

Alfresco 4.2.c - 201605GA

Migration use case

Mikel Asla
developer at keensoft



The case

Migrate two different Alfresco 4.2.c instances into one instance of Alfresco 201605-GA

The case

Two different production environments

Alfresco 4.2.f CE with Oracle 11g (instance 1)

- Integrated with an external expedient management system (SOAP)

Alfresco 4.2.f CE with Mysql 5.6 (instance 2)

- Integrated with organization's portals and internal applications (SOAP)
- Also used for collaboration through Share (about 10-15 concurrent users)

The case

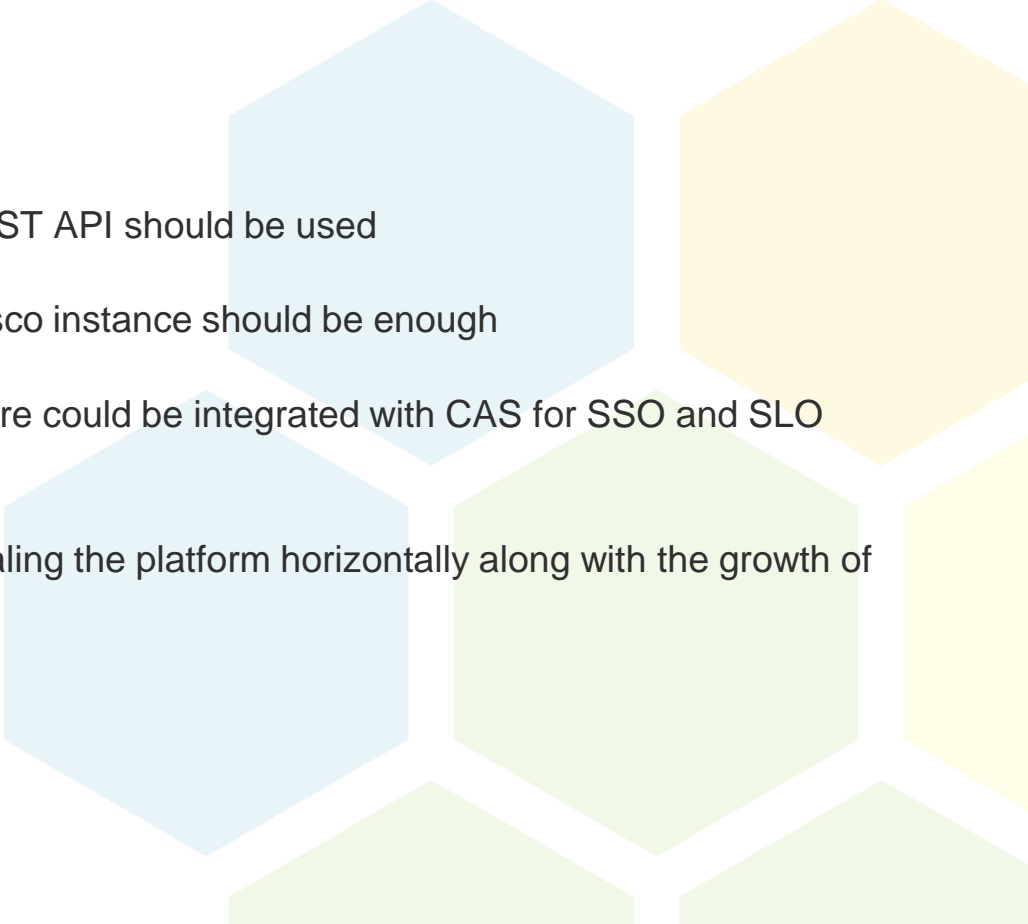
- The main use of Alfresco is to maintain and make available organization's sensitive data
- Alfresco integrates with organization using SOAP web services
- Collaboration trend to be more important in last years
- Despite most internal applications have authentication delegated to central CAS Server, Open LDAP authentication and synchronization is used for Share access.
- Alfresco repository, Share and LibreOffice are running standalone along with Lucene in-transaction indexing in a virtual VMWARE box

