Analisi del Codice Sorgente: [main.py](http://main.py)

Questo file è il cuore dell'applicazione frontend, sviluppata in Python con il framework Flask. Il suo ruolo principale è gestire l'interazione con l'utente e comunicare con il database Redis per registrare e visualizzare i voti.

1. Setup e Configurazione

La prima parte del codice si occupa di inizializzare l'applicazione e di configurare le variabili necessarie.

Importazioni: Vengono importate le librerie essenziali come Flask per il framework web e redis per la connessione al database. os viene usato per leggere le variabili d'ambiente, un'ottima pratica per la containerizzazione.

Caricamento delle Variabili: Il codice verifica la presenza di variabili d'ambiente (VOTE1VALUE, VOTE2VALUE, TITLE). Se queste non sono impostate, utilizza i valori predefiniti dal file di configurazione (config\_file.cfg).

Questo permette di personalizzare l'applicazione senza dover modificarne il codice.

2. Connessione a Redis

Questa sezione è cruciale per il funzionamento dell'applicazione.

Indirizzo del Server: L'indirizzo del server Redis viene letto dalla variabile d'ambiente REDIS. Nel nostro caso, questo valore è azure-vote-back, il nome del servizio Kubernetes che abbiamo definito.

Gestione della Connessione: Il blocco try...except tenta di stabilire una connessione a Redis e gestisce gli eventuali errori di connessione. Se la connessione fallisce, l'applicazione termina.

Inizializzazione dei Voti: Vengono inizializzati i conteggi dei voti per le due opzioni (Cats e Dogs) nel database Redis, assicurando che siano sempre disponibili.

3. Gestione delle Richieste Web

La funzione index() gestisce sia le richieste GET che POST inviate dall'utente.

Metodo GET: Quando un utente apre la pagina per la prima volta, viene eseguita la logica GET. L'applicazione si connette a Redis per recuperare i voti correnti per entrambe le opzioni (button1 e button2), e passa questi valori al template index.html per la visualizzazione.

Metodo POST: Quando l'utente clicca su un pulsante per votare, viene inviata una richiesta POST. Il codice verifica se il voto è per "Cats" o "Dogs" e incrementa il contatore corrispondente nel database Redis.

Successivamente, recupera i nuovi valori e aggiorna la pagina per mostrare i risultati. Il metodo POST gestisce anche un'opzione per azzerare i voti ('vote': 'reset').