Automating Juniper Devices
The Later Years

Rudolph Bott bott@sipgate.de @rbo_ne

sipgate

Who is sipgate?

sipgate

- VoIP Provider since 2006
- MVNO since 2012
- ~220 colleagues
- Düsseldorf, Germany

sipgate

Why Are
We Doing This?

sipgate

Our Environment

- ~830 servers
- ~300 hardware boxes
- Debian Linux all the way
- Multiple sites
- AS15594
- Public and private BGP peerings
- Ansible all the things!

sipgate

Our Network Environment

MX204

QFX5100

EX4300

JUNIPER NETWORKS

BROCADE

ProCurve Networking by HP

sipgate

Y tho?

- VoIP needs a stable network
- Some Juniper devices are slow
- Manual testing is tedious
- Automation should *save* time

sipgate

Network Automation Toolbox

- Git Repository
- Ansible (junos_config module)
- AWX
- Jinja2 templates
- python-yamale (YAML schema validation)
- pytest + ruby-junoser (template testing)




sipgate

How We Play
the Automation Game

sipgate

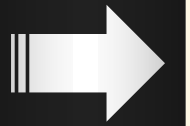
How We Play the Automation Game

- jinja2 template != Junos configuration
- Share code between your devices
- Build Ansible roles for device usecases, not for device types

sipgate

How We Play the Automation Game

```yaml
---
interfaces:
- type: "xe"
  fpc: 0
  id: 0
  mode: "layer3"
  ip4: 192.168.0.1/24
- type: "ge"
  fpc: 0
  id: 1
  mode: "trunk"
  tagged_vlans:
    - 100
    - 200
    - 300
```

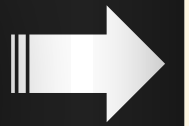


```
interfaces {
    {%- for int in interfaces %}
        replace:
        {{ int.type|default("xe") }}-0/{{ int.fpc|default("0") }}/{{ int.id }} {
        unit 0 {
        {% if int.mode == "layer3" %}
            {% if int.ip4 is defined %}
            family inet {
                address {{ int.ip4 }};
            }
            {% endif %}
        {% elif int.mode == "trunk" %}
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ {{ int.tagged_vlans|join(" ") }} ];
                }
            }
        [...]
```

sipgate

How We Play the Automation Game

```yaml
---
transit_name: "Best Transit Corp. [ID-123]"
transit_transfer_ip4: "192.168.0.1/30"
transit_transfer_ip6: "fc00::1/64"
```



```
interfaces {
    replace:
    xe-0/1/1 {
        description {{ transit_name }};
        mtu 1538;
        unit 0 {
            family inet {
                address {{ transit_transfer_ip4 }};
            }
            family inet6 {
                address {{ transit_transfer_ip6 }};
            }
        }
    }
}
```

sipgate

Ask Your Network
How It’s Doing

sipgate

Ask Your Network How It’s Doing

- Juniper devices support command results in plaintext, XML or JSON
- junos_command converts JSON to python data structures
- detect system alerts, interface or MC-LAG/VRRP/OSPF errors etc.

sipgate

Ask Your Network How It’s Doing

```yaml
- name: Fetch system alarms
  junos_command:
    commands: show system alarms
    display: json
  register: system_alarms

- name: Check system alarms
  assert:
    that:
      - system_alarms.stdout[0]["alarm-information"][0]["alarm-summary"][0]["no-active-alarms"] is defined
    success_msg: No system alarms detected
    fail_msg: "System alarms count is not 0! Please check the ouput of 'show system alarms'"
    quiet: yes
```

sipgate

Ask Your Network How It’s Doing

- Gain more confidence in everyday-deployments
- Validate a network operating system update
- Run entire testsuite against lab

sipgate

Test Driven
Template Development

Test Driven Template Development

- Test Driven...what?
  - Write your unit test first and let it fail
  - Adapt your software until the test passes
  - Change your test to reflect a new requirement and let it fail
  - Adapt your software until the test passes
  - Repeat

