IF BSI 07: Automated deployment with puppet

Patricia Jung
Informatica Feminale 2014

Automated deployment

Tasks

- Make sure hosts on OS level are always (not only after initial setup) configured according to specification
- Make sure applications/changes are reliably rolled out
- Make sure monitoring is being set up and configured appropriately
- Make sure setups are properly documented

Classical manual system administration

Problems:

- cannot ensure identical setups
- does not scale
- uhm... documentation

Solution

Configuration management, e.g. using Open Source software like

- Ansible (http://www.ansible.com/)
- Chef (http://www.getchef.com/)
- Cfengine (http://www.cfengine.com/)
- Puppet (http://www.puppetlabs.com/)
- SaltStack (http://www.saltstack.com/)
- ...

The Puppet software

- Dual license: Open Source and extended enterprise product
- Open Source edition (Apache 2.0 license, GPLed until v2.7) ships with most Linux distros, FreeBSD, Solaris from 11.2, Amazon EC2 (Linux AMI), ...
- Available for Windows, MacOS, AIX, HP-UX, ...
- Supports Nagios and compatible monitoring frameworks
- Certification programme

Puppet user interfaces

- Uses plain text configuration languages which can easily be stored in version control systems
- User interaction via command line (CLI)
- Browser user interfaces (BUIs) available in Puppet Enterprise version
- Declarative configuration management language
- Ruby to extend functionality
- YAML/JSON for storing node/site-specific data in Hiera

Helpers and documentation

- Reference: http://docs.puppetlabs.com/puppet/3/reference/
- Modules: https://forge.puppetlabs.com/
- Style checker: puppet-lint
- VIM support: vim-puppet

Puppet architecture

- **Master:**
  - takes care of set-up descriptions
  - runs as daemon, usually on port 8140
- **Agents:**
  - provide master with system-specific informations ("facts")
  - usually contact master to receive setup instructions
  - ... and obey them locally
- **Communication:** always SSL encrypted
- **Resource Abstraction Layer (RAL):** provides independence from OS/distribution specifics

Puppet (basic) commands

```
# puppet <subcommand> <options and args>
```

```
# [FACTERLIB=<path>] facter [-p] [<other options>][<fact>]
```

Puppet command examples

```
# puppet help

# puppet help resource

# puppet resource user

$ facter -h

$ facter

$ facter osfamily

# puppet master --verbose --no-daemonize --logdest /var/log/puppet/master.log

# puppet agent --server $(facter fqdn) --waitforcert 60 --onetime --noop

# puppet cert sign host.example.com

# puppet agent --server $(facter fqdn) --onetime --logdest /var/log/puppet/agent.log

# puppet master --genconfig | grep ssl

# puppet node clean host.example.com

# puppet module search stdlib
```

- puppet agent option --noop to test your code (won't change your system)
- puppet agent option --server <master> can be omitted if puppet master is running on puppet.<agent's domain>

The puppet configuration description language

```
$ puppet resource user root
user { 'root':
  ensure  => 'present',
  comment => 'root',
  gid     => '0',
  home    => '/root',
  shell   => '/bin/bash',
  uid     => '0',
}
```

Caution: puppet resource translates existing set-up into description language

Puppet manifests

```
/etc/puppet
|-- files
|-- manifests
|   |-- ...
|   `-- site.pp
|-- modules
|   |-- ...
...
|-- puppet.conf
...
`-- templates
```

Puppet modules

```
|-- modules
|   |-- <module_name>
|   |   |-- files
|   |   |   `-- ...
|   |   |-- lib
|   |   |   |-- facter
|   |   |   |   `-- <fact>.rb
|   |   |   ...
|   |   |-- manifests
|   |   |   |-- <class>.pp
|   |   |   |-- <class>
|   |   |   |   |-- <subclass>.pp
|   |   |   ... `-- <subclass>.pp
|   |   `-- templates
|   |       |-- <filename>.erb
... ...     ...
```

A simple configuration in a master-node setup

/etc/puppet/manifests/site.pp:

```
node 'host.example.com' {
    package { [ 'puppet-lint', 'vim-puppet' ] :
        ensure => 'latest',
    }
}
```

```
master# puppet master --verbose --no-daemonize --logdest /var/log/puppet/master.log

agent# puppet agent --server $(facter fqdn) --waitforcert 60 --onetime --logdest /var/log/puppet/agent.log

master# puppet cert sign host.example.com
```

- Certificate signing only after first contact of agent with master or when a new certificate was generated
- Always check your .pp files with puppet-lint

Facts, functions and conditionals

/etc/puppet/modules/puppettools/manifests/base.pp:

```
class puppettools::base {

    case $::osfamily {
        debian:  {
            package { [ 'puppet-lint', 'vim-puppet' ] :
                ensure => 'latest',
            }
        }
        default: { err("$::osfamily is currently not supported.") }
    }
}
```

/etc/puppet/manifests/site.pp:

```
node 'host.example.com' {
    include 'puppettools::base'
}
```

- Refer to facts with $::<fact> syntax
- $ prefix denotes a variable (in fact an invariable)
- :: denotes the root namespace (a fact is a "variable" in the root namespace)