Introduction to Puppet

Paul Waring (paul@xk7.net, @pwaring)

June 21, 2014

Configuration management and provisioning

- Define how a machine should be setup
- Configuration, software installed, users etc.
- Manage large numbers of machines - especially identical ones
- Quick redeployment of server
- Development matches production
- Ensure a consistent state, even if local edits made

What is Puppet?

- Configuration management and provisioning
- Apache 2.0 licence since v2.7
- Declarative vs imperative
- Describe desired state of server, Puppet makes it so

Puppet Labs

- Commercial company behind the software
- Enterprise platform available
- Support, training, conferences etc.

Support

- Community: mailing lists, IRC etc.
- Commercial: Puppet Labs, contractors etc.

Alternatives

- Ansible
- Chef
- cfengine

Why Puppet?

- Large ecosystem and community
- Lots of documentation (wikis, books etc.)

Why not Puppet?

- Requires an agent on all machines
- Extra firewall rules
- Bootstrapping problem
- You hate Ruby

Manifests

- Describe how a system should be configured
- Plain text files, Ruby syntax
- Write manifests once, run anywhere (mostly)

Vagrant Provisioning

- Start up a VM and configure it automatically
- Will be used in all examples

Resources

- Basic building blocks of manifests
- Standard types: `package`, `exec`, `service` etc.
- Define your own resource types
- Third party resource types: `mysql`, `apache` etc.

Generic example

```
resource_type { "identifier":
  attribute1 = value,
  attribute2 = value,
}
```

Package resource

Controls packages installed on the system.

Attributes

- `name`: Name (from package management system), defaults to identifier
- `ensure`: What state the package should be in

Examples

```
package { "nethack-common":
  ensure = present,
}

package { "php5":
  ensure = absent,
}
```

Exec resource

Execute specific commands which are not represented by resources (e.g. there is no ‘compressed’ resource type).

Examples

```
exec { "unpack_moodle_db":
  unless = "/usr/bin/test -f /home/vagrant/moodle.sql",
  command = "/bin/gunzip /home/vagrant/moodle.sql.gz",
}

exec { "unpack_moodle_code":
  cwd = "/home/vagrant/www/moodle2/htdocs",
  command = "/bin/tar --strip-components=1 \
   -xzf /home/vagrant/moodle-2.2.11.tgz",
}
```

Service resource

Ensure services are running (or not).

```
service { 'apache2':
  ensure = running,
  enable = true,
}
```

MySQL resource

Optional resource made available by Puppet Labs.

```
puppet module install puppetlabs-mysql
```

```
mysql_user { 'puppet@localhost':
  ensure = present,
}

mysql_database { 'puppet':
  ensure = present,
}

mysql_grant { 'puppet@localhost/puppet.*':
  ensure = present,
  options = ['GRANT'],
  privileges = ['ALL'],
  table = 'puppet.*',
  user = 'puppet@localhost',
}
```

Resource ordering

Occasionally resources need to be processed in a particular order which Puppet cannot determine.

Examples

```
file { "/home/vagrant/moodle-latest-26.tgz":
  ensure = present,
  source = "/vagrant_data/moodle-latest-26.tgz",
  before = Exec["unpack_moodle_code"],
}

exec { "unpack_moodle_code":
  cwd = "/home/vagrant/www/moodle2/htdocs",
  command = "/bin/tar --strip-components=1 \
    -xzf /home/vagrant/moodle-latest-26.tgz",
}
```

Questions

- Slides and scripts on GitHub under BSD Licence
- https://github.com/pwaring/puppet-talk