

Comprehensive Technical Guide: Docker-Based Deployment of an LLM-Powered Microsoft Teams Planner System

Report Date: 2025-10-06

1.0 Executive Summary & System Overview

This report provides a comprehensive technical guide for the architecture, containerization, and deployment of an intelligent, single-user management system for Microsoft Teams Planner. The proposed solution leverages a modular, microservices-inspired architecture, orchestrated entirely within Docker, to create a robust and extensible platform. The system is designed to provide a natural language interface for managing Planner tasks directly from Microsoft Teams, integrating multiple Large Language Model (LLM) providers through a central conversational hub. The core objective is to furnish developers with a detailed blueprint for implementation, covering service orchestration, LLM integration, secure API interaction, and performance optimization.

The recommended architecture is composed of several distinct, containerized services that communicate over a private network. The user interacts with a custom bot within Microsoft Teams, which serves as the primary user interface. This bot forwards user prompts to **OpenWebUI**, a feature-rich, self-hosted conversational AI platform that manages conversation history and orchestrates interactions with various LLMs. To enable task management capabilities, OpenWebUI communicates with a custom-built backend tool server. This backend is architected according to the **Model Context Protocol (MCP)**, an emerging open standard for AI tool interaction, ensuring modularity and future interoperability. A lightweight proxy server, **MCPO (MCP-to-OpenAPI)**, translates between OpenWebUI's native OpenAPI tool format and the MCP standard. The custom **Planner MCP Server** encapsulates all business logic, interacting directly with the **Microsoft Graph API** to perform create, read, update, and delete (CRUD) operations on Planner resources.

Deployment is managed via a single **Docker Compose** file, which defines and configures all system components, including the OpenWebUI application, the MCP servers, an optional local LLM runner (**Ollama**), and a **PostgreSQL** database for persistent data storage. This containerized approach ensures a portable, reproducible, and isolated environment, simplifying dependency management and streamlining the development-to-deployment workflow. The guide provides in-depth analysis of each component, detailed Docker Compose configurations, strategies for managing LLM provider rate limits and authentication, and best practices for service orchestration, data persistence, and security. By adhering to the principles and patterns outlined in this report, developers can construct a powerful, scalable, and intuitive conversational AI system for enhancing productivity within the Microsoft Teams ecosystem.

2.0 Core System Architecture and Design Principles

The foundation of the proposed Microsoft Teams Planner management system is a modular, microservices-inspired architecture deployed via Docker containers. This design philosophy is deliberately chosen to promote separation of concerns, independent scalability of components, and technological flexibility, which are significant advantages even in a single-user context. Each service within the architecture has a single, well-defined responsibility, simplifying development, testing, and maintain-

ance. The entire system is orchestrated using Docker Compose, which provides a declarative and reproducible method for defining and running the multi-container application. This approach ensures that the complex interplay between the user interface, the AI orchestration layer, and the backend tool is managed efficiently and reliably.

The end-to-end data and request flow begins with the user in the Microsoft Teams client. A custom bot, developed using the `microsoft/teams-ai` library, captures the user's natural language input. This bot acts as the frontend gateway, handling Teams-specific authentication and user experience elements before forwarding the message to the core of the system. The message is received by the OpenWebUI service, which functions as the central conversational AI hub. OpenWebUI is responsible for maintaining conversation state, managing interactions with the selected LLM, and invoking external tools when a user's request requires an action beyond simple text generation.

When the LLM determines that a Planner-related action is necessary—such as creating a task or listing plans—OpenWebUI triggers its registered Planner tool. Because OpenWebUI's plugin architecture is based on the OpenAPI specification, and the backend tool is built to the Model Context Protocol (MCP) standard for greater interoperability, an intermediary translation layer is required. This is the role of the MCPO (MCP-to-OpenAPI) proxy server. MCPO receives the OpenAPI-formatted tool call from OpenWebUI, translates it into a standardized MCP request, and forwards it to the appropriate backend service.

The final component in the chain is the custom Planner MCP Server. This dedicated backend service contains all the business logic for interacting with Microsoft Planner. It receives the MCP request, interprets the desired action, and constructs the corresponding API call to the Microsoft Graph API. This server manages the entire lifecycle of the interaction with Microsoft's services, including handling OAuth 2.0 authentication, managing API rate limits, and ensuring data consistency through mechanisms like ETags. The response from the Microsoft Graph API travels back through the same chain: the MCP server formulates a standardized MCP response, which the MCPO proxy translates back into an OpenAPI response for OpenWebUI. Finally, OpenWebUI uses this tool output to generate a coherent, natural language reply that is delivered to the user via the Microsoft Teams bot. This decoupled, containerized architecture ensures that each part of the system can be developed and updated independently, creating a resilient and maintainable solution.

3.0 LLM Integration and Provider Management

A central design tenet of this architecture is flexibility in the choice and use of Large Language Models. The system is architected to support multiple LLM providers simultaneously, including cloud-based services like OpenAI and Anthropic, as well as locally-hosted open-source models served via Ollama. This multi-provider strategy allows the user to select the most appropriate model for a given task based on factors such as performance, cost, privacy, or specific capabilities like function calling. OpenWebUI serves as the command center for this multi-model environment, providing a unified interface for managing connections, selecting models on a per-conversation basis, and even engaging multiple models in a single chat for comparative analysis.

Integration with cloud-based LLM providers like OpenAI and Anthropic is achieved through their respective APIs. OpenWebUI supports any OpenAI-compatible API endpoint, which allows for seamless connection to these services. Configuration is managed through the OpenWebUI admin panel or via environment variables in the Docker Compose file. The primary authentication mechanism for these services is the API key, which must be securely stored and provided to OpenWebUI. For a single-user system, managing these keys can be accomplished using a `.env` file alongside the `docker-compose.yml`, which Docker Compose automatically loads. It is critical to consider the operational con-

straints of these APIs, particularly rate limits and cost. Both OpenAI and Anthropic impose strict limits on requests per minute (RPM) and tokens per minute (TPM), which vary by model and user tier. The backend logic, particularly within the Planner MCP Server, must be designed to handle potential `429 Too Many Requests` errors gracefully by implementing an exponential backoff retry strategy. From a cost perspective, API usage is typically billed per token. While a single user's costs are likely to be modest, it is prudent to select cost-effective models (e.g., Anthropic's Haiku or OpenAI's GPT-4o mini) for routine tasks and reserve more powerful, expensive models for complex requests.

For users who prioritize privacy, offline capability, or wish to experiment with open-source models, the system fully supports local LLMs through Ollama. Ollama is a lightweight, user-friendly tool for running models like Llama, Mistral, and Gemma on local hardware. The recommended deployment pattern is to run Ollama as a dedicated service within the Docker Compose stack. The official `ollama/ollama` Docker image simplifies this process immensely. The Docker Compose configuration should define a service for Ollama, map a persistent volume to `/root/.ollama` to store downloaded models, and expose port `11434` to the internal Docker network. For users with compatible hardware, GPU acceleration can be enabled by installing the NVIDIA Container Toolkit and adding the `--gpus=all` flag to the service definition, which dramatically improves inference speed. Once the Ollama service is running, OpenWebUI can be configured to connect to it by setting the `OLLAMA_BASE_URL` environment variable to the Ollama service's network address (e.g., `http://ollama:11434`). This setup provides a powerful, private, and cost-effective alternative to cloud-based LLMs, fully integrated into the same conversational interface.

4.0 OpenWebUI: The Conversational AI Hub

OpenWebUI stands as the central nervous system of the intelligent Planner management system, providing a feature-rich and user-friendly interface for all conversational AI interactions. Its role extends far beyond a simple chat UI; it is a comprehensive platform for model management, conversation orchestration, and tool integration. The recommended deployment method for OpenWebUI is via its official Docker image, which ensures a consistent, isolated, and easily manageable runtime environment. The Docker deployment is configured through a service definition in the main `docker-compose.yml` file. A typical configuration involves running the `ghcr.io/open-webui/open-webui:main` image, mapping an external port (e.g., `3000`) to the container's internal port (`8080`), and mounting a named volume to `/app/backend/data` to ensure the persistence of user data, conversation history, and configuration settings.

Configuration of OpenWebUI is primarily handled through environment variables, which provide a flexible way to tailor the deployment without modifying the core application. For this specific architecture, several environment variables are critical. The `OLLAMA_BASE_URL` variable must be set to point to the internal network address of the Ollama service (e.g., `http://ollama:11434`) to enable access to locally hosted models. If the Ollama instance is running on the host machine rather than in a separate container, the special hostname `host.docker.internal:host-gateway` can be used to allow the container to connect back to the host. For a single-user system, it is often desirable to disable authentication and user sign-up for simplicity. This can be achieved by setting `WEBUI_AUTH=False` and `ENABLE_SIGNUP=False`. It is important to note that some OpenWebUI configurations are persistent; they are read from environment variables on the first run and then stored in an internal `config.json` file. Subsequent changes to these environment variables will not take effect unless persistence is disabled by setting `ENABLE_PERSISTENT_CONFIG=False` or the configuration is updated through the admin UI.

The most crucial aspect of the OpenWebUI configuration for this system is the integration of the custom Planner tool. OpenWebUI's extensibility is powered by its plugin system, which allows the registra-

tion of external tools that the LLM can invoke to perform actions. These tools are defined by an OpenAPI specification, which describes the available functions, their parameters, and what they return. The Planner tool will not be integrated directly but rather through the MCPO proxy. The MCPO server will expose an OpenAPI specification URL that describes the functions available from the Planner MCP Server (e.g., `create_task`, `list_my_tasks`). In the OpenWebUI settings, an administrator will register a new tool by providing this URL. Once registered, the tool becomes available for use in conversations. When a user issues a command like “Create a task to review the Q4 report,” the LLM, leveraging its function-calling capabilities, will identify the `create_task` function from the registered tool’s specification, formulate the correct JSON payload with the extracted parameters, and instruct OpenWebUI to execute the call. OpenWebUI then sends an HTTP request to the specified OpenAPI endpoint (the MCPO proxy), triggering the entire backend workflow.

