DBRE

Database Reliability Engineering

Designing and operating resilient database systems

Alexandre Fagundes

LAD Partner Enablement

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

About Our Live Sessions

- What is DBRE?
- IaC
  - Basics
  - 1st Deploy – Compute Instances
  - 2nd Deploy – DB with DG
- Performance ADB
  - Basics
  - Benchmarks

What is DBRE?

Database Reliability Engineering is a subfield of SRE.

The way SRE holistically deals with the reliability of all the systems for a company, DBRE deals with all the systems of the data infrastructure of a company.

- Service-Level Management
- Risk Management
- Operational Visibility
- Infrastructure Engineering
- Infrastructure Management
- Backup & Recovery
- Release Management
- Security
- Data Storage, Indexing, Replication

IaC

Infastructure-as-code

Infrastructure-as-Code → Terraform

- **Terraform**
  - **Open Source**
  - **Cloud Agnostic Tool**
  - **Build versioned Infra**
  - **Declarative Configuration**
  - **Deploy Faster**
  - **Ideal for repetitive tasks**
- **OCI Resource Manager**
- **OCI Cloud Shell**

- **Provision of OCI Resources**

Example:

- Networks
- Compute instances
- Databases

Terraform Basics

init - command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times.

plan - The terraform plan command is used to create an execution plan. Terraform performs a refresh, unless explicitly disabled, and then determines what actions are necessary to achieve the desired state specified in the configuration files.

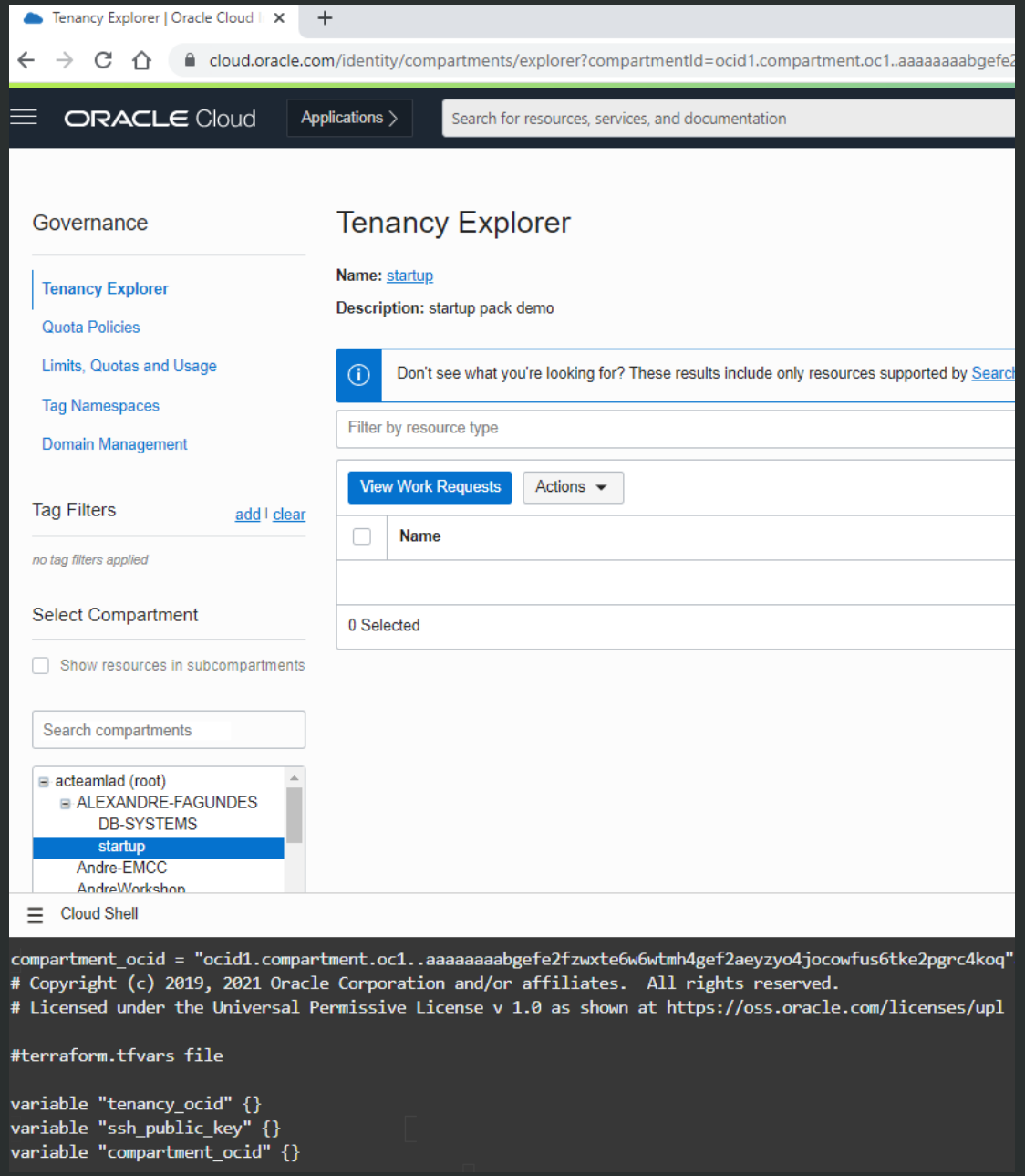
apply - The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.

