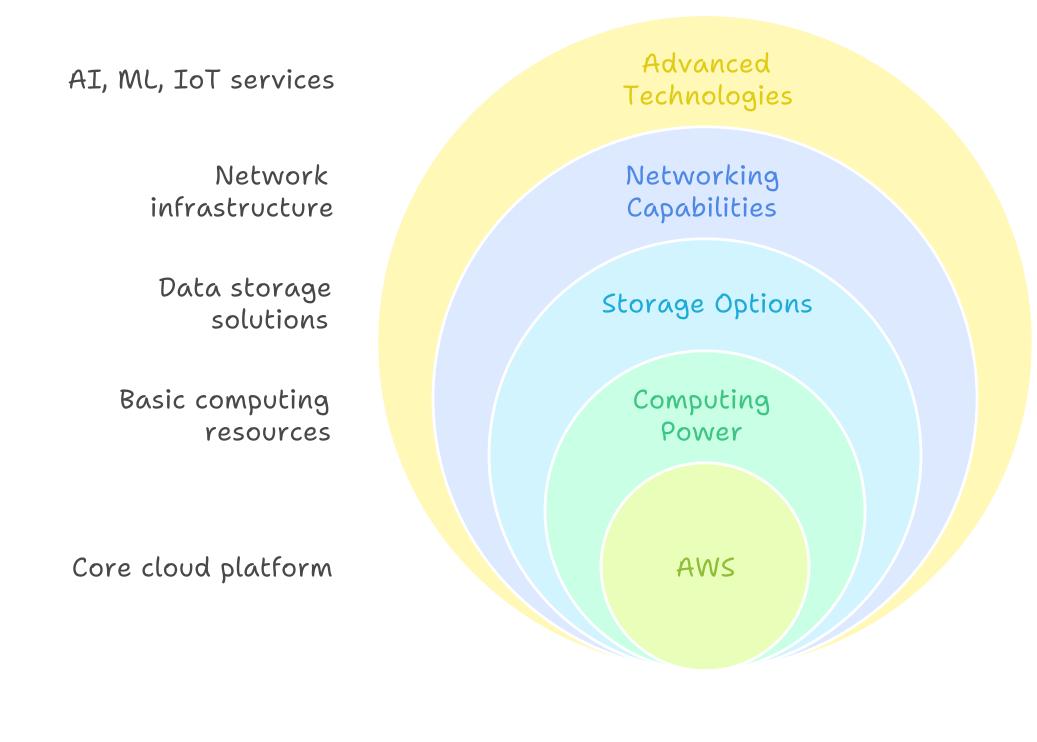
AWS

AWS Cloud Platform Ecosystem



• Amazon Bedrock (GA) provides a server-less foundation-model platform with

TPS at peak, with sub-200 ms latency[1].

Amazon Aurora Limitless Database and Amazon Bedrock:

on-demand and provisioned-throughput pricing. Input token costs range from \$0.0003 to \$0.0125 per 1,000 tokens depending on model choice.

Feature

• Amazon Aurora Limitless Database delivers automated horizontal scaling for

PostgreSQL workloads achieving up to 664 TPS in a 10-minute benchmark and 2,489

AWS Service Amazon Services Comparison **Features**

Limitless DB Bedrock Server-less Automated Scaling foundation-model horizontal scaling platform Up to 664-2,489 Throughput N/A **Aurora Limitless** TPS Amazon Bedrock Database Server-less **Automated horizontal** Latency N/A foundation-model Sub-200 ms scaling for platform with on-PostgreSQL demand and workloads, achieving provisioned-\$0.0003-\$0.0125/1,000 high transactions per Pricing N/A throughput pricing tokens second. options. • 1. Amazon Aurora Limitless Database 1.1 Architecture and Components • DB Shard Group: A container of routers and shards presenting a single cluster endpoint.

Aurora

Amazon

• Shards: Aurora PostgreSQL instances holding subsets of data for parallel processing.

• Table Types:

results.

• 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].

• Routers: Accept SQL connections, parse queries, route to shards, and aggregate

Shards

processing.

