

GOODUSDATA

TECHNOTE

Oracle GoldenGate 19c

New Feature

| | |
|-----------------------------------|------------|
| Author | 현하람 |
| Creation Date | 2020-06-12 |
| Last Updateed | |
| Version | 1.0 |
| Copyright(C) 2018 GoodusData Inc. | |
| All Rights Reserved | |

Document Reference

Change Record

| Date | Author | Version | Change Reference |
|------------|--------|---------|------------------|
| 2020-06-12 | 현하람 | 1.0 | 이전 문서 없음 |
| | | | |
| | | | |
| | | | |

Reviewers

| Name | Position |
|------|----------|
| | |
| | |
| | |
| | |

Distribution

| Copy No. | Name | Location |
|----------|------|----------|
| | | |
| | | |
| | | |
| | | |

Contents

| Chapter | PAGE |
|--|-----------|
| 1. Introduction | 4 |
| 2. Centralized Key Management Service | 5 |
| 2.1. KMS 란? | 5 |
| 2.1.1. OGG Encryption Key 를 저장하기 위해 KMS 를 사용하는 이유 | 5 |
| 2.1.2. Oracle Key Vault Capabilities | 5 |
| 2.2. KMS 란? OGG MA 에서 KMS 를 사용한 암호화 관리 | 5 |
| 2.2.1. Encryption Profile 이란? | 6 |
| 2.2.2. OGG 에서 OKV 를 구성하기 위한 필수 구성 요소 | 7 |
| 2.2.3. MA 에서 Encryption Profile 을 구성하는 방법 | 9 |
| 2.2.4. Oracle Key Vault 의 Key status 에 대한 client process 작동 방식 | 13 |
| 3. Target-Initiated Paths | 14 |
| 3.1. Target-Initiated Paths 개요 | 14 |
| 3.2. Target-Initiated Paths 작동 방식 | 14 |
| 4. New Heartbeat Table Command | 15 |
| 4.1. Upgrade Heartbeat Table 개요 | 15 |
| 4.2. Upgrade Heartbeat Table Practice | 15 |
| 5. Cross-Endian Support for Remote Integrated Extract | 24 |

1. Introduction

실시간 정보를 이용하기 위해 가용성과 성능에 영향을 주지 않으면서 기업 전체에 걸쳐 서로 다른 시스템의 정보를 통합할 수 있는 플랫폼이 필요합니다. Oracle GoldenGate(이하 "OGG") 19c 는 이기종 시스템과 클라우드 전반에 걸쳐 데이터베이스 트랜잭션의 실시간 캡처, 변환, 및 제공을 위한 고성능 소프트웨어 애플리케이션으로 양방향 데이터 복제를 제공합니다. OGG 19c 는 트랜잭션 무결성을 유지하면서 다양한 데이터베이스, 플랫폼 및 클라우드에 대한 짧은 지연 시간안에 고성능, 저영향으로 데이터 복제를 합니다. OGG 는 중요한 시스템이 연중무휴 작동이 가능하게하고, 관련 데이터가 기업 전체에 분산되어 의사결정을 최적화할 수 있도록 합니다. OGG 19c 에서는 Microservices Security 강화, Oracle Cross Endian Remote Capture 기능등 의 New feature 가 있습니다.

2. Centralized Key Management Service

2.1. KMS란?

Key Management Service(이하 "KMS")는 Encryption Key 관리를 중앙 집중화하는 유틸리티입니다. OGG Microservices Architecture(이하 "MA")는 KMS 를 지원하여 Encryption Key 가 OGG 에 의해 저장 또는 관리되지 않도록 보안을 지켜주면서 Encryption Key 와 Credential 관리에 대한 확장성을 제공합니다. OGG 는 캡슐화 방식을 사용하여 trail file 을 암호화합니다. 각 Trail file 에 대한 Data Encryption Key(이하 "DEK")를 생성하는데, 이를 Local Key 라고 합니다. Local Key 의 암호화된 버전이 trail file header 에 포함되고, Master Key 가 DEK 를 암호화하는 데 사용됩니다. 이 과정을 '캡슐 암호화 방식'이라고 합니다.

OGG 에서는 KMS 를 사용하여 기업 내의 암호키를 관리할 수 있습니다.

2.1.1. OGG Encryption Key 를 저장하기 위해 KMS 를 사용하는 이유

OGG 는 KMS 용도로 Oracle Key Vault(이하 "OKV")를 사용하여 Master Key 를 저장함으로써 Trail file 의 암호화를 강화합니다. Key Management 란 기업 내에서 사용하는 Encryption Key 를 관리하는 것을 말합니다. 필요에 따라 키를 생성, 교환, 저장, 사용 및 교체하는 작업을 다룹니다.

OGG 와 함께 KMS 를 사용하는 이점은 다음과 같습니다.

- Master Key 의 중앙 집중식 라이프사이클 관리가 가능합니다. 사용자가 지정하는 속성을 사용하여 Master Key 를 직접 생성하고 OKV 에 업로드할 수 있으며 KMS 내에서 라이프사이클 유지 관리 태스크를 직접 수행할 수 있습니다.
- OGG 는 Master Key 를 로컬에 저장할 필요가 없으며, Master Key 의 라이프사이클 관리에 관여하지 않습니다.
- OGG 는 여러 계층의 보안을 통해 키 관리를 제공하는 전문 KMS 기능을 활용할 수 있습니다.

