How to give everything away and still look expensive

"Fixing" Heroku Pricing

Peter van Hardenberg

Heroku Product / 6 year old-timer

A Tale in Five Acts

Act I: Launch

Act II: An Evolving Market

Act III: Wrangling Complexity

Act IV: Contact

Act V: Conclusions

Pricing defines product.

“What am I actually buying for how much?”

Act I: Launch


Koi 13
Database $15
Dynos $432
Workers $252
Add-ons $0
$699
estimated monthly cost
Database
Shared Cluster
Shared DBs offer highest peak performance.
Cost-effective, but performance is variable.
Blossom FREE
5MB database max
Free database for development
Koi $15
20GB database max
Fast but variable performance
Read about our recent
database pricing changes.
Dedicated
Dedicated DBs offer dedicated compute units
for predictable performance.
Ronin $200
1 compute unit
2TB database max
Fugu $400
5 compute units
2TB database max
Zilla $1600
20 compute units
2TB database max
Dynos and Workers
Crank your dynos to increase HTTP performance.
More dynos provide more concurrency. Crank your
workers to process background jobs from a queue.
More workers will empty the queue faster.
13 Dynos
$0.60/hour
7 Workers
$0.35/hour

Heroku Pricing 2009

$0.05 / dyno (first one is free)

databases can be added for an additional charge

pay-per-feature (logging, SSL, and so on)

What was good

focus on simplicity & ease of use

drove adoption effectively

matched ancient Ruby single-thread web servers

fit classic MVC design

abuse basically nonexistent: we lived below the radar

few serious customers anyway

Act II: An Evolving Market

Changes in the market

Heroku supports broad array of languages

Multi-threaded / asynchronous IO web servers

The rise of “microservices”

New and exciting workloads

The growth of abuse / gaming of the system

Friction from enterprise sales

Business Strategy Evolves

keep individual dev experience simple

support sophisticated production use-cases

small customers as strategic advocates

create a value axis & pricing experience centered around product features rather than commodity units (RAM/disk/CPU)

Adding simple stuff
makes simple stuff
complicated stuff.

Heroku Pricing mid 2015

dynos

databases

enterprise

add-ons

dynos

classic 1X dynos still nominally $0.05/hr

2X dynos for $0.10/hr

PX dynos for $0.80/hr (16x)