Aurora PostgreSQL

instances holding data subsets for parallel

Database Sharding Architecture Shared Tables

He **Routers Standard Tables** Accepts SQL connections and Stored on a single routes queries to shard for simplicity. shards. 1.2 Throughput and Latency

Amazon Aurora Limitless Database

Performance Over Time

511.5 TPS

300 s

360 s

519.9 TPS

40

574.4 TPS

400 s

448.9 TPS

180 s

Data partitioned by

shard key across

shards.

Reference Tables

Full copies on each

shard for optimized

120 s

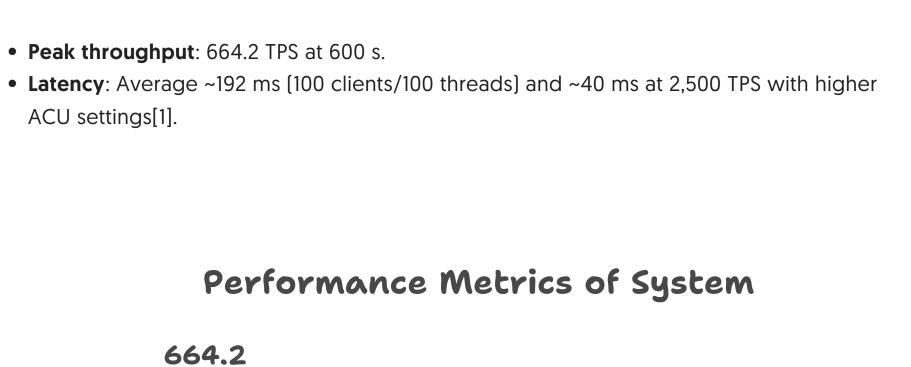
414.7 TPS

60 s

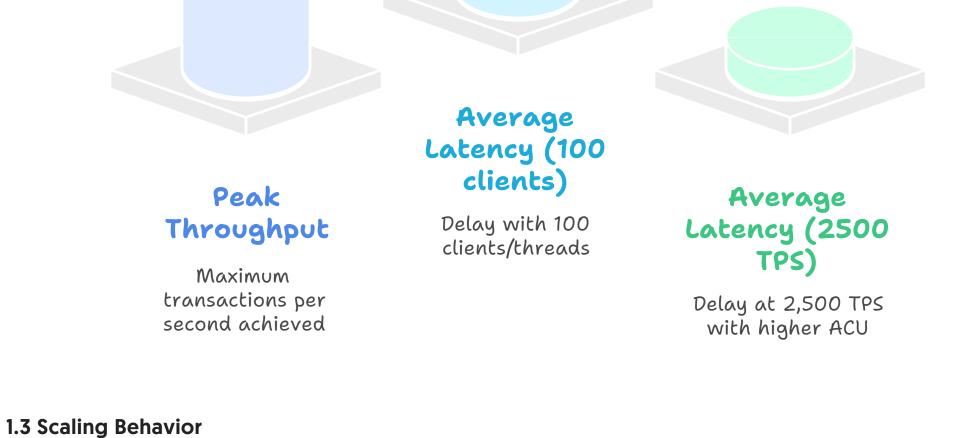
376.9 TPS

240 s

477.2 TPS



192



2.1 Service Overview

2.2 Pricing Models

Units (ACU).

growth.

based on Aurora analytics, and SaaS Capacity Units. platforms.

Horizontal Scaling

Automatically adds

routers and shards

• Horizontal scaling adds routers and shards automatically based on **Aurora Capacity**

• Use cases: High-scale OLTP, real-time analytic, SaaS platforms requiring seamless

Aurora Database Features

Use Cases

Suitable for highscale OLTP, real-time

• Supports multi-Region deployments with strong consistency.

Support Supports deployments across multiple regions with strong consistency. 2. Amazon Bedrock (GA) • Fully managed, serverless platform for foundation models (FMs) via unified API. • Access to models from Amazon (Titan), Anthropic (Claude), Meta (Llama 2), Al21, 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

Multi-Region

Key Capabilities Fine-tuning, RAG, and agent frameworks

Model Access

models

• **Provisioned Throughput**: Dedicated capacity (model units) for predictable workloads.

