

Welcome!

Hotels.com

From Fish To Fishing Rod

The Hotels.com® DevOps Journey



Chris Rice, Technology Director (Platform)
@iamchrisrice

Hotels.com

About Me – Chris Rice (@iamchrisrice)

- Working in Technology for 17 years
- Software development background
- At Hotels.com since 2011
- Led multiple feature development teams
- Moved to Platform group last year



About Hotels.com®



Global Business:

- Established in 1991 as the Hotels Reservations Network
- Part of Expedia Inc.
- Operating in 85 countries and 34 languages today

Global Technology team:

- More than 500 engineers
- More than 80 engineering teams
- Spread across 9 locations and 6 time zones

What does DevOps mean to Hotels.com®?

Hotels.com

There is no one-size-fits-all



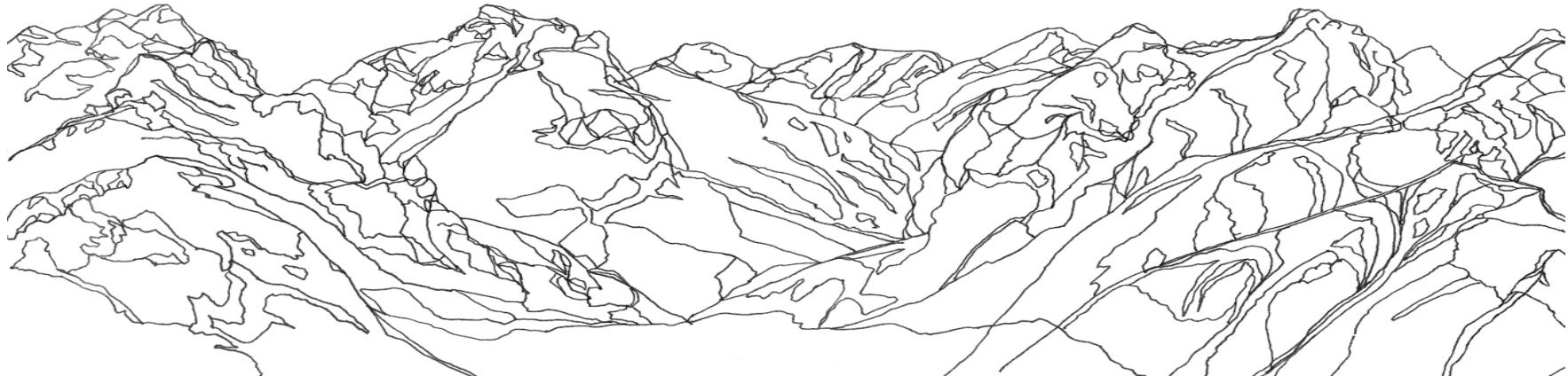
Our Context

Large, distributed technology team split across multiple locations

Smaller, centralised operations team based largely in London

Software deployed to primarily physical infrastructure

Number of compliance constraints (Sarbanes-Oxley, PCI-DSS etc.)



Our Desired Outcomes

Engineering teams should feel **empowered** and **responsible**

Reduction in **cycle time*** for feature delivery

Increase in **technical** and **product innovation**

Reduced **friction** and improved **developer experience**

Greater software **reliability** and **resiliency**

Remove **over-reliance** on smaller, central operations team

* defined as time between start of feature development to customer delivery

Where we came from

If you do not know where you come from, then
you don't know where you are, and if you don't
know where you are, then you don't know
where you're going.

- Terry Pratchett

Hotels.com

October 2011

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Hotels.com® Development in 2011

Monolithic Java web application

Multiple teams contributing to its development – no clear ownership or responsibility