2.1.2. Oracle Key Vault Capabilities

OGG 는 Oracle Key Vault 를 지원하며, 동작과 기능은 다음과 같습니다.

| KMS Name | KMS Type | Support Tags | Support Importing of Keys |
|------------------|-------------------------------|--------------|---------------------------|
| Oracle Key Vault | 버전 관리를 위한 keyname 및 사용자 지정 속성 | Yes | Yes |

2.2. KMS란? OGG MA에서 KMS를 사용한 암호화 관리

OGG MA 와 함께 KMS 를 사용할 경우의 이점과 OGG 에서 OKV 를 구성하기 위한 시스템 요구 사항, 프로세스, 파라미터에 대해 다루겠습니다.

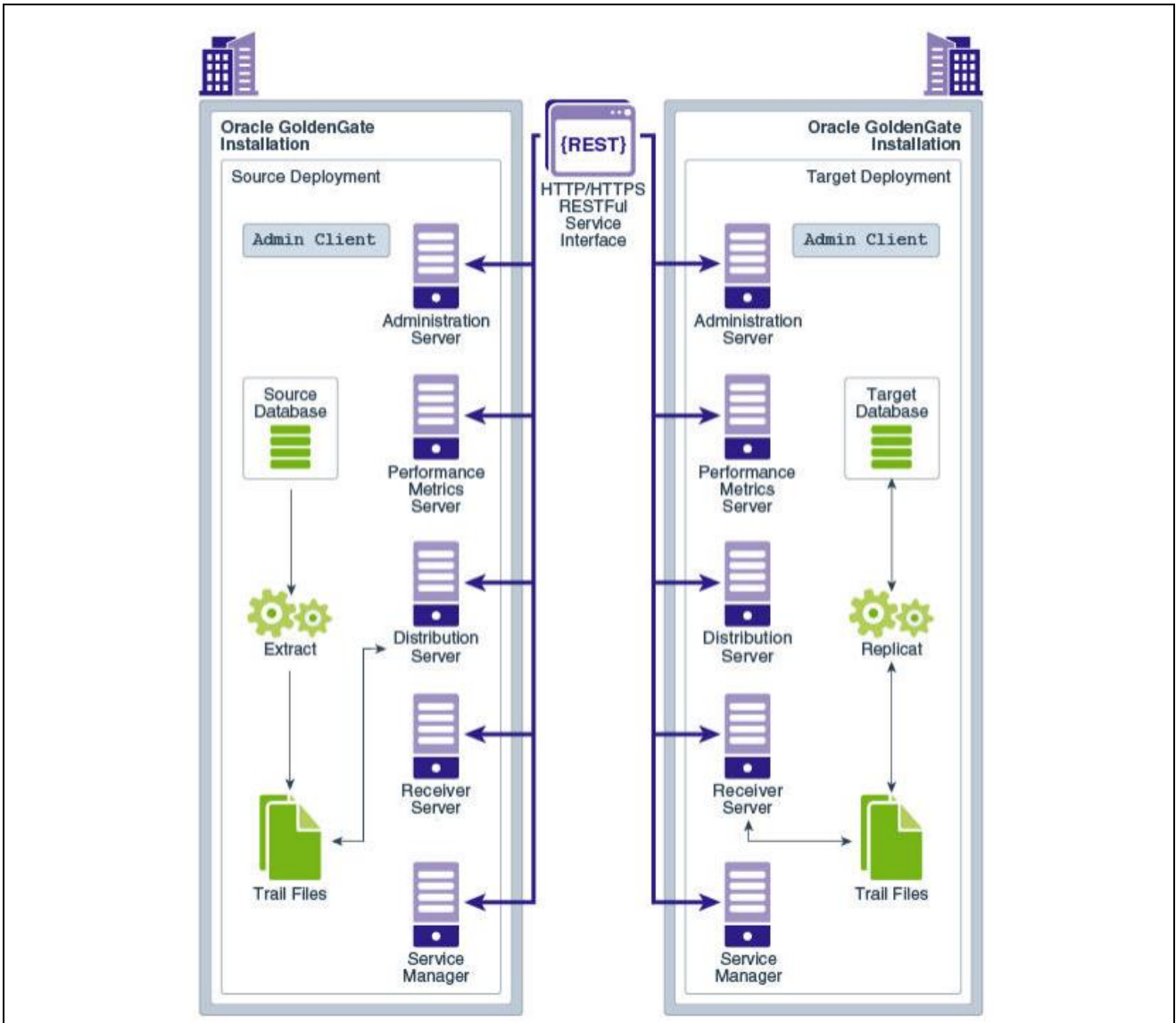
2.2.1. Encryption Profile 이란?

Encryption Profile 은 KMS 에서 Master Key 를 찾는데 사용되는 구성 정보입니다. Encryption Profile 에는 암호화와 복호화하는데 사용될 특정 Master Key 를 찾는데 필요한 모든 세부 정보와 KMS 서버에 연결하여 인증하는데 필요한 모든 정보가 포함되어 있습니다.

