Oracle 12c Multitenant and Encryption in Real Life

Christian Pfundtner
DB Masters GmbH
Stammersdorfer Str. 463
2201 Gerasdorf

Schlüsselworte

Oracle 12c, Multitenant, CDB, PDB, Database Vault DBV, Advanced Security, Transparent Data Encryption, TDE, Data Guard, Standby, Active Data Guard, Kundenprojekt

Einleitung

In diesem Vortrag werden die Erfahrungen und Erlebnisse mit der Kombination von Oracle Features wie Multitenant, Transparent Data Encryption, Database Vault und Active Data Guard im Rahmen eines Kundenprojektes beschrieben. Dabei zeigt sich, dass jedes Feature für sich alleine sehr gut funktioniert, nur in der Kombination ergeben sich Tücken und Fallen, die man zuerst erkennen und verstehen muss, um sie entweder zu beheben oder umgehen zu können.

Teil 1 – die genutzten Oracle Funktionalitäten im Überblick

Im ersten Teil stellen wir die im Rahmen des Projekts genutzten Oracle Funktionaltäten kurz vor und gehen auch darauf ein, warum genau diese in dem Kundenprojekt genutzt werden.

Oracle 12c Multitenant

Mit dieser Option kann man voneinander unabhängige Datenbanken unter einer Instanz zusammenfassen. Viele (aber nicht alle) der administrativen Aufgaben werden dadurch deutlich einfacher – beispielweise:

- Monitoring und Tuning
- Backup und Recovery
- Data Guard Nutzung
- Upgrades und Patches

Da im Kundenprojekt sehr viele „Kopien“ (Klone) von Datenbanken für Test-, Development, Schulung und Applikationsversionsvergleich benötigt werden (teilweise bis zu 70 Kopien der produktiven Datenbank), die aber immer nur selektive und zeitweise genutzt werden, kann man mit Multitenant (eine Instanz für alle Datenbanken) optimal die vorhandenen Ressourcen wie CPU und Memory für die gerade aktuell genutzten „Kopien“ (=PDBs) nutzen. Genau dies war der Grund für die Wahl dieser Option, dass damit das Erzeugen von Kopien sehr einfach geht, war nur der Zuckerguss.

Oracle 12c Database Vault

Database Vault gibt es schon seit 10g und dient dazu, den Zugriff auf die Daten innerhalb der Datenbank genau zu regeln. Damit kann man unter anderem sicherstellen, dass DBAs keine Daten sehen können, ohne in ihrer Arbeit behindert zu werden. Da der Kunde eine Art „Hosting“ für sensitive Daten für seine Kunden anbietet, ist dies eine essentielle Funktionalität. Neu mit Oracle 12c ist, dass man Database Vault nicht mehr auf Betriebssystemebene abschalten kann, sondern nur noch innerhalb der Datenbank.

Oracle 12 Advanced Security: Network und Datenverschlüsselung (TDE)

Damit die Daten sowohl im Netzwerk (LAN) als auch auf dem Datenträger (und auch im Backup) vor dem Zugriff geschützt sind, werden alle Daten mittels TDE – Transparent Data Encryption – verschlüsselt. Die dazugehörigen Schlüssel liegen in sogenannten Oracle Wallets. Gemeinsam mit Database Vault kann man somit sicherstellen, dass nur bei Zugriffen über die Applikation auf die Daten in der Datenbank zugegriffen werden kann. Ein Administrator (oder ein Hacker) hat keine Chance, die Daten im Klartext zu erhalten.

Oracle 12c (Active) Data Guard

Mittels Data Guard kann man eine Datenbank auf einen weiteren Standort „spiegeln“, um beispielsweise den Ausfall eines Rechenzentrums (desastertolerant) ohne Datenverlust überleben zu können. Da einige Verarbeitungen beim Kunden an zeitlich enge Termine gebunden sind, musste sichergestellt werden, dass der Ausfall eines Rechenzentrums zu keinen größeren Verzögerungen führt. Der Kunde hat – für eine spätere Ausbaustufe – die Active Data Guard Lizenz (lesender Zugriff auf die Kopie der Datenbank) gleich mitbestellt, allerdings ist der Applikationshersteller noch nicht soweit, diese schon zu optimal zu nutzen.

Teil 2 – das Kundenprojekt – von den Anforderungen und Erlebnissen in der Praxis

Wie schon bei den verschiedenen Optionen beschrieben, ist es für den Kunden in dem Projekt essentiell, dass niemand – wirklich niemand, auch nicht die eigenen Mitarbeiter – Zugriff auf die Kundendaten bekommen kann. Dies wird regelmäßig durch interne und auch externe Audits verifiziert.