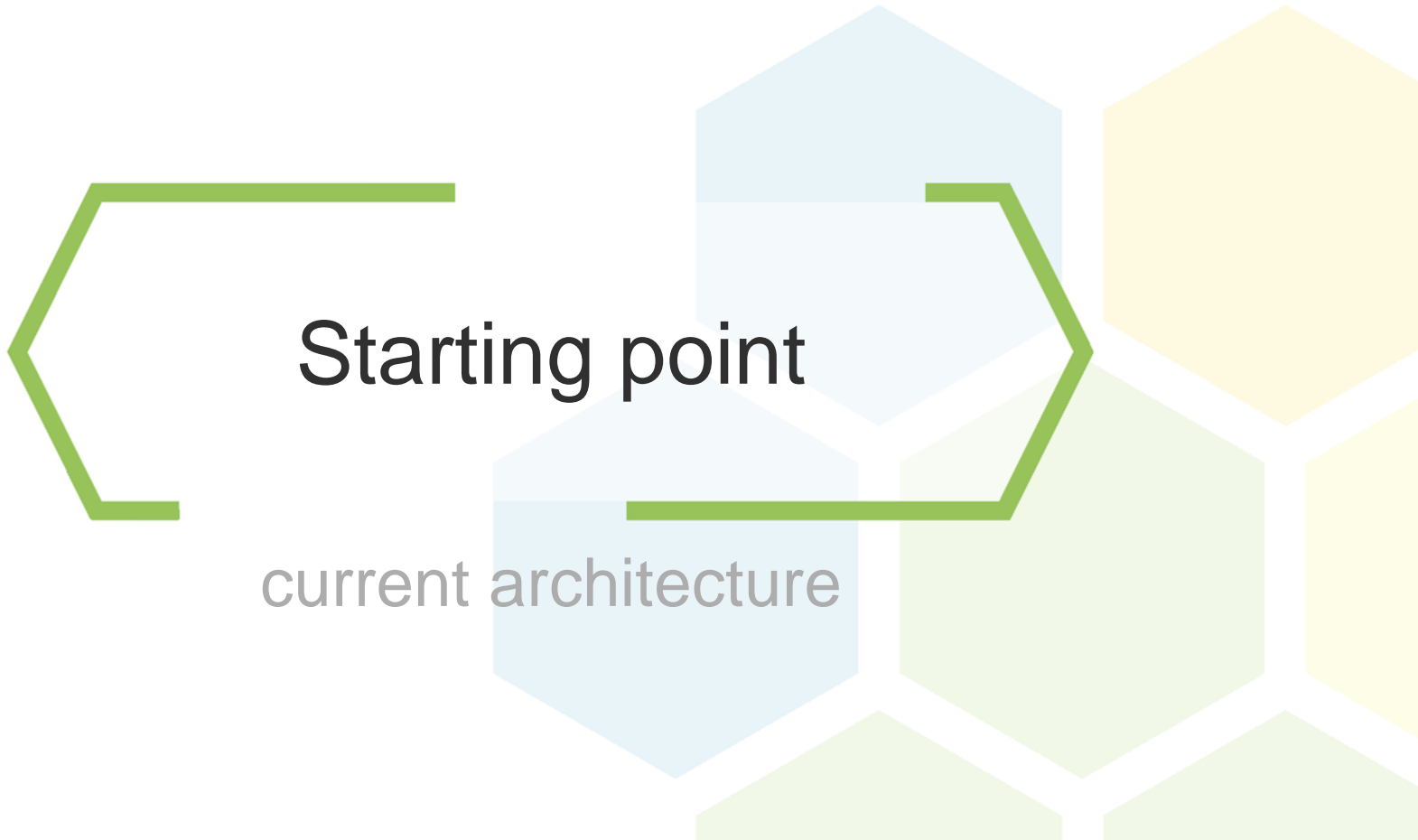


At first sight

in the end it was worth it

At first sight

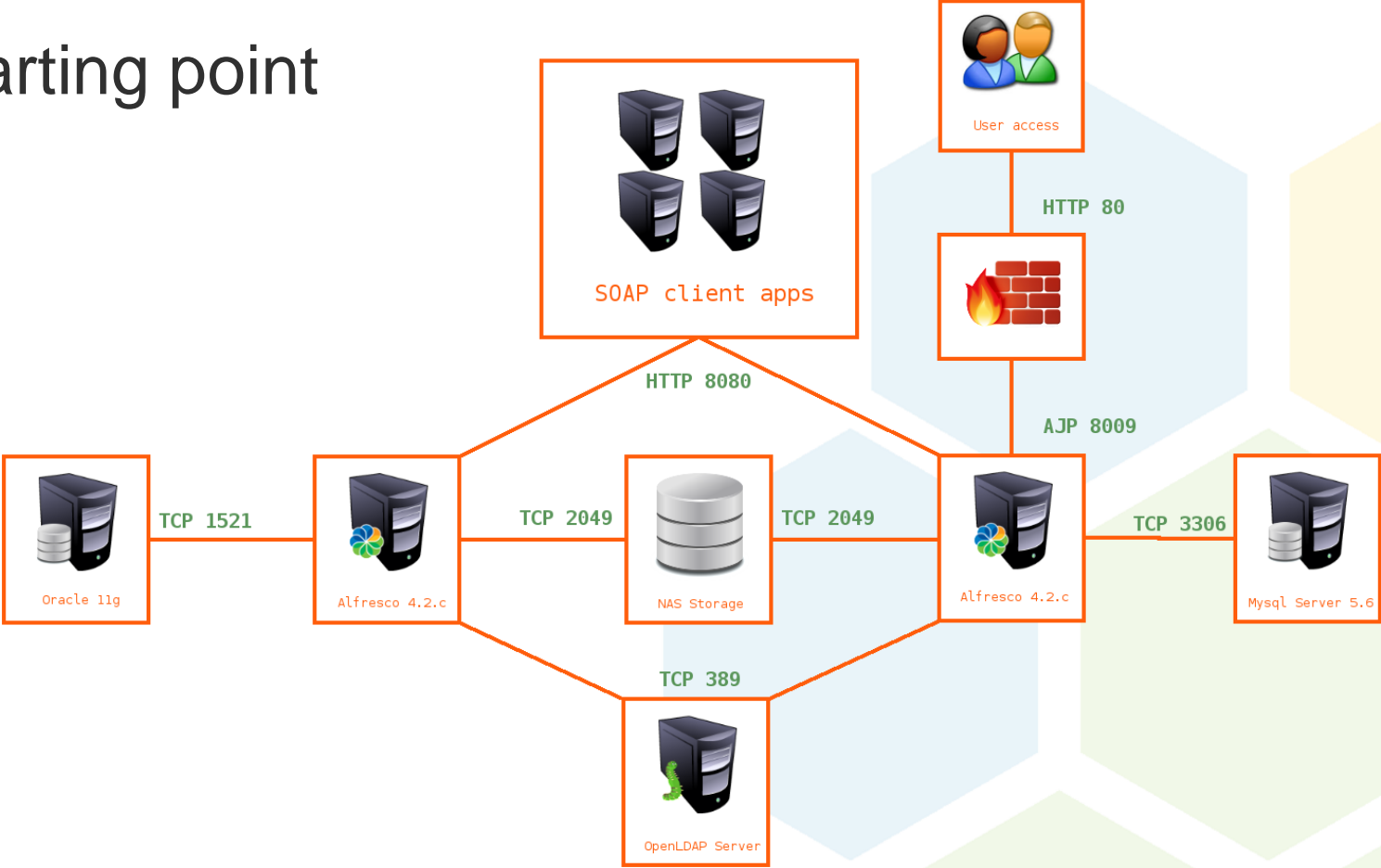
- 3th-party addons to talk to the database?
 - If interoperability is the centre point CMIS REST API should be used
 - With less than 50 concurrent users one Alfresco instance should be enough
 - In order to improve end user experience, Share could be integrated with CAS for SSO and SLO support
 - A more distributed deployment could help scaling the platform horizontally along with the growth of Share usage
- 



Starting point

current architecture

Starting point



Starting point

Instance 1

- Alfresco 4.2.f CE
 - Centos 5.7
 - 2x CPU Xeon E5649 @ 2.53GHz
 - 12 Gb RAM
 - 1 GB Oracle 11g database
 - 7 Gb Lucene indexes
 - 30 Gb repository
 - 7,2 million alf_node_properties records

Instance 2

- Alfresco 4.2.f CE
 - Centos 5.7
 - 2x CPU Xeon E5649 @ 2.53GHz
 - 12 Gb RAM
 - 600 MB Mysql 5.6 database
 - 1,2 Gb lucene indexes
 - 40 Gb repository
 - 0,7 million alf_node_properties records