모든 KMS 는 인증 토큰을 사용하여 API 에 접근합니다. OGG MA 는 이 인증 토큰을 Credential 로서 저장합니다. 이 Credential 은 MA 의 Encryption Profile 을 사용하여 생성됩니다. Encryption Profile 구성은 MA 에서만 가능합니다.

Encryption Profile 은 writer client 와 reader client 가 사용합니다. Writer client 는 정보를 암호화하는 반면, Reader client 는 정보를 복호화합니다. MA 에서 각 component 에 지정된 role 에 따라 정의됩니다.

- **Extract:** Writer client
- **Replicat:** Reader client
- **Distribution Server Path:** Witer and Reader client
- **LogDump:** Reader client



client 는 암호화 MA 를 설정할 때, 사용자가 선택한 Encryption Profile 을 사용합니다. Distribution server 는 writer 와 reader 의 role 을 모두 가지고 있으며, 단 하나의 Encryption Profile 만 사용됩니다. 그러나, Distribution server 가 PASSTHRU 모드에서 작동하는 경우에는 Encryption Profile 이 필요하지 않습니다. 복호화는 컬럼 필터링을 사용할 때만 필요합니다. 사용자는 다른 Encryption Profile 을 생성할 수도 있고, 모든 client 들은 필요한 Encryption Profile 에 접근할 수 있습니다. Client 는 필요할 때마다 관련 Encryption Profile 에 접근합니다. Reader client 는 새로운 trail file 을 읽을 때마다 Encryption Profile 에 접근할 것입니다. TTL 파라미터는 TTL(Time To Live)에 도달할 때까지 Key 를 Memory 에 유지하는데 사용됩니다.

MA 에서, 각 Extract 와 Replicat 프로세스는 Encryption Profile 과 연관되어 있습니다. 다른 Encryption Profile 을 default 로 지정하지 않은 경우, default Encryption Profile 은 Local Wallet 입니다.

이미 생성된 Extract, Replicat 과 Distribution Path 는 새로 생성된 Encryption Profile 이 아닌 관련 Encryption Profile 을 사용합니다. default Encryption Profile 이 변경된 후에 생성된 process 만 새로 생성된 Encryption Profile 을 사용합니다. 따라서, Extract, Replicat 과 Distribution Path process 에 대해 다른 Encryption Profile 을 지정하는 경우 Local Wallet 은 사용되지 않습니다.

Distribution path 는 다음과 같은 상황에 Encryption Profile 을 사용합니다:

- Source trail 이 암호화되어 있지 않고 encryption object 에 알고리즘 속성을 지정

```

"target": {
  "details": {
    "encryption": {
      "algorithm": "AES256"
    }
  },
  "uri": "ogg://localhost:13101/services/v2/targets?trail=b4"
}

```

- Source trail 이 암호화되어 있고, COLUMNVALUES 타입으로 정의된 filter 가 존재

Microservices 내의 관리 서버를 통해 Encryption Profile 을 관리할 수 있습니다. Encryption Profile 은 수정할 수 없으며, 변경이 필요한 경우, 관리 서버를 통해 Encryption Profile 을 삭제하고 새로운 Encryption Profile 을 추가해야 합니다.

2.2.2. OGG 에서 OKV 를 구성하기 위한 필수 구성 요소

OGG instance 가 실행중인 시스템에 OKV 를 구성하는 요소와 방법입니다.

- 1)OGG 를 실행하는 system user 로 OKV 서버에서 okvrestservices.jar 다운로드
- 2)OGG 를 실행하는 system user 로 OKV 서버에서 okvclient.jar 다운로드 및 설치

```
[ogguser@alice ~]$ java -jar okvclient.jar -d /u01/app/oracle/OKV
```

3)Key 생성

Wallet 의 이름은 OKV administrator 가 제공합니다.

```
[ogguser@alice ~]$ java -jar okvrestservices.jar kmip
--config /u01/app/oracle/OKV/conf/okvclient.ora
--service create_key
--algorithm AES
--length 256
--mask "ENCRYPT,DECRYPT,TRANSLATE_ENCRYPT,TRANSLATE_DECRYPT,TRANSLATE_WRAP,
TRANSLATE_UNWRAP"
--wallet OKV_WALLET76876ABA-B06D-4F35-BF7C-D9306D29764B
```

혹은 다음과 같이 유저 본인의 키를 등록할 수 있습니다.

```
[ogguser@alice ~]$ java -jar okvrestservices.jar kmip
--config ./conf/okvclient.ora
--service reg_key -
ENCRYPT,DECRYPT,TRANSLATE_ENCRYPT,TRANSLATE_DECRYPT,TRANSLATE_WRAP,
TRANSLATE_UNWRAP
--wallet OGG_WALLET
--object /u01/key.txt64B3AAD0-BE77-1821-E053-0100007FD178
```

4)OKV_HOME 환경변수 설정

OKV_HOME 의 하위에는 OKV 환경을 위한 필수 library, binary 와 configuration 파일이 있습니다.

```
[ogguser@alice ~]$ setenv OKV_HOME /u01/app/oracle/OKV
```

5)Key 활성화

```
[ogguser@alice ~]$ java -jar okvrestservices.jar kmip
--config /u01/app/oracle/OKV/conf/okvclient.ora
--service activate
--uid 76876ABA-B06D-4F35-BF7C-D9306D29764B
INFO: Success
```

6)OGG 관련 키 속성(KeyName, KeyVersion)을 구성에 추가

KeyName 은 OGG 에서 생성된 KMS Encryption Profile 의 Master Key 이름과 동일해야 합니다.

