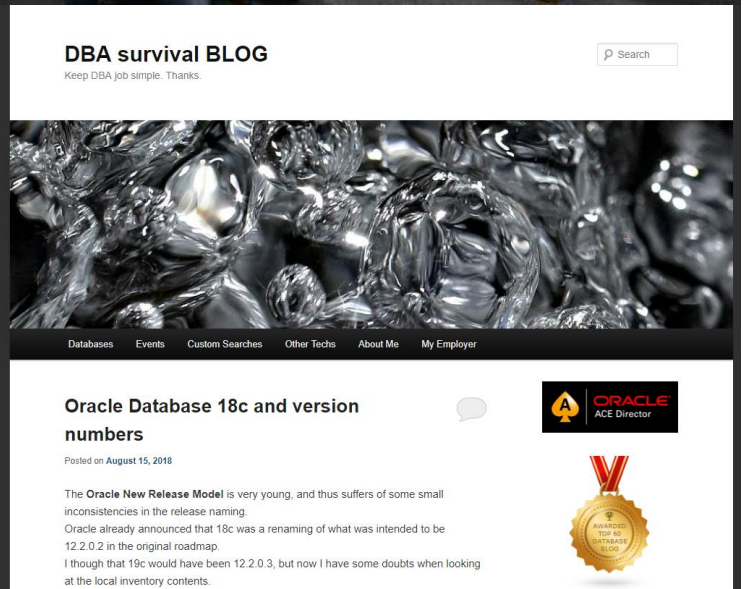
Let Your DBAs Get Some REST(api)

Ludovico Caldara - Computing Engineer @CERN, Oracle ACE Director



Ludovico Caldara

- Two decades of DBA experience (Not Only Oracle)
- ITOUG co-founder
- OCP (11g, 12c, MySQL) & OCE
- Italian living in Switzerland

http://www.ludovicocaldara.net

@ludodba

ludovicocaldara

ORACLE ACE Director

The Large Hadron Collider (LHC)

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

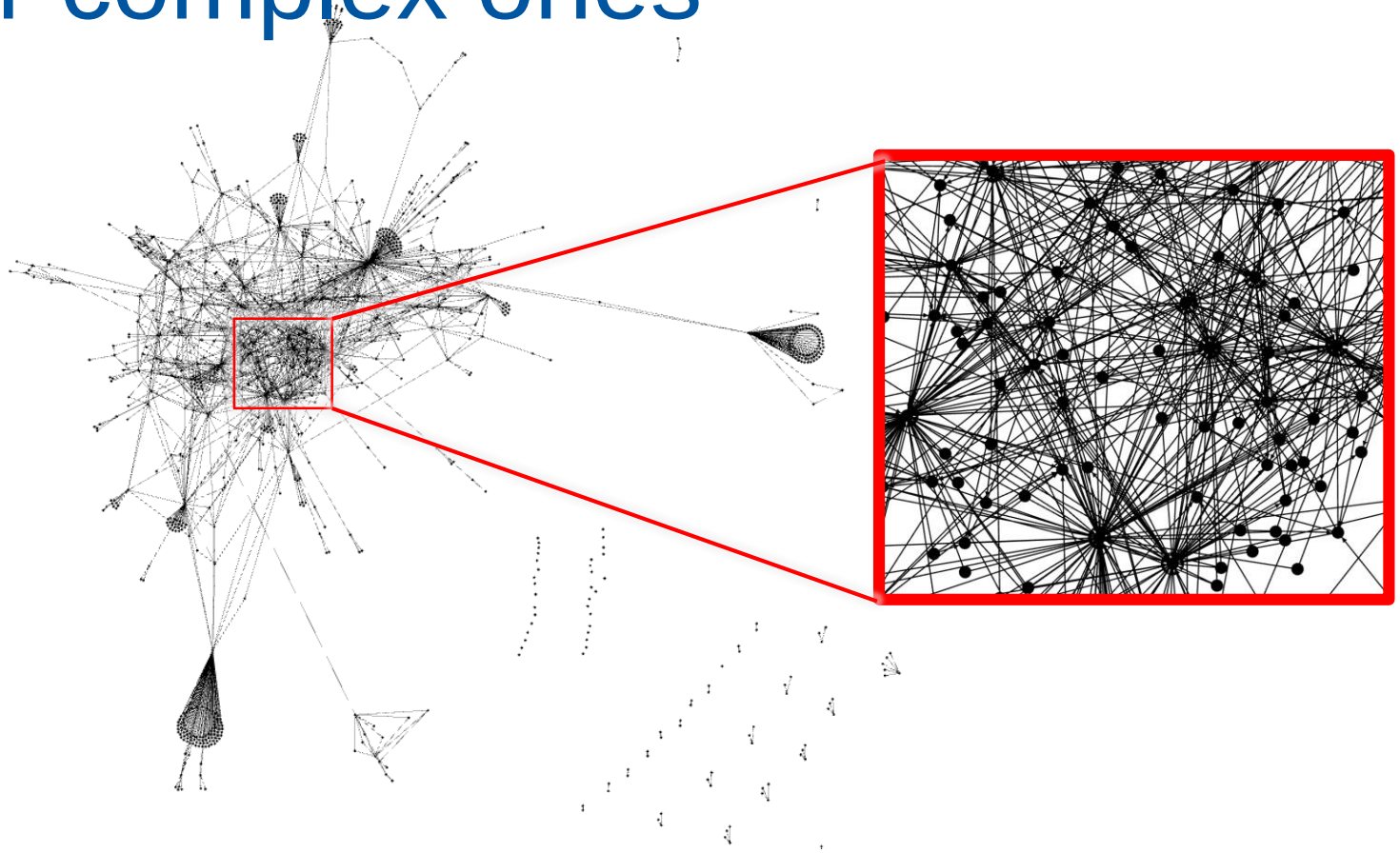
Hottest spot in the galaxy

During Lead ion collisions create temperatures 100 000x hotter than the heart of the sun

Large databases

```
SQL> select sum(bytes/power(1024,5)) as "PetaBytes"
> from dba_data_files;

     PetaBytes
--------------
1.052794738695
```

Or complex ones



Oracle Cloud Infrastructure

New Free Tier

oracle.com/gbtour

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services

Haaa, the joy of development!