sipgate

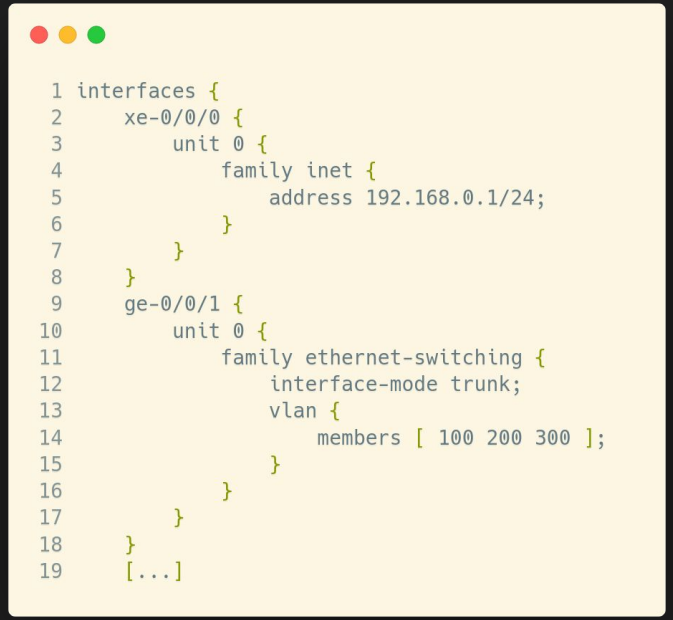
Test Driven Template Development

```
interfaces {
    {%- for int in interfaces %}
        replace:
        {{ int.type|default("xe") }}-0/{{ int.fpc|default("0") }}/{{ int.id }} {
        unit 0 {
        {% if int.mode == "layer3" %}
            {% if int.ip4 is defined %}
            family inet {
                address {{ int.ip4 }};
            }
            {% endif %}
        {% elif int.mode == "trunk" %}
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ {{ int.tagged_vlans|join(" ") }} ];
                }
            }
        [...]
```

Changing larger Junos/Jinja2 templates is not easy

sipgate

Test Driven Template Development



```
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 100 200 300 ];
                }
            }
        }
    }
    [...]
```

Step One: build your configuration (e.g. on a lab device)

sipgate

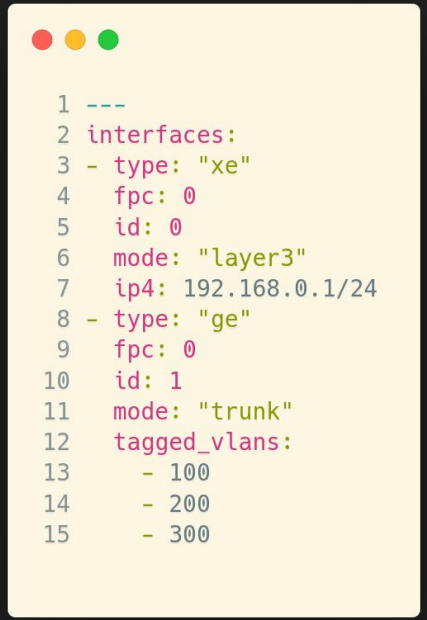
Test Driven Template Development

```
interfaces {
{%- for int in interfaces %}
    replace:
    {{ int.type|default("xe") }}-0/{{ int.fpc|default("0") }}/{{ int.id }} {
    unit 0 {
    {% if int.mode == "layer3" %}
        {% if int.ip4 is defined %}
        family inet {
            address {{ int.ip4 }};
        }
        {% endif %}
    {% elif int.mode == "trunk" %}
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members [ {{ int.tagged_vlans|join(" ") }} ];
            }
        }
    [...]
```

Step Two: build your template

sipgate

Test Driven Template Development



```yaml
---
interfaces:
- type: "xe"
  fpc: 0
  id: 0
  mode: "layer3"
  ip4: 192.168.0.1/24
- type: "ge"
  fpc: 0
  id: 1
  mode: "trunk"
  tagged_vlans:
    - 100
    - 200
    - 300
```

