

# RavenDB

# VS

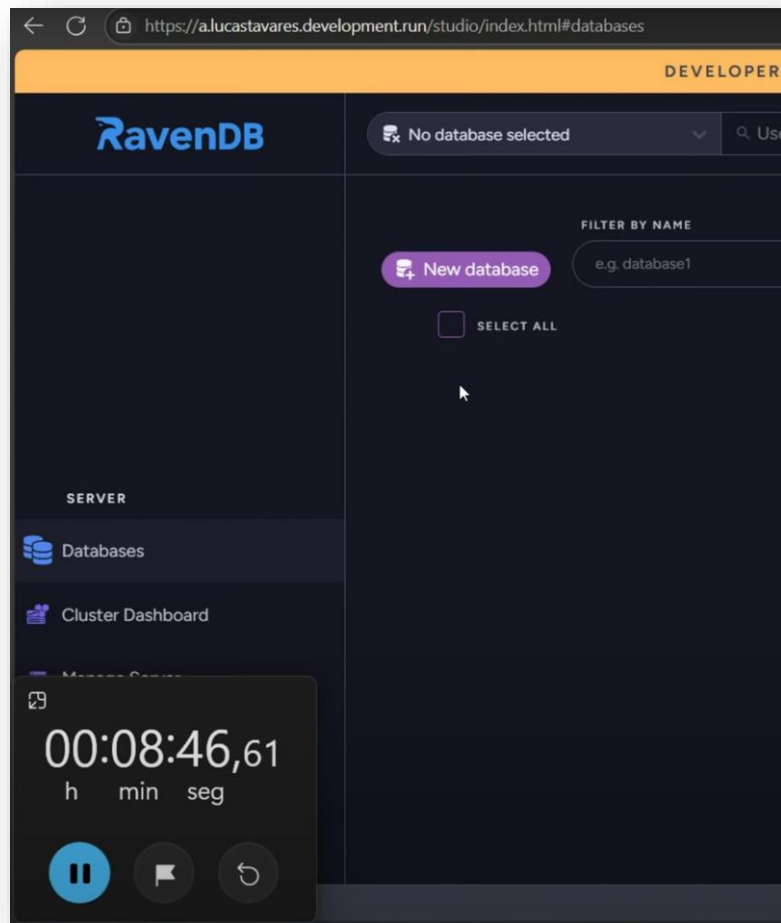


# PostgreSQL

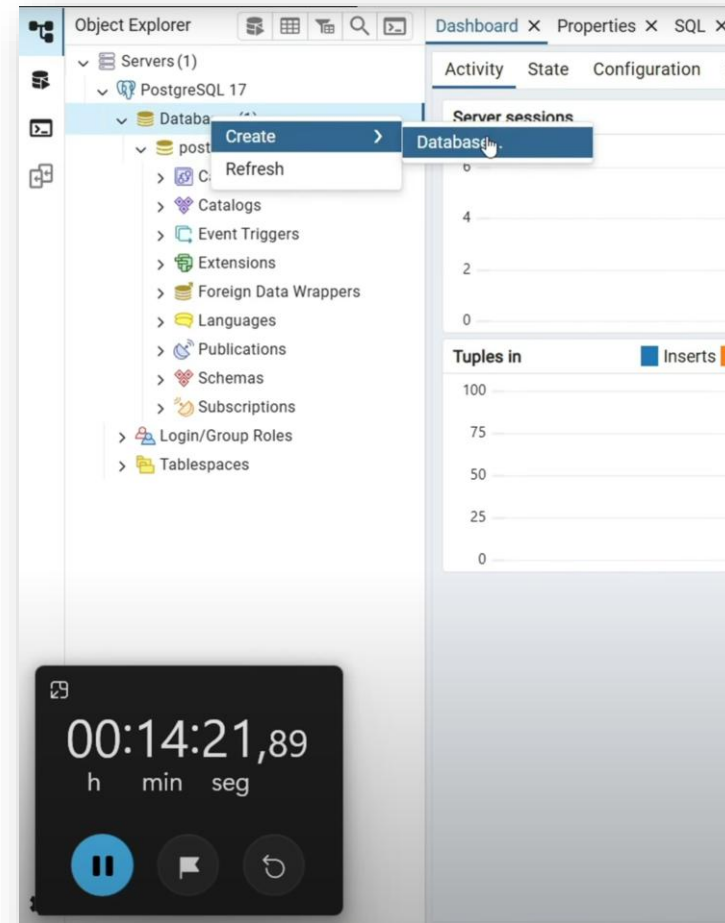
**WHO WINS  
ON SETUP  
EXPERIENCE?**



# Server setup experience

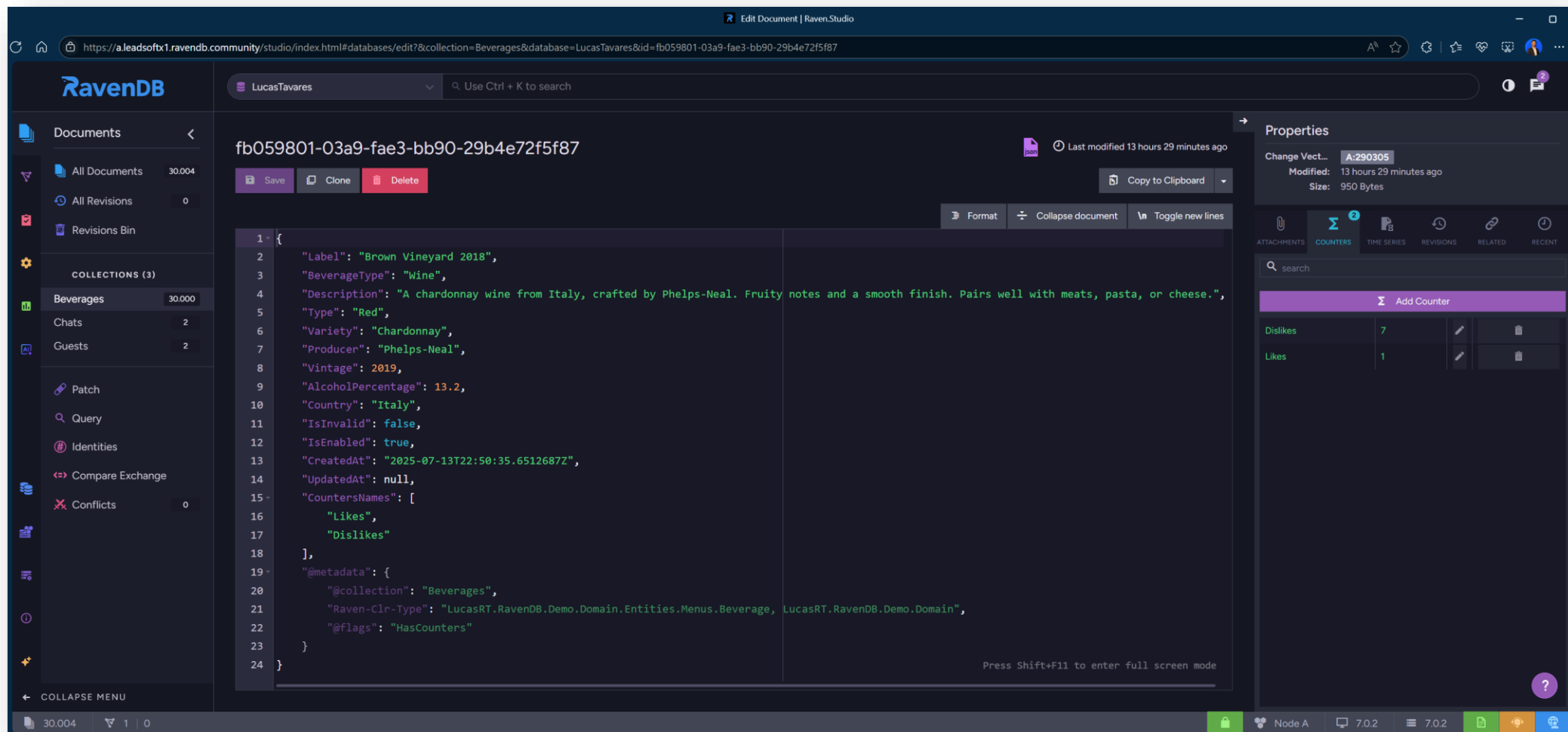


- No installation needed, just download server files and run or configure service (1 click).
- Less than 10 minutes to be ready to create database and connect your application.
- Web interface to manage database, no extra installation needed, even from your mobile browser.



