Using Salt to Make Infrastructure Consumable (Tasty, Even)

Warren Turkal, Infrastructure h4X0r
Introduction

- What SignalFx Does
- How We Organize Our Infrastructure
- Tooling
- Why We Chose Saltstack
- Integrating Salt with Our Infra
- Practical Choices
- Monitoring
- Q&A
What SignalFx Does

Monitoring Modern Infrastructure (as a Service)

- Consume metrics at any resolution (down to 1s)
- At most any scale (many many 10s of billions of datapoints a day)
- That are multi-dimensional (service, region, device type, canary, ...)
- For: visualization, ad hoc exploration, statistical analysis, alerting

- Special sauce: in-stream, “real-time” analytics
  - To build composite signals out of raw metrics → meaningful alerts
  - On current data as it’s arriving and historical data
  - Example: alert on (defined) week-over-week deviation in service-level metric like 90th percentile successful response latency over moving 1-hour window
How Our Infrastructure is Organized

Region (aws us-east-1)
How Our Infrastructure is Organized

Region (aws us-east-1)

Culture (aa)

Culture (bb)
How Our Infrastructure is Organized

Region (aws us-east-1)

Culture (aa)
Cell (aaaa)
Cell (aaab)

Culture (bb)
Cell (bbaa)
Cell (bbab)
How Our Infrastructure is Organized

Region (aws us-east-1)

Culture (aa)

Cell (aaaa)

Host

Cell (aaab)

Host

Culture (bb)

Cell (bbaa)

Host

Cell (bbab)

Host
How Our Infrastructure is Organized

Design goals:

• Cells fail independently
• TCP/IP connectivity between every host
• Ability to choose placement of hosts that take advantage of cost/latency/reliability concerns
How Our Infrastructure is Organized

*I gave a talk at SRECon15 with many more details: [link](#)*
Tooling - sfinfra

A library and a set of tools:

- sf-culture-tool
- sf-cell-tool
- sflb
- sfhost
Starting from Zero Production - Why We Chose Salt

- *Python*
- *Extensible*
Starting from Zero - Why We Chose Salt

- Extensible
- Python
- Ponies
Integration With Our Infra

Custom grains and pillars
Integration With Our Infra

Custom states

- HAProxy
- SSH
Practical Concerns

- one active salt-master in each culture
- ability to bootstrap a master from a different culture
- disaster recovery
- minimal use of public formulas
Monitoring

- Collectd plugin
- Important Metrics
  - Current highstates longer than 5m
  - Num nodes down (somewhat inaccurate and occasionally wildly fluctuates)
Future Work for Me

• master-tops
• move to saltstack packages
• investigate the ec2 external pillar module
• investigate multi-master again
Features I Would Like to See

- stackable pillars
- something like master tops for pillar data files
- use a stack of git branches to make up an environment’s fileserver contents
- port AWS related functionality to boto3 (especially s3fs)
Conclusion

- Our tooling ensures hosts are brought up in a normalized way
- Salt has some extension possibilities
- Salt has taken us from 0 to 600ish hosts in each culture with lots of automation
- Salt is working pretty well for us.
THANK YOU!
Follow me on Twitter @w00tSpeaks

Come see us at the booth for funtastic new features.

SIGN UP FOR A TRIAL AT:

signalfx.com