5.0 The Planner MCP Server: A Standardized Tool for Task Management

The core of the system’s task management functionality resides in the custom Planner MCP Server. This backend service is designed as a dedicated, single-responsibility component that acts as an intelligent wrapper around the Microsoft Graph API for Planner. Its primary purpose is to expose a set of high-level, task-oriented functions that an AI agent can easily understand and invoke. The strategic decision to build this server based on the Model Context Protocol (MCP) is a forward-looking one. MCP is an open, language-agnostic standard designed to standardize how AI agents discover and interact with external tools. By adhering to this protocol, the Planner tool becomes a reusable, interoperable component that is not tightly coupled to OpenWebUI and can be easily integrated with other MCP-compatible AI hosts or orchestrators in the future. The protocol uses a simple, string-based versioning scheme (e.g., `2025-06-18`) to manage backward-incompatible changes, ensuring stable and predictable interactions between clients and servers.

The design of the Planner MCP Server involves defining a set of functions, or “tools,” that map directly to logical user actions within Planner. These tools would include operations such as `planner.create_plan`, `planner.create_bucket`, `planner.create_task`, `planner.get_user_tasks`, `planner.update_task`, and `planner.delete_task`. Each tool definition includes a clear description of its purpose, its required parameters (with types), and what it returns. This metadata is crucial, as it is what the LLM uses to determine which tool to call and how to structure the request. Internally, the implementation of each tool contains the specific logic for interacting with the Microsoft Graph API. This includes constructing the correct HTTP requests, handling the OAuth 2.0 authentication flow to obtain and refresh user access tokens, and processing the API responses. A critical responsibility of this server is to manage the complexities of the Graph API, such as the mandatory use of ETags for optimistic concurrency control. For any update or delete operation, the server must first fetch the resource to get its current ETag, include that ETag in the `If-Match` header of the modification request, and be prepared to handle a `412 Precondition Failed` response by re-fetching and retrying. Furthermore, the server must implement robust error handling and a resilient retry mechanism, specifically an exponential backoff strategy, to gracefully manage the API’s strict throttling policies.

The Planner MCP Server is deployed as a standalone Docker container, defined as a service within the main `docker-compose.yml` file. This containerization pattern ensures that the server’s dependencies (e.g., Python libraries like the Microsoft Graph SDK) are isolated and that the service can be managed independently. The Dockerfile for this service would specify the base Python image, copy the application code, and install the necessary dependencies. The Docker Compose definition would then build and run this image, exposing the server’s port to the internal Docker network so that the MCPO proxy

can communicate with it. This containerized, protocol-driven approach results in a clean, modular, and highly maintainable backend architecture.

6.0 Bridging the Gap: The MCPO Proxy Architecture

A pivotal component in the system's architecture is the MCPO (MCP-to-OpenAPI) proxy server. This lightweight, intermediary service serves a single but critical function: to act as a translation bridge between the world of OpenWebUI's OpenAPI-based tool plugins and the standardized, protocol-driven world of the Planner MCP Server. OpenWebUI is designed to consume tools that are described by an OpenAPI (formerly Swagger) specification, a widely adopted standard for defining RESTful APIs. In contrast, the backend Planner tool is intentionally built as an MCP server to promote long-term reusability and interoperability with a broader ecosystem of AI agents. The MCPO proxy elegantly resolves this incompatibility without requiring modifications to either OpenWebUI or the MCP server itself.

The operational logic of the MCPO proxy is straightforward yet powerful. Upon startup, the MCPO server is configured to connect to one or more downstream MCP servers—in this case, the custom Planner MCP Server. It communicates with the MCP server using the standard MCP initialization and introspection protocol to discover the set of available tools, along with their functions, parameters, and descriptions. Once it has this information, MCPO dynamically generates a fully compliant OpenAPI 3.0 specification in JSON format. This specification accurately represents the MCP server's capabilities as a standard RESTful API. MCPO then exposes this generated specification via a stable HTTP endpoint (e.g., `/openapi.json`).

This generated OpenAPI specification is what gets registered within OpenWebUI. From OpenWebUI's perspective, it is simply interacting with a standard, well-defined API. When the LLM decides to call a Planner function, OpenWebUI constructs an HTTP request according to the OpenAPI specification and sends it to the corresponding endpoint on the MCPO server. The MCPO server receives this incoming HTTP request, translates its payload and path into a valid MCP tool call message, and forwards it to the underlying Planner MCP Server. The MCP server processes the request and returns an MCP response. MCPO then receives this response, translates it back into the appropriate HTTP response format expected by OpenWebUI, and sends it back. This entire process is transparent to both the frontend and the backend. This architectural pattern fully decouples the conversational hub from the tool implementation, allowing developers to build standardized, reusable MCP tools while still leveraging the powerful plugin ecosystem of platforms like OpenWebUI. The MCPO server is deployed as its own Docker container, defined as a service in the Docker Compose file, ensuring it is an isolated and independently manageable part of the overall system.

7.0 Interfacing with Microsoft Planner via the Graph API

The ultimate source of truth and the mechanism for all actions within Microsoft Planner is the Microsoft Graph API. This unified and comprehensive RESTful API serves as the gateway to data and services across the Microsoft 365 ecosystem. For the Planner management system, the Graph API provides a well-structured and powerful set of endpoints for programmatic interaction with all core Planner entities. A deep understanding of its capabilities and constraints is fundamental to building a reliable backend. The API exposes a hierarchical data model where a **plan** is the top-level container for tasks. Crucially, every plan must be associated with a Microsoft 365 Group, which governs its membership and permissions. Within each plan, tasks are organized into **buckets**, which function as columns on a Kanban-style board. The fundamental unit of work is the **task**, which contains properties like title, due date, and assignments. To optimize performance, the API separates core task properties from more de-

tailed information like descriptions, checklists, and attachments, which are stored in a `plannerTaskDetails` object and must be explicitly requested.

Secure access to Planner data is governed by the Microsoft identity platform using the standard OAuth 2.0 protocol. The application must be registered in Microsoft Entra ID, and for a system acting on behalf of a user, it must request and receive consent for **delegated permissions**. The essential permission scope for a fully functional system that can create, modify, and delete tasks is

`Tasks.ReadWrite`. The Planner MCP Server will be responsible for managing the entire OAuth 2.0 Authorization Code Flow, which involves obtaining an authorization code from the user, exchanging it for an access token and a refresh token, and using the access token in the `Authorization` header of every API call. The server must also handle token expiration and use the refresh token to seamlessly obtain new access tokens without requiring the user to sign in again.

Developers must architect the system to accommodate several critical operational constraints of the Graph API. First, the API is subject to strict **rate limits and throttling policies** to ensure service stability. If the application makes too many requests in a short period, the API will respond with an HTTP `429 Too Many Requests` error, including a `Retry-After` header. The Planner MCP Server must implement a resilient exponential backoff strategy that respects this header to handle throttling gracefully. To minimize the risk of being throttled, API calls should be optimized, for instance, by using JSON batching to combine up to 20 individual operations into a single HTTP request. Second, and most significantly, Planner resources are **not supported by the Microsoft Graph change notifications (webhooks) system**. This means the application cannot receive real-time push notifications when data changes in Planner. To maintain a synchronized state, the system must implement a polling mechanism. The most efficient method for this is using **delta queries**, which allow the application to request only the data that has changed since the last poll, thereby reducing network traffic and processing load. This architectural constraint means the system will have an inherent latency in reflecting external changes, determined by the polling interval.

8.0 Comprehensive Deployment with Docker Compose

The entire multi-service application is defined, configured, and orchestrated using a single `docker-compose.yml` file. Docker Compose is the ideal tool for this architecture as it provides a declarative, version-controllable, and portable way to manage the complete application stack. This approach simplifies the development environment setup to a single command (`docker compose up`) and ensures consistency across different machines. The Compose file will define each of the core components as a distinct service, along with the necessary networks for communication and volumes for data persistence.

The following is a representative `docker-compose.yml` configuration for the system, illustrating the relationships and configurations of each service.

```

# docker-compose.yml
version: '3.9'

services:
  # PostgreSQL Database for OpenWebUI
  postgres:
    image: postgres:16
    container_name: planner_postgres
    environment:
      - POSTGRES_USER=openwebui
      - POSTGRES_PASSWORD=openwebui
      - POSTGRES_DB=openwebui
    volumes:
      - postgres_data:/var/lib/postgresql/data
    restart: unless-stopped
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U openwebui -d openwebui"]
      interval: 10s
      timeout: 5s
      retries: 5

  # OpenWebUI Conversational Hub
  open-webui:
    image: ghcr.io/open-webui/open-webui:main
    container_name: planner_openwebui
    ports:
      - "3000:8080"
    environment:
      - 'DATABASE_URL=postgresql://openwebui:openwebui@postgres:5432/openwebui'
      - 'OLLAMA_BASE_URL=http://ollama:11434'
      - 'WEBUI_AUTH=false'
      - 'ENABLE_SIGNUP=false'
    volumes:
      - openwebui_data:/app/backend/data
    depends_on:
      postgres:
        condition: service_healthy
      ollama:
        condition: service_started # Ollama has no native healthcheck
    restart: always

  # Ollama for Local LLM Serving
  ollama:
    image: ollama/ollama
    container_name: planner_ollama
    # Uncomment the following lines to enable GPU acceleration
    # deploy:
    #   resources:
    #     reservations:
    #       devices:
    #         - driver: nvidia
    #           count: all
    #           capabilities: [gpu]
    volumes:
      - ollama_models:/root/.ollama
    restart: unless-stopped

  # Planner MCP Server (Custom Backend)
  planner-mcp-server:
    build:
      context: ./planner-mcp-server # Path to the server's Dockerfile and code
    container_name: planner_mcp_server

```

```

environment:
  - 'MS_CLIENT_ID=${MS_CLIENT_ID}'
  - 'MS_CLIENT_SECRET=${MS_CLIENT_SECRET}'
  - 'MS_TENANT_ID=${MS_TENANT_ID}'
restart: unless-stopped
healthcheck:
  test: ["CMD", "curl", "-f", "http://localhost:8000/health"]
  interval: 15s
  timeout: 5s
  retries: 3

# MCP0 Proxy Server
mcpo-proxy:
  image: ghcr.io/open-webui/mcpo:latest
  container_name: planner_mcpo_proxy
  command: --mcp-server-url http://planner-mcp-server:8000
  ports:
    - "8001:8000" # Expose MCP0's OpenAPI endpoint for registration in OpenWebUI
  depends_on:
    planner-mcp-server:
      condition: service_healthy
  restart: unless-stopped

volumes:
  postgres_data:
  openwebui_data:
  ollama_models:

```

This configuration demonstrates several best practices for service orchestration. The `depends_on` directive is used extensively to control the startup order. For instance, `open-webui` will not start until the `postgres` service is not just running but has passed its health check, ensuring the database is ready to accept connections. The `healthcheck` itself is defined within the `postgres` service, using the `pg_isready` utility to validate its status. This pattern prevents common race conditions during application startup. Services communicate over a default bridge network created by Docker Compose, using their service names (e.g., `postgres`, `ollama`) as hostnames. This provides reliable and isolated inter-service networking. Finally, named volumes (`postgres_data`, `openwebui_data`, `ollama_models`) are used to persist critical data. This ensures that even if the containers are removed and recreated, the application state, conversation history, and downloaded LLM models are preserved.

