Using Puppet and Cobbler to Automate Your Infrastructure

Phillip J. Windley, Ph.D
Founder and CTO
Kynetx
www.kynetx.com

Sleeping Through the Night

Phillip J. Windley, Ph.D
Founder and CTO
Kynetx
www.kynetx.com

(afford|scal|reli)ability

hire fewer people

meet demand quickly

make fewer mistakes

imall
INCORPORATED

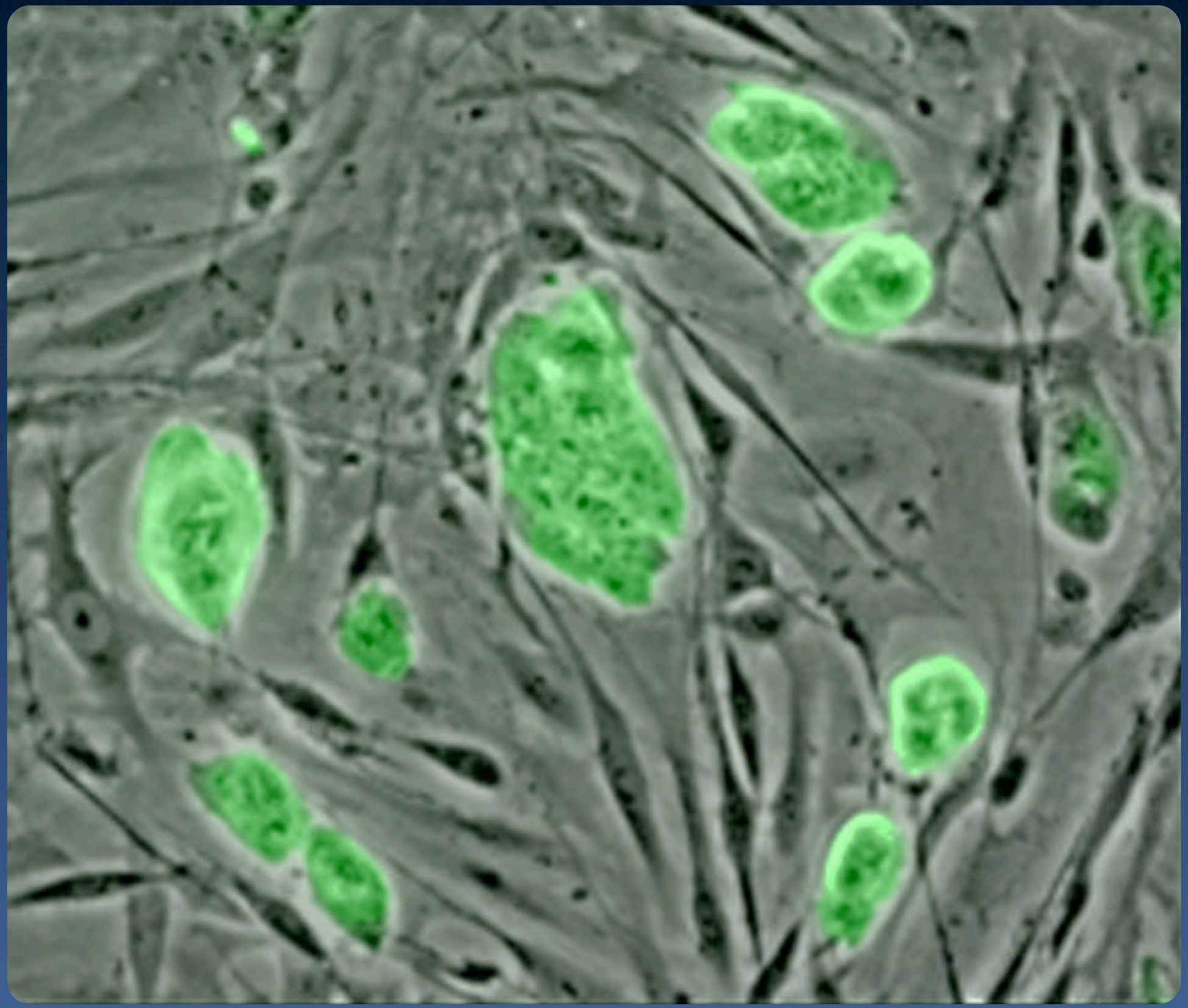
kynetx

this changes everything ™

1. machine provisioning

1. machine provisioning
2. system configuration

1. machine provisioning
2. system configuration
3. deployment



provisioning

machine provisioning

machine provisioning

- manage images & repositories

machine provisioning

- manage images & repositories
- kickstart machines

machine provisioning

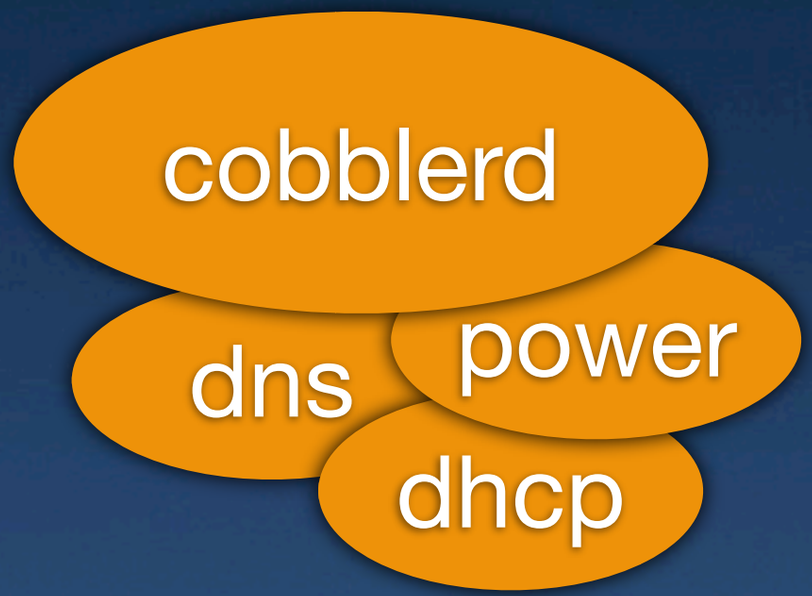
- manage images & repositories
- kickstart machines
- handle physical and virtual hardware

machine provisioning

- manage images & repositories
- kickstart machines
- handle physical and virtual hardware
- set up DHCP and DNS


cobbler

cobbler is a collection of tools that support machine provisioning


cobblerd
dns
power
dhcp

cobblerd

dns

power

dhcp

cobbler
web

images

repos

kickstart

koan

cobblerd

dns

power

dhcp

cobbler
web

repos

images

kickstart

cobbler uses a collection of specifications that define your systems

distro

profile

repo

distro

system

profile

repo

distro

import a distro

```
cobbler import --mirror ~/fc8 --name fc8
```

import a distro

```
cobbler import --mirror ~/fc8 --name fc8
```

create a profile

```
cobbler profile add --name=base-fc8
    --distro=fc8-xen-i386
    --kickstart=/root/base-fc8.ks
    --repos=fc8-newkey-repo
```

import a distro

```
cobbler import --mirror ~/fc8 --name fc8
```

create a profile

```
cobbler profile add --name=base-fc8
    --distro=fc8-xen-i386
    --kickstart=/root/base-fc8.ks
    --repos=fc8-newkey-repo
```

define a system

```
cobbler system add --name=log0
    --mac=00:16:3E:4B:40:00
    --ip=192.168.122.180 --profile=base-fc8
    --hostname=log0
```

building a machine

```
koan --server=cobbler.kobj.net --virt
  --nogfx --system=log0
```

configuration

system configuration

system configuration

- critical services on or off

system configuration

- critical services on or off
- security systems configured correctly

system configuration

- critical services on or off
- security systems configured correctly
- users created