Trello


Trello
Atom


Trello
Git
Atom


Trello
Atom
Git
Jira


Trello
Atom
Git
Jira
Jenkins


Trello
Docker
Atom
Git
Jira
Jenkins

Everything is beautiful and clean, and you feel more agile than ever

Still something is missing…
you need some magic from the DBA

Puppet

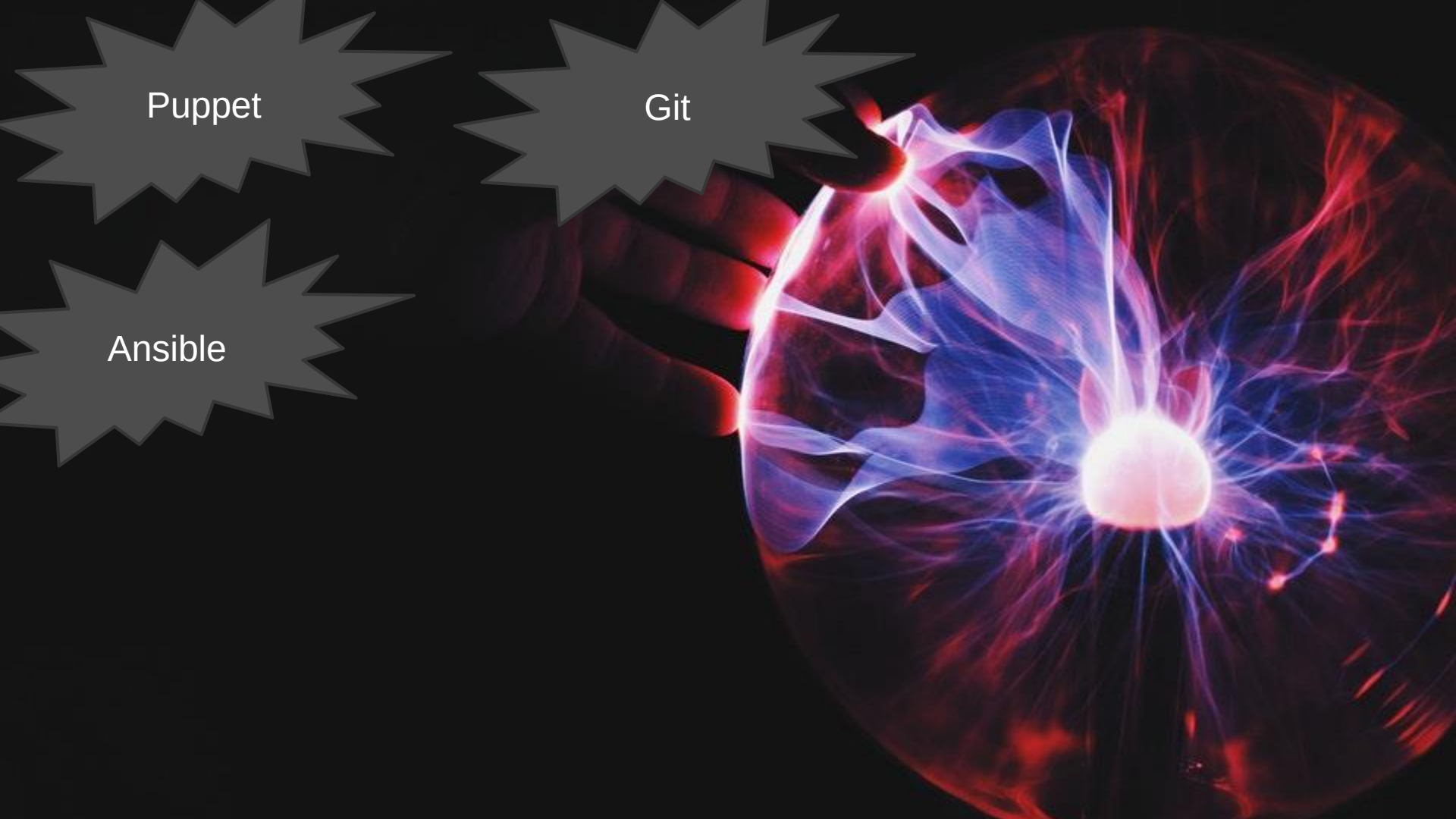
