



# **Under the dome (of failure driven pipeline)**

Maciej Lasyk

4developers – Warsaw

2015-04-20



Join Fedora Infrastructure!

- learn Ansible
- learn Docker with Fedora Dockerfiles

<http://fedoraproject.org/en/join-fedora>

Agenda?

Don't run away ;)

[...]

Situations like this only reinforce my deep suspicion of developers: They're often carelessly breaking things and then disappearing, leaving Operations to clean up the Mess.

[...]

“The Phoenix Project”

by Gene Kim, Kevin Behr and George Spafford



software developers are



software developers are **idiots**

software developers are **arrogant**

**how many** software developers are **there**

**how many** software developers are **in the world**

Press Enter to search



software developers are



software developers are **idiots**

software developers are **arrogant**

**how many** software developers are **there**

**how many** software developers are **in the world**

Press Enter to search



sysadmins are



sysadmins are **freaky**

sysadmins are **like firemen and cops**

Press Enter to search.



## **Conway's law (1968)**

organizations which design systems ... are  
constrained to produce designs which are copies  
of the communication structures of these  
organizations

[http://en.wikipedia.org/wiki/Conway%27s\\_law](http://en.wikipedia.org/wiki/Conway%27s_law)



## **Ruth Malan (2008)**

if the architecture of the system and the architecture of the organization are at odds, the architecture of the organization wins.

The organizational divides are going to drive the true seams in the system.

Yup, you're gut is telling truth...

Yup, you're gut is telling truth...

This will be another devops indoctrination

Yup, you're gut is telling truth...

This will be another devops indoctrination

What did you expect? ;)

This presentation includes gentle product placement

Yup, you're gut is telling truth...

This will be another devops indoctrination

What did you expect? ;)



This presentation includes gentle product placement

Yup, you're gut is telling truth...

This will be another devops indoctrination

What did you expect? ;)



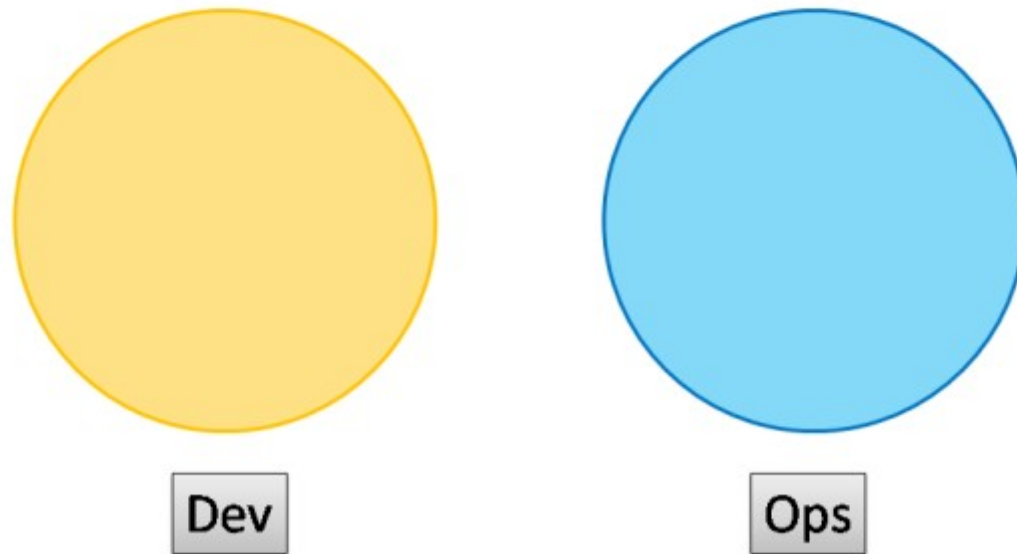
# **DevOps Anti-Types & patterns**

This is a copy/paste from  
<http://blog.matthewskelton.net/>  
w/my comments included

Great job Matthew! Thanks!

