Using Puppet to Manage Linux

(and nearly everything else)

Luke Kanies
luke@reductivelabs.com
Founder, Reductive Labs
Nashville, Tennessee
USA

The Life of a typical svsadmin

Debian

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

reductive labs

The Life of a typical svsadmin

Debian

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Red Hat

```
yum install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/sshd start
```

reductive labs

The Life of a typical svsadmin

Debian

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Red Hat

```
yum install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/sshd start
```

reductive labs

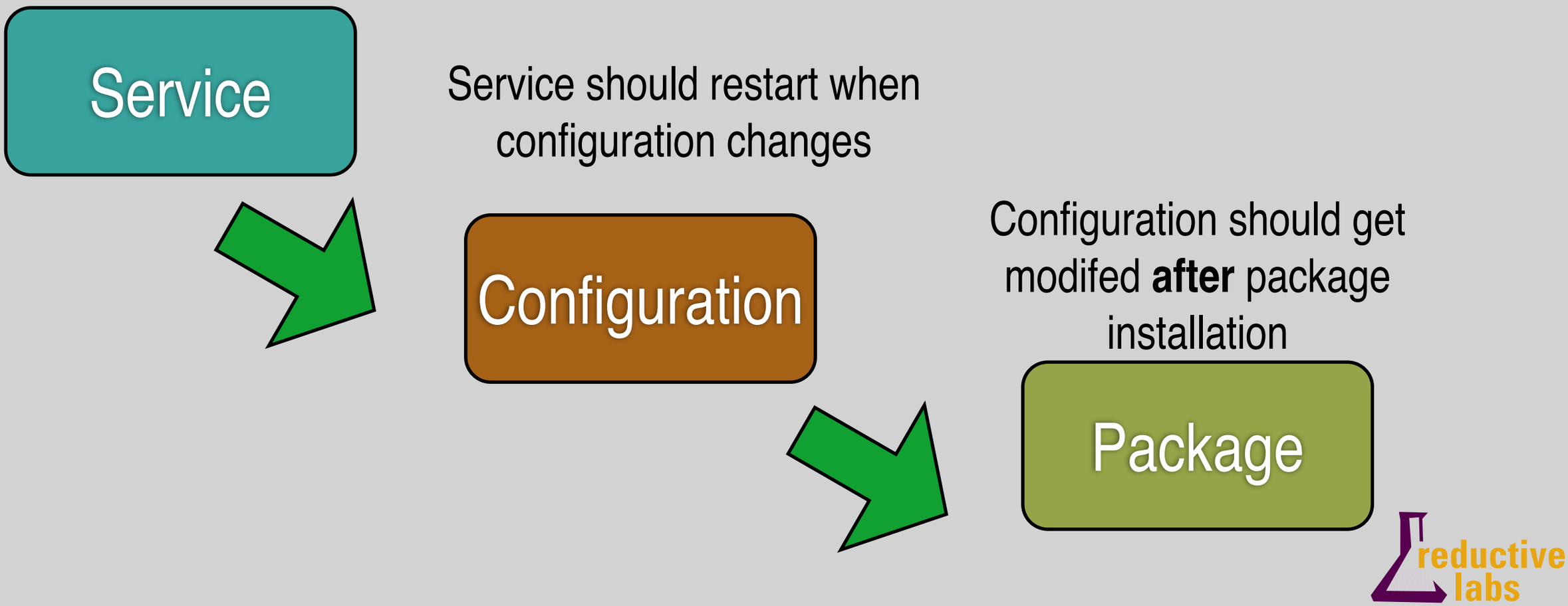
Relationships matter but are often implicit

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

reductive labs

Relationships matter but are often implicit

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```



Most of what we manage fits this model

Packages

Users

Groups

Services

FS Mounts

Cron Jobs

reductive labs

Puppet's primary advance is its Resource Abstraction Layer (RAL)

reductive labs

An Analogy

	Programming	**SysAdmin**
Low-level	Assembly	commands and files
Abstract	C	Resources

