

Using Oracle GoldenGate in a Multitenant Container Database

by Ahmed Baraka

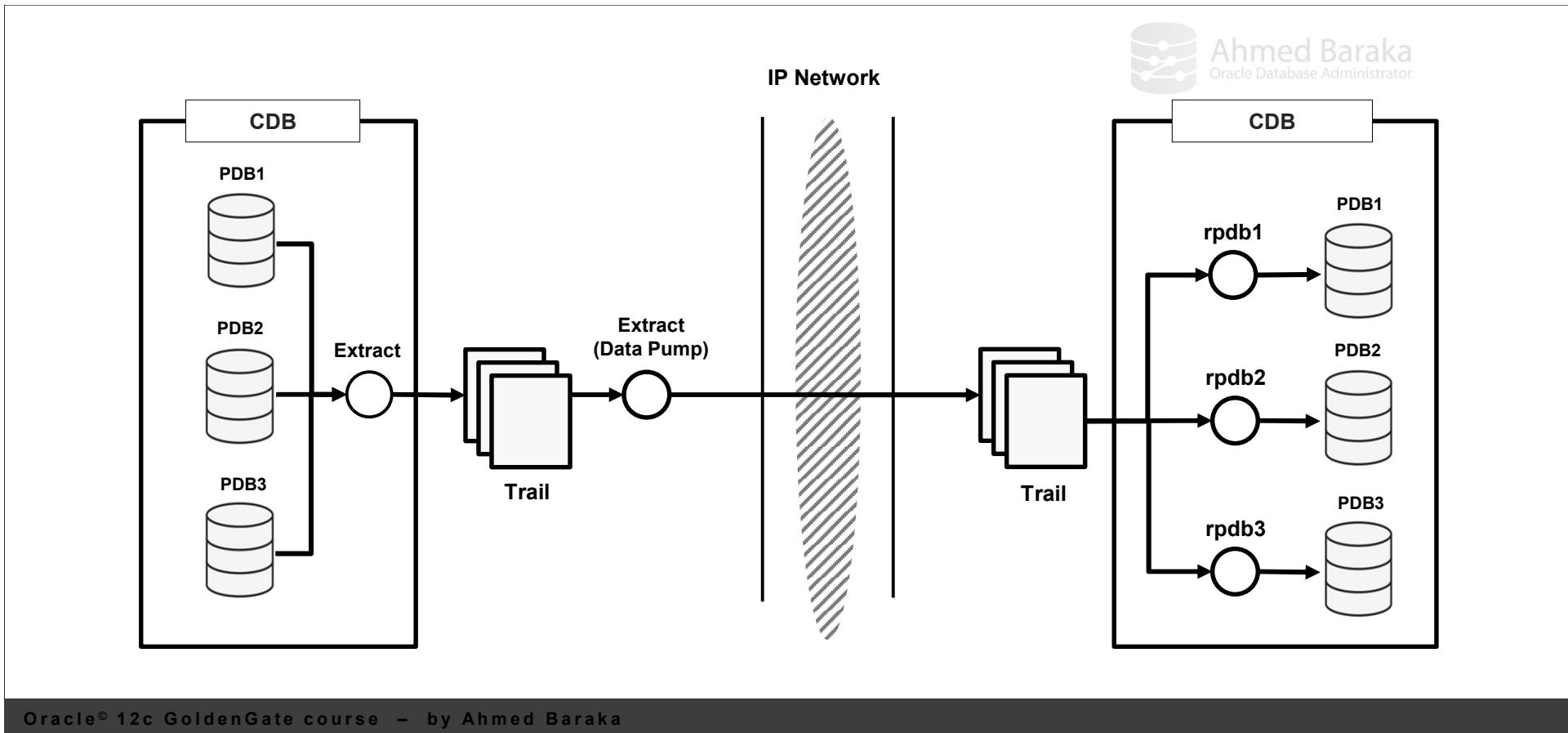
Objectives

By the end of this lecture, you should be able to perform the following:

- Configure a GoldenGate replication in a multitenant database

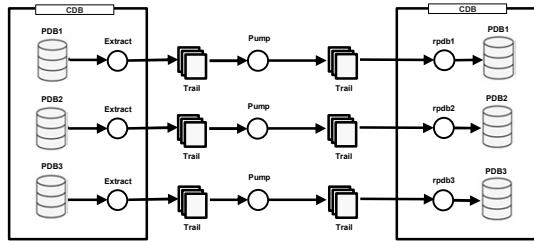


GoldenGate in CDB Databases

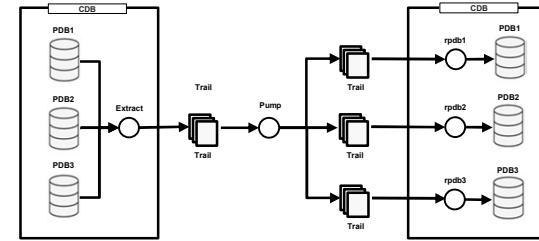


GoldenGate in CDB Databases: Possible Configurations

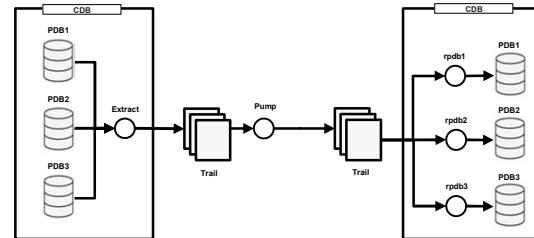
Parallel Extract and Replicat Streams



One Extract and Multiple Trails



One Extract and One Trail



Ahmed Baraka
Oracle Database Administrator

Using Oracle GoldenGate with Pluggable Databases

- The Extract must be configured in the integrated mode
- Extract must connect to the root container (cdb\$root) as a common user (c## user)
- One Extract group **can** capture from multiple PDBs
- Replicat can only connect and apply to one PDB
- A Replicat can read from a trail containing transactions for multiple PDBs
- PDBs with multiple character sets are supported

Creating a Common User for the Capture

- Connect to the root CDB:



```
sqlplus / as sysdba
SQL> CREATE USER C##GGADMIN IDENTIFIED BY ...;
```

- Grant the privileges to the user:

```
exec DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE (
    GRANTEE => 'C##GGADMIN', PRIVILEGE_TYPE => 'CAPTURE',
    GRANT_SELECT_PRIVILEGES => TRUE, DO_GRANTS => TRUE,
    CONTAINER => 'PDB1');

    ...
```

Specifying C## User in Extract Parameter File

```
alter credentialstore add user c##ggadmin password ***
alias c##ggadminogg1

edit param eogg1
...
USERIDALIAS c##ggadminogg1
```



Ahmed Baraka
Oracle Database Administrator

Specifying the Tables and Sequences in the Extract Parameter File

- Specify fully qualified three-part names:

```
pdb1.hr.employees  
pdb1.hr.emp_id_seq
```



- Specify a default pluggable database:

```
SOURCECATALOG pdb1  
TABLE hr.employees;  
SEQUENCE emp_id_seq;  
SOURCECATALOG pdb2  
TABLE hr.*;
```

Creating a Common User for the Apply

- Connect to the root CDB:



```
sqlplus / as sysdba
SQL> CREATE USER C##GGADMIN IDENTIFIED BY ...;
```

- Grant the privileges to the user:

```
exec DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE (
    GRANTEE => 'C##GGADMIN', PRIVILEGE_TYPE => 'APPLY',
    GRANT_SELECT_PRIVILEGES => TRUE, DO_GRANTS => TRUE,
    CONTAINER => 'PDB1');

    ...
```

