

A Good Start

Or How to Not Fail At Ops

[#dickpicsforfreedom](#)

LOPSA



WARNING

We are not speaking on behalf of our employer. All content you see here in no way reflects on our employer. They have not reviewed nor authorized any of this content.

Opinions shared are just that. Do not trust anything we say without verifying it for yourselves!

THE RULES

1. DO NOT AUTOMATE WHAT YOU DO NOT UNDERSTAND
2. DO NOT AUTOMATE WHAT YOU DO NOT UNDERSTAND!
3. YOU ARE THE LAST LINE OF DEFENSE FOR PRODUCTION

What Is Automation?

Taking human fingers off of
keyboards

Why Automate?

Consistent, reproducible environments

Reality Check

These Truths Might Be Difficult For Some
People

Reality Check...

There are no silver bullets,
not even automation.

Reality Check...

You are not a unique
snowflake.

Reality Check...

Do not reinvent the wheel.

Reality Check...

Complexity for its own sake is
a bad thing.

Reality Check...

Good ops skill never
goes out of style.

Reality Check...

This will be hard, and
take time...

Reality Check...

... but it will enable you and
your company to adapt
successfully

When Should We ...

- Use Source Code Management (SCM)?
- Manage your configuration in SCM?
- Manage your build artifacts?

Bootstrap

Get Your Bits On (The Disk)
Get It Up (On The Network)

Bad Idea Theatre

“I will build each one by hand!”

You will need...

- A bunch of optical drives
- Copies of the OS on DVDs
- A stack of pizzas
- Sons of Anarchy on Blu-Ray
 - You'll Need All Seven Seasons

What you will get...

Each node will come up *slightly*
differently

Good Idea Theatre

*“I will install off of the local network
using a pre-programmed configuration.”*

You will need...

- A functioning network
- A Basic Server
- A Thumb Drive containing:
 - OS Image
 - Kickstart file

A Word About Installation Scripts...

Only put code in here that you absolutely need to get the bits on the disk, the system on the network, and the config management system up.

Nothing more.

Single Rack

Caring For Systems

You Can Not Buy More Time

You should always be able to easily re-provision any system from scratch at any time

Hardware Fails

- How much will failure cost you?
- Could you have planned for it?
- Sometimes failure isn't a complete system failure

Software Fails

- “Bad” release
 - untested changes lead to unexpected behavior
 - broken dependencies
- Stupid Developer Tricks
 - local environment variables
 - sub-optimal artifact deployment
 - zip file + shell script

Fix or Replace?

- You should know the time cost of replacing a machine
- How much time do you have to investigate?
- Is replacement an option?

Configuration Management

Your server is too
complicated to remember
how you built it.

Configuration Management

Don't join a cult

but if you do...

Benefits of CM

*Consistently
reproducible
configuration*

Configuration Management

Only make changes in
your CM system

Single Datacenter

Caring About the Whole Application

Working With A Team

- Documentation
- Run Books
- Architecture

Keeping an Eye on Things

- Collecting Metrics
- Monitoring for Alerting



**Y'all motherfuckers
need metrics!**

Monitoring and Metrics

- **LOOK AT YOUR GRAPHS DAILY!**
- Metrics (**informative**)
 - Useful primarily in aggregate
 - Graphed over time
- Trending
 - Analysis of metrics over time
 - Identifying patterns
 - Measuring Distributed Databases Across the Globe