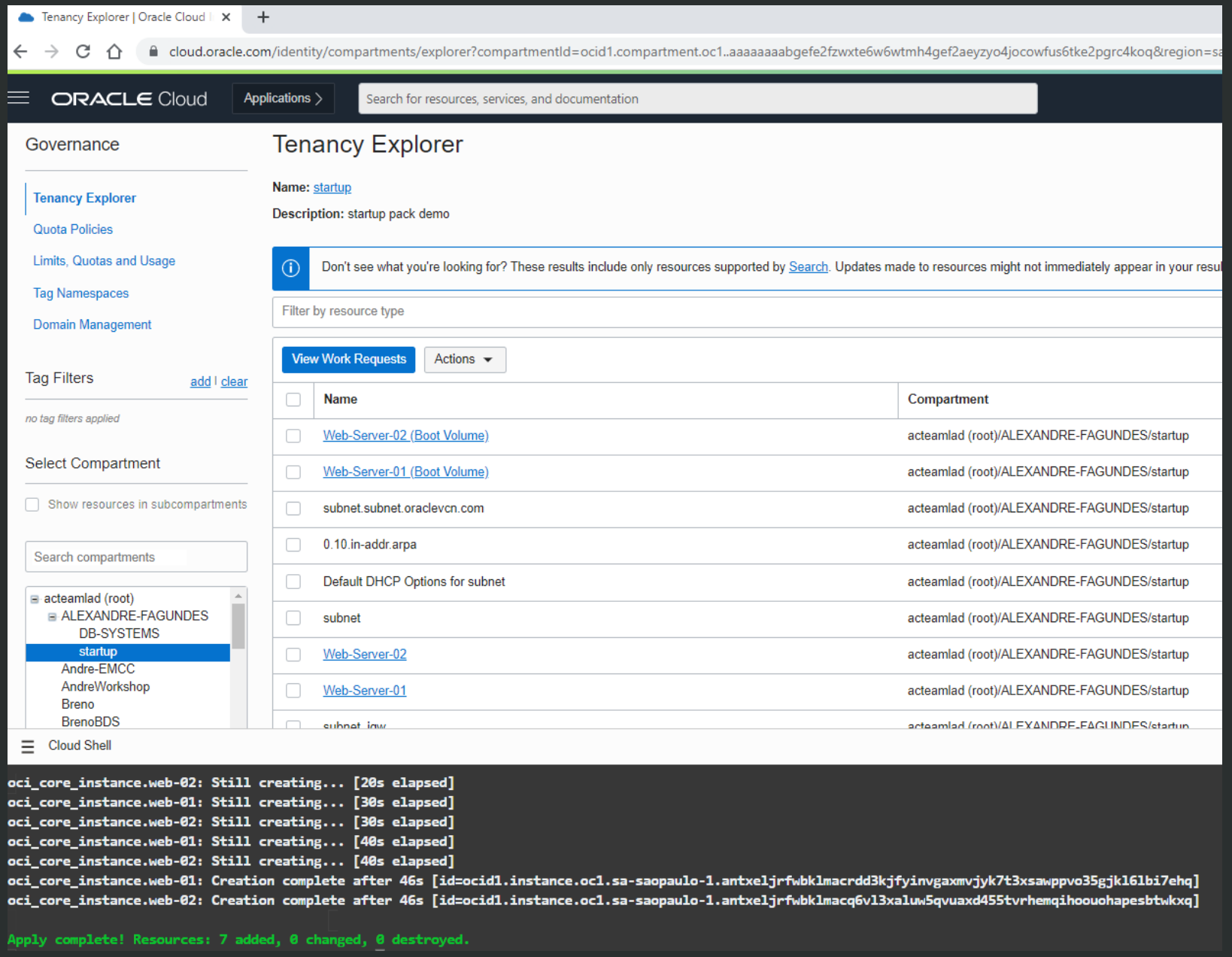
destroy - The terraform destroy command is used to destroy the Terraform-managed infrastructure.

