# Cumulus Automation Modules

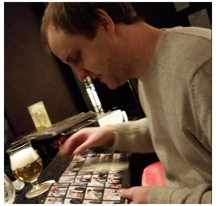
Leslie Carr

# Customer Engineering

- Prototyping, integration and tools development
- Distributed team

Former

**Barry Peddycord III**
Raleigh Durham, US



**Kristian Van Der Vliet**
Bristol, UK

**Leslie Carr**
San Francisco, US

**Nat Morris**
West Wales, UK

**Stanley Karunditu**
Raleigh Durham, US

# Foundations - Best practices

- **Treat like your servers**
  - No need to use custom tools and expect scripts
- **Partner with your server system administrators**
  - Use what they are using and collaborate !
- **Unified Monitoring**
  - Use server tools and existing monitoring systems

# Automation tools overview - Tools

- **Ansible**
  - Agent-less (via SSH from mgmt host)
  - No CL package required.

- **Puppet**
  - Puppet agent & Facter in addons repo
  - 3.6.2 (puppet as of CL2.5)
  - Enterprise version also available (amd64 only)

- **Chef**
  - amd64 & powerpc packages in addons repo
  - 11.6.2 as of CL2.5

# Unified automation modules

- **Unified automation modules**
  - Consistency across all toolsets - identical behaviour
  - Built for production use, not demo-ware
  - Easy to install via:
    - **Ansible Galaxy**
    - **Chef Supermarket**
    - **Puppet Forge**

# Automation tools overview - Modules

- **Modules**
  - cl-license
  - cl-ports
  - cl-interfaces
  - cl-interface-policy

# Automation tools overview - Modules - cl-license

- **Background:**
  - To enable switchd a license needs to be installed
  - Normally an engineer would have to run “cl-license”

- **Behaviour:**
  - Checks if license is installed and  up to date.
  - Checks expiration date. If license has not expired, then do nothing.
  - Confirms the source parameter is of the correct format.
  - Installs license if either license has expired or does not exist.

# Automation tools overview - Modules - cl-license

```
cumulus_license:
   src: 'http://mgmtserver.corpnet.com/cumulus.lic'
   force: true
notify: reload switchd
```

```
cumulus_license { 'license':
    src => 'http://mgmtserver.corpnet.com/cumulus.lic',
    force => true,
    notify => service[switchd]
}
```

```
cumulus_license 'corpnet' do
  source 'http://mgmtserver.corpnet.com/cumulus.lic'
  force true
  notifies :reload, "service[networking]"
end
```

# Automation tools overview - Modules - cl-ports

- **Background:**
  - Switch port attributes are configurable
    - 4 x 10G ports combined (or ganged) into one 40G
    - 40G ports can be split (or unganged) into 4 x 10G
  - Defined in /etc/cumulus/ports.conf

- **Behaviour:**
  - Write out /etc/cumulus/ports.conf replacing everything with the declared config
  - Supports port ranges

**ports.conf from a T+ 48x10G+4x40G**

```
1=40G/4
2=40G/4
3=40G/4
4=40G/4
5=10G
6=10G
7=10G
```

4 x 10G combined to 40G

…

```
45=10G
46=10G
47=10G
48=10G
49=40G
50=4x10G
```

40G broken out to 4 x 10G

# Automation tools overview - Modules - cl-ports

ANSIBLE

```
cumulus_ports:
      speed_40g_div_4: ["swp1-4"]
      speed_10g: ["swp5-48"]
      speed_4_by_10g: ["swp49-50"]
      speed_40g: ["swp51-52"]
notify: restart switchd
```

puppet labs

```
cumulus_ports { 'speeds':
      speed_40g_div_4 => ["swp1-4"],
      speed_10g => ["swp5-48"],
      speed_4_by_10g => ["swp49-50"],
      speed_40g => ["swp51-52"]
    notify => Service['switchd']
}
```

CHEF

```
cumulus_ports 'speeds' do
   speed_40g_div_4 ["swp1-4"],
   speed_10g ["swp5-48"],
   speed_4_by_10g ["swp49-50"],
   speed_40g ["swp51-52"]
  notifies :restart, "service[switchd]"
end
```

# Automation tools overview - Modules - cl-interfaces

- **Background:**
  - Assumes interface config is stored in separate files under a directory.
  - Doesn’t support config merging. overwrites existing config.

