Autviz Solutions

Azure Cost Estimator Chatbot

Conversational Azure Pricing · Real-Time Cost Estimation · Session History · Secure Auth

Streamlit	FastAPI	OpenAI GPT-4
PostgreSQL	Azure Pricing API	Azure VM

#AzureCost · #CloudPricing · #OpenAI · #FastAPI · #Streamlit · #ChatbotAutomation

Overview

What is the Azure Cost Estimator Chatbot?

The Azure Cost Estimator Chatbot is a web-based conversational application that enables users to interactively estimate Azure cloud service costs through natural language.

Built on a three-tier architecture — Streamlit frontend, FastAPI backend, PostgreSQL database — it uses OpenAI GPT-4 to understand intent and calls real-time Azure Pricing APIs for accurate, itemised cost breakdowns.

Conversational Chat UI

Natural language queries via Streamlit

Secure Authentication

Hashed passwords & session tokens

Real-Time Azure Pricing

Live API calls — no cached data

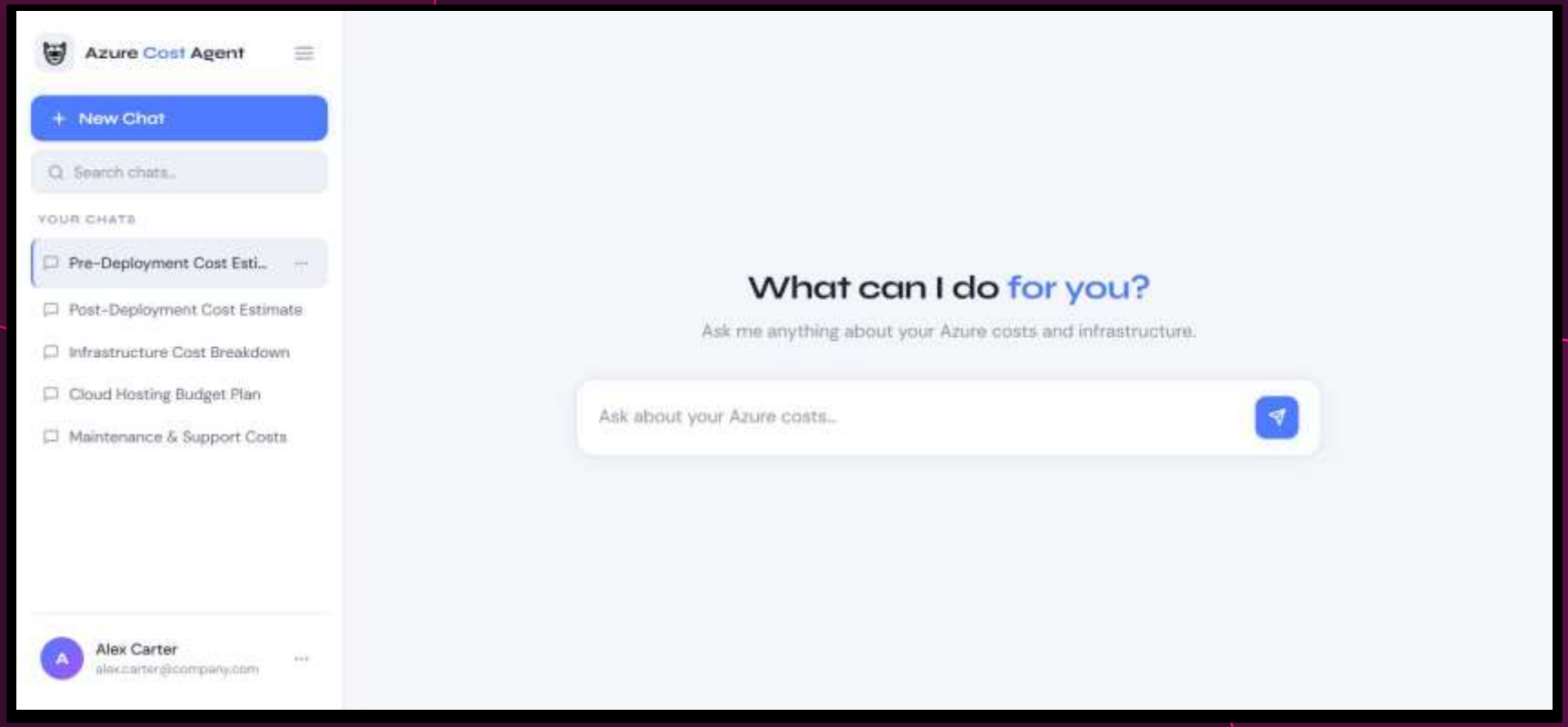
Chat History Persistence

All sessions stored per user in PostgreSQL

< 30s	24×7	100%	∞
Response Time	Always On	Live Pricing	Scalability

The Interface

Azure Cost Agent — Conversational cloud cost estimation in action



Challenges

Pain points that led to this solution

1 Complex Pricing Docs

Engineers spent hours navigating Azure's vast documentation to estimate costs for even simple deployments.

2 Manual Estimation Errors

Spreadsheet estimates were error-prone due to version mismatches, wrong tier assumptions, and stale pricing data.

3 No Conversational Interface

Azure calculators require structured form inputs — a steep barrier for non-technical stakeholders.

4 No Session Continuity

Pricing discussions over email or calls were lost immediately with no structured way to revisit past estimates.

5 Slow Cross-Team Alignment

DevOps, finance, and product teams had no shared tool to estimate and agree on cloud costs in real time.