Key value 는 Master Key 의 version number 와 동일해야 합니다.

```
[ogguser@alice ~] java -jar okvrestservices.jar kmip
--config /u01/app/oracle/OKV/conf/okvclient.ora
--service add_custom_attr
--uid 76876ABA-B06D-4F35-BF7C-D9306D29764B
--attribute x-OGG-KeyName
--type TEXT
--value OGG_Masterkey
```


INFO: Success

2.2.3. MA 에서 Encryption Profile 을 구성하는 방법

OGG MA 에서 사용할 수 있는 다양한 KMS 옵션에 대한 Encryption Profile 을 구성하는 단계입니다. Admin 서버 혹은 Admin 클라이언트에서 Encryption Profile 을 구성할 수 있습니다.

Extract, Replicat 과 Distribution path 에 대한 Encryption Profile 을 설정하는데 사용되는 Admin 클라이언트 명령어는 아래와 같습니다.

```
ADD ENCRYPTIONPROFILE
ALTER ENCRYPTIONPROFILE
DELETE ENCRYPTIONPROFILE
INFO ENCRYPTIONPROFILE
```

Extract, Replicat, Distribution path 를 ADD 하거나 ALTER 하는 명령어는 ENCRYPTIONPROFILE encryption-profile-name 을 포함하도록 변경되었습니다.

Master Key 를 관리하는 옵션에는 두 가지가 있습니다.

1)Local Wallet Encryption Profile

OGG MA 를 설치하거나 OGG 19c(19.1.0)으로 업그레이드 한 후, Default Encryption Profile 은 Local Wallet 에 설정됩니다.

Master Key 는 OGG Home 하위 directory 인 dirwlt directory 에 있는 Oracle Wallet 에 보관됩니다.

■ Create the Wallet

Wallet 은 CREATE WALLET command 로 생성합니다.

```
[ogguser:ALICE12201:/home/ogguser> ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Oct 18 2019 01:38:51
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (alice) 1 > create wallet

Created wallet.

Opened wallet.
```

■ Add the Master Key

Master Key 는 ADD MASTERKEY command 로 생성합니다.

Master Key 에 관한 정보를 확인하려면 info masterkey command 를 사용합니다.

```
GGSCI (alice) 2> add masterkey

2020-06-12T07:10:14Z INFO OGG-06142 Created version 1 of master key
'OGG_DEFAULT_MASTERKEY' in Oracle Wallet.

GGSCI (alice) 3> info masterkey

Masterkey Name: OGG_DEFAULT_MASTERKEY

Version          Creation Date          Status
1                2020-06-12T16:10:14.000+09:00 Current
```

■ Check the Master Key

```
[ogguser:ALICE12201:/home/ogguser/ogg> cd dirwlt
[ogguser:ALICE12201:/home/ogguser/ogg/dirwlt> ls -al
total 12
drwxr-x--- 2 ogguser dba 25 Jun 12 16:10 .
drwxr-xr-x 29 ogguser dba 4096 Jun 12 15:17 ..
-rw-r----- 1 ogguser dba 765 Jun 12 16:10 cwallet.sso
```

■ Copy the Master Key

Master Key 는 GGHOME 하위 directory 인 dirwlt 에 생성됩니다. target system 의 dirwlt directory 에 wallet file(cwallet.sso)을 copy 해주어야 합니다.

```
<Source>
[ogguser:ALICE12201:/home/ogguser/ogg/dirwlt> scp cwallet.sso
ogguser@172.40.40.244:/home/ogguser/ogg/dirwlt/
ogguser@172.40.40.244's password:
cwallet.sso                                100% 765      1.6MB/s   00:00

<Target>
[ogguser@alice dirwlt]$ ls -al
total 12
drwxr-x--- 2 ogguser dba 25 Jun 12 16:10 .
drwxr-xr-x 29 ogguser dba 4096 Jun 12 15:17 ..
-rw-r----- 1 ogguser dba 765 Jun 12 16:19 cwallet.sso
```

■ Open the Wallet

target system 에 wallet file 이 copy 가 완료되었으면, 반드시 open wallet command 를 사용하여 wallet 을 open 상태로 두어야 합니다.

```
[ogguser@alice dirwlt]$ ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.2 OGGCORE_19.1.0.0.0_PLATFORMS_190823.0013_FBO
Linux, x64, 64bit (optimized), Oracle 18c on Aug 23 2019 22:28:03
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (alice) 1 > open wallet

Opened wallet.

GGSCI (alice) 2 > info masterkey
Masterkey Name: OGG_DEFAULT_MASTERKEY

Version          Creation Date          Status
-----          -
1                2020-06-12T15:29:34.000+09:00  Current
```

■ Modify Parameter

Master Key 와 Wallet 을 이용한 OGG data encryption 을 제공하는 마지막 단계는 parameter 를 수정하는 것입니다. trail file 의 data 를 extract 나 pump 를 통하여 생성되는 trail 을 암호화하려는 경우 source 에서 암호화 할 수 있습니다.

a. Encryption in the Extract

extract parameter file 에 ENCRYPTTRAIL parameter 를 사용합니다. 해당 parameter 를 설정하면 extract 에 의해 생성되는 trail file 이 encrypt 됩니다.

