



Network Provisioning and Integration Testing

(USING ANSIBLE AND JENKINS)

Our solution

Ansible:

Generates the configs using the *template module*

Gerrit:

Git server for code review and version control

Jenkins:

Runs the ansible playbooks(and a lot more)

ZTP/POAP:

Configs are served up to switches automatically over HTTP/TFTP

* Images are also served to switches

Why automate provisioning?

- ▶ **Humans are slow**

Computers can speed up config generation

- ▶ **Humans make mistakes**

Computers can eliminate configuration errors

- ▶ **Humans are fickle**

Everybody has there own standard (which changes on a daily basis)

Result:

- ▶ Simpler network
- ▶ Easier to troubleshoot
- ▶ Humans can focus on what they are good at

Ansible

Automation tool used for:

- ▶ Provisioning
- ▶ Config management
- ▶ Software deployment

Low learning curve:

- ▶ YAML
- ▶ Basic linux

Flexible:

- ▶ Python, loops, conditionals, etc

Our Use Case:

- ▶ Config generation

Example playbook

```
---
- name: generate role for new tr switch and update siteconfig file
  connection: local
  hosts: localhost

  vars_prompt:
  - name: "switch"
    prompt: "switch name"
  vars:
  - switch_u: "{{ switch.upper() }}"
  - site: "{{ switch_u.split('-')[0] }}"
  vars_files:
  - ../../vars/{{ site }}.yaml
    ../../vars/bgp.yaml

  tasks:
  - name: create BGP config
    template: src=bgp.j2 dest=bgp.conf

  - name: create tasks, templates and vars directories
    file: path=../../roles/{{ switch_u }}/{{ item }} state=directory
    with_items:
      - tasks
      - templates
      - vars
```

Template Module (jinja2)

Variable file (bgp.yaml)

```
---
bgp_as: 65084

bgp_peers:
- ip: 1.1.1.1
  as: 65084
  description: "*** BGP Peer 1 ***"
- ip: 2.2.2.2
  as: 65084
  description: "*** BGP Peer 2 ***"
```

BGP Config file

```
!
router bgp 65084
  neighbor 1.1.1.1 remote-as 65084
  neighbor 1.1.1.1 description *** BGP Peer 1 ***
  neighbor 2.2.2.2 remote-as 65084
  neighbor 2.2.2.2 description *** BGP Peer 2 ***
!
```