- Long **installation wizard** and many additional options.
- **Over than 10 minutes** to be ready to create database and connect your application.
- **Installed** IDE to view data, thirdy party web or from console. It would be challenging to access from mobile.

# Easy web document navigation



- Very simple and powerful web interface that might save your time
- Inclusive to non technical personel to navigate on data.

# Easy web document navigation

The screenshot displays the RavenDB Studio web interface. On the left is a sidebar with navigation options: Indexes, Query, List of Indexes, Indexing Performance, Map-Reduce Visualizer, Index Cleanup, and Index Errors (0). The main area is titled 'Query' and contains a SQL-like query:

```
1 from "Beverages"
2 where startsWith(Label, "Gold") and Type = "White"
```

Buttons for 'Save...', 'Load Query', and 'Settings' are visible. Below the query editor is a 'Results' section showing the execution details and a table of results. The execution used index 'Auto/Beverages/ByIsEnabledAndIsInvalidAndSearch(BeverageType)AndSearch(Country)AndSearch(Label)AndSearch(Producer)AndSearch(Type)AndSearch(Variety)AndSearch(Vintage)AndVector.search(embedded)' and took 44 ms.

Preview	@id	Label	BeverageType	Description	Type	Variety	Producer	Vintage
	fb059801-f3ac-1e47-...	Golden Cellars 2018	Wine	A chardonnay wine fro...	White	Chardonnay	Miller-Mcgrath	2.015
	fb059801-a7ad-a971-...	Golden Cellars 2022	Wine	A syrah wine from Spa...	White	Syrah	Bullock, Page and Cam...	2.023

At the bottom, there are controls for 'Delete documents', 'Statistics', 'Export to file', 'Display', and 'Expand results'. The footer shows the URL 'https://a.leadsoftx1.ravendb.community/studio/index.html#clusterDashboard' and system information like 'Node A', '7.0.2'.

- Very simple to query with SQL semantic based (LINQ).
- RQL is very Strong to Search and update data in a readable way.



# Dynamic data model for AI

```

1 {
2   "Name": "Lucas T.",
3   "Email": "lucas@leadsoft.inf.br",
4   "Nationality": "",
5   "Other_Relevant_Information": {},
6   "SocialNetworks": [
7     {
8       "Name": "WhatsApp",
9       "Url": "+55 41 991777488"
10    }
11  ],
12  "ValidationResults": [
13    {
14      "MemberNames": [
15        "Nationality"
16      ],
17      "ErrorMessage": "The Nationality field is required."
18    }
19  ],
20  "@metadata": {
21    "@collection": "Guests",
22    "Raven-Clr-Type": "LucasRT.RavenDB.Demo.Domain.Entities.Guests.Guest, LucasRT.RavenDB.Demo.Domain"
23  }
24 }

```

```

1 {
2   "Name": "David Gilmour",
3   "Email": "dg@pinkfloyd.com",
4   "Nationality": "English",
5   "Other_Relevant_Information": {
6     "Drinks": {
7       "Night": "Wine",
8       "Day": "Beer"
9     }
10  },
11  "SocialNetworks": [
12    {
13      "Name": "Instagram",
14      "Url": "@davidgilmourofficial"
15    }
16  ],
17  "@metadata": {
18    "@collection": "Guests",
19    "Raven-Clr-Type": "LucasRT.RavenDB.Demo.Domain.Entities.Guests.Guest, LucasRT.RavenDB.Demo.Domain"
20  }
21 }

```

```

namespace LucasRT.RavenDB.Demo.Domain.Entities.Guests
{
    [Serializable]
    [DataContract]
    9 references
    public partial class Guest : AsValidCollection
    {
        [Required]
        [DataMember]
        2 references
        public string Name { get; set; } = string.Empty;

        [EmailAddress]
        [DataMember]
        2 references
        public string Email { get; set; } = string.Empty;

        [Required]
        [DataMember]
        2 references
        public string Nationality { get; set; } = string.Empty;

        [Required]
        [DataMember]
        2 references
        public dynamic Other_Relevant_Information { get; set; }

        [DataMember]
        2 references
        public IList<SocialNetwork> SocialNetworks { get; set; } = [];
    }
}

```

- If your AI can get information from a chat, it might get more or less information than a form.
- It's possible to include model validation on documents, so you can keep tracked business invalid documents to fix further, without losing data or losing your client.

