**CXT 6.1 – data seeding to the CHC Cloud on AWS**

DBAs : Oscar Pagan Date : 10/21/2019

John Semencar

Jinru Wen

DBA Mgr : Raju Alluri

Db Name : CXTHEPRD

Db Server : **xcxtdbm2p**

DBA scripts location : /orahome/u01/aetna/scripts/stage/DBA/CHC\_Cloud

DBA export destination : /oraexport/u01/CXTHEPRD

Notes: The prod db will be cloned, possibly be creating an additional standby db, and used as the source

for the data seeding.

The steps to extract and encrypt the data for transfer will be performed one year at a time, for the

latest 3 years.

The preWorks steps are performed once. The Work steps are performed for each of the 3 years.

We are using gpg (GNU version of OpenPGP) to encrypt/compress. A separate doc exists on our

Sharepoint for the gpg QA test.

\*\*\* AWS CLI is installed on xcxtdbw3q:/usr/bin/aws (kind of like a man page 🡪 aws help)

[ I asked Rich Zeender to install there and Rohan is aware....he will work with us .... per Oscar ]

🡪 need AWS CLI and prod standby (m2p) server...

\*\*\* Naming the files to ship: naming convention need at least target env, date-time stamp and kind of content:

eg; Claims-extract-QA2-10182019-14-46-15.dat.gpg

\*\*\* Exclude File : Jim said it will take 2 hours to generate.

\*\*\* Consider running long processes in the background (eg; nohup myscript.ksh & )

**In a nutshell :**

1. one time preWork tasks:

- UNDO t/s should be 50 gig

- undo\_retention should be 9000 (9000 seconds = 2.5 hrs, for the largest table)

- create tables/indexes

- UPDATE Batch table (Batch table is small, under 1meg)

- load Exclude table - 10 mins (exclude list was obtained from Jim Harty)

2. do for each year:

- ensure driver table is empty

- edit/create proc (edit for a year's worth of data)

- run proc (loads the driver table; size is under 10gig)

- export (largest year is 265gig, smallest is 220gig)

- encrypt/compress export files (avg 85gig per year : 50 dmp files x 1.7gig)

- <<< ship final files >>>

Timing from testing per year, using average results from the 3 one-year tests ( x = runProc + exp + gpg )

7 - 7 hours to execute the proc

2 - 2 hours to export the data

+ 3 - 3 hours to encrpypt/compress the export files

-----

= 12 hours per year

**PreWork**

1. ensure UNDO t/s is at least 50g
2. ensure undo\_retention is set to 9000

SQL> show parameter undo\_retention

SQL> alter system set undo\_retention = 9000 scope=both;

1. create the export user and grant privs:

$> cd /orahome/u01/aetna/scripts/stage/DBA/CHC\_Cloud/preWork

SQL> cr\_dpdba.sql

SQL> cr\_dp\_dir.sql

1. create tables and indexes :

SQL> cr\_tabs\_indexes-2.sql

1. update the BATCH table :

SQL> manually\_update\_BATCH-2.sql

1. load the excluded claims ( run time is apx 10 mins ) :

\*\*\* ensure flat file “ods\_full\_exclude\_file.txt” exists in the preWork dir ( from Jim Harty)

$> load\_exclude.ksh CXTHEPRD

1. config and prepare for encryption using gpg. (the following are the results from testing in QA)

\*\*\* the gpg encryption tool is located here : /usr/bin/gpg

1. Create a key pair.

$> cd $HOME

$> gpg --gen-key

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection? **1**

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Requested keysize is 2048 bits

Please specify how long the key should be valid.

1. = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0) **0**

Key does not expire at all

Is this correct? (y/N) **y**

GnuPG needs to construct a user ID to identify your key.

