	
32119931	23943202	12.2.0.1	NJ	APPLY	SUCCESS	15	2021/09/02 09:32:53,937320	OJVM RELEASE UPDATE 12.2.0.1.210119	
25292893	23964462	12.2.0.1	N	APPLY	SUCCESS	15	2021/09/02 09:32:54,390651	PHSB: RMAN SHOULD NOT CATALOG OTHER DATABASE...	
32269634	23964501	12.2.0.1	N	APPLY	SUCCESS	15	2021/09/02 09:32:54,413456	MERGE REQUEST ON TOP OF DATABASE JAN 2021 RU...	

```
$ZDM_HOME/bin/zdmcli resume job -jobid <jobid>
```

Migration post-steps

- Verify that the services are running on the new CDB

```
srvctl status service -db c90cslecc
```

- Remove the default « place holder » service

```
srvctl remove service -db c90cslecc -service c90csle_pdb1.paas.oracle.com
```

- Close the former CDB on-premises

```
. oraenv <<< c90csle1  
srvctl stop database -db c90csle  
srvctl disable database -db c90csle
```

Migration post-steps

- Restart the Weblogic managed servers
- Discover the new CDB target in OEM
- Configure and enable the backup
- Launch a FULL backup
- On both nodes on ExaCC, remove the crontab entry that is backing up the archived logs since we are doing it externally

```
[root@exac ~]# cat /etc/crontab
...
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | |
# * * * * * user-name command to be executed
15 * * * * oracle /var/opt/oracle/misc/backup_db_wallets.pl
15 * * * * oracle /var/opt/oracle/dbaascli/dbaascli tde backup --alldb
14,44 * * * * oracle /var/opt/oracle/bkup_api/bkup_api bkup_archlogs --cron --dbname=c90csle <== remove this line
0 */12 * * * oracle /var/opt/oracle/cleandb/cleandbblogs.pl
0 04 * * * grid /var/opt/oracle/cleandb/cleandbblogs.pl
```



Repeat the migration!

34 CDBs remaining...

33

32

...

...

3

2

1

Finish!



ZDM lessons learned



Direction des technologies
de l'information



Issues that we met...

- During the pause process:
 - The Fast Recovery Area (FRA) on the target database was filling up (db_recovery_file_dest_size was too small)
 - The archived logs were not applied on the target db when the CDB was restarted (start option = read only)
- After the switchover:
 - We had to retry the datapatch step because it failed in 12.2

Lessons learned

- We should have discovered the target CDB in OEM prior the switchover to monitor the dataguard process
- Make sure to use the right job_id when you switch a production database!!!! We have once switched the wrong production Database!



Register the target in
DataSafe



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

Create a VM for datasafe on-prem connector

- We use Datasafe to check our database security posture
- To use Datasafe in our datacenter we install an Oracle Linux VM
- This vm will be the link between OCI console and the ExaCC databases that are in our Datacenter

