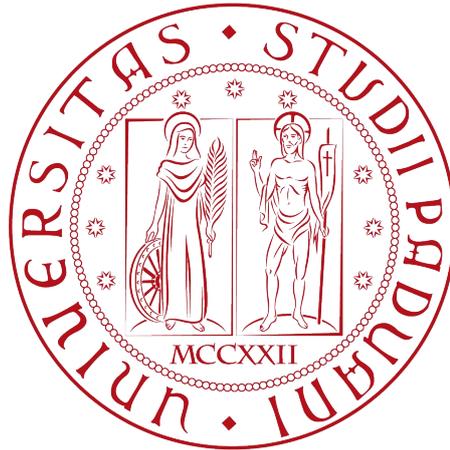


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Zero Downtime Migration di database
Oracle su Cloud**

Tesi di laurea

Relatore

Prof. Alessandro Sperduti

Laureando

Manuel Pagotto

ANNO ACCADEMICO 2021-2022

Manuel Pagotto : *Zero Downtime Migration di database Oracle su Cloud*, Tesi di laurea, © Luglio 2022.

A mio Nonno.

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di trecento ore, dal laureando Manuel Pagotto presso l'azienda TEOD S.r.l., con sede a Napoli (NA).

L'obiettivo dello stage consisteva nella migrazione di un database Oracle on-premise su Oracle Cloud Infrastructure, utilizzando la tecnologia Oracle Zero Downtime Migration (ZDM).

La prima parte dello stage è stata riservata allo studio delle diverse tecnologie che Oracle offre per i diversi tipi di migrazione verso cloud e non. Sono stati studiati, anche se in maniera puramente didattica, i diversi metodi di migrazione database Oracle, in modo da avere un quadro il più completo ed esaustivo possibile.

La seconda parte è stata dedicata puramente allo studio e approfondimento della tecnologia Zero Downtime Migration, alla sua installazione e configurazione. In seguito è stato configurato l'ambiente cloud in cui il database on-premise sarebbe stato migrato. Una volta completata la fase di installazione e configurazione, e completata la fase di test, è stata effettuata la migrazione effettiva del database on-premise verso il cloud Oracle.

“Un pazzo non è che una minoranza formata da una sola persona.”

— George Orwell

Ringraziamenti

Vorrei esprimere la mia gratitudine al mio relatore, il Prof. Alessandro Sperduti, per il supporto che mi ha fornito nella stesura di questa tesi.

Un ringraziamento speciale va ai miei genitori e alla mia famiglia, per l'affetto e il sostegno che mi hanno fornito durante tutti questi anni.

Desidero ringraziare i Delta-cappuccio e gli sGabelli per i bellissimi anni passati insieme. Per terminare, voglio ringraziare i miei amici e tutte le persone a me più care, che sono sempre presenti, nel bene e nel male.

Padova, Luglio 2022

Manuel Pagotto

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Offerta di stage	2
1.3	Progetto di stage	2
1.4	Problematiche riscontrate	2
1.5	Pianificazione	3
1.6	Organizzazione del testo	4
2	Tecnologie	5
2.1	Oracle Cloud Infrastructure	5
2.1.1	Buckets	5
2.1.2	Macchine Virtuali (VM)	5
2.1.3	Oracle Base Database	6
2.2	Sistema Operativo - Oracle Linux	6
2.3	Oracle Zero Downtime Migration	7
2.4	Oracle RMAN	7
2.5	SQL	7
2.6	Comunicazione	8
2.6.1	Microsoft Outlook	8
2.6.2	Microsoft Teams	8
2.7	MobaXterm	9
3	Progetto	11
3.1	Introduzione al progetto	11
3.2	Obiettivi Stage	11
3.3	Requisiti	11
3.3.1	Notazione	11
3.3.2	Obiettivi obbligatori	12
3.3.3	Obiettivi desiderabili	12
3.3.4	Obiettivi facoltativi	12
3.4	Analisi dei Requisiti	13
3.4.1	Requisito 01	13
3.4.2	Requisito 02	13
3.4.3	Requisito 03	13
3.4.4	Requisito 04	14
3.4.5	Requisito D1	14
3.4.6	Requisito F1	14

4	Progettazione e migrazione database su Oracle Cloud	15
4.1	Studio tecnologico	15
4.1.1	Certificazione Oracle Cloud Database Migration and Integration	15
4.2	Configurazione	16
4.2.1	Preparazione host Zero Downtime Migration (ZDM)	17
4.2.2	Configurazione host Sorgente	20
4.2.3	Creazione Object Storage e generazione Authentication Token	21
4.2.4	Creazione e configurazione host target	24
4.3	Migrazione Database	26
4.3.1	Dry Run	26
4.3.2	Go-live	28
5	Conclusioni	33
5.1	Consuntivo finale	33
5.2	Raggiungimento degli obiettivi	34
5.3	Prodotti ottenuti	34
5.4	Conoscenze acquisite	34
	Glossario	35
	Acronimi	39
	Bibliografia	41

Elenco delle figure

1.1	Logo TEOD S.r.l.	1
2.1	Logo OCI.	6
2.2	Logo Outlook.	8
2.3	Logo Teams.	8
2.4	Logo MobaXterm.	9
4.1	Badge Certificazione Oracle Cloud Database Migration and Integration Specialist.	16
4.2	Schermata OCI - Object Storage.	22
4.3	Schermata OCI - Creazione Bucket.	23
4.4	Schermata OCI - Creazione Bucket.	23
4.5	Schermata OCI - Users.	24
4.6	Schermata OCI - Oracle Database.	25

Elenco delle tabelle

1.1	Pianificazione oraria prevista dal piano di lavoro.	3
2.1	Informazioni SO.	6
3.1	Tabella Requisiti Obbligatori.	12
3.2	Tabella Requisiti Desiderabili.	12
3.3	Tabella Requisiti Facoltativi.	12
4.1	Informazioni SO.	25
4.2	Tabella descrizione Response file.	26

5.1	Consuntivazione oraria.	33
5.2	Raggiungimento obiettivi.	34
5.3	Prodotti ottenuti.	34

Capitolo 1

Introduzione

1.1 L'azienda

TEOD è un'azienda con sede a Napoli (NA), che da anni lavora nel settore della consulenza informatica. Operando nei settori delle Banche, Assicurazioni e Pubblica Amministrazione, è in grado di fornire supporto allo sviluppo di progetti, gestione di sistemi informativi e servizi di [Help Desk](#)^[g] in ambito dipartimentale e host. In particolare l'azienda pone in primo piano i seguenti ambiti per la sua consulenza:

- * **Marketing e visual consulting:** l'azienda offre costante supporto nelle politiche di marketing, di contenuti e di supporto visuale e UX.
- * **Consulenza informatica:** viene offerta consulenza informatica sia applicativa che sistemistica, nel supporto costante al cliente, nella proposizione di valore.
- * **Help Desk:** servizi dedicati di help desk di 1 e 2 livello in ambito dipartimentale e host al servizio delle differenti visioni tecniche e di dominio dei clienti.

Il team in TEOD è formato da professionisti di vari settori, dal marketing alla finanza, passando per professionisti in ambito [IT](#)^[g], in modo da offrire al cliente finale una consulenza il più completa e professionale possibile. Professionisti con esperienza pluriennale nel settore ma anche tirocinanti o neo laureati, le persone in TEOD sono molte e variegata, sia in termini di conoscenza che culturalmente.



Figura 1.1: Logo TEOD S.r.l.

1.2 Offerta di stage

Svolgendo già un lavoro inerente al corso di studi, è stato facile reperire il contatto di un'azienda che proponeva un progetto di stage interessante ed in linea con quello che desideravo fare.

Il progetto di stage scelto è rivolto soprattutto al futuro dei database; il cloud infatti sta diventando sempre più importante, diverse aziende stanno portando le proprie infrastrutture da on-premise, quindi anche i loro database, al cloud, principalmente per la facile gestione, per un supporto sempre disponibile e per i costi, che paragonati ad un ambiente on-premise, si riducono drasticamente. Lo spostamento su cloud sembra quindi inevitabile in alcuni ambiti aziendali; con questo tirocinio si affronta appunto una problematica di spostamento di database su cloud, in questo caso Oracle, offrendomi la possibilità poter apprendere le nozioni necessarie per lavorare in questo tipo di ambiente.

1.3 Progetto di stage

Lo scopo principale dello stage era quello di migrare un database, contenente dati reali ed usati da applicazioni esterne, da una soluzione on-premise al Oracle Cloud Infrastructure, possibilmente con un [downtime](#)^[g] del database vicino allo zero.

Per poter realizzare la migrazione è stato necessario configurare i diversi ambienti e macchine che sono andate ad ospitare le tecnologie usate.

Il progetto dunque si può racchiudere nelle seguenti macro-attività:

- * studio della struttura e della gestione dei database Oracle in cloud e on-premise.
- * studio delle tecnologie proprietarie Oracle per la migrazione di database.
- * creazione di una infrastruttura adeguata alla migrazione di database Oracle on-premise a [Oracle Cloud Infrastructure \(OCI\)](#) con [downtime](#) minimo.
- * migrazione del database.

A termine delle seguenti attività sarà prodotto un database su Oracle Cloud con struttura uguale al database on-premise migrato, con relative applicazioni che ora si interfacceranno al nuovo database su cloud.

1.4 Problematiche riscontrate

Una prima problematica riscontrata durante lo stage è stata la distanza fisica con l'azienda ospitante.

Infatti, abitando io a Treviso (TV), non ho avuto la possibilità di svolgere lo stage fisicamente in azienda; il lavoro da remoto e le riunioni svolte tramite videoconferenza, tuttavia, sono riusciti a mitigare questo problema in modo efficiente ed efficace, anche grazie all'ottima corrispondenza con il tutor aziendale.

Un altro problema riscontrato all'inizio dello stage è stato la mia mancanza di conoscenze adeguate per svolgere il progetto. Questo è stato prontamente risolto fornendomi dispense, appunti, video corsi e documentazione ufficiale dalle quali sono riuscito a ricavare tutte le informazioni di base necessarie per avviare in modo consapevole il progetto di stage e per portarlo a termine.