# DevOps Anti-Types

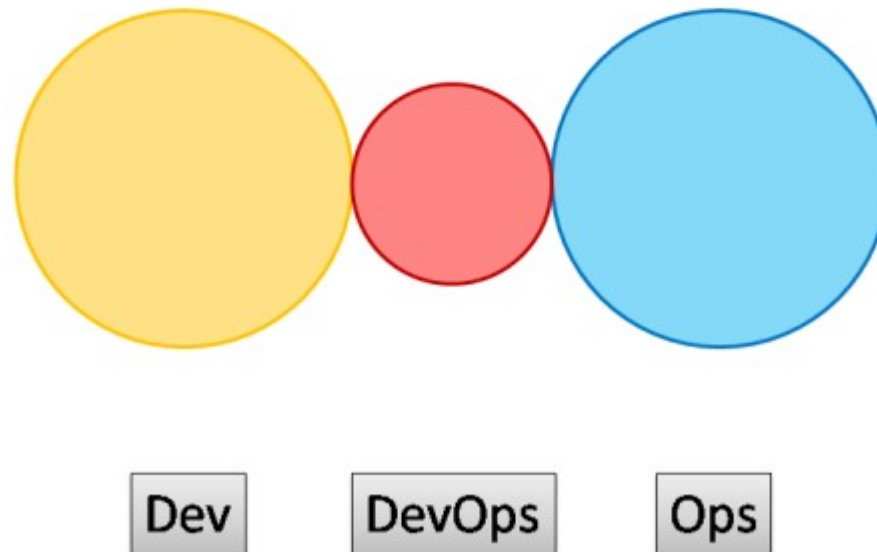
## Anti-Type A – Separate Silos





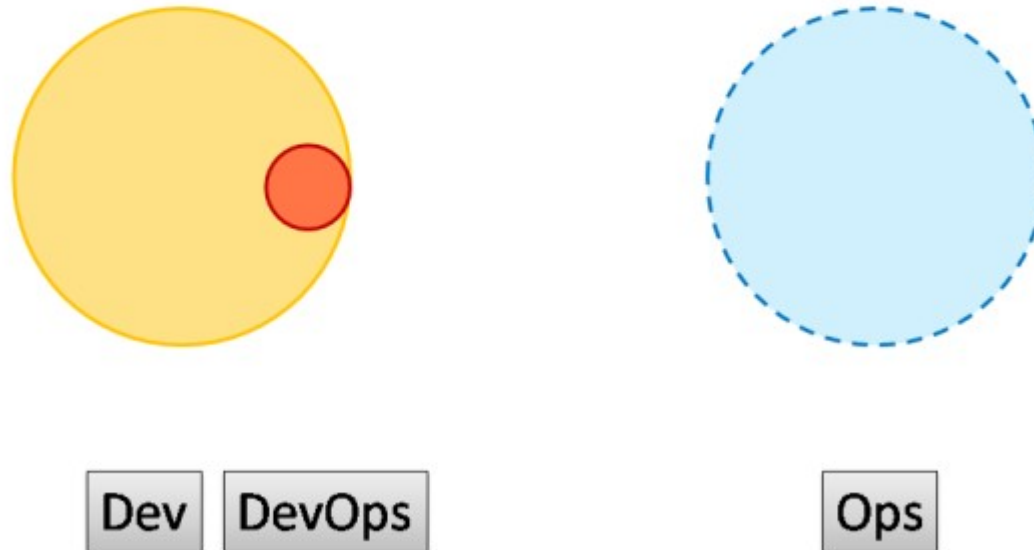
# DevOps Anti-Types

## Anti-Type B – Separate DevOps Silo



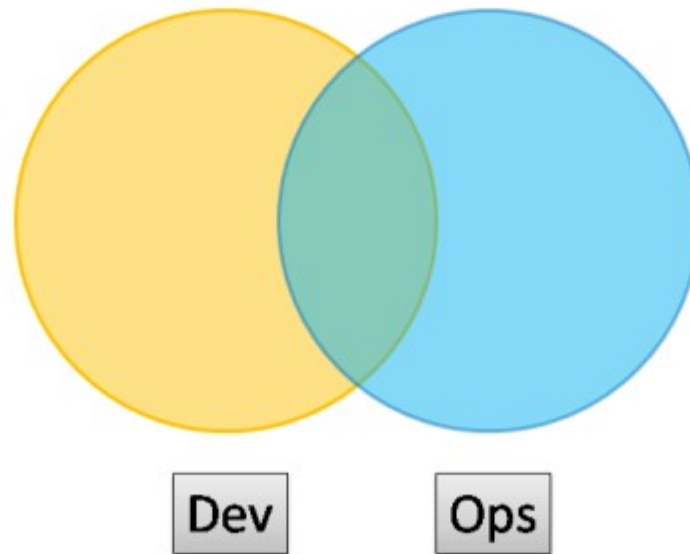
# DevOps Anti-Types

## Anti-Type C – “We Don’t Need Ops”



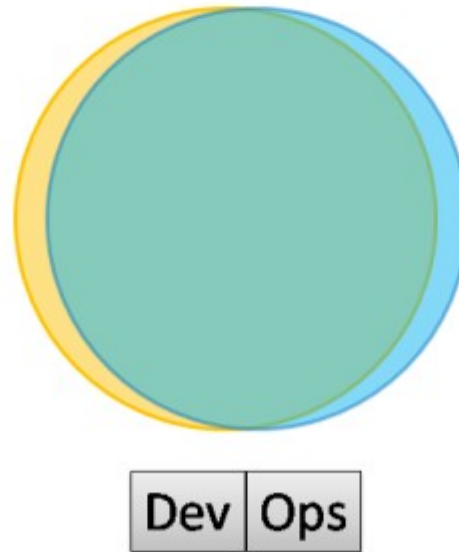
# DevOps Patterns

## Type 1 – Smooth Collaboration



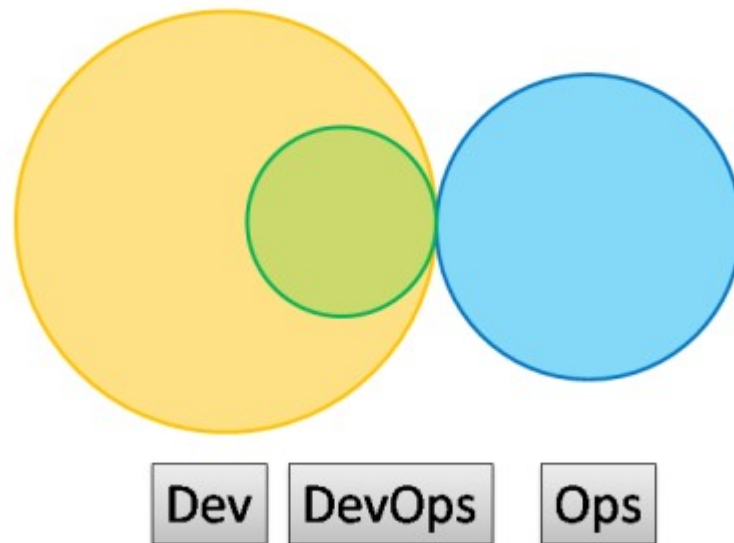
# DevOps Patterns

## Type 2 – Fully Embedded



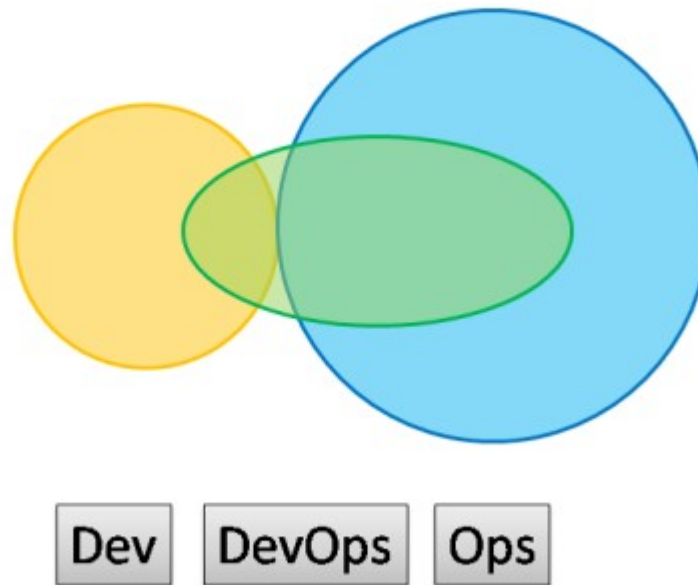
# DevOps Patterns

## Type 3 – Infrastructure-as-a-Service



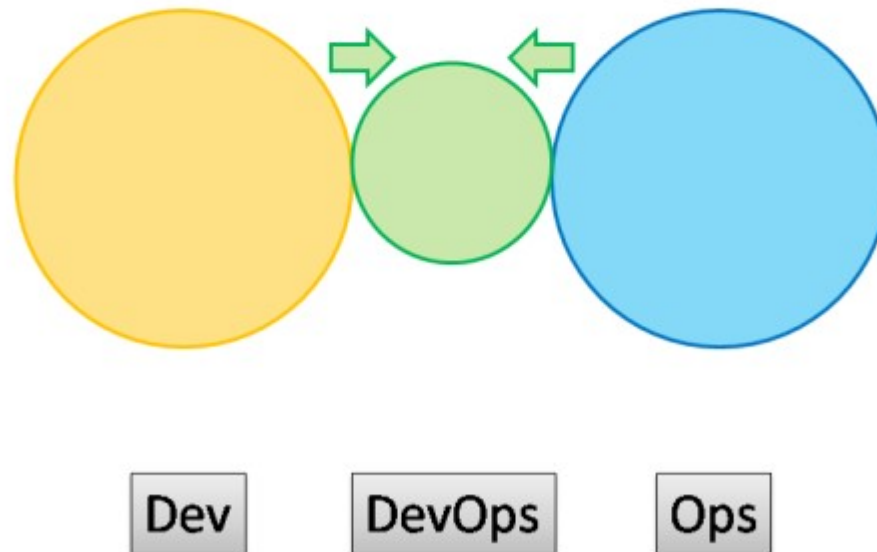
# DevOps Patterns

## Type 4 – DevOps-as-a-Service



# DevOps Patterns

## Type 5 – Temporary DevOps Team



Ok let's CAMS

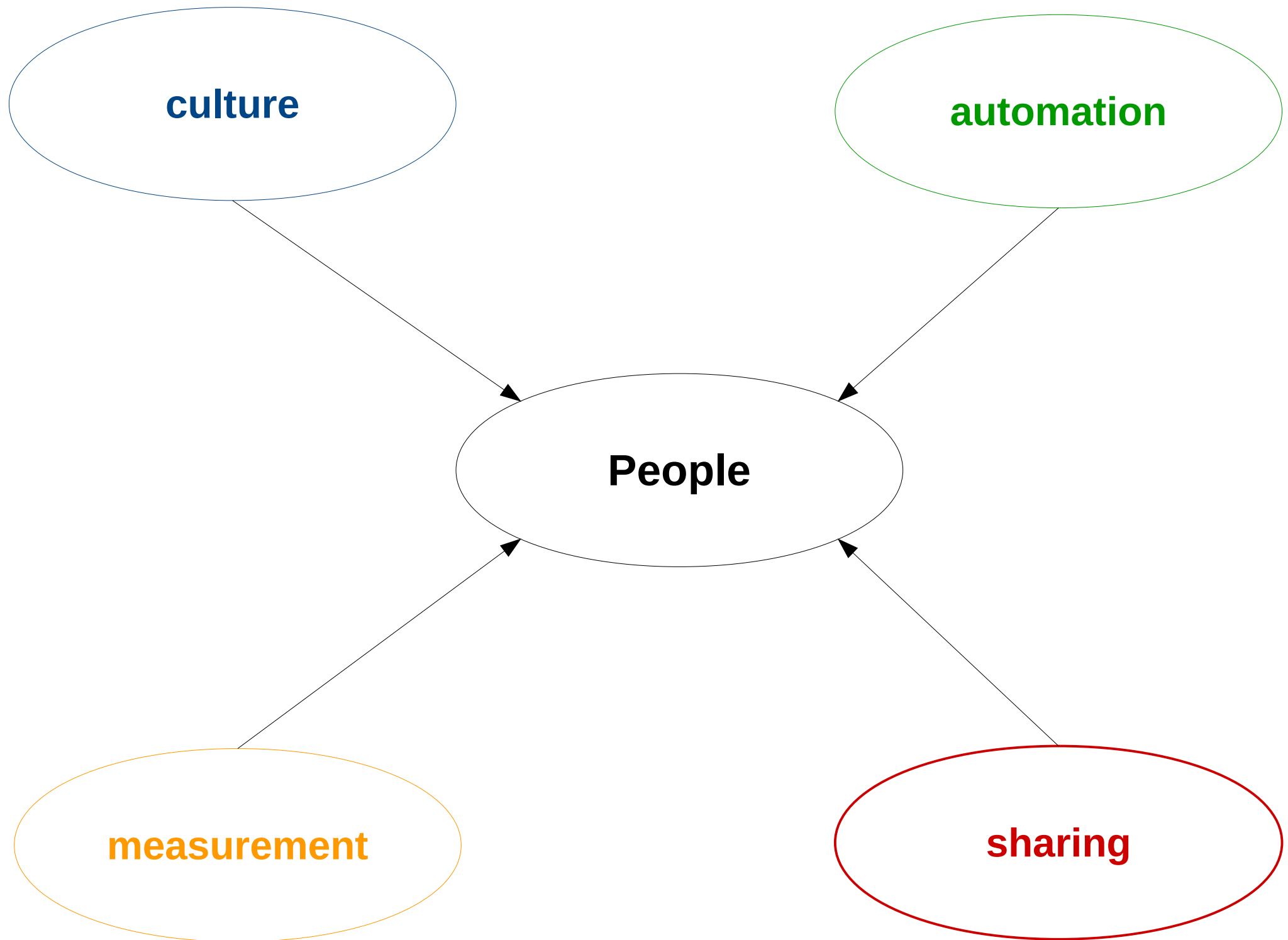


**DevOPS ?== CAMS**

(culture, automation, measurement, sharing)

**DevOPS !== CAMS**

**DevOPS === people!**

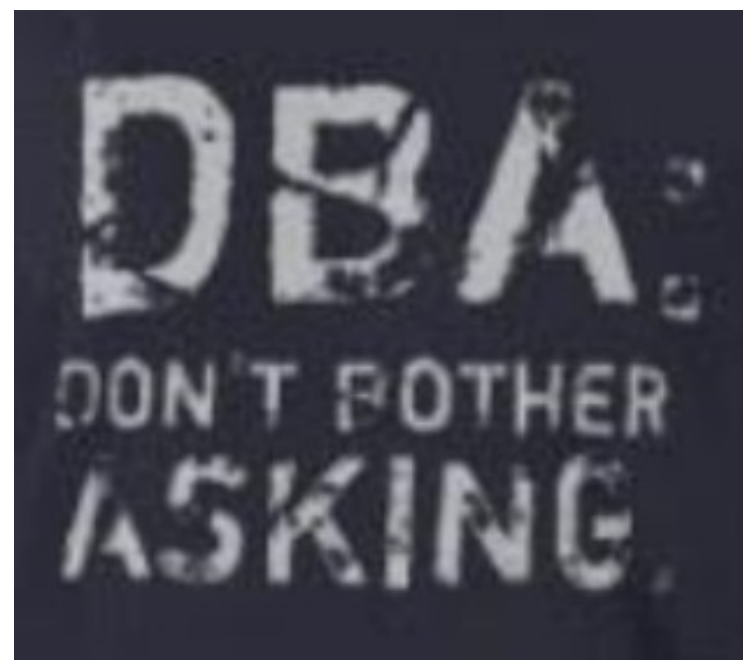


C for Culture

A for Automation

M for Monitoring

S for Sharing





# **Is there a need for change?**

“agile” and “cloud”:

- focus on delivery
- close collaboration
- lightweight environment and components

## **cultural change**

modification of a society through innovation,  
invention, discovery, or contact with other  
societies



# **Dead sea effect**

→ most talented evaporates

→ the residue

→ maintenance experts & bus factor == 1

- talk. often. and get along
- take responsibility - from beginning to the end
  - continuous improvement. seriously
  - be brave. don't be silent
- it's better to be unpolite I/German than polite I/Englishman

GTD? (getting things done)

GTD? (getting things done)

JFDI? (just fuckin' do it)

GTD? (getting things done)

JFDI? (just fuckin' do it)

MFBT? (move fast, break things)

$$\text{GTD} + \text{JFDI} + \text{MFBT} = \text{FCH}$$

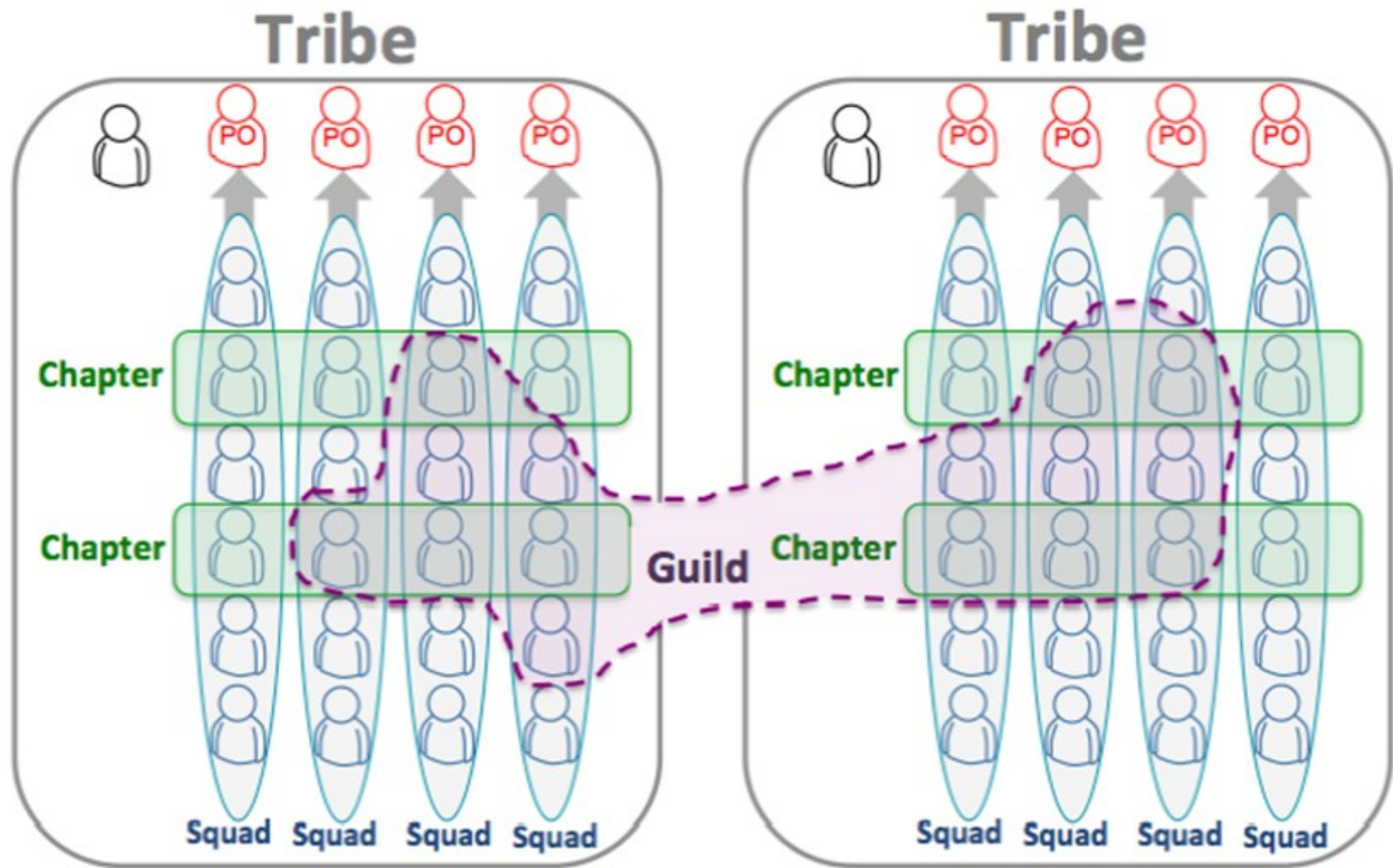
GTD + JFDI + MFBT = FCH

(Fuckin' Customer Happy)