Intelligence and Metrics

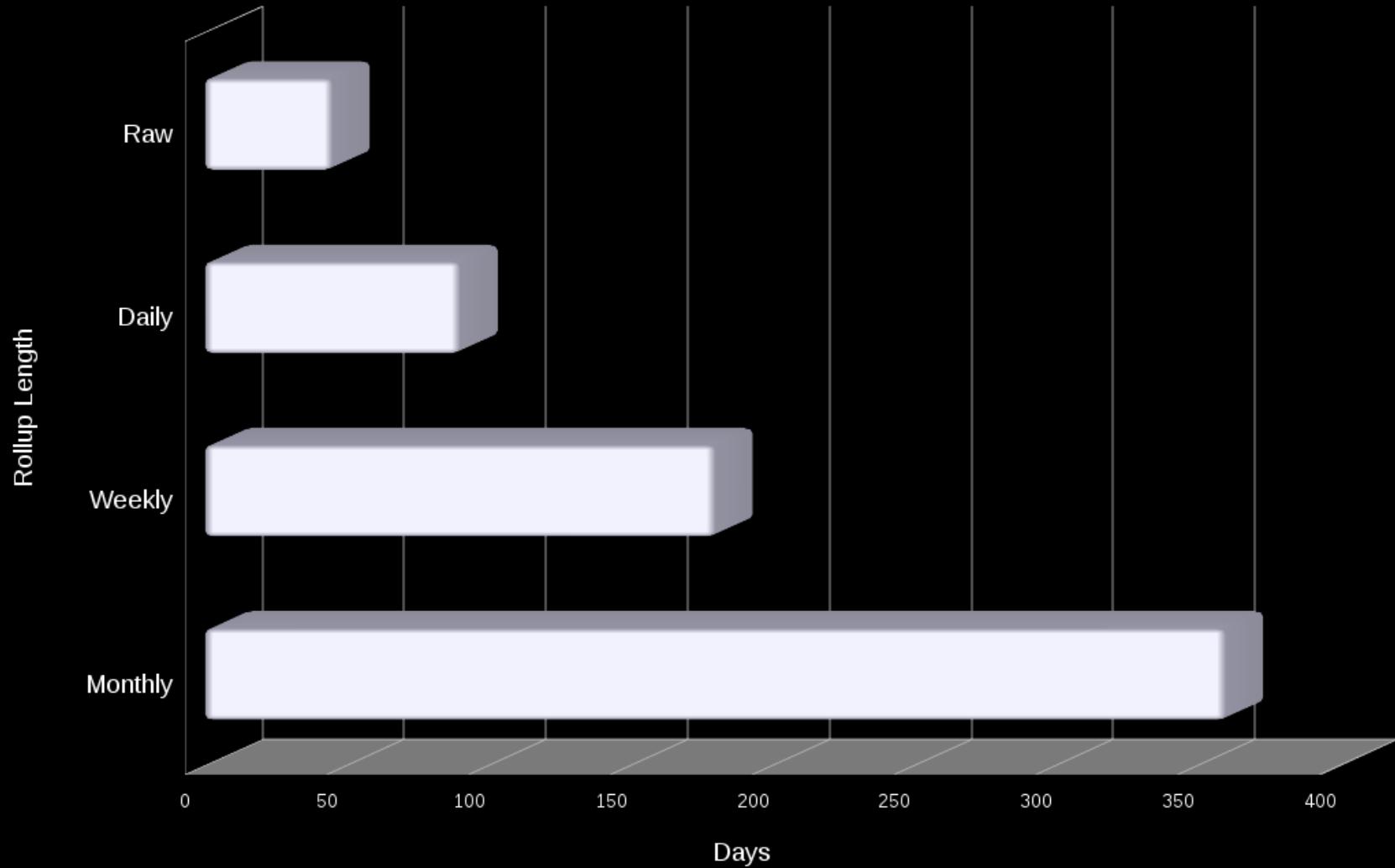
- **Metrics:** More than one metric recorded over time.
- **Data:** composed of many metrics.
- **Intelligence:** What you discern from the data.
 - AKA: Trending
 - Shapes in graphs
 - Patterns