SYNTAX> ENCRYPTTRAIL { AES128 | AES192 | AES256 }

```
GGSCI (alice) 1 > edit param ext01
GGSCI (alice) 2 > view param ext01

EXTRACT ext01
USERID ogguser PASSWORD ogguser
EXTTRAIL ./dirdat/ea, FORMAT RELEASE 12.2
```

```
ENCRYPTTRAIL AES256
```

```
TABLE alice.*;
```

b. Encrypting in the Pump

Pump parameter file 에 RMTHOSTOPTIONS ENCRYPT parameter 를 사용합니다.

```
SYNTAX> RMTHOSTOPTIONS ENCRYPT { AES128 | AES192 | AES256 }
```

```
GGSCI (alice) 3> edit param pmp01
```

```
GGSCI (alice) 4> view param pmp01
```

```
EXTRACT pmp01
```

```
USERID ogguser PASSWORD ogguser
```

```
RMTHOST 172.40.40.245, MGRPORT 7809
```

```
--RMTTRAIL ./dirdat/pa, FORMAT RELEASE 12.2
```

```
RMTTRAIL ./dirdat/pa
```

```
RMTHOSTOPTIONS ENCRYPT AES256
```

```
TABLE alice.*;
```

■ Master Key Encryption

Data encryption 을 위해 Master Key 와 wallet 을 setup 만 하면, source 와 target system 간의 data encryption 을 지원할 수 있습니다. 자동으로 trail file 의 암호화와 복호화가 되기 때문에, OGG trail file 의 보안을 유지하기 위해 가장 간단한 방식입니다.

2) OKV Encryption Profile

OKV 의 경우, Encryption Profile Credential 를 위해 다음 내용이 필요합니다.

- Name: OKV Encryption Profile 의 이름 명시
- Type: KMS type 을 OKV 로 지정
- Home Path: OKV 가 설치된 directory location 명시
- Key Name Attribute: Custom attribute 에서 사용하고 있는 Encryption Key 이름 명시. 이 값은 OGG 의 KMS parameter 값에 있는 key name 과 동일해야 하며, replicat 이 시작되면 변경할 수 없습니다. Key Version Attribute: Custom attribute 에서 사용하고 있는 Encryption Key 의 버전을 명시. 반드시 숫자로 명시해야 합니다.
- MasterKey Name: Master Key 의 이름 명시
- MasterKey Version: OKV 의 버전 명시. Default value 는 LATEST 혹은 18.1 과 같이 version number 를 명시 할 수 있습니다.

- Time to live(TTL): KMS 에서 Extract 가 발견한 key 에 대한 유효기간 명시. 다음 trail file 이 암호화 되었을 때, Extract 는 TTL 이 expired 되지 않았는지 체크합니다. expired 되었다면, Extract 는 Master Key 의 최신 버전을 찾습니다. Default value 는 24 시간 입니다.

* Note: Key Name 과 Key version 이 중복되는 값을 가진 Key 를 업로드하지 말아야 합니다. OGG 가 startup, restart, rollover 시에, 가장 높은 버전의 Key Version 값을 찾습니다.

2.2.4. Oracle Key Vault 의 Key status 에 대한 client process 작동 방식

Encryption Key status 에 따른 reader 혹은 writer client process 의 작동 방식입니다.

- **Active:** Trail writer 는 암호화를 위해 활성화 되어있는 가장 높은 버전의 Key 를 선택합니다. Trail reader 는 trail 을 복호화하기 위해 해당 Key(Key, Version number)를 사용합니다.
- **Preactive:** Trail writer 는 Preactive 상태인 key 와 version number 를 무시합니다.
- **Deactivated:** Trail writer 는 Deactivated 상태인 key 와 version number 를 무시합니다. Trail file reader 는 해당 Key 가 deactivated 거나 Compromised 상태인 경우, trail 을 복호화하기 위해 이 Key 를 찾아서 사용하게 됩니다.
- **Compromised:** Trail writer 는 Compromised 상태인 key 와 version number 를 무시합니다. Trail file reader 는 해당 Key 가 deactivated 거나 Compromised 상태인 경우, trail 을 복호화하기 위해 이 Key 를 찾아서 사용하게 됩니다.
- **Destroyed:** Trail writer 는 Destroyed 상태인 key 와 version number 를 무시합니다. Trail file reader 는 Destroyed 상태에서 복호화하기 위해 key 와 version number 가 필요하다면, 에러를 발생시키고 Abended 됩니다.
- **Destroyed-Compromised:** Trail writer 는 Destroyed 상태인 key 와 version number 를 무시합니다. Trail file reader 는 Destroyed 상태에서 복호화하기 위해 key 와 version number 가 필요하다면, 에러를 발생시키고 Abended 됩니다.