https://www.terraform.io/docs/commands/

How can we do that?

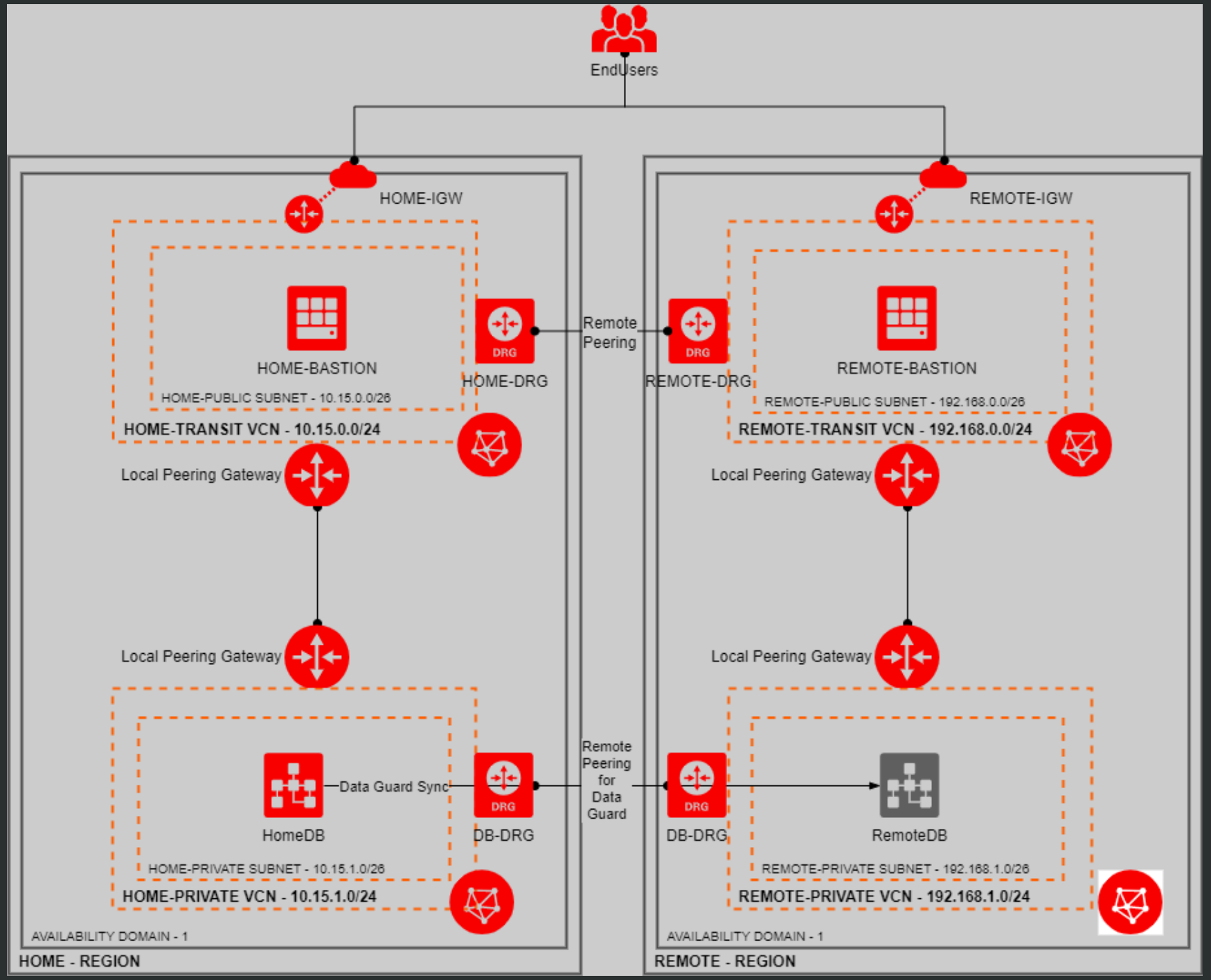
DEMO





1st sample Deployment

VCN
Subnet
Security List
Route Table
Internet Gateway
Boot Volumes
vNICs
Compute instances



DB with DG Association

2nd Deployment

2 VCNs
2 Subnets
2 Security Lists
2 Route Table
2 Internet Gateway
2 Boot Volumes
4 vNICs
2 Compute instances
1 Primary DB
1 Stand-by DB (Different Region)
Dataguard Association

Wrap Up – Deploying Resources

Using IaC & Cloud Shell

Rapidly
Easily
If necessary, repeatedly

1st deploy
https://github.com/alefagun/code

2nd deploy
https://github.com/alefagun/startup

Performance

Performance

Oracle Autonomous Database through good Database Design

Benchmarks

- The total transactions per second achieved (TPS)
- The average elapsed time in seconds across the 8 benchmark sessions (ELA)
- A cumulative breakdown of where the 8 sessions utilized the elapsed time