L'installazione del software [ZDM](#)^[g] e la sua configurazione, insieme a quella della sua

Virtual Machine (VM)^[g], sono state fatte per la prima volta. Questo ha prodotto uno studio sulla documentazione ufficiale Oracle, che ha risolte le diverse lacune tecniche dovute all'inesperienza.

Diverse applicazioni interrogavano il database proprietario dell'azienda esterna a *Teod*; per questo nell'ultima fase, quella dello *switchover*^[g], è stato necessario interfacciarsi con i diversi team al fine di modificare la stringa di connessione al database, il che ha richiesto un livello moderato di gestione delle richieste e delle tempistiche, il quale è stato raggiunto con il maturare del livello di comunicazione aziendale.

1.5 Pianificazione

La pianificazione oraria è stata svolta prima dell'inizio dello stage. Nella tabella seguente è possibile visionare il Piano di lavoro.

Durata in ore	Descrizione dell'attività
100	Formazione sulle tecnologie
80	Definizione architettura di riferimento e relativa documentazione
16	Analisi del problema e del dominio applicativo
48	Scelta della soluzione tecnologica di migrazione
16	Stesura documentazione
100	Realizzazione della migrazione
24	Installazione software
16	Configurazione ambiente
40	Dry-run
20	Go live
20	Collaudo Finale
16	Stesura documentazione finale
4	Live demo di tutto il lavoro di stage
Totale ore	300

Tabella 1.1: Pianificazione oraria prevista dal piano di lavoro.

Il piano di lavoro descritto nella tabella 1.1 è servito come indicazione di massima del lavoro che andava svolto. Durante il corso dello stage sono state apportate leggere modifiche alla pianificazione in funzione delle mie necessità e capacità e tarate in base al carico di lavoro effettivo.

1.6 Organizzazione del testo

Nel **secondo capitolo** vengono approfondite le tecnologie e i software utilizzati per la realizzazione del progetto.

Il **terzo capitolo** descrive e approfondisce tutti gli obiettivi e i requisiti del progetto.

Il **quarto capitolo** descrive il lavoro svolto per portare a termine la migrazione del database.

Il **quinto capitolo** è dedicato alle conclusioni riguardanti l'esperienza di stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * Gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * Per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * I termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Tecnologie

In questo capitolo verranno approfondite le tecnologie e i software utilizzati per la realizzazione del progetto.

2.1 Oracle Cloud Infrastructure

Oracle Cloud Infrastructure è un insieme di servizi cloud complementari che consentono di creare ed eseguire un'ampia gamma di applicazioni e servizi in un ambiente scalabile, sicuro, ad alte prestazioni ed a elevata disponibilità. Oracle Cloud Infrastructure (OCI) offre capacità di elaborazione ad alte prestazioni (come istanze hardware fisiche) e capacità di archiviazione in una rete virtuale flessibile accessibile in modo sicuro da qualsiasi rete locale. Per lo stage è stato utilizzato Oracle Cloud Infrastructure aziendale, creando un account personale e collegandolo al sopraccitato. Esistono una grande varietà di servizi, qui verranno descritti quelli utilizzati per lo svolgimento dello stage.

2.1.1 Buckets

I buckets sono contenitori logici per la memorizzazione dei dati. Un bucket è associato ad un singolo [compartimento](#)^[g], il quale possiede delle [policies](#)^[g]. Alla prima inizializzazione di un bucket in OCI, viene deciso quale storage tier (Archive o Standard) è più appropriato per i dati che ci verranno salvati all'interno. La scelta dello storage tier è importante poiché una volta impostato non è possibile cambiarlo; quindi se viene impostato lo storage tier come Archive non sarà possibile convertirlo in Standard, e viceversa.

2.1.2 Macchine Virtuali (VM)

OCI permette la creazione di istanze di macchine virtuali, le quali offrono molte forme di risorse di calcolo. Oracle offre la possibilità di scegliere il taglio della VM, come il numero di processori, la quantità di RAM e la quantità di storage.

2.1.3 Oracle Base Database

Oracle, con la piattaforma Cloud Infrastructure, permette la creazione di [database](#)^[g] single-node in sistemi virtuali o in sistemi [bare metal](#)^[g], con la possibilità di creare sistemi di [database](#)^[g] in [RAC](#)^[g] a 2-nodi, in questo caso solo in macchine virtuali. La gestione di questi database può essere effettuata tramite console, [Application Program Interface \(API\)](#)^[g], [OCI CLI](#)^[g], [Database cli](#)^[g], [Enterprise Manager](#)^[g] o [SQL Developer](#)^[g]. Per questo progetto è stato creato un sistema single-node, con database Oracle versione 19.13; i dettagli verranno spiegati nel [quarto capitolo](#)



Figura 2.1: Logo OCI.

2.2 Sistema Operativo - Oracle Linux

Oracle Linux è un [Sistema Operativo \(OS\)](#)^[g] distribuito da Oracle parzialmente sotto licenza GNU General Public License dal 2006. Codici sorgente, binari e aggiornamenti sono completamente gratuiti e fruibili a chi desidera.

È compilato a partire dalla versione Linux distribuita da Red Hat (RHEL). La versione Oracle Linux viene appunto usata nel Cloud Oracle e in generale in tutti i sistemi Oracle.

Per tutte le macchine presenti nel progetto è stato utilizzato il sistema operativo Oracle Linux, principalmente per una scelta di compatibilità verso i prodotti Oracle, oltre che di semplicità di gestione del sistema operativo ed eventuale scrittura di script.

Di seguito viene riportata una tabella con le informazioni di ciascuna macchina.

	Sorgente	ZDM	Target
OS	Oracle Linux Server 8.5	Oracle Linux Server 7.9	Oracle Linux Server 7.9
Kernel	Linux 4.18.0-348.7.1.el8-5.x86-64	Linux 5.4.17-2136.301.1.3.el7uek.x86-64	Linux 4.1.12-124.58.2.el7uek.x86-64
Architettura	x86-64	x86-64	x86-64

Tabella 2.1: Informazioni SO.

2.3 Oracle Zero Downtime Migration

Zero Downtime Migration è la soluzione Oracle che permette la migrazione di database da ambiente on-premise a OCI, Exadata Cloud @ Customer e Exadata Cloud Service, in modo facile e garantendo un tempo di inattività (Downtime) pari allo zero o in tempo trascurabile, appoggiandosi su altre tecnologie Oracle (come ad esempio [Data Guard](#)^[g]).

ZDM usa meccanismi come effettuare il backup del database sorgente in Oracle Cloud Infrastructure Object Storage ([Buckets](#)), successivamente, nell'ambiente target, crea un database di standby, sincronizza il primario con lo standby e in fine effettua uno switch over verso il database di standby, che diventerà il nuovo primario.

ZDM supporta anche la migrazione offline (backup e recovery). La parte del backup del primario è uguale per la migrazione offline descritta precedentemente; il backup effettuato sul bucket OCI viene usato per istanziare un nuovo database nel Cloud Oracle.

Questo prodotto offre quindi un robusto e flessibile processo di migrazione, inoltre è facile effettuare un [roll back](#)^[g].

Come vedremo più avanti nel capitolo [quattro](#), il software ZDM va installato in un server apposito, il quale non contiene né il database sorgente né quello target; questo server viene chiamato ZDM service host. All'interno del ZDM service host possono essere lanciati uno o più processi di migrazione.

2.4 Oracle RMAN

[Oracle Recovery Manager \(RMAN\)](#) è un software proprietario Oracle. [RMAN](#) è in grado di effettuare backup, restore e ripristini efficienti dei database Oracle.

Progettato per lavorare a stretto contatto con il server sul quale è presente il database Oracle, fornendo un rilevamento a livello di [blocco](#)^[g] durante le fasi di backup e restore. Inoltre ottimizza le performance e l'uso di spazio del database nella fase di backup. Per interfacciarsi con il software è possibile utilizzare una line di comando oppure attraverso [Enterprise Manager](#); con [RMAN](#) si possono schedulare task di backup in ambienti con un sistema operativo diverso l'uno dall'altro, oltre ad offrire features che non sarebbero disponibili in un sistema gestito senza questo software, come ad esempio la parallelizzazione dei data stream dei backup o dei restore, policy di mantenimento dati di backup e log di tutti i backup effettuati.

[RMAN](#) viene usato inoltre per facilitare la migrazione di database da on-premise verso il Cloud Oracle.

2.5 SQL

Linguaggio standard per l'interrogazione, creazione, gestione e amministrazione di un database.

Nasce nel 1974 nei laboratori IBM, e presto diventa uno standard per l'interrogazione dei database.

Così come la maggior parte dei produttori di database relazionali, anche Oracle supporta pienamente il linguaggio SQL.

2.6 Comunicazione

Come già descritto nel [primo capitolo](#), una delle problematiche riscontrate è stata la distanza fisica con l'azienda, questa è stata colmata con alcuni software di comunicazione che vedremo di seguito.

2.6.1 Microsoft Outlook

Outlook ([Figura 2.2](#)) è un programma di posta elettronica, contiene un calendario per le attività e riunioni pianificate, agenda dei contatti. Essendo parte dell'ecosistema Microsoft, ha delle integrazioni con [Microsoft Teams](#), come la possibilità di pianificare un riunione teams ed invitare i partecipanti tramite e-mail.



Figura 2.2: Logo Outlook.

2.6.2 Microsoft Teams

Teams ([Figura 2.3](#)) è un software di comunicazione che racchiude in una singola piattaforma chat, videochiamate, condivisione file (con lavoro in simultaneo sui file) e integrazione di applicazioni (es. calendario, turni, attività). È inoltre possibile creare gruppi di più persone con il quale è possibile fare videoconferenze e chiamate di gruppo. Teams permette anche la condivisione dello schermo da parte dei partecipanti ad una videoconferenza, con la possibilità di prendere il controllo dello schermo condiviso. Questo programma è stato particolarmente utile durante lo stage per le sue funzionalità ed integrazioni con altri software del mondo Microsoft.