- All that with a very simple object
- No need to create tables or database Sync with migrations.
- Business rules focused in your code.
- Easier merges, less bugs, faster GoLive.

# Semantics seach (Vector Search Embedding text)

```
Request URL
https://localhost:7198/Menus/Vector?searchtext=gorgonzola%20europe&currentPage=0

Server response

Code    Details
200     Response body
{"Description": "A tempranillo wine from France, crafted by Schmidt-Holland. Fruity notes and a smooth finish. Pairs well with meats, pasta, or cheese",
  },
  {
    "Id": "fb059801-97aa-21d6-6715-8cabafb6a345",
    "Item": "Vaughan Vineyard 2018 | Italy | White | Tempranillo | Schmidt-Holland",
    "BeverageType": "Wine",
    "Description": "A tempranillo wine from Italy, crafted by Schmidt-Holland. Fruity notes and a smooth finish. Pairs well with meats, pasta, or cheese",
  },
  {
    "Id": "fb059801-4ea9-5b35-80a2-167f90d92458",
    "Item": "Wong Cellars 2020 | France | Ros\u00e9 | Tempranillo | Gonzalez LLC",
    "BeverageType": "Wine",
    "Description": "A tempranillo wine from France, crafted by Gonzalez LLC. Fruity notes and a smooth finish. Pairs well with meats, pasta, or cheese."
  },
  }
```

