4. OPERATIONS CANDIDATE

1. Random Access Memory is very distinct from disk based storage, but disk-based storage can at times function as what we refer to as “Random Access Memory”. Random Access Memory is a faster to read/write means of storing data that is not persistent once the computer is shutdown or the application using it is exited. It is used as a means of speeding up computer access by having a location for data is to be repeatedly and/or rapidly read/written to, for example a fast-running program or an arrays values that are being algorithmically manipulated.

Disk-based storage is completely persistent storage that is read to or written slower (although advances in SSD technology “may”change that) and is intended for data that is retrievable after a computer restarts or is bigger than RAM is able to store. Examples would be your MP3 collection or a Photoshop Scratch Disk that is too big for RAM and does not need to be read/written to rapidly. Furthermore, OS’s such as Windows may use disk-based storage for RAM that is accessed infrequently as Virtual RAM, especially in machines will low Physical RAM (for example 2GB of RAM in a OS that requires a minimum of 2GB).

1. I have built my own computers for both desktop and server usage 10 times. Over the course of building computers, I have learned so much. One begins to learn which physical components of a computer are likely to fail, how they might fail, and why they might do so, in addition to how they interact with each other. This is useful when troubleshooting hardware errors. Furthermore, I have learned a significant amount regarding how to properly balance the operating power of each component. Although my first machine featured seriously imbalanced components (1000W PSU for 450W GPU and HDD bottlenecking CPU/RAM for example), I have learned how to properly allocate the strength of components based on how they will be limited by each other in order to get the fastest machine possible that is not bottlenecked by any of its component specifically. Furthermore, I have learned to not spend too little or not enough on any various parts based on these bottlenecks. I have also learned how specific applications require more resources in one area than another, and can tailor my builds accordingly.  
    As for running a server, I began to get an appreciation how it is important to minimize what is running at any given time, so that you can accommodate all the necessary traffic to the server and not waste system resources. Furthermore, I learned to always account for your load traffic, not your average traffic, because the instant you reach your maximum capacity your users will start to experience slowdowns in unnecessary ways and it will detract from their experience. Finally, one learns that the INSTANT you are visible to the internet as a whole in an indexable way (say running a website whose IP can be obtained by DNS crawlers), you will be under attack. Bots will try to bombard your server with a multitude of attacks (password bruteforcing on various accounts, exploits, etc) and I was forced to protect your server accordingly (IP banning, stealthing ports by sharing ports between services, disabling admin accounts, long passwords). It taught me the need for server security, and I am sure on an IP address that is connected to the government this will be even more aggressive.
2. I do not have any physical datacenter experience, but I run a MySQL database on Debian.