# Put the "Ops" in "Dev"

What Developers Need to Know About DevOps

OPEN SOURCE LAB

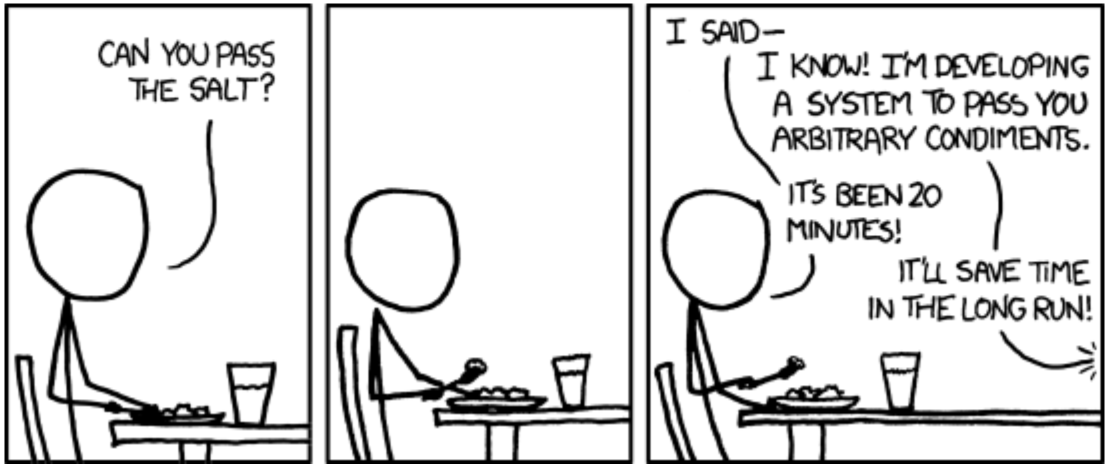
# Introductions

Lance Albertson
Rudy Grigar
Ken Lett
Greg Lund-Chaix

source: http://shesawake.com/

# A show of hands if you...

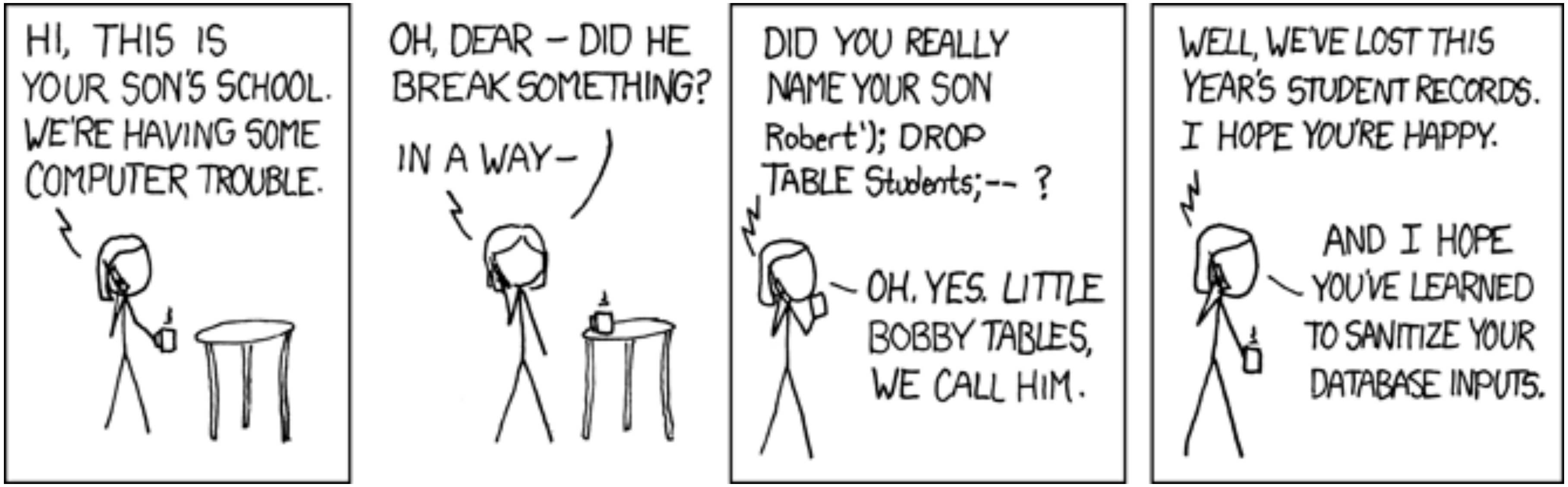
- develop?
- operate?
- devoperate?*


CAN YOU PASS THE SALT?
I SAID—
I KNOW! I'M DEVELOPING A SYSTEM TO PASS YOU ARBITRARY CONDIMENTS.
IT'S BEEN 20 MINUTES!
IT'LL SAVE TIME IN THE LONG RUN!

HTTP://XKCD.COM/974/

* Already consider yourself a DevOp

# Goals for this Talk


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

HTTP://XKCD.COM/327/

## Topics

**Ken**

How knowing your infrastructure helps you develop and test your code.

