Microsoft Azure

Build Smarter Apps and Agents with the AI-ready Azure Database for PostgreSQL

01 /

PostgreSQL popularity continues to grow among developers

02 /

Innovate with a fully managed, AI-ready PostgreSQL database

03 /

Bring AI into your PostgreSQL development workflow

04 /

Chart the course from RAG to AI agents

05 /

Boost AI agent intelligence with improved information retrieval

06 /

What you can build today with Azure Database for PostgreSQL

07 /

Start building AI-powered agents and apps today with Azure Database for PostgreSQL

PostgreSQL popularity continues to grow among developers

PostgreSQL has become one of the most popular open-source databases for good reason. Developers prize it for its standards compliance, broad extension ecosystem, and reliable performance across a wide range of use cases from web apps and analytics platforms to finance and manufacturing systems.

But even a well-loved tool like PostgreSQL comes with growing pains. In a recent study, 60% of organizations said their environment had become more complicated over the past two years: Data volumes are ballooning with the rise of AI and real-time apps, Security requirements are tightening, and IT teams are under pressure to move 50% faster than they did just three years ago.

In on-premises setups, those challenges are amplified. Teams must manually patch systems, monitor performance, and provision capacity for peak usage, which leads to overprovisioning and underuse. Routine management tasks eat up time that could be spent on innovation. At the same time, organizations struggle to attract and retain skilled database administrators. Additionally, as more teams look to integrate AI, analytics, and other modern tools, existing infrastructure often can't keep up.

It's no surprise that many companies are rethinking how they manage their PostgreSQL environments. Migrating to the cloud—especially to a fully managed, AI-ready platform—can help ease operational burdens, improve scalability, and free up IT to focus on delivering business value.

Innovate with a fully managed, AI-ready PostgreSQL database

Azure Database for PostgreSQL offers a powerful way forward. It's a fully managed PostgreSQL service that combines the flexibility of open-source tools with the performance, scalability, and AI-readiness of the cloud.

Engineered for enterprise scale and next-generation workloads, the service helps teams stay agile in the face of rising data volumes and expanding use of AI. Instead of maintaining infrastructure, managing patches, or manually tuning performance, developers can focus on building and move faster with tools and features designed for modern apps.

Azure Database for PostgreSQL is built to support today's challenges and tomorrow's innovation.

AI at the core

- pgvector extension enables vector search for Information Retrieval (IR) for agents and RAG apps
- Built-in capabilities that support implementation of advanced IR techniques that improve scale, accuracy and performance
- Azure_AI extension brings the power of LLMs to your data with Azure OpenAI, Azure ML, and Azure Cognitive Services
- The extension also introduces Semantic Operators that go beyond traditional vector search and enable GenAI capabilities directly within PostgreSQL to uncover deeper semantic relationships in your data
- Integrations with LLM and agentic frameworks enable seamless agentic app development

