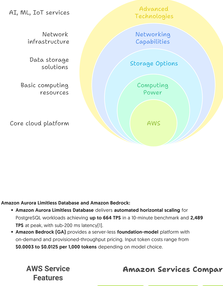
Rohit Singh Chouhan

AWS

AWS Cloud Platform Ecosystem



Amazon Aurora Limitless Database and Amazon Bedrock:

- **Amazon Aurora Limitless Database** delivers **automated horizontal scaling** for PostgreSQL workloads achieving **up to 664 TPS** in a 10-minute benchmark and **2,489 TPS** at peak, with sub-200 ms latency[1].
- **Amazon Bedrock (GA)** provides a server-less **foundation-model** platform with on-demand and provisioned-throughput pricing. Input token costs range from **$0.0003 to $0.0125 per 1,000 tokens** depending on model choice.

AWS Service Features

Amazon Services Comparison

Feature	Aurora Limitless DB	Amazon Bedrock
Scaling	Automated horizontal scaling	Server-less foundation-model platform
Throughput	Up to 664 (2.489 TPS)	N/A
Latency	5ms-200 ms	N/A
Pricing	N/A	$0.0003-$0.0125/1,000 tokens

- **1. Amazon Aurora Limitless Database**

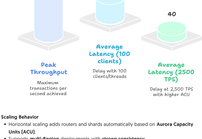
1.1 Architecture and Components

- **DB Shard Group**: A container of **routers** and **shards** presenting a single cluster endpoint.
- **Routers**: Accept SQL connections, parse queries, route to shards, and aggregate results.
- **Shards**: Aurora PostgreSQL instances holding subsets of data for parallel processing.
- **Table Types**:
  - *Shared tables*: Data partitioned by shard key across shards.
  - *Reference tables*: Full copies on each shard for optimized joins.
  - *Standard tables*: Stored on a single shard for simplicity[2][3].

Database Sharding Architecture

1.2 Throughput and Latency

Amazon Aurora Limitless Database Performance Over Time



- **Peak throughput**: 664.2 TPS at 600 s.
- **Latency**: Average ~182 ms (100 clients/100 threads) and ~40 ms at 2,500 TPS with higher ACU settings[1].

Performance Metrics of System

1.3 Scaling Behavior

- Horizontal scaling adds routers and shards automatically based on **Aurora Capacity Units (ACU)**.
- Supports **multi-Region** deployments with **strong consistency**.
- **Use cases**: High-scale OLTP, real-time analytic, SaaS platforms requiring seamless growth.

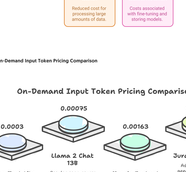
Aurora Database Features

2. Amazon Bedrock (GA)

2.1 Service Overview

- **Fully managed**, serverless platform for **foundation models (FMs)** via unified API.
- Access to models from **Amazon (Titan), Anthropic (Claude), Meta (Llama 2), AI21, Cohere, Stability AI**, and others[4].
- **Key capabilities**: Fine-tuning, Retrieval Augmented Generation (RAG) with Knowledge Bases, agent frameworks to orchestrate multi step tasks.

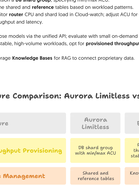
Foundation Model Platform Hierarchy



2.2 Pricing Models

- **On-Demand**: Pay per input/output tokens.
- **Batch**: Lower cost for bulk requests.
- **Provisioned Throughput**: Dedicated capacity (model units) for predictable workloads.
- **Customization**: Charges for fine-tuning (tokens × epochs) and storage.

Pricing Options



2.3 On-Demand Input Token Pricing Comparison

On-Demand Input Token Pricing Comparison

3. Implementation Guidance

1. **Aurora Limitless**
   - Provision a **DB shard group**, specifying min/max ACU.
   - Define sharded and **reference** tables based on workload patterns.
   - Monitor **router** CPU and shard load in Cloud-watch; adjust ACU for desired throughput and latency.
2. **Bedrock**
   - Choose models via the unified API; evaluate with small on-demand calls.
   - For stable, high-volume workloads, opt for **provisioned throughput** to ensure SLA.
   - Leverage **Knowledge Bases** for RAG to connect proprietary data.

Feature Comparison: Aurora Limitless vs. Bedrock

Feature	Aurora Limitless	Bedrock
Throughput Provisioning	DB shard group with min/max ACU	Provisioned throughput for stable workloads
Table Management	Shared and reference tables	Knowledge Bases for RAG
Resource Monitoring	Router CPU and shard load in CloudWatch	Unified API for model evaluation

4. Business Considerations

Cloud Service Comparison