9.0 System Operations and Best Practices

Effective operation of the deployed system requires adherence to best practices in three key areas: secure management of sensitive credentials, efficient allocation of system resources, and robust strategies for data backup and recovery. These operational considerations are crucial for maintaining the stability, security, and integrity of the application.

The secure management of secrets, such as the Microsoft Graph API client ID and client secret, as well as API keys for cloud-based LLM providers, is paramount. Hardcoding these credentials into the `docker-compose.yml` file or application code is a significant security risk. The recommended approach is to use a `.env` file located in the same directory as the Docker Compose file. Docker Compose automatically reads this file and substitutes the variables into the `environment` section of the service definitions, as shown with `${MS_CLIENT_ID}` in the example. This `.env` file should be included in the project's `.gitignore` file to prevent it from being accidentally committed to version control. For more advanced or production-like environments, Docker Secrets provides a more secure mechanism. Secrets

are managed by the Docker daemon, encrypted at rest, and mounted into containers as in-memory files, reducing the risk of exposure through environment variables or logs.

Resource management is another critical aspect, even for a single-user deployment. Unconstrained Docker containers can consume all available CPU and memory on the host machine, leading to poor performance or system instability. Docker Compose allows for the definition of resource limits for each service. Using the `deploy.resources.limits` and `deploy.resources.reservations` keys, developers can specify hard limits on CPU and memory usage, as well as reservations that guarantee a certain amount of resources to a service. Best practice involves profiling the application's typical resource consumption and setting reasonable limits to prevent runaway processes while ensuring sufficient resources for normal operation. For example, the Ollama service, which can be memory-intensive, should have a memory limit set to prevent it from impacting the performance of other services or the host system.

Finally, a comprehensive data persistence strategy must include procedures for backup and restore. The architecture uses named Docker volumes to persist data for PostgreSQL, OpenWebUI, and Ollama. While volumes ensure data survives container restarts, they do not protect against data corruption, accidental deletion, or host failure. A reliable backup strategy involves periodically creating archives of these volumes. This can be achieved by running a temporary container that mounts the target volume and a local host directory, then using a utility like `tar` to create a compressed archive of the volume's contents. This process can be scripted and scheduled using cron jobs or other automation tools. The restore process is the reverse: a temporary container is used to extract the backup archive into a new or existing volume. It is crucial to stop any containers using the volume before performing a restore operation to ensure data consistency. These backup archives should be stored securely, ideally in a separate physical location or a cloud storage service, to protect against catastrophic data loss.

10.0 Performance, Optimization, and Data Augmentation

To ensure a responsive and effective user experience, several performance optimization strategies should be considered throughout the system's architecture. At the database layer, the PostgreSQL instance backing OpenWebUI can be tuned for better performance. While the default configuration is generally sufficient for a single-user system, performance can be enhanced by adjusting parameters in the `postgresql.conf` file, such as `shared_buffers` and `work_mem`, to better match the host machine's available RAM. Proper indexing, which is managed by the OpenWebUI application itself, is also critical for ensuring that queries for conversation history and user data remain fast as the amount of data grows over time.

A significant opportunity for enhancing the system's intelligence lies in the implementation of a Retrieval-Augmented Generation (RAG) pipeline. RAG allows the LLM to access and incorporate information from an external knowledge base before generating a response, leading to more accurate, context-aware, and up-to-date answers. This is particularly useful for augmenting Planner tasks with relevant project documents, technical specifications, or external web content. The proposed architecture can be extended to support RAG by introducing several new components, orchestrated by a framework like **LangChain**. The core of the RAG pipeline would be a vector database. The recommended **PostgreSQL** database can be easily extended with the **pgvector** extension, allowing it to store and perform efficient similarity searches on high-dimensional vector embeddings of documents. This integrated approach simplifies the architecture by avoiding the need for a separate, dedicated vector database.

The data ingestion process for the RAG pipeline would involve processing various types of documents to create these embeddings. **Unstructured.io** is the recommended tool for this task, as it is specifically designed to parse complex, unstructured files (like PDFs and Word documents) and partition them into clean, semantically meaningful chunks that are ideal for LLM consumption. To acquire external data, such as documentation from a project wiki or relevant articles from the web, a web crawling component is necessary. **Playwright** is the ideal tool for this, as its browser automation capabilities allow it to reliably scrape content from modern, JavaScript-heavy websites. This entire RAG workflow would be managed by the orchestration service, likely built with LangChain, which would handle the steps of document loading, chunking, embedding, storing in pgvector, and then retrieving relevant context to inject into the LLM prompt during a conversation. This powerful extension transforms the system from a simple task manager into a knowledgeable project assistant.

11.0 Conclusion

The architecture and deployment strategy detailed in this report provide a robust and comprehensive blueprint for creating a powerful, LLM-driven management system for Microsoft Teams Planner. By leveraging a modular, containerized design orchestrated with Docker Compose, the system achieves a high degree of flexibility, maintainability, and portability. The strategic integration of best-in-class open-source technologies—including OpenWebUI as a versatile conversational hub, Ollama for private and cost-effective local LLM serving, and the Model Context Protocol for building standardized and reusable AI tools—creates a future-proof platform that is not locked into any single vendor or technology.

The successful implementation of this system hinges on a clear understanding of the capabilities and constraints of its core components. The detailed analysis of the Microsoft Graph API highlights the need for careful management of authentication, rate limiting, and data synchronization through a delta query polling mechanism. The guide's emphasis on operational best practices, including secure secrets management, resource allocation, and data backup strategies, provides the necessary foundation for building a stable and reliable application. Furthermore, the outlined path for future enhancement through the implementation of a Retrieval-Augmented Generation (RAG) pipeline demonstrates the architecture's capacity for growth and increased sophistication. By following the technical guidance, integration patterns, and deployment workflows presented herein, development teams can efficiently construct an intelligent and intuitive tool that streamlines task management and significantly enhances user productivity within the Microsoft Teams environment.

References

- [30 Days of Microsoft Graph - Day 21: Use case: Create plans, buckets, and tasks in Planner - Microsoft 365 Developer Blog](https://devblogs.microsoft.com/microsoft365dev/30daysmsggraph-day-21-use-case-create-plans-buckets-and-tasks-in-planner/) (<https://devblogs.microsoft.com/microsoft365dev/30daysmsggraph-day-21-use-case-create-plans-buckets-and-tasks-in-planner/>)
- [A Comparative Analysis of LLM Application Frameworks for Enterprise AI - ijgisonline.com](https://ijgis.pubpub.org/pub/6yecqicl) (<https://ijgis.pubpub.org/pub/6yecqicl>)
- [A Developer's Friendly Guide to Qdrant Vector Database - cohorte.co](https://www.cohorte.co/blog/a-developers-friendly-guide-to-qdrant-vector-database) (<https://www.cohorte.co/blog/a-developers-friendly-guide-to-qdrant-vector-database>)
- [AI-Powered Web Scraping Solutions 2025 - firecrawl.dev](https://www.firecrawl.dev/blog/ai-powered-web-scraping-solutions-2025) (<https://www.firecrawl.dev/blog/ai-powered-web-scraping-solutions-2025>)
- [Active 'apache-tika' Questions - stackoverflow.com](https://stackoverflow.com/questions/tagged/apache-tika?tab=Active) (<https://stackoverflow.com/questions/tagged/apache-tika?tab=Active>)
- [An Empirical Study of Docker Compose Configuration Patterns - arxiv.org](https://arxiv.org/html/2305.11293v2) (<https://arxiv.org/html/2305.11293v2>)
- [Announcing the GitHub integration with Microsoft Teams - GitHub Blog](https://github.blog/news-in-) (<https://github.blog/news-in->)

sights/product-news/announcing-the-github-integration-with-microsoft-teams/)

[Announcing updates to the Planner API in Microsoft Graph - Microsoft 365 Developer Blog](https://devblogs.microsoft.com/microsoft365dev/announcing-updates-to-the-planner-api-in-microsoft-graph/) (https://devblogs.microsoft.com/microsoft365dev/announcing-updates-to-the-planner-api-in-microsoft-graph/)

[Apache Tika – a content analysis toolkit - tika.apache.org](https://tika.apache.org/) (https://tika.apache.org/)

[Apache Tika, an underrated alternative to Unstructured.io for RAG and fine-tuning - reddit.com](https://www.reddit.com/r/LocalLLaMA/comments/1aq1qkd/apache_tika_an_underrated_alternative_to/) (https://www.reddit.com/r/LocalLLaMA/comments/1aq1qkd/apache_tika_an_underrated_alternative_to/)

[Authentication and authorization basics for Microsoft Graph - Microsoft Graph](https://learn.microsoft.com/en-us/graph/auth/auth-concepts) (https://learn.microsoft.com/en-us/graph/auth/auth-concepts)

[Awesome-GraphRAG - github.com](https://github.com/DEEP-PolyU/Awesome-GraphRAG) (https://github.com/DEEP-PolyU/Awesome-GraphRAG)

[BeautifulSoup4 vs Scrapy Comparison - firecrawl.dev](https://www.firecrawl.dev/blog/beautifulsoup4-vs-scrapy-comparison) (https://www.firecrawl.dev/blog/beautifulsoup4-vs-scrapy-comparison)

