

# Cobbler and Puppet

Controlling your server builds

Eric Mandel and Jason Ford  
BlackMesh



# Agenda

- Introductions
- The Saga of Server Builds
- Cobbler
- Puppet
- The Ease of Server Builds
- The Business Benefits
- Questions/Discussion

# Brief Introduction of Us

- Eric Mandel, BlackMesh
- Jason Ford, BlackMesh
  
- Managed Hosting Service Provider located in Northern Virginia outside of Washington, DC
- BlackMesh works with development companies, ASPs, SaaS providers, small businesses, large corporations

# Brief Introduction of You

- System Administrators?
- Developers?
- Currently use Cobbler?
- Currently use Puppet?

# The Saga of Server Builds

- We found ourselves always building new servers
- Most used the same applications, but configured differently



# The Saga of Server Builds

- A time consuming, repetitive, tedious task
- Time is money



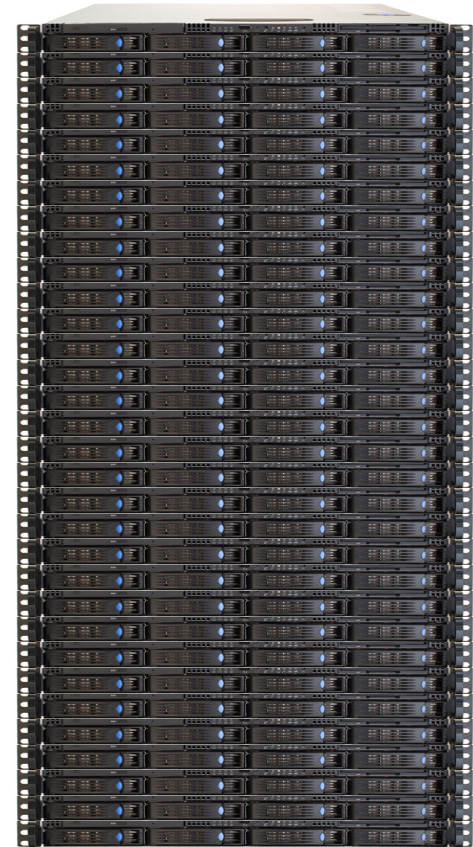
# The Saga of Server Builds

- A time consuming, repetitive, tedious task
- Time is money
- Lots of shell scripts were written



# The Saga of Server Builds

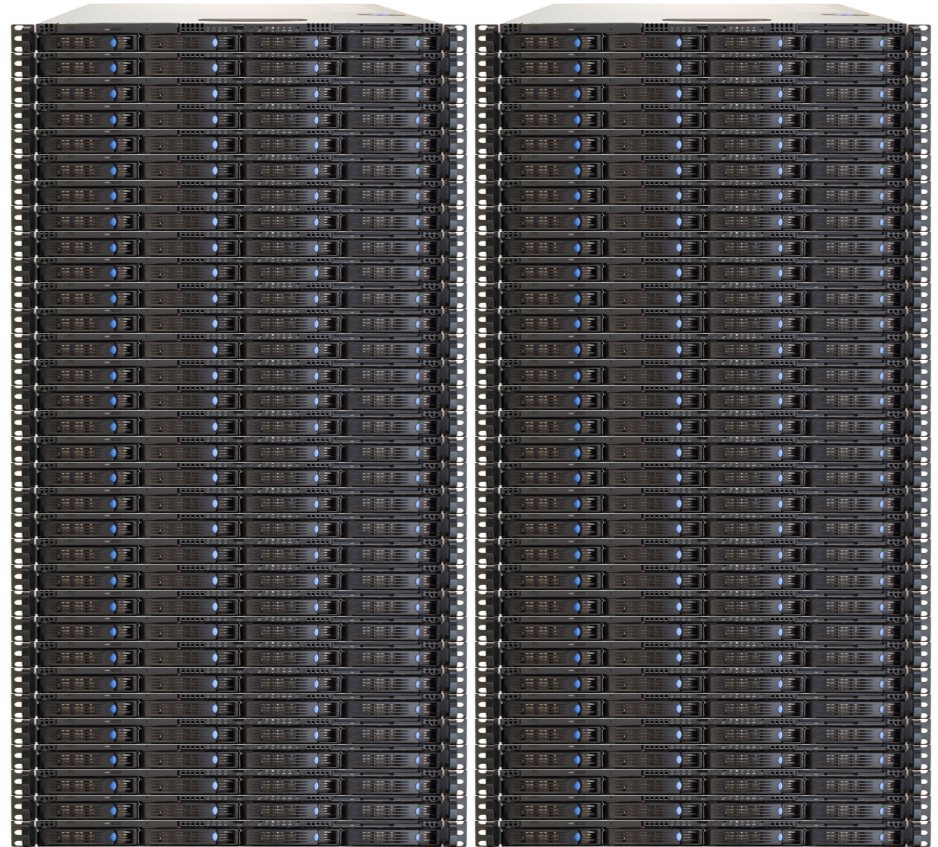
- A time consuming, repetitive, tedious task
- Time is money
- Lots of shell scripts were written
- Temporary is permanent