Further Privileges Required by Common User

- Applicable for C## users used in Extracts and Replicats

```
GRANT CREATE SESSION TO c##ggadmin CONTAINER=all;
GRANT ALTER SYSTEM TO c##ggadmin CONTAINER=all;
GRANT RESOURCE TO c##ggadmin CONTAINER=all;
GRANT CONNECT TO c##ggadmin CONTAINER=all;
GRANT SELECT ANY DICTIONARY TO c##ggadmin ...
GRANT EXECUTE on DBMS_FLASHBACK TO c##ggadmin ...
GRANT LOCK ANY TABLE TO c##ggadmin ...
GRANT ALTER USER TO c##ggadmin ...
```



Further Privileges Based on Process Type

- Further privileges required by Extract user:



Ahmed Baraka
Oracle Database Administrator

```
GRANT SELECT ANY TRANSACTION TO c##ggadmin
CONTAINER=all;
```

- Further privileges required by Replicat user:

```
GRANT INSERT ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT UPDATE ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT DELETE ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT CREATE TABLE TO c##ggadmin CONTAINER=all;
```

Specifying C## User in Replicat Parameter File

```
alter credentialstore add user c##ggadmin password  
oracle alias c##ggadminogg2
```

```
alter credentialstore add user c##ggadmin@pdb1logg2  
password oracle alias c##ggadminnpdb1
```

```
edit params rpdb1
```

```
...
```

```
USERIDALIAS c##ggadminnpdb1
```



Specifying the Tables and Sequences in the Replicat Parameter File

- Specify fully qualified three-part names in the MAP:

```
MAP pdb1.hr.employees, TARGET hr.employees;  
MAP pdb1.hr.emp_id_seq, TARGET hr.emp_id_seq;
```

- Specify a default pluggable database:

```
SOURCECATALOG pdb1  
MAP hr.employees, TARGET 1;  
MAP hr.employee_id_seq, TARGET 1;  
SOURCECATALOG pdb2  
MAP hr.* , TARGET 2;
```

Replicating Sequences in Multitenant Databases

1. (s&d) Create a separate PDB user:



```
ALTER SESSION SET CONTAINER=pdb1;
CREATE USER GGATE IDENTIFIED BY *** DEFAULT
TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA
UNLIMITED ON USERS CONTAINER=CURRENT;
```

2. (s&d) Grant Privileges to it:

```
GRANT CONNECT, RESOURCE, DBA TO GGATE;
```

Replicating Sequences in Multitenant Databases

3. (s&d) Run **sequence** script from the GG home:

```
SQL> @sequence
```

4. (s) Grant execute on **UPDATESEQUENCE** to the C## user:

```
GRANT EXECUTE on GGATE.UPDATESEQUENCE TO c##ggadmin
```

5. (d) Grant execute on **REPLICATESEQUENCE** to the C## user:

```
GRANT EXECUTE on GGATE.REPLICATESEQUENCE TO  
c##ggadmin
```

Replicating Sequences in Multitenant Databases

6. (s) Run the following code (as sysdba) from the root CDB:

```
ALTER TABLE sys.seq$ ADD SUPPLEMENTAL LOG DATA  
  (PRIMARY KEY) COLUMNS;
```

7. (d) Grant the following privileges to the c## account (*):

```
GRANT SELECT ANY SEQUENCE TO C##GGADMIN  
CONTAINER=ALL;  
  
GRANT ALTER ANY SEQUENCE TO C##GGADMIN  
CONTAINER=ALL;
```

(*) Not part of the standard procedure.

Replicating Sequences in Multitenant Databases

8. (s&d) Set the GGSHEMA in the GLOBALS:



Ahmed Baraka
Oracle Database Administrator

```
ggsci> EDIT PARAMS ./GLOBALS  
GGSHEMA ggate
```

9. (s) Flush the sequences after starting the Extract:

```
ggsci> start extract eogg1  
ggsci> flush sequence pdb1.hr.*
```

Summary

In this lecture, you should have learnt how to perform the following:

- Configure a GoldenGate replication in a multitenant database