Puppet

Git

Puppet

Git

Ansible

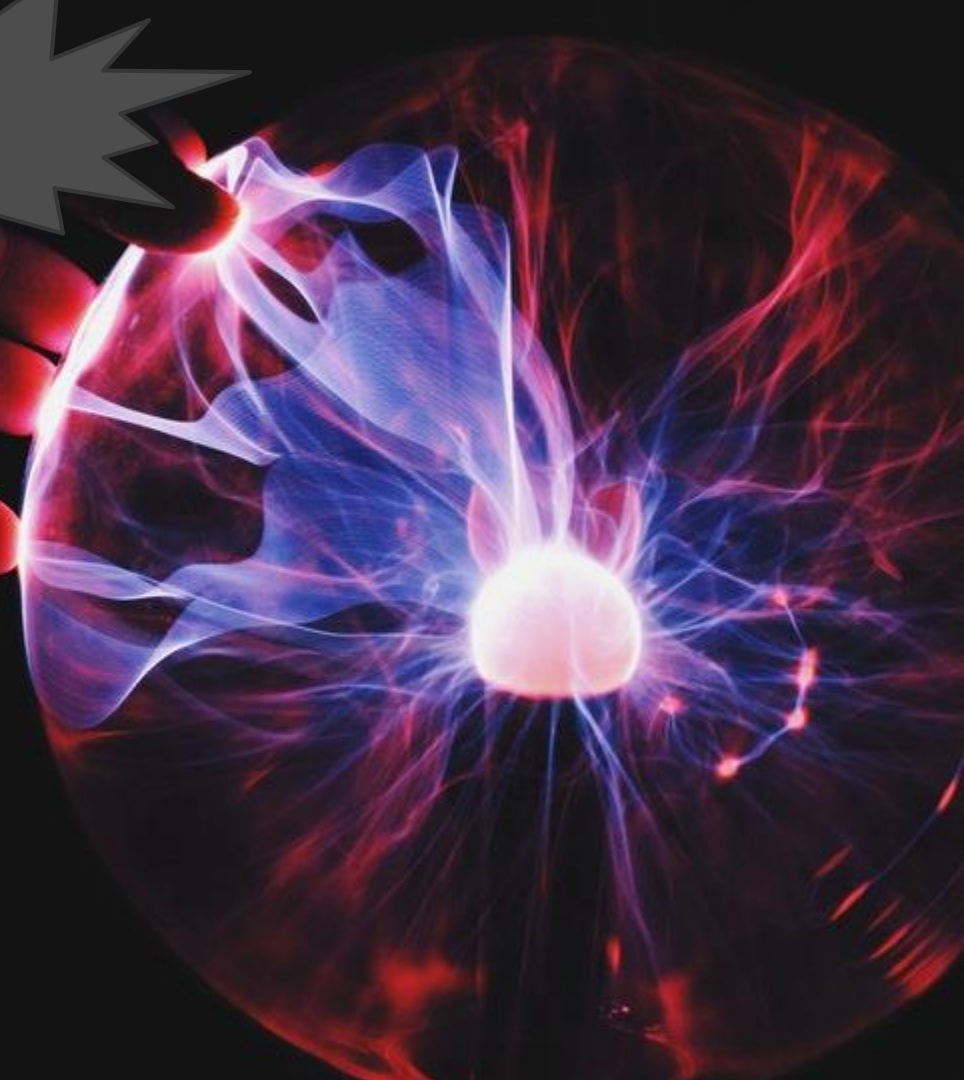


Puppet

Git

Ansible

Rundeck



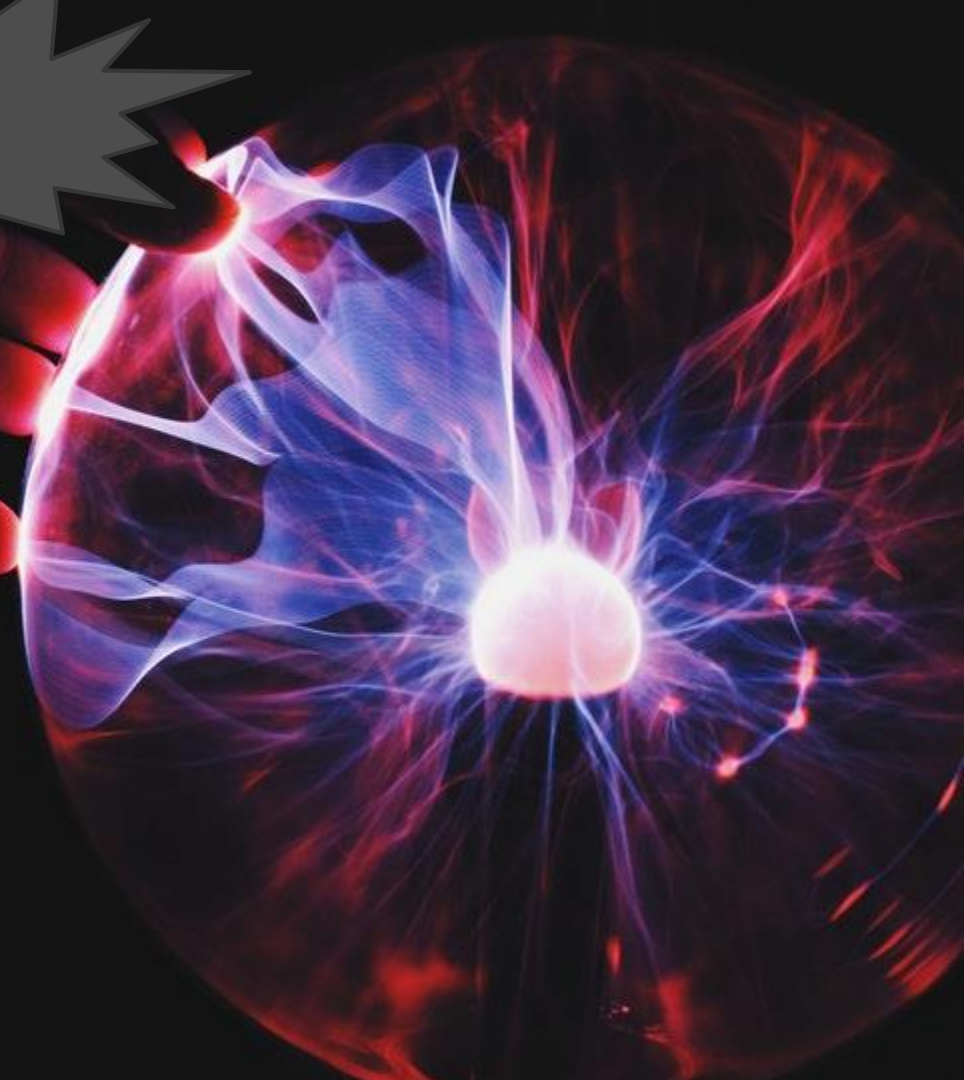
Puppet

Git

Ansible

Rundeck

ELK



Puppet

Git

Ansible

Rundeck

ELK

REST

Now, if your DBAs use those agile methods you are in the wrong room!