Destination

post-migration architecture

Destination



Destination

- Alfresco 201605-GA CE instance

4x CPU 2.6GHz

12 Gb RAM

Postgresql 9.4 database

~70 Gb repository

~8 Million entries alf_node_properties

- Solr4 & Share 201605-GA CE instance

4x CPU 2.6 GHz

8 Gb RAM

~7 Gb Solr 4 indexes

- LibreOffice 5.2 instance

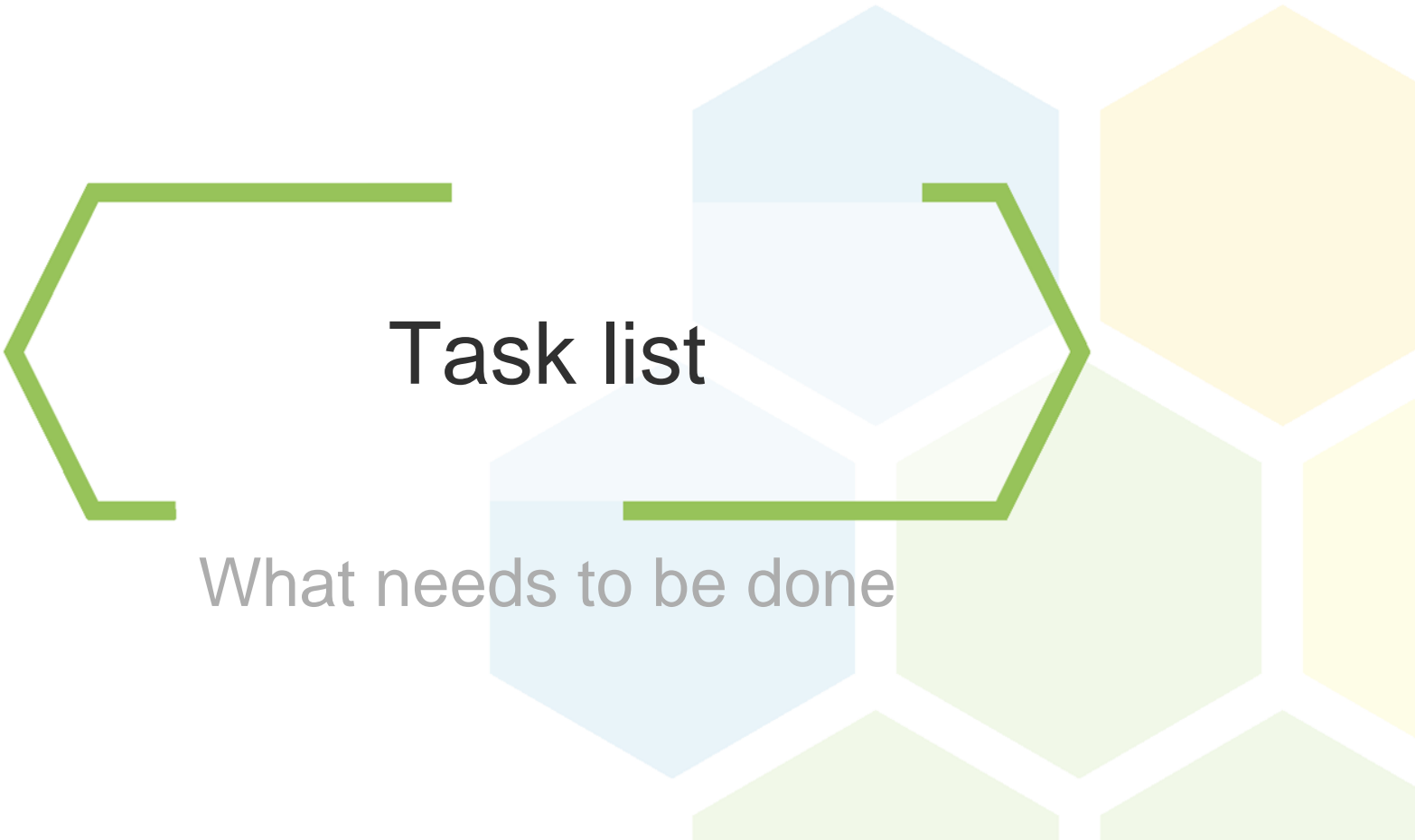
2x CPU 2.6 GHz

Gb RAM

Destination

Post migration architecture

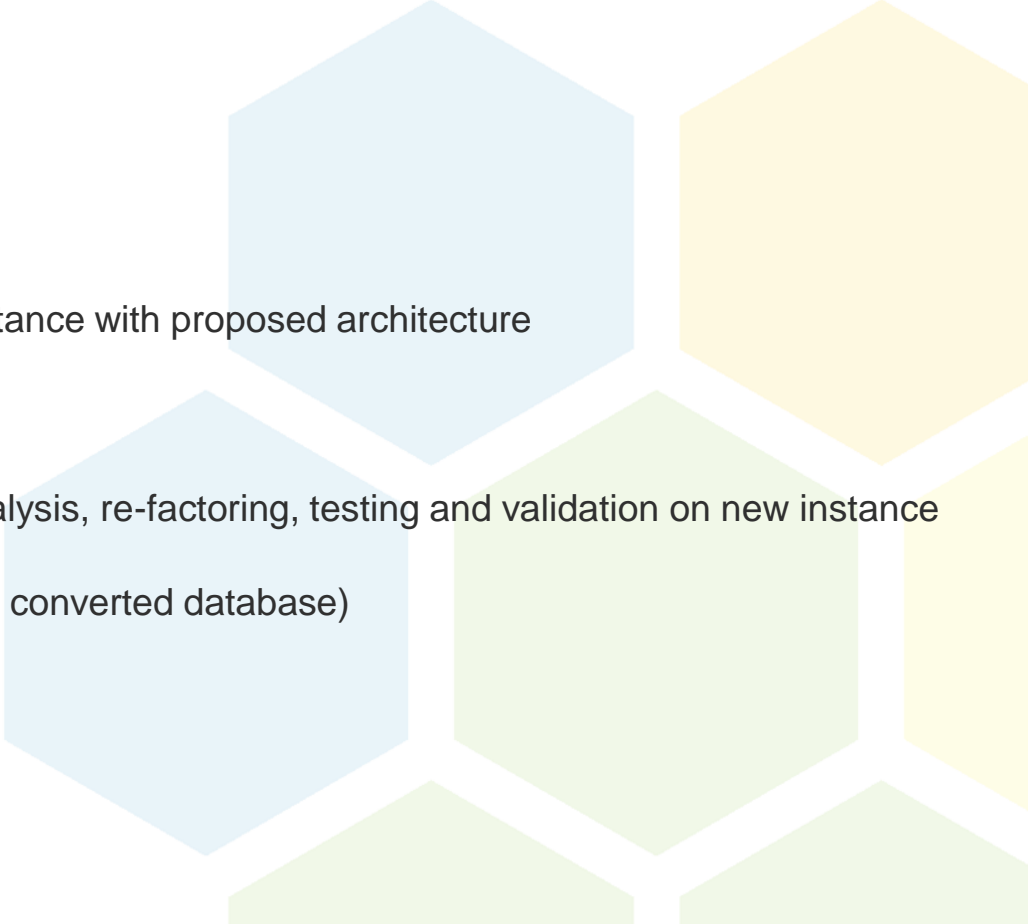
- Dedicated server for Alfresco repository and PostgreSQL database
 - Isolated data silo
 - Correct backup policies becomes key
 - Unique integration point for other applications (CMIS)
- Dedicated server for Solr 4 and Share
 - Solr memory consumption monitoring is encouraged
 - As collaboration grows Share migration to a dedicated server is easy
 - Authentication is delegated to Central Authentication Service
- Dedicated server for LibreOffice 5.2
 - As collaboration grows transformations do too, monitor/watchdog the process




Task list

What needs to be done

The task list I

- Full backup of production instance 1
 - Oracle to PostgreSQL database conversion
 - Alfresco 201605-GA installation in a new instance with proposed architecture
 - New instance validation
 - Instance 1 and instance2 customizations analysis, re-factoring, testing and validation on new instance
 - Instance 1 restoring on new instance (repo + converted database)
 - New instance bootstrapping
- 

The task list II

- Client applications re-factoring using CMIS
 - Testing
 - Go live with new instance (only instance 1 migrated)
 - Instance 2 ACP export (integrated applications one at a time)
 - New instance ACP import (with node UUID preservation)
 - Share and CAS integration (Ian Wright's github project, <https://github.com/wrighting/alfresco-cas>)
 - Instance 2 Sites migration (share-import-export tool)
 - Testing + testing + testing
- 