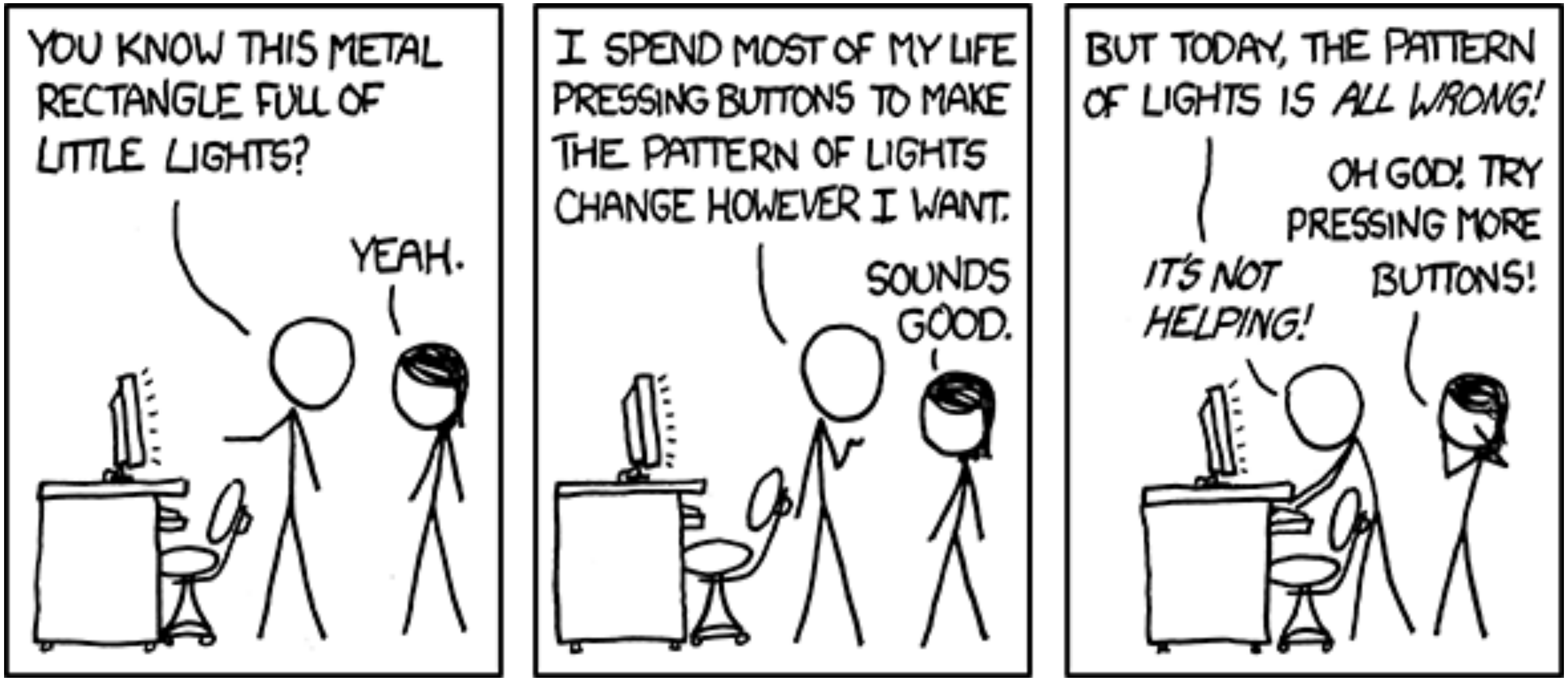
**Rudy**

Getting ready for deployment, testing changes, and release strategy.

**Greg**

The difference between development and production environments.

# Are SysOps and Devs really any different?


YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?
YEAH.
I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.
SOUNDS GOOD.
BUT TODAY, THE PATTERN OF LIGHTS IS *ALL WRONG!*
*IT'S NOT HELPING!*
OH GOD! TRY PRESSING MORE BUTTONS!

HTTP://XKCD.COM/722/

## Know your Infrastructure

*A developer should know:*

- Where will my code live?
- What resources will it need?
- How does it get deployed?
- How does it get maintained?

*In other words...*

**Ops**

## Know your Infrastructure

*Knowing the system in which your code lives means you can write better code.*

- What is the bandwidth situation?
- What is the storage situation?
- How slow is this going to be, anyway?
- Can I use a CDN?
- Can I cache it? On disk? In memory?

## Know your Infrastructure

*Knowing the system on which your code runs means you can plan.*

- What version of PHP? Drupal?
- What libraries are available?
- What tools are available?
- What versions did you say again?
- Seriously, PHP is still 5.2?

## Know your Infrastructure

*Knowing the system in which your code lives means you can coordinate maintenance.*

- They want to upgrade that now?!
- There's a security hole in what?
