Puppet

An Introduction

OlinData

About Me

```
user { "rajkissu":
    name     => "Raj Kissu Rajandran",
    employer => "OlinData",
    role     => "Trainer / Engineer",
    contact  => {
        "email"   => [ "rajkissu@gmail.com", "raj@olindata.com" ],
        "twitter" => "@rajkissu",
        "skype"   => "rajkissu"
    },
    github   => "https://github.com/rajkissu",
    gravatar => "http://gravatar.com/rajkissu"
}
```

OlinData

- Founded in 2008
- Open source consulting & training
- Remotely run, location independent

Puppet experts

- Official Puppet Labs partner
- Experience with small, medium and large infrastructures

Other Technologies

- Drupal, MySQL
- Acquia authorized partner
- More technologies to come!

What is Puppet?

- IT automation software for system administrators
- Developed by Puppet Labs
- Open source & commercial version – Puppet Enterprise
- Easy to learn
- Strong community backing
- Support for major operating systems

Puppeteers

Why Puppet?

The Ol' Days

Manually Configure

Golden Images

All-or-Nothing Software Packages

Custom One-off Scripts

Enter Puppet ...
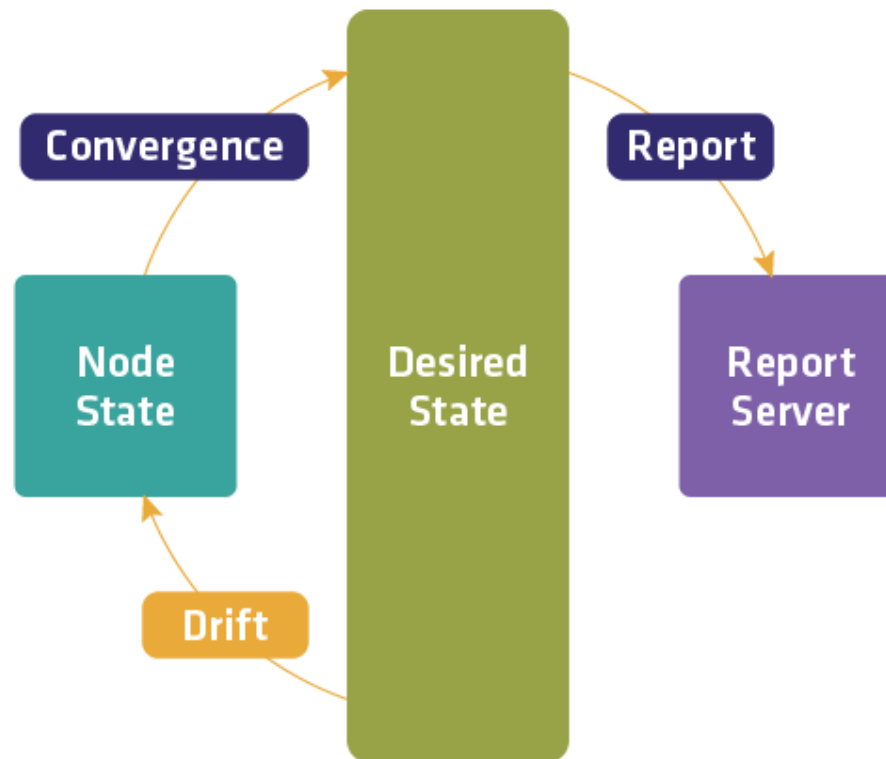
	Current State	With Puppet
Productivity	50-100 nodes/admin	100s-1000s nodes/admin
Provisioning	Days/Weeks	Minutes/Hours
Configuration Drift	Ever-Present	Eliminated
Visibility	Limited, if any	All Changes
Software Releases	Weeks or months	Days, even hours

How Puppet Works

1. **Define** the desired state of the infrastructure's configuration using Puppet's declarative language.

2. **Simulate** configuration changes before enforcing them.

3. **Enforce** the deployed desired state automatically, correcting any configuration drift.

4. **Report** on differences between actual and desired states, and any changes made enforcing the desired state.

Iterate & Increase Coverage

Current State   Desired State

Configuration Drift

Puppet Basics

A Simple Scenario
You need to manage a user

Some things to think about:

- user's existence
- primary group
- home directory

Concerns

Platform idiosyncrasies

- What OS are we working with?
- Does it use useradd or adduser?