Figura 2.3: Logo Teams.

2.7 MobaXterm

Software per sistema operativo Windows, permette la gestione delle connessioni [Secure Shell \(SSH\)](#)^[g] oltre che altri vari network tool per la connessione a terminali remoti. **MobaXterm** ([Figura 2.4](#)) è gratuito ma contiene delle limitazioni, come ad esempio il numero di connessioni remote che è possibile salvare; la versione a pagamento permette lo sblocco di tutte le funzionalità. Per lo svolgimento dello stage non è stato necessario l'acquisto della versione a pagamento.

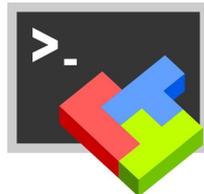


Figura 2.4: Logo MobaXterm.

Capitolo 3

Progetto

In questo capitolo vengono analizzati progetto e requisiti.

3.1 Introduzione al progetto

La migrazione di un database da on-premise a cloud ha diversi benefici, principalmente a livello di budget. Il mantenimento di un ambiente cloud infatti è meno oneroso in termini monetari, ma anche per quanto riguarda la gestione dell'infrastruttura e del database stesso.

3.2 Obiettivi Stage

L'obiettivo principale dello stage è la realizzazione di una migrazione di un database reale su cloud. Il database in oggetto costituisce il repository dello strumento di monitoraggio dei database Oracle aziendali. Contiene, oltre alle metriche di performance e di configurazione dell'infrastruttura cliente, anche i dati storici relativi ai consumi rilevati sui circa 200 database aziendali. La versione del database è Oracle Enterprise Edition 18c. Il numero di tabelle è circa 7000 tabelle con una dimensione totale di 250 GB. Questo database è necessario agli strumenti di monitoraggio enterprise del cliente e presenta vincoli in termini di continuità operativa.

3.3 Requisiti

Per il progetto di stage sono stati individuati degli obiettivi di massima da raggiungere nella durata del tirocinio.

3.3.1 Notazione

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- * **Ox** per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- * **Dx** per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;

- * **F_x** per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Dove **x** è un numero univoco che identifica individualmente l'obiettivo nella sua categoria.

3.3.2 Obiettivi obbligatori

Codice requisito	Breve descrizione
O1	Apprendere le nozioni fondamentali per la gestione di un database Oracle
O2	Conoscere le soluzioni per i database Oracle su OCI (Oracle Cloud Infrastructure) e saper effettuare provisioning di database
O3	Conoscere le principali tecniche di migrazione di database su cloud
O4	Realizzare un caso concreto di migrazione di un database su cloud Oracle

Tabella 3.1: Tabella Requisiti Obbligatori.

3.3.3 Obiettivi desiderabili

Codice requisito	Breve descrizione
D1	Conoscere e saper gestire Oracle Autonomous DB

Tabella 3.2: Tabella Requisiti Desiderabili.

3.3.4 Obiettivi facoltativi

Codice requisito	Breve descrizione
F1	Realizzare la migrazione di un database su Oracle Autonomous DB

Tabella 3.3: Tabella Requisiti Facoltativi.

3.4 Analisi dei Requisiti

Verranno analizzati più nel dettaglio i requisiti descritti nel sottocapitolo [Requisiti](#).

3.4.1 Requisito 01

Devono essere acquisite le conoscenze fondamentali della struttura di un database Oracle e la sua gestione. In particolare devono essere chiari i concetti di:

- * **Database relazionale**
- * **Tabelle e indici**
- * **Transazione (unità atomiche)**
- * **Struttura fisica del database**
- * **Struttura logica del database**
- * **Tool di backup e recovery dati da database**

3.4.2 Requisito 02

Viene appresa la struttura di [OCI](#), con l'obiettivo di poter creare e gestire un Oracle Database System. Vengono quindi approfonditi concetti di base, creazione, mantenimento di un Database System Oracle. Implementazione di [Data Guard](#) nell'Oracle Database System per l'implementazione di sistemi con high availability e disaster recovery. Gestione della sicurezza del database in base ai requisiti e alle esigenze del cliente o del database stesso.

3.4.3 Requisito 03

Vengono studiate e approfondite le principali tecniche di migrazione di un database da on-premise a Cloud Oracle:

- * **Oracle Data Integrator**
- * [RMAN](#)
- * [Data Pump](#)
- * **Clonazione di database**
- * [GoldenGate](#)^[g]
- * [ZDM](#)

Inoltre viene richiesto allo studente di individuare la strategia di migrazione che meglio si adatta al tipo di richiesta dello stage.

3.4.4 Requisito 04

La migrazione di un database on-premise usato da un'azienda cliente dell'azienda ospitante viene effettuata sulla base della tecnologia di migrazione individuata dallo studente. Essendo un database necessario agli strumenti di monitoraggio enterprise del cliente, viene stabilito un disservizio non superiore a 15 minuti.

Nel caso in cui la migrazione non dovesse andare a buon fine, il database sorgente dovrà continuare a rimanere attivo e funzionante. A fine migrazione il risultato atteso sarà che il database sorgente diventerà il secondario e l'Oracle Cloud Database sarà il primario; le applicazioni di monitoraggio aziendali non dovranno cessare il loro normale funzionamento, puntando al nuovo database attraverso una stringa di connessione multipla.

3.4.5 Requisito D1

Oracle Autonomous DB è studiato e approfondito.

- * **Tipi di licenza**
- * **Creazione Autonomous DB**
- * **Scalabilità e auto scalabilità**
- * **Clonare [Autonomous DB \(ADB\)](#)**
- * **Monitorare ed amministrare [ADB](#)**

3.4.6 Requisito F1

Viene richiesta la stessa migrazione del [Requisito 04](#); il database di arrivo (o target) dovrà essere un [ADB](#) situato nell'[Oracle Cloud Infrastructure](#).

La tecnologia di migrazione potrà essere diversa da quella scelta precedentemente se non potrà essere utilizzata per la migrazione di un database on-premise ad [ADB](#).

Capitolo 4

Progettazione e migrazione database su Oracle Cloud

Nel capitolo seguente verrà analizzata la progettazione e l'implementazione dell'infrastruttura OCI e successivamente la migrazione del database da on-premise a OCI

4.1 Studio tecnologico

La prima fase del progetto, come è stato descritto nell'[Analisi dei Requisiti](#), consiste nello studio delle tecnologie che verranno poi usate per il completamento del progetto di stage, quindi la migrazione del database.

Vengono fornite delle guide e dispense ufficiali Oracle per le nozioni base e la gestione di un database. In particolare sono state fornite dal tutor aziendale le seguenti documentazioni:

- * **Oracle Database Concepts**¹: descrive il funzionamento dell'Oracle database server, inoltre pone le basi concettuali per le informazioni pratiche contenute in altri manuali.
- * **Oracle Database 2 Day DBA**²: è una guida che ha lo scopo di descrivere quali attività un amministratore di database deve effettuare giornalmente, incluse le basi di [troubleshooting](#)^[g] e di monitoraggio delle performance.

4.1.1 Certificazione Oracle Cloud Database Migration and Integration

La necessità di apprendere le tecnologie e le tecniche necessarie per una migrazione di un database da on-premise a Oracle Cloud, ha portato il tutor e il manager alla scelta di far frequentare un corso³ Oracle svoltosi online. Il [Product Manager \(PM\)](#) dell'azienda

¹[site:base_db](#).

²Roopesh Ashok Kumar. *Oracle Database 2 Day DBA*. URL: <https://docs.oracle.com/en/database/oracle/oracle-database/19/admqsl/>.

³*Oracle Cloud Database Migration and Integration Specialist*. URL: <https://learn.oracle.com/ols/learning-path/become-an-oracle-cloud-database-migration-and-integration-specialist/35644/92250>.

ha ritenuto opportuno che, una volta terminato il corso, venisse svolto l'esame per la certificazione Oracle Cloud Database Migration and Integration Specialist. Il corso prepara alla migrazione di database su Oracle Cloud. Vengono analizzati i requisiti e l'architettura della migrazione con differenti metodologie, come [RMAN](#), [Data Pump](#)^[g], tramite duplicazione del database, SQL Dev, [GoldenGate](#) e [ZDM](#), in base a quelle che sono le caratteristiche del database sorgente e di quello target.

Al superamento dell'esame, per certificare le conoscenze, Oracle rilascia un [badge](#) nominale valido per circa un anno.



Figura 4.1: Badge Certificazione Oracle Cloud Database Migration and Integration Specialist.

4.2 Configurazione

In seguito allo studio delle varie tecnologie offerte da Oracle per la migrazione di database, viene proposto Oracle [ZDM](#) al tutor aziendale, principalmente perchè questa tecnologia presenta un automatismo che permette una gestione della migrazione migliore e rispetto alle altre tecnologie, dove esiste sì un automatismo ma molte azioni e controlli vanno effettuati a mano dal [Database Administrator \(DBA\)](#)^[g];

[ZDM](#) supporta nelle sue procedure [Data Guard](#), seguendo i principi di [Oracle Maximum Availability Architecture \(MAA\)](#), [RMAN](#) e in [ZDM](#) versione 21c⁴ vengono aggiunti i metodi di migrazione logici usando [Data Pump](#) o [GoldenGate](#).

Un database può essere migrato in due modi, ognuno con metodologia offline o online:

- * **Physical Online:** l'intero database è migrato attraverso il restore del backup del database sorgente, il nuovo database viene istanziato come standby. Il nuovo database viene mantenuto aggiornato tramite [redo](#)^[g] apply. Viene definito online perchè l'unico momento di [downtime](#) è durante lo switchover.
- * **Physical Offline:** l'intero database è migrato attraverso il restore del backup del database sorgente, il nuovo database viene istanziato come standby. Viene definito offline perchè esiste un periodo di [downtime](#) durante la fase di creazione del backup, trasferimento dati e restore in OCI.
- * **Logical Online:** uno o più schemi del database sorgente vengono migrati usando [Data Pump](#). Inoltre, [GoldenGate](#) viene usato per mantenere i due database allineati evitando [downtime](#). Definito online poichè esiste un piccolo [downtime](#) solo nella fase di switchover.

