Networkshop_49

Automating the Jisc Managed Router Service and Janet Access Programme

Managed Router Service

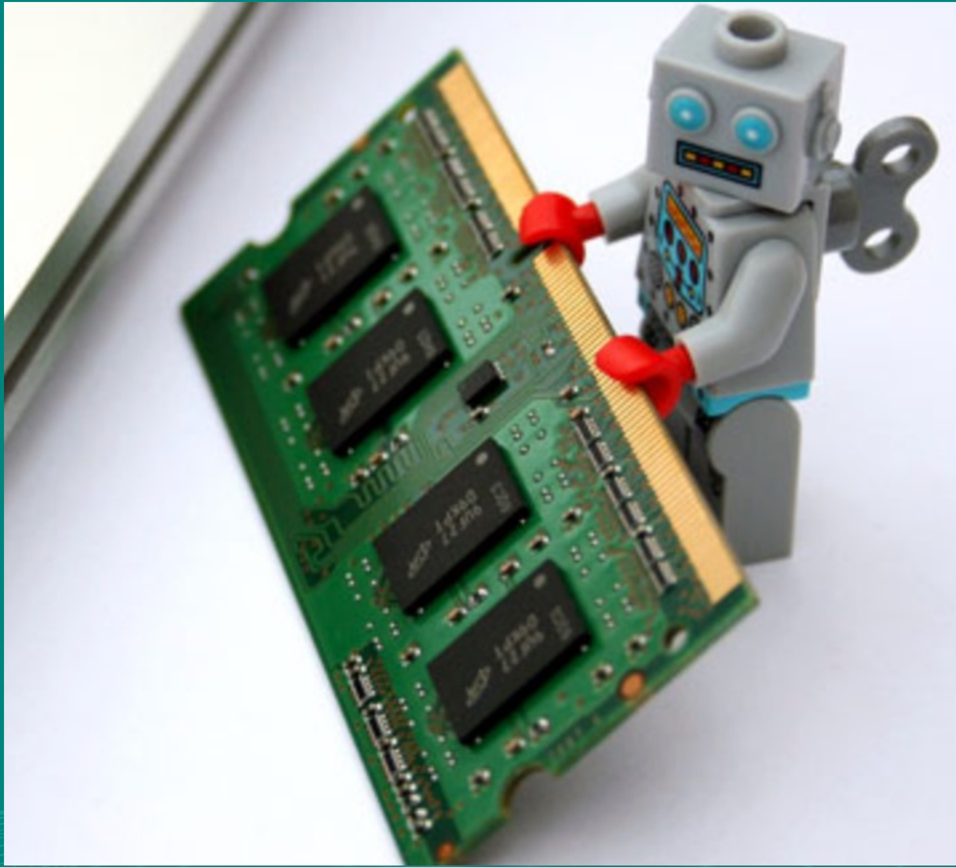
Choosing the best platform for the task

- Why Automate?
- Why Ansible?
  - Open Source
  - Agentless
  - Well documented
  - Vendor agnostic
  - Extensible
  - Flexible