more dyno types were on the horizon

workers, web, one-off/interactive jobs, cron

per-application credit for 750 1X-dyno-equivalent-hours-per-month

databases

four tiers: hobby, standard, premium, enterprise

two database technologies: redis & postgres

many different sizes

major legacy of different plan names & limits

enterprise

different (additional SKUs) & discounting matrices

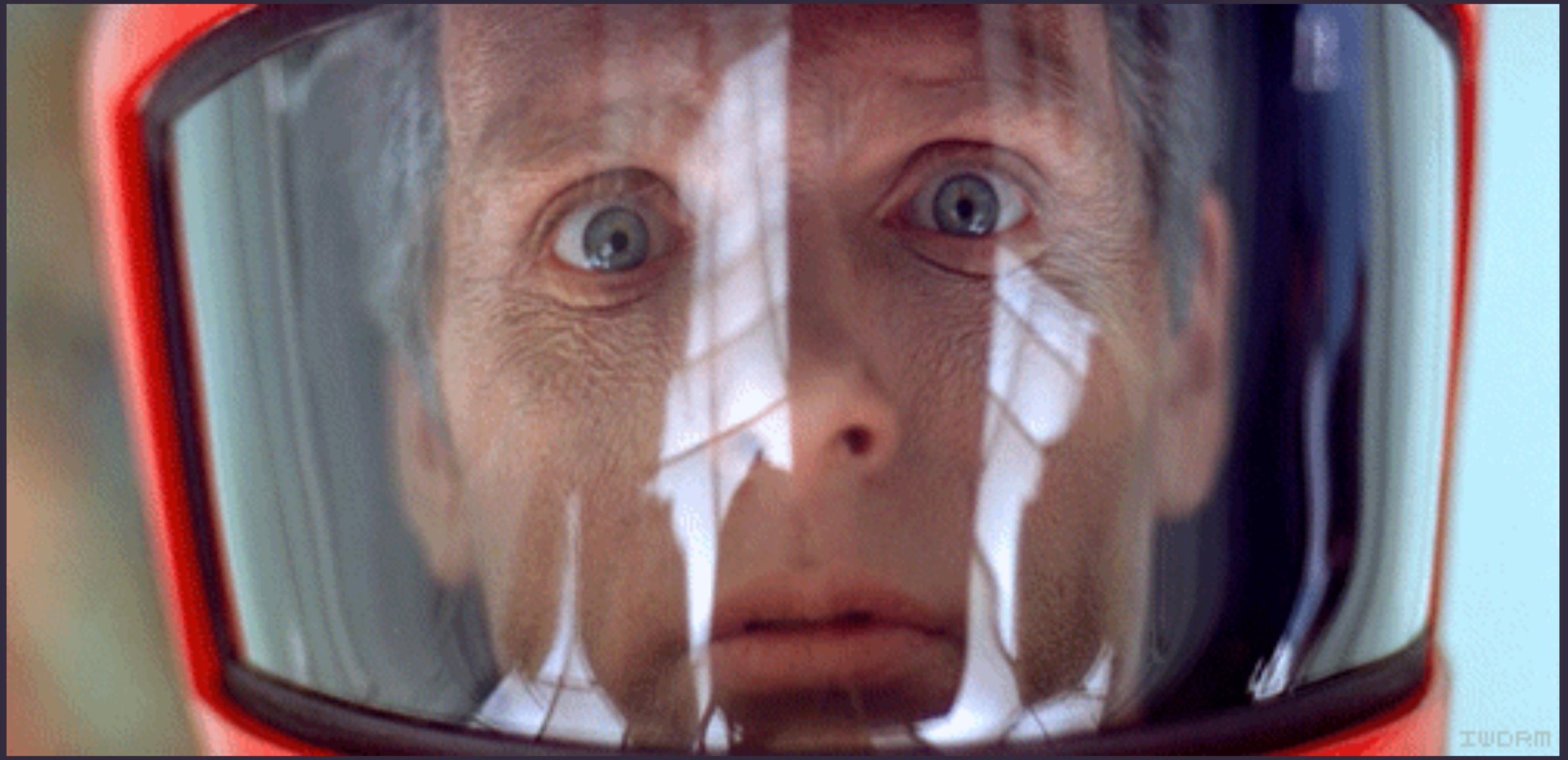
Technical Account Managers

Premium Support

custom terms

add-ons

total anarchy.