- Did anyone happen to back up that database yesterday?

## Know your Infrastructure

*Knowing the system in which your code lives means you can develop and test your code safely.*

- Does your dev environment match production?
- Does your test environment?
- You do have dev and test environments, right?

## Own your Infrastructure

*The fastest way to know your infrastructure is to build it.*

- Virtual Machines make it easy (if you know Ops)
- VirtualBox, Vagrant, Vmware, Parallels are your friends
- Configuration management tools make you the sysadmin

## Own your Infrastructure

*Virtual Machines allow you to replicate your target environment.*

- Virtual Machines are disposable
- Build machines that replicate your production systems
- Use the same tools that Ops uses
- Make lots of boxes
- Destroy them all!

## Be a DevOp

*It's just better that way.*

## Getting Ready for Deployment

How frequently are you releasing?

How difficult are upgrades?

How do you know when things break?

## Getting Ready for Deployment

*Know your upgrade path and document it to save headaches.*

- Reduce your moving parts:
  - Dependencies required, etc.
- Database schema changes:
  - Backward compatibility? (Drupal upgrades)
- Goal:
  - Decouple all the things! (To enable incremental changes that can be easily tested)

## Getting Ready for Deployment

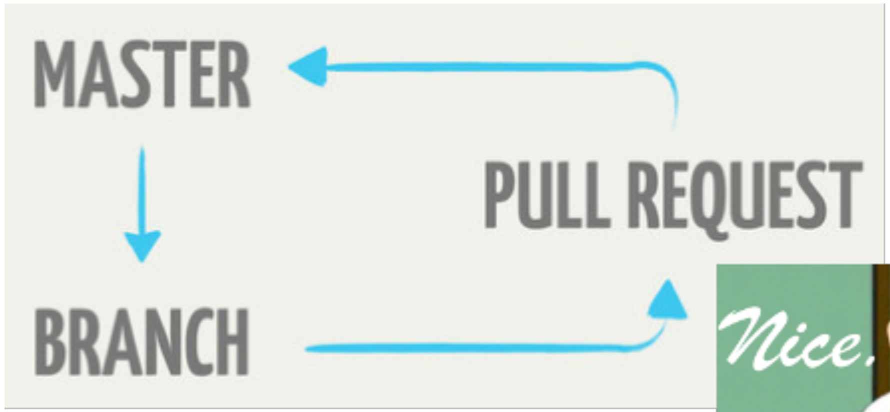
*When do we really need to upgrade software?*

