Salt – A Scalable Systems Management Solution for Datacenters

FrOSCon 2016, St. Augustin

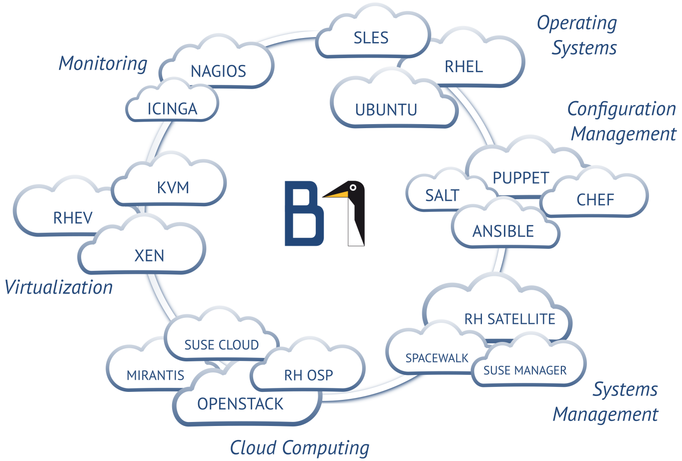
August 21, 2016

Sebastian Meyer
Linux Consultant & Trainer
B1 Systems GmbH
meyer@b1-systems.de

Introducing B1 Systems

- founded in 2004
- operating both nationally and internationally
- nearly 100 employees
- provider for IBM, SUSE, Oracle & HP
- vendor-independent (hardware and software)
- focus:
    - consulting
    - support
    - development
    - training
    - operations
    - solutions

Areas of Expertise



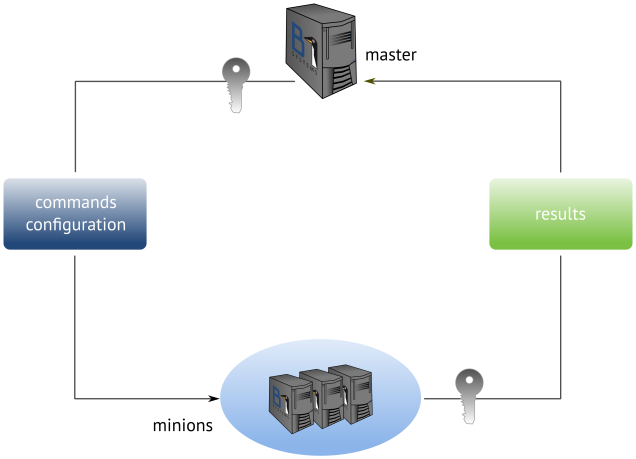
Salt – Introduction

Yet Another Systems Management Solution?

- takes inspiration from Puppet, Chef or Ansible
- focuses on the entire system life cycle
- easily scalable to a few thousand systems
- convenient and easy to learn
- configuration management and remote execution