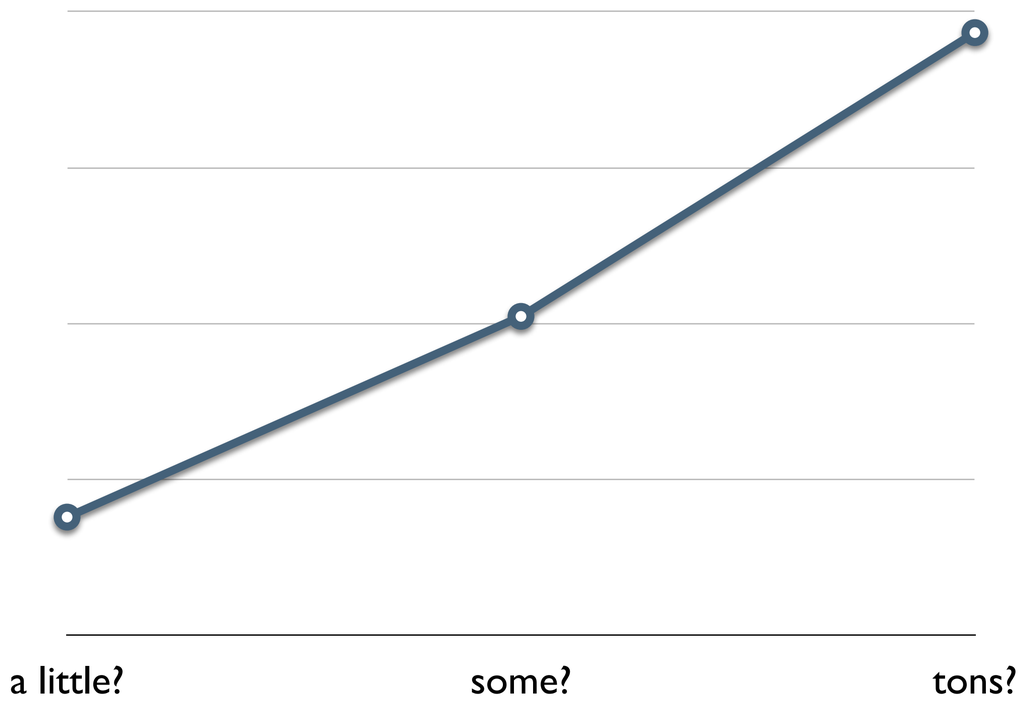

IWDRM

A user could get 75,000 dyno hours per month free.

(And still perceive us as “too expensive!”)

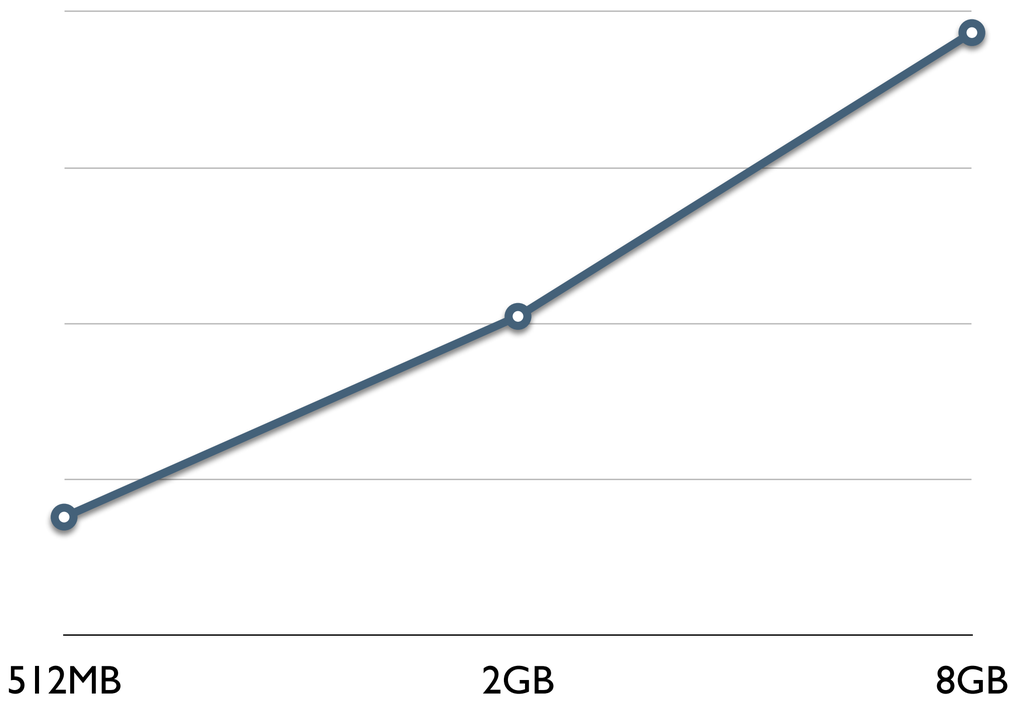
Inter-mezzo: Value Axis

Value Axis



What are they **buying**?

