SALTCONF 15

SaltStack Configuration Management Best Practices

Seth House, SaltStack Software Engineer
Michael Ryabushkin, cars.com Engineer

#SaltConf15

Writing Formulas

Style

- Use a descriptive State ID
- Use module.function notation
- Specify the name parameter
- Comment state files

Easy on the Jinja!

- Know the evaluation and execution order
- Avoid changing the underlying system with Jinja
- Avoid heavy logic and programming

When to use Jinja

- Inspect the local system
- Gather external data
- Light conditionals and looping
- Avoid inter-mixing states with Jinja
- Jinja Macros

Abstracting static defaults into a lookup table

- Collecting common values
- Overriding values in the lookup table
- When to use lookup tables
  - Platform-specific information
  - Sane defaults
  - Environment specific information

Single-purpose SLS files

- Do one thing and do it well
- Mix-and-match:
  - Top files
  - Orchestrate
  - Reactor
- ...or use sequential ordering with `listen`

Parameterization

- As much variable data as possible

Configuration

- Platform defaults
- Upstream defaults
- Sane defaults
- Allow Pillar overrides

Pillar Overrides

- Automatic merges
  - `pillar_source_merging_strategy`
- Manual merges
  - `update` method
  - `pillar.get`
  - `filter_by`

Scripting

- Python renderer
- Execution modules

Organization

- Document!
- Shallow directories
- Group related files

Testing Formulas

SALTCONF 15

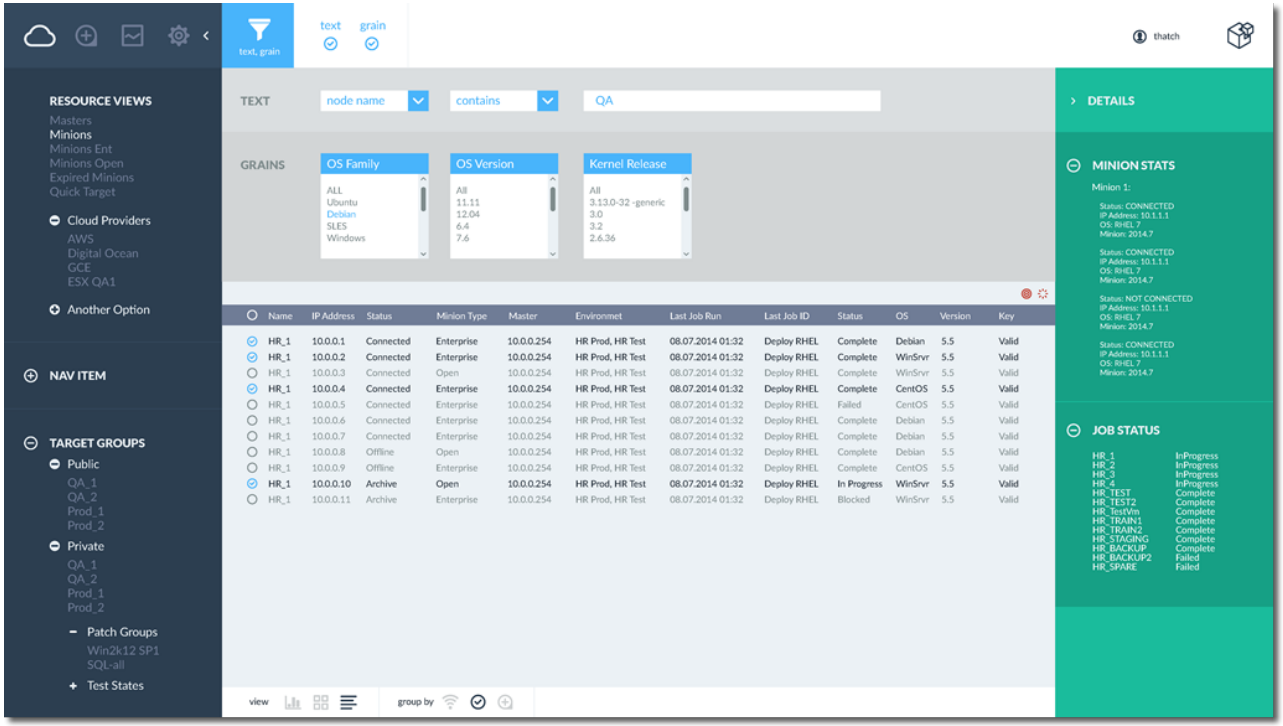
Development Steps

- One piece at a time
- What are you testing right now?
- Combine into a whole

All About the Data Structure

- Syntax errors
  - Jinja errors
  - YAML errors
  - Salt errors

SaltStack Enterprise GUI



Thank you

Please provide session feedback in the SaltConf15 mobile event guide

#SaltConf15