Intelligence and Metrics

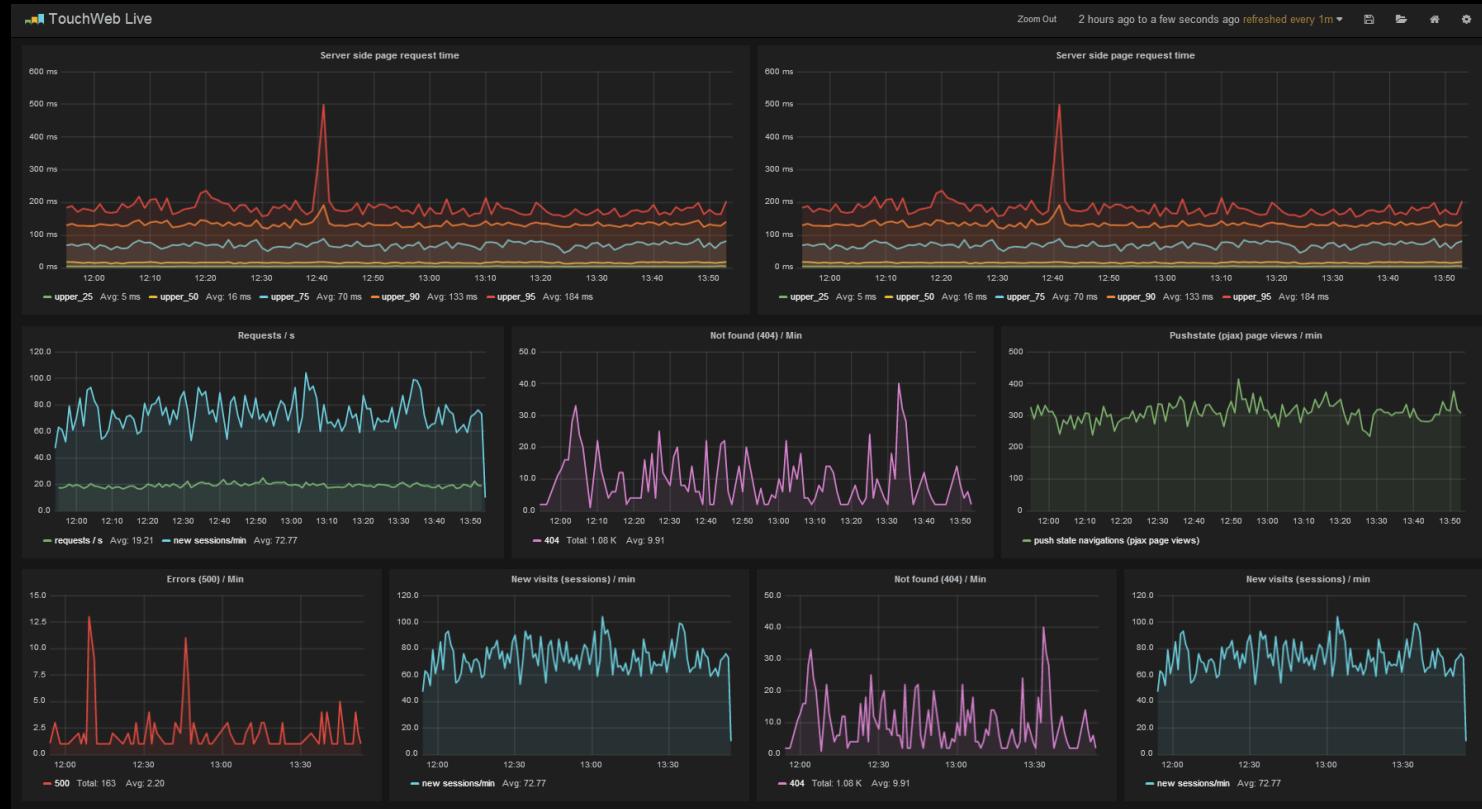
- Basic alerting thresholds
- Overlaying events over time (annotations)
lead to better understanding
- Bigger ideas grow from collected data:
 - **capacity planning**
 - **impact of operations**
 - **traffic behavior**