Database transformation

Oracle 11g to PostgreSQL 9.4

Ora2pg

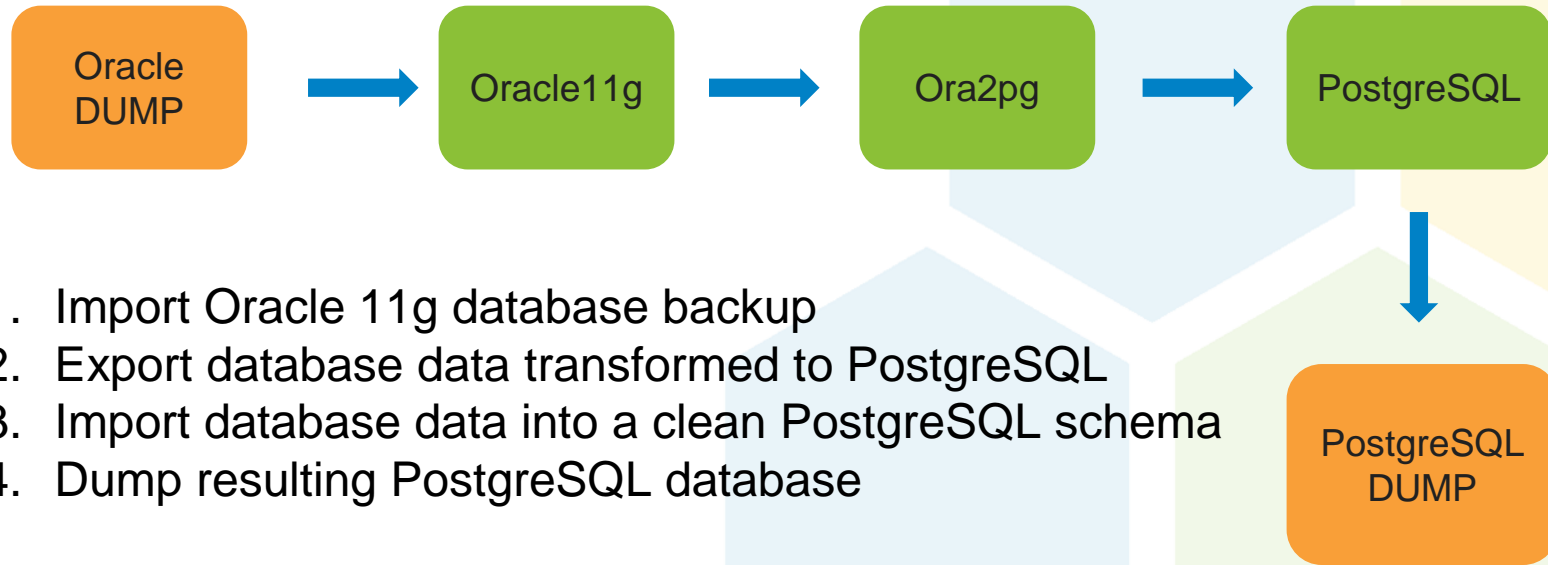
<http://ora2pg.darold.net/>

Ora2Pg is a free tool used to migrate an Oracle or MySQL database to a PostgreSQL compatible schema. It connects your Oracle database, scan it automatically and extracts its structure or data, it then generates SQL scripts that you can load into your PostgreSQL database.



ora2pg

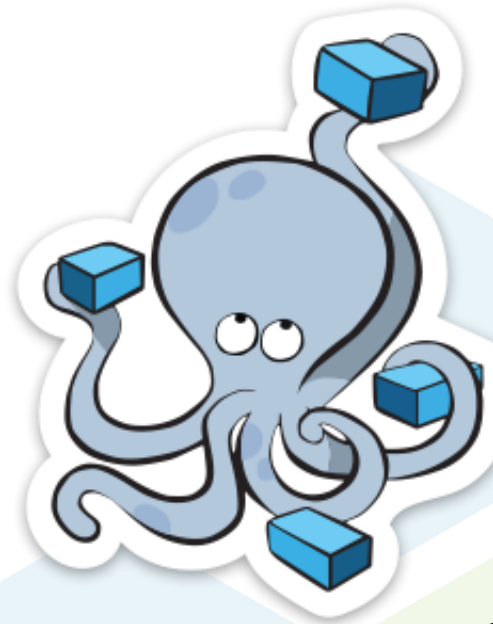
Transformation process



1. Import Oracle 11g database backup
2. Export database data transformed to PostgreSQL
3. Import database data into a clean PostgreSQL schema
4. Dump resulting PostgreSQL database

The docker project

<https://github.com/mikelasla/database-transformations>



ora2pg



The docker project

ora2pg
container



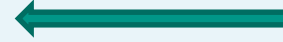
Volume for
output result



Compose file



```
▶ 201605-postgres
▶ 4.2.c-oracle
▼ ora2pg
  ▼ ora2pg
    ▶ assets
      Dockerfile
  ▼ oracle11g
    ▶ assets
      Dockerfile
  ▼ out
    .gitignore
  ▼ postgres
    Dockerfile
    .gitignore
    docker-compose.yml
    update_sequences.sql
▶ postgres-dump-splitter
```



Oracle 11g
container

ora2pg
container

ora2pg

docker-compose.yml

```
1  version: "3"
2  services:
3    ora2pg:
4      build: ora2pg
5      depends_on:
6        - postgres
7        - oracle11g
8      links:
9        - postgres
10       - oracle11g
11     volumes:
12       - ./out:/home/ora2pg/workdir/out
13
14     oracle11g:
15       build: oracle11g
16
17     postgres:
18       build: postgres
19       depends_on:
20         - oracle11g
21       environment:
22         - POSTGRES_DB=alfresco
23         - POSTGRES_USER=alfresco
24         - POSTGRES_PASSWORD=alfresco
25
```

ora2pg

Dockerfile

```
FROM centos:centos7
RUN set -x && yum update -y \
    && yum -y localinstall https://download.postgresql.org/pub/repos/yum/9.4/redhat/rhel-7-x86_64/pgdg-centos94-9.4-2.noarch.rpm \
    && yum install -y \
        gcc \
        libaio \
        make \
        perl \
        perl-ExtUtils-MakeMaker \
        perl-CPAN \
        postgresql94 \
        telnet \
        unzip \
        vim \
        wget
    && yum clean all

COPY assets/instantclient-basic-linux.x64-11.2.0.4.0.zip /tmp/instantclient-basic-linux.x64-11.2.0.4.0.zip
COPY assets/instantclient-sdk-linux.x64-11.2.0.4.0.zip /tmp/instantclient-sdk-linux.x64-11.2.0.4.0.zip
COPY assets/instantclient-sqlplus-linux.x64-11.2.0.4.0.zip /tmp/instantclient-sqlplus-linux.x64-11.2.0.4.0.zip
```

ora2pg

Dockerfile

```
RUN set -x \  
    && mkdir /opt/oracle \  
    && for zip in /tmp/*.zip ; do unzip $zip -d /opt/oracle ; rm $zip ; done \  
    && echo "/opt/oracle/instantclient_11_2/" > /etc/ld.so.conf.d/oracle.conf \  
    && ldconfig  
  
ENV ORACLE_HOME /opt/oracle/instantclient_11_2  
ENV LD_LIBRARY_PATH $LD_LIBRARY_PATH:$ORACLE_HOME  
ENV PATH $PATH:$ORACLE_HOME  
  
RUN set -x \  
    && wget https://github.com/darold/ora2pg/archive/v16.2.tar.gz \  
    && tar xzf v16.2.tar.gz \  
    && cd ora2pg-16.2 \  
    && perl Makefile.PL \  
    && make && make install \  
    && rm -rf v16.2.tar.gz ora2pg-16.2  
  
RUN set -x \  
    && perl -MCPAN -e 'install DBI' \  
    && perl -MCPAN -e 'install DBD::Oracle'
```


ora2pg

Dockerfile

```
RUN set -x \  
    && useradd -ms /bin/bash ora2pg \  
    && ora2pg --project_base /home/ora2pg --init_project workdir  
  
COPY assets/ora2pg.conf.sample /etc/ora2pg/ora2pg.conf  
COPY assets/ora2pg.conf.sample /home/ora2pg/workdir/config/ora2pg.conf  
COPY assets/create_schema_ /home/ora2pg/workdir/create_schema_  
COPY assets/*.sh /home/ora2pg/workdir/  
COPY assets/entrypoint.sh /home/ora2pg/workdir/entrypoint.sh  
  
RUN chmod +x /home/ora2pg/workdir/entrypoint.sh  
  
RUN set -x \  
    && echo "postgres:5432:alfresco:alfresco:alfresco" > /home/ora2pg/.pgpass \  
    && chmod 0600 /home/ora2pg/.pgpass \  
    && chown -R ora2pg:ora2pg /home/ora2pg  
  
WORKDIR /home/ora2pg/workdir  
USER ora2pg  
ENTRYPOINT ["/home/ora2pg/workdir/entrypoint.sh"]  
CMD ["run_export_data"]
```