[Benchmarks - qdrant.tech](https://qdrant.tech/benchmarks/) (https://qdrant.tech/benchmarks/)

[Best AI Web Scraping Tools - scrapeops.io](https://scrapeops.io/web-scraping-playbook/best-ai-web-scraping-tools/) (https://scrapeops.io/web-scraping-playbook/best-ai-web-scraping-tools/)

[Best Open Source Web Scraping Libraries - firecrawl.dev](https://www.firecrawl.dev/blog/best-open-source-web-scraping-libraries) (https://www.firecrawl.dev/blog/best-open-source-web-scraping-libraries)

[Best python library for generating PDFs? - reddit.com](https://www.reddit.com/r/Python/comments/82z6cw/best_python_library_for_generating_pdfs/) (https://www.reddit.com/r/Python/comments/82z6cw/best_python_library_for_generating_pdfs/)

[Build a basic AI chatbot in Teams - Microsoft Teams](https://learn.microsoft.com/en-us/microsoftteams/platform/toolkit/build-a-basic-ai-chatbot-in-teams) (https://learn.microsoft.com/en-us/microsoftteams/platform/toolkit/build-a-basic-ai-chatbot-in-teams)

[Build a document question-answering system with Docling and Granite - ibm.com](https://www.ibm.com/think/tutorials/build-document-question-answering-system-with-docling-and-granite) (https://www.ibm.com/think/tutorials/build-document-question-answering-system-with-docling-and-granite)

[Build an interactive app with change notifications via webhooks - Microsoft Graph](https://learn.microsoft.com/en-us/graph/patterns/interactive-app-with-change-notifications-via-webhooks) (https://learn.microsoft.com/en-us/graph/patterns/interactive-app-with-change-notifications-via-webhooks)

[Building a basic MCP server with Python - Medium](https://medium.com/data-engineering-with-dremio/building-a-basic-mcp-server-with-python-4c34c41031ed) (https://medium.com/data-engineering-with-dremio/building-a-basic-mcp-server-with-python-4c34c41031ed)

[Building an MCP server as an API developer - Medium](https://heeki.medium.com/building-an-mcp-server-as-an-api-developer-cfc162d06a83) (https://heeki.medium.com/building-an-mcp-server-as-an-api-developer-cfc162d06a83)

[Building custom MCP tool and integrate it to your self-hosted LLM through OpenWebUI \(Part 3\) - Medium](https://juniarto-samsudin.medium.com/building-custom-mcp-tool-and-integrate-it-to-your-self-hosted-llm-through-openwebui-part-3-3268c4fcac6e) (https://juniarto-samsudin.medium.com/building-custom-mcp-tool-and-integrate-it-to-your-self-hosted-llm-through-openwebui-part-3-3268c4fcac6e)

[Building Temporal Knowledge Graphs with Graphiti - falkordb.com](https://www.falkordb.com/blog/building-temporal-knowledge-graphs-graphiti/) (https://www.falkordb.com/blog/building-temporal-knowledge-graphs-graphiti/)

[Case Study: GenAI Tools - tuhinsharma.com](https://tuhinsharma.com/blogs/case-study-genai-tools/) (https://tuhinsharma.com/blogs/case-study-genai-tools/)

[Change notifications for Microsoft Graph resources - Microsoft Graph v1.0](https://learn.microsoft.com/en-us/graph/api/resources/change-notifications-api-overview?view=graph-rest-1.0) (https://learn.microsoft.com/en-us/graph/api/resources/change-notifications-api-overview?view=graph-rest-1.0)

[Choosing the Right Technology Stack for Your Microservices - medium.com](https://medium.com/@jpro-manonet/choosing-the-right-technology-stack-for-your-microservices-a863e34bf755) (https://medium.com/@jpro-manonet/choosing-the-right-technology-stack-for-your-microservices-a863e34bf755)

[Choose the Best Embedding Model for Your Retrieval-Augmented Generation \(RAG\) System - enterprisebot.ai](https://www.enterprisebot.ai/blog/choose-the-best-embedding-model-for-your-retrieval-augmented-generation-rag-system) (https://www.enterprisebot.ai/blog/choose-the-best-embedding-model-for-your-retrieval-augmented-generation-rag-system)

[claude-task-master - GitHub](https://github.com/eyaltoledano/claude-task-master) (https://github.com/eyaltoledano/claude-task-master)

[Combine multiple HTTP requests using JSON batching - Microsoft Graph](https://learn.microsoft.com/en-us/graph/json-batching) (https://learn.microsoft.com/en-us/graph/json-batching)

[compose - github.com](https://github.com/docker/compose) (https://github.com/docker/compose)

[Convert DOC to PPT in Python - blog.aspose.com](https://blog.aspose.com/total/convert-doc-to-ppt-python/) (https://blog.aspose.com/total/convert-doc-to-ppt-python/)

[Convert DOC to PPT in Python - gist.github.com](https://gist.github.com/aspose-com-gists/3d5bf343526692ab0cf44ad9c0487ece) (https://gist.github.com/aspose-com-gists/3d5bf343526692ab0cf44ad9c0487ece)

[Create PDF Documents in Python with ReportLab - pythonassets.com](https://pythonassets.com/posts/) (https://pythonassets.com/posts/)

[create-pdf-documents-in-python-with-reportlab/](#)

[Create selectable PDF files with Lambda, Python, and ReportLab - dev.to](#) (<https://dev.to/aws-builders/create-selectable-pdf-files-with-lambda-python-and-reportlab-5gp0>)

[Creating and Updating PowerPoint Presentations in Python using python-pptx - geeksforgeeks.org](#) (<https://www.geeksforgeeks.org/python/creating-and-updating-powerpoint-presentations-in-python-using-python-pptx/>)

[Creating Various PDF Files via Python - medium.com](#) (<https://akpolatcem.medium.com/creating-various-pdf-files-via-python-eba91a40df9d>)

[DataWalk as a Neo4j Alternative - datawalk.com](#) (<https://datawalk.com/datawalk/neo4jalternative/>)

[Deploy Neo4j with Docker - render.com](#) (<https://render.com/deploy-docker/neo4j>)

[Deployment of Neo4j Docker container with APOC and Graph-Algorithms plugins - medium.com](#) (<https://medium.com/swlh/deployment-of-neo4j-docker-container-with-apoc-and-graph-algorithms-plugins-bf48226928f4>)

[Development - OpenWebUI Docs](#) (<https://docs.openwebui.com/getting-started/advanced-topics/development/>)

[Docling - docling-project.github.io](#) (<https://docling-project.github.io/docling/>)

[Docling - docling.ai](#) (<https://www.docling.ai/>)

[Docling - github.com](#) (<https://github.com/docling-project/docling>)

[Docling: An Efficient Open-Source Toolkit for AI-Driven Document Conversion - research.ibm.com](#) (<https://research.ibm.com/publications/docling-an-efficient-open-source-toolkit-for-ai-driven-document-conversion>)

[Docling: An open-source toolkit for AI-driven document conversion - research.ibm.com](#) (<https://research.ibm.com/blog/docling-generative-AI>)

[Docling's rise: How IBM is making data LLM-ready - ibm.com](#) (<https://www.ibm.com/think/news/doclings-rise-llm-ready-data>)

[Documentation - qdrant.tech](#) (<https://qdrant.tech/documentation/>)

[Docker - neo4j.com](#) (<https://neo4j.com/docs/operations-manual/current/docker/>)

[Docker Compose overview - docs.docker.com](#) (<https://docs.docker.com/compose/>)

[docker-neo4j - github.com](#) (<https://github.com/neo4j/docker-neo4j>)

[Docxtemplater - docxtemplater.com](#) (<https://docxtemplater.com/>)

[Events - OpenWebUI Docs](#) (<https://docs.openwebui.com/features/plugin/events/>)

[Examples - Model Context Protocol](#) (<https://modelcontextprotocol.io/examples>)

[Features - OpenWebUI Docs](#) (<https://docs.openwebui.com/features/>)

[Firecrawl vs. BeautifulSoup - blog.apify.com](#) (<https://blog.apify.com/firecrawl-vs-beautifulsoup/>)

[Flask PDF Generation: ReportLab, WeasyPrint, and PDFKit Compared - codingeasypeasy.com](#) (<https://www.codingeasypeasy.com/blog/flask-pdf-generation-reportlab-weasyprint-and-pdfkit-compared>)

[Free/cheaper alternatives to Neo4j for a personal project? - reddit.com](#) (https://www.reddit.com/r/Data-base/comments/jpmnxp/freecheaper_alternatives_to_neo4j_for_a/)

[From RAG to GraphRAG: A Technical Review - arxiv.org](#) (<https://arxiv.org/abs/2506.05690>)

[Functions - OpenWebUI Docs](#) (<https://docs.openwebui.com/features/plugin/functions/>)

[Generate embeddings with Azure OpenAI - learn.microsoft.com](#) (<https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-generate-embeddings>)

[Generate PDFs in Python with Libraries - templated.io](#) (<https://templated.io/blog/generate-pdfs-in-python-with-libraries/>)

[Generating PDF in Python - glukhov.org](#) (<https://www.glukhov.org/post/2025/05/generating-pdf-in-python/>)

[Get access on behalf of a user - Microsoft Graph](#) (<https://learn.microsoft.com/en-us/graph/auth-v2-user>)

[Get access without a user - Microsoft Graph](#) (<https://learn.microsoft.com/en-us/graph/auth-v2-service>)

[Get change notifications in push mode - Microsoft Graph](#) (<https://learn.microsoft.com/en-us/graph/patterns/notifications-in-push-mode>)