Step Three: generate sample data

sipgate

Test Driven Template Development

Step Four: render template and compare

sipgate

Test Driven Template Development

```
interfaces {
    xe-0/0/0 {
        mtu 9000;
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 100 200 300 ];
                }
            }
        }
    }
    [...]
```

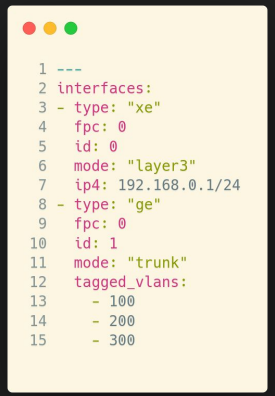
Rinse and repeat: Build/change your desired config, break the test, fix the template (& sample data)

sipgate

Test Driven Template Development

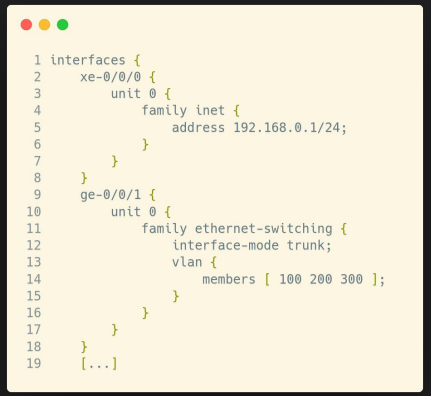
```
interfaces {
{%- for int in interfaces %}
    replace:
    {{ int.type|default("xe") }}-0/{{ int.fpc|default("0") }}/{{ int.id }} {
    unit 0 {
    {% if int.mode == "layer3" %}
        {% if int.ip4 is defined %}
        family inet {
            address {{ int.ip4 }};
        }
        {% endif %}
    {% elif int.mode == "trunk" %}
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members [ {{ int.tagged_vlans|join(" ") }} ];
            }
        }
    [...]
```

Jinja2 Template



```
---
interfaces:
- type: "xe"
  fpc: 0
  id: 0
  mode: "layer3"
  ip4: 192.168.0.1/24
- type: "ge"
  fpc: 0
  id: 1
  mode: "trunk"
  tagged_vlans:
    - 100
    - 200
    - 300
```

Test Data



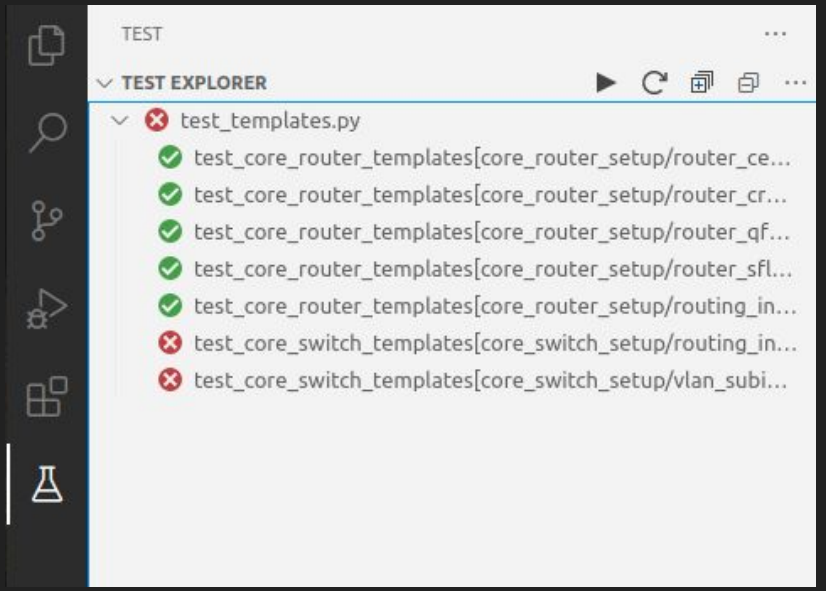
```
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 100 200 300 ];
                }
            }
        }
    }
    [...]
```