system configuration

- critical services on or off
- security systems configured correctly
- users created
- necessary libraries in place

system configuration

- critical services on or off
- security systems configured correctly
- users created
- necessary libraries in place
- right packages built & installed

puppet is a language for specifying desired system configuration

install
package

install package

configure

install package

configuration should be modified **after** package installation

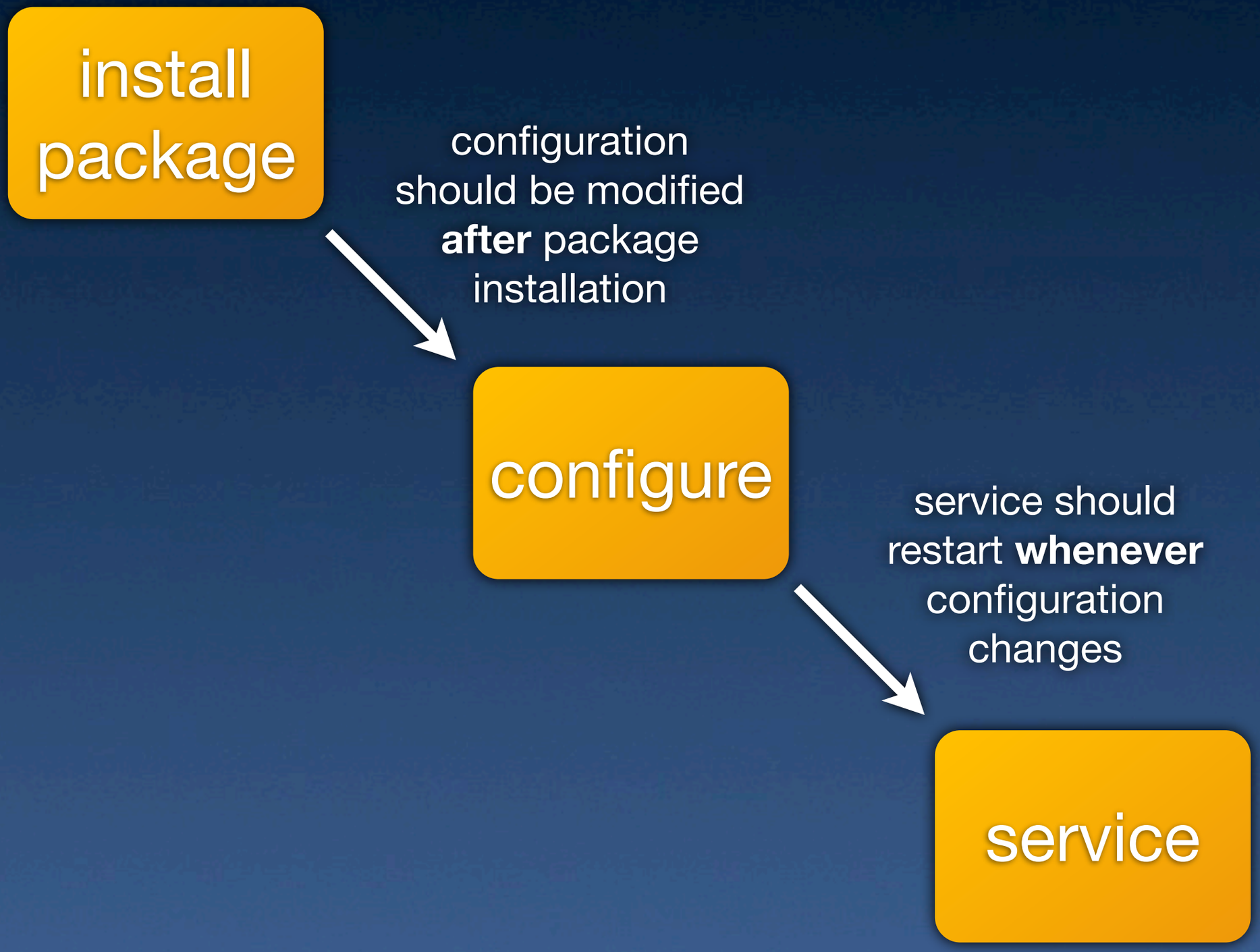
configure

install package

configuration should be modified **after** package installation

configure

service


install package
configuration should be modified **after** package installation
configure
service should restart **whenever** configuration changes
service

