

Matthew Campbell

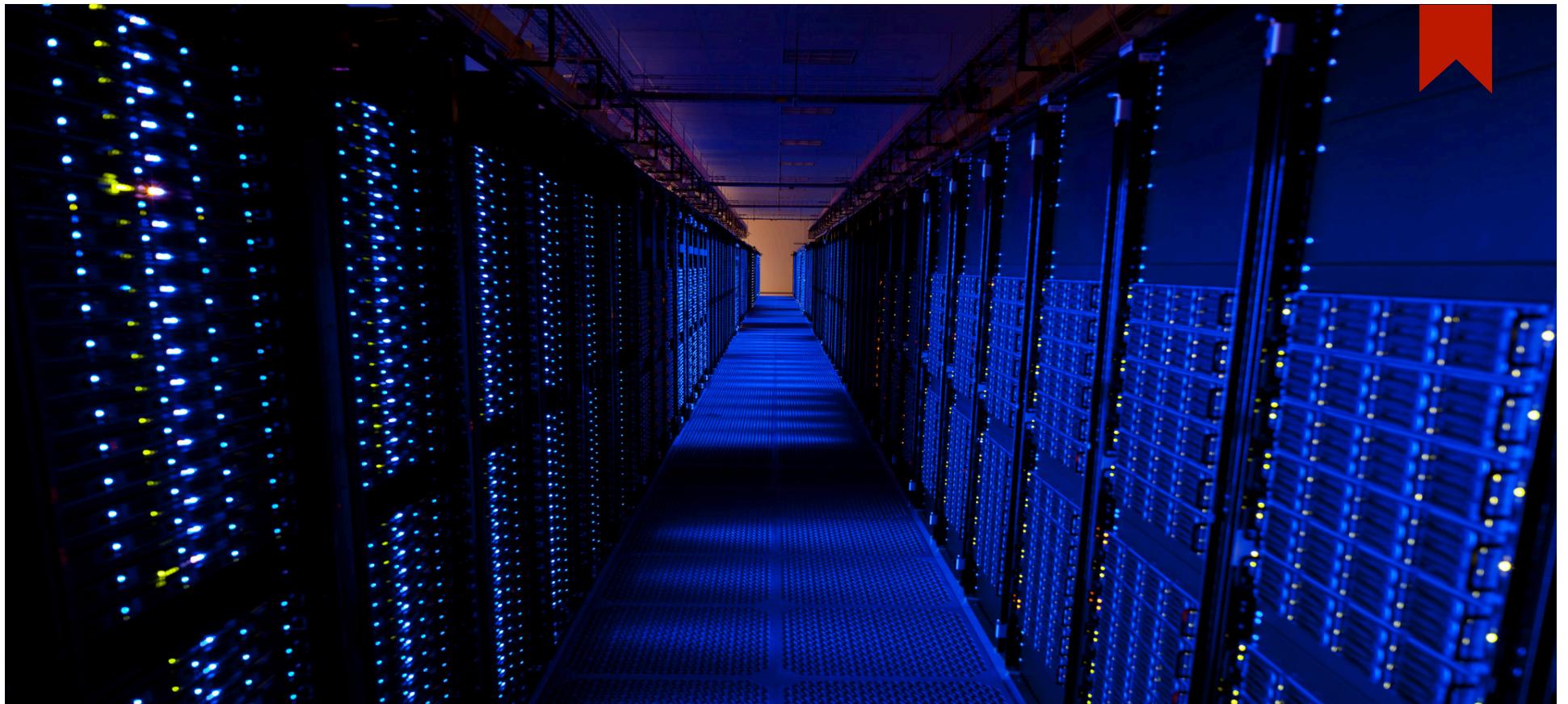
## AUTOMATING BASIC IT FUNCTIONS

- 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?

- 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”

- 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

- 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

- Modules
  - Manifests
    - Classes vs. Defines

```
include linux::users
linux::administrators { 'matt': uid => 1000, shell => '/bin/tcsh' }
linux::administrators { 'ian': uid => 1010, shell => '/bin/bash' }
```
    - Files
    - Templates
    - Metadata.json
  - Environments
  - Puppetforge™, Puppet documentation and Puppet Types

## Automating basic IT functions

## Configuration

```
/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
}
```

- /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
}
```

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

[http://mattswiki.seilcampbells.com/index.php5?title=Automation\\_Server](http://mattswiki.seilcampbells.com/index.php5?title=Automation_Server)

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

<https://forge.puppetlabs.com/>

<http://www.infoworld.com/article/2609482/data-center/data-center-review-puppet-vs-chef-vs-ansible-vs-salt.html>

## References