amazon
we pioneer

Systems Design
Interview Preparation Document

"Random forests, naïve Bayesian estimators, Restful services, gossip protocols, eventual consistency, data sharing, anti-entropy, Byzantine quorum, erasure coding, vector clocks ... walk into certain Amazon meetings, and you may momentarily think you've stumbled into a computer science lecture."
- Jeff Bezos, 2010 Shareholder letter

Amazon—a place where builders can build. We hire the world's brightest minds and offer them an environment in which they can invent and innovate to improve the experience for our customers. We want employees who will help share and shape our mission to be Earth's most customer-centric company. Amazon's evolution from Web site, to e-commerce partner, to development platform, is driven by the spirit of invention that is part of our DNA. We do this every day by solving complex technical and business problems with ingenuity and simplicity. We're making history, and the good news is that we've only just begun.

Thank you taking the time to speak with us. The tips below are intended to enhance your candidate experience.

Our employees tackle some of the most complex challenges in large-scale computing. Software development engineers, technical program managers, test engineers, technical program managers and user-interface experts work in small teams across the company to create experiences that our customers will be thrilled with.

Answering the Systems Design Question

When interviewing for a **Software Development Engineer, a Software Development Manager, or a Technical Program Manager** position at Amazon, you will likely have at least one interview focused on software systems design. This is a very important interview, so it's critical to prepare for it thoroughly. You'll know when you're being asked the systems design question because you'll be asked to design a software system.

Answering this question will be very interactive; the interviewer will ask you lots of questions related to the design and you are encouraged to ask the interviewer any necessary questions to complete your design. If you are suggesting technology to solve a problem, please make sure you understand how that technology works; it is more important that you understand how your solution solves the problem than specific technology solutions. It helps to think out loud and take hints from the interviewer. You will most likely be diagramming your design on a white board, so if you have access to a white board at home (or even just a pen and paper), writing these sort of designs out by hand can be great practice.

Topics to Review

- Often times, software systems need software components, something to store data, something to make decisions (such as business logic) and APIs, component relationships, and data flows. Reviewing software systems design diagrams (especially SOA or distributed software systems) can be helpful for preparation.

- Scaling is a critical component of software design at Amazon. It's important to consider scaling when diagramming and designing your software system. Prior to your interview be sure to research scalability concepts and technology prior to your interview such as caching, load balancing, non-relational databases, microservices and sharding.

- Knowledge of distributed systems, SOA, and n-tiered software architecture is very important in answering systems design questions. If you don't work with these concepts regularly, be sure to review them prior to your interview.

- Large software system designs often need to trade off availability, consistency, and other desirable performance characteristics. Be prepared to discuss these and other types of tradeoffs.