the hard way

```
yum install openssh-server
vi /etc/ssh/sshd_config
service sshd start
```

the puppet way

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

the puppet way

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

the puppet way

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

wait a minute…
that looks like a **lot**
more lines to me!

deployment

requirements

requirements

- deployment happens over & over again

requirements

- deployment happens over & over again
- controlled, not continuous

requirements

- deployment happens over & over again
- controlled, not continuous
- role-based

requirements

- deployment happens over & over again
- controlled, not continuous
- role-based
- remotable

now for deployment...

now for deployment...

now for deployment...

now for deployment...

in the end…
I just wrote it in Perl
in a few hours

```
[root@ops deploy]# ./deploy.pl -d

The following tasks are configured:
deploy              | Export a new copy of the code
install             | deploy, initialize, restart
uninstall           | rollback code, initialize,restart
start_httpd         | Start the HTTP server
rollback            | Rollback to the deploy
stop_httpd          | Stop the HTTP server
test_server         | Run the appropriate server test
cleanup             | Remove old copies of code
test_code           | Run the all tests
configure_httpd     | Build the httpd.conf file
install_init        | Install the init JS files
restart_httpd       | Restart the HTTP server
```

```
[root@ops deploy]# ./deploy.pl -s

 server                    | version
---------------------------|-----------------
 init0.kobj.net            | 340M
 init1.kobj.net            | 340M
 log.kobj.net              | 340
 log0.kobj.net             | 340
 log1.kobj.net             | 340
 krl.kobj.net              | 340
 cs0.kobj.net              | 341
 cs1.kobj.net              | 341
 cs2.kobj.net              | 341
 cs3.kobj.net              | 341
```

```
[root@ops deploy]# ./deploy.pl -m krl -t install

Performing install on krl with role krl...
A    /web/lib/releases/perl_0910091229/ops
...
A    /web/lib/releases/perl_0910091229/startup.pl
A    /web/lib/releases/perl_0910091229/Kynetx.pm
A    /web/lib/releases/perl_0910091229/README
Checked out revision 342.
Writing /web/conf/httpd.conf
Stopping httpd: [  OK  ]
Starting httpd: [  OK  ]
Testing RuleManager.....ok
All tests successful.
Files=1, Tests=73, 8 wallclock secs ...
Result: PASS
```