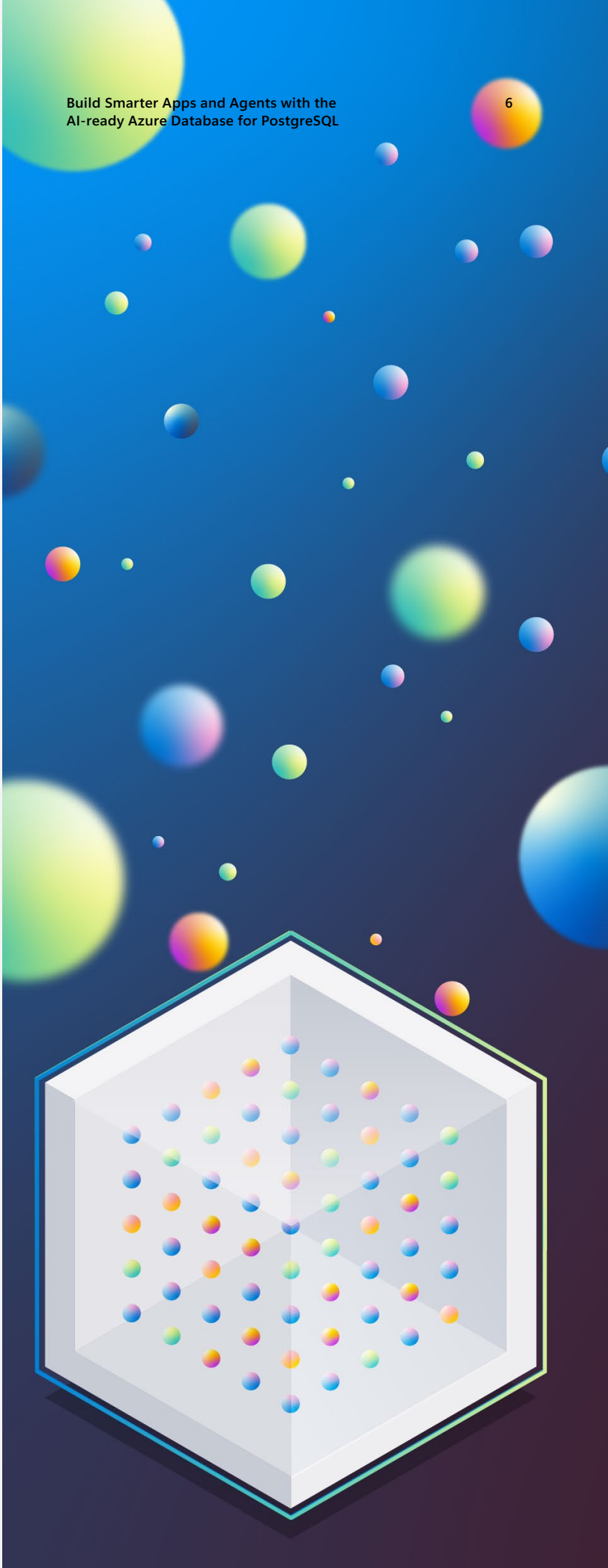
Frictionless developer workflows

- Familiar developer tooling integrations, like the Visual Studio (VS) Code extension with GitHub Copilot capabilities for PostgreSQL, let teams build and optimize PostgreSQL with AI assistance in the development environment
- Hosting on Azure services such as Azure Container Apps and Azure Kubernetes Service along with integrations with monitoring and optimization tools like Azure Monitor and Azure Advisor streamline modern app development
- Built on Community Postgres, the service supports development in any language and all PostgreSQL-compatible frameworks, while also offering support for multiple PostgreSQL extensions to further expand functionality and integration

Resilience out of the box

- Premium SSD v2 with high availability delivers sub-millisecond latency and zero RPO failover to support critical workloads with minimal disruption
- Enterprise-grade security and compliance features include customer-managed key rotation, confidential computing SKUs, and long-term backup deliver confidence in the cloud

Azure Database for PostgreSQL provides a seamless foundation for building intelligent, AI-enhanced applications, without the complexity of managing a full stack. Its elastic scaling, rich extension ecosystem, and native integration with Azure AI services make it ideal for evolving workloads. And because it's built on open-source PostgreSQL, teams can migrate existing workloads without rewriting applications or restructuring data models. That means less risk and faster time to value.



Bring AI into your PostgreSQL development workflow

Developers are under pressure to deliver more intelligent applications, faster. Traditionally, adding AI meant stitching together external tools, managing orchestration layers, or retraining on entirely new platforms. Azure Database for PostgreSQL eliminates that complexity by infusing AI directly into the development process. Developers can streamline their workflows using Visual Studio (VS) Code and GitHub Copilot integrations tailored for Azure Database for PostgreSQL.