- A Database that understands what you mean and find information for you.
  - Searched text for a Beverages Menu with 30.000 items: "Gorgonzola europe"
  - Query time: 82ms
- None of that literal text information is in Description field, where the vector search occurred in this sample.
- Similar experience searching for "American Burger"

```
Request URL
https://localhost:7198/Menus/Vector?searchtext=american%20burger&currentPage=0

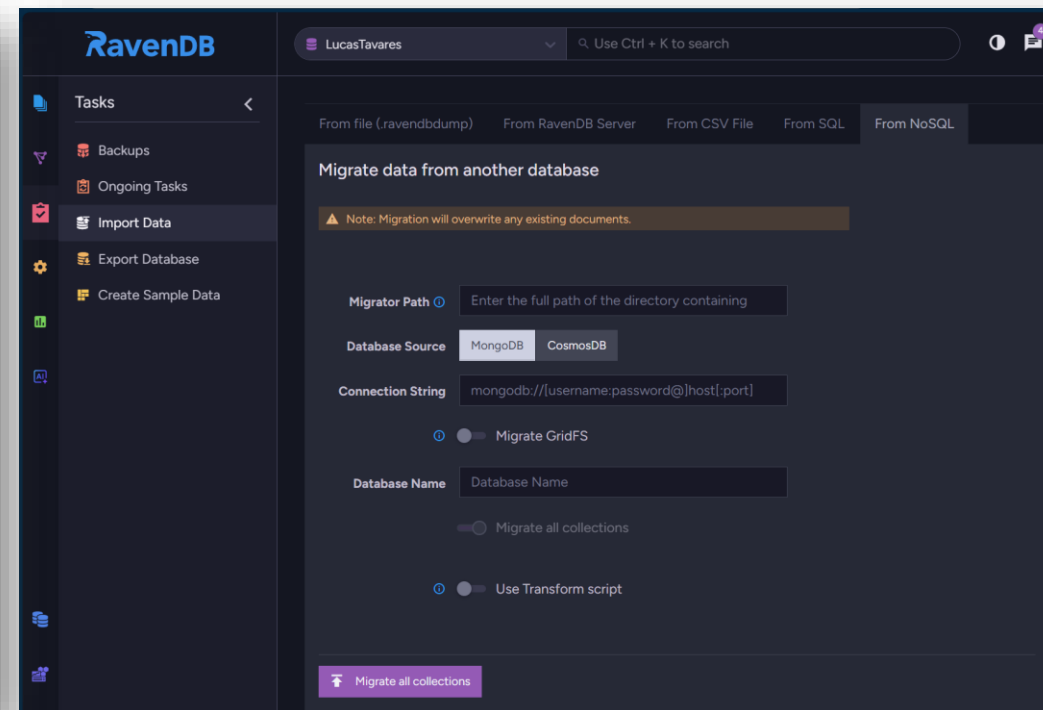
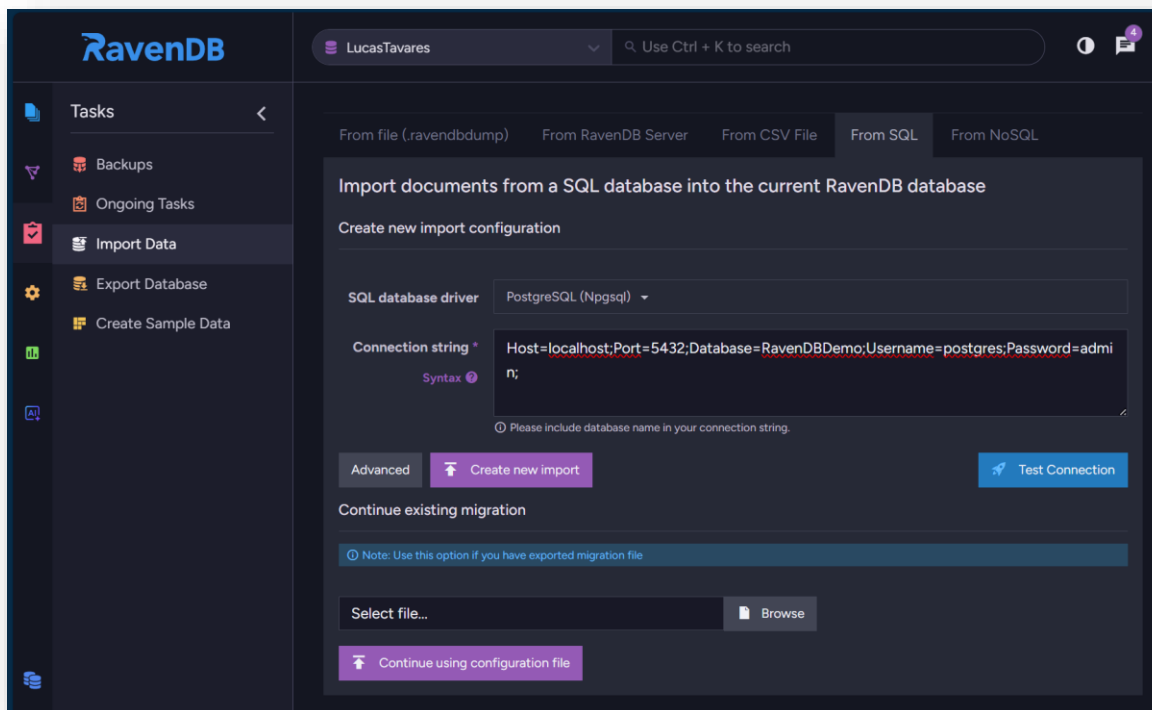
Server response

Code    Details
200     Response body
{"Id": "fb059801-3403-7073-2a03-220072c0d418",
  "Item": "Golden Dog | USA | Lager | | Hammond-Ross",
  "BeverageType": "Beer",
  "Description": "Lager style beer brewed by Hammond-Ross in USA. Refreshing and flavorful with balanced bitterness. Great with snacks or burgers."
},
{
  "Id": "fb059801-76b3-60a9-275c-aae44e4506e4",
  "Item": "Lazy River | USA | Lager | | Rodriguez-Gross",
  "BeverageType": "Beer",
  "Description": "Lager style beer brewed by Rodriguez-Gross in USA. Refreshing and flavorful with balanced bitterness. Great with snacks or burgers."
},
{
  "Id": "fb059801-9ab2-5dde-4aba-0ba4c34d343f",
  "Item": "Frosty River | USA | Lager | | Jackson, Stein and Carney",
  "BeverageType": "Beer",
  "Description": "Lager style beer brewed by Jackson, Stein and Carney in USA. Refreshing and flavorful with balanced bitterness. Great with snacks or burgers."
},
{
  "Id": "fb059801-0bb3-d5af-d17e-6480ac0ad845",
  "Item": "Hoppy Dog | USA | Lager | | Scott-Hudson",
  "BeverageType": "Beer",
  "Description": "Lager style beer brewed by Scott-Hudson in USA. Refreshing and flavorful with balanced bitterness. Great with snacks or burgers."
},
}
```