⁴Ricardo Gonzalez. *Oracle Zero Downtime Migration 21c*. 2021. URL: <https://blogs.oracle.com/maa/post/new-oracle-zero-downtime-migration-21c>.

- * **Logical Offline:** è molto simile alla migrazione Logical Online ma senza [Golden-Gate](#) a supporto. Offline perchè durante la fase di export e import il database non è usabile.

Per questa migrazione è stata scelta la migrazione tramite il metodo **Physical Online**, utilizzando come database target un'istanza di database Oracle creata appositamente su [OCI](#).

Questo metodo è stato scelto per diversi motivi

- * Per completare la migrazione è necessario semplicemente uno [switchover](#)^[g] tramite [Data Guard](#), software ampiamente utilizzato da molte aziende, che garantisce zero perdita di dati nel processo. Inoltre il [downtime](#) atteso per questa operazione è vicino allo zero.
- * La possibilità di effettuare un fallback in qualsiasi momento. L'aggiornamento del database standby viene effettuato tramite un flusso [redo](#). Al momento dello [switchover](#) su [OCI](#), il flusso [redo](#) viene invertito e passa al database on-premise che diventa il nuovo standby. In qualsiasi momento questo processo può essere invertito nuovamente senza troppo effort.
- * Il backup iniziale del database on-premise viene effettuato tramite [RMAN](#), il quale usa compressione che riduce drasticamente la grandezza del backup e quindi i dati che verranno mandati su Oracle Cloud

4.2.1 Preparazione host [ZDM](#)

Requisiti minimi

Oracle consiglia di avere una macchina dedicata per l'installazione del software [ZDM](#), separata dai server in cui sono presenti database target e sorgente. Viene quindi create una [VM](#) in [OCI](#) con le seguenti specifiche minime:

- * Oracle Linux 7 o superiore
- * 100 GB di spazio di archiviazione libero
- * Nessuna istanza di [Oracle Grid Infrastructure \(OGI\)](#)^[g]
- * Deve poter accedere in [SSH](#) (porta 22) nei server dei database target e sorgente

L'istanza [VM](#) creata su [OCI](#) viene acceduta tramite connessione [SSH](#) con una chiave privata che viene fornita unicamente al momento della creazione della macchina. Per il collegamento all'istanza [ZDM](#) viene usato il programma [MobaXterm](#). Viene aggiornato il file `/etc/hosts` con gli indirizzi ip dei nodi *target* e *source*. Gli indirizzi ip delle macchine *source* e *target* sono stati oscurati per questioni di sicurezza.

```
[opc@itgzdm01 ~]$ cat /etc/hosts
7X.XX.XXX.X0 itgoemrep.italgas.it source
1XX.XX.XXX.X2 target-scan.subnet.vcn.oraclevcn.com target
10.0.0.100 itgzdm01.subnet.vcn.oraclevcn.com itgzdm01
```

Installazione pacchetti

Come pre-requisito per la macchina [ZDM](#) è necessario inoltre installare i seguenti pacchetti.

- * glibc-devel
- * expect
- * unzip
- * libaio
- * oraclelinux-developer-release-el7

Viene usato [Yellow dog Update, Modified \(YUM\)](#)^[g], invece di [Red Hat Package Manager \(RPM\)](#)^[g], perché al contrario di quest'ultimo, gestisce tutte le dipendenze di un determinato pacchetto al posto dell'utente, stampando a schermo tutte le dipendenze richieste e il volume dei file che verranno scaricati. Se viene accettato di proseguire, tutte le dipendenze verranno scaricate e installate, oltre che al pacchetto specificato.

Creazione gruppo e user

Si creano il gruppo *zdm* e l'user *zdmuser* attraverso i seguenti comandi

```
[root@itgzdm01 ~]$ groupadd zdm
[root@itgzdm01 ~]$ useradd -g zdm zdmuser
```

Successivamente vengono create le cartelle che serviranno in futuro per l'installazione di [ZDM](#) e per il suo corretto funzionamento

```
[zdmuser@itgzdm01 ~]$ mkdir -p /u01/app/oracle/zdm21
[zdmuser@itgzdm01 ~]$ mkdir -p /u01/zdm21-inst
```

Impostazione variabili ambientali

Per una più semplice gestione futura del sistema le variabili d'ambiente necessarie a [ZDM](#) vengono inserite nel file *.bashrc*, che si può descrivere come uno script che viene eseguito ogni volta che si effettua un login con un determinato user (ogni user ha un proprio file *.bashrc*).

```
[zdmuser@itgzdm01 ~]$ echo "INVENTORY_LOCATION=/u01/app/oraInventory;
export INVENTORY_LOCATION" >> ~/.bashrc
[zdmuser@itgzdm01 ~]$ echo "ORACLE_BASE=/u01/app/oracle; export
ORACLE_BASE" >> ~/.bashrc
[zdmuser@itgzdm01 ~]$ echo "ZDM_BASE=\$ORACLE_BASE; export ZDM_BASE" >>
~/.bashrc
[zdmuser@itgzdm01 ~]$ echo "ZDM_HOME=\$ZDM_BASE/zdm21; export ZDM_HOME" >>
~/.bashrc
[zdmuser@itgzdm01 ~]$ echo "ZDM_INSTALL_LOC=/u01/zdm21-inst; export
ZDM_INSTALL_LOC" >> ~/.bashrc
[zdmuser@itgzdm01 ~]$ source ~/.bashrc
```

Installazione ZDM

Il passaggio successivo è l'installazione effettiva del software ZDM, viene scaricato dal sito ufficiale Oracle⁵ sotto forma di file zip^[5] spostato nella directory `/u01/zdm21-inst`. Successivamente viene estratto il file dall'archivio zip. Viene lanciato lo script di installazione del software, le variabili d'ambiente che sono state impostate precedentemente sono utili anche in questa fase dove una svista può compromettere la corretta installazione.

```
[zdmuser@itgzdm01 ~]$ ./zdminstall.sh setup \
    oraclehome=$ZDM_HOME \
    oraclebase=$ZDM_BASE \
    ziploc=$ZDM_INSTALL_LOC/zdm_home.zip -zdm
```

`ziploc` si riferisce al path completo in cui è stato estratto l'archivio zip scaricato in precedenza.

Lancio del servizio ZDM

Per usare ZDM ed effettuare le varie operazioni necessarie per la migrazione del database on-premise, è necessario attivare il servizio `zdm` service. Per effettuare un ulteriore controllo, sull'installazione e sul suo corretto completamento, una volta attivato il servizio viene lanciato il comando di verifica dello status di quest'ultimo; con *Running: true* il software è stato configurato correttamente.

```
[zdmuser@itgzdm01 ~]$ $ZDM_HOME/bin/zdm service start
[zdmuser@itgzdm01 ~]$ $ZDM_HOME/bin/zdm service status
```

Configurazione connettività

Come già detto nel paragrafo [Requisiti minimi](#), la macchina ZDM ha la necessità di comunicare via SSH con l'host *sorgente* e quello *target*.

Per questo motivo è stato necessario inserire nella cartella `/home/zdmuser/.ssh` le chiavi private di entrambi gli host.

Le chiavi sono state generate tramite il comando `ssh-keygen`, senza passphrase e in formato RSA^[5]; una volta portate sulla macchina ZDM sono state rinominate con i nomi assegnati nel file `/etc/hosts`.

```
#host sorgente
[oracle@itgoemrep ~]$ ssh-keygen

#host target
[opc@target ~]$ ssh-keygen
```

Al fine di testare le connettività SSH, vengono effettuate delle connessioni verso gli host *sorgente* e *target* con le chiavi precedentemente generate.

```
[zdmuser@itgzdm01 ~]$ ssh -i /home/zdmuser/.ssh/source root@source
[zdmuser@itgzdm01 ~]$ ssh -i /home/zdmuser/.ssh/target opc@target
```

⁵ Oracle Zero Downtime Migration. URL: <https://www.oracle.com/database/technologies/rac/zdm-downloads.html>.

4.2.2 Configurazione host Sorgente

Connessione **Single Client Access Name (SCAN)**^[g]

Gli host dei database *sorgente* e *target* devono poter connettersi in modo bilaterale tramite **SQL*net**^[g] in modo che le operazioni gestite da **Data Guard** possano essere effettuate.

Quindi viene modificato il file `/etc/hosts` della macchina sorgente con le seguenti stringhe

```
#file /etc/hosts
13X.XX.XXX.X2 target.subnet.vcn.oraclevcn.com target
13X.XX.XXX.X2 target-vip.subnet.vcn.oraclevcn.com target-vip
13X.XX.XXX.X2 target-scan.subnet.vcn.oraclevcn.com target-scan
```

Viene effettuato un test attraverso il comando **tnsping**^[g] sulla porta 1521 (porta di default per database Oracle) per capire se le modifiche sono andate a buon fine

```
[oracle@itgoemrep ~]$ tns ping target:1521

TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 09-JUL-2022
18:46:38

Copyright (c) 1997, 2021, Oracle. All rights reserved.

Used parameter files:
/app/oracle/product/19.0.0/dbhome_1/network/admin/sqlnet.ora

Used HOSTNAME adapter to resolve the alias
Attempting to contact (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=))
                      (ADDRESS=(PROTOCOL=tcp)(HOST=13X.XX.XXX.X2)(PORT=1521)))
OK (0 msec)
```

Transparent Data Encryption (TDE)^[g] wallet

In **OCI** tutti i database vengono criptati di default, e per fare ciò viene usato **TDE**. Il database *sorgente* da migrare non è criptato, ma il database *target*, essendo collocato su **OCI**, viene criptato con **TDE** durante il processo di migrazione su cloud. Per permettere la corretta criptazione del database *target*, **TDE** va configurato nell'host *sorgente*.