Better tools are about higher quality, not less effort

reductive labs

Resource Abstraction Layer

Resource Types

Package

User

Service

apt

dpkg

useradd

init.d

SMF

yum

ports

netinfo

Providers

reductive labs

Portable Abstractions

- 23 package types
- Users in NetInfo, useradd, pw
- Support for Debian, Ubuntu, Red Hat, Solaris, OS X, Gentoo, SuSE, FreeBSD, and more

reductive labs

It's an Operating System API

```
luke@culain(0) $ ralsh user luke
user { 'luke':
     uid => '100',
     gid => '1000',
     shell => '/bin/bash',
     home => '/home/luke',
     ensure => 'present',
     groups => ['sysadmin','audio','video','puppet'],
     comment => 'Luke Kanies'
}
luke@culain(0) $
```

reductive labs

Everything's Explicit

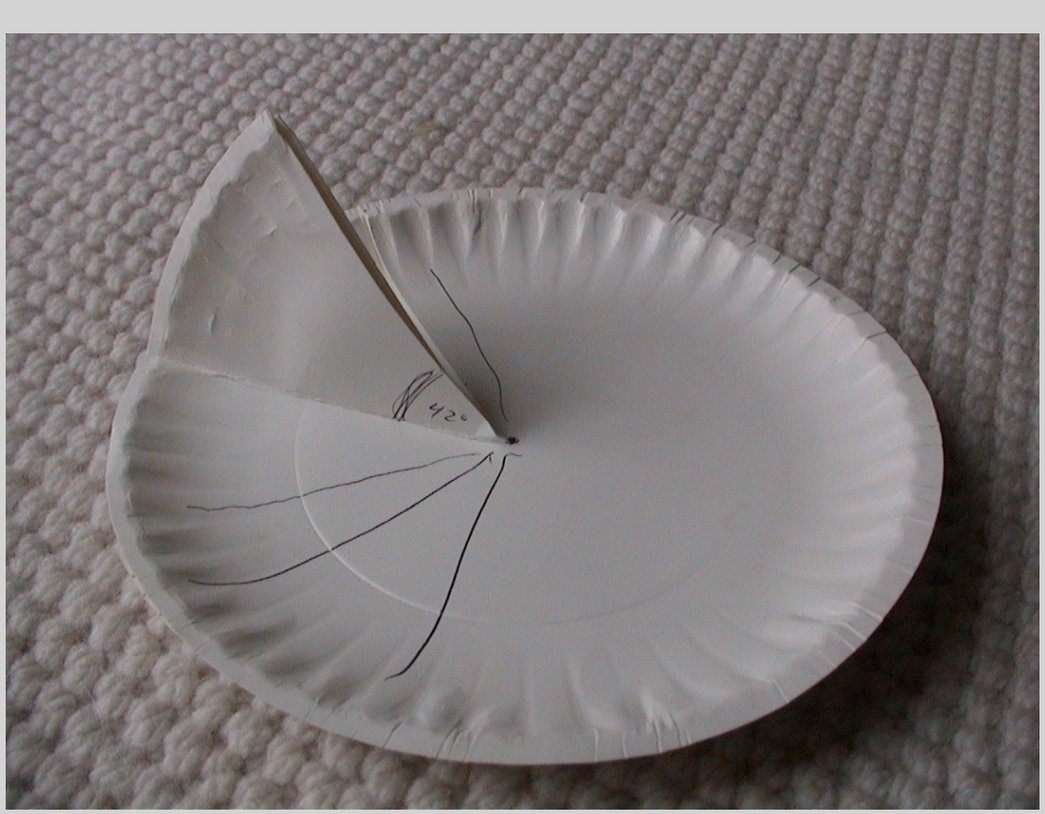
```
class ssh {
    package { ssh: ensure => installed }
    file { sshd_config:
        name => "/etc/ssh/sshd_config",
        owner => root,
        group => root,
        source => "puppet://server/apps/ssh/sshd_config
        before => Package[ssh]
    }
    service { sshd:
        ensure => running,
        subscribe => [Package[ssh], File[sshd_config]]
    }
}
```