Storing Metrics

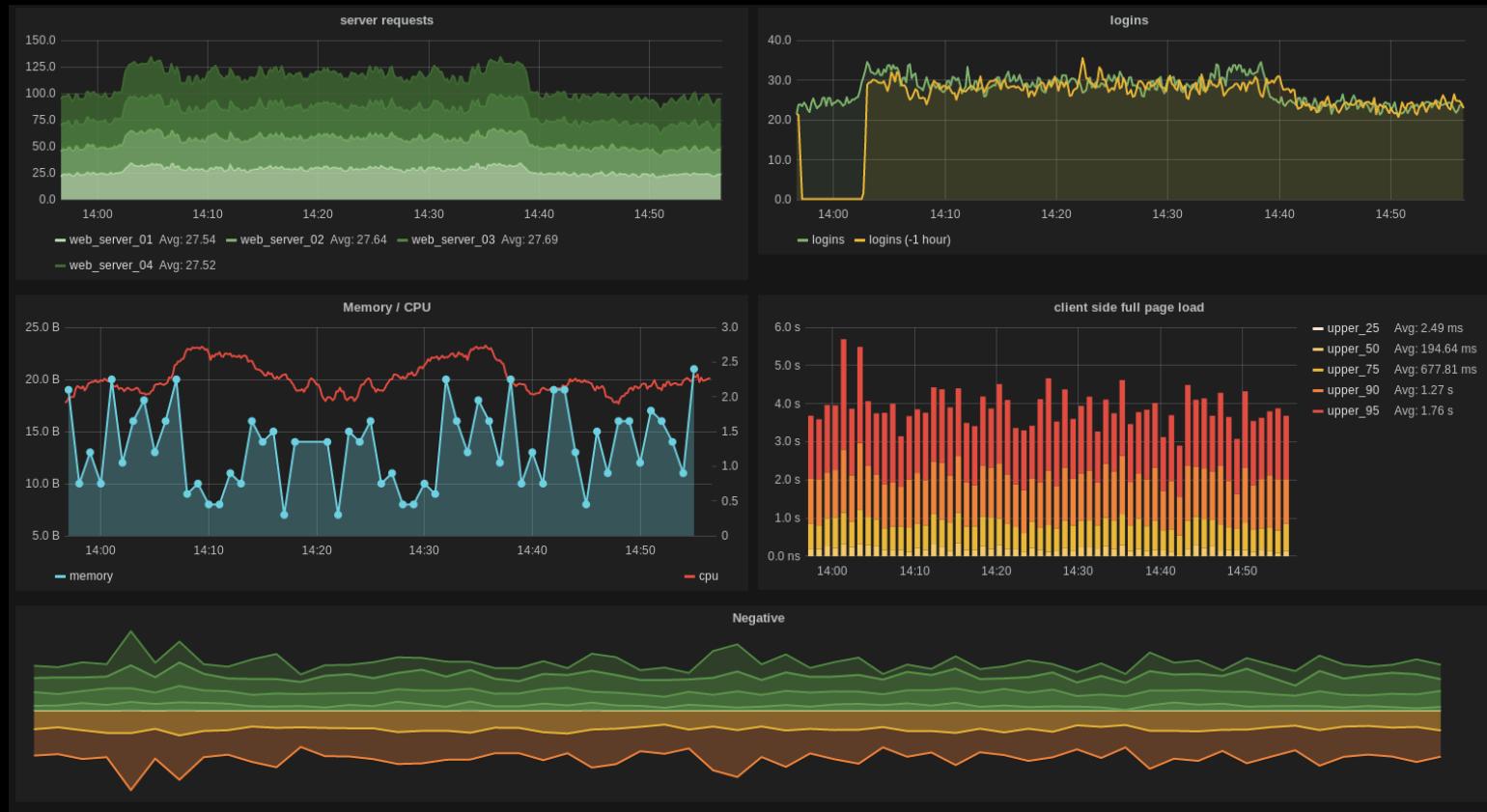
- Storing metric data is cheap
 - Reading it back is expensive
- What are you willing to store?
- How detailed do you need the data?
- For how long?
 - 30 days?
 - 6 months?
 - 1 year?



Graphana Example



Graphana Example



Monitoring and Alerting

- **What** do you alert on?
 - **Informative** and **actionable** problems!
- **When** do you alert?
- **To whom** do you alert?
 - Based on impact and criticality
- **How** do you alert?
 - Cell phone if it's P1, email otherwise.