With the VS Code extension, developers can connect to PostgreSQL databases, browse tables, run queries, and even visualize schemas without leaving the editor. The extension offers IntelliSense autocompletion for SQL, making query writing faster and reducing errors. And because it integrates directly with Azure Database for PostgreSQL, developers can securely connect to cloud-hosted databases using Entra ID authentication—no connection strings or manual login required.

Integrated GitHub Copilot capabilities further enhances developer productivity by providing AI assistance with PostgreSQL context in the editor. Developers can interact with PostgreSQL databases through the @pgsql GitHub Copilot agent, which brings contextual understanding to schema design, query optimization, and app development. The Copilot agent even supports agent-mode actions like creating databases, modifying schema, or debugging queries based on natural language instructions. This AI assistance reduces the PostgreSQL learning curve and helps ensure developers follow best practices for performance and security.

The VS Code extension and GitHub Copilot support work with any PostgreSQL database, not just Azure-hosted instances, giving developers flexibility to adopt AI-assisted workflows across environments. By integrating these tools, developers can more quickly and confidently build intelligent applications and agents backed by PostgreSQL. The result is a more efficient workflow: less time spent on manual query writing and troubleshooting, and more time delivering features enabled by AI.

Real gains for real-world dev teams

- **Faster iterations.** Schema updates, query tuning, and debugging don't require tool switching or manual SQL just natural language prompts in familiar environments.
- **Reduced ramp up.** New team members can get started quickly with Copilot guidance and intuitive visual tooling.
- **Secure collaboration**. Entra ID based access removes the need for shared passwords or hardcoded credentials.

Chart the course from RAG to AI agents

What is RAG?

Retrieval-Augmented Generation (RAG) is an AI pattern where a large language model (LLM) is supplemented with relevant data fetched from a knowledge base to produce more accurate, context-aware responses. Instead of relying solely on what the model memorized during training, a RAG system retrieves information at query time and feeds it into the model's prompt. PostgreSQL is often used as the backing store in RAG architectures. An app will embed documents or facts into vectors and store them in PostgreSQL using an extension like pgvector, then, at runtime, find which entries are most relevant to the user's question. This grounds the AI's answer in up-to-date, specific data.

A step ahead with AI agents

An AI agent goes one step further. It can generate responses and autonomously plan actions, use tools, and maintain memory to achieve a goal. Agents are powered by LLMs but augmented with frameworks that let them do things like call APIs, query databases, or perform calculations as part of a larger task. Unlike basic RAG, an agent can handle multi-step workflows with branching decisions. It has flexibility and memory adapting its plan based on intermediate results and remembering earlier interactions or data to inform later steps.

For a developer, using agents enables building far more interactive and dynamic AI-driven features compared to static Q&A bots.

Leveraging Azure Database for PostgreSQL for AI Agents

Azure Database for PostgreSQL plays a pivotal role in implementing AI agents. It can serve dual roles:

- **Knowledge store for retrieval:** agents can query embedded vectors stored in PostgreSQL to retrieve relevant facts using pgvector, and use advanced indexing for scalability and enhancements like graphRAG and Semantic Operators to improve accuracy. These facts can then be fed into AI prompts with built-in capabilities from the azure_ai extension bringing the power of LLMs directly into the database.
- **Memory for an agent:** PostgreSQL can store conversation history, user profiles, or intermediate results that the agent can query as it works. Because Azure Database for PostgreSQL is fully managed and scalable, it can handle the high query throughput that complex agent interactions might generate.

Integration with agentic frameworks

Developers don't have to build these capabilities from scratch. There are agentic frameworks like Microsoft's Semantic Kernel, LangChain/LangGraph, LlamaIndex, and others which provide patterns and libraries for creating AI agents. Azure Database for PostgreSQL integrates seamlessly with these frameworks.