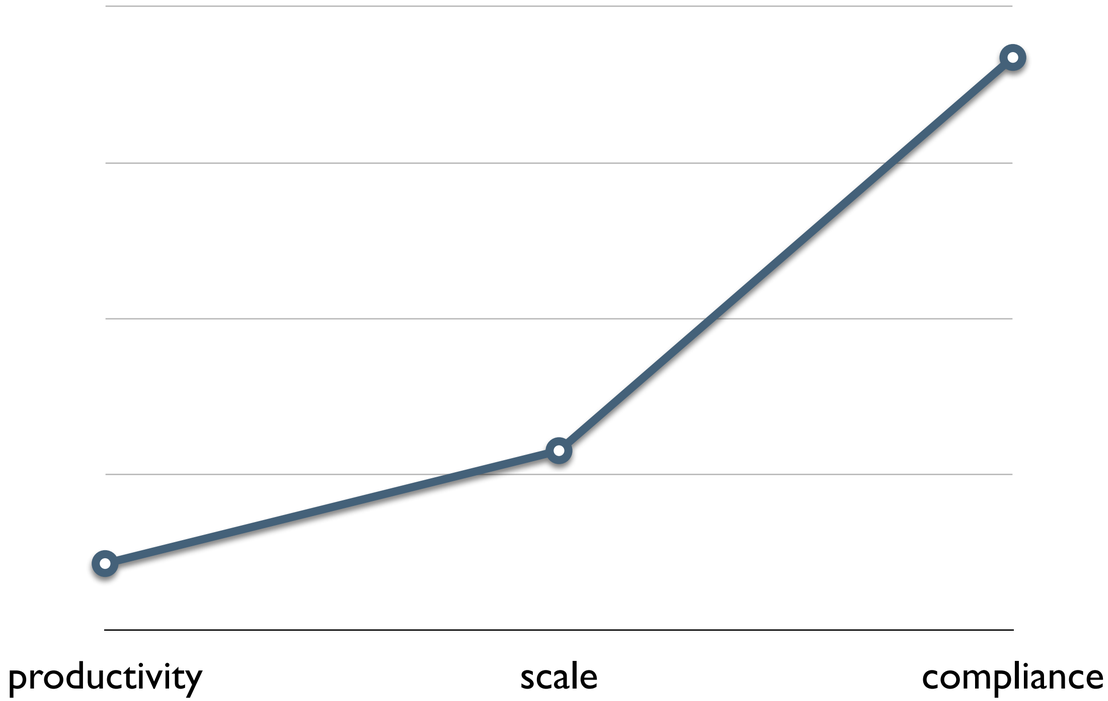
Value Axis



“Heroku costs more than DigitalOcean per GB of RAM.”

The ultimate indignity: your product compared to a VPS unfavorably.

Value Axis



Act III: Wrangling Complexity

Work from first principles.

You're not Heroku.
Or Stripe.
Or GitHub.
Or AWS.
Or RedHat.
Or Slack.

Don’t cargo-cult
your business model.

Example time!

(Adoption) Market Strategy

get lots of individual users

get many of them to pay a small amount

these people will love and feel invested in Heroku

they will help drive adoption in business contexts

Similar models

GitHub

DropBox

Google Apps

Slack

Project Scope Directives

make one decisive change

constrain scope to dynos (if possible)

don’t put us out of business

constrain (most) scope to self-serve audience

Early example goals

Dyno Tiers

Goals:

+ Introduce competitive price point for individual developers
+ Remove “free dyno” distraction from co-prime sales
+ Create market perception of better value, price decrease
+ Articulate product value more through features than resources
+ Separate personal & production use cases distinctly
+ Avoid large short- (or long-term) revenue losses
+ Reduce free tier gaming
+ (WIP) Avoid perception we are increasing prices (HN rage)
+ (WIP) Reduce gap between personal usage and prod. usage
+ (WIP) Encourage best practices in prod (2+ dynos)