# Scaling Agile @ Spotify

## with Tribes, Squads, Chapters & Guilds

Henrik Kniberg & Anders Ivarsson  
Oct 2012





C for Culture

**A for Automation**

M for Monitoring

S for Sharing

Automation is big for most sysadmins. We're inherently lazy, so the idea of pushing a button and making programs work for us? Appealing.

## **Standalone Sysadmin**

<http://www.standalone-sysadmin.com/blog/2011/04/view-from-the-other-side/>

- it has to be simple
- don't reinvent the wheel. don't fabric
- automate from very beginning

→ repeatable tasks leads to automation



- repeatable tasks leads to automation
  - automation leads to consistency



- repeatable tasks leads to automation
  - automation leads to consistency
    - consistency reduces errors



- repeatable tasks leads to automation
  - automation leads to consistency
    - consistency reduces errors
- reducing errors leads to stable environment



- repeatable tasks leads to automation
  - automation leads to consistency
    - consistency reduces errors
- reducing errors leads to stable environment
- stable environment leads to less unplanned work



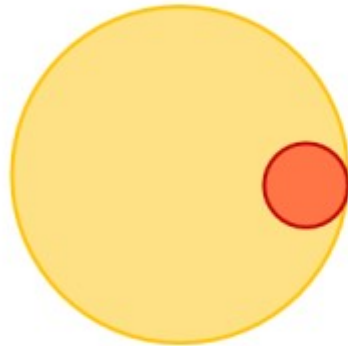


- repeatable tasks leads to automation
  - automation leads to consistency
    - consistency reduces errors
- reducing errors leads to stable environment
- stable environment leads to less unplanned work
- less unplanned work leads to focus on delivery

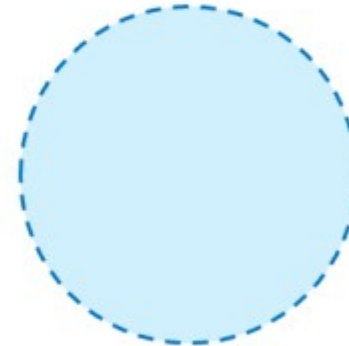


# Remember?

Anti-Type C – “We Don’t Need Ops”



Dev DevOps



Ops

# Short story of Anti-Type C

## “we don't need ops”

```
# it's madness with paths for different users and such option as:  
# sudo su  
# sudo -i  
# su -  
# su  
# that is why we add variables to two places
```

```
ENVIRONMENT_FILE = '/etc/environment'  
PROFILE_FILE      = '/etc/profile'  
INITIAL_PATH      = '/usr/local/bin:/usr/bin:/bin'
```

```
# due to sudo issues (resetting PATH by /etc/sudoers)  
# we have to add PATH to /root/.profile as well
```

# Short story of Anti-Type C

## “we don't need ops”

# it's madness with paths for different users and such option as:

# sudo su

# sudo -i

# su -

# su

# that is why we add variables to two places

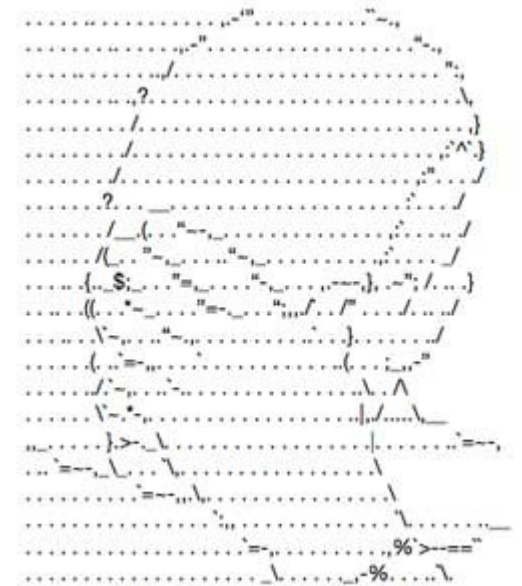
ENVIRONMENT\_FILE = '/etc/environment'

PROFILE\_FILE = '/etc/profile'

INITIAL\_PATH = '/usr/local/bin:/usr/bin:/bin'

# due to sudo issues (resetting PATH by /etc/sudoers)

# we have to add PATH to /root/.profile as well



# Short story of Anti-Type C

## “we don't need ops”

Shells:

- login
- non-login
- interactive
- non – interactive

# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- **su**
  - sudo su: interactive, non-login, .bashrc
  - sudo su -: interactive, login, /etc/profile;/root/.profile;/root/.bashrc
  - sudo -i: interactive, login, /root/.profile;/root/.bashrc;/root/.login
  - sudo /bin/bash: interactive, non-login, ~/.bashrc
  - sudo -s: reads \$SHELL and executes it

# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- su
  - sudo su: **interactive, non-login, .bashrc**
  - sudo su -: interactive, login, /etc/profile;/root/.profile;/root/.bashrc
  - sudo -i: interactive, login, /root/.profile;/root/.bashrc;/root/.login
  - sudo /bin/bash: interactive, non-login, ~/.bashrc
  - sudo -s: reads \$SHELL and executes it

# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- su
  - sudo su: interactive, non-login, .bashrc
  - sudo su -: **interactive, login, /etc/profile;/root/.profile;/root/.bashrc**
  - sudo -i: interactive, login, /root/.profile;/root/.bashrc;/root/.login
  - sudo /bin/bash: interactive, non-login, ~/.bashrc
  - sudo -s: reads \$SHELL and executes it



# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- su
  - sudo su: interactive, non-login, .bashrc
  - sudo su -: interactive, login, /etc/profile;/root/.profile;/root/.bashrc
  - sudo -i: **interactive, login, /root/.profile;/root/.bashrc;/root/.login**
  - sudo /bin/bash: interactive, non-login, ~/.bashrc
  - sudo -s: reads \$SHELL and executes it

# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- su
  - sudo su: interactive, non-login, .bashrc
  - sudo su -: interactive, login, /etc/profile;/root/.profile;/root/.bashrc
  - sudo -i: interactive, login, /root/.profile;/root/.bashrc;/root/.login
  - sudo /bin/bash: **interactive, non-login, ~/.bashrc**
  - sudo -s: reads \$SHELL and executes it

# Short story of Anti-Type C

## “we don't need ops”

### Shells:

- login
  - non-login
  - interactive
  - non – interactive
- 
- su
  - sudo su: interactive, non-login, .bashrc
  - sudo su -: interactive, login, /etc/profile;/root/.profile;/root/.bashrc
  - sudo -i: interactive, login, /root/.profile;/root/.bashrc;/root/.login
  - sudo /bin/bash: interactive, non-login, ~/.bashrc
  - sudo -s: reads \$SHELL and executes it

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #download if necessary
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1" /etc/issue maybe?

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #download if necessary
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #download if necessary
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"): ldconfig maybe?
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #download if necessary
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #download if necessary
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```



```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian What about RHEL, Fedora, Slackware, Gentoo?
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #downl. if necessary So whole this is for particular distro version?
            url = "http://.../libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

    if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
        sudo('chmod ug+x %s' % store_file_path)
        sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #downl. if necessary
            url = "http://libssl0.9.8_0.9.8o-squeeze14_amd64.deb"
```

```
if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path)
    sudo("dpkg -i %s" % store_file_path)
```

```
def is_ubuntu():
    return run("uname -a | grep Ubuntu | wc -l") == "1"

def install_apache_fix():
    if is_ubuntu():
        if exists("/lib/x86_64-linux-gnu/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            sudo("apt-get -y install libssl0.9.8")
    else:
        #Debian
        if exists("/usr/lib/libssl.so.0.9.8"):
            print "libssl.so.0.9.8 already installed - SKIPPING"
        else:
            #downl. if necessary
            url = "http://libssl0.9.8_0.9.8o-squeeze14_amd64.deb"

if download.sync_opt_download(_download_libssl_lock, url, store_file_path):
    sudo('chmod ug+x %s' % store_file_path) # declarative madness
    sudo("dpkg -i %s" % store_file_path)
```

# Imperativeness vs declarativeness

# Imperativeness vs declarativeness

```
def configure(dst_dir, config_properties, installer_file):  
    _copy_conf_file(dst_dir, properties)  
def _copy_conf_file(dst_dir, properties):  
    sudo("cp %s %s" % (srcConfigPath, targetConfigPath))  
    change_directory_owner(targetConfigPath)  
    sudo('chmod ug+x %s' % store_file_path)
```

```
- name: configure this  
hosts: all  
tasks:  
  - name: copy conf file  
    file: >  
      src={{ some_source }}  
      dest={{ some_destination }}  
      perms=0750
```

# Imperativeness vs declarativeness

```
def configure(dst_dir, config_properties, installer_file):  
    _copy_conf_file(dst_dir, properties)  
def _copy_conf_file(dst_dir, properties):  
    sudo("cp %s %s" % (srcConfigPath, targetConfigPath))  
    change_directory_owner(targetConfigPath)  
    sudo('chmod ug+x %s' % store_file_path)
```

- name: configure this
- hosts: all
- tasks:
  - name: copy conf file
  - file: >
    - src={{ some\_source }}
    - dest={{ some\_destination }}
    - perms=0750

≡ ANSIBLEWORKS



→ flat learning curve



# ≡ ANSIBLEWORKS

- flat learning curve
- doesn't required additional resources

# ≡ ANSIBLEWORKS

- flat learning curve
- doesn't required additional resources
- fit for maintenance jobs / procedures

# ANSIBLEWORKS

- flat learning curve
- doesn't required additional resources
- fit for maintenance jobs / procedures
- great for any containers as non-daemon

# ANSIBLEWORKS

- flat learning curve
- doesn't required additional resources
- fit for maintenance jobs / procedures
- great for any containers as non-daemon
- deals with “deployment specs”

# ANSIBLEWORKS

- flat learning curve
- doesn't required additional resources
- fit for maintenance jobs / procedures
- great for any containers as non-daemon
- deals with “deployment specs”
- might be easily adopted as universal language

Windows Update



Restart your computer to finish installing  
important updates

Windows can't update important files and services while the  
system is using them. Make sure to save your files before  
restarting.