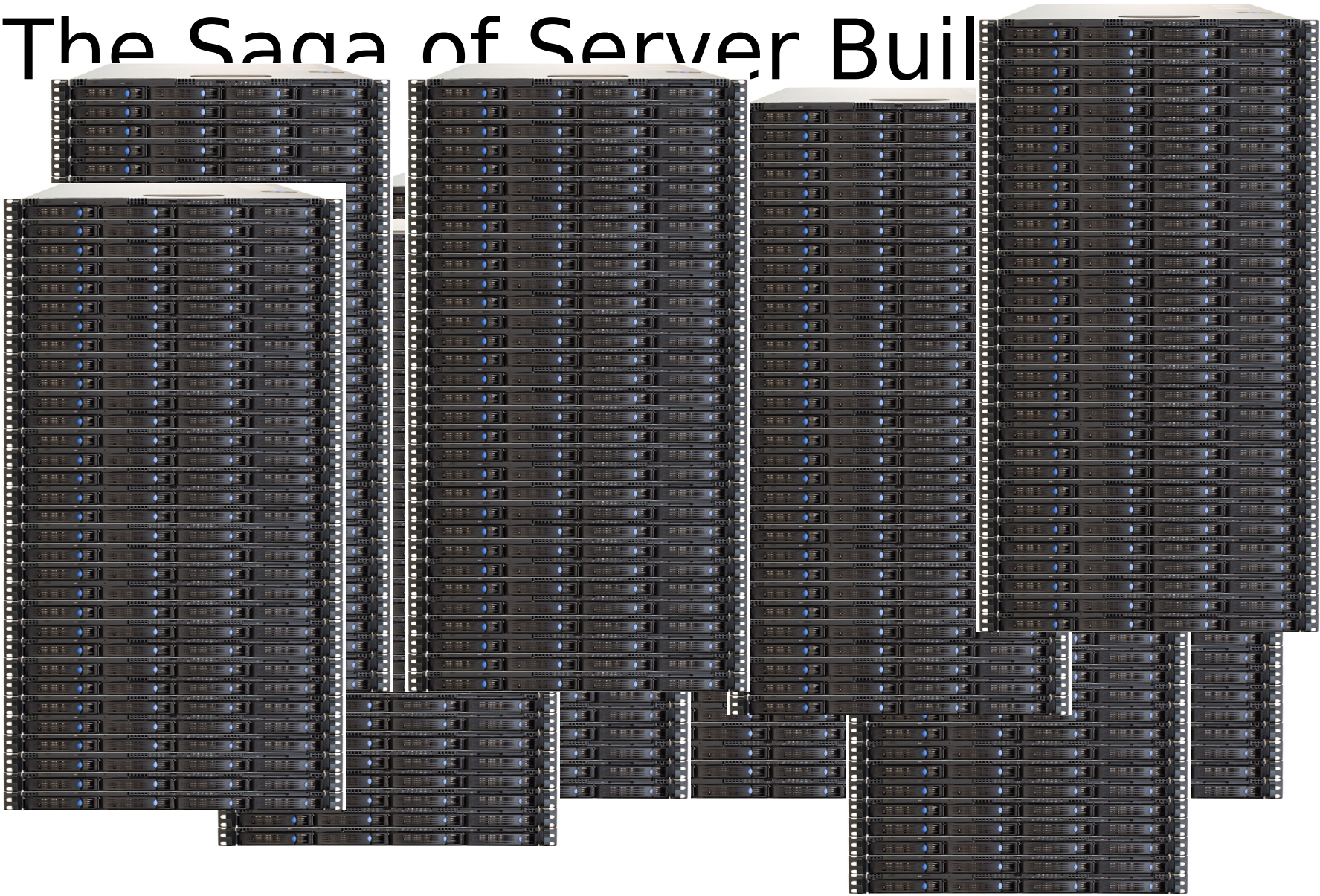


# The Saga of Server Builds

- A time consuming, repetitive, tedious task
- Time is money
- Lots of shell scripts were written
- Temporary is permanent



# The Saga of Server Build



OSCON, July 2009

# The Saga of Server Builds

- New server deployments can be time-critical
- Ensuring that all of the packages are the latest versions
- We did not want to reinvent the wheel
- Need to know what we are deploying is same as what is out there

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added.