Da die Applikation laufend an neue Anforderungen und Regeln angepasst wird, gibt es aktuell alle 1-2 Wochen neue Software Deployments – wie jeder weiß, werden die nicht immer fehlerfrei sein, daher ist eine Anforderung / Bedingungen, dass vor einem Deployment der aktuellen Stand „eingefroren“ wird, was dank der Multitenant Funktionalität mittels Kopieren (Klonen) der PDB erfolgt. Parallel dazu werden sofort Kopien für weitere Entwicklung, für Schulung und für Tests durchgeführt, was die Anzahl der PDBs innerhalb einer CDB bis auf über 70 ansteigen lässt. Erst nach einem sogenannten Abrechnungszyklus dürfen alte PDBs gelöscht werden. Dass man die ganze Multitenant Datenbank mittels EINER Standby Datenbank mit Data Guard schützt, kommt den Kunden extrem entgegen, so müssen während eines Deploymentzyklus nicht auch gleich noch sehr viele Data Guards erzeugt und aufgebaut werden.

Aktuell hat der Kunde ca. 20 CDBs (Multitenant Datenbanken) mit in Summe über 500 PDBs.

<u>Real Life Begins – Starten einer Datenbank Instanz (Multitenant + Oracle Wallet für TDE)</u>

Als erster kleiner Stolperstein auf dem Weg stellt sich die Verschlüsselung heraus, für die die Schlüssel in sogenannten Oracle Wallets abgelegt werden müssen. Solang so ein Wallet nicht geöffnet wird, kann man auf die verschlüsselten Daten nicht zugreifen. Da sowohl die CDB als auch jede PDB über eigene Keys verfügen, muss man sicherstellen, dass für alle „offenen“ PDBs auch die Wallet Keys geladen werden, bevor man auf die Daten zugreifen kann.

Neben der Möglichkeit die Oracle Wallets „manuell“ (mit SQL Befehl) zu öffnen, gibt es seitens Oracle auch eine sogenannten „Auto Login“ Funktionalität, um das Wallet zu öffnen - allerdings nur Read Only. Beim manuellen Weg ist es wichtig, dass der DBA das dafür nötige Passwort NICHT kennt. Sowohl das Wallet als auch ein etwa vorhandenes Auto Login File muss manuell auf die

Standby (Data Guard) Seite kopiert werden, da sonst kein Recovery erfolgen kann. Bei jeder Änderung (zb: neue PDB) muss das Wallet auch wieder manuell kopiert werden.

Und da sind wir schon beim eigentlichen Problem angelangt: Wenn eine PDB kopiert/geklont wird, muss ein neuer Key für diese PDB erzeugt und muss im Wallet abgespeichert werden – das ist aber nicht möglich, wenn das Wallet mittels Auto Login nur Read Only geöffnet ist. Die Lösung dazu (der Workaround) ist, dass man vorsieht, dass bei einem Restart der Instanz (oder einem Öffnen einer PDB) dies nur gemeinsam von einem DBA und einem Security Administrator gemacht werden kann.

Das ist zwar nicht immer einfach, kann aber organisiert werden. Nur gibt es hierzu gleich eine schmerzhafe Ausnahme: Wenn man Point in Time Recovery für eine PDB (oder einen Tablespace oder einen Table) durchführen möchte, wird dabei automatisiert eine Kopie einer PDB erzeugt, die lediglich lesenden Zugriff auf den Key des Originals braucht (um die Daten bereitstellen zu können). Diese Funktionalität benötigt somit zwingend ein Auto Login Wallet !!! Die Lösung ist, dass man in den Workflow für Point in Time Recovery vorsieht, dass ein Auto Login Wallet angelegt und nach dem Recovery sofort wieder gelöscht wird.

Real Life Begins – Multitenant und Data Guard

Immer wenn man in einer Multitenant Datenbank eine weitere PDB anlegt (zB: durch kopieren), werden die Datenbank Files durch Data Guard auf der Standby Seite ebenfalls erzeugt – genau das, was der Kunde benötigt (man muss nur noch dafür sorgen, dass das Wallet ebenfalls transferiert wird).