Remind me in:

10 minutes ▼

Restart now

Postpone



- selinux enforcing i -rw-r--r--. stash stash  
unconfined\_u:object\_r:mysql\_db\_t:s0 authorized\_keys
- /etc/ssh/sshd\_config && /etc/network/interfaces
- iptables-save nope?
- broken \_netfs ?





**It's now safe to turn off  
your computer.**



What if...

- `./configure && make && make install` → `.zip`
- Dev & Ops have 2 different build & installation methods?

Plz..

- pkg repos (or Nexus)
- use fpm for creating pkgs if needed (demo)

C for Culture

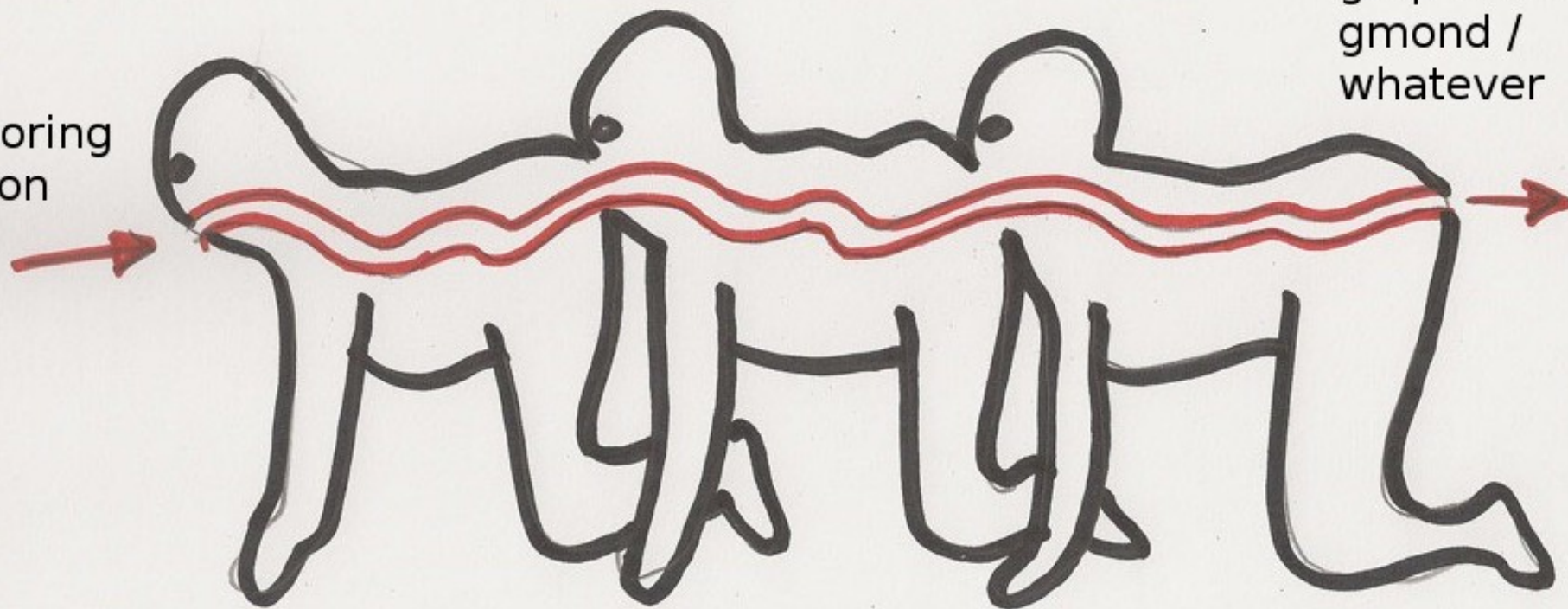
A for Automation

**M for Monitoring**

S for Sharing

monitoring  
daemon  
on  
node

graphana /  
gmond /  
whatever



- make developers create monitoring
- find yourself between RRD and InfluxDB
- will product team be able to query your monitoring DB?
- Etsy case (Ganglia / Graphite)

- make developers create monitoring
- **find yourself between RRD and InfluxDB**
- will product team be able to query your monitoring DB?
- Etsy case (Ganglia / Graphite)



- make developers create monitoring
- find yourself between RRD and InfluxDB
- **will product team be able to query your monitoring DB?**
- Etsy case (Ganglia / Graphite)

- make developers create monitoring
- find yourself between RRD and InfluxDB
- will product team be able to query your monitoring DB?
- **Etsy case (Ganglia / Graphite)**

C for Culture

A for Automation

M for Monitoring

**S** for **Sharing**

- **learn on OPS mistakes**
- Major Incident Reports – source of improvement
- Learn developers about change management
- Make CM an easy process. Use simple tools.

- learn on OPS mistakes
- **Major Incident Reports – source of improvement**
- Learn developers about change management
- Make CM an easy process. Use simple tools.

- learn on OPS mistakes
- Major Incident Reports – source of improvement
- **Learn developers about change management**
- Make CM an easy process. Use simple tools.

- learn on OPS mistakes
- Major Incident Reports – source of improvement
- Learn developers about change management
- **Make CM an easy process. Use simple tools.**

Let's arch the infrastructure



## Addressing the space

→ VLSM

→ DHCP & DDNS

→ KISS: flat networks!

→ stop /24!

## Addressing the space

→ VLSM

→ **DHCP & DDNS**

→ KISS: flat networks!

→ stop /24!

## Addressing the space

→ VLSM

→ DHCP & DDNS

→ **KISS: flat networks!**

→ stop /24!

## Addressing the space

→ VLSM

→ DHCP & DDNS

→ KISS: flat networks!

→ stop /24!

Prefix	1st octet	2nd octet	3rd octet	4th octet	Max.Subnets /0	Max Subnets for /0,8,16,24	Hosts	IPs	0th subnet	1st subnet
/30	255	255	255	252	1073741824	64 for /24 network	2	4	0.0.0.0	0.0.0.4
/29	255	255	255	248	536870912	32 for /24 network	6	8	0.0.0.0	0.0.0.8
/28	255	255	255	240	268435456	16 for /24 network	14	16	0.0.0.0	0.0.0.16
/27	255	255	255	224	134217728	8 for /24 network	30	32	0.0.0.0	0.0.0.32
/26	255	255	255	192	67108864	4 for /24 network	62	64	0.0.0.0	0.0.0.64
/25	255	255	255	128	33554432	2 for /24 network	126	128	0.0.0.0	0.0.0.128
/24	255	255	255	0	16777216	0 for /24 network	254	256	0.0.0.0	0.0.1.0
/23	255	255	254	0	8388608	128 for /16 network	510	512	0.0.0.0	0.0.2.0
/22	255	255	252	0	4194304	64 for /16 network	1022	1024	0.0.0.0	0.0.4.0
/21	255	255	248	0	2097152	32 for /16 network	2046	2048	0.0.0.0	0.0.8.0
/20	255	255	240	0	1048576	16 for /16 network	4094	4096	0.0.0.0	0.0.16.0
/19	255	255	224	0	524288	8 for /16 network	8190	8192	0.0.0.0	0.0.32.0
/18	255	255	192	0	262144	4 for /16 network	16382	16384	0.0.0.0	0.0.64.0
/17	255	255	128	0	131072	2 for /16 network	32766	32768	0.0.0.0	0.0.128.0
/16	255	255	0	0	65536	0 for /16 network	65534	65536	0.0.0.0	0.1.0.0
/15	255	254	0	0	32768	128 for /8 network	131070	131072	0.0.0.0	0.2.0.0
/14	255	252	0	0	16384	64 for /8 network	262142	262144	0.0.0.0	0.4.0.0
/13	255	248	0	0	8192	32 for /8 network	524286	524288	0.0.0.0	0.8.0.0
/12	255	240	0	0	4096	16 for /8 network	1048574	1048576	0.0.0.0	0.16.0.0
/11	255	224	0	0	2048	8 for /8 network	2097150	2097152	0.0.0.0	0.32.0.0
/10	255	192	0	0	1024	4 for /8 network	4194302	4194304	0.0.0.0	0.64.0.0
/9	255	128	0	0	512	2 for /8 network	8388606	8388608	0.0.0.0	0.128.0.0
/8	255	0	0	0	256	0 for /8 network	16777214	16777216	0.0.0.0	1.0.0.0
/7	254	0	0	0	128	128 for /0 network	33554430	33554432	0.0.0.0	2.0.0.0
/6	252	0	0	0	64	64 for /0 network	67108862	67108864	0.0.0.0	4.0.0.0
/5	248	0	0	0	32	32 for /0 network	134217726	134217728	0.0.0.0	8.0.0.0
/4	240	0	0	0	16	16 for /0 network	268435454	268435456	0.0.0.0	16.0.0.0
/3	224	0	0	0	8	8 for /0 network	536870910	536870912	0.0.0.0	32.0.0.0
/2	192	0	0	0	4	4 for /0 network	1073741822	1073741824	0.0.0.0	64.0.0.0
/1	128	0	0	0	2	2 for /0 network	2147483646	2147483648	0.0.0.0	128.0.0.0
/0	0	0	0	0	1	0 for /0 network	4294967294	4294967296	-	-

What about DNS?

→ BIND roxx (views etc)