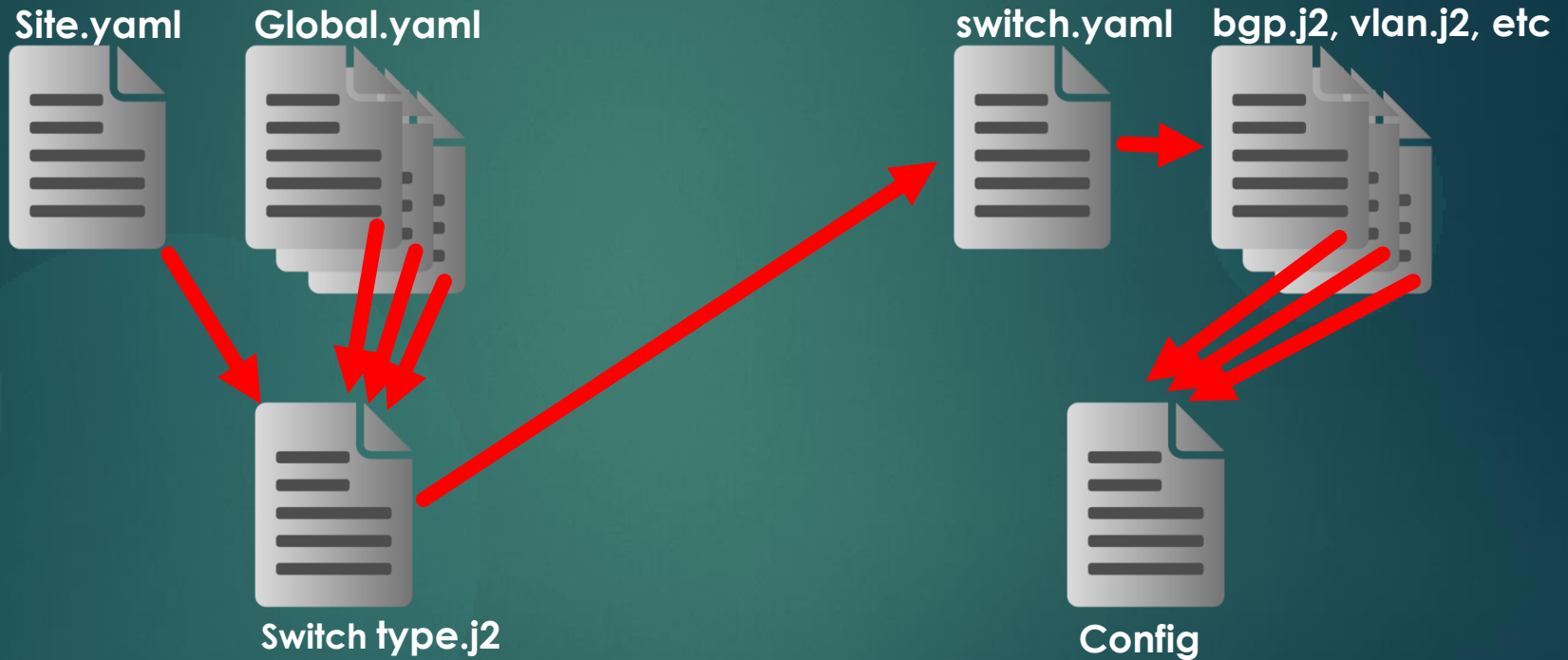
Template file (bgp.j2)

```
!
router bgp {{ bgp_as }}
{% for peer in bgp_peers %}
  neighbor {{ peer.ip }} remote-as {{ peer.as }}
  {% if peer.description is defined %}
  neighbor {{ peer.ip }} description {{ peer.description }}
  {% endif %}
{% endfor %}
!
```



Our Implementation

(Matryoshka Dolls)



Next Steps

Programmatic Tools to configure network:

- ▶ Python scripts, web GUI's or ansible playbooks can now easily be written to reconfigure network devices via 'switch.yaml' variable files
- ▶ Yaml files are human and machine readable

Testing and Validation:

- ▶ Ansible playbooks can use the same 'switch.yaml' variables as input to validate the network is in expected state
- ▶ Variables can change but playbooks stay the same

Jenkins & the Network



What is Jenkins?

- ▶ Cron on Steroids.
- ▶ Automation server.
- ▶ Continuous integration server.
- ▶ Its free

Installation & Configuration

- Quick Installation
- Easy Configuration
- ▶ Hundreds of Plugins

Git , SVN , perforce , Jira , Ansible , Gerrit etc

Jenkins and Integration testing

Integration testing verifies the different pieces of configuration management system works well together.

Problem: Someone's checks in a bad commit that breaks your config generation modules that you need but you discover this many days later ☹️

Jenkins Can help

Create a Jenkins project in a simple 3 step process :

- ▶ Defines SCM (Git , SVN etc.)
- ▶ Specify Integration test as a build step
 - ▶ Ansible playbook.
 - ▶ Python or your favourite scripting language.
- ▶ Specify a Post Build action
 - ▶ Run another Jenkins project
 - ▶ Creating or Update Jira
 - ▶ Email notification

Demo



Other use-cases

- ▶ We use Jenkins for trending , visibility & centralisation benefits rather than deploying cron landmines across our environment.
- ▶ Automatically update DNS server once a new network device is checked into our Git Repo
- ▶ Jenkins Enables Zero Touch Replacement strategy in our environment.

What else are we working on ?

- ▶ Using Jenkins to build Continuous Deployment Pipeline for network infrastructure.
- ▶ Continuous Compliance orchestration