reductive labs

It has a few other goals

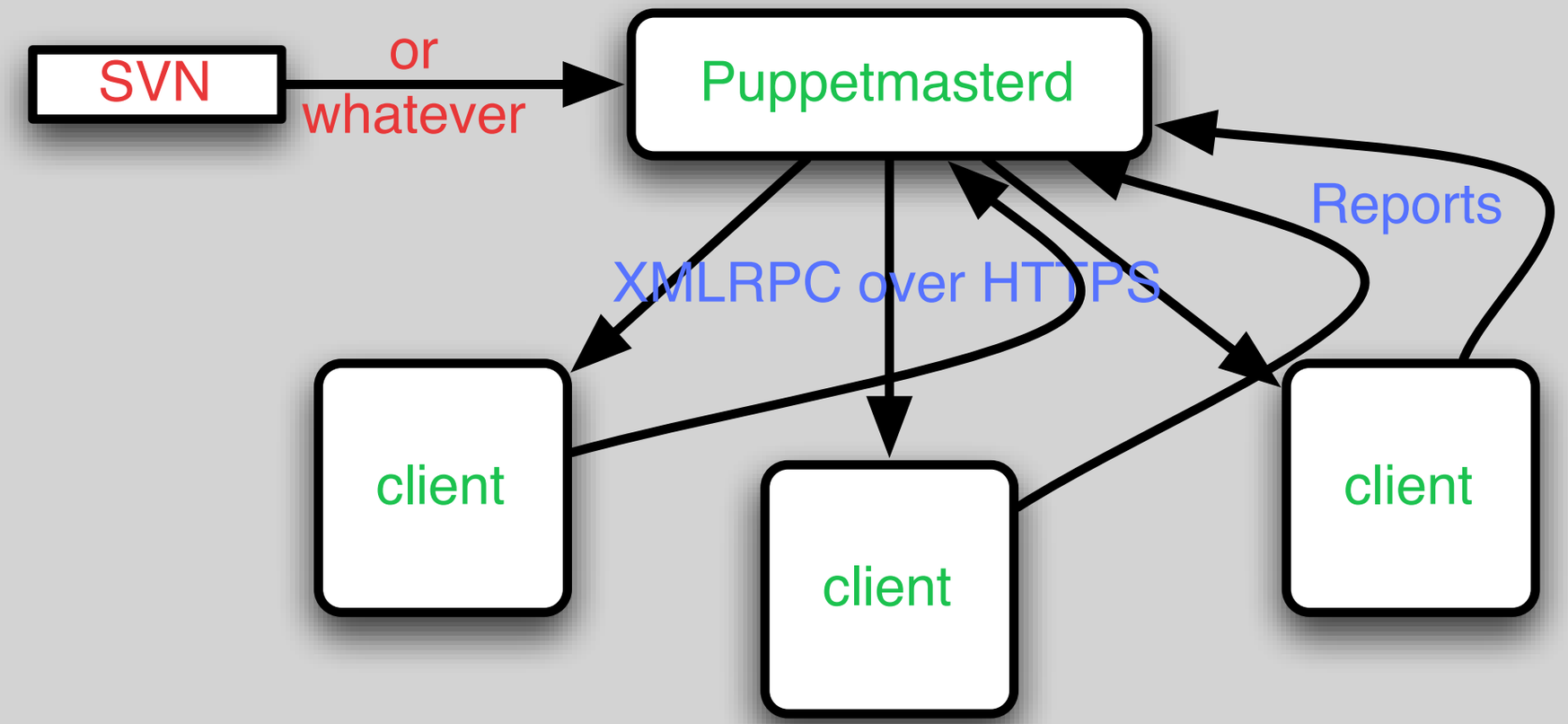
reductive labs

SSH and a *for* loop is not a solution



reductive labs

Centralized Management



reductive labs

An open development community

reductive labs

If the product sucks, I don't eat

reductive labs

Revisiting Abstraction

reductive labs

Resources are high-level abstractions for managing systems

reductive labs

Via **specifying** and **applying** resource catalogs

reductive labs

What resources are you managing?

reductive labs

How are they related?

reductive labs

What resource classes is the given host a member of?

