

Building a reasonably popular web
application for the first time |

Who am I?

- Co-founder and chief architect at Hotjar.
- My development career started at the age of six when I wrote my first game - I've been hooked on programming ever since.
- Transitioned between different tech stacks throughout the years. Started with Python about 7 years ago and it's definitely my favorite language so far.

My definition of *reasonably popular*.

- We process around 400 000 API requests every minute.
- Our CDN delivers about 10 TB of data per day.
- We have roughly 3 TB of data in our primary data store (PostgreSQL), another 2 TB in our Elasticsearch cluster, and roughly 35 TB on Amazon S3.

We still use reasonably standard solutions.

NGINX

memCached

uWSGI

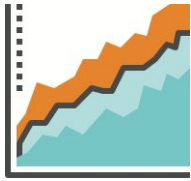
python

elastic

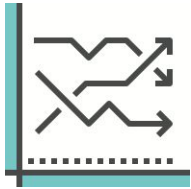
redis

PostgreSQL

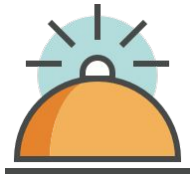
Lua



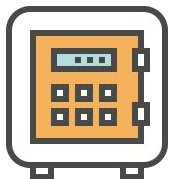
Log and monitor from day one.



Have a way to profile API calls.



Know when things fail.



Have a way to keep secrets.



Everything needs a limit
(even if it's really big).



Don't get too attached to a framework.



Choose components which allow for language interoperability.



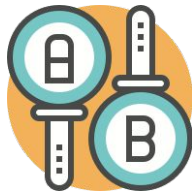
Plan for database downtime.



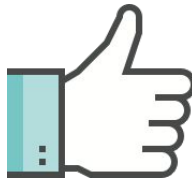
Have a way to share settings between backend and frontend.



Have a way to go into maintenance mode.



Feature flags are a great way to test things out before releasing them to everyone.



Accept different quality of code for different parts of the system.

Thank you for listening

@eriknhj | www.hotjar.com | careers.hotjar.com