[Get change notifications through Azure Event Hubs - Microsoft Graph](https://learn.microsoft.com/en-us/graph/change-notifications-delivery-event-hubs) (https://learn.microsoft.com/en-us/graph/change-notifications-delivery-event-hubs)
[Getting Started - OpenWebUI Docs](https://docs.openwebui.com/getting-started/) (https://docs.openwebui.com/getting-started/)
[Getting Started with Microsoft Graph API Webhooks: Real-Time Notifications for Your App - Medium](https://medium.com/@mohamedamine.hajji/getting-started-with-microsoft-graph-api-webhooks-real-time-notifications-for-your-app-6b53169373bb) (https://medium.com/@mohamedamine.hajji/getting-started-with-microsoft-graph-api-webhooks-real-time-notifications-for-your-app-6b53169373bb)
[Getting labels from planner with microsoft graph api - Stack Overflow](https://stackoverflow.com/questions/71420902/getting-labels-from-planner-with-microsoft-graph-api) (https://stackoverflow.com/questions/71420902/getting-labels-from-planner-with-microsoft-graph-api)
[GitHub and Microsoft Planner Integration - Make.com](https://www.make.com/en/integrations/github/microsoft-planner) (https://www.make.com/en/integrations/github/microsoft-planner)
[GitHub Integration with Microsoft Teams - GitHub](https://github.com/integrations/microsoft-teams) (https://github.com/integrations/microsoft-teams)
[GitHub Topic: ai-chatbot - GitHub](https://github.com/topics/ai-chatbot) (https://github.com/topics/ai-chatbot)
[GitHub Topic: conversational-ai - GitHub](https://github.com/topics/conversational-ai) (https://github.com/topics/conversational-ai)
[GitHub Topic: microsoft-planner - GitHub](https://github.com/topics/microsoft-planner) (https://github.com/topics/microsoft-planner)
[GitHub Topic: task-management - GitHub](https://github.com/topics/task-management) (https://github.com/topics/task-management)
[GitHub Topic: task-management-app - GitHub](https://github.com/topics/task-management-app) (https://github.com/topics/task-management-app)
[GitHub Topic: task-oriented-dialogue - GitHub](https://github.com/topics/task-oriented-dialogue) (https://github.com/topics/task-oriented-dialogue)
[Graph API - Increase Rate Limiting / Throttling - Microsoft Q&A](https://learn.microsoft.com/en-us/answers/questions/219222/graph-api-increase-rate-limiting-throttling) (https://learn.microsoft.com/en-us/answers/questions/219222/graph-api-increase-rate-limiting-throttling)
[Graph API Integration for SaaS Developers - Microsoft Tech Community](https://techcommunity.microsoft.com/blog/fasttrackforazureblog/graph-api-integration-for-saas-developers/4038603) (https://techcommunity.microsoft.com/blog/fasttrackforazureblog/graph-api-integration-for-saas-developers/4038603)
[Graph RAG - huggingface.co](https://huggingface.co/papers?q=Graph+RAG) (https://huggingface.co/papers?q=Graph+RAG)
[GraphRAG Explained: Enhancing RAG with Knowledge Graphs - medium.com](https://medium.com/@zilliz_learn/graphrag-explained-enhancing-rag-with-knowledge-graphs-3312065f99e1) (https://medium.com/@zilliz_learn/graphrag-explained-enhancing-rag-with-knowledge-graphs-3312065f99e1)
[Graphiti: Building Temporal Knowledge Graphs for Agentic Memory - neo4j.com](https://neo4j.com/blog/developer/graphiti-knowledge-graph-memory/) (https://neo4j.com/blog/developer/graphiti-knowledge-graph-memory/)
[graphiti - github.com](https://github.com/getzep/graphiti) (https://github.com/getzep/graphiti)
[How can I automate the generation of a PowerPoint with Python? - stackoverflow.com](https://stackoverflow.com/questions/71359430/how-can-i-automate-the-generation-of-a-powerpoint-with-python) (https://stackoverflow.com/questions/71359430/how-can-i-automate-the-generation-of-a-powerpoint-with-python)
[How does Haystack differ from other search frameworks like LangChain and LlamaIndex? - milvus.io](https://milvus.io/ai-quick-reference/how-does-haystack-differ-from-other-search-frameworks-like-langchain-and-llamaindex) (https://milvus.io/ai-quick-reference/how-does-haystack-differ-from-other-search-frameworks-like-langchain-and-llamaindex)
[How to avoid Microsoft Graph API throttling and optimize network traffic - DEV Community](https://dev.to/this-is-learning/how-to-avoid-microsoft-graph-api-throttling-and-optimize-network-traffic-5c2g) (https://dev.to/this-is-learning/how-to-avoid-microsoft-graph-api-throttling-and-optimize-network-traffic-5c2g)
[How to build an AI Chatbot with Redis, Python, and GPT - freeCodeCamp](https://www.freecodecamp.org/news/how-to-build-an-ai-chatbot-with-redis-python-and-gpt/) (https://www.freecodecamp.org/news/how-to-build-an-ai-chatbot-with-redis-python-and-gpt/)
[How to Run Multiple Containers with Docker Compose - freecodecamp.org](https://www.freecodecamp.org/news/run-multiple-containers-with-docker-compose/) (https://www.freecodecamp.org/news/run-multiple-containers-with-docker-compose/)
[How to use Tools with Open WebUI - Medium](https://medium.com/write-a-catalyst/how-to-use-tools-with-open-webui-9725db2724bb) (https://medium.com/write-a-catalyst/how-to-use-tools-with-open-webui-9725db2724bb)
[IBM Automation Document Processing - ibm.com](https://www.ibm.com/products/document-processing) (https://www.ibm.com/products/document-processing)
[IBM Granite-Docling: End-to-end document conversion for AI - ibm.com](https://www.ibm.com/new/announcements/granite-docling-end-to-end-document-conversion) (https://www.ibm.com/new/announcements/granite-docling-end-to-end-document-conversion)
[ibm-granite/granite-docling-258M - huggingface.co](https://huggingface.co/ibm-granite/granite-docling-258M) (https://huggingface.co/ibm-granite/granite-docling-258M)
[Integrate GitHub with Microsoft Planner - Appy Pie Automate](https://www.appypieautomate.ai/integrate/apps/github/integrations/microsoft-planner) (https://www.appypieautomate.ai/integrate/apps/github/integrations/microsoft-planner)
[Introduction - neo4j.com](https://neo4j.com/docs/operations-manual/current/docker/introduction/) (https://neo4j.com/docs/operations-manual/current/docker/introduction/)
[Introducing the Azure MCP Server - Azure SDK Blog](https://devblogs.microsoft.com/azure-sdk/introducing-the-azure-mcp-server/) (https://devblogs.microsoft.com/azure-sdk/introducing-the-azure-mcp-server/)

[Is it possible to get real time update for updates via microsoft graph webhooks? - Stack Overflow](https://stackoverflow.com/questions/62297678/is-it-possible-to-get-real-time-update-for-updates-via-microsoft-graph-webhooks) (<https://stackoverflow.com/questions/62297678/is-it-possible-to-get-real-time-update-for-updates-via-microsoft-graph-webhooks>)

[LangChain, LlamaIndex, or Haystack: Which Framework Suits Your LLM Needs? - medium.com](https://dkaarthick.medium.com/langchain-llamaindex-or-haystack-which-framework-suits-your-llm-needs-7408fee8ab1e) (<https://dkaarthick.medium.com/langchain-llamaindex-or-haystack-which-framework-suits-your-llm-needs-7408fee8ab1e>)

[Live-Chatbot-for-Final-Year-Project - GitHub](https://github.com/Vatshayan/Live-Chatbot-for-Final-Year-Project) (<https://github.com/Vatshayan/Live-Chatbot-for-Final-Year-Project>)

[LlamaIndex vs LangChain vs Haystack - medium.com](https://medium.com/@heyamit10/llamaindex-vs-langchain-vs-haystack-4fa8b15138fd) (<https://medium.com/@heyamit10/llamaindex-vs-langchain-vs-haystack-4fa8b15138fd>)

[LlamaIndex vs LangChain vs Haystack: Choosing the right one for your LLM application - linkedin.com](https://www.linkedin.com/pulse/llamaindex-vs-langchain-haystack-choosing-right-one-subramaniam-yvere) (<https://www.linkedin.com/pulse/llamaindex-vs-langchain-haystack-choosing-right-one-subramaniam-yvere>)

[LlamaIndex vs LangChain: Which is better? - blog.n8n.io](https://blog.n8n.io/llamaindex-vs-langchain/) (<https://blog.n8n.io/llamaindex-vs-langchain/>)

[MS Graph API Planner - Query all task details for a plan - Stack Overflow](https://stackoverflow.com/questions/71084675/ms-graph-api-planner-query-all-task-details-for-a-plan) (<https://stackoverflow.com/questions/71084675/ms-graph-api-planner-query-all-task-details-for-a-plan>)

[Mastering RAG: How to Select an Embedding Model - galileo.ai](https://galileo.ai/blog/mastering-rag-how-to-select-an-embedding-model) (<https://galileo.ai/blog/mastering-rag-how-to-select-an-embedding-model>)

[MCP integration into OpenWebUI - Reddit](https://www.reddit.com/r/OpenWebUI/comments/1jaidh4/mcp_integration_into_openwebui/) (https://www.reddit.com/r/OpenWebUI/comments/1jaidh4/mcp_integration_into_openwebui/)

[MCP Pipe - OpenWebUI Community](https://openwebui.com/f/haervwe/mcp_pipe) (https://openwebui.com/f/haervwe/mcp_pipe)

[MCP server step-by-step guide to building from scratch - Composio DevBlog](https://composio.dev/blog/mcp-server-step-by-step-guide-to-building-from-scrтч) (<https://composio.dev/blog/mcp-server-step-by-step-guide-to-building-from-scrтч>)

[MCPO: Supercharge Open WebUI with MCP Tools - Medium](https://mychen76.medium.com/mcpo-supercharge-open-webui-with-mcp-tools-4ee55024c371) (<https://mychen76.medium.com/mcpo-supercharge-open-webui-with-mcp-tools-4ee55024c371>)

[Microservice Architecture Tech Stack Essentials - springfuse.com](https://www.springfuse.com/microservice-architecture-tech-stack-essentials/) (<https://www.springfuse.com/microservice-architecture-tech-stack-essentials/>)

[Microservices architecture style - learn.microsoft.com](https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices) (<https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>)

[Microservices Technology Stack Delivers Content as a Service - contentstack.com](https://www.contentstack.com/cms-guides/microservices-technology-stack-delivers-content-as-a-service) (<https://www.contentstack.com/cms-guides/microservices-technology-stack-delivers-content-as-a-service>)

