SaltStack – Brownbag

(Not) just another
Automation & Remote Execution Tool

Arnold Bechtoldt
Karlsruhe, 16.08.13
Agenda

1. Project
2. Quickstart
3. States
4. Grains
5. Pillar
6. Modules
- Was born in February, 2011
- Written in Python, Apache License v2
- Runs on Linux, Arch Linux, FreeBSD, OS X, Solaris, Windows
- > 10k commits, ~ 340 contributors, Top 10 of „GitHubs notable OSS“ (2012)
- Commercial support by the company behind SaltStack
1. Setting the repository source
   - deb http://debian.saltstack.com/debian wheezy-saltstack main

2. Importing the package signing key
   - wget -q -O - "http://debian.saltstack.com/debian-salt-team-joehealy.gpg.key" | apt-key add -

3. Updating the local cache & Installation of Salt Client + Master
   - apt-get update && apt-get install salt-minion salt-master
4. Make domain 'salt' resolvable (Client)
   - echo "192.168.2.1 master.domain.de salt" >> /etc/hosts

5. Restart the client (Client)
   - service salt-minion restart

6. Accept client's public key (Master)
   - salt-key -a client.domain.de
Salt States
Postfix Satellite Example

- Salt State File Tree (/srv/salt/):
  - top.sls
  - postfix/
    - init.sls
    - satellite.sls
  - files/
    - etc/
      - postfix/
        - satellite.main.cf
Salt States

top.sls – Assigning nodes to SLS modules

development:

’mx-*’:

- postfix.satellite

development:

’webservers’:

- match: nodegroup
- apache
- curl

Environment

Match nodes (hostname)

Module

Global defined group of nodes
postfix/init.sls – Initial Module States

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Function</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>postfix</td>
<td>pkg</td>
<td>- installed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- names:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- postfix</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- postfix-pcre</td>
<td></td>
</tr>
<tr>
<td>service:</td>
<td></td>
<td>- running</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- require:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- pkg: postfix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- file: /etc/postfix/main.cf</td>
</tr>
</tbody>
</table>
Salt States
postfix/satellite.sls – Special Module States

/etc/postfix/main.cf:

  file.managed:
    - source: salt://postfix/files/etc/postfix/satellite.main.cf
    - user: root
    - group: postfix
    - mode: 640
    - require:
      - pkg: postfix
... myhostname = {{ grains['fqdn'] }}
myorigin = $myhostname
inet_interfaces = {{ inet_interfaces }}
{% if use_postscreen == True -%}
postscreen_bare_newline_action = ignore
postscreen_blacklist_action = drop
{% endif %}
{% endif %}
## Salt States

### Predefined SaltStack State Types (Puppet: Resources)

<table>
<thead>
<tr>
<th>alias</th>
<th>cmd</th>
<th>cron</th>
<th>svn</th>
<th>disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>gem</td>
<td>git</td>
<td>grains</td>
<td>user</td>
</tr>
<tr>
<td>group</td>
<td>mercurial</td>
<td>hosts</td>
<td>iptables</td>
<td>kmod</td>
</tr>
<tr>
<td>libvirt</td>
<td>locale</td>
<td>lvm</td>
<td>mdadm</td>
<td>mongodb</td>
</tr>
<tr>
<td>mount</td>
<td>mysql</td>
<td>network</td>
<td>pkg</td>
<td>postgres</td>
</tr>
<tr>
<td>quota</td>
<td>selinux</td>
<td>service</td>
<td>ssh</td>
<td>...</td>
</tr>
</tbody>
</table>
Salt Grains

Static node-specific information (Puppet: Facts)

- Contain node-specific information like
  - BIOS (release, version)
  - CPU (manufacturer, arch, model, flags)
  - Hostname, Domain, FQDN
  - LSB distribution / OS info (OS, codename, release, id, kernel type + version)
  - IP configuration (Interfaces, IP addresses)
  - Salt runtime environment data (version, $PATH, Python version, master)
  - System type (virtual/ physical)

- Can be extended

- But do not have to: → Pillar
Salt Pillar
Dynamic node-specific information

- Node-specific information defined by the user
- Uses the same structure as SLS tree (top.sls, several environments, …)
- Provides the same targeting possibilities (globbing, regex, node groups, lists, grains)
- Will be sent to the client by the master during SLS execution only
- Simple YAML syntax
Salt Modules
Remote Execution

- To be used on the CLI (Master):
  
  ```
  salt \\
  -L 'web1.domain.de,web2.domain.de,web3.domain.de' \\
  pkg.install \\
  name=foobar \\
  refresh=true \\
  fromrepo=wheezy-backports \\
  version=4.2
  ```

- Or within SLS files: {% if salt['pkg.upgrade_available']('foobar') == true %}
## Salt Modules

### Predefined SaltStack Modules

<table>
<thead>
<tr>
<th>pkg</th>
<th>alias</th>
<th>apache</th>
<th>timezone</th>
<th>at</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>cp</td>
<td>cron</td>
<td>debconf</td>
<td>dig</td>
</tr>
<tr>
<td>disk</td>
<td>dnsutil</td>
<td>pip</td>
<td>extfs</td>
<td>file</td>
</tr>
<tr>
<td>gem</td>
<td>git</td>
<td>group</td>
<td>logrotate</td>
<td>mdadm</td>
</tr>
<tr>
<td>mongodb</td>
<td>mount</td>
<td>mysql</td>
<td>network</td>
<td>service</td>
</tr>
<tr>
<td>quota</td>
<td>ps</td>
<td>puppet</td>
<td>S3</td>
<td>ssh</td>
</tr>
<tr>
<td>solr</td>
<td>tls</td>
<td>tomcat</td>
<td>user</td>
<td>...</td>
</tr>
</tbody>
</table>
Vielen Dank für Ihre Aufmerksamkeit

Kontakt

Arnold Bechtoldt
Systems Engineer

inovex GmbH
Office Karlsruhe
Zur Gießerei 16
76227 Karlsruhe

+49 (173) 3181 117
arnold.bechtoldt@inovex.de