Nella directory `/app/oracle/product/19.0.0/dbhome_1/network/admin` è necessario inserire `ENCRYPTION_WALLET_LOCATION` e valorizzarlo (come vediamo nel [blocco di codice](#)) all'interno del file `sqlnet.ora` nel percorso specificato precedentemente.

```
[oracle@itgoemrep admin]$ pwd
/app/oracle/product/19.0.0/dbhome_1/network/admin
[oracle@itgoemrep admin]$ cat sqlnet.ora
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
(METHOD_DATA=(DIRECTORY=/app/oracle/product/19.0.0/dbhome_1/network/admin/)))
```

Il passo successivo è creare un **keystore**^[g]. Per prima cosa è necessario collegarsi al database con:

```
[oracle@itgoemrep admin]$ sqlplus / as sysdba
```

Successivamente, il seguente comando, crea un una serie di file in cui vengono salvate chiavi protette da una *master password* che sarà decisa in fase di creazione della [keystore](#)

```
SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE  
      '/app/oracle/product/19.0.0/dbhome_1/network/admin/' IDENTIFIED BY  
      password;
```

Il [keystore](#) appena creato viene posto in modalità OPEN, in modo da poter modificare le chiavi al suo interno

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY password;
```

Ora che [keystore](#) è in modalità OPEN, viene creata la *master encryption key* usata per la criptazione del database

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY password WITH BACKUP;
```

Secondo le indicazioni ufficiali Oracle, il [TDE](#) wallet è accettato che sia impostato sia su *PASSWORD* sia su *AUTOLOGIN*. Per questo progetto è stata presa la strada del wallet su *AUTOLOGIN* per una questione puramente pratica, infatti con questa modalità non è richiesto nessun input per la password da parte dell'operatore quando vengono richiamate le chiavi. L'ultimo passaggio quindi è impostare la modalità *AUTOLOGIN* sulla [keystore](#) creata in precedenza

```
SQL> ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE  
      '/app/oracle/product/19.0.0/dbhome_1/network/admin/' IDENTIFIED BY  
      password;
```

4.2.3 Creazione Object Storage e generazione Authentication Token

Creazione Object Storage - Bucket

[ZDM](#) con migrazione di tipo *Physical Online* crea un backup del database *sorgente* attraverso il software [RMAN](#). Il backup creato viene inviato attraverso connessione [HyperText Transfer Protocol over Secure Socket Layer \(HTTPS\)](#)^[5] all'interno di object storage creati su [OCI](#).

Per la creazione dell'object storage è necessario accedere alla piattaforma [OCI](#), e andare nella sezione **Storage > Object Storage** del menù a tendina.

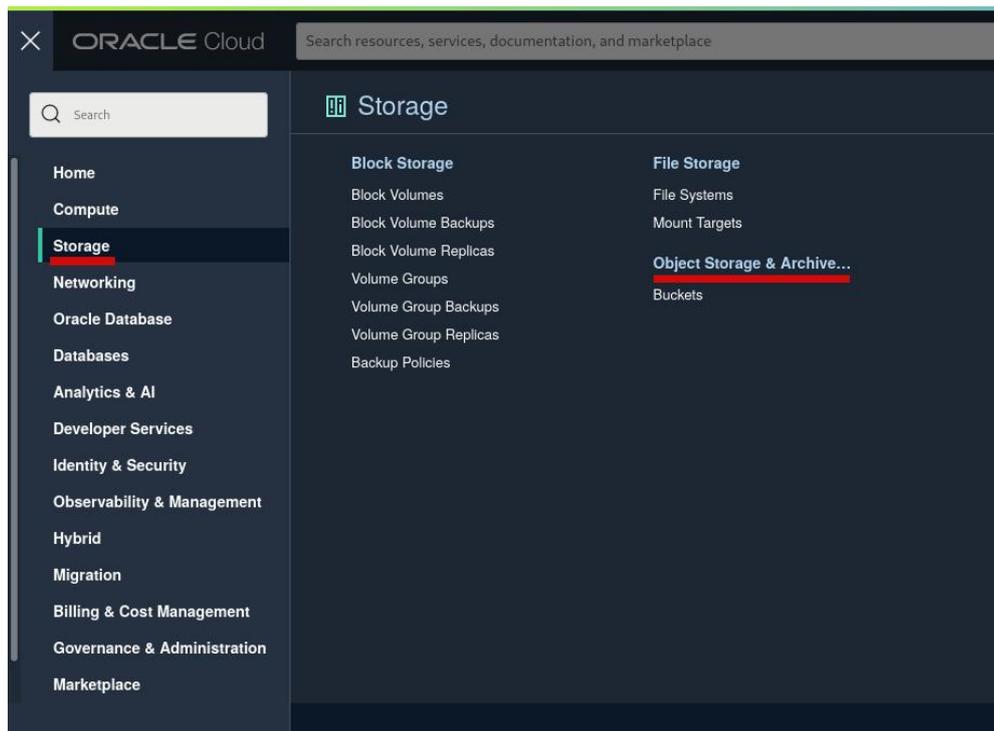


Figura 4.2: Schermata OCI - Object Storage.

Nella schermata successiva possiamo creare il *Bucket* che useremo per contenere il backup del database sorgente.

Con il bottone **Create Bucket** si crea quindi l'*Object Storage*; si seleziona il nome del *Bucket*, si imposta *storage tier* come "Standard" e nell'opzione Encryption si seleziona "ENCRYPT USING ORACLE MANAGED KEYS" poiché verrà sfruttato il [TDE](#) impostato precedentemente nel database *source*.

Create Bucket [Help](#)

Bucket Name

Default Storage Tier
 Standard
 Archive
The default storage tier for a bucket can only be specified during creation. Once set, you cannot change the storage tier in which a bucket resides. [Learn more about storage tiers](#)

Enable Auto-Tiering
Automatically move infrequently accessed objects from the Standard tier to less expensive storage. [Learn more](#)

Enable Object Versioning
Create an object version when a new object is uploaded, an existing object is overwritten, or when an object is deleted. [Learn more](#)

Emit Object Events
Create automation based on object state changes using the [Events Service](#).

Uncommitted Multipart Uploads Cleanup
Create a lifecycle rule to automatically delete uncommitted multipart uploads older than 7 days. [Learn more](#)

Encryption
 Encrypt using Oracle managed keys
Leaves all encryption-related matters to Oracle.
 Encrypt using customer-managed keys
Requires a valid key from a vault that you have access to. [Learn more](#)

Tags
 Optional tags to organize and track resources in your tenancy. [How do I use tags?](#)

Tag Namespace Tag Key Tag Value

None (add a free-form tag) ×

[+ Another Tag](#)

[Create](#) [Cancel](#)

Figura 4.3: Schermata OCI - Creazione Bucket.

Ritornando alla sezione **Object Storage**, il *Bucket* è stato creato correttamente ed è pronto ad ospitare i file di backup

ORACLE Cloud Search resources, services, documentation, and marketplace Germany Central (Frankfurt)

Object Storage & Archive Storage

Buckets *in tecit_ope Compartment*

Object Storage provides unlimited, high-performance, durable, and secure data storage. Data is uploaded as objects that are stored in buckets. [Learn more](#)

[Create Bucket](#)

Name	Default Storage Tier	Visibility	Created
bucket-target	Standard	Public	
	Standard	Private	
	Standard	Private	
	Standard	Public	
	Standard	Public	

Showing 5 Items < 1 of 1 >

Figura 4.4: Schermata OCI - Creazione Bucket.

Generazione Authentication Token

L'Authentication Token serve per autenticare l'accesso a determinati servizi [OCI](#) su una determinata utenza.

Nel nostro caso servirà per l'accesso al [Bucket](#), creato in precedenza, al software [ZDM](#). Attraverso la piattaforma web [OCI](#), si accede all'area **Identity & Security > Users** dal menù a tendina.

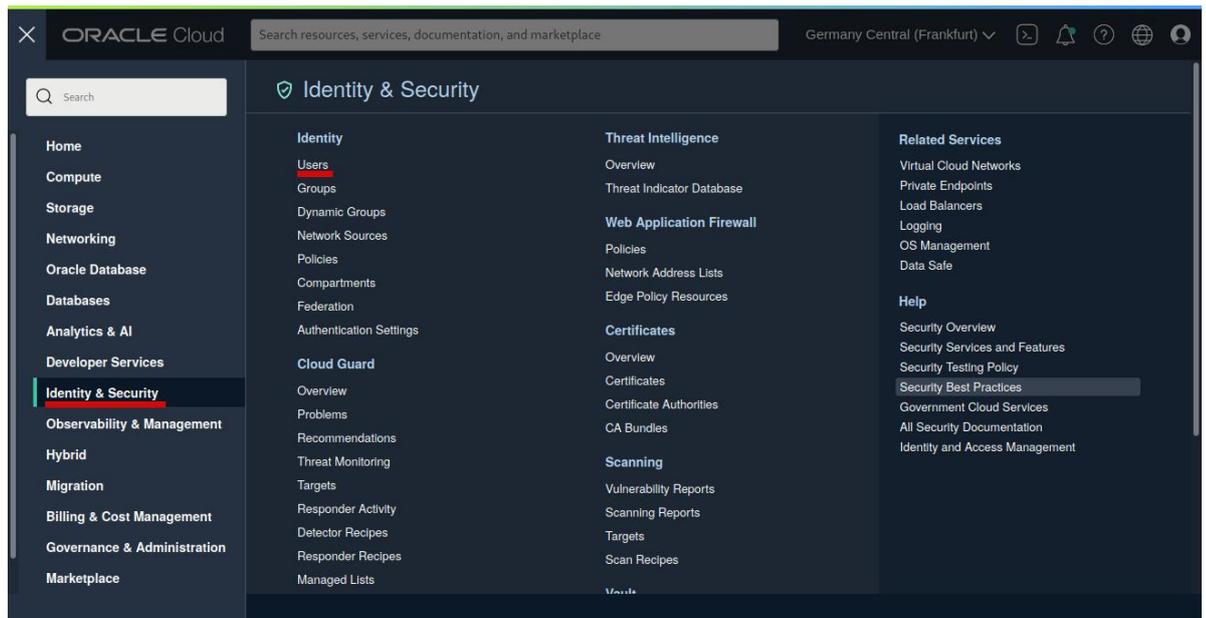


Figura 4.5: Schermata OCI - Users.