But if you enter the DBA office and the DBA is like…

The not-so-agile DBA problem…

- In one word: BASH
- DBA's script are not complex
- Underlying operations are complex
- OS commands everywhere
- Not-so-agile-DBAs do not need anything else

Expectation vs reality

Quick backup before going production:

- You need to ask it at least 4h in advance
- The backup takes ages
- The DBA must check the outcome

Expectation vs reality

Refresh of the test database:

- You need to ask it at least 24h in advance
- “It will not be accessible for the whole next working day”

Expectation vs reality

Killing a session urgently:

- “Urgently is dead”
- “Test your (*) code before going production”

* Put your favorite bad word here

Expectation vs reality

Check why the database is so slow:

- “The database is fine”
- “The database would be just fine without your crappy code”
- Eventually, no action at all

Who might help?

JSON (/ˈdʒeɪsən/)

REST endpoints

- Standard, widely used
- Flexible
- De-coupled
- Tons of libraries and 3rd party integration

REST for DBAs?

REST for DBAs? ORDS!

- Definition in the database
- PL/SQL: DBA friendly
- Powerful
- Damn simple

REST for DBAs? ORDS!

REST for DBAs? ORDS!

INGREDIENT

4 shades of REST

- ~~DEV -> DEV~~
- DEV -> DBA
- DBA -> DEV
- DBA -> DBA

DEV -> DBA

Developer’s endpoint for the DBA

- **Can the app connect to the database?**

Can the app connect to the database?

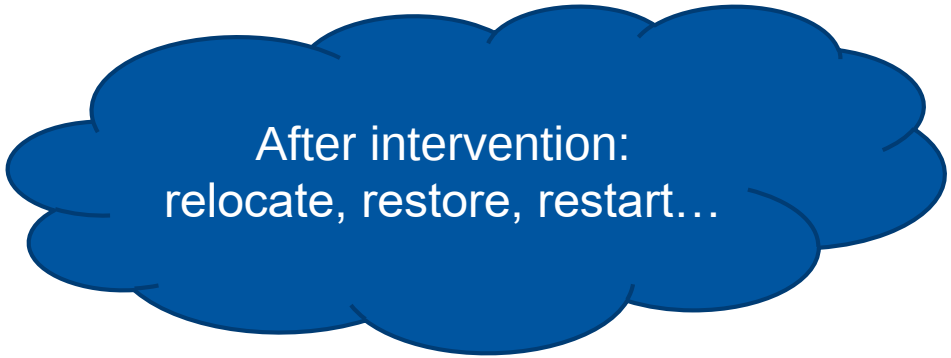
```
$ curl https://portal_prod/check/dbstatus/
{
  "database_error": "ORA-12541: TNS:no listener",
  "database_connection": "error",
  "application": "portal_prod"
}
```

Can the app connect to the database?

```
$ eval $(curl https://.../ |jq --raw-output '. | @sh
"L_DB_CONN=\([.database_connection]);
 L_DB_ERR=\([.database_error])"
')

$ echo $L_DB_CONN
error

$ echo $L_DB_ERR
ORA-12541: TNS:no listener
```

Can the app connect to the database?

```
$ eval $(curl https://.../ |jq --raw-output '. | @sh
"L_DB_CONN=\([.database_connection]);
 L_DB_ERR=\([.database_error])"
')

$ echo $L_DB_CONN
error

$ echo $L_DB_ERR
ORA-12541: TNS:no listener
```



Developer’s endpoint for the DBA

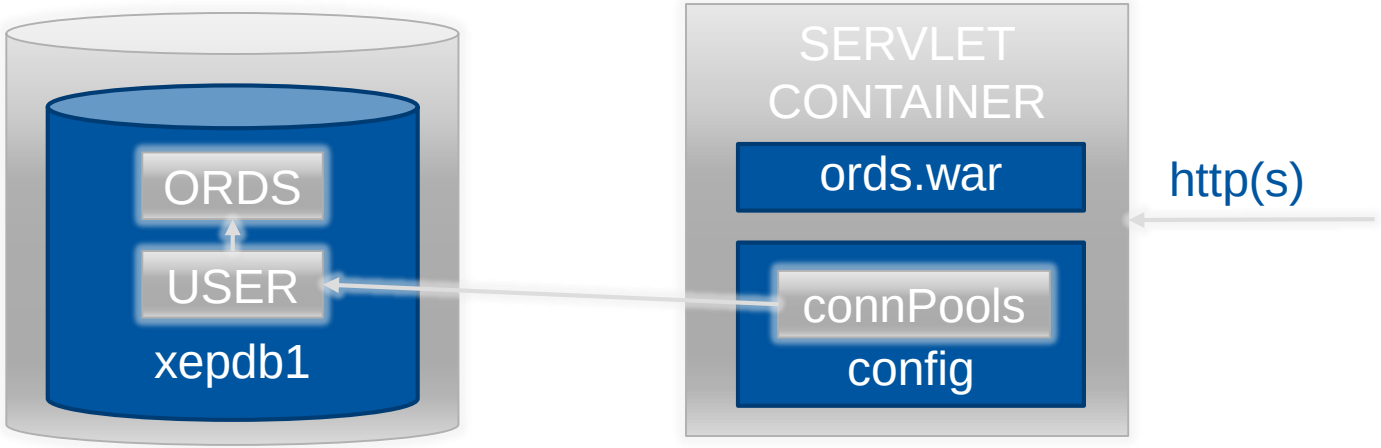
- Can you connect to the database?
- **Put the application in maintenance mode**

Put the application in maintenance mode

```
# put the app in maintenance
curl -X POST \
 --header "Content-Type: application/json“ \
 -d "{\"maintenance\":\"true\"}" \
http://myapp/backend/maintenance_mode

# do maintenance
...
# put the app online
...
```

DBA -> DEV
(ORDS)

ORDS configuration



Schema/module definition

```
BEGIN
  ORDS.ENABLE_SCHEMA(
      p_enabled             => TRUE,
      p_schema              => 'DBAREST',
      p_url_mapping_type    => 'BASE_PATH',
      p_url_mapping_pattern => 'dbarest',
      p_auto_rest_auth      => FALSE);

  ORDS.DEFINE_MODULE(
      p_module_name    => 'dbarest',
      p_base_path      => '/dbarest/',
      p_items_per_page =>  0,
      p_status         => 'PUBLISHED',
      p_comments       => NULL);
```

https://oraweb.cern.ch/ords/xepdb1/**dbarest**/...