[Microsoft Graph API - Oauth 2.0 scopes - Microsoft Q&A](https://learn.microsoft.com/en-us/answers/questions/28515/microsoft-graph-api-oauth-2-0-scopes) (<https://learn.microsoft.com/en-us/answers/questions/28515/microsoft-graph-api-oauth-2-0-scopes>)

[Microsoft Graph API daily limitation - Stack Overflow](https://stackoverflow.com/questions/57107987/microsoft-graph-api-daily-limitation) (<https://stackoverflow.com/questions/57107987/microsoft-graph-api-daily-limitation>)

[Microsoft Graph API limits for Microsoft 365 Copilot connectors - Microsoft Graph](https://learn.microsoft.com/en-us/graph/connecting-external-content-api-limits) (<https://learn.microsoft.com/en-us/graph/connecting-external-content-api-limits>)

[Microsoft Graph API limits? - Reddit](https://www.reddit.com/r/Intune/comments/1jp31de/microsoft_graph_api_limits/) (https://www.reddit.com/r/Intune/comments/1jp31de/microsoft_graph_api_limits/)

[Microsoft Graph API Overview - Microsoft Learn](https://learn.microsoft.com/en-us/graph/overview) (<https://learn.microsoft.com/en-us/graph/overview>)

[Microsoft Graph Dev Center - Microsoft Developer](https://developer.microsoft.com/en-us/graph) (<https://developer.microsoft.com/en-us/graph>)

[Microsoft Graph Documentation - GitHub](https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/concepts/planner-concept-overview.md) (<https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/concepts/planner-concept-overview.md>)

[Microsoft Graph GitHub Repositories - GitHub](https://github.com/microsoftgraph) (<https://github.com/microsoftgraph>)

[Microsoft Graph REST API - Microsoft Developer](https://developer.microsoft.com/en-us/graph/rest-api) (<https://developer.microsoft.com/en-us/graph/rest-api>)

[Microsoft Graph Webhooks & Delta Query: Like Peanut Butter & Jelly - Voitanos](https://www.voitanos.io/blog/microsoft-graph-webhook-delta-query/) (<https://www.voitanos.io/blog/microsoft-graph-webhook-delta-query/>)

[Microsoft Graph change notifications API - Microsoft Q&A](https://learn.microsoft.com/en-us/answers/tags/304/microsoft-graph-change-notifications-api) (<https://learn.microsoft.com/en-us/answers/tags/304/microsoft-graph-change-notifications-api>)

[Microsoft Graph permissions reference - Microsoft Graph](https://learn.microsoft.com/en-us/graph/) (<https://learn.microsoft.com/en-us/graph/>)

permissions-reference)

[Microsoft Graph service-specific throttling limits - Microsoft Graph](https://learn.microsoft.com/en-us/graph/throttling-limits) (https://learn.microsoft.com/en-us/graph/throttling-limits)

[Microsoft Graph throttling guidance - Microsoft Graph](https://learn.microsoft.com/en-us/graph/throttling) (https://learn.microsoft.com/en-us/graph/throttling)

[Microsoft Planner Premium REST API - Microsoft Q&A](https://learn.microsoft.com/en-us/answers/questions/2119208/microsoft-planner-premium-rest-api) (https://learn.microsoft.com/en-us/answers/questions/2119208/microsoft-planner-premium-rest-api)

[Microsoft Planner Tag on Microsoft 365 Developer Blog - Microsoft](https://devblogs.microsoft.com/microsoft365dev/tag/microsoft-planner/) (https://devblogs.microsoft.com/microsoft365dev/tag/microsoft-planner/)

[Microsoft identity platform and OAuth 2.0 On-Behalf-Of flow - Microsoft Entra](https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-on-behalf-of-flow) (https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-on-behalf-of-flow)

[Microsoft identity platform and OAuth 2.0 authorization code flow - Microsoft Entra](https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-auth-code-flow) (https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-auth-code-flow)

[Microsoft identity platform and OAuth 2.0 client credentials flow - Microsoft Entra](https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-client-creds-grant-flow) (https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-client-creds-grant-flow)

[Microsoft identity platform and OAuth 2.0 implicit grant flow - Microsoft Entra](https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-implicit-grant-flow) (https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth2-implicit-grant-flow)

[MinIO and Tika for Text Extraction - blog.min.io](https://blog.min.io/minio-tika-text-extraction/) (https://blog.min.io/minio-tika-text-extraction/)

[Model Context Protocol - Anthropic](https://www.anthropic.com/news/model-context-protocol) (https://www.anthropic.com/news/model-context-protocol)

[Model Context Protocol - A Complete Tutorial - Medium](https://medium.com/@nimritakoul01/the-model-context-protocol-mcp-a-complete-tutorial-a3abe8a7f4ef) (https://medium.com/@nimritakoul01/the-model-context-protocol-mcp-a-complete-tutorial-a3abe8a7f4ef)

[Model Context Protocol - OpenCV Blog](https://opencv.org/blog/model-context-protocol/) (https://opencv.org/blog/model-context-protocol/)

[Model Context Protocol - seangoedecke.com](https://www.seangoedecke.com/model-context-protocol/) (https://www.seangoedecke.com/model-context-protocol/)

[Model Context Protocol \(MCP\) Tutorial - DataCamp](https://www.datacamp.com/tutorial/mcp-model-context-protocol) (https://www.datacamp.com/tutorial/mcp-model-context-protocol)

[Model Context Protocol \(MCP\) Tutorial - DigitalOcean](https://www.digitalocean.com/community/tutorials/model-context-protocol) (https://www.digitalocean.com/community/tutorials/model-context-protocol)

[Model Context Protocol \(MCP\) Tutorial: Build Your First MCP Server in 6 Steps - Towards Data Science](https://towardsdatascience.com/model-context-protocol-mcp-tutorial-build-your-first-mcp-server-in-6-steps/) (https://towardsdatascience.com/model-context-protocol-mcp-tutorial-build-your-first-mcp-server-in-6-steps/)

[Model Context Protocol GitHub Organization - GitHub](https://github.com/modelcontextprotocol) (https://github.com/modelcontextprotocol)

[Model Context Protocol Specification - modelcontextprotocol.io](https://modelcontextprotocol.io/specification/2025-06-18) (https://modelcontextprotocol.io/specification/2025-06-18)

[Model Context Protocol/servers - GitHub](https://github.com/modelcontextprotocol/servers) (https://github.com/modelcontextprotocol/servers)

[Multi-container applications with Docker Compose - learn.microsoft.com](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/multi-container-applications-docker-compose) (https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/multi-container-applications-docker-compose)

[MySQL vs. PostgreSQL: What's the Difference? - aws.amazon.com](https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/) (https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/)

[MySQL vs. PostgreSQL: A 360-Degree Comparison of Syntax, Performance, Scalability, and Features - enterprisedb.com](https://www.enterprisedb.com/blog/postgresql-vs-mysql-360-degree-comparison-syntax-performance-scalability-and-features?lang=en) (https://www.enterprisedb.com/blog/postgresql-vs-mysql-360-degree-comparison-syntax-performance-scalability-and-features?lang=en)

[MySQL vs. PostgreSQL: A Comprehensive Comparison - portable.io](https://portable.io/learn/psql-2-mysql) (https://portable.io/learn/psql-2-mysql)

[MySQL vs Postgres in 2024 - dbconvert.com](https://dbconvert.com/blog/mysql-vs-postgres-in-2024/) (https://dbconvert.com/blog/mysql-vs-postgres-in-2024/)

[Neo4j Alternative: Open-Source, Distributed, and Lightning-Fast - nebula-graph.io](https://www.nebula-graph.io/posts/Neo4j_Alternative_Open-Source_Distributed_and_Lightning_Fast) (https://www.nebula-graph.io/posts/Neo4j_Alternative_Open-Source_Distributed_and_Lightning_Fast)

[Neo4j Alternative: What Are My Open-Source DB Options? - memgraph.com](https://memgraph.com/blog/neo4j-alternative-what-are-my-open-source-db-options) (https://memgraph.com/blog/neo4j-alternative-what-are-my-open-source-db-options)

[Neo4j Alternatives & Competitors - gartner.com](https://www.gartner.com/reviews/market/cloud-data-) (https://www.gartner.com/reviews/market/cloud-data-)

base-management-systems/vendor/neo4j/product/neo4j-graphdatabase/alternatives)

[Neo4j Alternatives - puppygraph.com](https://www.puppygraph.com/blog/neo4j-alternatives) (https://www.puppygraph.com/blog/neo4j-alternatives)

[Neo4j Deployment Center - neo4j.com](https://neo4j.com/deployment-center/) (https://neo4j.com/deployment-center/)

[Neo4j Documentation - neo4j.com](https://neo4j.com/docs/) (https://neo4j.com/docs/)

[Neo4j Graph Database Alternatives - softwarereviews.com](https://www.softwarereviews.com/categories/119/products/4481/alternatives) (https://www.softwarereviews.com/categories/119/products/4481/alternatives)

[Neo4j Graph Database Competitors and Alternatives - g2.com](https://www.g2.com/products/neo4j-graph-database/competitors/alternatives) (https://www.g2.com/products/neo4j-graph-database/competitors/alternatives)

[Neo4j vs. Memgraph - memgraph.com](https://memgraph.com/blog/neo4j-vs-memgraph) (https://memgraph.com/blog/neo4j-vs-memgraph)

[neo4j - hub.docker.com](https://hub.docker.com/_/neo4j) (https://hub.docker.com/_/neo4j)

[Obsidian BMO Chatbot - AIBase](https://www.aibase.com/repos/project/obsidian-bmo-chatbot) (https://www.aibase.com/repos/project/obsidian-bmo-chatbot)

[Open WebUI Tool Creator - OpenWebUI Community](https://openwebui.com/m/mhwhgm/open-web-ui-tool-creator) (https://openwebui.com/m/mhwhgm/open-web-ui-tool-creator)

[OpenAPI Servers for Open WebUI - GitHub Discussions](https://github.com/open-webui/openapi-servers/discussions/58) (https://github.com/open-webui/openapi-servers/discussions/58)