Command differences

- What's the difference between -l and -L?

What will happen if I run the command again?

DIY

```sh
#! /bin/sh
USER=$1; GROUP=$2; HOME=$3
if [ 0 -ne $(getent passwd $USER > /dev/null)$? ]
then useradd $USER --home $HOME --gid $GROUP -n; fi

OLDGID=`getent passwd $USER | awk -F: '{print $4}'`
OLDGROUP=`getent group $OLDGID | awk -F: '{print $1}'`
OLDHOME=`getent passwd $USER | awk -F: '{print $6}'`

if [ "$GROUP" != "$OLDGID" ] && [ "$GROUP" != "$OLDGROUP" ]
then usermod --gid $GROUP $USER; fi

if [ "$HOME" != "$OLDHOME" ]
then usermod --home $HOME $USER; fi
```
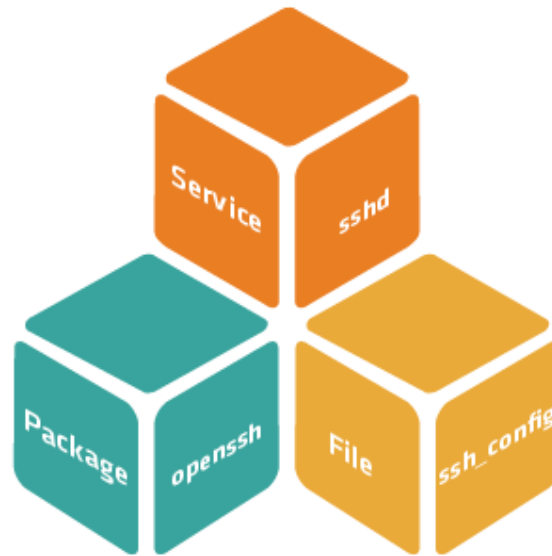
But what about...

- Robust error checking?
- Solaris and Windows support?
- Robust logging of changes?
- Readable code?

And managing users is *easy*.

Resources

Resources

- Basic building blocks
- Represents system assets

```
user { "elvis":
    ensure     => present,
    gid        => 'sysadmin',
    home       => '/mnt/home/elvis',
    managehome => true
}
```