Monitoring and Alerting

- Alert Priority Should Guide Timeframe
- Page only the most important issues

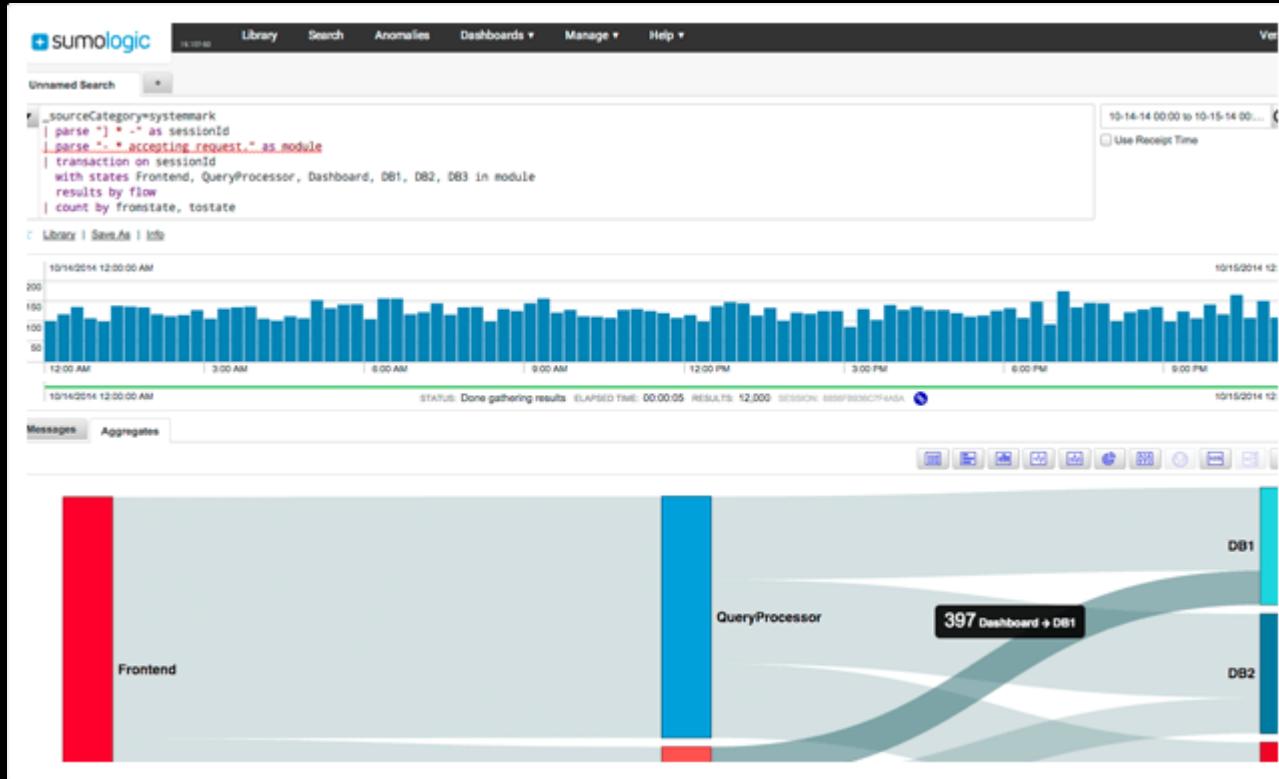
Centralized Alerting

All of your alerts in one place.