Real name: aetna

Email address: aetna@aetna.com

Comment:

You selected this USER-ID: "aetna <aetna@aetna.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? **O**

2. Export the public key.

$> gpg --armor --export aetna --output aetna\_pubkey.asc

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v2.0.22 (GNU/Linux)

mQENBF1v1j0BCADLAFWAVsWxnxbxZW+8AcdmfIuuVjOq+BJ8AcsUvPQ1qCwJ2NM0

8/+Q2ASloKivl9+jcXmNSXTneTHUPNu6zqH2ZbAx1mRD0x36JJALIr1UcnJxcz+3

5RAS3y+YpACbljaW02LbrXMt2h14aJLOh71l82LDXlgil6iMbKma60EXhwnVidm8

05taa8pUJAtK3RxE7C7vnfVpT7TpZBbe9eSyiNBU9vmYSVWd8MFA922paPzhwPwf

69Co05qNPGBK9qOkhggAlLwHSqbeLAe9XmpXD4X4ACmajcu/AAFHSAT7eyS0VXRP

urfarWlNUxbiDYnJg/L0WHxuhkrPPyctXblPABEBAAG0F2FldG5hIDxhZXRuYUBh

ZXRuYS5jb20+iQE5BBMBAgAjBQJdb9Y9AhsDBwsJCAcDAgEGFQgCCQoLBBYCAwEC

HgECF4AACgkQufIDzkrk/48OFQf/YA+B1G4DVO/Akz/ImoojDqbzzo3DdnPYGxaS

qfVzdMeYgTnHhL1poR8909qQonu3ROqoDP5Q9EkHWyKXYAtymZe+8hXl/oWD7NWN

0S9MI/VFztKj0Yn+NDAgkhE844Tvldd8MoYhHJWv5CNq3I0Xex4fF30oXlztbhJq

ujhx8iVXPDnXNgcf+0btpKHHVQ8XHQ+RlUcARZ30A8kWHH+IcgDzKL3a82ZMcTZw

jEjxRhFL0We+XQPsD/k5nBFT6STo9SCt5XBLAJHbvfW18gvRqlpHfJflSJnupfNc

CGf8hoF+ceDeSJdf1Ell5duR7tpLuzU/uncxZIA167igBsZM8rkBDQRdb9Y9AQgA

vc9fRc7tzwaORHUi+2uRyXL2+XkQJHQM2eKMjjqppPgqDcjWFkNV9TFET8vXYg5l

r0JE87V7IfTWzja0YSB0J2+rjotknl9SZekszr8tU2quNuhqOMZI8/6L0dAV+jbE

SUEOHynczN5MUeAFjnxpPb+nv+rugjpdTa49DMY454WrcqWB1LzKp+z+nvICMmZc

XU22rFwDDoDAuPcpFxeW7mbq1bIf2kn7hhCJPxBqw94ejXkrTVICeYW+cj/vWlIy

xA9oWOUDrFyDACSv2f2VAjO45r2whLLm72d3vZcOCaraGU8YUhazMu3SZrfuMmQ1

FjlxGMhr+Y8s3k1SXS0pFwARAQABiQEfBBgBAgAJBQJdb9Y9AhsMAAoJELnyA85K

5P+PwocH/34B9BYG7pJM1F7/AWg7Y5PUDreb0eB/A5NVmoX+iZgr7RMjE0TCt/+M

Tj2DfGBB/OPe0x7CD/TaYHcGfnpav8p9gi9HB/p54LwyJpher407+wRfk69PMidN

mrffP883gWIE5eW70Q8e5JbqLo14dTF6+kA/hayi7pNWMNHVtRGdKS4upIJ+W2YW

WmJqICNBeWnv7y0LnLNAV5qwAEoTx4Yt+Oy1h8D56XigqTApVoA+c2Z/qEgSuLdA

PKenuIediDJGc+jFGJc1cb+vNljxxGle3fmULfi4I2WtCZYexL88Ed7tYK9bp1di

O6I2eGEfG8ahqfP7mjB1baW7eFkOfvU=28eM

-----END PGP PUBLIC KEY BLOCK-----

3. Email CHC the public key (above).

🡪 copy/paste into email and wait to receive CHC public key to pair with Aetna.