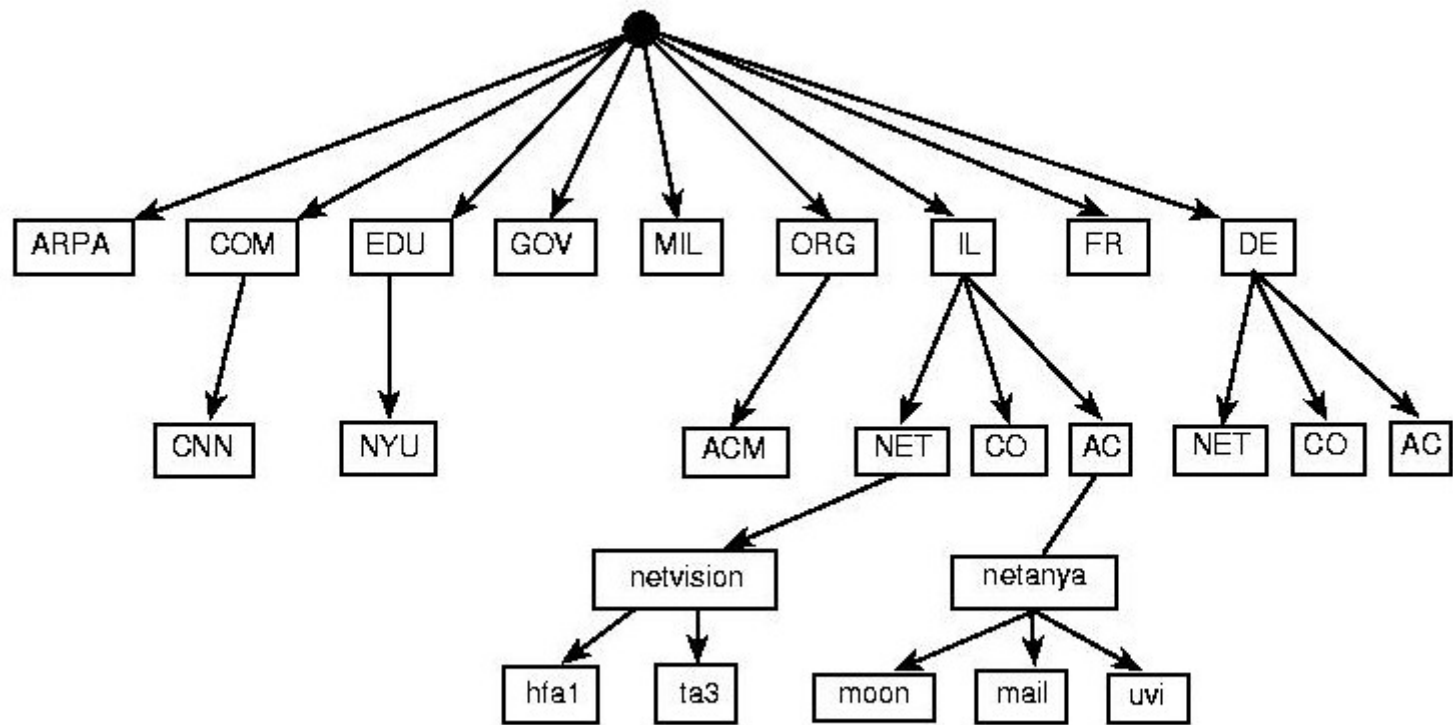
→ KISS: maybe decentralized w/Ansible?

```
view "internal-view" {  
    match-clients { internal; };  
    recursion yes;  
  
    zone "lasyk.info" IN {  
        type master;  
        file "internal.lasyk.info.conf";  
        allow-transfer { any; }  
    };  
};
```

```
view "external-view" {  
    match-clients { any; };  
    recursion no;  
  
    zone "lasyk.info" IN {  
        type master;  
        file "external.lasyk.info.conf";  
        allow-transfer { none; };  
    };  
};
```

```
view "internal-view" {  
    match-clients { internal; };  
    recursion yes;  
  
    zone "lasyk.info" IN {  
        type master;  
        file "internal.lasyk.info.conf";  
        allow-transfer { any; }  
    };  
  
view "external-view" {  
    match-clients { any; };  
    recursion no;  
  
    zone "lasyk.info" IN {  
        type master;  
        file "external.lasyk.info.conf";  
        allow-transfer { none; };  
    };
```





# Linux containers equation

Linux Containers = namespaces + cgroups + storage

# control groups (cgroups)

Control Groups provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behavior

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting



# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

# control groups (cgroups)

- grouping processes
- allocating resources to particular groups
  - memory
  - network
  - CPU
  - storage bandwidth (I/O throttling)
  - device whitelisting

control groups (cgroups)

little demo?

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname



# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

Providing a unique views of the system for processes.

- PID – PIDs isolation
- NET – network isolation (via virt-ifaces; demo)
- IPC – won't use this
- MNT – chroot like; deals w/mountpoints
- UTS – deals w/hostname

# Kernel Namespaces

little demo?

# OverlayFS

- hell fast (you'll see)
- page cache sharing
- finally in upstream kernel (in rhel from 7.2)
- finally supported by docker (-s overlay)
- SELinux not there yet (but will be)

# OverlayFS

- hell fast (you'll see)
- **page cache sharing**
- finally in upstream kernel (in rhel from 7.2)
- finally supported by docker (-s overlay)
- SELinux not there yet (but will be)

# OverlayFS

- hell fast (you'll see)
- page cache sharing
- finally in upstream kernel (in rhel from 7.2)
- finally supported by docker (-s overlay)
- SELinux not there yet (but will be)

# OverlayFS

- hell fast (you'll see)
- page cache sharing
- finally in upstream kernel (in rhel from 7.2)
- finally supported by docker (-s overlay)
- SELinux not there yet (but will be)

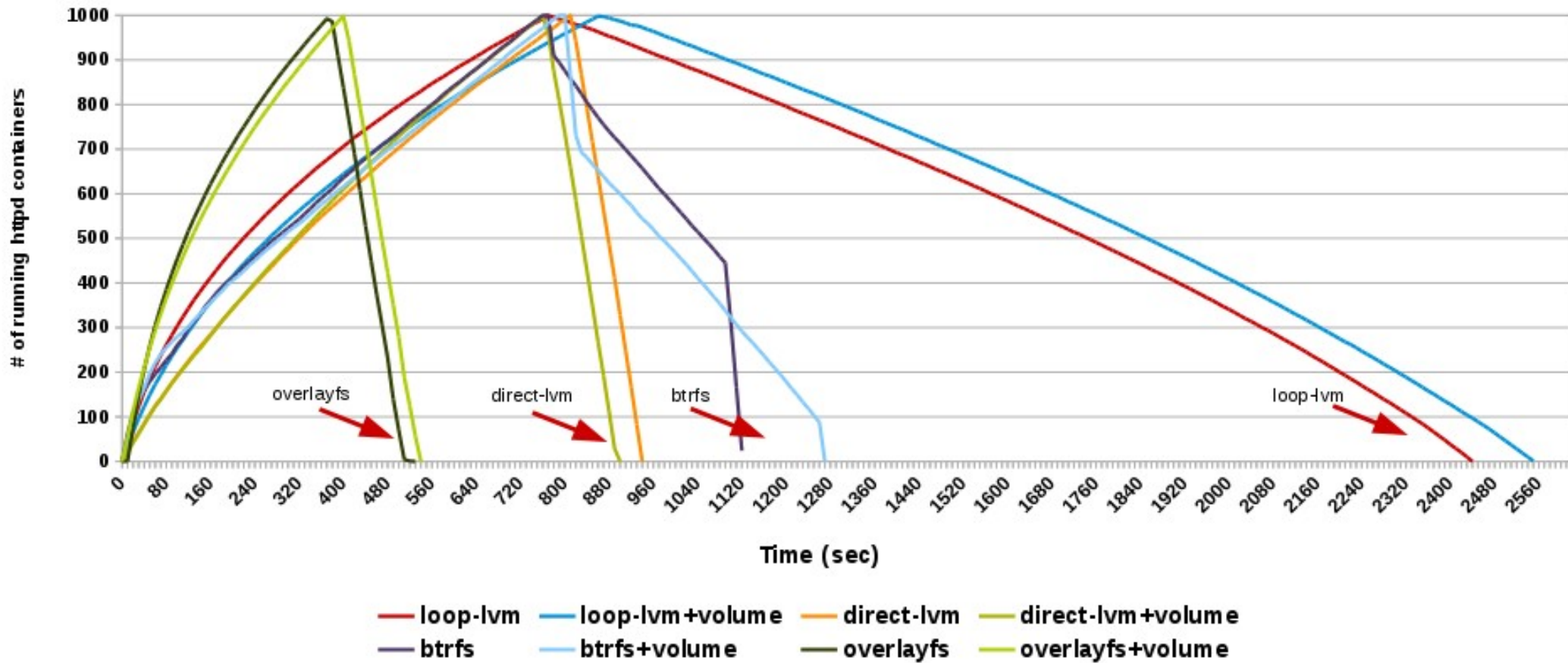
# OverlayFS

- hell fast (you'll see)
- page cache sharing
- finally in upstream kernel (in rhel from 7.2)
- finally supported by docker (-s overlay)
- SELinux not there yet (but will be)