IOS & Junos upgrades

Required to keep software in-line with vendor recommendations

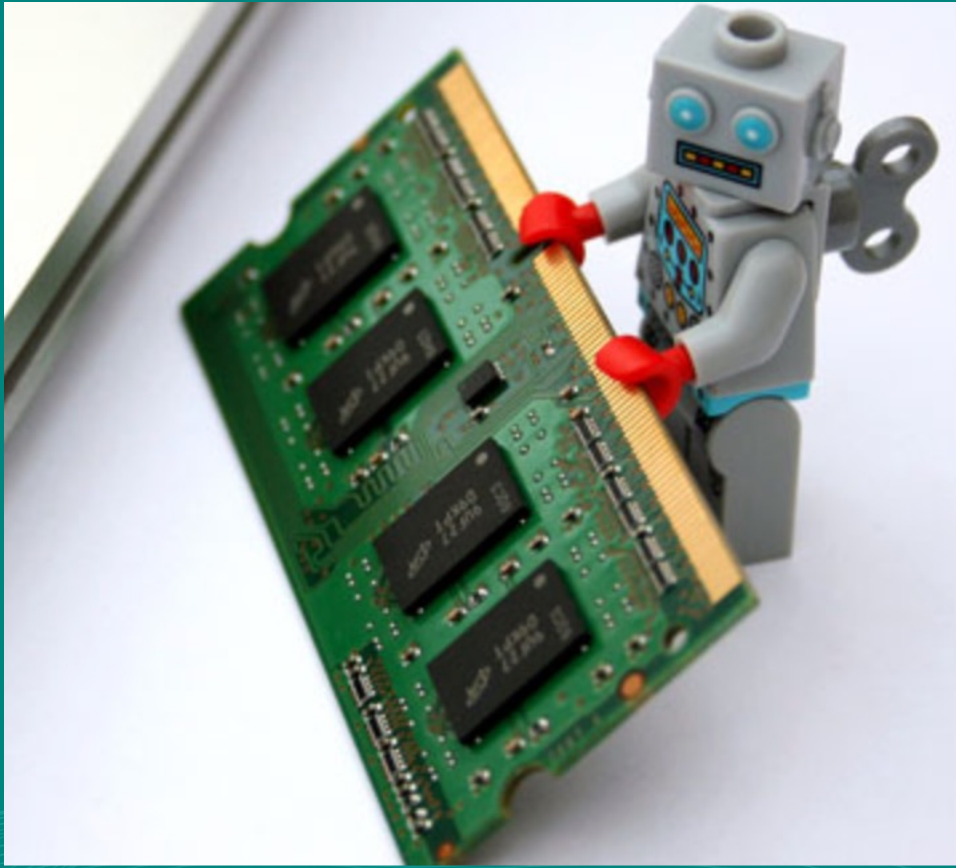
- Previously:
  - Repetitive manual process
  - Each engineer responsible for upgrading a set of devices
  - Upgrades took place over a period of months



IOS & Junos upgrades

With Ansible

- Automated checks and logging
- Upgrades caried out in batches
- Approx. 140 hours saved on this deployment
- Ansible plays reusable



Hardening Device Security

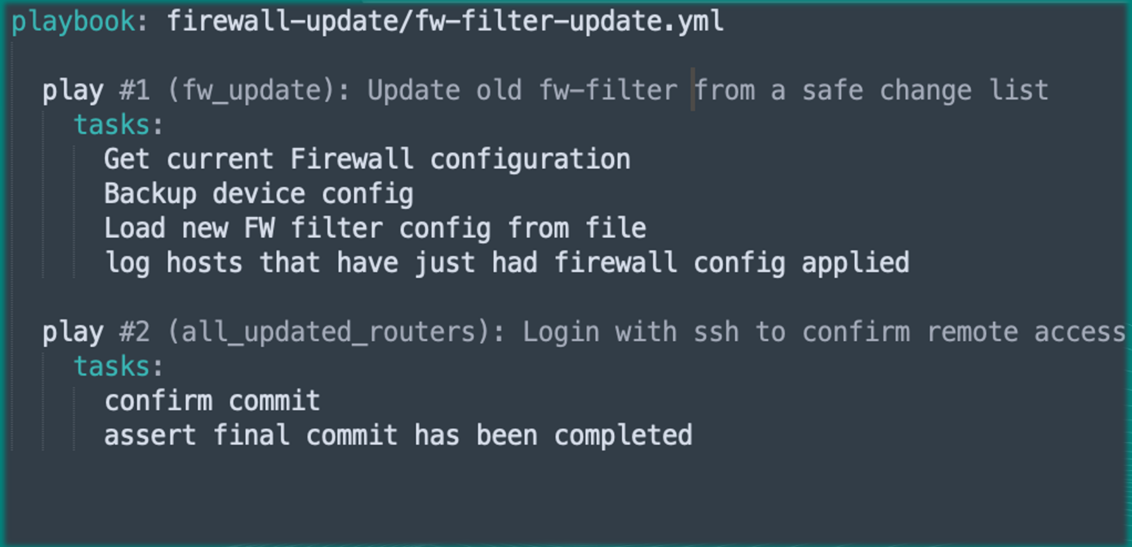
Opportunity to standardize

- ACLs protecting the MRS device management
- Change in policy led to need to tighten further
- Lots of text to validate before making changes
- Potential for loss of mgmt. access if mistakes made!