3. Target-Initiated Paths

3.1. Target-Initiated Paths 개요

Microservice 에 대한 Target-Initiated paths 설정을 하면 Receiver Server 는 Target 에 대한 Distribution service 경로를 정하고 trail file 을 끌어올 수 있습니다. 이 기능을 통해 네트워크 보안 정책으로 인해 Source server 에서 target server 로의 네트워크 연결이 불가능한 cloud to on-premise 와 같은 환경에서 target-initiated path 를 생성할 수 있습니다.

Distribution server 가 Receiver server 에는 연결할 수 없지만, Receiver server 가 운영중인 Distribution server 에 연결이 가능할 경우, Receiver server 는 OGG 를 사용하여 방화벽을 통해 Distribution server 로 secure/non-secure target initiated path 를 설정할 수 있으며, 요청된 trail file 을 끌어올 수 있습니다.

3.2. Target-Initiated Paths 작동 방식

OGG 는 Source 와 target 시스템 간의 alias 를 통한 연결을 설정하기 위해 passive alias 구성을 사용하고 있습니다. MA 에서는 이 기능이 향상되었지만, Receiver server, Distribution server, Admin Client 명령어로 관리하는 target-initiated distribution path 에서는 사용할 수 없습니다.

path 는 target 의 Receiver server 에서 생성되며, TARGET_INITIATED 속성을 가지고 있습니다. 해당 path 는 read-only 이며, Distribution server 에서도 접근할 수 있습니다. path 에 대한 정보는 Target 시스템에 저장됩니다.

connection 이 끊긴 경우, target host 의 receiver server 는 connection 을 다시 연결하기 위해 path definition 이 필요합니다. 이 정보는 path 가 running 상태일 때, Distribution server 와 공유됩니다.

Path 가 stop 되거나 disconnected 되었을 때, Distribution Server 는 path definition 을 포함한 path 에 대한 모든 정보를 삭제합니다. 하지만, checkpoint 는 old trail file 이 purge 될 수 있는지 여부를 결정하는데 사용되기 때문에 checkpoint file 은 보존됩니다. path 를 의도적으로 삭제하지 않는 한 old trail 파일은 purge 하지 않는 것이 좋습니다.

4. New Heartbeat Table Command

4.1. Upgrade Heartbeat Table 개요

UPGRADE HEARTBEATTABLE 커맨드를 사용하여 19c 이전 버전의 OGG 에서 생성된 heartbeat table 과 lag view 를 alter 하여 extra column 을 추가할 수 있습니다. 해당 extra column 은 extract restart position 을 추적하는데 사용됩니다. UPGRADE HEARTBEATTABLE 커맨드는 이미 extra column 이 추가되어있는 table 과 view 의 경우 heartbeat table 의 설정에서 실패하게 됩니다.

해당 명령어는 DBLOGIN 이 필요하며 CDB 의 경우 PDB 로그인 이 필요합니다.

추가되는 Column 은 아래와 같습니다.

- INCOMING_REPLICAT_LW_CSN
- INCOMING_EXTRACT_HEARTBEAT_CSN
- INCOMING_EXTRACT_RESTART_CSN
- INCOMING_EXTRACT_RESTART_TS

4.2. Upgrade Heartbeat Table Practice

Heartbeat Table 을 Upgrade 하여 table recovery 시, extract 와 replicat 의 restart position 을 추적하여 빠르게 recovery 하도록 합니다.

■ Oracle Version 확인

```
[oracle:ALICE11204:/home/oracle> sqlplus "/as sysdba"

SQL*Plus: Release 12.2.0.1.0 Production on Fri Jun 12 15:13:49 2020

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> select version, status from v$instance;

VERSION          STATUS
-----
12.2.0.1.0       OPEN
```

■ Oracle GoldenGate Version Check

```
[ogguser:ALICE12201:/home/ogguser> ggsci -v
```

```
Oracle GoldenGate Command Interpreter for Oracle
```

```
Version 12.2.0.1.1 OGGCORE_12.2.0.1.0_PLATFORMS_151211.1401_FBO
```

```
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
```

```
Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.
```

■ Heartbeat table 추가

```
GGSCI (alice) 1> dblogin userid oraogg password oraogg
```

```
Successfully logged into database.
```

```
GGSCI (alice as oraogg@ALICE12201)2> add heartbeatable
```

```
2020-06-12 14:52:30 INFO OGG-14001 Successfully created heartbeat seed table  
["GG_HEARTBEAT_SEED"].
```

```
2020-06-12 14:52:30 INFO OGG-14032 Successfully added supplemental logging for  
heartbeat seed table ["GG_HEARTBEAT_SEED"].
```

```
2020-06-12 14:52:30 INFO OGG-14000 Successfully created heartbeat table  
["GG_HEARTBEAT"].
```

```
2020-06-12 14:52:30 INFO OGG-14033 Successfully added supplemental logging for  
heartbeat table ["GG_HEARTBEAT"].
```

```
2020-06-12 14:52:30 INFO OGG-14016 Successfully created heartbeat history table  
["GG_HEARTBEAT_HISTORY"].
```

```
2020-06-12 14:52:30 INFO OGG-14023 Successfully created heartbeat lag view ["GG_LAG"].
```

```
2020-06-12 14:52:30 INFO OGG-14024 Successfully created heartbeat lag history view  
["GG_LAG_HISTORY"].
```

```
2020-06-12 14:52:30 INFO OGG-14003 Successfully populated heartbeat seed table with  
[ALICE122].
```

```
2020-06-12 14:52:30 INFO OGG-14004 Successfully created procedure
```


["GG_UPDATE_HB_TAB"] to update the heartbeat tables.