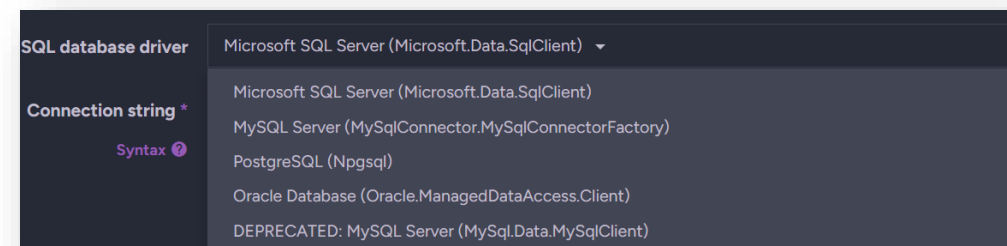
So, imagine if you have your data details available to be understood and queried?

Fantastic.

# SQL or NoSQL data migration tool



- Connect on your current databases and bring data automatically to RavenDB
- Your database schema will be automatically mapped and converted to a Valued Object based on tables relationship. So you won't lose your existing application domain and your current business rules



# Licensing

**Provider**

- aws** Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

**Region**

US East (N. Virginia) - us-east-1

**SUMMARY**

aws

us-east-1

Production Tier

2 vCPU 0.5 GB RAM 10 GB\* STORAGE

**\$38.35 / MONTH**

HOURLY MONTHLY

Estimated price for 3-node cluster

[Copy config link](#) [Start](#)

Interested in subscribing through Aws Marketplace? [See our offer](#)

**Tier**

FREE DEVELOPMENT **PRODUCTION**

**Production**

- 3 nodes
- up to 48 CPU cores and 192 GB of RAM
- up to 4000 GB of disk space
- up to 2h SLA on support
- automatic backups
- Cloud License restrictions

**Instance**

CPU priority

**SHARED** BASIC STANDARD PERFORMANCE

Shared

Instances with limited data and workload amount capabilities, suitable for very small applications.

**DATA TRANSFER COST**

Same region	\$0.01 /GB
Different region	\$0.02 /GB
Internet	\$0.09 /GB
Public IP	\$0.005 /hour

**BACKUP COST \***

1 GB	FREE
After 1 GB	\$1.00 /GB/month

**ADDITIONAL FEATURES COSTS**

PowerBI	15% of hosting cost
...	15%

- On the cloud you wish.
- You don't need a very large machine to get extremely high performance.
- RavenCloud includes
  - Secure by default (SSL)
  - Automated backups by default
  - Customizable AutoScaling
  - 3 Nodes replicance by default
  - Storage auto expand
  - Embeddings for vectors
- Many AI direct plugs
  - OpenAI
  - AzureOpenAI
  - Ollama
  - Google AI
- Everything smooth and easy to configure from Studio or Cloud Panel.
- Dynamic and UI friendly cluster configuration

**RavenDB**

LucasTavares

Use Ctrl + K to search

Manage Server

Cluster

Client Configuration

Studio Configuration

Server Settings

Admin JS Console

Certificates

Server-Wide Tasks

Server-Wide Analyzers

Server-Wide Sorters

Admin Logs

Traffic Watch

Gather Debug Info

Storage Report

IO Stats

Stack Traces

Running Queries

Advanced

Cluster Topology

Cluster Debug

Cluster gate

LEGEND

Node Type: Leader, Member, Promotable

CONNECTION STATUS: Active Connection, Broken Connection

**Connector**

Select connector

- Azure OpenAI Enterprise
- Google AI Enterprise
- Hugging Face Enterprise
- Ollama Enterprise
- OpenAI Enterprise
- Mistral AI Enterprise
- Embedded (bge-micro-v2)



# Resume

What RavenDB can bring to your business?