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Becomes:

```
package { ssh: ensure => installed }
file { sshd_config:
     name => "/etc/ssh/sshd_config",
     source => "puppet://server/apps/ssh/sshd
}
service { sshd: ensure => running, }
```

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Becomes:

```
package { ssh: ensure => installed }
file { sshd_config:
     name => "/etc/ssh/sshd_config",
     source => "puppet://server/apps/ssh/sshd
}
service { sshd: ensure => running, }
```

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Becomes:

```
package { ssh: ensure => installed }
file { sshd_config:
     name => "/etc/ssh/sshd_config",
     source => "puppet://server/apps/ssh/sshd
}
service { sshd: ensure => running, }
```

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Becomes:

```
package { ssh: ensure => installed }
file { sshd_config:
     name => "/etc/ssh/sshd_config",
     source => "puppet://server/apps/ssh/sshd
}
service { sshd: ensure => running, }
```

reductive labs

A simple example

This:

```
apt-get install openssh-server
vi /etc/ssh/sshd_config
/etc/init.d/ssh start
```

Becomes:

```
package { ssh: ensure => installed }
file { sshd_config:
     name => "/etc/ssh/sshd_config",
     source => "puppet://server/apps/ssh/sshd
}
service { sshd: ensure => running, }
```

reductive labs

Portability and Naming

```
file { sshd_config:
    name => $operatingsystem ? {
        darwin => "/etc/sshd_config",
        default => "/etc/ssh/sshd_config",
    },
    owner => root,
    group => root,
    source => "puppet://server/apps/ssh/sshd_config",
}
service { sshd:
    ensure => running,
    subscribe => File[sshd_config]
}
```

reductive labs

Portability and Naming

```
file { sshd_config:
    name => $operatingsystem ? {
        darwin => "/etc/sshd_config",
        default => "/etc/ssh/sshd_config",
    },
    owner => root,
    group => root,
    source => "puppet://server/apps/ssh/sshd_config",
}
service { sshd:
    ensure => running,
    subscribe => File[sshd_config]
}
```

reductive labs

Adding Relationships

```
package { ssh: ensure => installed }
file { sshd_config:
    name => "/etc/ssh/sshd_config",
    source => "puppet://server/apps/ssh/sshd_config",
    after => Package[ssh]
}
service { sshd:
    ensure => running,
    subscribe => [Package[ssh], File[sshd_config]]
}
```

reductive labs

Adding Relationships

```
package { ssh: ensure => installed }
file { sshd_config:
    name => "/etc/ssh/sshd_config",
    source => "puppet://server/apps/ssh/sshd_config",
    after => Package[ssh]
}
service { sshd:
    ensure => running,
    subscribe => [Package[ssh], File[sshd_config]]
}
```

reductive labs

Resource Classes

```
class ssh {
    package { ssh: ensure => installed }
    file { sshd_config:
        name => "/etc/ssh/sshd_config",
        owner => root,
        group => root,
        source => "puppet://server/apps/ssh/sshd_config",
        after => Package[ssh]
    }
    service { sshd:
        ensure => running,
        subscribe => [Package[ssh], File[sshd_config]]
    }
}
```

reductive labs

Resource Classes

```
class ssh {
    package { ssh: ensure => installed }
    file { sshd_config:
        name => "/etc/ssh/sshd_config",
        owner => root,
        group => root,
        source => "puppet://server/apps/ssh/sshd_config",
        after => Package[ssh]
    }
    service { sshd:
        ensure => running,
        subscribe => [Package[ssh], File[sshd_config]]
    }
}
```

reductive labs

Associating Classes with Nodes

```
node culain inherits basenode {
    include puppet::server, mailserver
    include webserver, webserver::mail
}

node rh3a inherits basenode {
    include os::redhat
}

node laeg inherits basenode {
    include yum::repository, puppet::server
    include webserver, webserver::trac
}
```