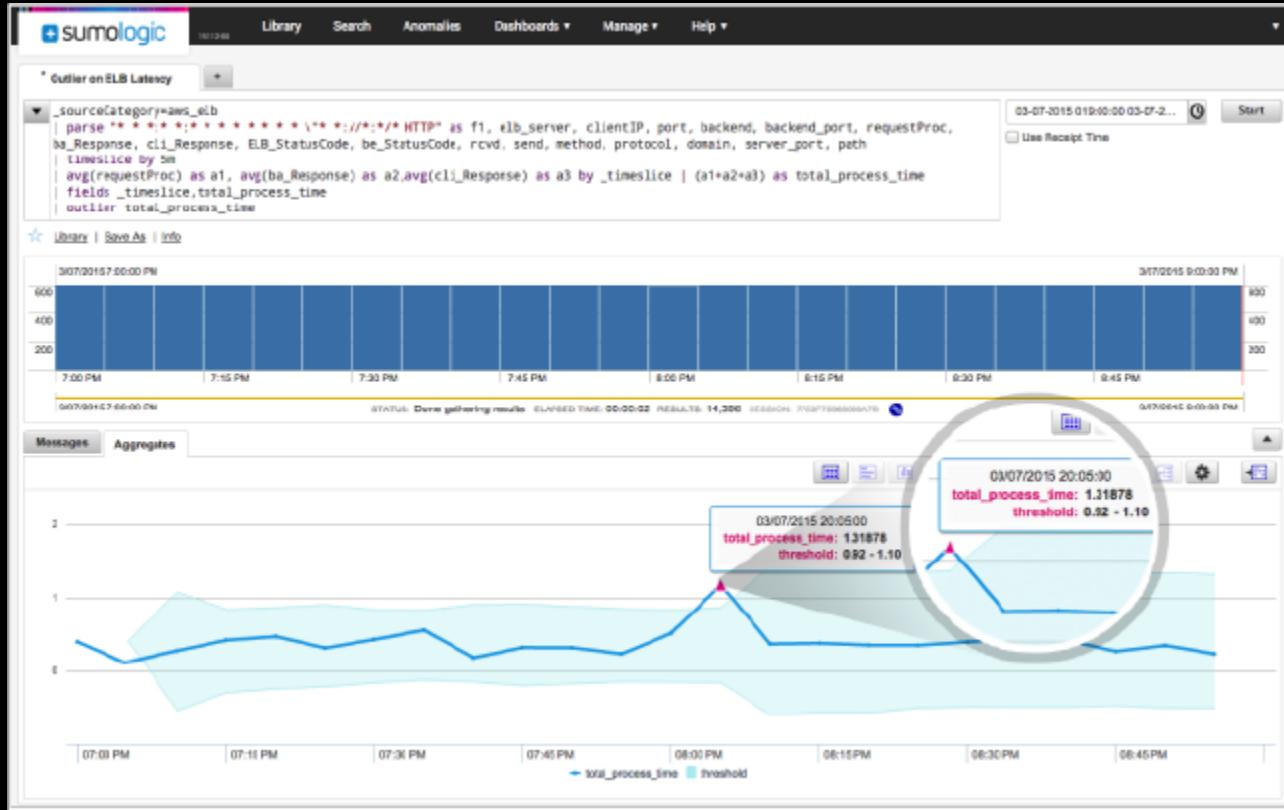
Centralized Logging

All of your logs in one place.