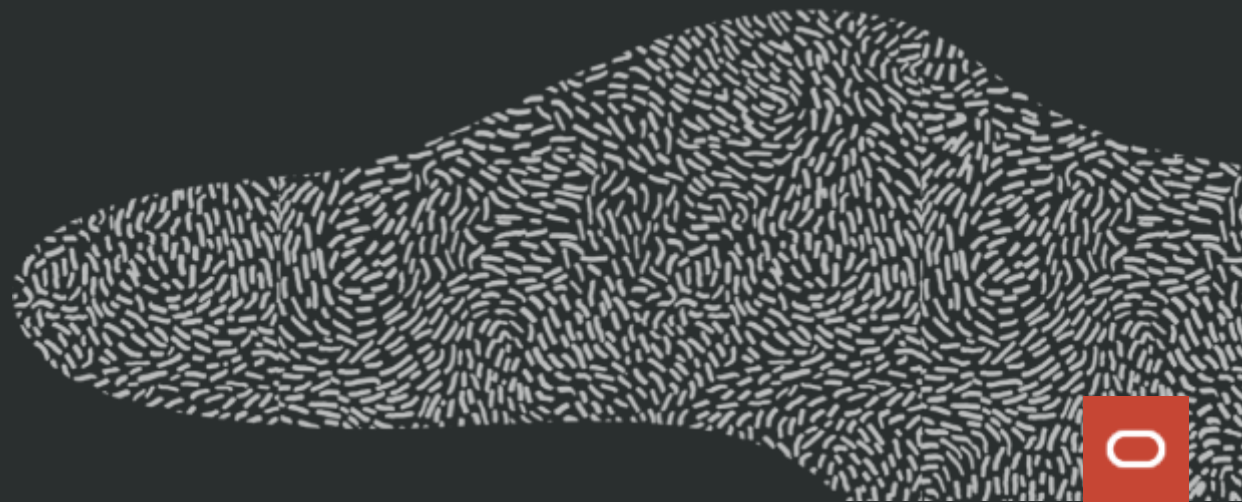
Performance Benchmarks

Benchmark 1: Launches 8 parallel sessions

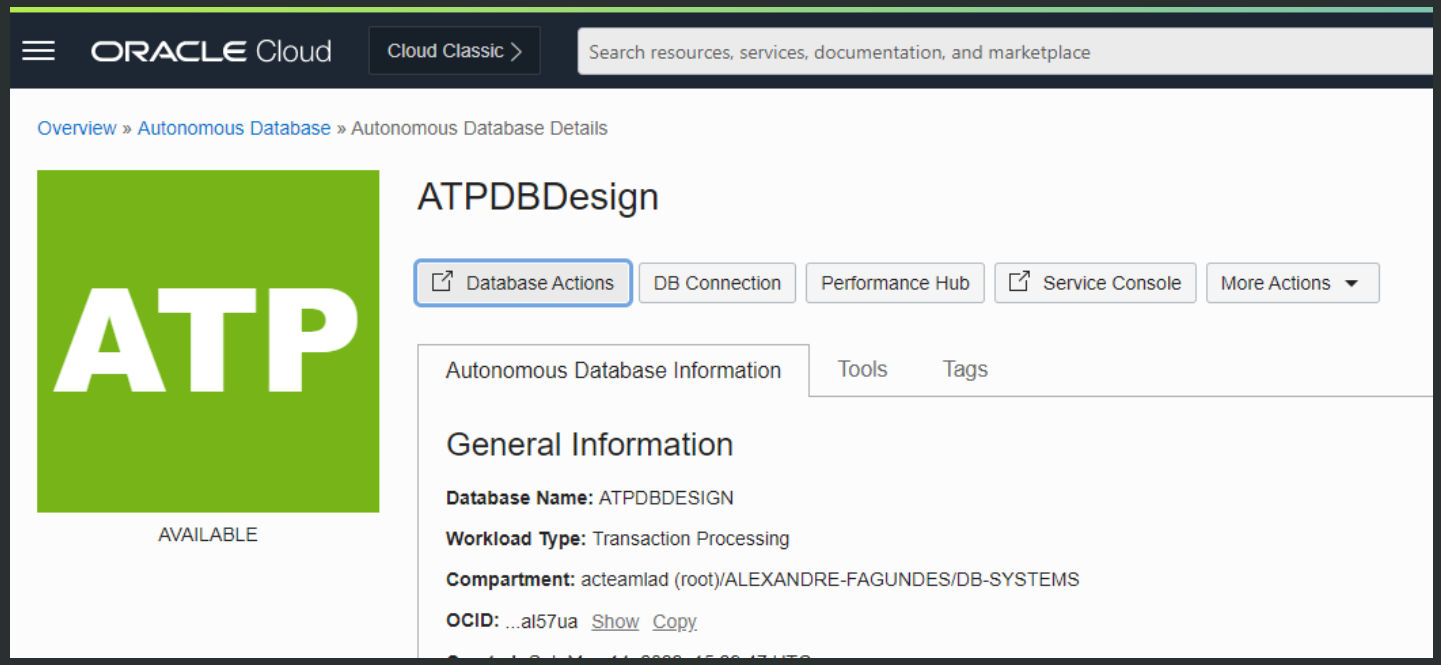
Benchmark 2: 1 + Reduces DML triggers (table creation with DEFAULT ON NULL)

Benchmark 3: 2 + Improve sequences CACHE, drop few indexes

Benchmark 4: 3 + Partitioning



How can we do that?



DEMO

Thank you

Alexandre Fagundes

LAD Partner Enablement