We need rate limiter to save the server from DDOS attack. In DDOS attack there are lots of request are being sent within a very small amount of time. But the server has limited RAM and disk capacity. So in DDOS attack servers get down.

There are various kinds of rate limiter algorithm. They are: token bucket, leaking bucket, fixed window counter, sliding window log, sliding window counter.

In token bucket, the bucket has fixed capacity of token. For example the bucket has capacity to have 4 token. If the 5th tokens arrives it will be overflowed. That means the request will be denied. But there is another term called refiller. The bucket can be refilled with new 2 tokens after each 1 min. So after each 1 min it can handle two new request. You can config the number of refiller. After one minutes old two token will be overflowed. When a new request comes it will check the counter value then it will decrease the counter value by 1 until the counter value reach to 0. If the requests get counter value as 0 the request will be denied with status code 429 which means excess request in a time limit. But after 1 min the value of the counter will be 2. So it can again entertain requests.

In a leaking bucket, the bucket has fixed size and the leaking is constant that means the request will be processed in a constant rate. If it gets more request than its capacity, there will be an overflow. Queue is used in this algorithm. Before sending the request in the queue it will check whether the queue is filled up or not. If not filled up it will send the request to the queue. If it is filled up it will deny the request with 429. There is a disadvantage of leaking bucket. For example it may not  handle the request of amazon prime. Because there is some constant burst of request. It can not handle it.

In fixed window counter, time is divided into fixed slot. Within each slot it can handle certain amount of request. If it gets more request than that it will deny that request. For example if there is a 5 min slot that means there be slot from 8 to 8 and 5, 8 and 5 to 8 and 10, 8 and 10 to 8 and 15 so on and so on. Lets say it can handle 3 request in each time slot. There is a disadvantage of this algorithm. Lets say it gets 3 requests in 8 and 4 so the request will be processed and again gets 3 requests in 8 and 6 so these will also be processed as they come in different time slot. But if we consider 5 min around this request it actually handles 6 request at that span of time. For this system might not be ready.

This problem is solved by sliding window log. Whenever there is a request, there will be created a log in log file with the timestamp of the request. For example the sliding window size is 1 min. So if a request comes at [10.00.20](tel:100020) then the staring point of the window will be [10.00.20](tel:100020). Lets say it can handle 3 request in a slide so it will except at most 3 request within [10.00.20](tel:100020) to [10.01.20](tel:100120). If there more requests come, they all will be rejected. If after [10.01.20](tel:100120) next request comes in [10.01.45](tel:100145) then the starting point of the sliding window will be [10.01.45](tel:100145). And it will remove the first log from the file. There is a point to be noticed, there will be log for also denied request. As a result the file becomes large and it takes unnecessary storage.

Sliding window counter is combination of two algorithms one is fixed window counter and another one is sliding window log. It uses the benefits of both of the window. Lets say the window size is 1 min and the request can be handle in 1 min is 5. Lets say there are 6 requests from [08.00.00](tel:080000) to [08.01.00](tel:080100) and 2 requests from [08.01.01](tel:080101) to [08.01.50](tel:080150). So how many requests are there from [08.00.50](tel:080050) to [8.01.50](tel:80150)  
2 (this the number of requests from [08.01.01](tel:080101) to [08.01.50](tel:080150)) + (6 (number of requests from [08.00.00](tel:080000) to [08.00.01) / 60](tel:08000160) (seconds in the minute from [08.00.00](tel:080000) to [08.00.01](tel:080001)))  \* 10 (10 seconds from [08.00.50](tel:080050) to [08.01.00](tel:080100)) = 2 + 1 = 3.

Components for designing rate limiter. We have seen so far we need counter. We will implement counter using redis. We need config files. If there is not gate way in the mid of client and server we can host rate limiter with server. If there is a gateway, the rate limiter can be hosted with it. So the client request will first go to the rate limiter. Based on its response the request will hit the server. There will be redis with rate limiter for counter and cache to store config file. There can be distributed rate limiter. Means there can be multiple rate limiter. Redis is centralized data store. The requests from rate limiters will go to the redis. But redis has no atomicity. That means if multiple requests come parallelly from rate limiters to redis and the counter value is 2. What will happen? It does not have atomicity but we can add atomicity.

====================  
Questions  
====================  
1. What rate limiter actually is?  
2. What is redis?  
3. How does redis work?  
4. Deeply understand dvantage and disadvantage of different rate limiter algorithms.