Connector installation

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.el7_9.x86_64/jre/
python3 setup.py install --connector-port=1560
Enter install bundle password:
Data Safe on-premises connector installation in progress...
Data Safe on-premises connector successfully installed
```

Add each PDB as a Datasafe target in OCI Console

The screenshot shows the Oracle Cloud Data Safe Target Databases interface. At the top, there's a navigation bar with 'Data Safe' and 'Target Databases'. Below it, a table header includes columns for 'Target Database Name', 'Status', 'Registration Time', and 'Database Type'. A red box highlights the 'Register Database' button. On the left, a sidebar has 'Target Databases' selected. The main area is titled 'Register Target Database' and contains several input fields: 'Database Type' (set to 'Oracle Cloud@Customer Database'), 'Data Safe Target Information' (with 'Select VM Cluster in uavalca (root)' and 'Home Available' dropdowns), 'Data Safe Target Display Name' (empty), 'Description (Optional)' (empty), 'Compartment' (set to 'uavalca (root)'), 'Data Safe Target Connection Details' (with 'On-Premises Connector' dropdown set to 'Select On-Premises Connector in uavalca (root) (Change Compartment)'), and a note about selecting a Data Safe on-premises connector. At the bottom are 'Register' and 'Cancel' buttons.

OCI API can be used to add hundred of PDBS

Datasafe Dashboard

Security Center

Dashboard

Key security indicators for all the registered target databases

Dashboard

Security Assessment

