Network Configuration Automation

Dale W. Carder
ESnet Network Engineering
Lawrence Berkeley National Laboratory
Acknowledgements

Indira Kassymkhanova

Scott Richmond

- Richard Cziva
- Jon Dugan
- Nehemya McCarter-Ribakoff
- John MacAuley
- Michael O’Connor
- Sam Oehlert
- Brendan White
Example ESnet5 automation capabilities

- L2 circuits & bandwidth calendaring
- IGP traffic engineering
- eBGP Peering & prefix automation
- iBGP full mesh config
- Topology & inventory
- RANCID++, "netlint" static checker
- User account & ssh key sync from LDAP
- Device boilerplate & policy sync
Existing tools

- Written in multiple languages
- Ships in the night
- Reimplement business rules
- Individually talk to hardware
- Do their own verification & diff checks
- Really hard to delegate / federate functions
- Very "box" centric
- The network is the Source of Truth!
Approach

• Punt on Provisioning, use tools in existence
  – configuring devices is not really the hard part
  – existence proof:
    • state of the art 20yr old Expect scripts

• Move Source of Truth up to yaml, jinja2 files
  – Move to database sometime later
Findings

Templating systems are way too naive

- A form 1:1 with configuration elements has only minor value. *We want to describe intent.*
- Ansible configuration (yaml) is likely turing-complete

Use Ansible/Salt systems as a glorified Makefile

- Do "real" logic in Python: software libraries, more reusable components
- Write out .yaml files for import into Ansible
  - worry about integration later

- Focus on the hard part: defining all of our business rules
Use / Exploit 'git'

- git repo defines the network state
  - create a branch for change to network
  - update files, run tooling
  - merge the branch
  - update network to match

- As such, some features come for "free"
  - engineering design review is really a merge request
  - the state of the network is known snapshot of the repo
    - can also use git tagging!
  - built-in diff functions
  - tracking of state over time
    - (this can be exploited, for example to audit prefix announcements)
Putting it together: Automation Design Pattern
Network Configuration Automation

Dale W. Carder
dwcarder@es.net
ESnet Network Engineering
Lawrence Berkeley National Laboratory
Example

- Snapshot database state
- Merge w/ auxiliary data stored in yaml input files
- `esgen` & `ipsvc` script wrappers
  - scripts pull data in from external sources
    - can easily be individually run by hand for troubleshooting
    - simple command-line flags
    - json or flat text stdout
    - warnings to stderr
- saves configuration state to various intermediate and output files
  - json file per-peer fully defines the service for a peer
  - yaml file per-peer, peer- network element facilitates templates
- calls esgen to ram the result through j2 templates
- load configurations into devices
- Diff generated config w/ running config
- Any of these steps are composable for use in ansible