**Integrate RAG Chatbot into Microsoft Teams**

**Goal:**

Enable users to interact with our Retrieval-Augmented Generation (RAG) chatbot directly inside Microsoft Teams for a seamless, integrated conversational experience.

**Approach 1: Azure Bot Service with Teams Channel**

This approach involves exposing your Streamlit app’s RAG logic as an API and connecting it to a bot built with Azure Bot Service, configured to communicate via the Microsoft Teams channel.

Detailed Steps

1. Expose Streamlit Backend as an API
   * Modify your Streamlit app to include an API endpoint using a framework like FastAPI or Flask.
   * Host the API on a server with HTTPS (e.g., Azure Web App, Heroku, or your existing Streamlit server).
   * Ensure the API accepts JSON payloads with user queries and returns RAG-generated responses.
2. Create an Azure Bot Service Resource
   * In the Azure Portal, navigate to “Create a resource” > “AI + Machine Learning” > “Bot Services” > “Azure Bot”.
   * Choose “Multi-tenant” for general use or “Single-tenant” for restricted access.
   * Provide a bot handle, subscription, resource group, and region.
   * Select the “F0” (free) or “S1” (standard) pricing tier (see cost details below).
   * Generate an App ID and password for the bot, stored in Azure Key Vault for security.
3. Develop the Bot Using Microsoft Bot Framework SDK
   * Use the Bot Framework SDK (Python) to create a bot that calls your API.
4. Deploy the Bot to Azure
   * Create a bot application structure
5. Add the Teams Channel
   * In the Azure Portal, go to your bot resource > “Channels” > “Microsoft Teams”.
   * Accept the Teams terms and enable the channel.
   * Note the App ID for Teams integration.
6. Create a Teams App Manifest
   * In the Teams Developer Portal (https://dev.teams.microsoft.com), create a new app.
   * Add a bot capability and use the App ID from Azure.
   * Configure bot settings (e.g., name, description, commands like /ask).
   * Download the app manifest (ZIP file).
7. Deploy and Test in Teams
   * Upload the app manifest to Teams:
     + For personal testing, sideload the app in Teams (“Apps” > “Upload a custom app”).
     + For organization-wide use, share the manifest with your Teams admin for tenant app catalog deployment.
   * Test the bot by sending messages in a personal chat or team channel.
   * Verify that RAG responses and citations are displayed correctly (e.g., using markdown or adaptive cards).

Cost Considerations for Approach 1

* Azure Bot Service:
  + F0 Tier (Free): 10,000 messages/month, suitable for testing or low-traffic bots. Limited to one bot per subscription.
  + S1 Tier (Standard): ~$0.50/1,000 messages (pay-as-you-go). For higher traffic, expect $50–$500/month depending on usage (e.g., 100,000–1M messages).
* Azure Web App (API Hosting):
  + Free Tier: Limited to 60 CPU minutes/day, suitable for testing.
  + B1 Tier: ~$15–$50/month for small-scale apps (1 core, 1.75 GB RAM).
  + Higher Tiers: $100–$500/month for production apps with auto-scaling.
* Azure Key Vault: ~$0.03/10,000 transactions, typically <$5/month for small bots.
* Note: For precise pricing, check the Azure Pricing Calculator (https://azure.microsoft.com/pricing/calculator/). Costs vary by region and usage.

Approach 2: Microsoft 365 Agents Toolkit with Teams AI Library

Description

This approach uses the Microsoft 365 Agents Toolkit (Teams AI Library) to build an AI-powered bot with native RAG support. The bot connects to your Streamlit backend or Microsoft 365 data sources (e.g., SharePoint) and is deployed to Teams.

Detailed Steps

1. Set Up Development Environment
   * Install Visual Studio Code and the Microsoft 365 Agents Toolkit extension.
   * Install Node.js (for Teams AI Library dependencies) and Python (for compatibility with your Streamlit app).
   * Ensure access to an LLM provider (e.g., Azure OpenAI or another model compatible with your RAG setup).
2. Create a Teams AI Bot
   * In Visual Studio Code, open the Microsoft 365 Agents Toolkit extension.
   * Select “Create a New Agent/App” > “Agent for Teams” > “AI Agent” > “Python”.
   * Name the project (e.g., rag-teams-bot) and configure with your LLM credentials (e.g., Azure OpenAI endpoint and key).
3. Integrate RAG Logic
   * Create a custom DataSource to connect to your Streamlit RAG backend.
4. Configure RAG Data Source
   * If using Microsoft 365 data (e.g., SharePoint), configure Azure AI Search as a data source:
     + Create an Azure AI Search resource and index your data (e.g., SharePoint documents).
     + Update config.json to use the Azure AI Search data source.
   * For your Streamlit backend, ensure the API returns structured responses compatible with the Teams AI Library.
5. Deploy the Bot to Azure
   * Deploy the bot to Azure Web App:
   * Configure the bot’s messaging endpoint in Azure Portal (e.g., https://rag-teams-bot.azurewebsites.net/api/messages).
   * Register the bot in Azure AD and note the App ID.
6. Add the Teams Channel
   * In the Azure Portal, go to the bot resource > “Channels” > “Microsoft Teams”.
   * Enable the Teams channel and accept the terms.
7. Create and Deploy Teams App Manifest
   * In the Teams Developer Portal, create a new app and add a bot capability.
   * Use the App ID from Azure and configure bot settings (e.g., commands, description).
   * Download the app manifest and upload it to Teams for testing or tenant-wide deployment.

Cost Considerations for Approach 2

* Azure Web App (Bot Hosting):
  + Free Tier: Limited to 60 CPU minutes/day, suitable for testing.
  + B1 Tier: ~$15–$50/month for small-scale apps.
  + Higher Tiers: $100–$500/month for production with auto-scaling.
* Azure AI Search (Optional for RAG):
  + Basic tier: ~$75/month for vector search and indexing.
  + Standard tier: ~$200–$500/month for higher capacity.
* Azure OpenAI (If Used):
  + Pay-as-you-go: ~$0.001–$0.01 per 1,000 tokens, depending on model (e.g., GPT-4o).
  + Example: 100,000 queries/month with 1,000 tokens/query ≈ $100–$1,000/month.
* Azure Key Vault: ~$0.03/10,000 transactions, typically <$5/month.
* Note: Use the Azure Pricing Calculator (https://azure.microsoft.com/pricing/calculator/) for precise estimates.