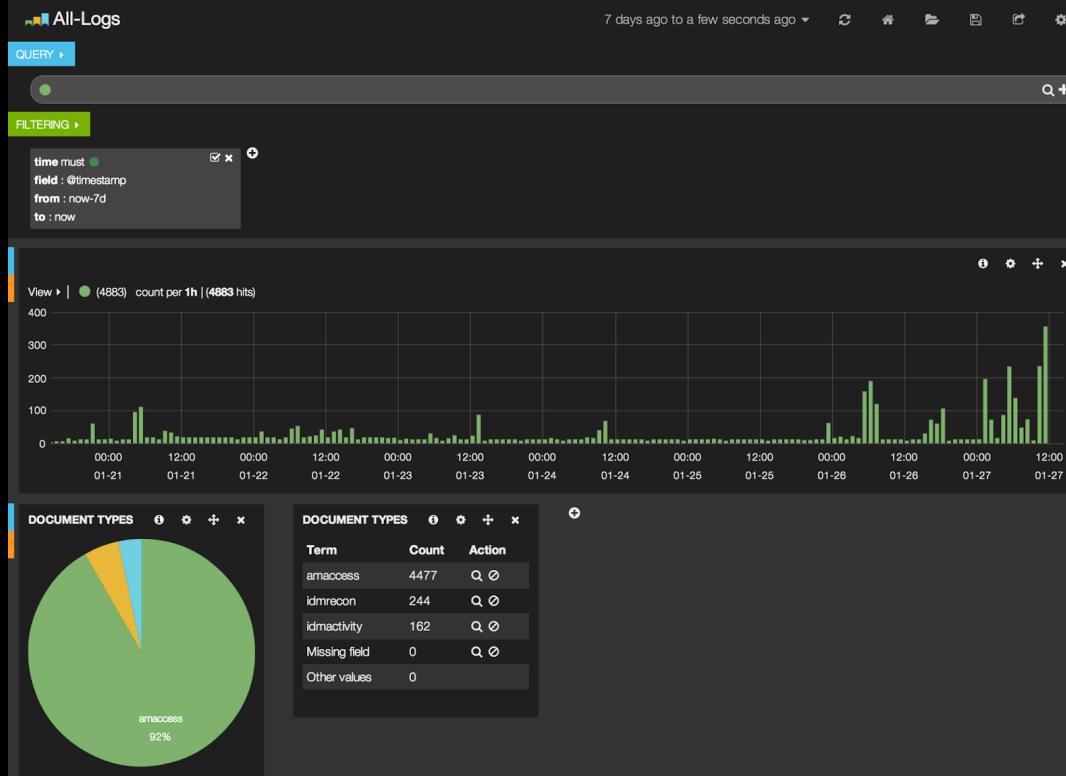
Sumologic Example



Sumologic



Logstash Example



“Basic” Loggins

rsyslog / syslog-ng



Advanced Loggins

- GUI
- Search
- Reporting
- Graphs
- Alerts



Force Multipliers

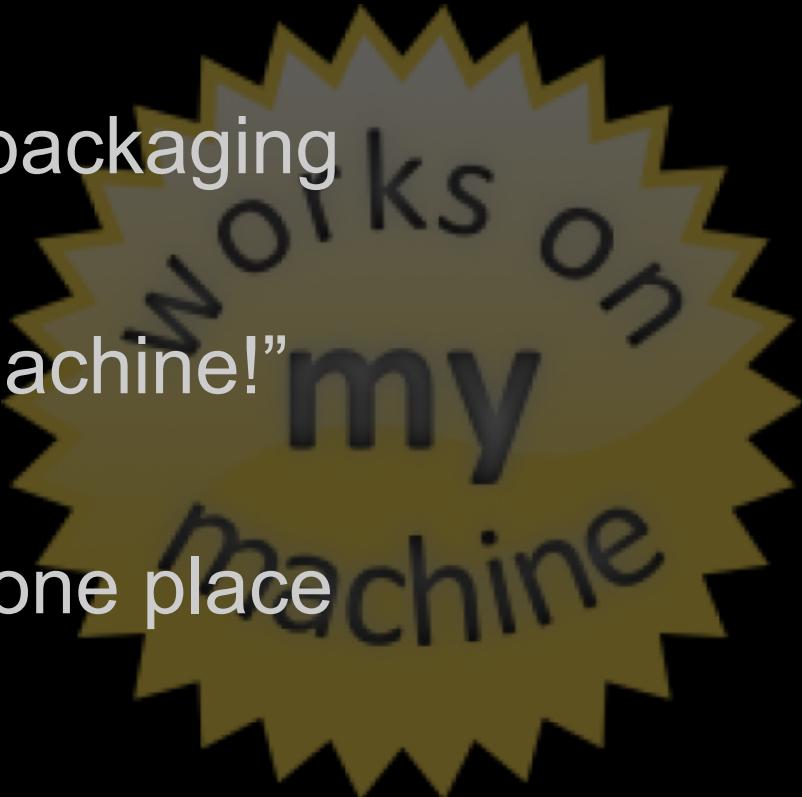
You Have Room For Improvement

Packaging

- Tarballs are not a deployment artifact
- Manage source RPMs
- Make packaging part of your build process
- Don't create package gurus
 - Your team should be able to hit the 'package' button

Build Servers

- Clearly define your build + packaging pipeline
- Prevent “It worked on my machine!”
- Observe the process
- Put all of your packages in one place



Proxies & Caches

- Thou shalt not hammer free software repositories as part of your automation!
- If you rely on a package repository to do business, mirror or at least cache that repo
- Consider your own local mirrors
 - Your builds can go here too!