User Assessment

Data Discovery

Data Masking

Activity Auditing

Alerts

List Scope

Compartment

ulavalca (root)

Include child compartments

Filters

Target Databases

ORDV

Security Assessment

52 Risks

Severity	Count
High	0
Low	5
Evaluate	33
Medium	3
Advisory	11

User Assessment

43 Users

Severity	Count
Critical	27
High	4
Medium	0
Low	12

Data Discovery

No data to display

Top 5 Targets

No data to display

All Activity

Last 1 Week

Audit Events

Admin Activity

Last 1 Week

Audit Events

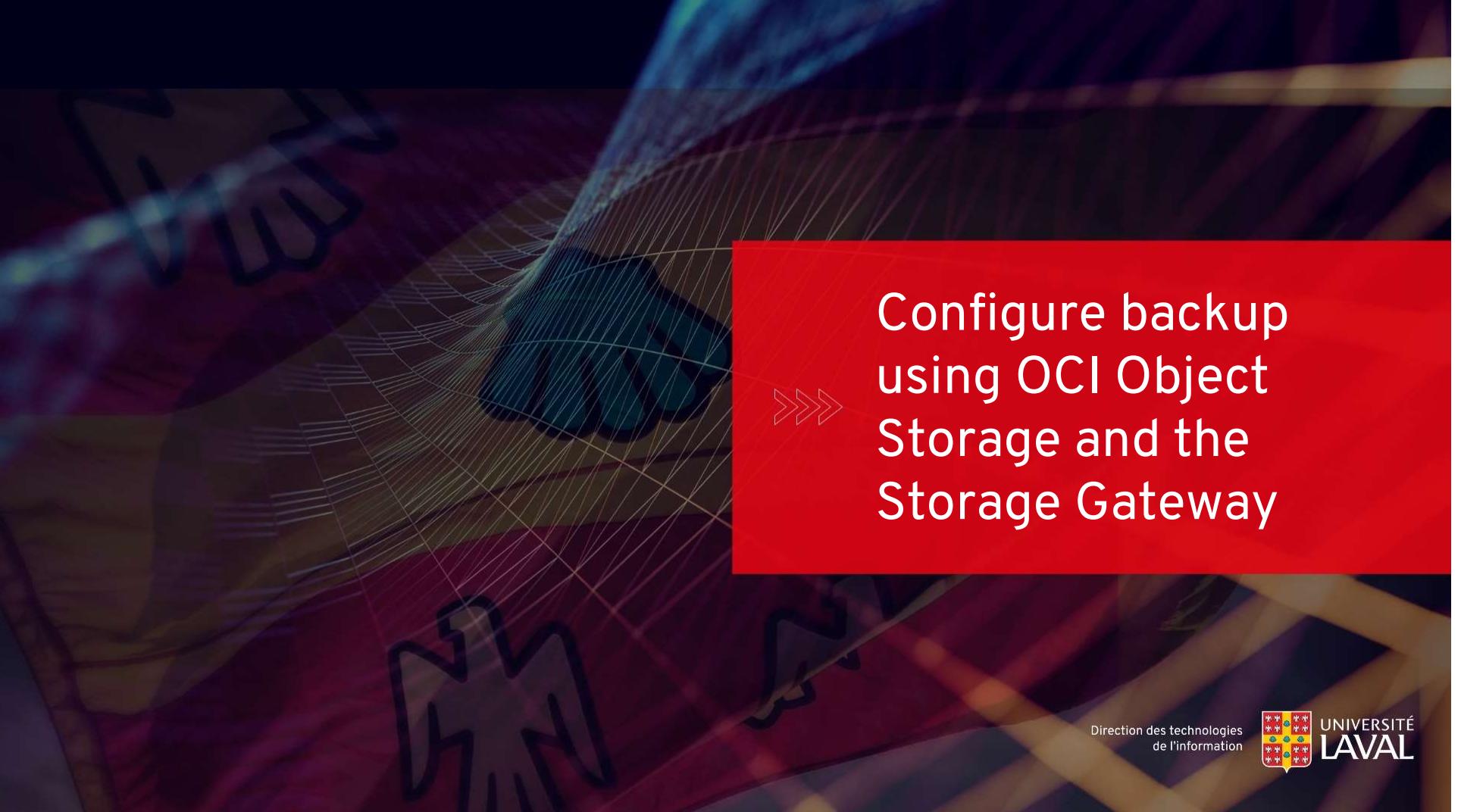
Open Alerts

Last 1 Week

No data to display

Direction des technologies
de l'information





Configure backup
using OCI Object
Storage and the
Storage Gateway



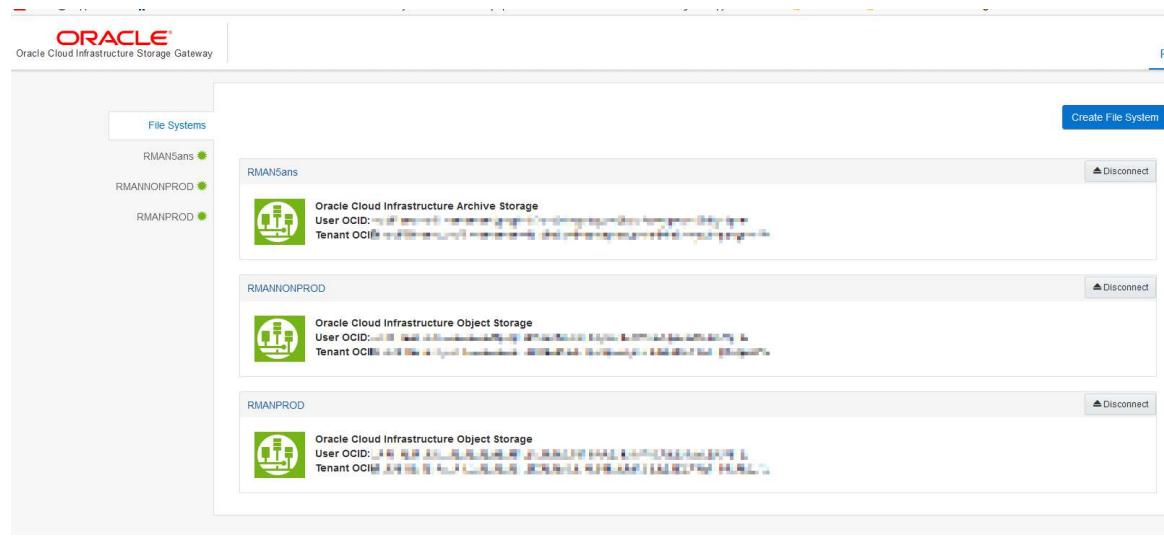
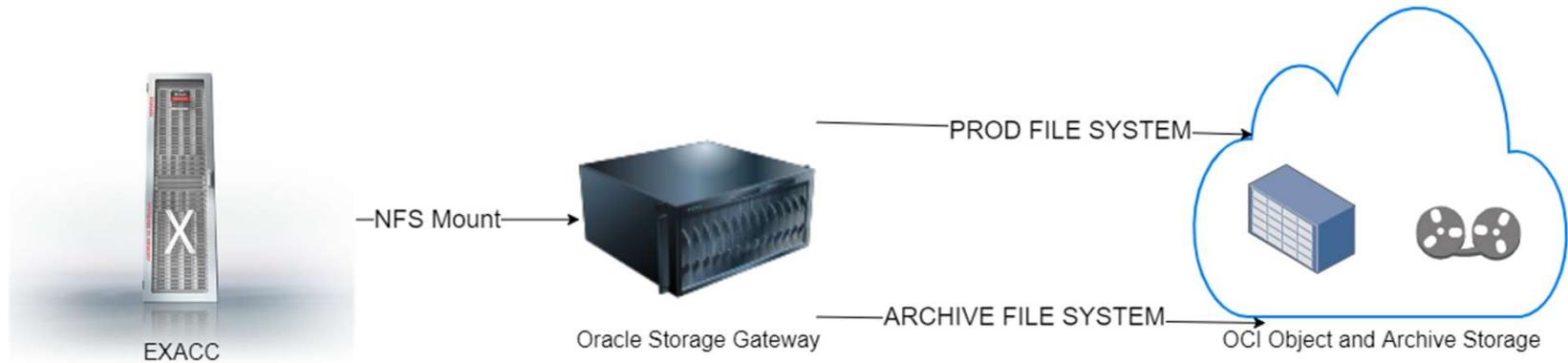
Direction des technologies
de l'information



UNIVERSITÉ
LAVAL

RMAN backup on OCI object storage with the storage gateway

- All our rman backup are stored on OCI Object storage or Archive Storage
- We use Oracle Storage Gateway on a VM in our Datacenter to transfer files on the OCI object storage / archive storage
- The Storage Gateway is installed on a simple Oracle Linux VM in our Datacenter.



Direction des technologies
de l'information

Oracle Storage Gateway benefits

- Local cache on the storage gateway and files are sync to the cloud asynchronously
- Possibility to sync files on the archive storage and bypass the object storage

Thanks

Direction des technologies
de l'information





QUESTIONS?



Direction des technologies
de l'information



UNIVERSITÉ
LAVAL