Guia de Implantação do ChatTCU

TRIBUNAL DE CONTAS DA UNIÃO

1. Introdução

Este manual tem como objetivo orientar órgãos e empresas interessadas na implantação do ChatTCU, uma solução baseada em inteligência artificial (IA) desenvolvida pelo Tribunal de Contas da União (TCU). Este documento oferece orientações técnicas sobre infraestrutura, configuração, operação e personalização da ferramenta.

2. Visão Geral da Solução

O ChatTCU é um assistente virtual que utiliza IA para auxiliar servidores e colaboradores no acesso a informações institucionais e na automação de tarefas. A ferramenta é composta pelos seguintes componentes:

- **Backend**: Desenvolvido em Python, utilizando o framework FastAPI.
- **Frontend**: Construído em React.js para interação com o usuário.
- **Serviços de Nuvem**: Para armazenamento e processamento de dados.
- **Modelos de IA**: Integra modelos de linguagem como GPT-4o (Azure OpenAI), Gemini 2.5 Pro (Google Cloud) e Claude 3.7 Sonnet (Amazon Web Services).
- **Modelo de embeddings**: Utiliza o text-embedding-ada-003-large.

3. Funcionalidades Principais

- **RAG (Retrieval-Augmented Generation):** Permite ao ChatTCU buscar informações em bases de dados internas antes de gerar respostas, aumentando a precisão e relevância.
- **Memória de Conversação:** O sistema mantém um histórico de interações dentro de um chat para contextualizar respostas, mas não entre diferentes sessões de chat.
- **Upload de Documentos:** Usuários podem fazer upload de documentos em formatos como PDF, Word, Excel, CSV, MP3 e MP4, que são processados e indexados para consultas futuras.
- **Arquitetura de Decisão:** O LLM atua como agente principal para decidir entre diferentes ferramentas (tools) baseadas na pergunta do usuário.
- **Agentes especializados:** A aplicação disponibiliza agentes de IA visando a otimização do trabalho dos usuários, como agente de geração de ata de reunião, agente de criação de minuta de voto e agente de análise de dados de certificados/diplomas de cursos.

4. Requisitos de Infraestrutura

O ChatTCU opera em uma estrutura híbrida, com parte dos serviços hospedados no data center do TCU em Brasília, e outra parte na nuvem.

Serviços Principais:

Serviço	Tecnologia
Armazenamento de Arquivos	Azure Blob Storage
Banco de Dados Vetorial	Azure AI Search
Banco de Dados NoSQL	ElasticSearch, Cosmos DB
Computação e Orquestração	Kubernetes (gerenciado via Rancher)
Gerenciamento de chaves	Azure Key Vault
Gerenciamento de Identidade e Acesso	Microsoft Entra ID (OIDC)

⚠️ Todos os serviços mencionados podem ser substituídos por equivalentes, desde que sejam realizados os devidos ajustes no código e na configuração da aplicação.

5. Arquitetura Geral

1. **Frontend:**
   - Desenvolvido em TypeScript com ReactJS, sem frameworks adicionais.
   - Utiliza o React Query para gerenciamento de requisições e Axios para chamadas HTTP.
   - Desenvolvido como um Single Page Application (SPA) que utiliza o Nginx para servir arquivos estáticos.
   - Armazena informações temporárias no LocalStorage.

2. **Backend**:

- Desenvolvido em Python 3.1, utilizando o framework FastAPI para a criação de endpoints.
- Hospedado em um cluster Kubernetes gerenciado pelo Rancher.
- Organização em microserviços responsáveis por:
    - Controle das interações.
    - Integração com bases internas.
    - Geração de respostas por meio de agentes inteligentes.

3. **Bases de Dados e Armazenamento**:

- **Azure AISearch**: Buscas vetoriais das bases internas do TCU.
- **CosmosDB**: Armazenamento de metadados e documentos indexados.
- **Blob Storage**: Armazenamento de documentos enviados pelos usuários.
- **Elasticsearch**: Armazenamento de histórico de interações dos usuários.