Declarative Modelling Language

- Model the desired state
- Let Puppet figure out how to get it done

Imperative

```
if [ 0 -ne $(getent passwd elmo > /dev/null)$? ]
    useradd elmo --gid sysadmin -n
then fi

GID=`getent passwd elmo | awk -F: '{print $4}'`
GROUP=`getent group $GID | awk -F: '{print $1}'`

if [ "$GROUP" != "$GID" ] && [ "$GROUP" != "sysadmin" ]
    usermod --gid $GROUP $USER
then fi
```
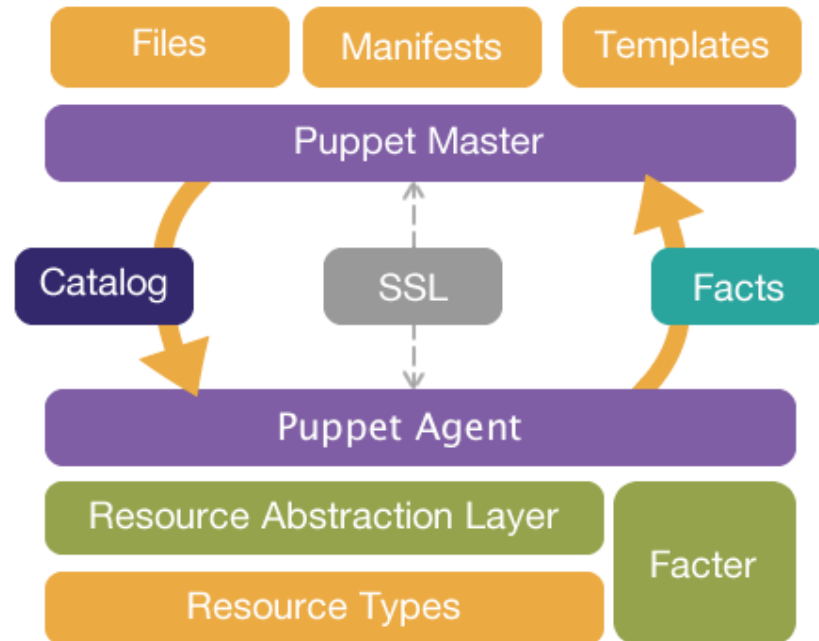
Declarative

```
user { 'elmo':
    ensure => present,
    gid    => 'sysadmin'
}
```
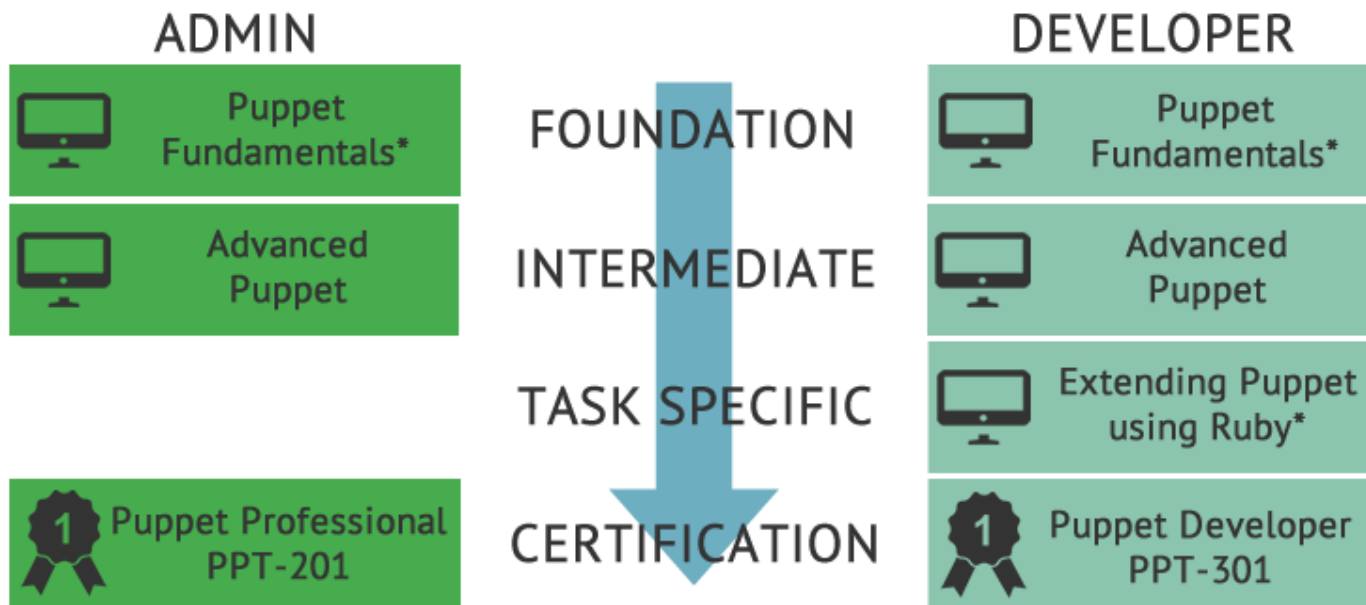
Master - Agent Setup

Demo

What To Do Next ...

Self Learning

- Tutorials **http://docs.puppetlabs.com/learning/**

- Learning VM **http://info.puppetlabs.com/download-learning-puppet-VM.html**

Extending Puppet

- Puppet Enterprise **http://info.puppetlabs.com/download-pe.html**

- Online Documentation **http://docs.puppetlabs.com/**

- Puppet Forge **http://forge.puppetlabs.com/**

Training Resources

ADMIN

	FOUNDATION
Puppet Fundamentals*	
Advanced Puppet	INTERMEDIATE
	TASK SPECIFIC
Puppet Professional PPT-201	CERTIFICATION

DEVELOPER

Puppet Fundamentals*

Advanced Puppet

Extending Puppet using Ruby*

Puppet Developer PPT-301

* recommended preparation for Certification Exam

http://olindata.com/training/upcoming/

We have worked & trained with

KL Puppet Fundamentals
September 9 - 11 2013

http://www.amiando.com/puppet-training-kualalumpur-sep-2013.html

Q & A

Thank You!

Raj Kissu

raj@olindata.com

http://www.olindata.com

http://fb.com/olindata

http://twitter.com/olindata