# OverlayFS

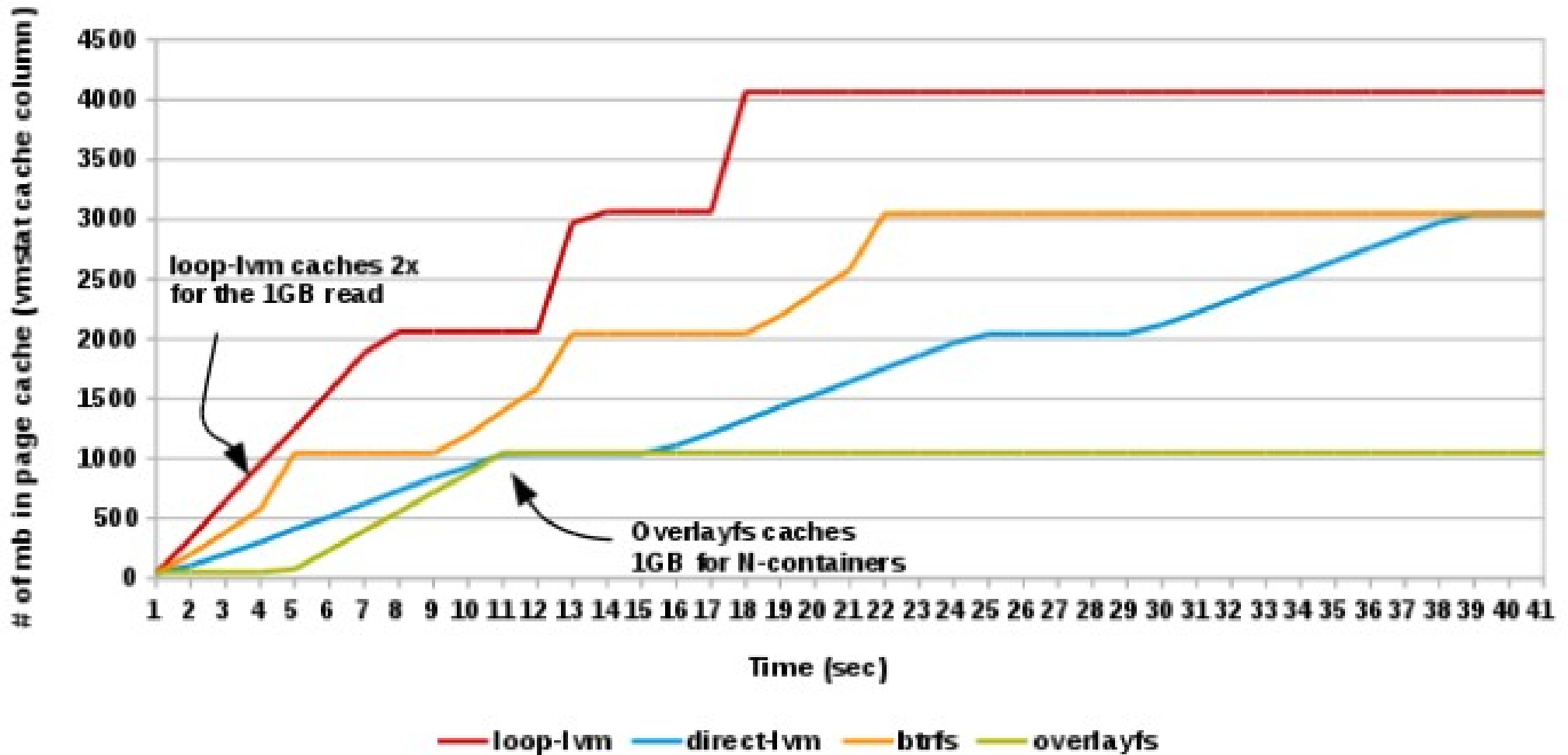
Container Create/Destroy Times



# OverlayFS

## Docker Page Cache Usage Test

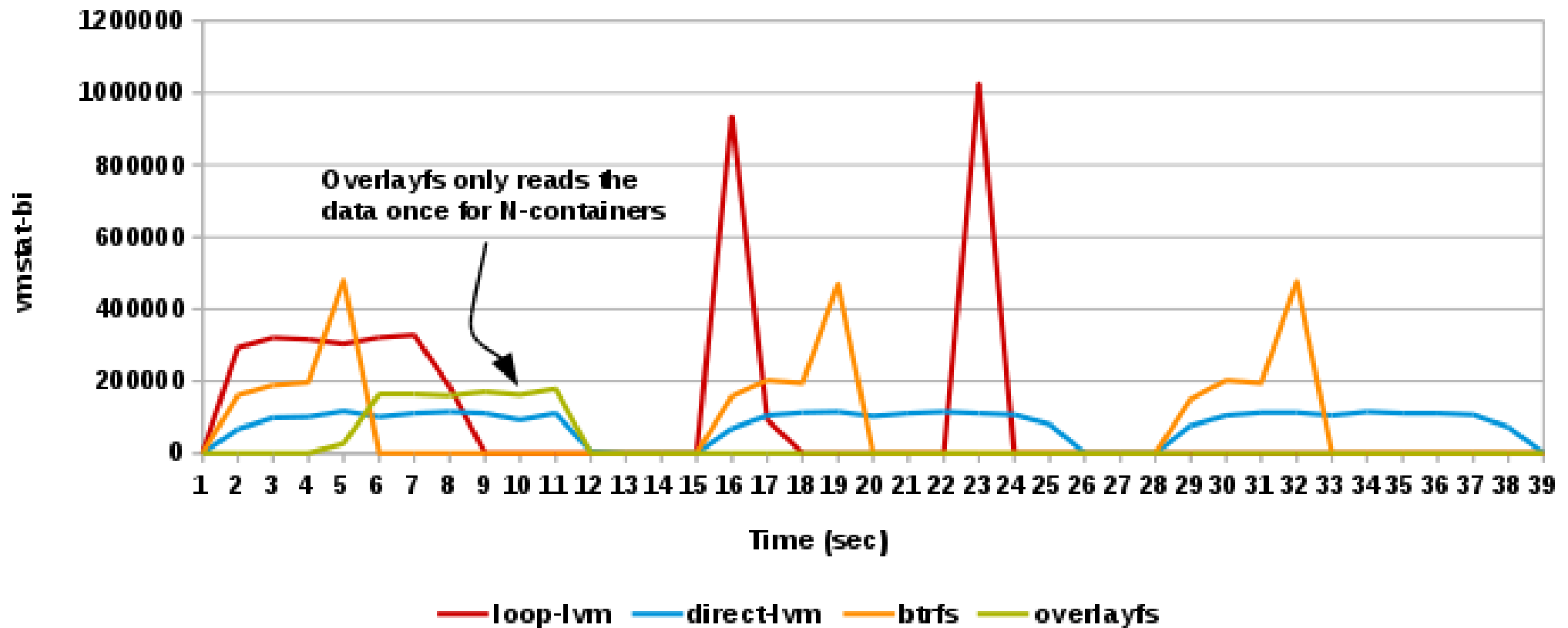
docker-1.1 + 3.17-rc1



# OverlayFS

## Docker Page Cache Usage Test

docker-1.1 + 3.17-rc1



## Developers' envs?

- use containers!
- configure cgroups
- use LXC / LXC Web Panel
- use Ansible for spinning up anything!

## Developers' envs?

- use containers!
- **configure cgroups**
- use LXC / LXC Web Panel
- use Ansible for spinning up anything!

## Developers' envs?

- use containers!
- configure cgroups
- use LXC / LXC Web Panel
- use Ansible for spinning up anything!

## Developers' envs?

- use containers!
- configure cgroups
- use LXC / LXC Web Panel
- use Ansible for spinning up anything!

## GENERAL

[Overview](#)

## CONTAINERS

**server1**

server2

server3

server4

## LXC SETTINGS

[Networking](#)[Check config](#)

## LXC WEB PANEL

[Users](#)[About](#)

## Server1

▶ Start

■ Stop

|| Freeze

Hostname

web

Start at boot



Network flag

Up

Network type

veth

Network link

lxcbr0

HW address

00:16:3e:ac:5e:45

IP address

10.0.3.10

Memory limit \*



1024



MB

Memory + Swap limit \*



2048



MB

CPUs \*\*

All

(e.g 0 or 0-1,3 or 0,3)

CPU Shares \*\*

1024



Root FS

/var/lib/lxc/server1/rootfs

(e.g /var/lib/lxc/server1/rootfs)

✓ Apply

## INFOS

Status : **Running**

Pid : 18188

Network : **Up**Mem. usage : **176 MB**

Arch : amd64

\* Set to max to unset (unlimited)

\*\* Leave empty to unset



Containers embraces granularity → microservices!

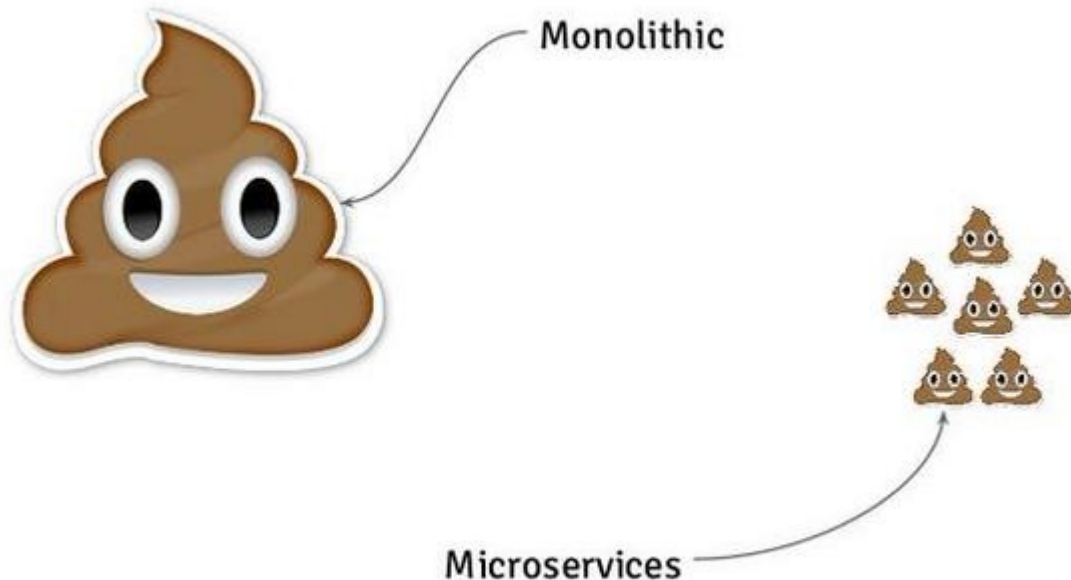
Containers embraces granularity → microservices!

Watch out for microservices architecture, or...

# Containers embraces granularity → microservices!

## Watch out for microservices architecture, or...

### Monolithic vs Microservices



*You might as well just kill  
yourself right now*

# Web Development With Assembly



O'REILLY®

*Bob Johnson  
with His Therapist*

Who knows FHS?

Who knows FHS?

→ 'temp' – what it consist?

Who knows FHS?

→ 'temp' – what it consist?

→ actually: “This Entity Must Persist” ;)

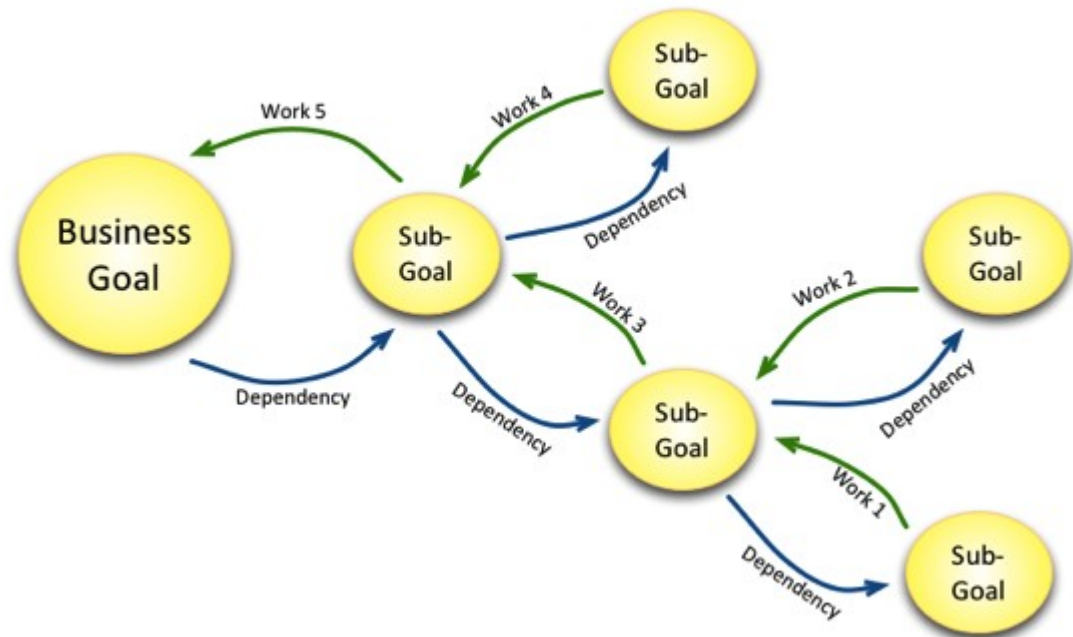
## Who knows FHS?

- 'temp' – what it consist?
- actually: “This Entity Must Persist” ;)
- Define your FHS!



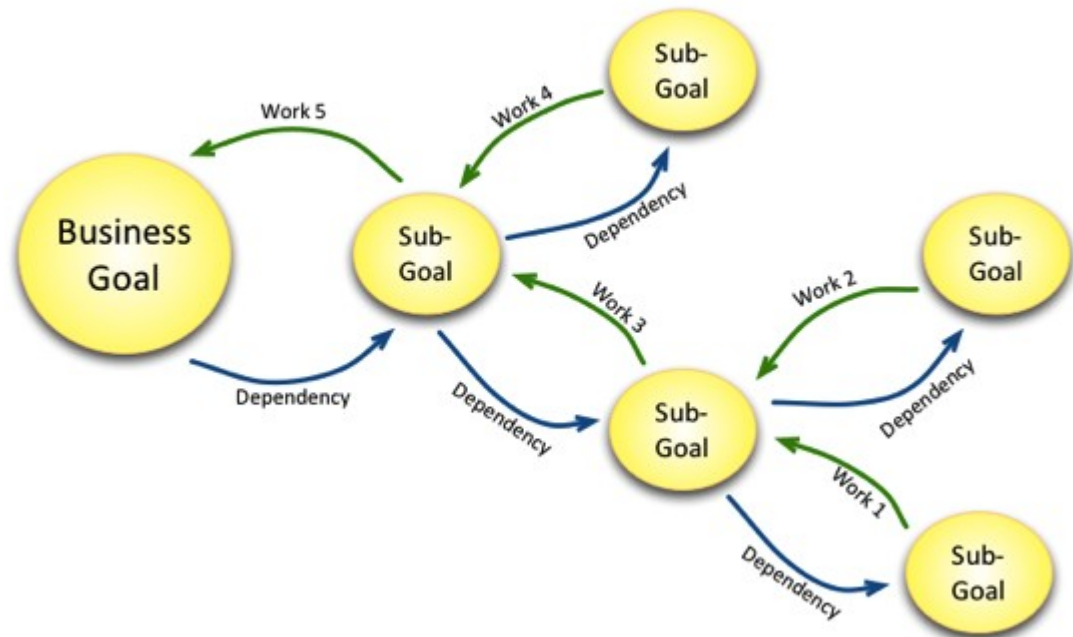
# Mikado Method for the win!