Dreaded merge before release

Release cadence typically weeks or months



Hotels.com[®] Operations in 2011

No common automation for release of software

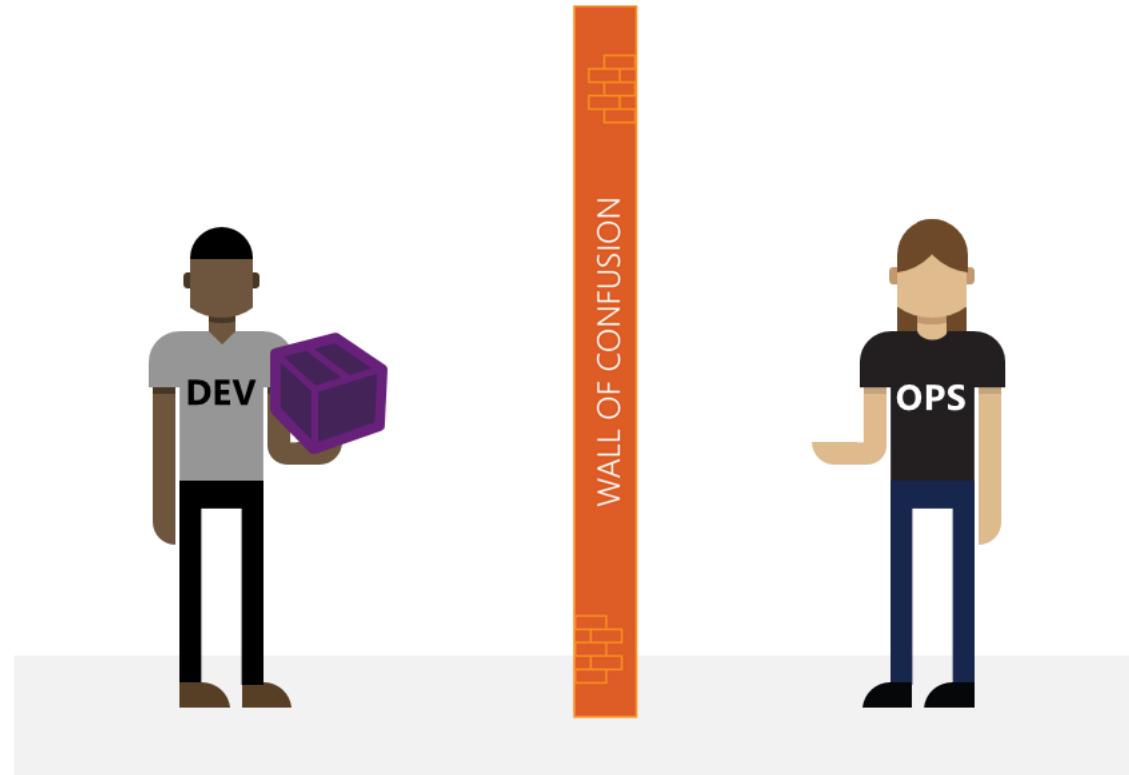
No source control in place for scripts and tools

Typically a team of reactive fire-fighters

Little-to-no collaboration with development teams



Hotels.com® State of DevOps in 2011



Where we went

Hotels.com

Breaking Apart The Monolith



Splitting into **multiple applications**

Giving individual teams or groups **ownership** of code base

Elimination of multi-team **merge** before release

First step toward **independent** release of applications

More **complexity** due to increased number of applications

Improving Release Cadence



Opt-in **release vehicles** scheduled
every two weeks

Manual Ops To Automation



A N S I B L E

First steps toward **automating** our infrastructure put in place

Ansible playbooks replacing ad hoc scripting for common tasks

Playbooks committed to and maintained via **source control**

Where we are

Hotels.com

Shipping Containers



Feature teams delivering containers rather than snowflake applications

Huge simplification of tooling

Allows teams greater room to innovate with technology choices – as long as it runs in a container, it can run on our platform...

Self Service Automation



Shift from ticket-driven operations team to providers of multiple platform capabilities

Capabilities made available to development teams via **self-service interfaces** and **automation**

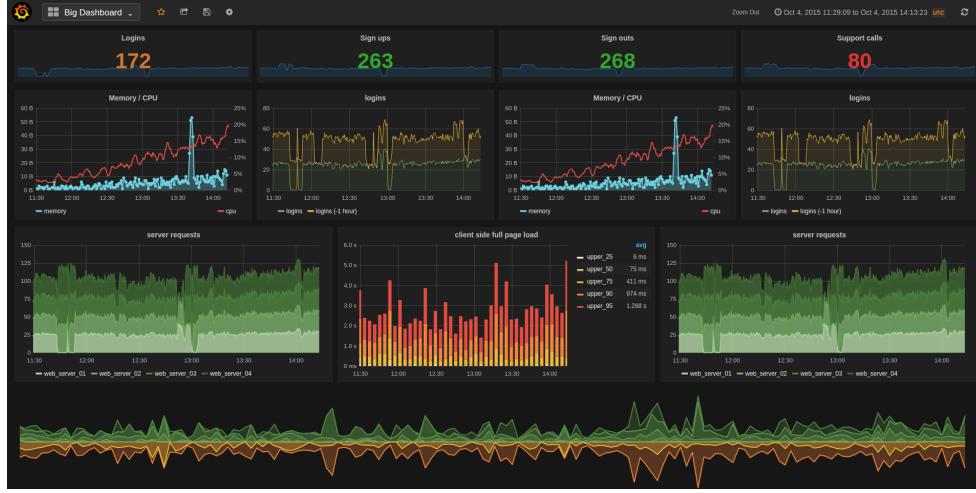
Deployment Pipelines



Capability for feature teams to **build** and **release** their software to multiple environments on demand

Canary releases to enable verification of new releases with real customer traffic

Dashboards And Alerting



Dashboard infrastructure built out to enable teams to visualize metrics about application health and performance