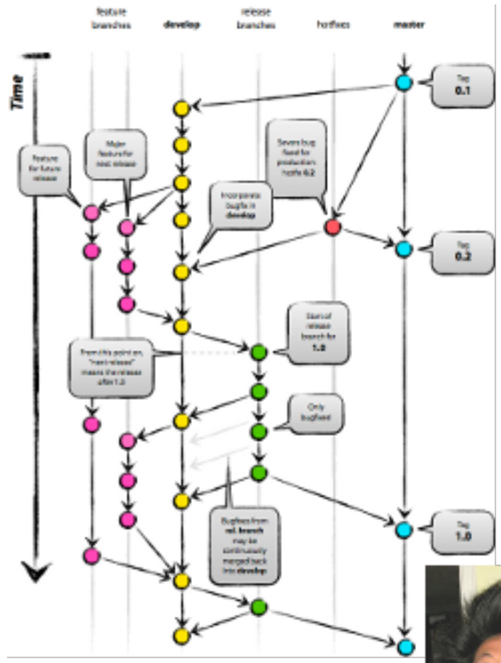
- Security fixes
- Critical bugs
- New features
- Remember: Keep it simple

## Getting Ready for Deployment

*A simple development process may lead to a simple upgrade process.*

K.I.S.S.





## Getting Ready for Deployment

*Drush is your friend.*

- Do you really need packages?
  - webapps: probably not, unless...
  - drupal: drush!
  - everything else: probably
- Drush can automate many pieces of the development and deployment process.
  - sql-sync
  - runserver and qd (core-quick-drupal)
  - rsync

## Getting Ready for Deployment

*Are you continuously testing?*

- Jenkins-CI and drush+SimpleTest.
- Continuously test your software.
    - Fixed a bug?  Don't forget to add a test to confirm it doesn't break again.
- Ops: We should test our infrastructure too.
    - Does my web server VM successfully launch with httpd serving content?
    - Does my database server start mysqld with the proper configuration?

## Going from dev to production

http://www.flickr.com/photos/dlr_de/5475486539

- Common development environments:
  - Small VPS (1-2 proc, 256-512MB RAM, 10-20G virtual disk)
  - Laptop with XAMPP or Vagrant
  - Full stack on one machine or VM
- Common production environment:
  - Load balancer
  - Multiple webnodes
  - Proxy cache (Varnish, Squid)
  - Internal cache (Memcache, APC)
  - Database cluster
  - Network storage (SAN, NFS)

# Avoiding pitfalls

- Agnosticism
- HTTP daemons
- Databases
  - Clusters,
  - Split read/write
- Cache
  - Reverse proxy - Varnish, Squid
    - Set those HTTP headers!
    - Be careful with cookies
  - Internal - Memcache, APC
    - Shared or individual?
    - Keys

http://www.flickr.com/photos/baggis/3860802929

## Avoiding pitfalls

- Load balancing & clusters
  - Session management
  - Shared, non-local storage
- Scaling
  - What happens when someone deploys 1,000 copies on a server?
  - What happens when 10,000 users hit it all at once?
- Multisite vs. single
  - Don't assume /sites/default or /sites/all

http://www.flickr.com/photos/baggis/3860802929

## Don't hack core!

No, really.

Don't.

*Think. Of. The.*
***Kittens.***

*Please!*

http://www.flickr.com/photos/gideonvanderstelt/3833757689

# Questions? Flames? Angry Mobs?

Lance Albertson - lance@osuosl.org - @ramereth
Rudy Grigar - rudy@osuosl.org - @basic_
Ken Lett - kennric@osuosl.org - @KenLett
Greg Lund-Chaix - gchaix@squishymedia.com - @gchaix

***http://osuosl.org***

*Follow OSUOSL*

@osuosl | fb.com/OSUOSL

G+ "Open Source Lab"