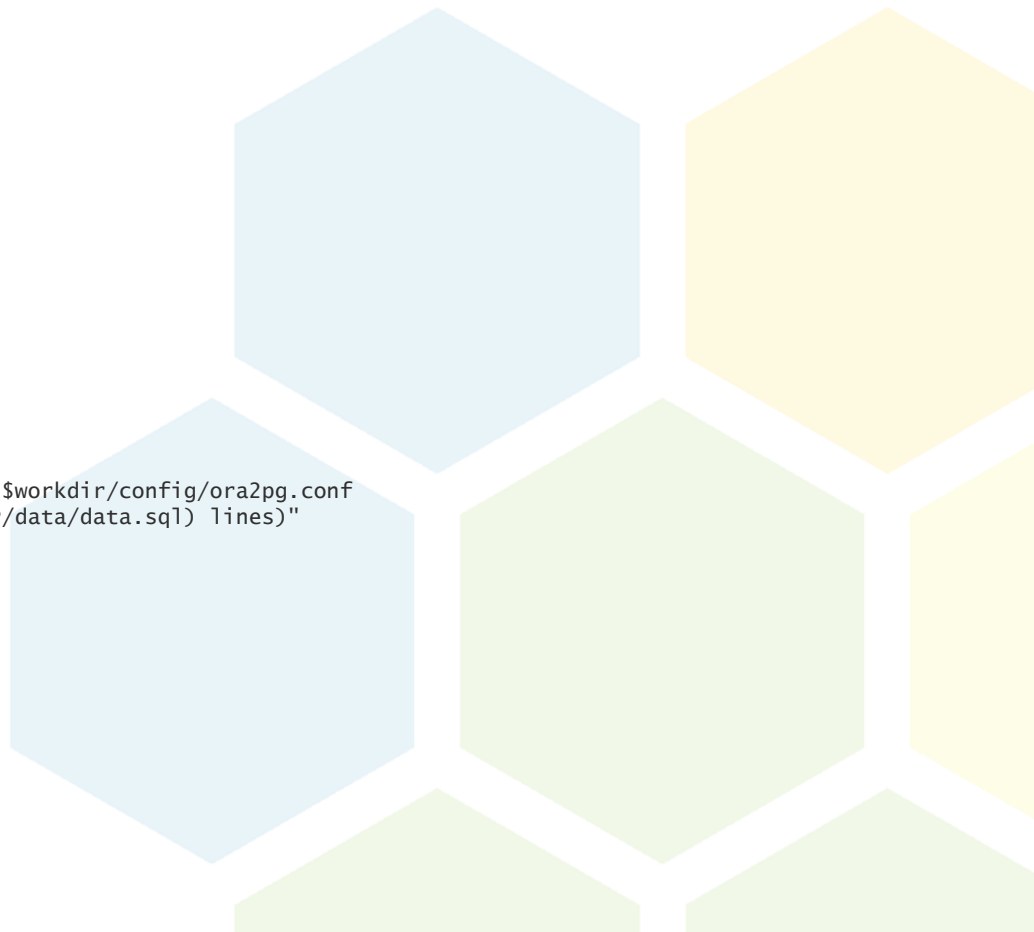
ora2pg

Entrypoint

```
#!/bin/bash

workdir=/home/ora2pg/workdir
host=oracle11g
port=1521

if [ "$1" == "run_export_data" ]
then
    bash -x clean.sh
    echo "Export Data from oracle11g"
    ora2pg -t COPY -o data.sql -b $workdir/data -c $workdir/config/ora2pg.conf
    echo "Import Data to postgres ($(wc -l $workdir/data/data.sql) lines)"
    bash do_dump.sh
else
    exec "$@"
fi
```



ora2pg

do_dump.sh

```
#!/bin/bash

set -ue

host=postgres
user=alfresco
database=alfresco

_ARGS="-h $host -U $user -d $database"
_PSQL=$(which psql)

if [ "$_PSQL" == "" ]
then
    echo "no psql found"
    exit 1
fi

# EXEC SQL FILE
_sql_f() {
    $_PSQL $_ARGS -f $1
}

# EXEC SQL COMMAND
_sql_c() {
    $_PSQL $_ARGS -c "$1"
}
```



ora2pg

do_dump.sh

```
echo "-- clean schema"
_sql_c "drop schema public cascade;"
_sql_c "create schema public authorization $user;"
echo

echo "-- Create tables and sequences"
_sql_f create_schema_/tables.sql
_sql_f create_schema_/sequences.sql
echo

echo "-- Data tranformations"
# Remove junk tables not in destination schema
for s in alf_attributes alf_audit_config alf_audit_date alf_audit_source alf_global_attributes alf_list_attribute_entries
alf_map_attribute_entries
do
    sed -i "/^.*$s.*$/Id" data/data.sql
done
```

ora2pg

do_dump.sh

```
# Remove sequence update statements, we will update them at the end
sed -i -e '/SEQUENCE/{w data/update_sequences.sql' -e 'd}' data/data.sql

echo "-- Import data"
_sql_f data/data.sql

echo "-- Delete tagScope entries (alf_lock , alf_lock_resource)"
_sql_c "delete from alf_lock_resource;"
_sql_c "delete from alf_lock;"

echo "-- Create indexes and constraints"
_sql_f create_schema_/index_constraints.sql

echo "-- Update sequences"
sed -i 's,alf_prop_ser_value_seq,alf_prop_serializable_value_seq,g' data/update_sequences.sql
_sql_f data/update_sequences.sql

echo "-- Dump resulting database"
TIMESTAMP=$((date +%s%N)/1000000)
pg_dump $_ARGS -O --format plain --file out/alfresco_${TIMESTAMP}_postgresql.sql --verbose --encoding UTF8
```

ora2pg

Configuration (ora2pg.conf)

```
ORACLE_HOME /opt/oracle
ORACLE_DSN dbi:Oracle:host=oracle11g;sid=xe
ORACLE_USER ALFRESCO
ORACLE_PWD alfresco
```

```
EXPORT_SCHEMA 0
SCHEMA ALFRESCO
CREATE_SCHEMA 0
COMPILE_SCHEMA 0
PG_SCHEMA public
TYPE TABLE
DISABLE_COMMENT 0
ALLOW ACT_.* ALF_.* AVM_.* JBPM_.*
```

```
REPLACE_AS_BOOLEAN alf_child_assoc:is_primary alf_access_control_entry:allowed jbpm_processdefinition:isterminationimplicit_
jbpm_token:isabletoactivateparent_ jbpm_token:isterminationimplicit_ jbpm_token:issuspended_ alf_tenant:enabled
jbpm_job:issuspended_ jbpm_job:isexclusive_ jbpm_node:isasync_ jbpm_node:isasyncexcl_ jbpm_node:createtasks_ jbpm_node:endtasks_
jbpm_processinstance:issuspended_ act_ru_execution:is_active_ act_ru_execution:is_concurrent_ act_ru_execution:is_scope_
act_ru_execution:is_event_scope_ alf_access_control_list:latest alf_access_control_list:inherits alf_access_control_list:is_versioned
alf_access_control_list:requires_version jbpm_task:isblocking_ jbpm_task:issignalling_ jbpm_taskinstance:iscancelled_
jbpm_taskinstance:issuspended_ jbpm_taskinstance:isopen_ jbpm_taskinstance:issignalling_ jbpm_taskinstance:isblocking_
alf_applied_patch:was_executed alf_applied_patch:succeeded alf_attributes:bool_value alf_attributes:byte_value act_ru_job:exclusive_
act_re_procdef:has_start_form_key_ jbpm_action:ispropagationallowed_ jbpm_action:isasync_ act_ge_bytearray:generated_
avm_node_properties:multi_valued avm_node_properties:boolean_value avm_store_properties:multi_valued
avm_store_properties:boolean_value avm_nodes:is_root alf_node_properties:boolean_value
```

ora2pg

Use

```
$ docker-compose up -d postgres oracle11g
```

```
$ docker-compose ps
```