Alerting capabilities to enable teams to quickly detect problems

Champions



Community of champions established to work closely with platform teams

Acting as customer representatives and local expertise on platform topics

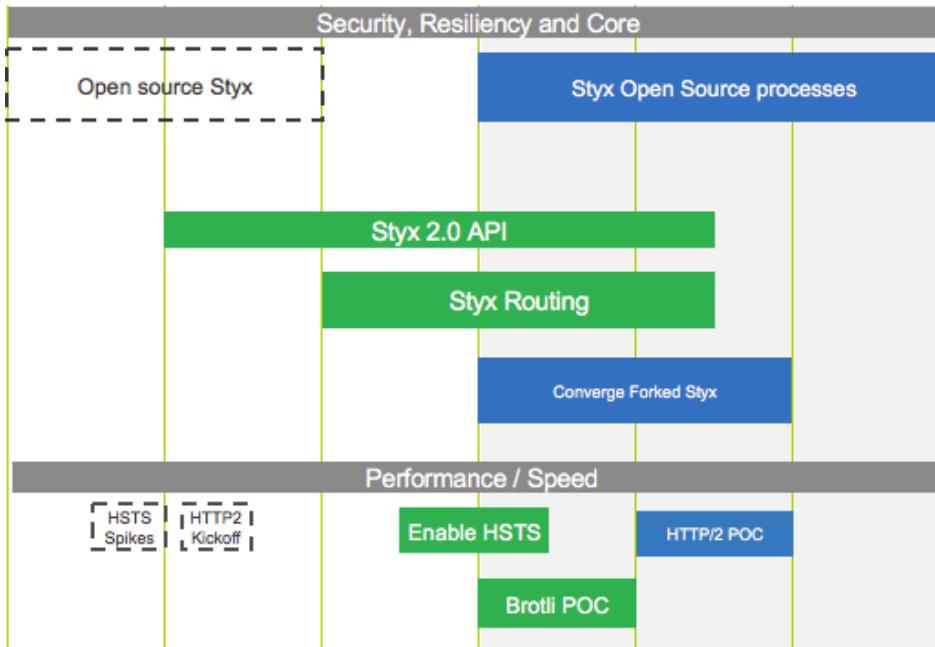
Building empathy with platform and operations teams

Helping steer future direction

Where we're going

Hotels.com

Agile Delivery and Product Management



Formation of Agile Coaching Team within Platform to help drive best practices around delivery and product management

Teams publishing roadmaps reflecting planned work

Surfacing team metrics around planned vs. unplanned work balance

Helping solicit customer satisfaction feedback

Leaving The Data Centre



Major project in flight to move out of the data centre and into **AWS**

Taking advantage of previous investment in containers by leveraging **Kubernetes** for resource scheduling

Cloud infrastructure configuration expressed **as code** and committed to source control

Going Big on Self Service

Chris Rice · 11:02
argo help

Flyte · argo · 11:02

Hey @ChrisRl argo currently support the following:

- argo artifact get [environment] [component] : Get the current artifact and metadata
- argo deploy [environment] [component] [version] [CHG ref - optional, required for prod] : Deploy any version of an app. When production, please specify your change request after version e.g. CHG01234567
- argo redeploy [environment] [component] [CHG ref - optional, required for prod] : Redeploy the current version of an app. When production, please specify your CHG ref as per deploy.
- argo rollback [environment] [component] [CHG ref - optional, required for prod] : Rolls back to the previous version - note it won't roll back runtime config, so you must do this if needed! When production, please specify your CHG ref as per deploy.
- argo kill deploy [environment] [component] : Aborts a deploy if it is running.
- argo instance logs [environment] [component] [index] : get the link to the instance logs in yer browser
- argo status [environment] [component] : get the deploy status for a component
- argo instance get [environment] [component] [index] : return information about the given instance
- argo instances [environment] [component] : get a list of instances
- argo component metadata put [environment] [component] [key] [value] : adds metadata to a component
- argo server inventory [server] : list all instances on a given server
- argo who deployed [environment] [component] : ask who triggered the deploy of something

Missing something you need? Have a chat with your pipeline champ or hop onto the HCOM Self Service Platform room!

Investment in self-service automation platform, Flyte, to trivialise exposure of capabilities

Chat Ops used at primary interface into Flyte automation

Flyte to be Open Sourced soon!

From Fish To Fishing Rod

People love to say, "Give a man a fish, and he'll eat for a day. Teach a man to fish, and he'll eat for a lifetime."

What they don't say is, "And it would be nice if you gave him a fishing rod." That's the part of the analogy that's missing.

- Trevor Noah

Summary

Hotels.com

Our Desired Outcomes

Engineering teams should feel **empowered** and **responsible**

Reduction in **cycle time** for feature delivery

Increase in **technical** and **product innovation**

Reduced **friction** and improved **developer experience**

Greater software **reliability** and **resiliency**

Remove **over-reliance** on smaller, central operations team

Summary: DevOps Journey Considerations

There is no **one-size-fits-all** for DevOps – consider your **context** and the **outcomes** you desire, and take small, achievable steps in the right direction

Not an overnight transformation, but a long and sometimes challenging **journey** involving both development and operations functions

Continually **assess** where you are in your journey – you may need to make adjustments

Introduce tools to support your objectives

One more thing...

Hotels.com

Open Source @ Hotels.com

Hotels.com Technology open sourced its largest project yet last week...

- **Styx** – a programmable, asynchronous, event-based reverse proxy for the JVM.
- Used by multiple Expedia Inc. brands, servicing well over **1 billion** requests per month

More projects to follow, including **Flyte** - the framework underpinning our **self-service** efforts.

Check us out at:

<https://github.com/HotelsDotCom/>

Thank You

P.S. We're hiring: <https://lifeatexpedia.com/>

Hotels.com