DBA’s endpoint for the developer

- **Check backup before deployment**

```
GRANT SELECT ON v_$rman_status TO dbarest;
CONN dbarest/dbarest
BEGIN
ORDS.DEFINE_TEMPLATE(
      p_module_name    => 'dbarest', p_pattern  => 'lastbackup/',
      p_priority       => 0,         p_etag_type      => 'HASH');

  ORDS.DEFINE_HANDLER(
      p_module_name    => 'dbarest',      p_pattern        => 'lastbackup/',
      p_method         => 'GET',          p_source_type    => 'json/query',
      p_items_per_page =>  0,             p_mimes_allowed  => '',
      p_source         =>
'SELECT   operation, status,  object_type,
    TO_CHAR(start_time, ''YYYY/MM/DD:HH24:MI:SS'') as start_time,
    TO_CHAR(end_time, ''YYYY/MM/DD:HH24:MI:SS'') as end_time
   FROM sys.v_$rman_status WHERE start_time=(
     SELECT MAX(start_time) FROM sys.v_$rman_status WHERE  operation = ''BACKUP'')'
);
COMMIT;
END;
```

```
GRANT SELECT ON v_$rman_status TO dbarest;
CONN dbarest/dbarest
BEGIN
ORDS.DEFINE_TEMPLATE(
      p_module_name    => 'dbarest', p_pattern  => 'lastbackup/',
```

```
$ curl https://.../dbarest/lastbackup/
{"items":[
 {"operation":"BACKUP",
 "status":"COMPLETED",
 "object_type":"ARCHIVELOG",
 "start_time":"03/24/2019:12:29:30",
 "end_time":"03/24/2019:12:33:20"}
]}
```

```
',
,
'S
ACKUP'')'
);
COMMIT;
END;
```

```
GRANT SELECT ON v_$rman_status TO dbarest;
CONN dbarest/dbarest
BEGIN
ORDS.DEFINE_TEMPLATE(
      p_module_name    => 'dbarest', p_pattern  => 'lastbackup/',
```

```
$ curl
{"item
 {"ope                                                    ',
 "stat                                                    ,
 "obje
 "star
 "end_
]}
```

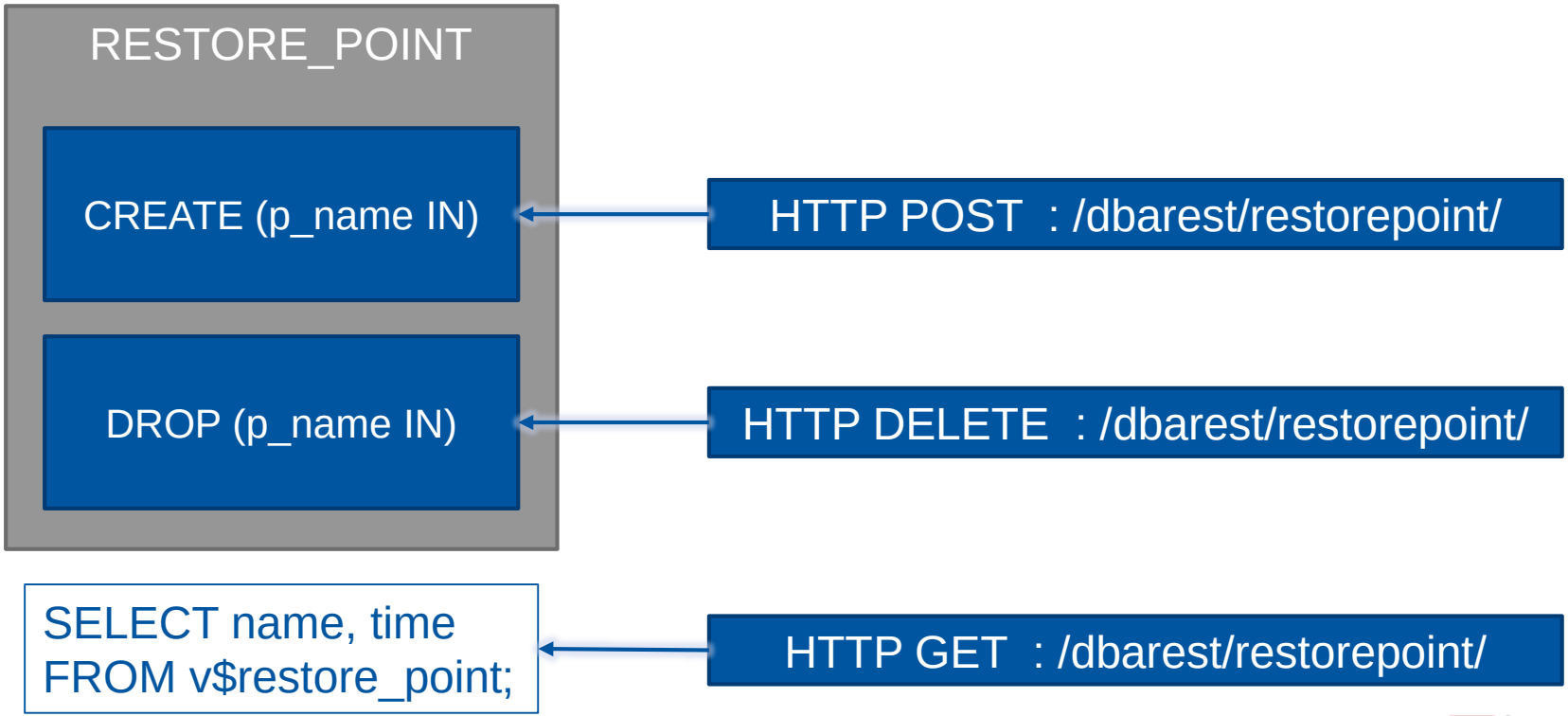
DEMO?

```
ACKUP'')'
);
COMMIT;
END;
```