- Semantic Kernel offers connectors to store and retrieve embeddings or chat memories from Azure Database for PostgreSQL
- Frameworks like LlamaIndex can use PostgreSQL to index documents and then let an agent query that index during conversations

Support for emerging protocols

Azure is also at the forefront of new standards like the Model Context Protocol (MCP). MCP is an open protocol that allows AI agents to discover and interact with various tools and data sources in a standardized way. Azure MCP Server is an implementation that exposes Azure services, including Azure Database for PostgreSQL, to AI agents through this protocol. In practice, this means an AI agent can use MCP to safely execute a SQL query on the PostgreSQL database when needed, all through a consistent interface. This greatly simplifies connecting agents to enterprise data in a governed manner.

To build advanced AI agents, you need a combination of reliable data retrieval, lasting memory, and tool integration. Azure Database for PostgreSQL integrated with Azure's ecosystem, frameworks and MCP, provides the necessary foundation. Using these together, developers can create AI agents that are grounded in facts, capable of complex reasoning, and integrated with real business processes.

Boost AI agent intelligence with improved information retrieval

AI agents and RAG-based applications are only as good as the information they can retrieve. Azure Database for PostgreSQL offers powerful features to support efficient and accurate information retrieval:

- **Vector search with pgvector:** Azure Database for PostgreSQL supports the pgvector extension, which allows you to store high-dimensional vectors and query them by similarity. With vectors stored directly in PostgreSQL, your application can perform similarity search across structured, unstructured and graph data, powering use cases like 'chat with your data', contextual retrieval for agents, and personalized recommendations without relying on an external vector database. You can also combine vector similarity filtering with traditional SQL conditions in one query. This tight integration of relational data and AI search reduces complexity and latency for RAG pipelines.
- **Azure AI extension for built-in AI services:** Beyond basic vector search, the azure_ai extension enables direct calls to AI services from within the database. You can generate embeddings using Azure OpenAI Service's models right inside a SQL query. You can also invoke Azure Cognitive Services for tasks like sentiment analysis, language detection, or key phrase extraction via simple function calls. This means your PostgreSQL database itself can preprocess and enrich data using AI or even perform re-ranking of results. Additionally, Semantic Operators help you go one step beyond vector search by bringing advanced GenAI functionality directly into PostgreSQL queries for faster development of AI apps.

These capabilities simplify building intelligent applications because much of the heavy AI lifting can happen alongside your data in the database.

Resolve scale challenges with advanced indexing

As your knowledge store grows to millions of entries, scaling query performance is crucial. Traditional indexing methods can struggle with very large vector sets or complex filters. Azure Database for PostgreSQL addresses this with the DiskANN vector index, which is unique to Azure, in addition to the two pgvector supported indexes, IVFFlat and HNSW.

- **DiskANN:** This algorithm that's unique to Azure efficiently indexes vectors by storing them partly on disk and intelligently caching portions in memory. It enables fast approximate nearest-neighbor searches on billions of vectors while drastically reducing memory requirements so you can perform vector searches over massive document collections with low latency and at lower cost, without running into memory bottlenecks. DiskANN has been shown to outperform popular approaches like HNSW in speed, especially as dataset size grows, and can be combined with PostgreSQL's filtering.

Resolve accuracy challenges with better retrieval results

It's not enough to retrieve data quickly; the results must also be relevant and useful. Three features help improve the accuracy of information retrieval for AI applications:

- **Semantic Operators:** These empower developers to harness the capabilities of LLMs directly within PostgreSQL queries—unlocking semantic relationships in operational data and delivering more accurate results without relying on external orchestration layers or services. Available through the azure_ai extension for Azure Database for PostgreSQL, these four operators—is_true, extract, generate, and rank—enable truth evaluation, high-level data extraction, intelligent text generation, and document reranking, respectively. By embedding these capabilities into SQL workflows, developers can uncover deeper insights and elevate the performance of AI-powered applications with minimal operational overhead.