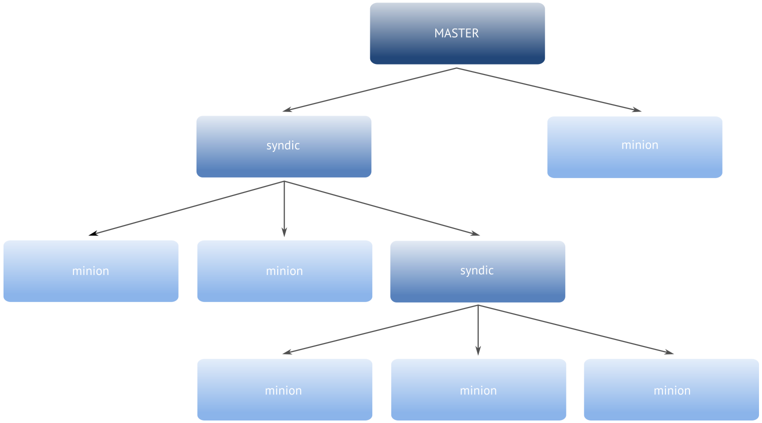
Salt – Concept

Master & Minions

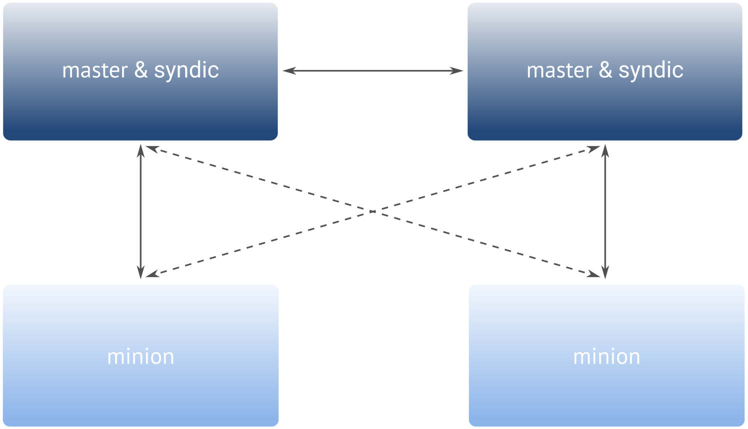


Source: https://docs.saltstack.com/en/getstarted/fundamentals/install.html

Scalability: Masters, Syndics & Minions



High Availability: Multiple Masters& Minions

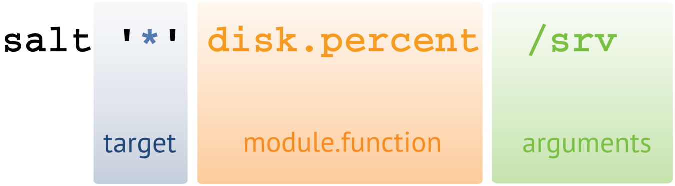


Salt Modes

- minions pull from master
- master pushes to Minions
- minions apply states locally
- master applies states on minions via SSH

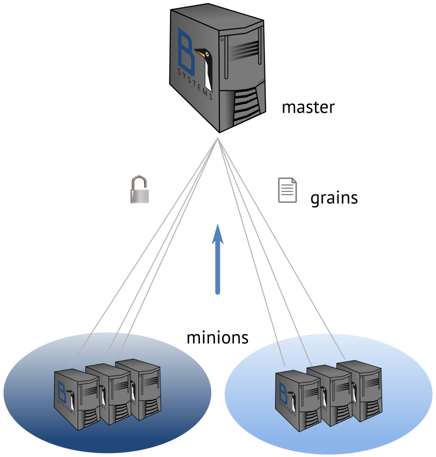
Remote Execution System

Salt Command



```
salt '*' disk.percent /srv
```

Grains



Source: https://docs.saltstack.com/en/getstarted/overview.html

Configuration Management

States

```
ID:
  module.function:
    - name: name
    - argument1: value
    - argument2:
      - value1
      - value2
```

Top File

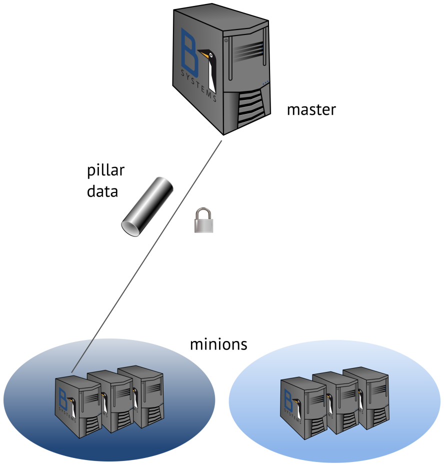
```
base:
  '*':
    - monitoring
    - ssh
    - syslog

  '*lan*':
    - ntp.lan

  '*dmz*':
    - ntp.dmz
    - firewall
```

- all servers:
  - monitoring
  - ssh config
  - syslog
- servers in LAN:
  - ntp config
- servers in DMZ:
  - ntp config
  - firewall

Pillars



Source: https://docs.saltstack.com/en/getstarted/overview.html

Pillar Data

Pillar Example

```
ntp:
  {% if grains['id'].startswith('myntpserver') %}
  ntpservers: ["0.us.pool.ntp.org","1.us.pool.ntp.org"]
  comment: ''
  {% else %}
  ntpservers: ["10.1.1.20","10.1.1.21"]
  comment: 'myinternalservers'
  {% endif %}
```