Name	Command	State	Ports
ora2pg_oracle11g_1	/bin/sh -c /start.sh	Up	1521/tcp, 8080/tcp
ora2pg_postgres_1	docker-entrypoint.sh postgres	Up	5432/tcp

```
$ docker-compose logs -f oracle11g
```

```
oracle11g_1 | Starting Oracle Net Listener.
```

```
oracle11g_1 | Starting Oracle Database 11g Express Edition instance.
```

```
oracle11g_1 |
```

```
oracle11g_1 | Database init...
```

```
...  
oracle11g_1 | Starting "SYSTEM"."SYS_IMPORT_FULL_01": system/***** dumpfile=alfresco.dmp full=yes
```

```
...
```

```
oracle11g_1 |
```

```
oracle11g_1 | Import done!
```

ora2pg

Use

```
$ docker-compose up -d ora2pg
```

```
$ docker-compose ps
```

Name	Command	State	Ports
ora2pg_ora2pg_1	/home/ora2pg/workdir/entry ...	Up	
ora2pg_oracle11g_1	/bin/sh -c /start.sh	Up	1521/tcp, 8080/tcp
ora2pg_postgres_1	docker-entrypoint.sh postgres	Up	5432/tcp

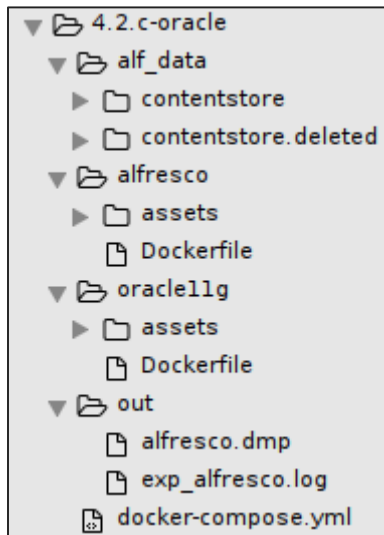
```
$ docker-compose logs -f ora2pg
```

```
ora2pg_1      | Ora2Pg version: 16.2
ora2pg_1      | Trying to connect to database: dbi:Oracle:host=oracle11g;sid=xe
ora2pg_1      | Isolation level: SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
ora2pg_1      | Retrieving table information...
...
ora2pg_1      | -- Dump resulting database
...
ora2pg_ora2pg_1 exited with code 0
```


4.2.c-oracle

Volume for
alf_data
persistence

Oracle 11g
container



Alfresco 4.2.c
container

Volume to
persist the
export

Compose file

4.2.c-oracle

docker-compose.yml

```
1  version: '3'
2  services:
3    alfresco:
4      build: ./alfresco
5      links:
6        - libreoffice
7      ports:
8        - 8080:8080
9        - 1521:1521
10     volumes:
11       - ./alf_data/contentstore:/usr/local/alfresco/alf_data/contentstore
12       - ./alf_data/contentstore.deleted:/usr/local/alfresco/alf_data/contentstore.deleted
13
14     db:
15       build: ./oracle11g
16       network_mode: service:alfresco
17       volumes:
18         - ./out:/u01/app/oracle/admin/XE/dpdump/exp_alfresco
19
20     libreoffice:
21       image: xcgd/libreoffice
```

4.2.c-oracle

Use

```
$ docker-compose up -d
$ docker-compose ps
```

Name	Command	State	Ports
42coracle_alfresco_1	catalina.sh run	Up	1521/tcp, 0.0.0.0:8080->8080/tcp
42coracle_db_1	/bin/sh -c /start.sh	Up	
42coracle_libreoffice_1	/opt/libreoffice/startoo.sh	Up	8997/tcp

```
$ docker-compose logs -f alfresco
```

```
...
alfresco_1 | 2017-04-22 08:59:10,231 INFO [domain.schema.SchemaBootstrap] [localhost-startStop-1] Connecting to database:
jdbc:oracle:thin:@localhost:1521:XE, UserName=ALFRESCO, Oracle JDBC driver
alfresco_1 | 2017-04-22 08:59:10,231 INFO [domain.schema.SchemaBootstrap] [localhost-startStop-1] Schema managed by
database dialect org.alfresco.repo.domain.hibernate.dialect.AlfrescoOracle9Dialect.
...
alfresco_1 | 2017-04-22 09:02:33,893 INFO [service.descriptor.DescriptorService] [localhost-startStop-1] Alfresco started
(Community). Current version: 4.2.0 (4576) schema 6.022. Originally installed version: 4.2.0 (4576) schema 6.022.
...
alfresco_1 | INFORMACIÓN: Server startup in 290217 ms
```