Viene selezionato l'utente che accede alla risorsa *Bucket*; nella sotto sezione **Resources > Auth Tokens** viene creato il token che verrà usato da [ZDM](#). Il *token* viene copia e salvato in un file protetto da password; questa operazione è stata effettuata immediatamente dopo la generazione poiché, l'*auth token*, viene visualizzato una sola volta dopodiché, se non salvato, sarà necessario rigenerarlo.

4.2.4 Creazione e configurazione host target

C'è la necessità di creare un ambiente su [OCI](#) che conterrà il database *target*. Operazione che viene fatta con semplicità tramite interfaccia web di [OCI](#).

Per questo progetto si è cercato di mantenere pari o superiori le risorse del server on-premise, in modo da poter gestire il database al meglio. La creazione dell'[Oracle Base Database](#) viene effettuata interamente tramite interfaccia web; tramite il menù a tendina, nella sezione **Oracle Database > Oracle Base Database (VM, BM)** è possibile creare il [DB System](#)^[g].

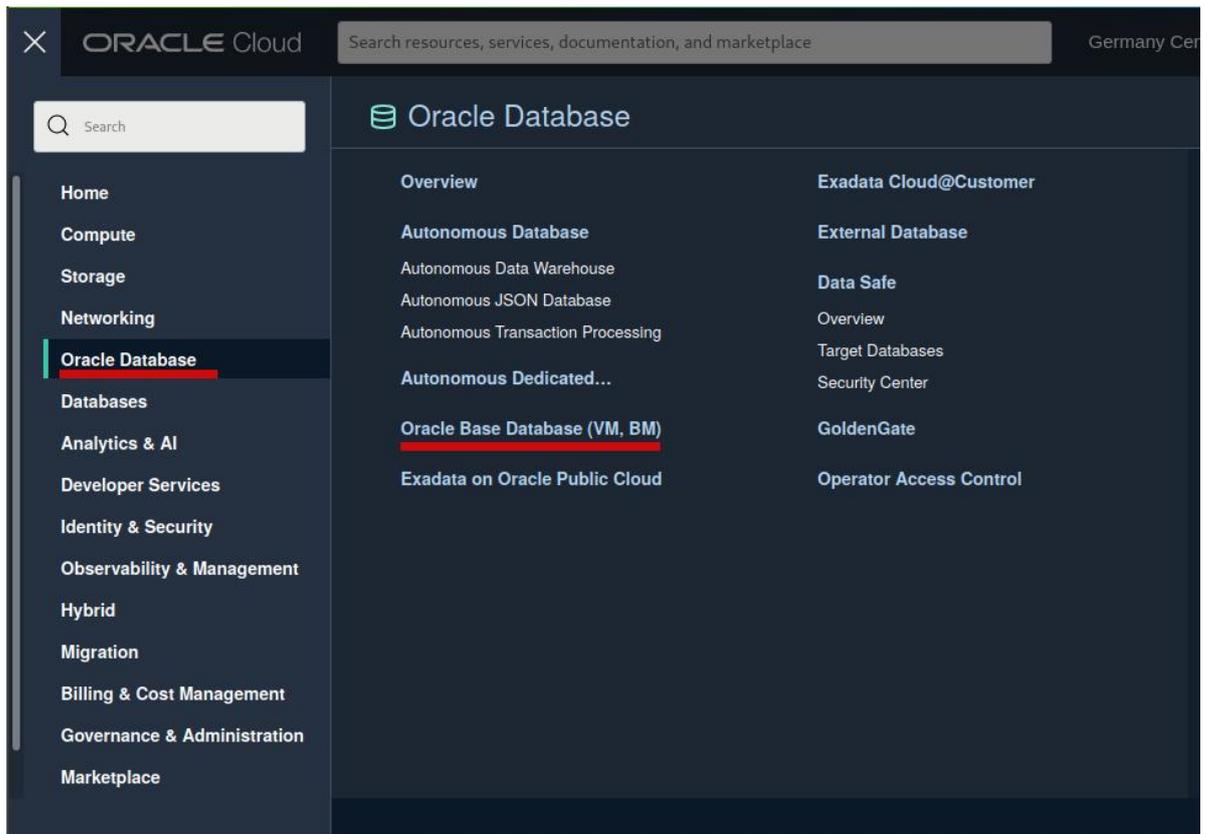


Figura 4.6: Schermata OCI - Oracle Database.

Oracle impone alcuni prerequisiti per il database *target*:

- * La versione del db target deve essere la stessa del db source
- * Il parametro DB_NAME deve essere uguale sul db target e sorgente
- * Il parametro DB_UNIQUE_NAME deve essere diverso in entrambi i database ([Data Guard](#) può identificare il source dal target)

Rispettando questi prerequisiti avremmo un database target con le seguenti caratteristiche

	Sorgente	Target
Versione DB	19.13.0.0.0	19.13.0.0.0
DB_NAME	EMDB	EMDB
DB_UNIQUE_NAME	EMDB	EMDB_target

Tabella 4.1: Informazioni SO.

4.3 Migrazione Database

Nella fase di preparazione precedente sono stati svolti i lavori di configurazione e installazione con lo scopo di migrare un database da un ambiente on-premise a Oracle Cloud.

Seguono due fasi, quella di [dry run](#)^[g] e di [go-live](#)^[g] che vedremo nel dettaglio.

4.3.1 Dry Run

In questa fase viene effettuato un test di migrazione con parametri identici alla fase di migrazione vera e propria; **ZDM** mette a disposizione un comando di valutazione della bontà della migrazione, in questa fase non viene modificato nulla né nel database *sorgente* né in quello *target*.

All'interno dell'host **ZDM** viene preparato il file (Response file) con alcuni parametri usati da **ZDM** per la migrazione.

```
[zdmuser@itgzdm01 template]$ pwd
/u01/app/oracle/zdm21/rhp/zdm/template

[zdmuser@itgzdm01 template]$ cat physical_online.rsp
MIGRATION_METHOD=ONLINE_PHYSICAL
DATA_TRANSFER_MEDIUM=OSS
HOST=https://swiftobjectstorage.eu-frankfurt-1.oraclecloud.com/v1/fr99gki0bywn
OPC_CONTAINER=bucket-target
TGT_DB_UNIQUE_NAME=EMDB_target
PLATFORM_TYPE=VMDB
```

Nella tabella 4.2 vengono descritti i parametri usati per creare il Response file.

Parametro	Breve descrizione
DATA TRANSFER MEDIUM	Il valore di default è Object Storage Service (OSS) . Specifica in che modo viene trasferito il backup del database <i>sorgente</i>
MIGRATION METHOD	Specifica con che metodologia viene migrato il database
HOST	Specifica il REST endpoint URL per accedere ad Oracle Cloud Object Storage (Bucket)
OPC CONTAINER	Specifica il nome del Object Storage Bucket in cui verrà caricato il backup del database <i>sorgente</i>
TGT DB UNIQUE NAME	Nome univoco del database <i>target</i>
PLATFORM TYPE	Specifica su che tipo di macchina si trova il database <i>target</i>

Tabella 4.2: Tabella descrizione Response file.

Il comando per effettuare la valutazione della migrazione è il seguente:

```
$ZDM_HOME/bin/zdmcli migrate database \
-rsp /u01/app/oracle/zdm21/rhp/zdm/template/physical_online.rsp \
-sourcedb EMDB \
```

```
-sourcnode source \
-srcauth zdmauth \
-srcarg1 user:oracle \
-srcarg2 identity_file:/home/zdmuser/.ssh/source \
-srcarg3 sudo_location:/usr/bin/sudo \
-targetnode target \
-tgtauth zdmauth \
-tgtarg1 user:opc \
-tgtarg2 identity_file:/home/zdmuser/.ssh/target \
-tgtarg3 sudo_location:/usr/bin/sudo \
-targethome /u01/app/oracle/product/19.0.0.0/dbhome_1 \
-backupuser "oracleidentitycloudservice/m.pagotto@reply.it" \
-eval
```

Alcuni commenti sul comando che viene lanciato per la fase di valutazione:

- * *sourcnode* e *targetnode* sono i nomi degli host dei database *sorgente* e *target*. Questi nomi sono gli stessi messi all'interno del file */etc/hosts*.
- * *sorucedb* è il **DB_UNIQUE_NAME** del database *sorgente*.
- * *srcarg1* e *tgtarg1* sono i nomi degli utenti con cui **ZDM** si collega, rispettivamente, all'host *sorgente* e *target*.
- * *srcarg2* e *tgtarg2* sono i path dove sono contenute le **chiavi private ssh**.

Tale comando va lanciato dall'utente *zdmuser* sull'host **ZDM**.

Viene chiesta la password per l'utente *SYS* per accedere al database *sorgente*, inoltre per *backupuser* è richiesto l'**auth token** creato in precedenza per l'accesso all'Object Storage Bucket. Fornite le credenziali, il processo di valutazione della migrazione passa alla fase di running.