- set a goal
- experiment
- visualize
- rollback



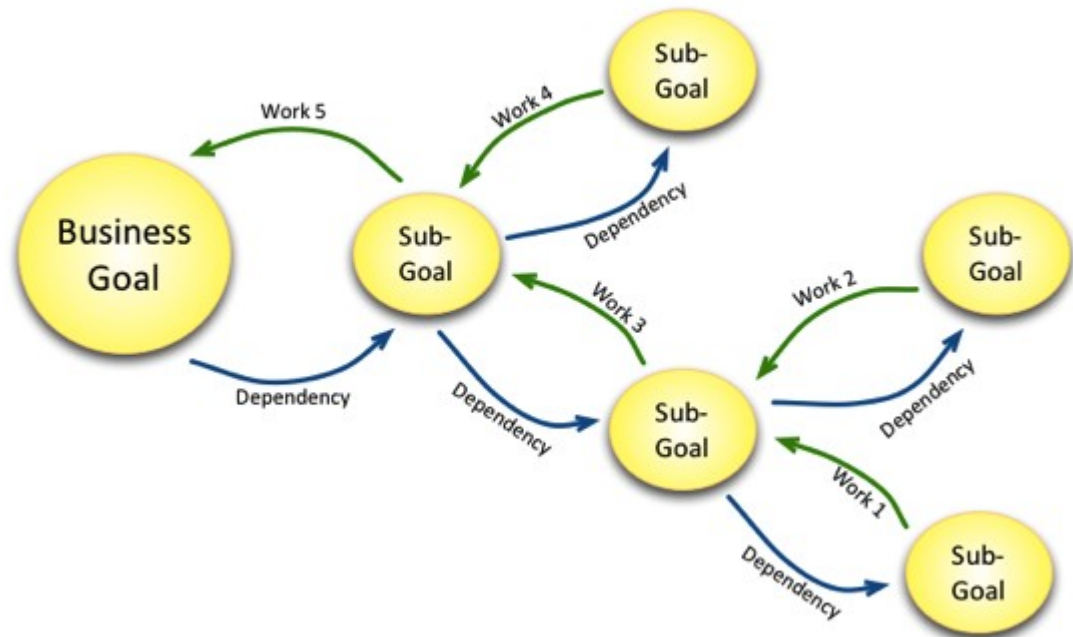
# Mikado Method for the win!

- set a goal
- **experiment**
- visualize
- rollback



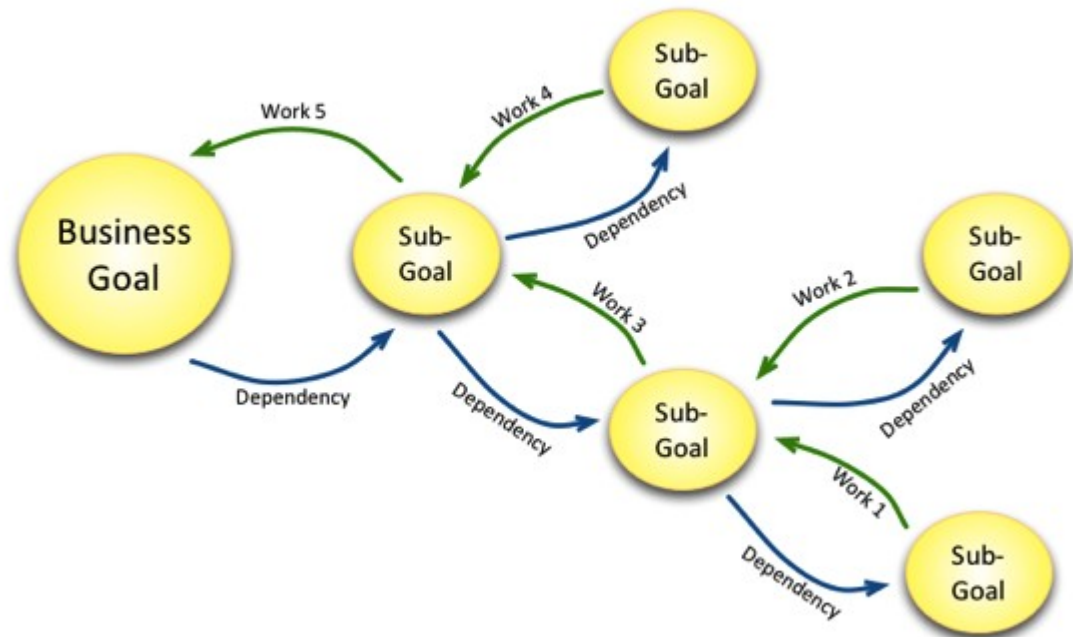
# Mikado Method for the win!

- set a goal
- experiment
- **visualize**
- rollback



# Mikado Method for the win!

- set a goal
- experiment
- visualize
- **rollback**



# Mikado Method for the win!

- before any work and rollbacks..
- remember: monitoring & tests are your friends!
- think about testing strategy – think heatmaps!

# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

Much simpler w/one, flat network (for small envs)!

# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

Much simpler w/one, flat network (for small envs)!

# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

Much simpler w/one, flat network (for small envs)!



# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

Much simpler w/one, flat network (for small envs)!

# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

Much simpler w/one, flat network (for small envs)!

# Ansible & infra layers

repo1

Layer 1: bare metal,

Networking

Hypervisor + VM provisioning

Storage

Network interfaces

Storage mounts

repo2

Layer 2: VM

Networking

Container's engine & provisioning

Resources allocation

repo3

Layer 3: container

Application build

Application env

**Much simpler w/one, flat network (for small envs)!**

# SmartStack

- automated service discovery and registration framework
- ideal for SOA architectures
- ideal for continuous integration & delivery
- solves “works on my machine” problem

# SmartStack

- automated service discovery and registration framework
- ideal for SOA architectures
- ideal for continuous integration & delivery
- solves “works on my machine” problem

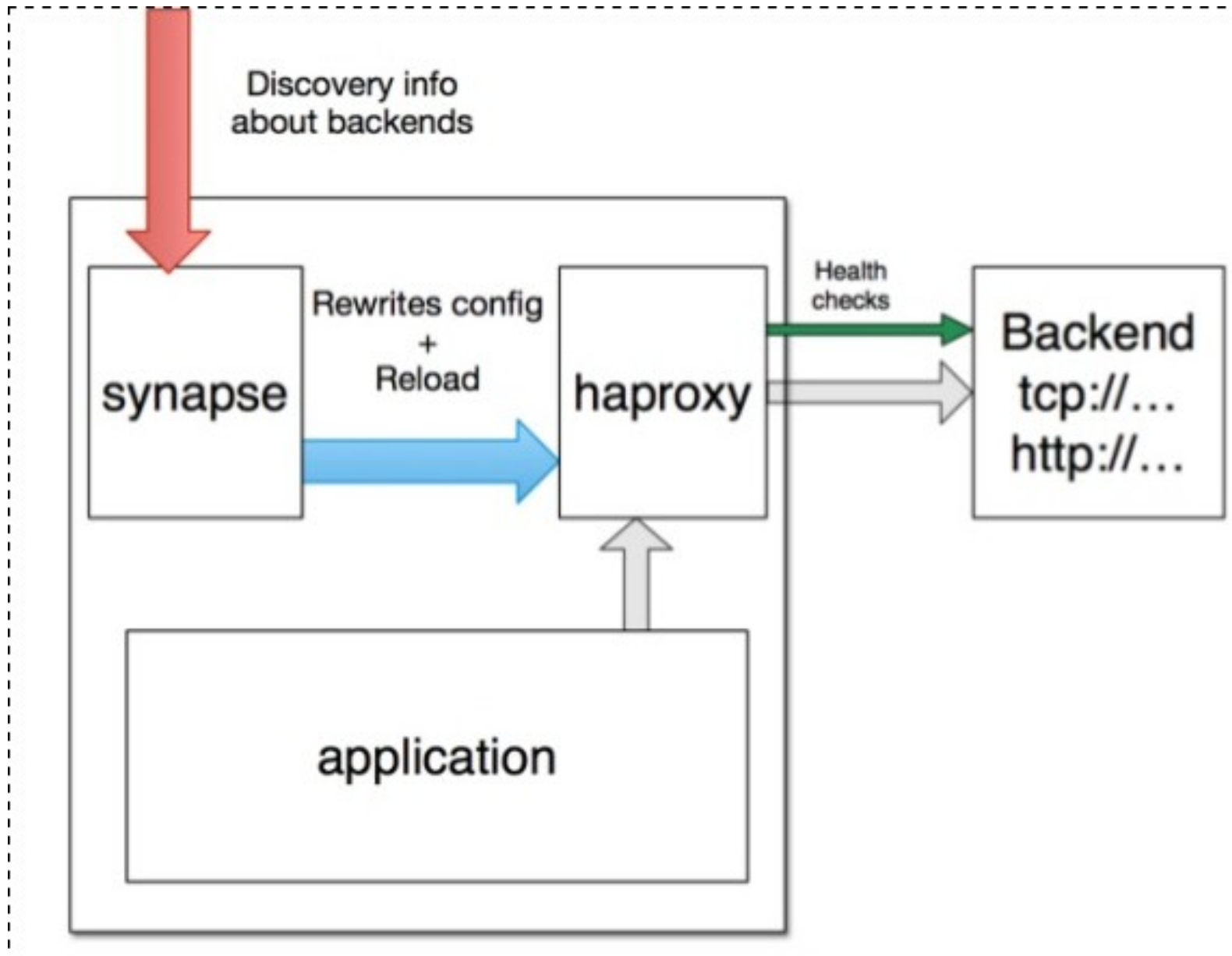
haproxy + nerve + synapse + zookeeper = smartstack

# SmartStack

## Synapse

- discovery service (via zookeeper or etcd)
- installed on every node
- writes haproxy configuration
- application doesn't have to be aware of this
- works same on bare / VM / docker
- <https://github.com/airbnb/nerve>