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.
- Applications get installed



# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.
- Applications get installed, get upgraded

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.
- Applications get installed, get upgraded, get uninstalled

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.
- Applications get installed, get upgraded, get uninstalled, and reinstalled and reconfigured

# Maintenance

- Over time, things happen
- Undocumented changes are put in place to resolve a urgent problem
- Users get added, and then deleted, and then re-added. And usually modified.
- Applications get installed, get upgraded, get uninstalled, and reinstalled and reconfigured
- Configuration files change

# Cobbler & Puppet to the rescue



# Cobbler

- Disk imager on steroids
  - Red Hat's Kickstart
  - Sun's Jumpstart
  - Norton's Ghost
- Admin interface to Kickstart
- Hardware aware
- <https://fedorahosted.org/cobbler/>



# Cobbler

- Brings efficiencies to our process:
  - no longer have a three step process of install OS, install repositories, and then yum update
  - Local repository, kept up to date with custom drivers
  - Automatically installs the standard applications (Apache, MySQL, PHP, etc.)
  - Automatically applies our security procedures (users, services, etc.)

# Cobbler

- You must connect the new server to the Cobbler server
  - This can be over network
  - Or with direct CAT5 cable
- Configure server to boot from Cobbler via PXE or static IP
- Builds server to point where Puppet can take over with Kickstart



# Puppet

- From the Puppet Wiki:

“Puppet is a **declarative language** for expressing system configuration, a **client and server** for distributing it, and a **library** for realizing the configuration.”

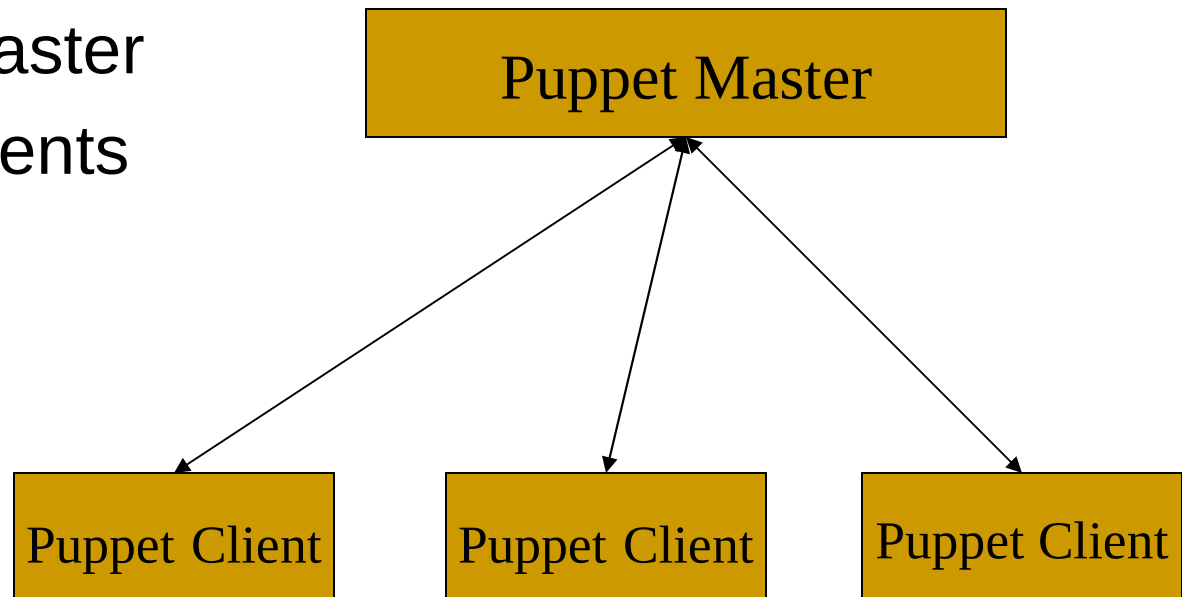
- <http://reductivelabs.com/trac/puppet>

# Puppet

- Written in Ruby
  - Very OO
  - Also uses **Facter**, a Ruby program to determine system information and parameters
- Active community
  - FOSS
  - Shared modules (Recipes from the Cookbook)
  - Community is growing

# Puppet

- Client/Server
  - Puppet Master
  - Puppet clients



# Puppet

- Three main pieces:
  - declarative language
  - client and server
  - library
  
- Main concept:
  - idempotent

# Puppet

- Three layers
- Each responsible for separate aspects of the system
- Providers

Configuration Language

Transaction Layer

Resource Allocation Layer

# Puppet

- Base unit is a resource
- Resource types are:
  - File
  - User
  - Package
  - Service
  - Cron
  - Mount (file system)
  - The “exec” resource allows you to create your own resources