- **Behaviour:**
  - Module acts only on a single interface.
  - Checks JSON formatted existing config and compares with generated JSON desired config
  - Overwrites existing config using ifquery call, that converts desired JSON hash into regular text and copies to single interface text file.

# Automation tools overview - Modules - cl-interfaces

```
cumulus_bridge: name=br0 ports=”swp11-12 swp32-33 bond0”
                alias_name=”vlan aware bridge” mtu=9000 vlan_aware=true
                vids=”1-300” mstpctl_treeprio=4096

cumulus_bond: name=bond0 slaves=”swp3-4” ipv4=”10.1.1.1/24”

cumulus_interface: name=br0.1 ipv4=”10.1.1.1/24”

cumulus_interface: name=swp33 speed=1000 alias_name=”only has vlans1-10”
                   vids: “1-10”

cumulus_interface: name=lo addr_method=’loopback’

cumulus_interface: name=eth0 addr_method=’dhcp’
```

# Automation tools overview - Modules - cl-interfaces

```
cumulus_bridge { 'br0':
    ports => ['swp11-12, 'swp32-33, ‘bond0’],
    alias_name => 'vlan aware bridge',
    mtu => '9000',
    vids => [‘1-300’],
    vlan_aware => true,
    mstpctl_treeprio => '4096'
}
cumulus_bond { 'bond0':
    slaves => ['swp3-4'],
    clag_id => 1
}
cumulus_interface { ‘br0.1:
   ipv4 => ‘10.1.1.1/24’
}
cumulus_interface { 'swp33':
   speed => '1000'
   alias_name => ‘only has vlans 1-10’
   vids => [‘1-10’]
}
cumulus_interface{ 'lo':
   addr_method => 'loopback'
}
cumulus_interface{ 'eth0':
   addr_method => 'dhcp'
}
```

# Automation tools overview - Modules - cl-interfaces

```
cumulus_bridge 'br10' do
  ports ['swp11-12', 'swp32-33', ‘bond0’]
  alias_name 'vlan aware bridge’
  vlan_aware true
  vids “1-300”
  mtu 9000
  mstpctl_treeprio 4096
end

cumulus_bond 'bond0' do
   slaves ['swp3-4']
end

cumulus_interface 'swp33' do
  ipv4 '10.30.1.1/24'
  speed 1000
  alias_name ‘only has vlans 1-10’
end

cumulus_interface 'lo' do
   addr_method 'loopback'
end

cumulus_interface 'eth0' do
   addr_method 'dhcp'
end
```

# Automation tools overview - Modules - cl-interface-policy

- **Background:**
  - Enforcing which interfaces are configured on a switch
  - Stops rogue interfaces being setup ad-hoc

- **Behaviour:**
  - Assumes interface config distributed in individual files specific directory
  - Checks that interface files listed exists in enforcement list
  - Example: ‘*lo, eth0, swp1-10*” is the enforcement policy list
  - If interface directory has “*swp11*” file,  it is deleted.

# Automation tools overview - Modules - cl-interface-policy

```
cl_interface_policy: allowed="lo eth0 swp1-33 br0 br0.1-10 bond0-10"
notify: reload networking
```

```
cumulus_interface_policy { 'policy':
   allowed => ['lo', 'eth0', 'swp1-33', ‘br0’, ‘br0.1-10’, ‘bond0-10’],
   notify => service['networking']
}
```

```
cumulus_interface_policy 'policy' do
  allowed ['lo', 'eth0', 'swp1-33', ‘br0’, ‘br0.1-10’, ‘bond0-10’ ]
  notifies :reload, "service[networking]"
end
```

# Foundations - Best practices

- **Treat like your servers**
  - No need to use custom tools and expect scripts
- **Partner with your server system administrators**
  - Use what they are using and collaborate !
- **Unified Monitoring**
  - Use server tools and existing monitoring systems

# Foundations - Single source of truth

## Configuration management tool should be the source of truth

## Scenario - switch needs replacing

- Remove old device
- Update MAC address in DHCPd on management server
- Clear any old certificates (Puppet/Chef)
- Install new device
- ONIE installs CL
- ZTP bootstraps automation tool
- Switch back in service