Hardening Device Security

Goal: apply standard filter on all devices

- Repeatable process
- Machines are good at pattern matching
- Ansible only applies changes when required
- Large majority of routers automatically verified and upgraded



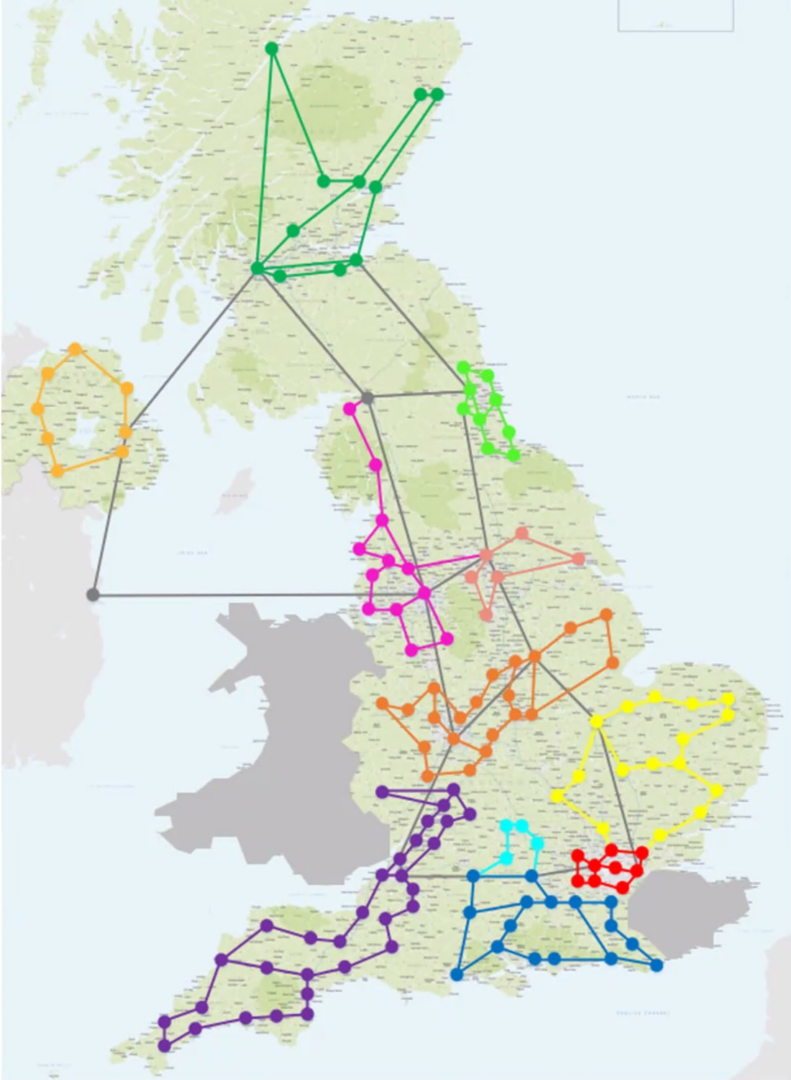
```
playbook: firewall-update/fw-filter-update.yml

  play #1 (fw_update): Update old fw-filter from a safe change list
    tasks:
      Get current Firewall configuration
      Backup device config
      Load new FW filter config from file
      log hosts that have just had firewall config applied

  play #2 (all_updated_routers): Login with ssh to confirm remote access
    tasks:
      confirm commit
      assert final commit has been completed
```

Janet Access Programme

Configuring new networks

Region by region

- Writing variables to Ansible inventory
- Variable validation within plays
- Using device based templates
- Reusing data in multiple templates
  - Configuration
  - Documentation
  - DNS