• Customization: Charges for fine-tuning (tokens × epochs) and storage.

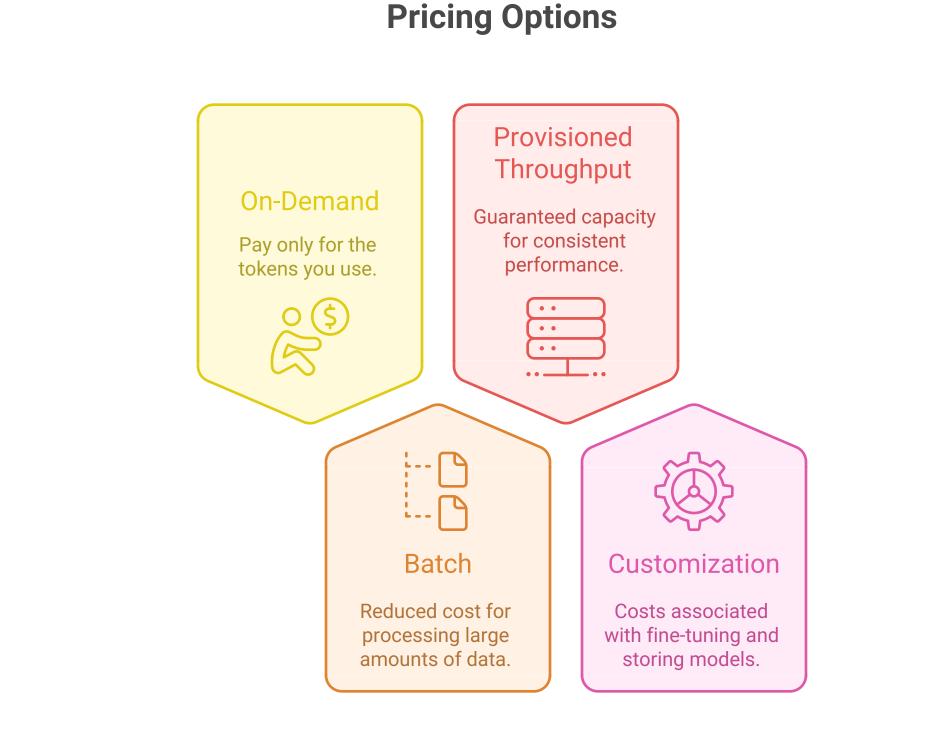
Managed Platform

Serverless platform for foundation

• On-Demand: Pay per input/output tokens.

• Batch: Lower cost for bulk requests.

Access to models from various providers



On-Demand Input Token Pricing Comparison

0.0003

Titan Text Lite

Cost-effective text

processing model

3. Implementation Guidance

1. Aurora Limitless

SLA.

Feature

2. Bedrock

throughput and latency.

2.3 On-Demand Input Token Pricing Comparison

13B Popular open-source Claude Instant chat model Fast and efficient AI assistant

• Choose models via the unified API; evaluate with small on-demand calls.

• Leverage **Knowledge Bases** for RAG to connect proprietary data.

• For stable, high-volume workloads, opt for **provisioned throughput** to ensure

Feature Comparison: Aurora Limitless vs. Bedrock

Aurora

Limitless

Bedrock

Provisioned

0.00075

Llama 2 Chat

• Provision a **DB shard group**, specifying min/max ACU. • Define shared and **reference** tables based on workload patterns. • Monitor router CPU and shard load in Cloud-watch; adjust ACU for desired

0.00163

0.0125

Jurassic-2 Mid

Advanced text generation model

- DB shard group Throughput Provisioning throughput for with min/max ACU stable workloads Shared and
- Knowledge Bases Table Management reference tables for RAG Router CPU and Unified API for Resource Monitoring shard load in model evaluation CloudWatch 4. Business Considerations