# Puppet

## ■ File resource type:

```
# Make sure the modes on the passwd file
are correct
file { "/etc/passwd":
    owner => "root",
    group => "root",
    mode  => 644
}
```

# Puppet

## ■ User resource type:

```
# Make sure the user blackmesh is on the
server
user { "blackmesh":
    ensure => present,
    password => "secretpass"
}
```



# Puppet

## ■ Package resource type:

```
# Make sure the httpd package is
installed
package { "httpd.${architecture}" :
  alias => "httpd",
  provider => "yum",
  ensure => "present",
  require => Exec["remi-enabled"]
}
```

# Puppet

- Resources are combined together in classes:

```
class httpd {
  package { "httpd.${architecture}" :
    alias => "httpd",
    provider => "yum",
    ensure => "present",
    require => Exec["remi-enabled"]
  }
  service { "httpd":
    ensure => running,
    hasstatus => true,
    require => Package["httpd"]
  }
  exec { "chkconfig_httpd":
    command => "/sbin/chkconfig --level 2345 httpd on",
    require => Package["httpd"]
  }
}
```

# Puppet

## ■ Classes can be combined together:

```
class base {  
    include sudo,  
            yum_exclude_kernel,  
            epel_enabled,  
            remi_enabled,  
            yum-updatesd,  
            denyhosts,  
            sshd_config,  
            sshd,  
            bestyum,  
            yum_update,  
            ntpd,  
            user_blackmesh,  
}
```

# Puppet

- **Nodes apply classes:**

```
node cust115_webserver {  
    include base, httpd  
}
```

# Puppet

- Providers implement resources on a local level
  - Red Hat-based: `useradd`
  - BSD-based: `adduser`
- Actions can cause new actions based on `notify` and `subscribe` properties
- Logs everything, so have a record, but not transactional

# Puppet

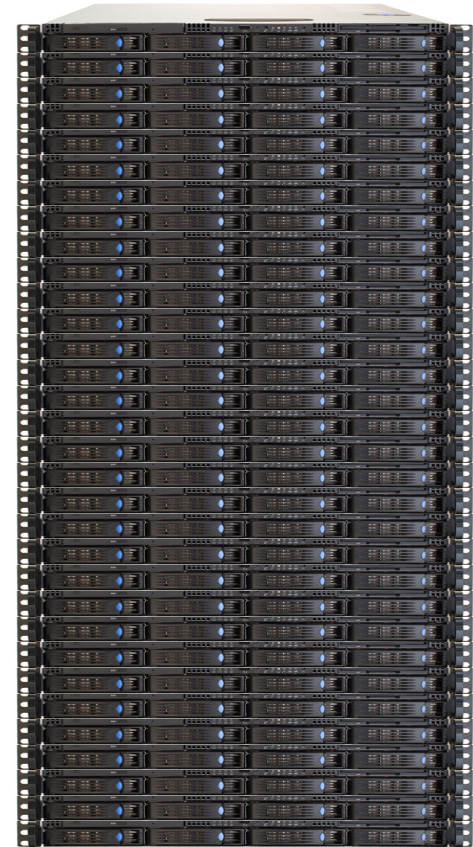
- Not transactional
- The run files are not processed sequentially
- Require statements very, very important
- Does not guarantee state at end of any run

# Puppet Implementation

- Install PuppetMasterd
- Configure sites for the Puppet config
- Install Puppet on client
- Create the puppet certificate on the client
- Sign the puppet certificate on the master
- Run puppet

# The Ease of Server Builds

- Entirely scripted
- New hardware to server on network in 10 minutes
- All servers built the same no matter who builds them





# The Ease of Maintenance

- Ensure consistency across servers:
  - User exists
  - Apache configuration
  - MySQL configuration
  - Security rules (who can ssh, etc.)
  - Watch for new accounts
  - Verify cron jobs are running; warn us about failures
  - Verify backups are running
  - Verify logrotate is working
  - Watch for excessively large files

# The Business Benefits

- Proactive maintenance
- Confidence knowing things are routinely verified
- Time
- Lower costs
- Repeatable
- Documented
- Ease of maintenance

# Web resources

- Cobbler

- <https://fedorahosted.org/cobbler/>

- Puppet

- <http://reductivelabs.com/trac/puppet>

---

# Summary

# Cobbler and Puppet Rock

# Questions

?

Eric Mandel, [emandel@blackmesh.com](mailto:emandel@blackmesh.com)

Jason Ford, [jford@blackmesh.com](mailto:jford@blackmesh.com)