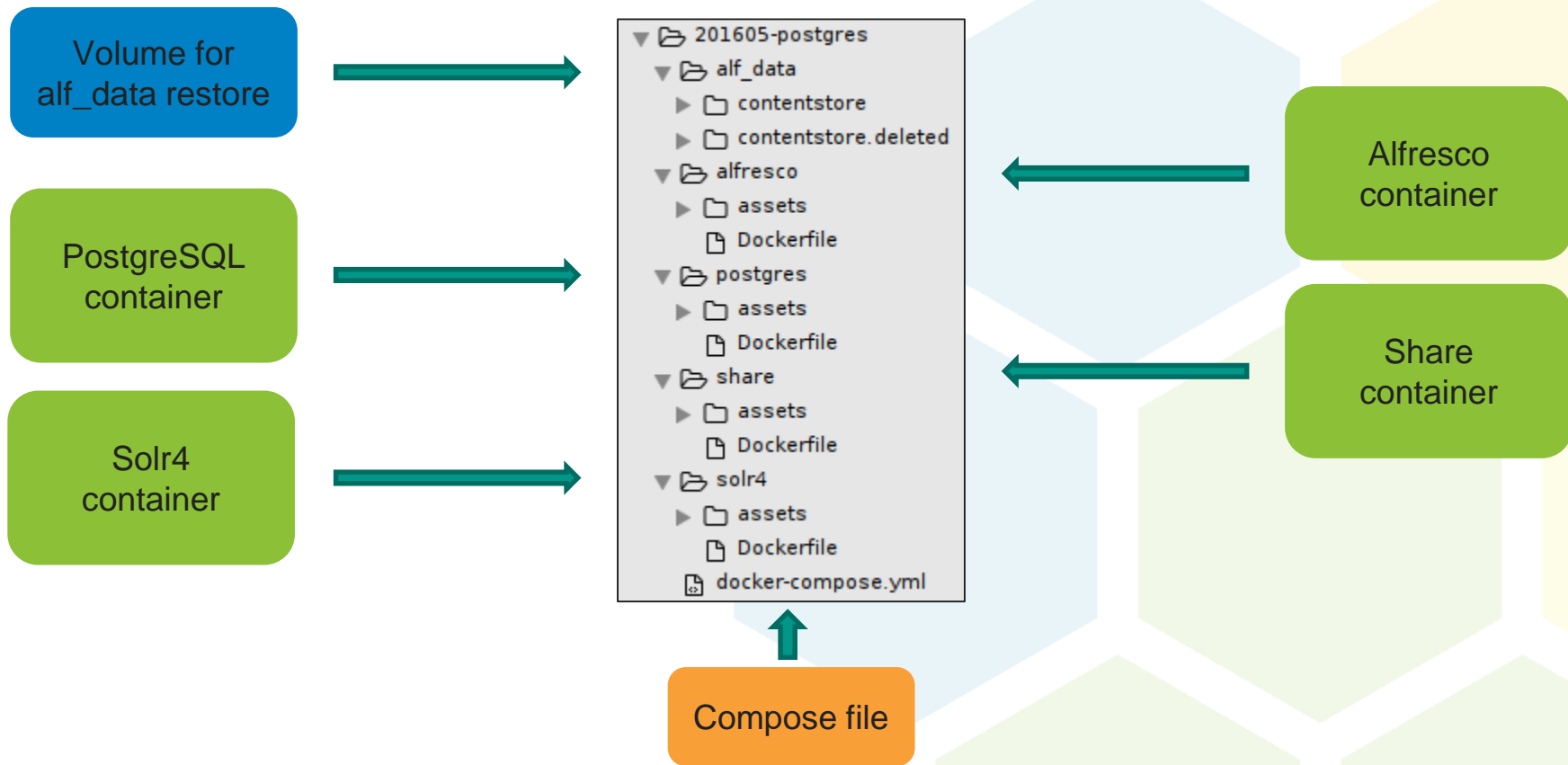
4.2.c-oracle

Database export

```
$ docker exec -it 42coracle_db_1 bash
$ root#> su - oracle
$ oracle#> sqlplus / as sysdba
$ SQL> CREATE DIRECTORY exp_alfresco AS '/u01/app/oracle/admin/XE/dpdump/exp_alfresco';
Directory created.
$ SQL> GRANT read,write ON DIRECTORY exp_alfresco TO ALFRESCO;
Grant succeeded.
$ SQL> quit
$ oracle#> expdp ALFRESCO/alfresco DIRECTORY=exp_alfresco DUMPFILE=alfresco.dmp LOGFILE=exp_alfresco.log TABLESPACES=ALFRESCO
...
Dump file set for ALFRESCO.SYS_EXPORT_TABLESPACE_01 is:
/u01/app/oracle/admin/XE/dpdump/exp_alfresco/alfresco.dmp
Job "ALFRESCO"."SYS_EXPORT_TABLESPACE_01" successfully completed at 11:34:47

$ oracle#> exit
$ root#> exit
$ docker-compose stop && docker-compose rm -vf
$ cp out/alfresco.dmp ../ora2pg/oracle11g/assets/backup
```

201605-postgres



201605-postgres

docker-compose.yml

```
1  version: '3'
2  services:
3  alfresco:
4    build: ./alfresco
5    links:
6      - db
7      - share
8      - solr4
9      - libreoffice
10   ports:
11     - 8081:8080
12   volumes:
13     - ./alf_data/contentstore:/usr/local/tomcat/alf_data/contentstore
14     - ./alf_data/contentstore.deleted:/usr/local/tomcat/alf_data/contentstore.deleted
15
16   db:
17     build: ./postgres
18     ports:
19       - 5432:5432
20     environment:
21       - POSTGRES_DB=alfresco
22       - POSTGRES_USER=alfresco
23       - POSTGRES_PASSWORD=alfresco
24
25   share:
26     build: ./share
27     ports:
28       - 8080:8080
29
30   solr4:
31     build: ./solr4
32     ports:
33       - 8082:8080
34
35   libreoffice:
36     image: xcgd/libreoffice
```

201605-postgres

Use

```
$ docker-compose up -d db
$ psql -h localhost -U alfresco -d alfresco -f ../ora2pg/out/alfresco_1492774782458_postgresql.sql
$ psql -h localhost -U alfresco -d alfresco -f ../ora2pg/update_sequences.sql
$ cp -R ../4.2.c-oracle/alf_data/contentstore* alf_data
$ docker-compose up -d -build
$ docker-compose logs -f alfresco
...
alfresco_1      | 2017-04-22 15:52:27,788 INFO [domain.schema.SchemaBootstrap] [localhost-startStop-1] Connecting to database:
jdbc:postgresql://db:5432/alfresco, UserName=alfresco, PostgreSQL Native Driver
alfresco_1      | 2017-04-22 15:52:27,788 INFO [domain.schema.SchemaBootstrap] [localhost-startStop-1] Schema managed by
database dialect org.hibernate.dialect.PostgreSQLDialect.
...
alfresco_1      | 2017-04-22 15:52:49,092 INFO [service.descriptor.DescriptorService] [localhost-startStop-1] Alfresco started
(Community). Current version: 5.1.0 (r127059-b7) schema 10.001. Originally installed version: 4.2.0 (4576) schema 6.022.
...
alfresco_1      | INFORMACIÓN: Server startup in 97089 ms
```

201605-postgres

Use

```
$ docker-compose ps
```

Name	Command	State	Ports
201605postgres_alfresco_1	catalina.sh run	Up	2121/tcp, 0.0.0.0:8081->8080/tcp
201605postgres_db_1	docker-entrypoint.sh postgres	Up	0.0.0.0:5432->5432/tcp
201605postgres_libreoffice_1	/opt/libreoffice/startoo.sh	Up	8997/tcp
201605postgres_share_1	catalina.sh run	Up	0.0.0.0:8080->8080/tcp
201605postgres_solr4_1	catalina.sh run	Up	0.0.0.0:8082->8080/tcp

Some more about docker

<https://github.com/keensoft/alfresco-docker-template>

```
$ git clone https://github.com/keensoft/alfresco-docker-template
$ cd alfresco-docker-template/templates/201702-GA
$ docker-compose up
```

```
$ docker-compose ps
```

Name	Command	State	Ports
201702ga_alfresco_1	catalina.sh jpda run	Up	8009/tcp, 0.0.0.0:9999-9999/tcp
201702ga_db_1	docker-entrypoint.sh postgres	Up	
201702ga_httpd_1	/bin/sh -c httpd -DFOREGROUND	Up	0.0.0.0:80->80/tcp
201702ga_libreoffice_1	/opt/libreoffice/startoo.sh	Up	8997/tcp
201702ga_share_1	catalina.sh run	Up	8009/tcp, 8080/tcp
201702ga_solr6_1	./run.sh run	Up	0.0.0.0:8983->8983/tcp
201702ga_swagger-editor_1	sh /usr/share/nginx/docker ...	Up	0.0.0.0:8080->8080/tcp



Migrating pieces

Export/Import content from different repositories

share-import-export

<https://github.com/wabson/share-import-export>

Python scripts for importing and exporting site-based content and user information held within Alfresco Share, and also provide some sample demo content.

- Sites
 - Site configurations
 - Site members (users only at present)
 - Site dashboards, including dashlet configuration
 - All content held within the site
 - Records Management sites (must have RM installed)
 - Web Quick Start sites (must have WQS installed)
 - Document categories and tags (specify `--export-tags` and `--import-tags`)
- Users
 - All profile information, including profile images
 - User dashboard configurations
 - User preferences
 - User groups and group memberships

alfresco-shell-tools

<https://github.com/ecm4u/alfresco-shell-tools>

Shell scripts that makes it more easy to interact with the repository and its exposed resources.