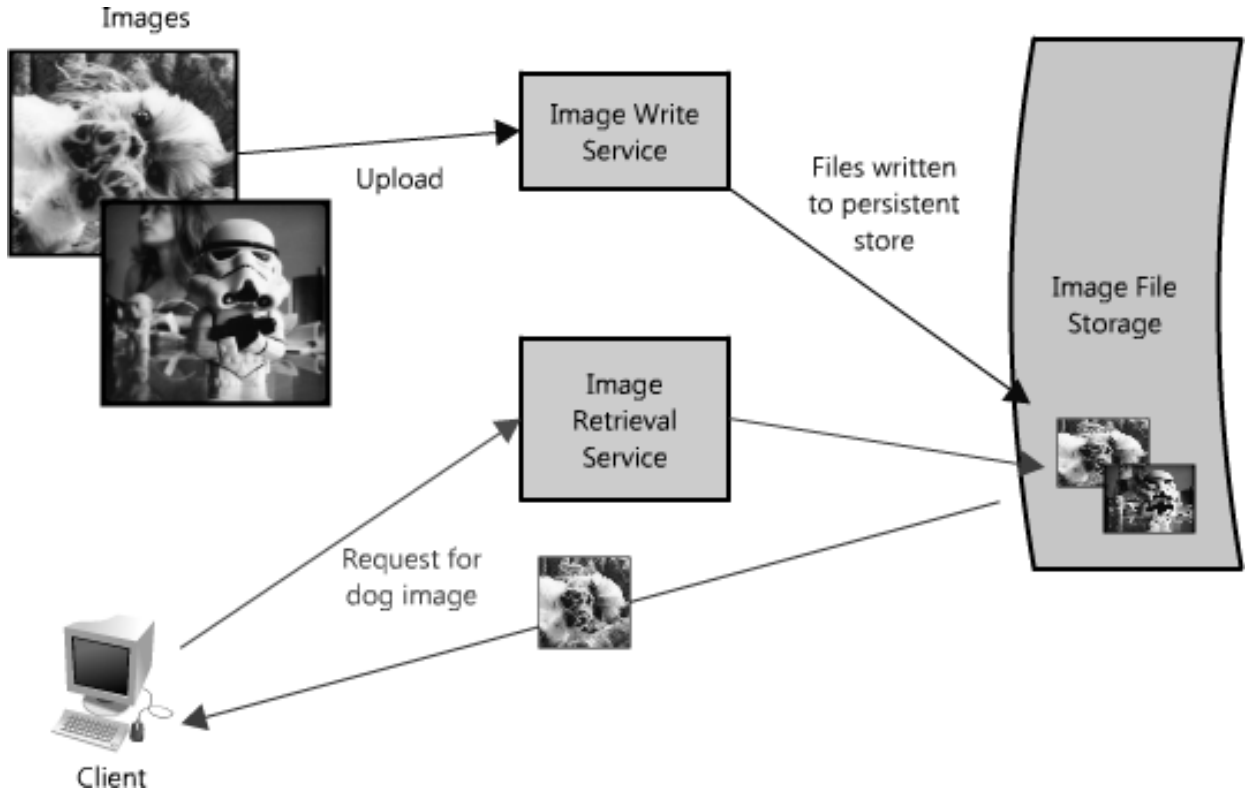
"Many of the problems we face have no textbook solution, and so we happily invent new ones."
- Jeff Bezos, 2010 Shareholder letter

Steps in the Systems Design Interview

1. Ask clarifying questions; while the interviewer won't try to trick you, they might be intentionally vague. It's important to know what sort of design the interviewer is looking for, so ask questions. When asking your questions, start with the customer in mind. Who is the customer and what problem are you solving for them?

2. As you ask clarifying questions, begin writing a list of requirements on the board. This should typically be the first thing you add to the white board.

3. Once you have a good idea on the sort of problems the system you are designing is supposed to solve, begin drawing a diagram on the white board to express your ideas. A great way to do this is to draw shapes to represent different software components and data sources and then arrows connecting them to show web services, APIs, and interactions between components.



4. Be prepared to discuss trade-offs in your design. With any software system there are multiple ways to design it. What advantages would yours have? Disadvantages? What if you were to change a component or process? Be prepared to discuss these questions.

5. Operational performance of your design is important as well. Be prepared to answer the following: how will you ensure this system is working at an acceptable level of performance? If a problem occurs, what will be involved in troubleshooting and resolving it quickly? What are the possible points of failure and how can they be made more robust against failure?

Amazon Press

- How Jeff Bezos built a business through experimentation
  http://fortune.com/2015/09/17/amazon-founder-ceo-jeff-bezos-skills/

- Why Maria Renz chooses to work for Amazon
  http://recode.net/2015/09/25/why-i-work-for-amazon-a-response/

- Amazon ranked in the Top 5 Most Attractive Employers for Computer Science by Universum
  http://top100.universumglobal.com/united-states-of-america/ranking/most-attractive-employers-ranking-for-computer-science-it

- Record Breaking Holiday Season 2014
  http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=2002024