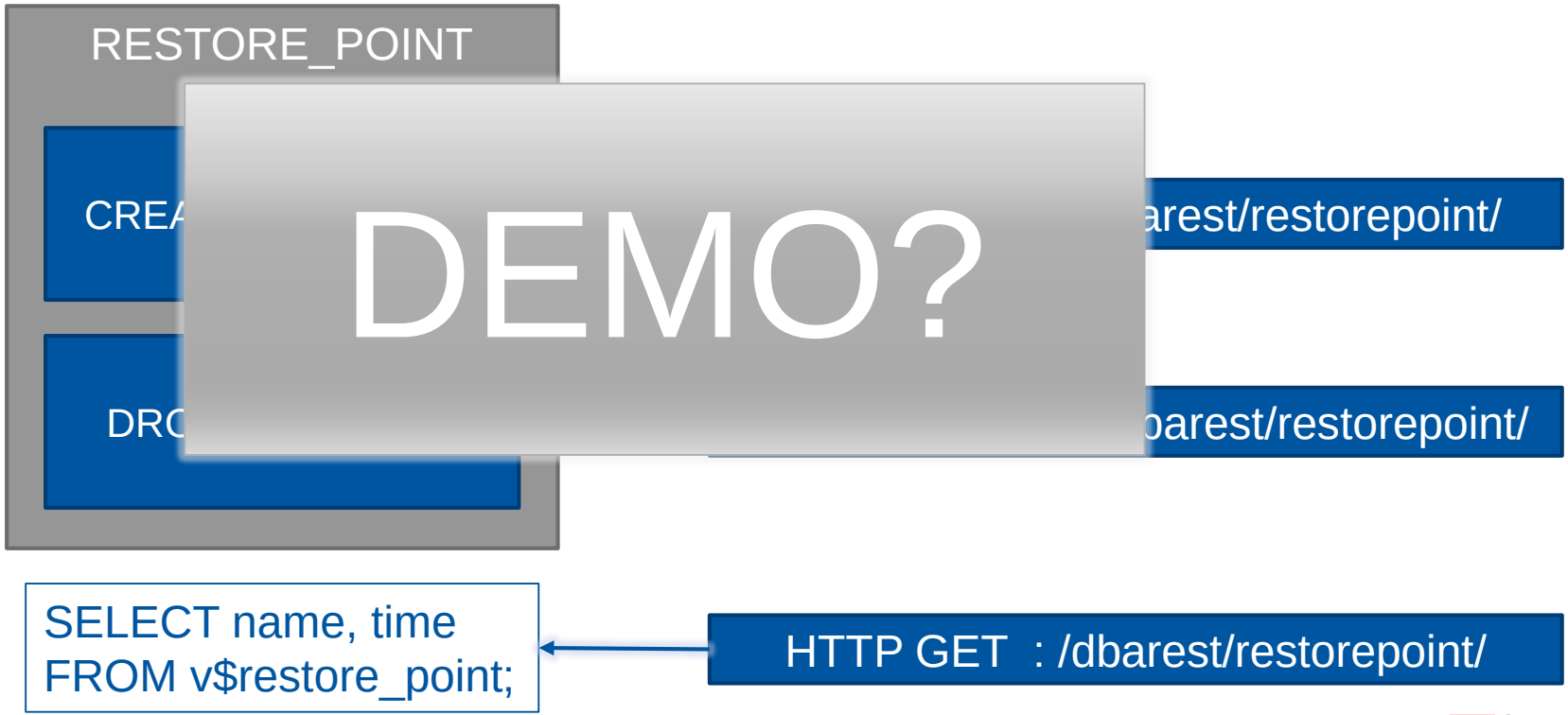
DBA’s endpoint for the developer

- Check backup before deployment
- **Create/delete restore point**

Create/delete restore point



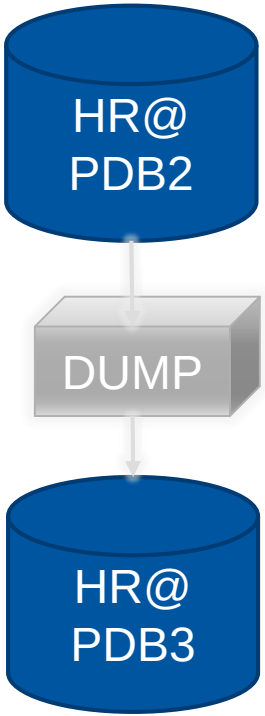
Create/delete restore point



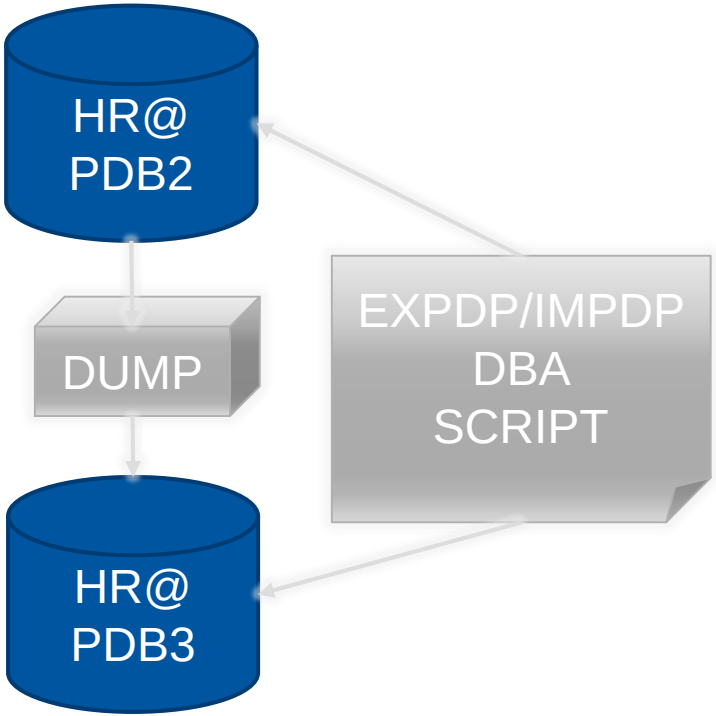
DBA’s endpoint for the developer

- Check backup before deployment
- Create/delete restore point
- **Refresh Schema**