2020-06-12 14:52:30 INFO OGG-14017 Successfully created procedure
["GG_PURGE_HB_TAB"] to purge the heartbeat history table.

2020-06-12 14:52:31 INFO OGG-14005 Successfully created scheduler job
["GG_UPDATE_HEARTBEATS"] to update the heartbeat tables.

2020-06-12 14:52:31 INFO OGG-14018 Successfully created scheduler job
["GG_PURGE_HEARTBEATS"] to purge the heartbeat history table.

GGSCI (alice as oraogg@ALICE12201) 3> info heartbeatable

HEARTBEAT table oraogg.gg_heartbeat exists.

HEARTBEAT table oraogg.gg_heartbeat_seed exists.

HEARTBEAT table oraogg.gg_heartbeat_history exists.

Frequency interval: 60 seconds.

Purge frequency interval: 1 days.

Retention time: 30 days.

■ OGG Heartbeat table 과 Description 확인

[oracle:ALICE12201:/home/oracle> sqlplus "/as sysdba"

SQL*Plus: Release 12.2.0.1.0 Production on Fri Jun 12 14:54:02 2020

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> conn oraogg/oraogg

Connected.

```
SQL> select table_name from user_tables;
```

```
TABLE_NAME
```

```
-----  
CHKPT_LOX  
CHKPT  
GG_HEARTBEAT_SEED  
GG_HEARTBEAT  
GG_HEARTBEAT_HISTORY
```

```
SQL> desc GG_HEARTBEAT
```

| Name | Null? | Type |
|-----------------------|-------|----------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_TIMESTAMP | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |
| INCOMING_EXTRACT | | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | | VARCHAR2(4000) |
| INCOMING_REPLICAT | | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT | | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | | VARCHAR2(4000) |
| OUTGOING_REPLICAT | | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | | TIMESTAMP(6) |

```
SQL> desc GG_HEARTBEAT_SEED
```

| Name | Null? | Type |
|---------------------|-------|---------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_TIMESTAMP | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |

| | |
|-----------------------|----------------|
| INCOMING_EXTRACT | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | VARCHAR2(4000) |
| INCOMING_REPLICAT | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | TIMESTAMP(6) |
| INCOMING_REPLICAT_TS | TIMESTAMP(6) |
| OUTGOING_EXTRACT | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | VARCHAR2(4000) |
| OUTGOING_REPLICAT | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | TIMESTAMP(6) |

SQL> desc GG_HEARTBEAT_HISTORY

| Name | Null? | Type |
|-----------------------|-------|----------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_RECEIVED_TS | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |
| INCOMING_EXTRACT | | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | | VARCHAR2(4000) |
| INCOMING_REPLICAT | | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT | | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | | VARCHAR2(4000) |
| OUTGOING_REPLICAT | | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | | TIMESTAMP(6) |

■ OGG version 12.2.0.2 -> 19.1.0.0.4 로 upgrade 후, vesion Check

```
[ogguser:ALICE12201:/home/ogguser/ogg> ggsci -v
```

```
Oracle GoldenGate Command Interpreter for Oracle  
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO  
Linux, x64, 64bit (optimized), Oracle 12c on Oct 18 2019 01:38:51  
  
Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.
```

■ OGG Heartbeat table Check

```
[ogguser:ALICE12201:/home/ogguser> ggsci
```

```
Oracle GoldenGate Command Interpreter for Oracle  
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO  
Linux, x64, 64bit (optimized), Oracle 12c on Oct 18 2019 01:38:51  
Operating system character set identified as UTF-8.
```

```
Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.
```

```
GGSCI (alice) 1> dblogin userid oraogg password oraogg
```

```
Successfully logged into database.
```

```
GGSCI (alice as oraogg@ALICE12201) 2> info heartbeatable
```

```
HEARTBEAT table oraogg.gg_heartbeat exists.
```

```
HEARTBEAT table oraogg.gg_heartbeat_seed exists.
```

```
HEARTBEAT table oraogg.gg_heartbeat_history exists.
```

```
HEARTBEAT table oraogg.gg_heartbeat supplemental logging ENABLED.
```

```
HEARTBEAT table oraogg.gg_heartbeat_seed supplemental logging ENABLED.
```

```
HEARTBEAT table oraogg.gg_heartbeat_history partitioning DISABLED.
```

```
Frequency interval: 60 seconds.
```

```
Purge frequency interval: 1 days.
```

```
Retention time: 30 days.
```