# SmartStack



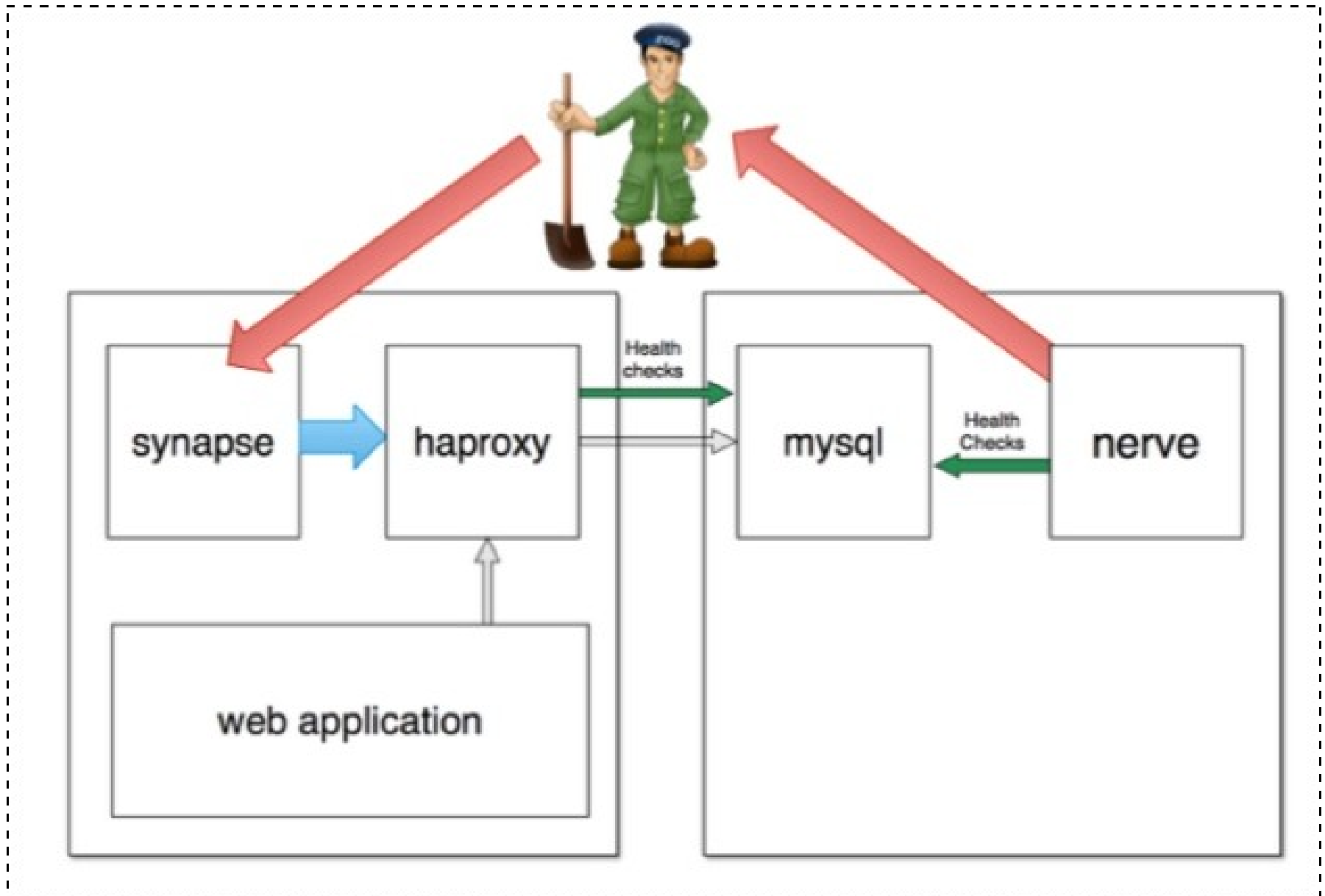
# SmartStack

## Nerve

- health checks (pluggable)
- register service info to zookeeper (or etcd)
- <https://github.com/airbnb/synapse>



# SmartStack



# SmartStack



Smartstack + Docker = <3

# Smartstack + Docker = <3

but also remember about Consul  
(come to #dockerkrk 2 meetup!)

questions?

# Archaeological workshop



# Archaeological workshop

- nmap, tcpdump, lsof, strace, sysdig, sar
- cgroups throttling on-the-fly

Do we have time for demo?

Hardware: disks?

→ RAID5 vs RAID10

→ Howto RAID over 1 disk ;)

→ Cheap SSD drives?



## Hardware: disks?

→ RAID5 vs RAID10

→ Howto RAID over 1 disk ;)

→ Cheap SSD drives?

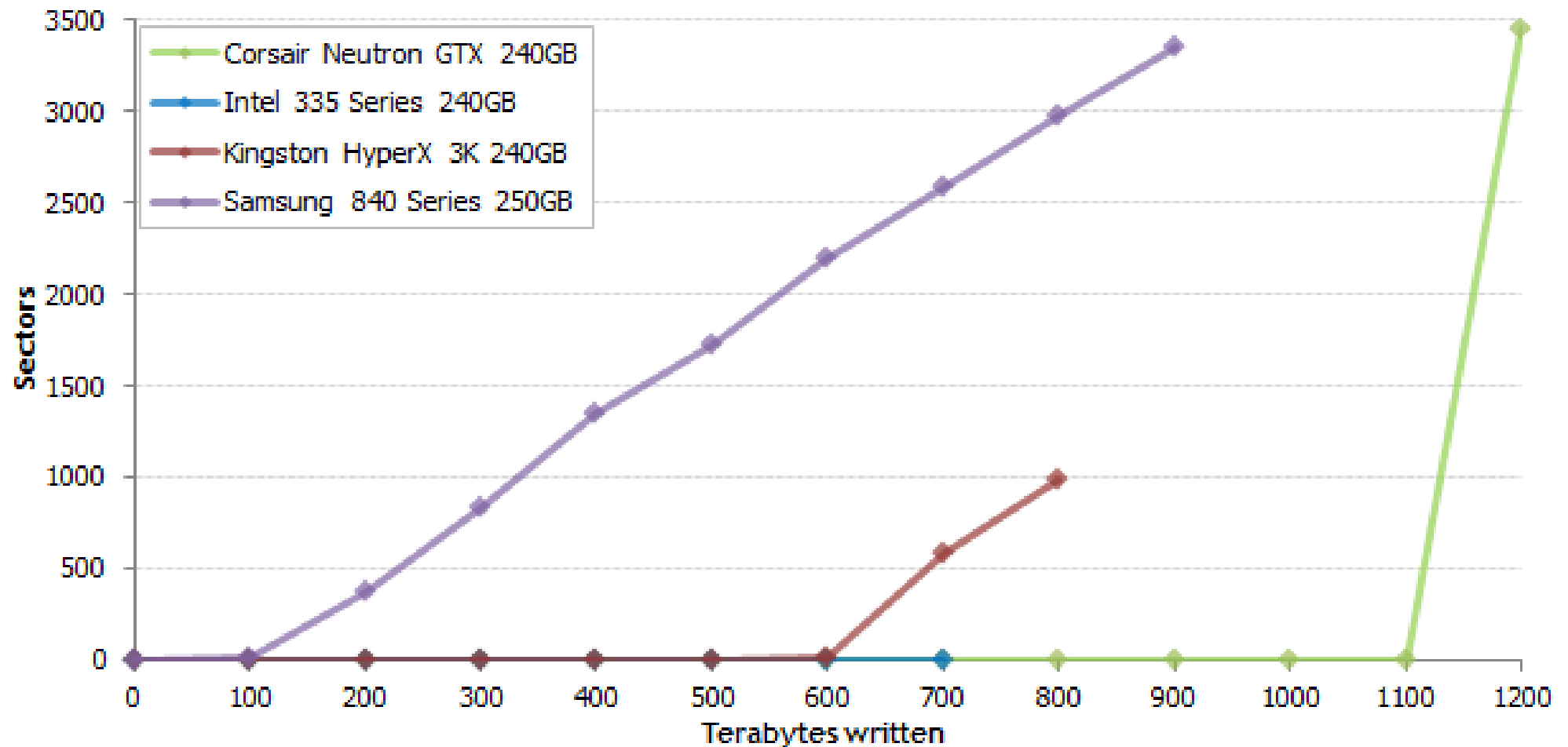
## Hardware: disks?

→ RAID5 vs RAID10

→ Howto RAID over 1 disk ;)

→ Cheap SSD drives?

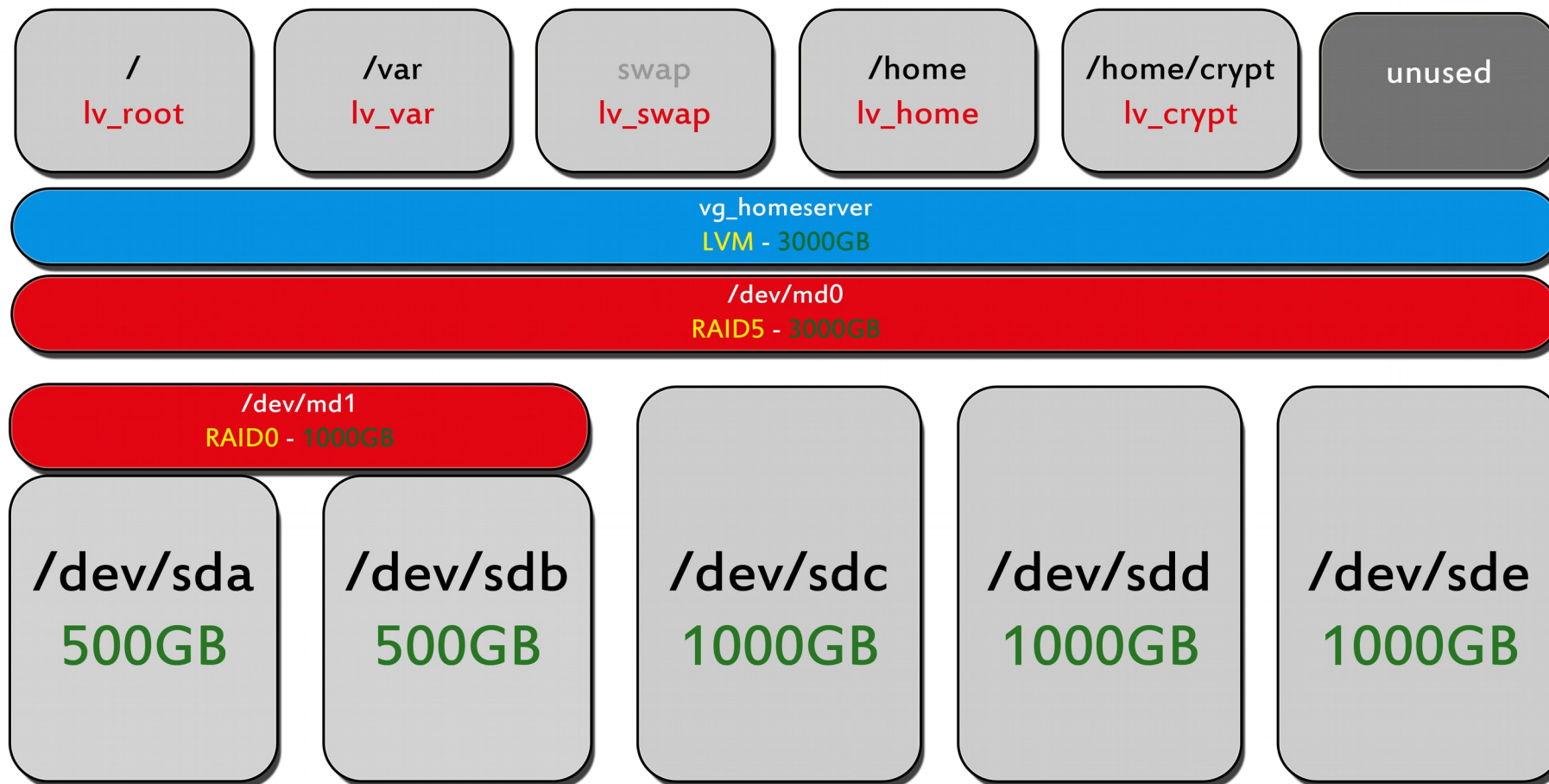
### Reallocated sectors: The early casualties



<http://techreport.com/review/27909/the-ssd-endurance-experiment-theyre-all-dead>

# Why use LVM?

- indexation (capacity, inodes check)
- capacity planning / iops per mount





# **Under the dome (of failure driven pipeline)**

Maciej Lasyk

4developers – Warsaw

2015-04-20