Dyno Tiers

Non-Goals / Future Work:

+ Fix SSL pricing
+ Introduce application based pricing / bundling
+ Align add-ons or other product areas to this pricing
+ Offer higher-price enterprise offerings

Early project proposal

Positive Customer Perception/ Pricing Change

Negative Customer Perception/ Pricing Change

Deltas: Dyno Tiers vs Current Dynos

Old	Dyno Credit	1X	1X	2X	PX
New	**Free Tier**	**Basic**	**Standard**	**Prod**	**Performance**
Duty Cycle	750 hrs/mo **350 hrs/mo**	750 hrs/mo 750 hrs/mo			
System Resources	512MB **512MB**	512MB **512MB**	512MB **512 MB**	1024MB **1024MB**	6GB **6GB**
Custom Domains	Yes **No**	Yes **Yes**			
Scale out	No **No**	Yes **No**	Yes **Yes**		
Scale up	Yes **No**			Yes **Yes**	
Custom SSL	Heroku app SSL **Heroku app SSL**	$20/mo **$20/mo**		$20/mo **Free**	
Price	Free **Free**	$35 **$7**	$35 **$25**	$73 **$50**	$576 **$500**
Goal Alignment	Customers can no longer run free tier 24x7. Less distraction for sales.	Basic fills a need for hobbyists, expands top of the funnel	Tricky: Customers with a mix of 1X and 2X/PX dynos will have to choose whether to go up or down.	Price cut for the majority of revenue producing customers. Introduce SLA. Enable single dyno apps.	Price cut for our best customers with most demanding apps.

All of this continued to evolve.

“You should solve [whichever problem]!”

“Okay… are the **goals** or the **proposal** wrong? Does the proposal **break** something **important**?”

Act IV: Contact

Modeling

SaaS is great: you have so much data

Built a succession of detailed models.

High-level price sensitivity.

Per-customer migration behavior.

First Contact

Informal interviews of friendly users and friends.

Good for practicing talking about the change.

Good for gut checking broad strokes & huge missteps.

Biased by past experience with product.

Tend to be invested already & give benefit of doubt.

Haters are never wrong!

The burden is on you to communicate,
not the world to understand.

Alpha

Responded to survey & informal feedback.

Small pilot by invite only.

Minimize cost of pivots.

Target different cohorts with different outcomes.

Expected leaks hinted at market response.

Beta

Responded to Alpha feedback

(simplify dynos / include domains on free)

Optional for all users.

New users received new pricing by default.

Existing users could port over.

GA

Responded to beta feedback.

(16 hours on free dynos, other tweaks)

All new users and apps only get new pricing.

Existing apps can migrate per preference.

Grandfathering and migration details announced.

Details finalized*.

Grandfathering & Migration

(Mostly) let users migrate themselves.

Encourage migration by gradual pressure.

Observe market response.

Finish migration & retire old dynos end of January.

Act V: Conclusions

What went well

Huge increase in new customers per month.

Large customers growing better.

Differentiated price points for individuals/hobbyists.

Product is now priced along value axis.

Isolate product complexity to professional segment.

What went poorly

Took a very long time. (Still ongoing!)

Purchasing experience confusing.

Organizationally painful to implement.

Third major attempt at this project.

Gaming reduced… but new gaming started.

Conclusion

Pricing **defines** product.

Have **explicit** strategy.

Approach **pricing** problems
as product **design** problems.

Support with **data**.
Gather diverse **feedback**,
model changes, and **adapt**.

Your pricing will
never be **perfect**.

fin

Bonus Time: Multiplication

“1X dynos cost $35/mo.”

— every developer

“Hobby dynos cost $84/yr.”

— every developer

Developers will multiply numbers together once.


WELP

fin

Questions!

What about interactions with enterprise sales?

But, but, but: Slack!??!!?

How does this change for licensed software?

There are only four of us, and I'm in charge!

pvh@pvh.ca / @pvh