Only update required

```
host sw-rack-88-tor-2 {
    hardware ethernet 00:11:22:33:44:55;
    option host-name "sw-rack-88-tor-2";
}
```

# Agenda

- Foundations
- Automation tool overview
- Unified automation modules
- Ansible
- Puppet
- Chef
- Infrastructure testing
- Continuous integration
- Summary

# Automation tools overview - Tools

- **Ansible**
  - Agent-less (via SSH from mgmt host)
  - No CL package required.
- **Puppet**
  - Puppet agent & Facter in addons repo
  - 3.6.2 (puppet as of CL2.5)
  - Enterprise version also available
- **Chef**
  - amd64 & powerpc packages in addons repo
  - 11.6.2 as of CL2.5

# Automation tools overview - Tools

- **Salt**
  - Agent-less deployment working
  - PowerPC packages for agent not yet available (working with Salt team)
    - Dependency on ZeroMQ for PowerPC

- **CFEngine**
  - Enterprise version working on Cumulus Linux
  - Community PowerPC packages not available

# Agenda

- Foundations
- Automation tool overview
- Unified automation modules
- Ansible
- Puppet
- Chef
- Infrastructure testing
- Continuous integration
- Summary

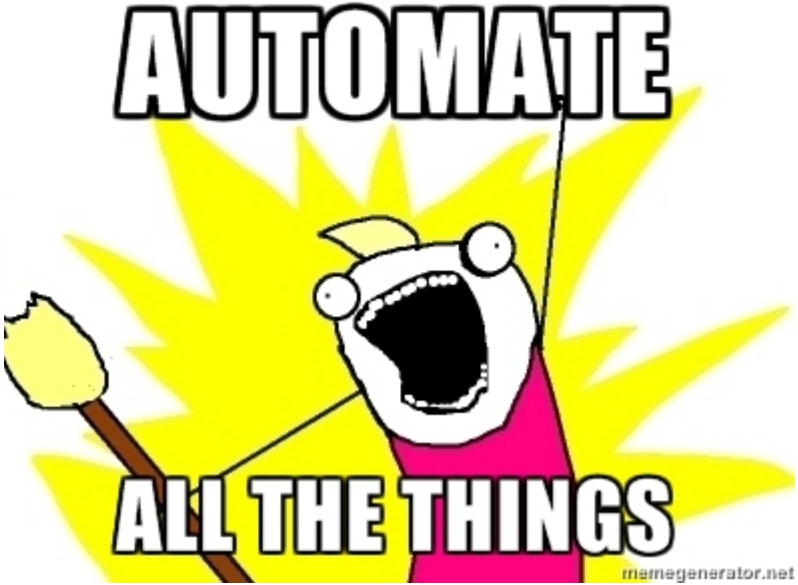
# Unified automation modules

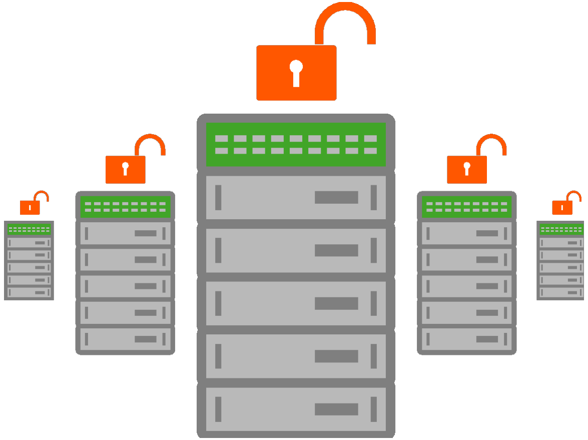
- **Unified automation modules**
  - Consistency across all toolsets - identical behaviour
  - Built for production use, not demo-ware
  - Easy to install via:
    - **Ansible Galaxy**
    - **Chef Supermarket**
    - **Puppet Forge**

# Automation tools overview - Modules

- **Modules**
  - cl-license
  - cl-ports
  - cl-interfaces
  - cl-interface-policy

# Summary





# Thank You!