- **Semantic ranking:** Often your initial search might bring back a set of candidate items, but not all are equally relevant. A semantic ranker uses an ML model to re-evaluate and sort those candidates based on how well each item truly matches the query intent. Microsoft provides a semantic ranking solution that can be integrated with Azure Database for PostgreSQL. This significantly improves answer quality for users.
- **Graph-based retrieval (GraphRAG):** Sometimes information is spread across multiple datasets or connected in ways that similarity search alone might not capture. GraphRAG is an advanced technique where you augment your text data with a knowledge graph structure. Using the Apache AGE extension, Azure Database for PostgreSQL can store graph data alongside your tables. An AI agent can then perform graph queries to fetch not just one document, but a network of related information. This helps in complex reasoning and improves accuracy by ensuring the AI considers the structured relationships in your data, not just raw text similarity. Apache AGE allows hybrid SQL + Cypher queries, so an agent could find a relevant starting node via vector search and then follow its connections via a graph query, all within PostgreSQL.

By leveraging these features – vector search, the Azure AI extension, DiskANN for scale, semantic ranking, and graph queries – developers can build retrieval systems that are both fast and intelligent. Your AI agents will be able to fetch precisely the information they need from enormous datasets, and do so in a way that the information is highly relevant and contextual, leading to better outcomes for end users.

What you can build today with Azure Database for PostgreSQL

With AI built into the database, teams can move from experimentation to execution faster. Azure Database for PostgreSQL makes it easy to bring generative AI and advanced search into real-world applications, using tools and languages your team already knows.

Whether you're building an internal assistant, a customer-facing chatbot, or a recommendation engine, you can do it right inside your PostgreSQL database. Here are some common use cases:

Customer support search
Use semantic operators and embeddings to surface the most relevant answers from past tickets and documentation, even when phrasing doesn't match exactly.

Personalized recommendation engines
Combine user behavior, content metadata, and high-dimensional vectors to generate contextual recommendations for content, products, or services.

Document classification and summarization
Automatically categorize, evaluate, or rewrite documents using in-database LLM calls. Ideal for knowledge management or content-heavy apps.

Fraud detection and relationship analysis
Use graph queries and vector search to identify suspicious patterns, uncover hidden connections, and flag anomalous behavior in real time.

UBS

UBS unlocks advanced AI techniques with PostgreSQL on Azure

Challenge
UBS needed a scalable, governed, and efficient infrastructure to support advanced AI techniques, particularly Retrieval Augmented Generation (RAG) and vector search for their financial AI use cases. Their existing setup made it difficult to manage large volumes of unstructured data, enforce data privacy rules, and streamline AI model development across teams.

Solution
Azure Database for PostgreSQL provided UBS with a governed, multi tenant vector store (VEGA), seamless integration with AI services, and support for advanced vector search and hybrid search techniques. This allowed UBS to:

- Build and deploy RAG based solutions efficiently
- Govern vector embeddings and AI data pipelines
- Enable self service AI development on a secure, compliant platform

The result: faster innovation, easier AI adoption, and stronger compliance all within their existing PostgreSQL skill set.

Read the full story.

Start building AI-powered agents and apps today with Azure Database for PostgreSQL

Ready to build more intelligent applications and agents with AI-powered PostgreSQL?

Leverage one of our Solution Accelerators to jump-start development using Azure Database for PostgreSQL.

Agentic shop accelerator

Create a sample multi-agent e-commerce application that uses Azure Database for PostgreSQL for storing product data and vector indexes, illustrating how an AI agent can handle tasks like product recommendation and inventory queries by interacting with the database.

Chat with your data accelerator

Combine Azure OpenAI with PostgreSQL's pgvector to create a Q&A chatbot over your documents.

GraphRAG accelerator

Construct an end-to-end pipeline for converting text data into a knowledge graph in PostgreSQL (with Apache AGE) and using it to answer complex questions.

Bring your own AI copilot accelerator

Build an AI-powered solution based on best practices utilizing RAG-based copilots on the Azure AI platform.

Azure Database for PostgreSQL combines the flexibility of open-source with the performance, scalability, and AI-readiness of Azure. Get started today.