4. Import CHC public key (CHC will send this via email).

$> gpg --import chc\_pub\_key.asc

gpg: key CCAC2367: public key "CHANGEHEALTHCARE

<changehealthcare@changehealthcare.com>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

5. Sign the CHC public key (that was imported) with the Aetna private key that was created above.

$> gpg -u aetna --sign-key [changehealthcare@changehealthcare.com](mailto:changehealthcare@changehealthcare.com)

pub 2048R/CCAC2367 created: 2019-09-04 expires: never usage: SC

trust: unknown validity: unknown

sub 2048R/8952735A created: 2019-09-04 expires: never usage: E

[ unknown] (1). CHANGEHEALTHCARE <changehealthcare@changehealthcare.com>

pub 2048R/CCAC2367 created: 2019-09-04 expires: never usage: SC

trust: unknown validity: unknown

Primary key fingerprint: 7E79 7148 98FD 1402 3F20 3705 2926 F371 CCAC 2367

CHANGEHEALTHCARE [changehealthcare@changehealthcare.com](mailto:changehealthcare@changehealthcare.com)

Are you sure that you want to sign this key with your

key "aetna <aetna@aetna.com>" (64617CC4)

Really sign? (y/N) **y**

OK

**Work**

\*\*\* Perform the following steps for each year.

1. ensure the driver table is empty :

$> cd /orahome/u01/aetna/scripts/stage/DBA/CHC\_Cloud

SQL> trunc\_CCES\_CLAIM\_VERSION.sql

1. edit the proc script to extract one year’s worth of data.

script name : cr\_proc-2a.sql

\*\*\* adjust the parallel hint in the proc as needed, especially if running some steps concurrently.

eg;

SELECT **/\*+ FULL(b) PARALLEL(b,8) \*/** CV.IC\_CLAIM\_VERSION\_ID,

CV.IC\_BATCH\_ID, CV.IC\_CLAIM\_ID, CV.CLAIM\_ID

FROM CLAIM\_VERSION CV, BATCH B ...

eg;

WHERE CV.IC\_PURGE\_DATE >= TO\_TIMESTAMP('**08/01/2018**','mm/dd/yyyy')

AND CV.IC\_PURGE\_DATE < TO\_TIMESTAMP('**08/01/2019**','mm/dd/yyyy') ...

1. execute the proc ( avg run time is 7 hours) :

SQL> exec aetna\_he\_tpp.PopulatDrivingTable ;

1. export the data ( avg run time is 2 hours ) :

$> exp\_dataSeed.ksh CXTHEPRD

1. encrypt/compress the export files using CHC public key ( avg run time is 3 hours ) :

$> cp encrypt\_files.ksh /oraexport/u01/CXTHEPRD

$> cd /oraexport/u01/CXTHEPRD

$> ls -1 > filelist

\*\*\* review “filelist” and edit if needed.

\*\*\* it should only contain the export file names.

$> encrypt\_files.ksh CXTHEPRD

\*\*\* you can tail the o/p file to monitor the progress.

1. Aetna uploads the files to the AWS S3 bucket. Pls see our DBA doc:

🡪 CXT\_fileTransfer\_fromQA1\_toAWS\_DBA

**Appendix**

**Notes** 🡪 exec aetna\_he\_tpp.PopulatDrivingTable

1st run (year 1) - 7.25 hrs

* cursor : 2.75 hrs
* inserts : 4.50 hrs
* aetna\_he\_tpp.CCES\_CLAIM\_VERSION : 7.5 gig

2nd run (year 2) – 6 hrs

* cursor : 2.5 hrs
* inserts : 3.5 hrs
* aetna\_he\_tpp.CCES\_CLAIM\_VERSION : 6.2 gig

3rd run (year 3) – 7.5 hrs

* cursor : 3.0 hrs
* inserts : 4.5 hrs
* aetna\_he\_tpp.CCES\_CLAIM\_VERSION : 7 gig

**Notes** 🡪 exp\_dataSeed.ksh

- year 1 (8/1/19 - 8/1/18) : 2.5 hrs to complete - 265 gig

- year 2 (8/1/18 - 8/1/17) : 1.5 hrs to complete - 220 gig

- year 3 (8/1/17 - 8/1/16) : 1.75 hrs to complete – 255 gig

**Notes** 🡪 encrypt\_files.ksh

- year 1 (8/1/19 - 8/1/18) : hrs to complete - gig

- year 2 (8/1/18 - 8/1/17) : hrs to complete - gig

- year 3 (8/1/17 - 8/1/16) : 2.5 hrs to complete - 65 gig

The end …