- Faster environment setup to your team
  - For large teams, that can reflect on a minor budget
- Ease of use: It's very simple to configure, start using and learn new things
  - No migration needed.
  - Fast and optimized by default
- Create applications with less code:
  - Less code = less bugs
  - Less code = minor merges
  - Minor merges = safer and faster GoLive
  - Safer and faster GoLive = Success
- Self contained service.
  - All you need is included by default in RavenDB and you can manage it from your phone, if you wish to.
- AI First!
  - Don't worry about data modeling for dynamic data that comes from AI.
  - You can query your data with semantics, the Database will understand your search, and find your data based on known information.
- Care about the business, not to the Database:
  - RavenDB was design to be "forgotten". It will make all the work do store, optimize and deliver your data while you just need to care of your backend.
  - You won't need a DBA anymore.
- Security and backup
  - RavenCloud will manage all of that for you
  - SSL secured



# Examples

- Check this [Repository on Git Hub](#) get practical examples:
  - [leadsoftlucas/ravendb-demo](#)
- Watch on YouTube dive in the comparison:



[README](#)
[MIT license](#)

# RavenDB

This repository purpose is to highlight three of RavenDB's most impressive and differentiating features rather than other database products.

## Intro

Welcome to this repository that will demonstrate how easy and amazing is to connect and use **RavenDB** into a .net 9.0 c# Rest Web API application and a short comparison with **PostgreSQL**. My goal here, is to bring the ease of use to achieve fantastic results on application features in an effortless code base.

You'll be able to experience how fast we can create an application with high performance and efficiency, both on coding and runtime results.

## Resume

Before diving into the key features I will demonstrate a change of mindset when we discuss data modeling and why it makes all the difference on our database and code style choice.

A great question to ask yourself before building a new application or solution is: "where do I want to spend my time?" . And a very quick response for that, is to focus on the core business, focus on true valued code, so the best approach would be modeling how you'll persist you data, right? Well... sort of! Maybe the business doesn't demand a very clear data model, and how would you define that and keep you code synced with your database? Seems like a mess if you're designing a SQL data model. Before I go deeper, remember one thing. We're in the beginning of the AI age, so I presume you don't want to maintain tables that might fit all possibilities of an AI response, that would be insane. What would be you idea? Please, don't say that you would store tons of json data into a table column. Please, don't. We have a more accurate solution for that.

Let's get back into the main problem: We want to store data, that fit's our business needs language (glossary) and we don't know exactly what kind of fields we might use. Well, we must track the `Resources` of this business and map the relationship between them, at least. ([Domain Driven Design](#), have you ever heard about it?) But for dynamic contents, we might need dynamic fields. So I guess we agree that a multimodel database is a great option here.

But, "how would we select this data?", or even more, how would we search and find data without killing my server cluster or my cloud fundings?.

Wait, you won't need NASA computers for that, you just need a proper database with incredible fast queries, that you won't need to preocupy to setup indexes. Well, have you ever heard about [RavenDB](#)?

Let's fly high with Rook, we'll show you.



# Lucas Tavares

## RavenDB Enthusiast

Location: Curitiba, Brazil

Contact: [lucas@leadsoft.inf.br](mailto:lucas@leadsoft.inf.br)

**Professional Summary:** Experienced technology professional working in the field since 2007, Lucas has held various developer roles in companies spanning sectors from ERP systems to payroll credit banking. A truly RavenDB enthusiast since the first usage.

**Vision:** Lucas believes strongly in agile, flexible software development focused on delivering tangible business value and real-world impact. He is an official partner and enthusiast of **RavenDB** in Brazil.

**Leadership:** A pragmatic leader, assertive communicator, and result-oriented professional, Lucas leads teams using agile and innovative methodologies. His approach fosters collaborative environments that consistently deliver strategic business outcomes.

**Purpose:** Lucas specializes in transforming complex ideas into practical, actionable solutions, driving business growth with intelligent technology.

**Technical Expertise:** Lucas is an expert in high-performance backend development, Web APIs using .NET C#, and databases using RavenDB. He is proficient in implementing Hexagonal Architecture and Domain-Driven Design (DDD), enhancing software scalability, maintainability, and clarity.

Specializing in data modeling DDD (Domain-Driven Design) orientated, he designs solutions with robust data acquisition capabilities, optimal performance, and significant information generation.

### Core Competencies:

- Agile Software Development
- Backend Engineering
- High-Performance Web APIs
- RavenDB Implementation
- Domain-Driven Design (DDD)
- Hexagonal Architecture
- Data Modeling & Performance Optimization
- Team Leadership & Communication

Thank you for your time!

Leadsoft