[OpenAPI Servers: MCP - OpenWebUI Docs](https://docs.openwebui.com/openapi-servers/mcp/) (https://docs.openwebui.com/openapi-servers/mcp/)

[OpenAPI Servers: Open WebUI - OpenWebUI Docs](https://docs.openwebui.com/openapi-servers/open-webui/) (https://docs.openwebui.com/openapi-servers/open-webui/)

[open-webui/mcpo - GitHub](https://github.com/open-webui/mcpo) (https://github.com/open-webui/mcpo)

[open-webui/open-webui - GitHub](https://github.com/open-webui/open-webui) (https://github.com/open-webui/open-webui)

[open-webui/open-webui/discussions/3134 - GitHub](https://github.com/open-webui/open-webui/discussions/3134) (https://github.com/open-webui/open-webui/discussions/3134)

[open-webui/open-webui/discussions/7363 - GitHub](https://github.com/open-webui/open-webui/discussions/7363) (https://github.com/open-webui/open-webui/discussions/7363)

[Optimize Open WebUI: Three Practical Extensions for a Better User Experience - Medium](https://medium.com/pythoneers/optimize-open-webui-three-practical-extensions-for-a-better-user-experience-cbe365af60b1) (https://medium.com/pythoneers/optimize-open-webui-three-practical-extensions-for-a-better-user-experience-cbe365af60b1)

[Overview - qdrant.tech](https://qdrant.tech/documentation/overview/) (https://qdrant.tech/documentation/overview/)

[Overview of change notifications in Microsoft Graph - Microsoft Graph](https://learn.microsoft.com/en-us/graph/change-notifications-overview) (https://learn.microsoft.com/en-us/graph/change-notifications-overview)

[Overview of the Azure MCP Server - Microsoft Learn](https://learn.microsoft.com/en-us/azure/developer/azure-mcp-server/overview) (https://learn.microsoft.com/en-us/azure/developer/azure-mcp-server/overview)

[PM-bot.ai - AI Project Management Consultant](https://pm-bot.ai/) (https://pm-bot.ai/)

[PMOtto.ai - AI Project Management Assistant](https://www.pmotto.ai/) (https://www.pmotto.ai/)

[Pattern: Microservice Architecture - microservices.io](https://microservices.io/patterns/microservices.html) (https://microservices.io/patterns/microservices.html)

[Pattern: Multiple technology stacks per service - microservices.io](https://microservices.io/articles/dark-energy-dark-matter/dark-energy/multiple-technology-stacks.html) (https://microservices.io/articles/dark-energy-dark-matter/dark-energy/multiple-technology-stacks.html)

[Permissions and consent in the Microsoft identity platform - Microsoft Entra](https://learn.microsoft.com/en-us/entra/identity-platform/scopes-oidc) (https://learn.microsoft.com/en-us/entra/identity-platform/scopes-oidc)

[Pgvector vs Qdrant: Which is the Best Vector Database? - tigerdata.com](https://www.tigerdata.com/blog/pgvector-vs-qdrant) (https://www.tigerdata.com/blog/pgvector-vs-qdrant)

[pgvector - github.com](https://github.com/pgvector/pgvector) (https://github.com/pgvector/pgvector)

[pgvector 0.5.0 Released - postgresql.org](https://www.postgresql.org/about/news/pgvector-050-released-2700/) (https://www.postgresql.org/about/news/pgvector-050-released-2700/)

[pgvector 0.7.0 Released - postgresql.org](https://www.postgresql.org/about/news/pgvector-070-released-2852/) (https://www.postgresql.org/about/news/pgvector-070-released-2852/)

[pgvector Tutorial: Storing and Querying Vector Embeddings in PostgreSQL - datacamp.com](https://www.datacamp.com/tutorial/pgvector-tutorial) (https://www.datacamp.com/tutorial/pgvector-tutorial)

[pgvector: Embeddings and vector similarity - supabase.com](https://supabase.com/docs/guides/data-) (https://supabase.com/docs/guides/data-

base/extensions/pgvector)

[Planner - Microsoft Graph Toolkit](https://learn.microsoft.com/en-us/graph/toolkit/components/planner) (https://learn.microsoft.com/en-us/graph/toolkit/components/planner)

[planner resource type - Microsoft Graph beta](https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/api-reference/beta/resources/planner-overview.md) (https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/api-reference/beta/resources/planner-overview.md)

[planner resource type - Microsoft Graph beta](https://learn.microsoft.com/en-us/graph/api/resources/planner-overview?view=graph-rest-beta) (https://learn.microsoft.com/en-us/graph/api/resources/planner-overview?view=graph-rest-beta)

[planner resource type - Microsoft Graph v1.0 - GitHub](https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/api-reference/v1.0/resources/planner-overview.md) (https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/api-reference/v1.0/resources/planner-overview.md)

[planner resource type - Microsoft Graph v1.0](https://learn.microsoft.com/en-us/graph/api/resources/planner-overview?view=graph-rest-1.0) (https://learn.microsoft.com/en-us/graph/api/resources/planner-overview?view=graph-rest-1.0)

[Plugin - OpenWebUI Docs](https://docs.openwebui.com/features/plugin/) (https://docs.openwebui.com/features/plugin/)

[Postgres vs MySQL - cyberpanel.net](https://cyberpanel.net/blog/postgres-vs-mysql) (https://cyberpanel.net/blog/postgres-vs-mysql)

[PostgreSQL as a Vector Database: Using pgvector - tigerdata.com](https://www.tigerdata.com/blog/postgresql-as-a-vector-database-using-pgvector) (https://www.tigerdata.com/blog/postgresql-as-a-vector-database-using-pgvector)

[PostgreSQL Extensions: pgvector - tigerdata.com](https://www.tigerdata.com/learn/postgresql-extensions-pgvector) (https://www.tigerdata.com/learn/postgresql-extensions-pgvector)

[PostgreSQL Vector Search Guide with pgvector - northflank.com](https://northflank.com/blog/postgresql-vector-search-guide-with-pgvector) (https://northflank.com/blog/postgresql-vector-search-guide-with-pgvector)

[PostgreSQL vs MySQL comparison 2024 2025 - bytebase.com](https://www.bytebase.com/blog/postgres-vs-mysql/) (https://www.bytebase.com/blog/postgres-vs-mysql/)

[PostgreSQL vs. MySQL - flatirons.com](https://flatirons.com/blog/postgresql-vs-mysql/) (https://flatirons.com/blog/postgresql-vs-mysql/)

[PostgreSQL vs. MySQL: Which One Is Better For Your Use Case? - integrate.io](https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/) (https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/)

[Python PDF Generation from HTML with WeasyPrint - dev.to](https://dev.to/bowmanjd/python-pdf-generation-from-html-with-weasyprint-538h) (https://dev.to/bowmanjd/python-pdf-generation-from-html-with-weasyprint-538h)

[Python Web Scraping Projects - firecrawl.dev](https://www.firecrawl.dev/blog/python-web-scraping-projects) (https://www.firecrawl.dev/blog/python-web-scraping-projects)

[python-pptx - github.com](https://github.com/scanny/python-pptx) (https://github.com/scanny/python-pptx)

[python-pptx - products.documentprocessing.com](https://products.documentprocessing.com/editor/python/python-pptx/) (https://products.documentprocessing.com/editor/python/python-pptx/)

[python-pptx - pypi.org](https://pypi.org/project/python-pptx/) (https://pypi.org/project/python-pptx/)

[python-pptx - python-pptx.readthedocs.io](https://python-pptx.readthedocs.io/en/latest/index.html) (https://python-pptx.readthedocs.io/en/latest/index.html)

[Qdrant - aws.amazon.com](https://aws.amazon.com/marketplace/pp/prodview-rtp4b42tydtzg) (https://aws.amazon.com/marketplace/pp/prodview-rtp4b42tydtzg)

[Qdrant - Vector Database for AI - qdrant.tech](https://qdrant.tech/) (https://qdrant.tech/)

[Qdrant Vector Database - qdrant.tech](https://qdrant.tech/qdrant-vector-database/) (https://qdrant.tech/qdrant-vector-database/)

[qdrant - github.com](https://github.com/qdrant/qdrant) (https://github.com/qdrant/qdrant)

[RAG and embeddings: is it better to embed text with labels or not? - community.openai.com](https://community.openai.com/t/rag-and-embeddings-is-it-better-to-embed-text-with-labels-or-not/604100) (https://community.openai.com/t/rag-and-embeddings-is-it-better-to-embed-text-with-labels-or-not/604100)

[RAG vs. GraphRAG: A Comprehensive Comparison for Question Answering - arxiv.org](https://arxiv.org/abs/2502.11371) (https://arxiv.org/abs/2502.11371)

[RAG vs. GraphRAG: A Comprehensive Comparison for Question Answering - arxiv.org](https://arxiv.org/html/2502.11371v1) (https://arxiv.org/html/2502.11371v1)

[Report on all Planner data in a tenant using PowerShell - Practical 365](https://practical365.com/report-planner-data/) (https://practical365.com/report-planner-data/)

[Resources for Advanced Document Processing - cwiki.apache.org](https://cwiki.apache.org/confluence/display/TIKA/Resources+for+Advanced+Document+Processing) (https://cwiki.apache.org/confluence/display/TIKA/Resources+for+Advanced+Document+Processing)

[Retrieval Augmented Generation \(RAG\) - docs.mistral.ai](https://docs.mistral.ai/guides/rag/) (https://docs.mistral.ai/guides/rag/)

[Retrieval-Augmented Generation \(RAG\) - promptingguide.ai](https://www.promptingguide.ai/research/rag) (https://www.promptingguide.ai/research/rag)

[Retrieving attachments from Microsoft Planner tasks through Graph API - Stack Overflow](https://stack-) (https://stack-

[overflow.com/questions/47940869/retrieving-attachments-from-microsoft-planner-tasks-through-graph-api](https://stackoverflow.com/questions/47940869/retrieving-attachments-from-microsoft-planner-tasks-through-graph-api))

[Run Neo4j in a Docker container - development.neo4j.dev](https://development.neo4j.dev/developer/docker-run-neo4j/) (<https://development.neo4j.dev/developer/docker-run-neo4j/>)