4. **Integrações**:

- **Azure Functions:** Utilizadas para processamento paralelo de documentos, garantindo escalabilidade e eficiência.
- **Azure key Vault:** Utilizado para gerenciamento de senhas.
- **Spring Cloud:** Utilizado para armazenamento de URLs e demais valores (como logins) que não sejam senhas.
- **Azure OpenAI:** Fornece modelos de linguagem natural, como o GPT-4o, para processamento de texto e imagem.
- **Amazon Web Services (AWS):** Fornece modelos de linguagem natural, como o Claude 3.7 Sonnet, para processamento de texto e imagem.
- **Google Cloud:** Fornece modelos de linguagem natural, como o Gemini 2.5 Pro, para processamento de texto e imagem.
- Microsoft Entra ID (OICD): Utilizado para gerenciamento de acesso.

6. Fluxo de Funcionamento da Aplicação

1. Usuário envia uma pergunta pelo frontend.
2. Backend processa a solicitação e consulta:
    - Memória do usuário (histórico de mensagens).
    - Bases relevantes via RAG (Retrieval-Augmented Generation).
3. Modelo de IA gera a resposta baseada nos dados retornados.
4. Resposta é exibida ao usuário.

7. Processamento de Documentos

- **Upload:** Arquivos são armazenados no Azure Blob Storage.
- **Indexação:** O conteúdo do arquivo é extraído, dividido em trechos (chunks) e indexado no Azure AI Search para buscas futuras.
- **Busca e Recuperação:** O sistema utiliza técnicas de busca semântica e busca vetorial, dependendo da natureza da consulta. Isso é gerenciado por um sistema de agentes que decide qual abordagem usar com base na consulta do usuário.

8. Integração com Microsoft Teams

O ChatTCU pode ser integrado ao Microsoft Teams, permitindo que os usuários acessem a ferramenta diretamente dentro do ambiente de colaboração.

Isso é feito através de uma configuração específica que replica o front-end no Azure Blob Storage.

9. Configuração e Personalização

Nuvem:

- Provisionar recursos na nuvem.
- Mapear serviços utilizados para equivalentes na nuvem contratada.

Nome e Identidade:

- Personalizar o nome da ferramenta e identidade visual.

Orquestração:

- Configurar Kubernetes para a aplicação.

Monitoramento:

- Utilizar ferramentas como API Management para controle de uso.
- Implementar logs detalhados para rastrear eventos e identificar erros.

Autenticação:

- Utilizar um sistema de autenticação compatível, como Azure Entra ID (OIDC), ou adaptar as credenciais para as políticas do órgão.

Backend

- Instalar Python 3.11 ou superior.
- Configurar o ambiente virtual (venv) e instalar dependências (pip install -r requirements.txt).
- Configurar integrações com APIs de IA.

Frontend

- Instalar Node.js (versão recomendada: LTS).
- Instalar dependências (npm install).
- Compilar a aplicação (npm run build).
- Hospedar o frontend em servidores web estáticos (ex: Nginx).

10. Boas Práticas e Manutenção

Treinamento de Usuários:

- Realizar workshops para introduzir o uso da ferramenta.

Atualizações:

- Monitorar lançamentos de novas versões e aplicar correções.

Monitoramento:

- Acompanhar uso e custo.
  - O custo principal está associado ao uso dos modelos de linguagem, com variações significativas dependendo do modelo e do volume de uso. A infraestrutura na nuvem permite escalabilidade conforme a demanda.
- Implementar logs e feedback para melhorar respostas.

Segurança:

- Configurar políticas de acesso.
- Monitorar vulnerabilidades.

7. Colaboração e Suporte

- **Canal de Suporte:** Enviar dúvidas, sugestões e relatos de problemas para o TCU por meio do e-mail nia@tcu.gov.br.
- **Comunidade:** Incentiva-se a criação de uma rede de usuários e desenvolvedores para troca de conhecimentos e melhorias contínuas no uso da IA no setor público.

12. Considerações Finais

A implementação do ChatTCU demonstra o compromisso do TCU com a modernização e transformação digital no setor público. A adoção dessa tecnologia requer não só infraestrutura, mas também responsabilidade no uso ético e seguro da inteligência artificial.

TRIBUNAL DE CONTAS DA UNIÃO