■ OGG UPGRADE Heartbeat table

```
GGSCI (alice as oraogg@ALICE12201) 3> upgrade heartbeattable
```

```
2020-06-12 15:17:53 INFO OGG-14089 Successfully tracking extract restart position with
heartbeat table ""oraogg"."GG_HEARTBEAT_SEED"".
```

```
2020-06-12 15:17:53 INFO OGG-14089 Successfully tracking extract restart position with
heartbeat table ""oraogg"."GG_HEARTBEAT"".
```

```
2020-06-12 15:17:53 INFO OGG-14089 Successfully tracking extract restart position with
heartbeat table ""oraogg"."GG_HEARTBEAT_HISTORY"".
```

```
2020-06-12 15:17:53 INFO OGG-14023 Successfully created heartbeat lag view
""oraogg"."GG_LAG"".
```

```
2020-06-12 15:17:53 INFO OGG-14024 Successfully created heartbeat lag history view
""oraogg"."GG_LAG_HISTORY"".
```

■ OGG OGG UPGRADE 된 Heartbeat table Column Check

```
[oracle:ALICE12201:/home/oracle> sqlplus "/as sysdba"
```

```
SQL*Plus: Release 12.2.0.1.0 Production on Fri Jun 12 15:18:32 2020
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
```

```
SQL> conn oraogg/oraogg
```

```
Connected.
```

```
SQL> select table_name from user_tables;
```

```
TABLE_NAME
```

```
-----
```

```
CHKPT_LOX
```

```
CHKPT
```

```
GG_HEARTBEAT_SEED
```

```
GG_HEARTBEAT
```

```
GG_HEARTBEAT_HISTORY
```

```
SQL> desc GG_HEARTBEAT
```

| Name | Null? | Type |
|--------------------------------|-------|----------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_TIMESTAMP | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |
| INCOMING_EXTRACT | | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | | VARCHAR2(4000) |
| INCOMING_REPLICAT | | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT | | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | | VARCHAR2(4000) |
| OUTGOING_REPLICAT | | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_LW_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_HEARTBEAT_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_TS | | TIMESTAMP(6) |

```
SQL> desc GG_HEARTBEAT_SEED
```

| Name | Null? | Type |
|-----------------------|-------|----------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_TIMESTAMP | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |
| INCOMING_EXTRACT | | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | | VARCHAR2(4000) |
| INCOMING_REPLICAT | | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | | TIMESTAMP(6) |

| | |
|--------------------------------|----------------|
| INCOMING_REPLICAT_TS | TIMESTAMP(6) |
| OUTGOING_EXTRACT | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | VARCHAR2(4000) |
| OUTGOING_REPLICAT | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | TIMESTAMP(6) |
| INCOMING_REPLICAT_LW_CSN | VARCHAR2(128) |
| INCOMING_EXTRACT_HEARTBEAT_CSN | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_CSN | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_TS | TIMESTAMP(6) |

SQL> desc GG_HEARTBEAT_HISTORY

| Name | Null? | Type |
|--------------------------------|-------|----------------|
| LOCAL_DATABASE | | VARCHAR2(512) |
| HEARTBEAT_RECEIVED_TS | | TIMESTAMP(6) |
| REMOTE_DATABASE | | VARCHAR2(512) |
| INCOMING_EXTRACT | | VARCHAR2(128) |
| INCOMING_ROUTING_PATH | | VARCHAR2(4000) |
| INCOMING_REPLICAT | | VARCHAR2(128) |
| INCOMING_HEARTBEAT_TS | | TIMESTAMP(6) |
| INCOMING_EXTRACT_TS | | TIMESTAMP(6) |
| INCOMING_ROUTING_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT | | VARCHAR2(128) |
| OUTGOING_ROUTING_PATH | | VARCHAR2(4000) |
| OUTGOING_REPLICAT | | VARCHAR2(128) |
| OUTGOING_HEARTBEAT_TS | | TIMESTAMP(6) |
| OUTGOING_EXTRACT_TS | | TIMESTAMP(6) |
| OUTGOING_ROUTING_TS | | TIMESTAMP(6) |
| OUTGOING_REPLICAT_TS | | TIMESTAMP(6) |
| INCOMING_REPLICAT_LW_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_HEARTBEAT_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_CSN | | VARCHAR2(128) |
| INCOMING_EXTRACT_RESTART_TS | | TIMESTAMP(6) |

5. Cross-Endian Support for Remote Integrated Extract

Intergrated extarct 가 실행중인 서버가 Oracle database 가 실행중인 서버와 다를 때 Cross-endian support 가 자동으로 실행됩니다. Remote system 에서 Intergrated Extarct 가 실행중인 경우, OGG 는 자동으로 Cross-Endian 운영을 가능하게 합니다. 즉, Intergrated extract 가 실행 중인 Endian 값이 Oracle Database 가 실행 중인 Endian 값과 다른 경우 Cross-Endian support 가 자동으로 실행됨을 의미합니다. Cross-Endian remote Integrated Extract 가 support 가능함에 따라 on-premise database 에서의 remote capture 를 cloud 로 migration 하는 것이 가능해졌습니다.