Result

sipgate

Test Driven Template Development

- pytest
  - Python-native testing framework
  - Magic glue for all components
  - Easy integration of YAML/Jinja2
  - Visual support in IDEs



sipgate

Test Driven Template Development

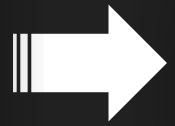
```
1 interfaces {
2     xe-0/0/0 {        unit 0 {
3             family inet {         address 192.168
4             }   }
5     }
6     ge-0/0/1 {        unit 0 {
7             family ethernet-switching {      inte
8                 vlan {         members [ 100 200
9             }  }
10        }
11    }
12    [...]
```


```
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 100 200 300 ];
                }
            }
        }
    }
    [...]
```

sipgate

Test Driven Template Development

```
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 100 200 300 ];
                }
            }
        }
    }
    [...]
```



```
set interfaces xe-0/0/0 unit 0 family inet address 192.168.0.1/24
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 100
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 200
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 300
```

sipgate

Test Driven Template Development

- What is *junoser*?
  - Parses any given Junos configuration
  - Syntax validation
  - **translates between config and set syntax**
  - http://xml.juniper.net/junos/18.3R3/junos *or* ask your device

sipgate

Test Driven Template Development

- The abstract of this talk said something about Docker?
  - Disclaimer: we are not a ruby shop
  - Hide magic in docker container
  - Integrate it with python-docker into pytest-based testsuite

```
1 def convert_junos_config_to_set(docker_client, config_path):
2     set_syntax_result = docker_client.containers.run('junos-config-validator',
3                                                      command="/junoser/exe/junoser -d /generated.txt",
4                                                      volumes={config_path: {'bind': '/generated.txt', 'mode': 'ro'}})
```

sipgate

Test Driven Template Development

- For each template, store sample data (YAML) and the expected result
- Read the sample data
- Render the template
- Convert to *set* syntax and compare
- Fail or move on to the next template

sipgate

Validate Your YAML Files

Validate Your YAML Files

- Someone introduces a new variable
- Someone forgets to remove a variable
- What the heck are the possible values for *this* variable?!
- Ansible templates choke on missing variables

sipgate

Validate Your YAML Files

- Basic types: integer, float, string, boolean, null, lists
- optional/required variables
- defined sets of values
- content like IPv4, IPv6 addresses or dates
- custom strings through regexes

sipgate

Validate Your YAML Files

- How to use python-yamale?
  - integrated in pytest
  - integrated in Ansible

```yaml
- name: Test schema of host_vars
  sipgate.yamale.yamale_validate:
    schema_path: "schema/hostvars_core-router.yml"
    data_path: "host_vars/{{ inventory_hostname }}"
```

github.com/sipgate/ansible-module-yamale

sipgate

Validate Your YAML Files

```yaml
---
interfaces:
- type: "xe"
  fpc: 0
  id: 0
  mode: "layer3"
  ip4: 192.168.0.1/24
- type: "ge"
  fpc: 0
  id: 1
  mode: "trunk"
  tagged_vlans:
    - 100
    - 200
    - 300
```

```
---
interfaces: list(include("interface_l3"),include("interface_trunk"))

# define include types in a separate YAML document:
---
interface_l3:
  type: enum("ge","xe")
  fpc: int()
  id: int()
  mode: str("layer3")
  ip4: ip(version=4)

interface_trunk:
  type: enum("ge","xe")
  fpc: int()
  id: int()
  mode: str("trunk")
  tagged_vlans: list(int())
```

sipgate

Wrap Up

Wrap Up

- Structure your Ansible playbooks / roles
- Avoid code duplication with abstract Ansible roles
- Avoid complicated templates
- Split into different Ansible roles before things get messy/complicated
- Use Ansible to get instant feedback from your network
- Use tests to validate templates before deploying them
- Use YAML schema validation to avoid extra/missing variables or illegal values

Rudolph Bott bott@sipgate.de @rbo_ne

sipgate

Get Your Hands Dirty

github.com/sipgate/ansible-juniper-cookbook

Rudolph Bott bott@sipgate.de @rbo_ne

sipgate