Attraverso il comando [4.3.1](#) è possibile monitorare lo stato di avanzamento del processo.

```
$ZDM_HOME/bin/zdmcli query job -jobid 1
```

```
itgzdm01.subnet.vcn.oraclevcn.com: Audit ID: 229944
Job ID: 1
User: zdmuser
Client: itgzdm01
Job Type: "EVAL"
Scheduled job execution start time: 2022-01-01T17:35:35Z. Equivalent local
time: 2022-01-01 17:35:35
Current status: EXECUTING
Current Phase: "ZDM_POSTUSERACTIONS"
Result file path:
"/u01/app/oracle/chkbase/scheduled/job-1-2022-01-01-17:35:52.log"
Metrics file path:
"/u01/app/oracle/chkbase/scheduled/job-1-2022-01-01-17:35:52.json"
Job execution start time: 2022-01-01 17:35:52
ZDM_GET_SRC_INFO ..... PRECHECK_PASSED
ZDM_GET_TGT_INFO ..... PRECHECK_PASSED
ZDM_PRECHECKS_SRC ..... PRECHECK_PASSED
ZDM_PRECHECKS_TGT ..... PRECHECK_PASSED
```

```

ZDM_SETUP_SRC ..... PRECHECK_PASSED
ZDM_SETUP_TGT ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS_TGT ..... PRECHECK_PASSED
ZDM_OBC_INST_SRC ..... PRECHECK_PASSED
ZDM_OBC_INST_TGT ..... PRECHECK_PASSED
ZDM_VALIDATE_SRC ..... PRECHECK_PASSED
ZDM_VALIDATE_TGT ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS_TGT .... STARTED
ZDM_CLEANUP_SRC ..... PENDING
ZDM_CLEANUP_TGT ..... PENDING

```

Terminata la valutazione correttamente, il comando si ferma con status *SUCCEEDED*.

```

Current status: SUCCEEDED
Result file path:
  "/u01/app/oracle/chkbase/scheduled/job-1-2022-01-01-17:35:52.log"
Metrics file path:
  "/u01/app/oracle/chkbase/scheduled/job-1-2022-01-01-17:35:52.json"
Job execution start time: 2022-01-01 17:35:52
Job execution elapsed time: 5 minutes 28 seconds
ZDM_GET_SRC_INFO ..... PRECHECK_PASSED
ZDM_GET_TGT_INFO ..... PRECHECK_PASSED
ZDM_PRECHECKS_SRC ..... PRECHECK_PASSED
ZDM_PRECHECKS_TGT ..... PRECHECK_PASSED
ZDM_SETUP_SRC ..... PRECHECK_PASSED
ZDM_SETUP_TGT ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS ..... PRECHECK_PASSED
ZDM_PREUSERACTIONS_TGT ..... PRECHECK_PASSED
ZDM_OBC_INST_SRC ..... PRECHECK_PASSED
ZDM_OBC_INST_TGT ..... PRECHECK_PASSED
ZDM_VALIDATE_SRC ..... PRECHECK_PASSED
ZDM_VALIDATE_TGT ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS ..... PRECHECK_PASSED
ZDM_POSTUSERACTIONS_TGT .... PRECHECK_PASSED
ZDM_CLEANUP_SRC ..... PRECHECK_PASSED
ZDM_CLEANUP_TGT ..... PRECHECK_PASSED

```

Essendo la fase di test terminata senza errori, è possibile passare alla fase [go-live](#).

4.3.2 Go-live

Questa è la fase più delicata dell'intero progetto. Il [dry run](#) si è concluso correttamente, viene usata nuovamente il comando lanciato in precedenza con la sola differenza che il parametro *-eval*, che forzava [ZDM](#) a effettuare solo la valutazione della migrazione, ora viene omesso. Viene aggiunto un altro parametro, *-pauseafter* che permette di mettere in pausa [ZDM](#) in un determinato punto della migrazione; in questo caso la pausa avviene subito dopo la fase di *ZDM_CONFIGURE_DG_SRC*, quindi prima che avvenga lo [switchover](#) nella fase *ZDM_SWITCHOVER_SRC*.

```
itgzdm01.subnet.vcn.oraclevcn.com: Audit ID: 230387
```

```

Job ID: 2
User: zdmuser
Client: itgzdm01
Job Type: "MIGRATE"
Scheduled job execution start time: 2022-02-11T19:05:45Z. Equivalent local
time: 2022-02-11 19:05:45
Current status: PAUSED
Current Phase: "ZDM_SWITCHOVER_SRC"
Result file path:
"/u01/app/oracle/chkbase/scheduled/job-2-2022-02-11-19:05:53.log"
Metrics file path:
"/u01/app/oracle/chkbase/scheduled/job-2-2022-02-11-19:05:53.json"
Job execution start time: 2022-02-11 19:05:53
ZDM_GET_SRC_INFO ..... COMPLETED
ZDM_GET_TGT_INFO ..... COMPLETED
ZDM_PRECHECKS_SRC ..... COMPLETED
ZDM_PRECHECKS_TGT ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_SETUP_TGT ..... COMPLETED
ZDM_PREUSERACTIONS ..... COMPLETED
ZDM_PREUSERACTIONS_TGT ..... COMPLETED
ZDM_OBC_INST_SRC ..... COMPLETED
ZDM_OBC_INST_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_BACKUP_FULL_SRC ..... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC ..... COMPLETED
ZDM_COPYFILES ..... COMPLETED
ZDM_PREPARE_TGT ..... COMPLETED
ZDM_SETUP_TDE_TGT ..... COMPLETED
ZDM_CLONE_TGT ..... COMPLETED
ZDM_FINALIZE_TGT ..... COMPLETED
ZDM_CONFIGURE_DG_SRC ..... COMPLETED
ZDM_SWITCHOVER_SRC ..... PENDING
ZDM_SWITCHOVER_TGT ..... PENDING
ZDM_POST_DATABASE_OPEN_TGT .... PENDING
ZDM_DATAPATCH_TGT ..... PENDING
ZDM_POST_MIGRATE_TGT ..... PENDING
ZDM_POSTUSERACTIONS ..... PENDING
ZDM_POSTUSERACTIONS_TGT ..... PENDING
ZDM_CLEANUP_SRC ..... PENDING
ZDM_CLEANUP_TGT ..... PENDING

Pause After Phase: "ZDM_CONFIGURE_DG_SRC"

```

A questo punto della migrazione il database secondario di standby è stato costruito in [OCI](#). La sincronizzazione sta avvenendo tramite stream di file [redo](#), i due database sono quindi allineati.

```

#database SORGENTE
SQL> select database_role from v$database;
DATABASE_ROLE

```

```

-----
PRIMARY

#database TARGET
SQL> select database_role from v$database;
DATABASE_ROLE
-----
PHYSICAL STANDBY

```

Mentre il job [ZDM](#) è in stato "PAUSED" e il database *target* su [OCI](#) (PHYSICAL STANDBY) è costantemente allineato con il database *sorgente* (PRIMARY), l'area aziendale che si occupa dello sviluppo degli applicativi viene avvisata della situazione attuale permettendo di sostituire la stringa di connessione ad un singolo database con la stringa per database multipli. Per motivi di copyright non è possibile fornire il codice sorgente originale delle applicazioni aziendali, tuttavia viene fornito un esempio di stringa di connessione a database multipli.

```

(DESCRIPTION=
  (ADDRESS_LIST =
    (ADDRESS=(PROTOCOL=TCP) (HOST=source) (PORT=1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=target) (PORT=1521))
  )
  (CONNECT_DATA=
    (SERVICE_NAME=EMDB)
  )
)

```

In seguito alla modifica della applicazioni che puntavano al database *sorgente* è possibile riprendere il job [ZDM](#) da dove era stato interrotto, quindi subito prima dello [switchover](#).

```
[zdmuser@itgzdm01 ~]$ $ZDM_HOME/bin/zdmcli resume job -jobid 2
```

Una volta completato del tutto il job, viene ri-effettuato il controllo del ruolo del database.

```

#database SORGENTE
SQL> select database_role from v$database;
DATABASE_ROLE
-----
PHYSICAL STANDBY

#database TARGET
SQL> select database_role from v$database;
DATABASE_ROLE
-----
PRIMARY

```

La migrazione è quindi terminata con successo; le applicazioni, grazie ad una semplice modifica alla stringa di connessione, puntano al primario, che in seguito allo [switchover](#) effettuato da [ZDM](#), è il database *target* in [OCI](#). Il database on-premise rimane il *PHYSICAL STANDBY*, allineato costantemente con il database Oracle Cloud. Abbiamo quindi completato la migrazione con un tempo di down di molto

inferiore ai 15 minuti come previsto da requisito.

Capitolo 5

Conclusioni

5.1 Consuntivo finale

La pianificazione oraria preventivata non è stata rispettata del tutto; le ore effettive sono state distribuite diversamente in quanto la pianificazione è stata effettuata su basi puramente teoriche e senza una precedente esperienza. La **Formazione sulle tecnologie** ha rispettato il preventivo in quanto il materiale teorico era molto e le conoscenze con cui partivo erano basiche. In più come già spiegato in [sottosezione 4.1.1](#), è stata conseguita la certificazione Oracle Cloud Database Migration and Integration. Nella **Definizione architettura di riferimento e relativa documentazione** sono state preventivate ore in più rispetto a quelle effettivamente lavorate, soprattutto nella *Scelta della soluzione tecnologica di migrazione* la cui quasi totalità delle ore è stata assorbita dalla **Formazione sulle tecnologie**. **Realizzazione della migrazione** complessivamente ha impiegato più ore di quanto preventivato, infatti le ore effettivamente lavorate nelle varie fasi si discostano dal preventivo iniziale; in particolare la *Configurazione ambiente* ha impiegato più tempo del previsto poiché l'architettura, nonostante non sia molto complessa, doveva essere predisposta nei migliori dei modi così che in fase di [sottosezione 4.3.1](#) e [sottosezione 4.3.2](#) ci fossero meno problemi possibile. La fase di dry run infatti è stato sovrastimata, mentre quella di go-live ha richiesto più tempo del previsto a causa dell'iter dietro le modifiche delle stringe di connessione delle applicazioni.

Per finire, il **Collaudo Finale** si è rivelato in linea con le stime iniziali.

Ore preventivate	Ore consuntivate	Descrizione dell'attività
100	100	Formazione sulle tecnologie
80	62(-18)	Definizione architettura di riferimento e relativa documentazione
16	16	Analisi del problema e del dominio applicativo
48	30(-18)	Scelta della soluzione tecnologica di migrazione
16	16	Stesura documentazione
100	118(+18)	Realizzazione della migrazione
24	26(+2)	Installazione software
16	30(+14)	Configurazione ambiente
40	30(-10)	Dry-run
20	32(+12)	Go live
20	20	Collaudo Finale
16	16	Stesura documentazione finale
4	4	Live demo di tutto il lavoro di stage
Totale ore	300	300

Tabella 5.1: Consuntivazione oraria.