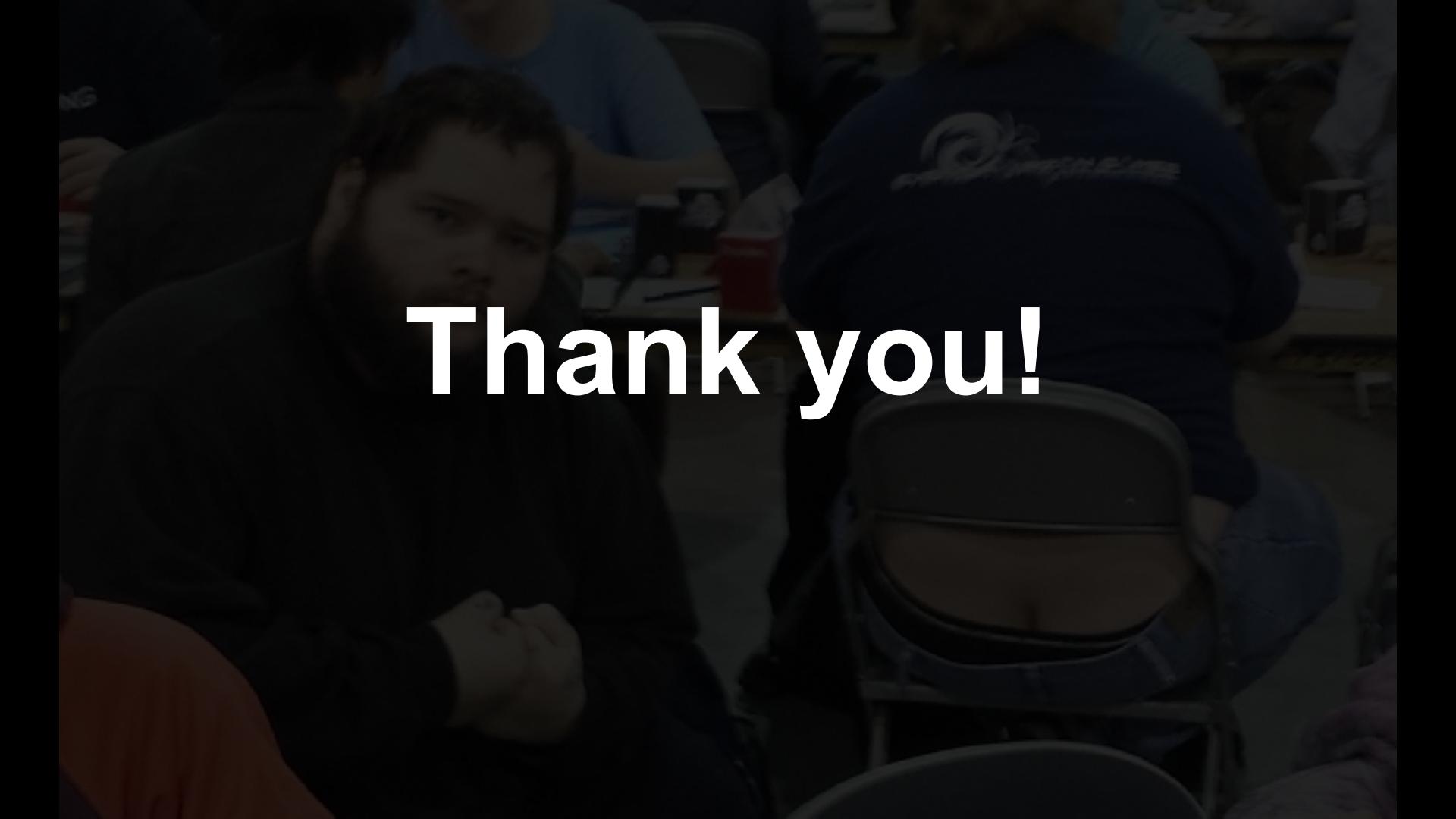
Questions?

Contact Information

Bobby - @bobbylikeslinux

Materials: <https://github.com/vectorsigma/opstalk-materials>

#swift on freenode.net

A dark, grainy photograph of a group of people at a social gathering. In the foreground, a person's hand is visible holding a large, light-colored cup. In the background, several other people are visible, some wearing hats and jackets. The overall atmosphere is casual and suggests a social event.

Thank you!