Rate limiting is a technique where the number of requests made by clients to an API or server are limited over a set time period. This helps prevent abuse and ensures fair usage among all clients.  
  
Redis can be used for rate limiting by performing two actions: keeping track of the number of calls made within a certain time frame, and rejecting additional calls once that limit has been reached.  
  
Here's how it works:  
  
1. Set up Redis as your data store.  
2. For each client request, increment a counter in Redis with their IP address (or another identifier).  
3. Check if this count exceeds the allowed limit; if so, reject any subsequent requests until the reset timer expires.  
4. If not exceeded yet, continue processing the request normally while updating counters accordingly.  
  
Another approach is using Nginx as reverse proxy to act as gateway between web server like ExpressJS(HTTP) or Python(Flask/Django).  
  
Nginx allows us to define rules which we use during forwarding handling e.g  
  
server