

Salty DNS: Automate all the things

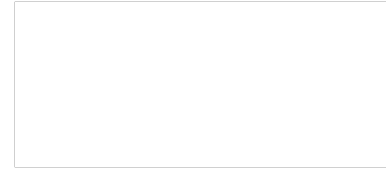
Joe Crowe – joseph_crowe@comcast.com

Daniel Boughton – daniel_boughton@comcast.com

NANOG 69

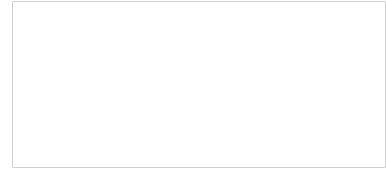
High level background

What we do



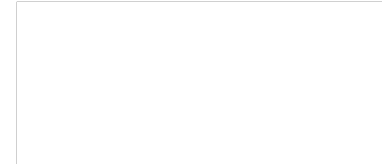
- Main focus is on large complex DNS Infrastructure serving over 20 Million customers
- Took lead on moving from one automation platform to SaltStack

Why a new automation tool?



- No access to many features
 - No Marionette
 - No r10k – featured based releases are important
- Oversubscribed and growth issues
- We were using ansible with a hosts script to run jobs

Other factors ..



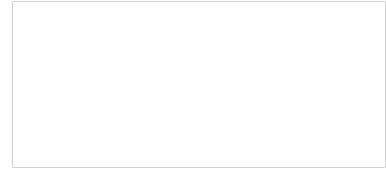
- Difficulty extending puppet in support of newer integrated infrastructure
- Difficulty with puppet dependency loops, especially with built-in dependencies.
- Who doesn't want to control their own destiny?

How did we get there?

What's the plan?

- Replace aging puppet tool with something we could control.
- Replace centralized kickstart server with something we could control
- Extend the capabilities of a new change management tool in support of a new infrastructure

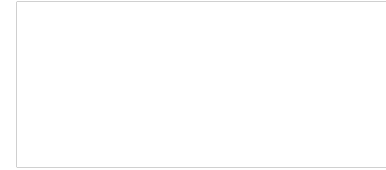
The plan - continued



- We
- Want
- Control
- This helps ensure we know when/why something is broken

We got Salty

Turned to SaltStack

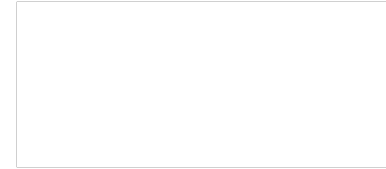


- Included things we liked from Ansible and Puppet
- State files similar to puppet
- Data separation was built-in
- Looked very extensible
- Individual states or groups of states can be applied easily

Conversion seemed easy...

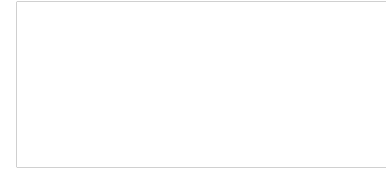
- Initial conversion and tests took about 1 week
- Puppetfiles to state files
- Facts to grains
- Hieradata to Pillar
 - Both yml, win/win
- ERB to Jinja or Mako
 - Complexity dependent

Converting isn't really that quick.



- Actual conversion took months
- Added module to extend network device capabilities.
 - Looked at NAPALM
 - Created our own module
- Created module for various network “math”
 - Role, host, and pillar based
- Added module to integrate with multiple DNS vendor libraries

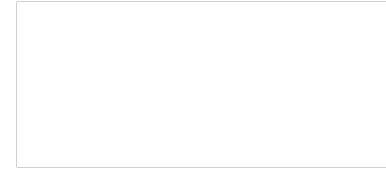
Couldn't kick the pillar habit...



- Refined our pillar code to be able to pull data from one source.
- Rolled out new infrastructure using SaltStack
- Moved to own kickstart servers local to market
 - Better performance
 - More Control (Did we mention we want control?)

**Salt doesn't dissolve right
away**

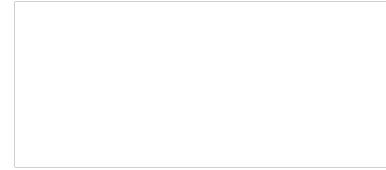
Bumps in the road



- Python
- Unit testing with pillar without using virtual machines
- Jinja and Mako without python experience
- Design changes
- Continuous Improvement

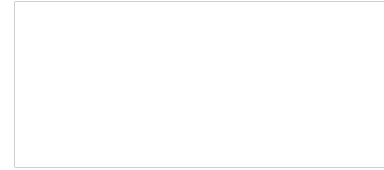
Results

We have control!



- More integration with our infrastructure
 - Network devices
 - Servers
 - Vendor software
- Easier integration with anything
- Good unit tests for hand-built modules

We have control – continued.



- One source for given method
 - Module based
 - Helps eliminate duplicate code
- Much better targeted information gathering.
- Faster bare metal to production light up times.



Thank you!