reductive labs

Associating Classes with Nodes

```
node culain inherits basenode {
    include puppet::server, mailserver
    include webserver, webserver::mail
}

node rh3a inherits basenode {
    include os::redhat
}

node laeg inherits basenode {
    include yum::repository, puppet::server
    include webserver, webserver::trac
}
```

reductive labs

Associating Classes with Nodes

```
node culain inherits basenode {
    include puppet::server, mailserver
    include webserver, webserver::mail
}

node rh3a inherits basenode {
    include os::redhat
}

node laeg inherits basenode {
    include yum::repository, puppet::server
    include webserver, webserver::trac
}
```

reductive labs

So You've Got a Resource Catalog

reductive labs

The Configuration Process

The Configuration Process

1. Retrieve resource catalog from central server

The Configuration Process

1. Retrieve resource catalog from central server
2. Do a topological sort based on resource dependencies

reductive labs

The Configuration Process

1. Retrieve resource catalog from central server
2. Do a topological sort based on resource dependencies
3. Iterate across each resource in turn, fixing any resources that are out of sync

reductive labs

The Configuration Process

1. Retrieve resource catalog from central server
2. Do a topological sort based on resource dependencies
3. Iterate across each resource in turn, fixing any resources that are out of sync
4. This process repeats every 30 mins by default

reductive labs

Configuration Retrieval is via XMLRPC over HTTPS

And moving to REST
over HTTPS

reductive labs

Puppet has its own Certificate Authority

reductive labs

Resource sorting is done via dependencies

In this context, I sometimes call the Resource Catalog the ‘Resource Graph’

Transactions (for each resource)

Transactions (for each resource)

1. Retrieve current state (e.g., by querying dpkg db or doing a *stat)*

reductive labs

Transactions (for each resource)

1. Retrieve current state (e.g., by querying dpkg db or doing a *stat)*
2. Compare to desired state

reductive labs

Transactions (for each resource)

1. Retrieve current state (e.g., by querying dpkg db or doing a *stat)*
2. Compare to desired state
3. If there are any discrepancies, fix them (or log)

reductive labs

A Simple Transaction

```
file { "/tmp/testing":
     content => "Something", mode => 755
}
```

reductive labs

A Simple Transaction

```
file { "/tmp/testing":
     content => "Something", mode => 755
}
```

```
debug: //File[/tmp/testing]: Changing mode
debug: //File[/tmp/testing]: 1 change(s)
notice: //File[/tmp/testing]/mode:
     mode changed '644' to '755'
debug: Finishing transaction 10644270 with 1 changes
```

reductive labs

A Simple Transaction

```
file { "/tmp/testing":
    content => "Something", mode => 755
}
```

```
debug: //File[/tmp/testing]: Changing mode
debug: //File[/tmp/testing]: 1 change(s)
notice: //File[/tmp/testing]/mode:
    mode changed '644' to '755'
debug: Finishing transaction 10644270 with 1 changes
```

reductive labs

Configurations are
idempotent

reductive labs

Configurations are
idempotent

reductive labs

Logs go to syslog (by default)

You can configure clients to send simple reports to the server

reductive labs

An Irony

Puppet exposes **Change Management** as your next biggest problem

reductive labs

About the implementation

reductive labs

Written in Ruby

- 1 to 1 test code to real code (and pretty good coverage)
- Plugins are nearly always drop-in (resource types, providers, reports, etc.)

reductive labs

It's in production usage around the world right now

- Two sites managing more than 6,000 hosts
- Tens of other known sites
- Very active user community

reductive labs

Working toward 1.0

- Refactoring the network APIs so they're stable and maintainable
- Refactoring the Resource Abstraction Layer
- A few other tweaks here and there

reductive labs

Using Modules

```
luke@laeg(0) $ find . | grep -v .bzr
.
./templates
./templates/tracconfig.erb
./templates/tracsite.erb
./manifests
./manifests/init.pp
./manifests/tracsite.pp
./files
luke@laeg(0) $
```

reductive labs

Reductive Labs

- Open Source Software Startup
- Consulting, training, custom development, and support
- Other projects in process, e.g., change management and visualization
- My full-time job since 2005, two part time partners

Questions?

reductive labs