[Running multi-container applications - docs.docker.com](https://docs.docker.com/get-started/docker-concepts/running-containers/multi-container-applications/) (<https://docs.docker.com/get-started/docker-concepts/running-containers/multi-container-applications/>)

[SDKs Overview - Microsoft Graph](https://learn.microsoft.com/en-us/graph/sdks/sdks-overview) (<https://learn.microsoft.com/en-us/graph/sdks/sdks-overview>)

[Scrapy vs. Playwright - brightdata.com](https://brightdata.com/blog/web-data/scrapy-vs-playwright) (<https://brightdata.com/blog/web-data/scrapy-vs-playwright>)

[Set up notifications for changes in user data - Microsoft Graph](https://learn.microsoft.com/en-us/graph/change-notifications-delivery-webhooks) (<https://learn.microsoft.com/en-us/graph/change-notifications-delivery-webhooks>)

[Simple Graph Database Setup with Neo4j and Docker Compose - medium.com](https://medium.com/@matthewghannoum/simple-graph-database-setup-with-neo4j-and-docker-compose-061253593b5a) (<https://medium.com/@matthewghannoum/simple-graph-database-setup-with-neo4j-and-docker-compose-061253593b5a>)

[TeamsBridge GitHub Repository - amoreng](https://github.com/amoreng/TeamsBridge) (<https://github.com/amoreng/TeamsBridge>)

[Technology Stack for Microservices - aalpha.net](https://www.aalpha.net/blog/technology-stack-for-microservices/) (<https://www.aalpha.net/blog/technology-stack-for-microservices/>)

[The Conversational AI Pipeline - GitHub](https://github.com/kunan-sa/the-conversational-ai-pipeline) (<https://github.com/kunan-sa/the-conversational-ai-pipeline>)

[Throttling and batching - Microsoft Graph](https://learn.microsoft.com/en-us/graph/throttling) (<https://learn.microsoft.com/en-us/graph/throttling>)

[Throttling MS Graph API/Sharepoint Online - Reddit](https://www.reddit.com/r/AZURE/comments/16hu3es/throttling_ms_graph_apisharepoint_online/) (https://www.reddit.com/r/AZURE/comments/16hu3es/throttling_ms_graph_apisharepoint_online/)

[throttling-limits.md - GitHub](https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/concepts/throttling-limits.md) (<https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/concepts/throttling-limits.md>)

[Tips for using WeasyPrint for PDF generation - reddit.com](https://www.reddit.com/r/Python/comments/jmfbwk/tips_for_using_weasyprint_for_pdf_generation/) (https://www.reddit.com/r/Python/comments/jmfbwk/tips_for_using_weasyprint_for_pdf_generation/)

[To Do API Overview - Microsoft Graph](https://learn.microsoft.com/en-us/graph/todo-concept-overview) (<https://learn.microsoft.com/en-us/graph/todo-concept-overview>)

[Tools - OpenWebUI Community](https://openwebui.com/tools) (<https://openwebui.com/tools>)

[Tools - OpenWebUI Docs](https://docs.openwebui.com/features/plugin/tools/) (<https://docs.openwebui.com/features/plugin/tools/>)

[Tools Development - OpenWebUI Docs](https://docs.openwebui.com/features/plugin/tools/development/) (<https://docs.openwebui.com/features/plugin/tools/development/>)

[Top 10 Open-Source Graph Databases - index.dev](https://www.index.dev/blog/top-10-open-source-graph-databases) (<https://www.index.dev/blog/top-10-open-source-graph-databases>)

[Top 10 Open-Source RAG Evaluation Frameworks You Must Try - medium.com](https://walseis-arel.medium.com/top-10-open-source-rag-evaluation-frameworks-you-must-try-15c40d0ba2c0) (<https://walseis-arel.medium.com/top-10-open-source-rag-evaluation-frameworks-you-must-try-15c40d0ba2c0>)

[Top 10 Tools for Web Scraping - firecrawl.dev](https://www.firecrawl.dev/blog/top_10_tools_for_web_scraping) (https://www.firecrawl.dev/blog/top_10_tools_for_web_scraping)

[Top 10 Ways to Generate PDFs in Python - nutrient.io](https://www.nutrient.io/blog/top-10-ways-to-generate-pdfs-in-python/) (<https://www.nutrient.io/blog/top-10-ways-to-generate-pdfs-in-python/>)

[Top 5 Open-Source Web Scraping Tools for Developers - firecrawl.dev](https://www.firecrawl.dev/blog/top-5-open-source-web-scraping-tools-for-developers) (<https://www.firecrawl.dev/blog/top-5-open-source-web-scraping-tools-for-developers>)

[Top Unstructured Data Tools - wajidkhan.info](https://wajidkhan.info/top-unstructured-data-tools/) (<https://wajidkhan.info/top-unstructured-data-tools/>)

[Troubleshoot Microsoft Graph authorization errors - Microsoft Graph](https://learn.microsoft.com/en-us/graph/resolve-auth-errors) (<https://learn.microsoft.com/en-us/graph/resolve-auth-errors>)

[Tutorial: Deploy a multi-container group using Docker Compose - learn.microsoft.com](https://learn.microsoft.com/en-us/azure/container-instances/tutorial-docker-compose) (<https://learn.microsoft.com/en-us/azure/container-instances/tutorial-docker-compose>)

[Unstructured Answer: Structured vs. Unstructured Data in AI - restack.io](https://www.restack.io/p/unstructured-answer-structured-vs-unstructured-cat-ai) (<https://www.restack.io/p/unstructured-answer-structured-vs-unstructured-cat-ai>)

[Unstructured Data, Apache Tika, and Beer - dankeelely.wordpress.com](https://dankeelely.wordpress.com/2016/05/18/unstructured-data-apache-tika-and-beer/) (<https://dankeelely.wordpress.com/2016/05/18/unstructured-data-apache-tika-and-beer/>)

[Unstructured is an ETL solution for LLMs - news.ycombinator.com](https://news.ycombinator.com/item?id=36616799) (<https://news.ycombinator.com/item?id=36616799>)

[unstructured - github.com](https://github.com/Unstructured-IO/unstructured) (<https://github.com/Unstructured-IO/unstructured>)

[Use Docker Compose to deploy multiple containers - learn.microsoft.com](https://learn.microsoft.com/en-us/azure/ai-services/containers/docker-compose-recipe) (https://learn.microsoft.com/en-us/azure/ai-services/containers/docker-compose-recipe)

[Use GitHub with Loop - Microsoft Support](https://support.microsoft.com/en-us/office/use-github-with-loop-5a4d95d5-3c59-4de8-a208-c9c8ab05a4fb) (https://support.microsoft.com/en-us/office/use-github-with-loop-5a4d95d5-3c59-4de8-a208-c9c8ab05a4fb)

[Use the API - Microsoft Graph](https://learn.microsoft.com/en-us/graph/use-the-api) (https://learn.microsoft.com/en-us/graph/use-the-api)

[Use the Planner REST API - Microsoft Graph](https://learn.microsoft.com/en-us/graph/planner-concept-overview) (https://learn.microsoft.com/en-us/graph/planner-concept-overview)

[Use the pgvector extension in Azure Database for PostgreSQL - Flexible Server - learn.microsoft.com](https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/how-to-use-pgvector) (https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/how-to-use-pgvector)

[Using Awesome-Compose to Build and Deploy Your Multi-Container Application - docker.com](https://www.docker.com/blog/using-awesome-compose-to-build-and-deploy-your-multi-container-application/) (https://www.docker.com/blog/using-awesome-compose-to-build-and-deploy-your-multi-container-application/)

[Using PostgreSQL over MySQL in 2024 - reddit.com](https://www.reddit.com/r/PostgreSQL/comments/1e4wq0p/using_postgresql_over_mysql_in_2024/) (https://www.reddit.com/r/PostgreSQL/comments/1e4wq0p/using_postgresql_over_mysql_in_2024/)

[vercel/ai-chatbot - GitHub](https://github.com/vercel/ai-chatbot) (https://github.com/vercel/ai-chatbot)

[What is a microservice architecture? - contentful.com](https://www.contentful.com/resources/microservice-architecture/) (https://www.contentful.com/resources/microservice-architecture/)

[What is a Microservice Tech Stack? - hygraph.com](https://hygraph.com/blog/what-is-microservice-tech-stack) (https://hygraph.com/blog/what-is-microservice-tech-stack)

[What is a vector database? — Qdrant - medium.com](https://medium.com/@qdrant/what-is-a-vector-database-qdrant-fb8cd9b3b524) (https://medium.com/@qdrant/what-is-a-vector-database-qdrant-fb8cd9b3b524)

[What is pgvector? - enterprisedb.com](https://www.enterprisedb.com/blog/what-is-pgvector) (https://www.enterprisedb.com/blog/what-is-pgvector)

[What is the best scraper tool right now? Firecrawl, Scrapy, etc. - reddit.com](https://www.reddit.com/r/LocalLLaMA/comments/1jw4yqv/what_is_the_best_scraper_tool_right_now_firecrawl/) (https://www.reddit.com/r/LocalLLaMA/comments/1jw4yqv/what_is_the_best_scraper_tool_right_now_firecrawl/)

[What are the differences between LangChain and other LLM frameworks like LlamaIndex or Haystack? - milvus.io](https://milvus.io/ai-quick-reference/what-are-the-differences-between-langchain-and-other-llm-frameworks-like-llamaindex-or-haystack) (https://milvus.io/ai-quick-reference/what-are-the-differences-between-langchain-and-other-llm-frameworks-like-llamaindex-or-haystack)

[wokelo-docs - pypi.org](https://pypi.org/project/wokelo-docs/) (https://pypi.org/project/wokelo-docs/)

[Working with Docker Compose - code.visualstudio.com](https://code.visualstudio.com/docs/containers/docker-compose) (https://code.visualstudio.com/docs/containers/docker-compose)

[anyone_know_what_the_long_term_trend_between - reddit.com](https://www.reddit.com/r/PostgreSQL/comments/1f8ta5p/anyone_know_what_the_long_term_trend_between/) (https://www.reddit.com/r/PostgreSQL/comments/1f8ta5p/anyone_know_what_the_long_term_trend_between/)