Objectives

What this chatbot was built to achieve

Provide an intuitive natural language interface for Azure cost estimation — no forms, no documentation navigation required.

Securely authenticate users with hashed password storage, ensuring private and protected session environments.

Persist chat history per user in PostgreSQL, enabling future reference, session continuity, and cost comparisons.

Use OpenAI GPT-4 to understand intent, extract service parameters, and generate structured, readable cost breakdowns.

Call Azure Pricing APIs dynamically as LLM tools to return accurate, real-time pricing — never static or cached.

Deploy on a Linux Azure VM with production-grade configuration and zero downtime — immediately operational for enterprise teams.

Solution & Workflow Logic

Three-tier modular architecture — end to end

1

User Login

Streamlit UI. Passwords hashed in PostgreSQL. Session token issued.

2

Query Submitted

User types a natural language question about Azure service pricing.

3

FastAPI Routing

Backend validates session & routes query to OpenAI processing layer.

4

GPT-4 Processing

Intent extraction — identifies services, region, tier, usage volume.

5

Azure API Tool Call

Real-time Azure Pricing API called with extracted parameters.

6

Response Generated

GPT-4 formats structured, itemised cost reply for the user.

7

History Saved

Full conversation turn saved to PostgreSQL. Reply rendered in chat.

Key Tools & Technologies

The technology stack powering the chatbot

Tool / Service	Purpose
Streamlit	Frontend chat UI, session state, history display
FastAPI	Backend API, authentication, routing & tool orchestration
OpenAI GPT-4	NL understanding, intent extraction, response generation, tool use
Azure Pricing API	Real-time Azure cost data fetched per user query dynamically
PostgreSQL	User accounts, hashed passwords, full chat history persistence
Azure VM (Linux)	Production deployment — Ubuntu, zero-downtime setup
REST APIs	Inter-service communication between Streamlit and FastAPI

Duration & Resources

4 days · 1 Engineer · Full documentation · Zero downtime

4 Days
Total Build Time

1
Automation Engineer

Zero
Downtime Deployment

24×7
Operating Mode

What Was Delivered

- Streamlit frontend — fully configured chat UI with session management and history panel
- OpenAI GPT-4 integration — prompt design, intent extraction & tool call setup
- Deployment configuration — Azure VM (Ubuntu) with zero downtime & env management
- FastAPI backend — authentication, routing & Azure Pricing API integration
- PostgreSQL schema — user accounts, hashed passwords & conversation persistence
- Full handover documentation — architecture, setup guide & API reference

Use Cases

Real-world applications of the Azure Cost Estimator Chatbot

VM Cost Estimation

"How much for 10 D8s v3 VMs in UK South?" — fetches live pricing, returns full itemised breakdown instantly.

Multi-Service Architecture

App Service + Azure SQL + Blob Storage estimated together in a single conversational response.

Region Comparison

"East US vs West Europe for AKS?" — API called for both regions, cost comparison returned automatically.

Budget Planning

Accumulates multiple service estimates across a session and provides a running total for launch planning.

Session History Review

A user returns days later to revisit a prior estimate — full conversation retrieved from PostgreSQL.

Non-Technical Access

A business director asks in plain English — chatbot guides, asks clarifying questions, returns accessible estimate.

Outcomes

Before vs after deployment — measurable impact

99%
Faster Estimation

100%
Live Pricing Data

∞
Query Scalability

Metric	Before	After Deployment	Improvement
Estimation Time	30–120 mins manual	< 30 seconds	**99% faster**
Pricing Accuracy	Error-prone (static docs)	Real-time API data	**100% live pricing**
Accessibility	Technical users only	Any stakeholder	**Fully democratised**
Session Continuity	None — lost after call	Full history in DB	**Permanent record**
Deployment	No prior tooling	Zero downtime on VM	**Production-ready**
Scalability	1 engineer, manual	Unlimited queries	**∞ capacity**

Key Features

What makes this system particularly powerful

Conversational Estimation

Plain English queries. GPT-4 handles all intent parsing — no forms or structured inputs needed.

Real-Time Azure Pricing Tool

Every query triggers a live API call. Pricing is always current — never cached or hardcoded.

Secure Authentication

Passwords hashed with industry-standard cryptography. No plaintext credentials stored.

Persistent Chat History

Every turn saved per user in PostgreSQL. Session continuity and side-by-side comparisons enabled.

Three-Tier Architecture

Streamlit, FastAPI, PostgreSQL — each tier independently replaceable and scalable.

Itemised Cost Breakdowns

Per-service line items, regional pricing, tier rates, and estimated monthly totals in every reply.

Multi-Service Queries

Single message can request pricing for multiple Azure services. GPT-4 decomposes and fetches each.

Zero-Downtime Deployment

Full stack deployable on Azure VM (Ubuntu) with production-grade startup scripts and env management.

Conclusion

The Azure Cost Estimator Chatbot demonstrates the practical power of combining large language models, real-time APIs, and a clean full-stack architecture to solve a genuinely time-consuming problem in cloud infrastructure planning.

By integrating OpenAI GPT-4, Azure Pricing API, FastAPI, Streamlit, and PostgreSQL, any stakeholder — technical or otherwise — can estimate Azure cloud costs through natural conversation, in seconds rather than hours.

Estimate smarter. Deploy faster. Plan with confidence.

Streamlit | FastAPI | OpenAI GPT-4 | PostgreSQL | Azure VM