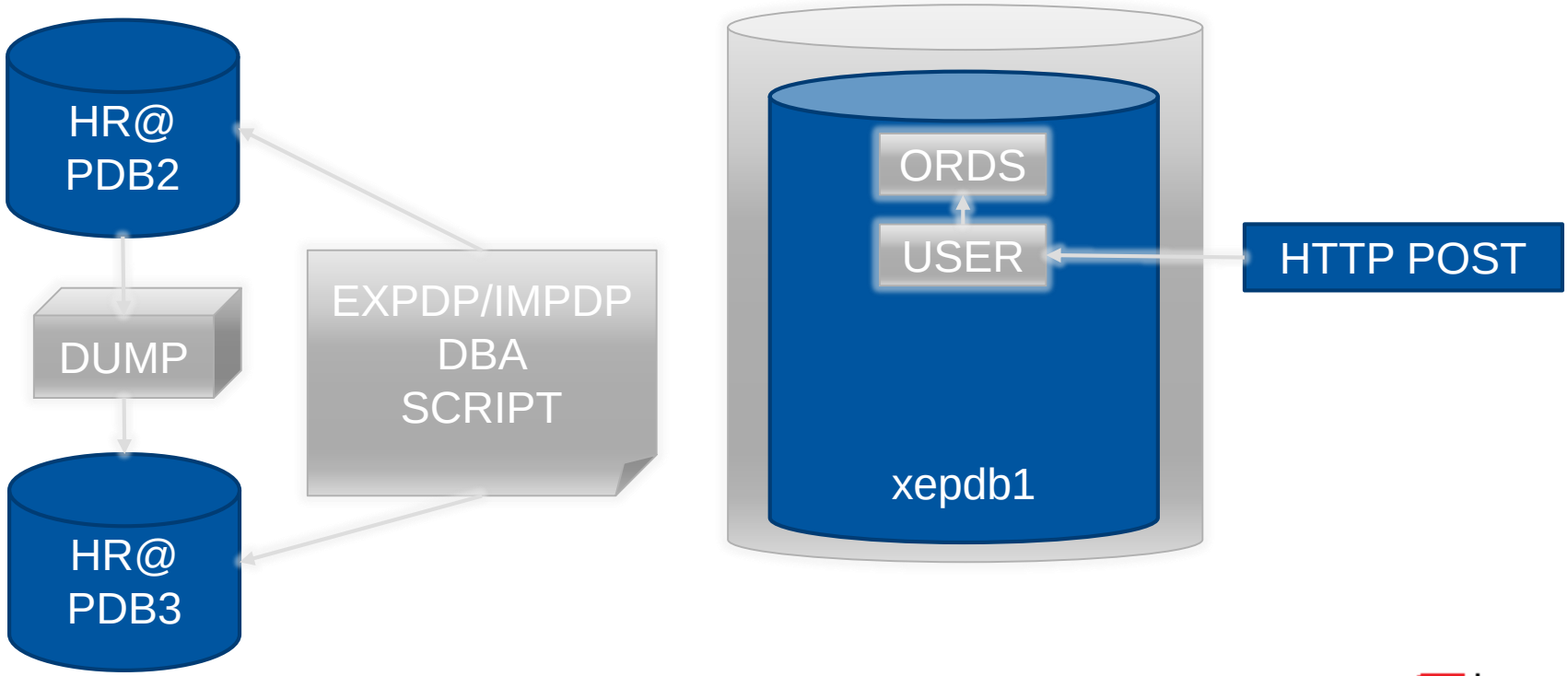
Refresh Schema



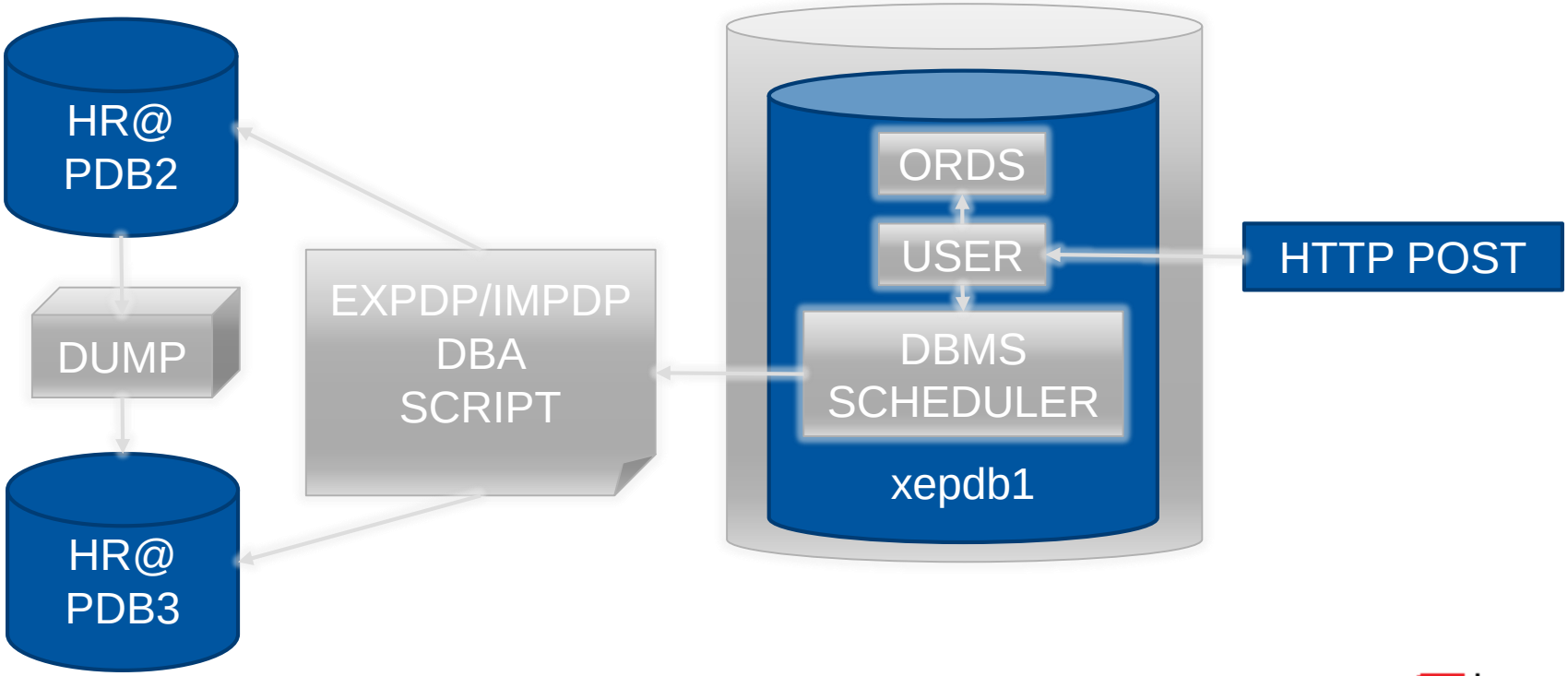
Refresh Schema



Refresh Schema



Refresh Schema



Refresh Schema

DBA’s endpoint for the developer

- Check backup before deployment
- Create/delete restore point
- Refresh Schema
- **Create new users**

DBA’s endpoint for the developer

- Check backup before deployment
- Create/delete restore point
- Refresh Schema
- Create new users
- **Create new databases**