Leider gibt es hier einen Bug, der in manchen Situationen verhindert, dass die Datenbank im Remotesystem alle Datenbank Files anlegt (der Bug ist aktuell noch nicht behoben). Manchmal „vergisst“ sie einfach einzelne Files anzulegen. Zufälligerweise sind wir auf den Workaround gestoßen: Wenn man die Standby Datenbank öffnet (benötigt Active Data Guard), dann erkennt Data Guard die Situation und sorgt dafür, dass die fehlenden Files ebenfalls transferiert werden (die Info muss in den verfügbaren Redo Logs (Archive Logs) noch vorliegen. Zum Glück hatte der Kunde Active Data Guard schon lizensiert, wodurch nur eine kleine Erweiterung des Workflows beim Kopieren/Clonen einer PDB nötig wurde (neben dem Transferieren des Wallets muss sicher gestellt werden, dass alle PDBs read only geöffnent werden).

Real Life Begins – Multitenant, Data Guard und Security

Wie schon erwähnt muss man das Oracle Wallet auf die Standby Seite manuell kopieren und immer dann erneut kopieren, wenn eine PDB neu dazu kommt oder gelöscht wird. Das bedeutet, dass mehrere Schritte erforderlich sind, damit es zu keinen Überraschungen kommt:

- Stoppen von Data Guard Apply
- Wallet auf der Standby Datenbank schließen
- Neue PDB (oder Kopie) erzeugen
- Wallet von der Primary auf die Standby kopieren
- Wallet auf der Standby Seite öffnen (für alle PDBs)
- Data Guard Apply neu starten

Real Life Begins – Multitenant und Database Vault

Die Nutzung und Konfiguration von Database Vault in einem Multitenant Environment ist leider relativ aufwändig. Es ist aber auf Grund der Architektur klar, dass es so sein muss – die Gründe dafür würden den Rahmen des Vortrages sprengen.

Wenn man DBV aktivieren oder deaktivieren muss, so muss:

- Zuerst auf der CDB konfiguriert und eingeschaltet werden
- Dann auf jeder PDB, die DBV nutzt (in unserem Fall: auf allen) dies jeweils konfiguriert und eingeschaltet werden
- Sobald das fertig ist, muss die Instanz neu gestartet werden (siehe das Thema rund um das Öffnen des Wallets).
- Beim Deaktivieren muss zuerst DBV auf allen PDBs und zuletzt auf der CDB deaktiviert und dann die Instanz neu gestartet werden.

Das Aktivieren und Deaktivieren sowie das Restarten der Instanz bedeutet aber, dass hier wieder sowohl der DBA als auch der Security Administrator beteiligt sein müssen. Zusätzlich gibt es bis zu 70 PDBs, an die man sich jeweils anmelden und wo man Befehle absetzen muss. Bei diesem Kundenprojekt kommt erschwerend hinzu, dass nicht aktiv benötigte PDBs auch oft bewusst nicht geöffnet sind (damit keiner versehentlich auf der falschen PDB arbeitet) – diese PBDs müssen dafür zuerst geöffnet werden, wobei hier jeweils auch das Wallet für die PDB geöffnet werden muss.

Je nach Anzahl und Status der PDBs müssen der DBA und der Security Administrator abwechselnd „ihre" Passwörter (SYS, DB_OWNER, Wallet Password) für jede PDB eingeben ... und dabei darf ja keine PDB übersehen werden!

Die Lösung ist hier eigentlich relativ einfach: ein Script, dass auch Spezialfälle abfängt, wie etwa einzelne PDBs aktuell aus irgendeinem Grund nur Read Only zu öffnen. Das Script erfragt am Anfang einmalig allenbenötigten Passwörter, dadurch dauert die ganze Prozedur nur noch einige Minuten (statt Stunden wie am Anfang, weil die Passwörter sehr komplex sind und sich die Kollegen auch das eine oder andere mal vertippt haben).

Real Life Begins – Database Vault und PSU Upgrade auf 12.1.0.2.3

Laut PSU Beschreibung und auch laut Oracle ACE Engineer muss man Database Vault für das PSU Upgrade nicht deaktivieren (für ein Patchset Upgrade schon, laut Dokumentation). Leider entsprach das nicht den Tatsachen, wie wir leidvoll erfahren mussten.

Zu diesem Zeitpunkt waren beim Kunden 12 CDBs im Einsatz, bei 4 CDBs hat das PSU-Upgrade auch problemlos funktioniert, leider aber nicht bei den anderen 8 CDBs. Dort ging das Upgrade derartig schief, dass die PDBs sich nur noch im RESTRICTED-Modus öffnen ließen. Auf Grund der Fehler war schnell klar, es hat etwas mit DBV zu tun, und die dazu passende Empfehlung vom Oracle Support war: „*Deaktivert doch DBV und führt das Upgrade Script nochmals durch*".