Source: https://github.com/saltstack-formulas/ntp-formula/blob/master/pillar.example

Pillars and States

States top.sls

```
base:
  '*':
    - monitoring
    - ssh
    - syslog
    - ntp

  '*dmz*':
    - firewall
```

Pillar top.sls

```
base:
  '*':
    - monitoring
    - ssh
    - syslog

  '*lan*':
    - ntp.lan

  '*dmz*':
    - ntp.dmz
    - firewall
```

Deploying the State

Master pushes to minions

```
salt '*' state.highstate
salt '*' state.sls mystate
```

Minions pull from master

```
salt-call state.highstate
salt-call state.sls mystate
```

Reusing States: Formulas

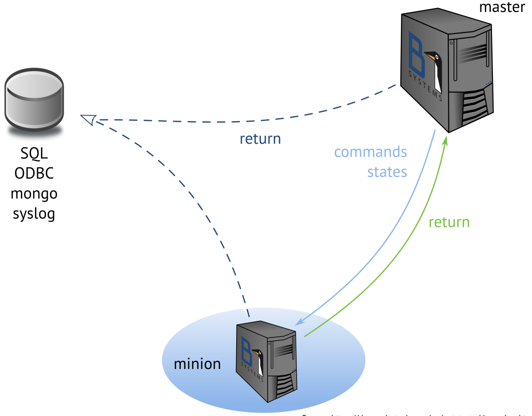
- reusing existing code
- roughly the same as Puppet modules/Ansible roles
- collection of States and files
- github.com/saltstack-formulas/ for "official" formulas

Using Formulas

- directly from VCS or local
- extendable via include
- configurable via Pillar data
- variables mapped via Jinja map
- requirements across Formulas possible

Demo

Returners



Source: https://docs.saltstack.com/en/getstarted/overview.html

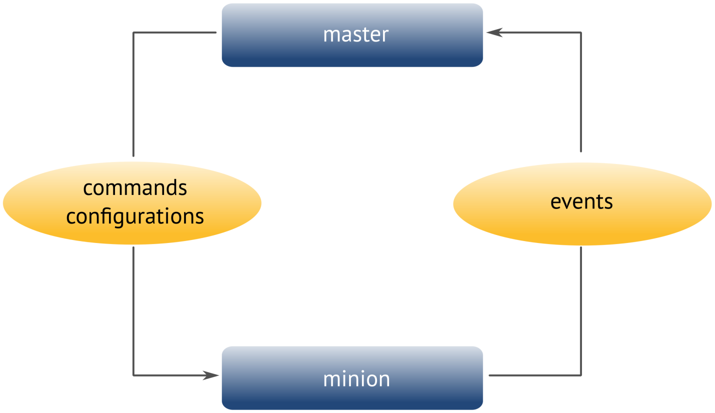
```
salt '*' disk.usage --return redis_return
```

Salts Event Driven Infrastructure

Overview

- actions trigger events
- events are communicated via the event bus
- reactors execute trigger actions responding to events

Event Bus



Actions & Events

```
master# salt ’salt-minion-01’ disk.percent /srv
salt-minion-01:
    11%
```

Actions & Events

```
20160422163250339970 {
    [...]
}
salt/job/20160422163250339970/new {
    "_stamp": "2016-04-22T14:32:50.340357",
    "arg": [ "/srv" ],
    "fun": "disk.percent",
    "jid": "20160422163250339970",
    "minions": [ "salt-minion-01" ],
    "tgt": "salt-minion-01",
    "tgt_type": "glob",
    "user": "root"
}
```

Actions & Events

```
salt/job/20160422163250339970/ret/salt-minion-01 {
    "_stamp": "2016-04-22T14:32:50.536877",
    "cmd": "_return",
    "fun": "disk.percent",
    "fun_args": [ "/srv" ],
    "id": "salt-minion-01",
    "jid": "20160422163250339970",
    "retcode": 0,
    "return": "11%",
    "success": true
}
```

Events in a State

```
b1/mystate/status/update:
  event.send:
    - data:
        status: "Installation done!"
```