DBA’s endpoint for the developer

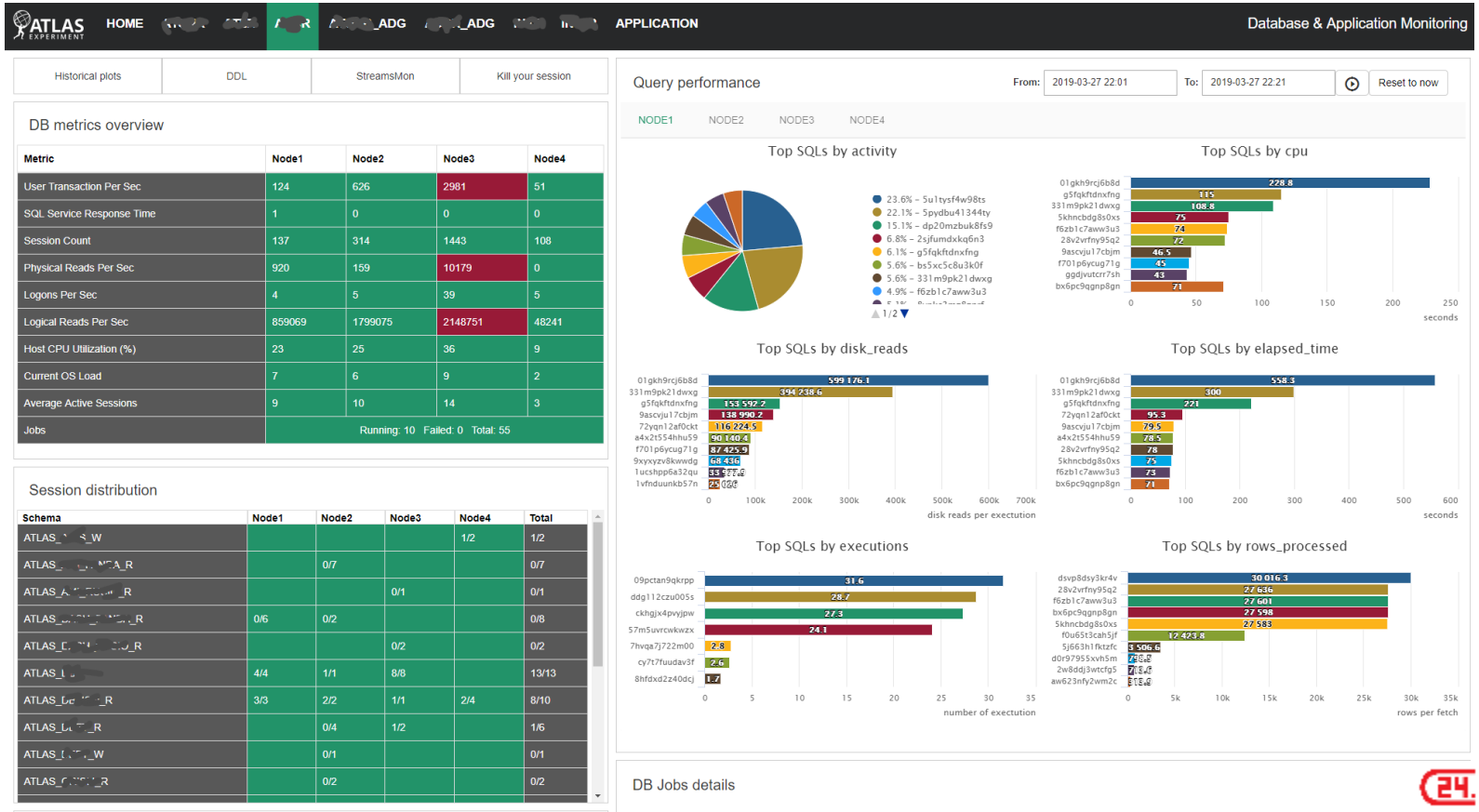
- Check backup before deployment
- Create/delete restore point
- Refresh Schema
- Create new users
- Create new databases
- **Create Guaranteed Restore Points**

DBA -> DBA
(ORDS)

DBA’s endpoint for the DBA

- Collect metrics
- Various metadata, e.g.:
  - Retrieve unsuccessful backup

Real production example @ CERN



Thank you!

Ludovico Caldara - Computing Engineer @CERN, Oracle ACE Director