AUTOMATING BASIC IT FUNCTIONS

Matthew Campbell
Automating basic IT functions

• Background
  – Unix OS design
  – Animation system hardware
  – Process automation
  – Enterprise scale

• The environment
  – The home router: modem, firewall, dhcp, named, wap, etc.
  – Home IT services: file sharing, media, environment, ddns+http, etc.
  – Costs of automation: bare metal or virtual? micro compute?
Automating basic IT functions

• Automation is designed to make it easier to configure and maintain hundreds or thousands of servers

• Major automation systems: Chef, Puppet, Ansible, Salt
  – Chef and Puppet are based on Ruby
  – “Puppet arguably enjoys the biggest mindshare” (at time of printing)
  – “Salt scales through delegation, Puppet through self-service”
  – “Puppet and Chef appeal to developers while Salt and Ansible are more attuned to Sysadmins”
Automating basic IT functions

• Common aspects of automation frameworks
  – Define “Infrastructure as Code”
  – Client / Server architecture
  – SSH communications
  – Certificate based authentication and encryption
  – Version control of configuration definitions
  – Can be extended by downloading modules
  – Operated through a Web UI and/or CLI
  – Cross platform solutions
Why automate your home IT?

- Good experience for your resume (home === enterprise)
- Reliability, Recovery, and End of Life
- Ease of maintenance and support
- Central definition for IP security
- Technology currency
- Documenting state and process
INSTALLATION – THE PUPPET EXAMPLE
Automating basic IT functions

Installation

- **Install the puppetmaster service**
  
  ```
  sudo rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
  sudo dnf install --y puppet-server
  ```

- **Install the puppet agent**
  
  ```
  sudo dnf install --y puppet
  ```

- **Firewall port 8140**

- **Configure the server**

- **“Node” Authentication**

- **Running an update manually**

- **Scheduling continuous updates**
Automating basic IT functions

- **Modules**
  - Manifests
    - Classes vs. Defines
      - include linux::users
        - linux::administrators { 'matt': uid => 1000, shell => '/bin/tcsh' }
        - linux::administrators { 'lan': uid => 1010, shell => '/bin/bash' }
  - Files
  - Templates
  - Metadata.json

- **Environments**

- **Puppetforge™**, Puppet documentation and Puppet Types
Automating basic IT functions

/etc/puppet/manifests/site.pp

node default {
    include common
}

node 'fileserver.myhome.com' {
    include common
    include nfs_server
}

node 'services.myhome.com', 'backupservices.myhome.com' {
    include common
    include dhcp_server
    include ntp_server
    include ddns_server
}
Automating basic IT functions

- `/etc/puppet/environments/production/modules/<name>/manifests/init.pp`

  # Definition: linux
  #
  # Description: Manage software and services common to all linux hosts
  #
  # Usage:
  #  include linux
  #
  # Options:
  #  None

  class linux {
    # Create common linux users
    include linux::users
    # Install default linux packages
    include linux::packages
  }
Automating basic IT functions

**Examples**
- Webserver/init.pp
- webserver::files.pp
- webserver::imageutils.pp
- firewall::templates/service.xml.erb
- /usr/lib/firewall/services/misterhouse.xml
- house::init.pp

https://docs.puppetlabs.com/references/latest/type.html#file

https://forge.puppetlabs.com/


References