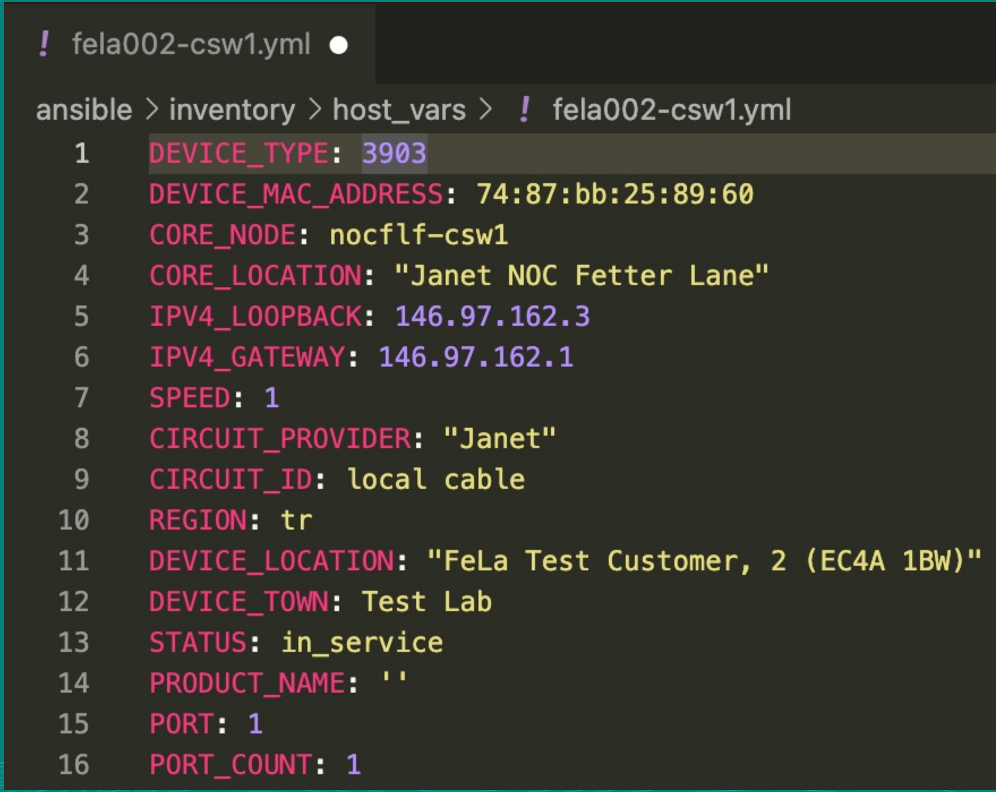
Piecing things together

- Ansible
  - Inventory – host data
  - Plays – task lists
- Jinja2
  - Configuration templates
- Python3 scripts
  - Adding host data to Ansible
- git
  - Version control

Where can we improve?

- Source of truth?
  - Network devices
  - Ansible inventory
  - Other?

- Data Management
  - Importing to ansible
  - Validity checks
  - Preventing duplicates



```yaml
DEVICE_TYPE: 3903
DEVICE_MAC_ADDRESS: 74:87:bb:25:89:60
CORE_NODE: nocflf-csw1
CORE_LOCATION: "Janet NOC Fetter Lane"
IPV4_LOOPBACK: 146.97.162.3
IPV4_GATEWAY: 146.97.162.1
SPEED: 1
CIRCUIT_PROVIDER: "Janet"
CIRCUIT_ID: local cable
REGION: tr
DEVICE_LOCATION: "FeLa Test Customer, 2 (EC4A 1BW)"
DEVICE_TOWN: Test Lab
STATUS: in_service
PRODUCT_NAME: ''
PORT: 1
PORT_COUNT: 1
```

Essentials

- Revision control
- An IDE
- Lab / testbed devices
- Good dialogue
- Coffee!

Future

- MRS standard config
  - Validation Checks
  - template / ZTP configuration of new routers

- Single source of truth
  - Netbox?

- Automated workflows

Thank you

David Richardson

Senior Network Engineer

david.richardson@jisc.ac.uk

15 Fetter Lane, London, EC4A 1BW

0203 006 6017

customerservices@jisc.ac.uk

jisc.ac.uk