TODO

TODO

- configuration database

TODO

- configuration database
- (more) automated testing

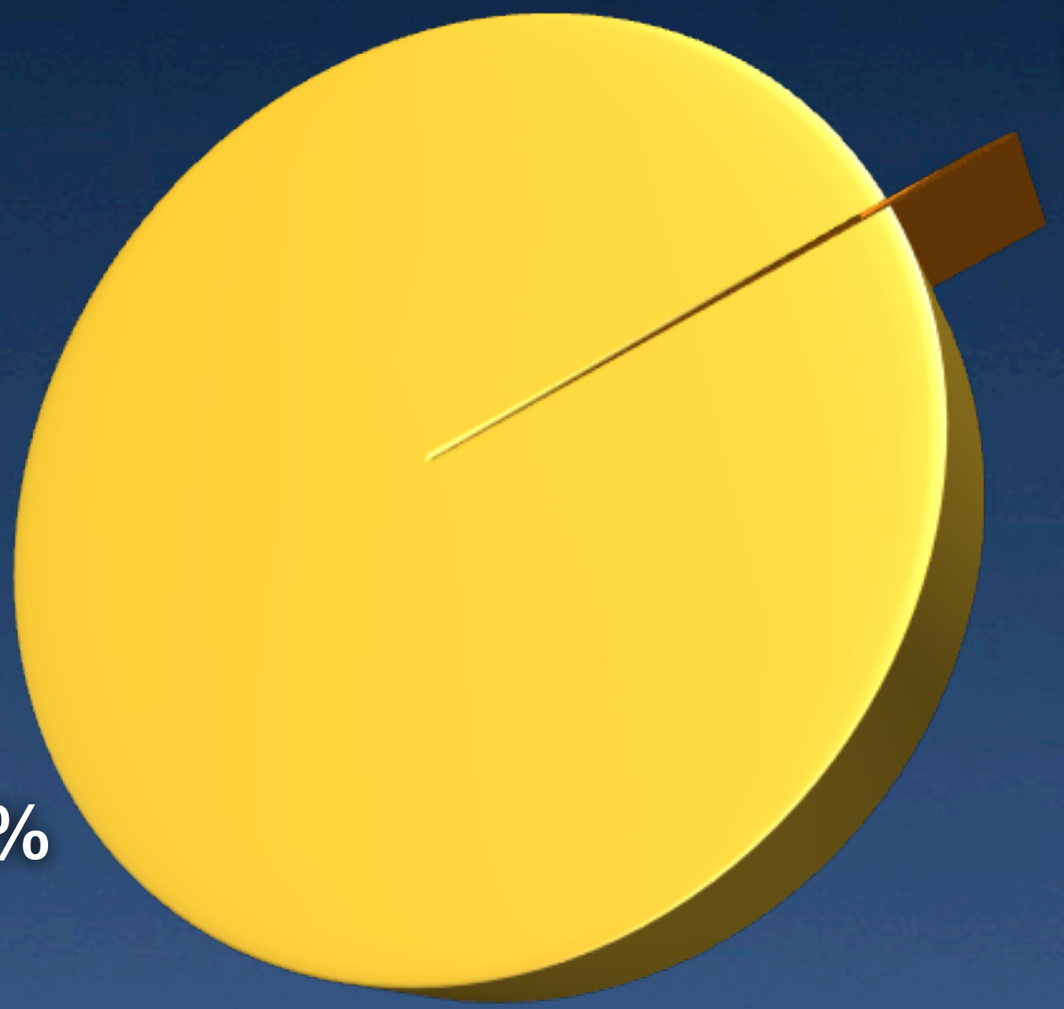
TODO

- configuration database
- (more) automated testing
- continuous integration

results

kynetx can stand up a new server in < 30 minutes

our servers stay up



downtime*
0.00229%

uptime
99.99772%

* includes scheduled maintenance

Warning!


Warning!

lessons learned

lessons learned

- architect for (afford|scal|reli)ability

lessons learned

- architect for (afford|scal|reli)ability
- insist on consistency & repeatability

lessons learned

- architect for (afford|scal|reli)ability
- insist on consistency & repeatability
- document process with code

lessons learned

- architect for (afford|scal|reli)ability
- insist on consistency & repeatability
- document process with code
- rolling releases and change control

lessons learned

- architect for (afford|scal|reli)ability
- insist on consistency & repeatability
- document process with code
- rolling releases and change control
- put ops procedures online

learning more

- Introduction to Cobbler
  - Derek Carter 2:30
- Puppet Workshop
  - Andrew Shafer 3:00
- Managing your minions with func
  - Daniel Hanks 3:45
- Cobbler power tools
  - Derek Carter 5:00

kynetx

IMPACT CONFERENCE 2009

Nov 18-19, 2009,
Provo UT

kynetx

IMPACT CONFERENCE 2009

Nov 18-19, 2009,
Provo UT

Use discount code
Windley50
www.kynetx.com

Sleeping Through the Night

Contact info:

pjw@kynetx.com
www.windley.com
@windley

FREE
Context Automation
White Paper
at Kynetx Booth

Sign up free: http://www.kynetx.com/signup