5.2 Raggiungimento degli obiettivi

Alla fine del tirocinio tutti gli obiettivi obbligatori e desiderabili sono stati completati. L'obiettivo facoltativo non è stato possibile completarlo per mancanza di tempo. Si può vedere un riassunto tabellare degli obiettivi raggiunti e non alla [Tabella 5.2](#).

Codice requisito	Completato
O1	Si
O2	Si
O3	Si
O4	Si
D1	Si
F1	No

Tabella 5.2: Raggiungimento obiettivi.

5.3 Prodotti ottenuti

Con il concludersi del progetto di stage, i seguenti prodotti sono stati ottenuti:

Prodotto	Quantità	Descrizione
Host Zero Downtime Migration	1	Virtual machine presente su Oracle Cloud. Configurata e utilizzabile per svolgere altre migrazioni su Oracle Cloud oltre a quella del progetto di stage.
Database Oracle Cloud	1	Database migrato da on-premise su Oracle Cloud. Completamente up&running. Utilizzato come database Primario da azienda esterna a Teod.
Database on-premise STANDBY	1	Database che era già presente prima della migrazione come primario. Viene utilizzato come secondario, costantemente allineato al primario. In qualsiasi momento è possibile effettuare un nuovo switchover per ripristinarlo come primario.
Documento step-by-step	1	Documento che descrive le attività tecniche passo per passo. Contiene informazioni utili su installazione software, configurazione vm ed errori comuni e non che si sono presentati durante il progetto. Documento interno utile per una futura migrazione.
Certificazione Oracle	1	Durante il progetto è stata conseguita la certificazione Oracle Cloud Database Migration and Integration con validità annuale.

Tabella 5.3: Prodotti ottenuti.

5.4 Conoscenze acquisite

Il tirocinio è stato molto formativo, ho visto da vicino il ciclo di vita di un progetto relativamente lungo e complesso, in una realtà professionale come quella di Teod e i suoi clienti. È stato particolarmente interessante la parte formativa, soprattutto per un possibile inserimento all'interno del team [DBA](#)^[8] dell'azienda. L'ambiente che ho trovato, nonostante lo svolgimento a distanza, è stato rilassato e la gestione da parte del tutor e del [PM](#) è stata ottimale. Il progetto svolto è stato utile e rivolto al futuro delle basi dati e del cloud, per questo mi ritengo soddisfatto dell'esperienza che ho avuto.

Glossario

API *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. [37](#)

Bare Metal Server in cui tutte le sue risorse sono messe a disposizione esclusivamente per un cliente; viene chiamato anche "Server dedicato". [6](#), [35](#)

Blocco Blocco di memoria in cui vengono salvati dati. [7](#), [35](#)

Compartimento Raccolta di risorse correlate a cui è possibile accedere solo dai gruppi che ha ricevuto l'autorizzazione da un amministratore dell'organizzazione. [5](#), [35](#)

Data Guard Software proprietario Oracle che assicura alta affidabilità, protezione dei dati e disaster recovery sui dati. [7](#), [13](#), [16](#), [17](#), [20](#), [25](#), [35](#)

Data Pump Software che permette il trasferimento di dati e metadati da un database all'altro. [13](#), [16](#), [35](#)

Database Insieme di informazioni (o dati) strutturate in genere archiviate elettronicamente in un sistema informatico. [6](#), [35](#)

Database CLI Interfaccia a linea di comando disponibile nei sistemi DB system su bare metal o virtual machine. [6](#), [35](#)

DB System Sistema composto da database e macchina su cui viene ospitato. [24](#), [35](#)

DBA Amministratore di sistemi informatici specializzato nell'organizzazione e gestione di dati. [37](#)

Downtime Periodo in cui il database non è disponibile. [2](#), [16](#), [17](#), [35](#)

Dry run Processo della fase di test dove gli effetti di un possibile fallimento vengono mitigati. [26](#), [35](#)

Enterprise Manager Software che utilizzato per il monitoraggio e la gestione di Oracle database situati su cloud o on-premise. [6](#), [7](#), [35](#)

- Go-live** Ultima fase di un progetto. In questa fase il prodotto viene reso disponibile al cliente in una forma usabile e funzionante. [26](#), [35](#)
- GoldenGate** Software proprietario Oracle che permette la replica, trasformazione, filtraggio dei dati da un database all'altro. [13](#), [16](#), [17](#), [35](#)
- Help Desk** Servizio professionale aziendale, in buona parte orientato alla risoluzione di problemi, volto a fornire assistenza/supporto tecnico e/o informativo, all'utente/cliente, relativamente all'acquisto e/o utilizzo di prodotti elettronici o servizi informatici, con lo scopo dunque di fornire indicazioni o risolvere problemi su prodotti hardware come computer, apparecchiature elettroniche o software. [1](#), [35](#)
- IT** Insieme dei metodi e delle tecnologie che vengono utilizzate in ambito pubblico, privato o aziendale per l'archiviazione, la trasmissione e l'elaborazione di dati e informazioni attraverso l'uso di reti (reti aziendali, internet ecc.), elaboratori (PC, server, mainframe ecc.) e attrezzature di telecomunicazione (datacenter, router, smartphone, tablet, GPS ecc.). [1](#), [36](#)
- Keystore** Insieme di file che contengono password criptate. [20](#), [21](#), [36](#)
- MAA** Fornisce le best practice per architetture, configurazioni e life cycle per Oracle database, abilitando servizi high-availability per database che risiedono su sistemi on-premise, cloud, o configurazioni ibride. [37](#)
- OCI CLI** Oracle Cloud Infrastructure Command Line Interface, o più semplicemente OCI CLI, è uno strumento che può essere utilizzato da solo o con la console per interfacciarsi con Oracle Cloud. [6](#), [36](#)
- OGI** Software Oracle che fornisce supporto di sistema ai database Oracle come gestione del volume dati, file system e restart automatici. [37](#)
- Gruppo di regole usate come base per azioni che si possono intraprendere. [5](#), [36](#)
- RAC** Opzione del database di Oracle che consente a due o più server di lavorare sulla stessa istanza del database Oracle. [6](#), [36](#)
- Redo** File che salvano tutti i cambiamenti in un database nel momento in cui avvengono. [16](#), [17](#), [29](#), [36](#)
- Roll back** Operazione che permette di riportare la base di dati a una versione o stato precedente. [7](#), [36](#)
- RPM** RPM è uno dei primi sistemi di gestione dei pacchetti creati per distribuzioni GNU/Linux. [37](#)
- RSA** Algoritmo usato per criptare chiavi ssh. [19](#), [36](#)
- SCAN** SCAN è una feature di Oracle RAC (Real Application Clusters) che fornisce un singolo nome per per i client per l'accesso ad Oracle Database. [37](#)
- Sistema Operativo** Software che gestisce le risorse hardware e software della macchina, fornendo servizi di base ai software applicativi. [37](#)

- SQL Developer** Ambiente di sviluppo integrato per lavorare con SQL nei database Oracle. [6](#), [36](#)
- SQL* net** Software proprietario Oracle, il quale permette l'accesso remoto ai dati tra programmi e database o tra database-database. [20](#), [36](#)
- SSH** Protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica. [37](#)
- Switchover** Cambio di ruolo tra un database primario e uno dei suoi database standby. [3](#), [17](#), [28](#), [30](#), [36](#)
- Tnsping** Programma che permette di determinare quando il listener per un servizio su rete Oracle Net può essere raggiunto correttamente. [20](#), [36](#)
- Troubleshooting** Processo di ricerca logica e sistematica delle cause di un problema su un prodotto o processo, in genere a partire dal monitoraggio del sistema, affinché possa essere successivamente risolto e il sistema torni ad essere nuovamente operativo. [15](#), [36](#)
- VM** Software che attraverso il processo di virtualizzazione, crea un ambiente virtuale che emula il comportamento di una macchina fisica attraverso l'assegnazione di risorse hardware. [37](#)
- YUM** Yum o Yellow dog Update, Modified è un manager di pacchetti sviluppato dalla Duke University per migliorare l'installazione degli [RPM](#). [37](#)
- Zip** Formato libero di compressione dei dati lossless. [19](#), [36](#)

Acronimi

- ADB** [Autonomous DB](#). 14, 37
- API** [Application Program Interface](#). 6
- DBA** [Database Administrator](#). 16, 34
- HTTPS** [HyperText Transfer Protocol over Secure Socket Layer](#). 21
- MAA** [Oracle Maximum Availability Architecture](#). 16
- OCI** [Oracle Cloud Infrastructure](#). 2, 17, 20, 21, 24, 29, 30
- OGI** [Oracle Grid Infrastructure](#). 17
- OS** [Sistema Operativo](#). 6
- OSS** [Object Storage Service](#). 26
- PM** [Product Manager](#). 15, 34
- RMAN** [Oracle Recovery Manager](#). 7, 16
- RPM** [Red Hat Package Manager](#). 18
- SCAN** [Single Client Access Name](#). 20
- SSH** [Secure Shell](#). 9, 17, 19
- TDE** [Transparent Data Encryption](#). 20–22
- VM** [Virtual Machine](#). 3, 17
- YUM** [Yellow dog Update, Modified](#). 18
- ZDM** [Zero Downtime Migration](#). x, 2, 16–19, 21, 24, 26–28, 30

Bibliografia

Siti web consultati

- [1] Roopesh Ashok Kumar. *Oracle Database 2 Day DBA*. URL: <https://docs.oracle.com/en/database/oracle/oracle-database/19/admqsl/> (cit. a p. 15).
- [2] *Oracle Cloud Database Migration and Integration Specialist*. URL: <https://learn.oracle.com/ols/learning-path/become-an-oracle-cloud-database-migration-and-integration-specialist/35644/92250> (cit. a p. 15).
- [3] Ricardo Gonzalez. *Oracle Zero Downtime Migration 21c*. 2021. URL: <https://blogs.oracle.com/maa/post/new-oracle-zero-downtime-migration-21c> (cit. a p. 16).
- [4] *Oracle Zero Downtime Migration*. URL: <https://www.oracle.com/database/technologies/rac/zdm-downloads.html> (cit. a p. 19).