[x]alfAddAuthorityToGroup.sh

[x]alfCreateGroup.sh

[]alfCreateSite.sh

[x]alfCreateTenant.sh

[x]alfCreateUser.sh

[x]alfDeleteAuthorityFromGroup.sh

[x]alfDeleteGroup.sh

[x]alfDelete.sh

[]alfDeleteSite.sh

[x]alfDeleteUser.sh

[x]alfGetCompanyHomeNodeRef.sh

[x]alfGet.sh

[x]alfGetThumbnail.sh

[x]alfGetUserHomeFolder.sh

[x]alfGetUser.sh

[x]alfListGroupMembers.sh

[x]alfListGroupMembers.sh

[x]alfListTenants.sh

[x]alfListUsers.sh

[x]alfMetadata.sh

[x]alfNodeRef2Path.sh

[x]alfPath2NodeRef.sh

[x]alfRename.sh

[x]alfResetAvatar.sh

[x]alfSearch.sh

[x]alfSetAvatar.sh

[x]alfUpdateUser.sh

[x]alfUpload.sh

(Custom scripts)

[x]alfCreateSiteSpace.sh

[x]alfCreateSpaceRule.sh

[x]alfSetNodePermissions.sh

[x]alfGetNodePermissions.sh

ACP export

```
#!/bin/bash
```

```
JAVA_OPTS="-XX:MaxPermSize=1512m -Xms1g -Xmx3g -server"
```

```
ALF_OPTS="-Davm.rmi.service.port=0 -Davmsync.rmi.service.port=0 -  
Dattribute.rmi.service.port=0 -Dauthentication.rmi.service.port=0 -Drepo.rmi.service.port=0 -  
-Daction.rmi.service.port=0 -Dwcm-deployment-receiver.rmi.service.port=0 -  
Dmonitor.rmi.service.port=0 -Dvti.server.port=0 -Dcifs.enabled=false -Dftp.enabled=false -  
Dnfs.enabled=false -Demail.server.enabled=false -Dldap.synchronization.active=false -  
Dimap.server.enabled=false -Daudit.enabled=false -Dtransferservice.receiver.enabled=false -  
Dal fresco.rmi.services.port=0 -Dooo.enabled=false -Dooo.exe= -Djodconverter.enabled=false"
```

```
CLASSPATH=$ALF_HOME/tomcat/webapps/alfresco/WEB-  
INF/classes/alfresco/module:$ALF_HOME/tomcat/shared/classes:$ALF_HOME/tomcat/webapps/alfres  
co/WEB-INF/classes  
for lib in $ALF_HOME/tomcat/webapps/alfresco/WEB-INF/lib/*.jar $ALF_HOME/tomcat/lib/*.jar  
do  
    CLASSPATH ="${CLASSPATH}:${lib}"  
done
```

ACP export

```
user=admin  
password=admin
```

```
path="/app:company_home/cm:foo/cm:bar"
```

```
outputdir="/tmp"  
outputfile="bar.ACP"
```

```
$ java $JAVA_OPTS $ALF_OPTS -classpath $CLASSPATH org.alfresco.tools.Export  
-user $user -pwd $password -zip -verbose -dir $outputdir -overwrite -store  
workspace://SpacesStore -p $path $outputfile
```

ACP export

```
$ nohup ./export_ACP.sh > export_ACP.log &  
$ tail -f export_ACP.log
```

```
Exporting node workspace://SpacesStore/a2945c93-025f-44dc-980b-  
5ecf1e55d98c
```

```
Exporting node workspace://SpacesStore/0a48dab8-648c-419c-bf7e-  
da7913e2446e
```

```
Exporting node workspace://SpacesStore/18a45579-d142-44cc-a10f-  
4563b10f0b91
```

```
...
```

```
Time to execute 19311.357 seconds
```

```
Alfresco Repository Exporter successfully completed.
```

ACP import

```
#!/bin/bash
```

```
HOST='localhost'
```

```
PORT=21
```

```
USER='admin'
```

```
PASSWD='admin'
```

```
DIR='some folder'
```

```
FILE='bar.ACP'
```

```
ftp -n $HOST $PORT <<END_SCRIPT
```

```
quote USER $USER
```

```
quote PASS $PASSWD
```

```
tick
```

```
binary
```

```
cd $DIR
```

```
put $FILE
```

```
quit
```

```
END_SCRIPT
```

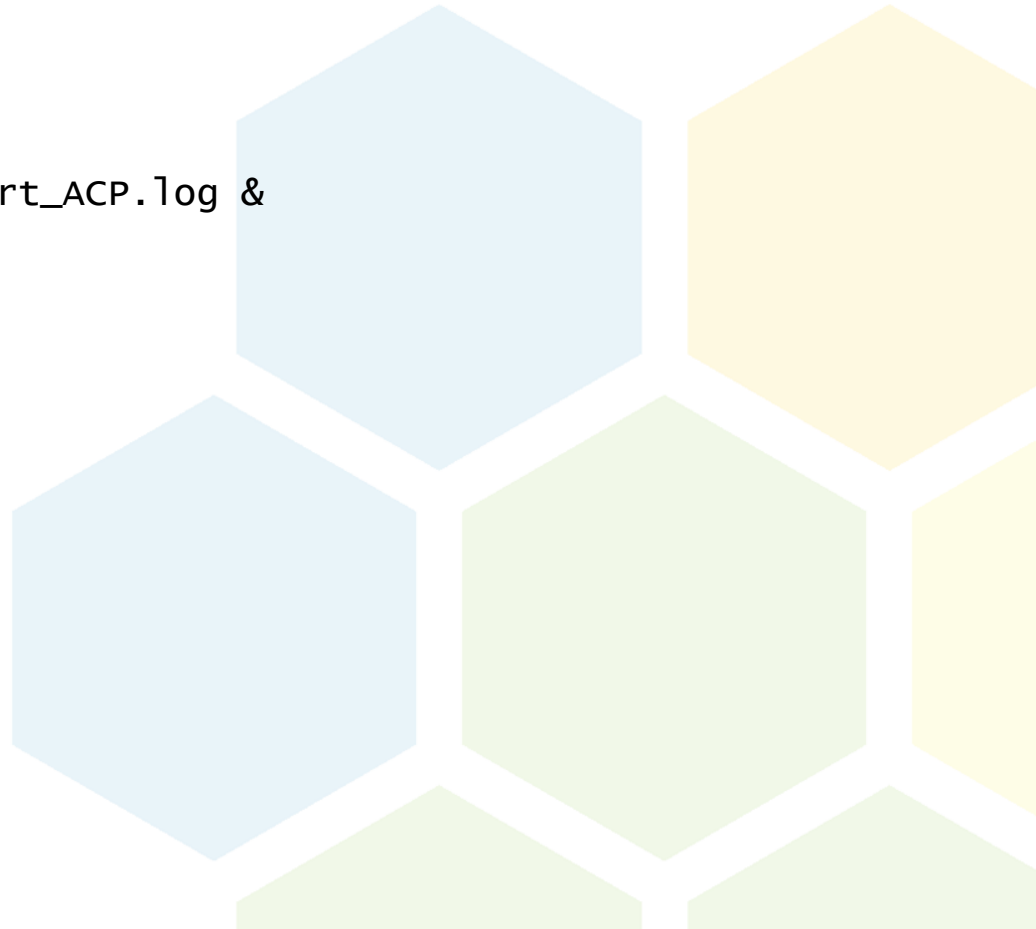
```
exit 0
```



ACP import

```
$ nohup ./import_ACP.sh > logs/import_ACP.log &  
$ tail -f logs/import_ACP.log
```

Bytes transferred: 27822193246



Node UUID preservation

```
public class CreateNewNodePreservingExistingImporterStrategy implements
NodeImporterStrategy {
    ...
    @Override
    public NodeRef importNode(final ImportNode node) {
        ...
        // Detect existing nodes
        String uuid = node.getUUID();
        boolean logChange = false;
        if (uuid != null && uuid.length() > 0)
        {
            NodeRef existingNodeRef = new
                NodeRef(StoreRef.STORE_REF_WORKSPACE_SPACESSTORE, uuid);
            if (nodeService.exists(existingNodeRef)) {
                // New UUID
                logChange = true;
            } else {
                // Use UUID from ACP
                initialProperties.put(ContentModel.PROP_NODE_UUID, node.getUUID());
            }
        }
    }
    ...
}
```

Site export Webscript

```
$ curl -u admin:admin  
http://localhost:8080/alfresco/service/api/sites/swsdp/export > swsdp.ACP
```





thanks!





@maculi33

mikel.asla@keensoft.es