Leider gibt es da nur ein Problem. Database Vault lässt sich nicht deaktivieren, wenn die Datenbank im Restricted Mode geöffnet ist. Also gut, dann halt die Datenbanken auf den Zeitpunkt vor dem Upgrade zurücksetzen (Restore und Recovery), Database Vault deaktivieren und nochmals starten. Leider sind wir auf diesem Weg bei 4 CDBs auf ein zusätzliches Problem gestoßen: Die PDB$SEED – ein integraler Bestandteil der Multitenant-Architektur ist immer read only offen. Wenn im RMAN jetzt auch noch BACKUP OPTIMIZATION aktiviert ist, werden read only Dateien nur einmal gesichert. Da die Backups auf ein gemeinsames Fileshare gelegt wurden und im Backup Script nur ein „Crosscheck" dafür gesorgt hat, dass RMAN mitbekommt, dass ein Backup nicht mehr verfügbar ist, war genau für diese 4 CBDs kein Backup der PDB$SEED mehr verfügbar. Oracle Support konnte leider keinen hilfreichen Input anbieten, daher mussten wir uns dann mit einigen Tricks behelfen, um das Problem zu lösen. Hier eine grobe Darstellung des Lösungsweges (ein wenig komplizierter war es leider schon):

- Restore der Multitenant Database und Recovery bis kurz vor dem Start des fehlgeschlagenen Upgrades
- Eliminieren der PDB$SEED Datenbank (nicht supportet, aber unsere einzige Chance)
- Öffnen der CDB ohne PDB$SEED
- Erzeugen einer neuen CDB (mit PDB$SEED)
- Alle PDBs read only öffnen
- UNPLUG der PDBs von der alten CDB und re-plug in die neue CDB, dabei das Oracle Wallet nicht vergessen
- Database Vault deaktivieren
- Das Upgrade durchführen
- Data Guard neue konfigurieren und aufbauen

Sicherheitshalber sollte im RMAN Backup Optimization deaktiviert werden und in den Backup Scripts zusätzliche CROSSCHECKS und DELETE OBSOLETE eingebaut werden, damit RMAN erkennt, wann es kein Backup mehr von der PDB$SEED gibt.

In den Workflow für ein Upgrade sollte fix ein Backup der CDB + PDB$SEED vor dem Upgrade eingeplant werden.

Teil 3 – das Kundenprojekt – lebt immer noch ;-)

Waren dies unsere einzigen Themen im Zusammenspiel mit den verschiedenen Optionen: Nein, definitiv nicht, aber es waren die interessantesten. Letztendlich war es immer möglich, eine praktikable Lösung oder einen Workaround zu finden – in einigen Fällen allein nur durch die exakte Definition von Arbeitsabläufen.

Das wichtigste für den Kunden: Seine Daten (eigentlich die Daten seiner Kunden) sind wirklich so sicher wie es nur geht – und darauf kommt es letztendlich an!

Jede der vorgestellten Funktionalitäten für sich arbeite sehr gut und (fast) fehlerfrei, nur die Kombination – vor allem mit der neuen Multitenant Option – birgt noch einige Stolperfallen. Selbst in der momentan aktuellsten Version (12.1.0.2) sind noch nicht alle beseitig.

Können wir guten Gewissens diese Lösung auch anderen Kunden empfehlen?

Die Antwort ist ein klares JA! Man muss sich nur im Klaren sein, dass etwas mehr Zeit für das Erforschen und Testen der verschiedenen Anwendungsfälle notwendig ist (verglichen mit einer Single Instanz Datenbank, wie wir diese schon seit mehreren Jahrzehnten kennen).

Letztendlich überwiegen die Vorteile und die Arbeitsersparnis der Lösung mit Hilfe der Multitenant Option im Vergleich zu dem Aufwand, der mit vielen Single Instanz Datenbanken zu betreiben ist. Viele Tätigkeiten muss man nur auf CDB (beim Kunden ca. 20 Stück) und nicht auf PDB (aktuell über 500 bei diesem Kunden) machen. Durch genau definierte Arbeitsabläufe und Scripting ist trotz einer großen Komplexität des Environments ein Patchen aller Datenbanken an einem Tag problemlos möglich – stellen Sie sich vor, Sie müssten 500 Single Instanz Datenbanken (plus Data Guard) an nur einem Arbeitstag durchpatchen...

Abschließende Worte:

Wir von DB Masters stehen Ihnen auch gerne jederzeit hilfreich zur Seite, wenn Sie Themen rund um die Oracle Datenbank haben.

Kontaktadresse:

Christian Pfundtner

DB Masters GmbH
Stammersdorfer Str. 463
2201 Gerasdorf
Österreich

Telefon: +43 699 15037884

E-Mail cp@dbmasters.at

Internet: www.dbmasters.at