Beacons

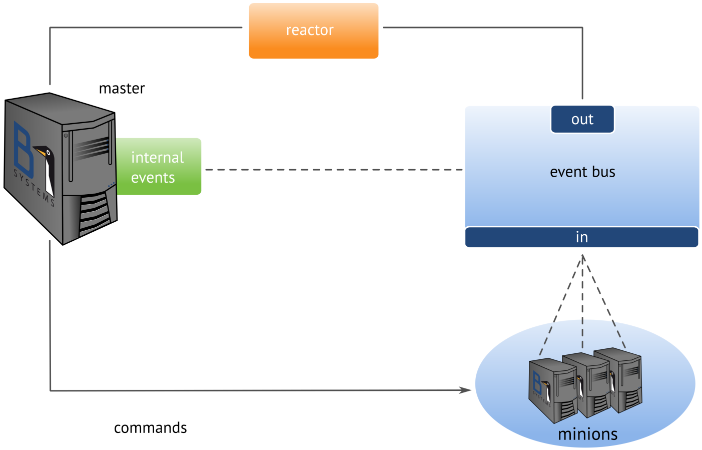
- hook into system on minion
- create events
- inotify, diskusage, load, journald ...

Beacons - Example

inotify Beacon

```
beacons:
  inotify:
    /etc/motd:
      mask:
        - modify
```

Reactors



Source: https://docs.saltstack.com/en/getstarted/overview.html

Calling Reactors on Events

Reactor Example

```
reactor:
  - 'salt/minion/*/start':
    - /srv/reactor/start.sls

  - 'b1/mystate/status/*':
    - salt://reactor/status.sls
```

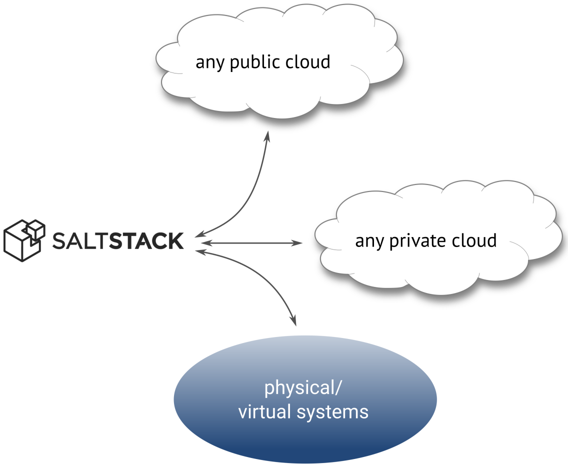
Demo

Use Cases?

- load-balancing
- job automation
- alerting

Salt Cloud

Overview



Source: https://docs.saltstack.com/en/latest/topics/cloud/

Providers

Amazon EC2 Provider Example

```
my-ec2:
  driver: ec2
  id: 'MYEC2ID'
  key: 'adsfrf453fMYKEYasdsadg43'
  private_key: /etc/salt/my_key.pem
  keyname: my_key
  securitygroup: default
  minion:
    master: saltmaster.example.com
```

Profiles

- profile name
- provider
- image or template
- options for the instance
- minion options

Profiles

LXC Profile Example

```
myfancyprofile:
  provider: lxc-host01
  lxc_profile:
    template: ubuntu
    options:
      release: trusty
    password: test123
```

Maps

Mapfile

```
profile1:
  - instance_name_1
  - instance_name_2
profile2:
  - instance_name_3:
      grains:
        mykey: myvalue
  - instance_name_4
```

Execute Mapfile

```
salt-cloud -m /path/to/mapfile
```

Bootstrapping a New Salt Environment

Mapfile

```
profile1:
  - instance_name_1:
      make_master: True
      minion:
        master: myoldmaster
        local_master: True
  - instance_name_2
  - instance_name_3
  - instance_name_4
...
```

Saltify Existing Machines 1/2

Saltify Provider

```
saltify-all-machines:
  driver: saltify
  minion:
    master: mysaltmaster
```

Saltify Profile

```
salt-machine:
  provider: saltify-all-machines
  ssh_username: root
  key_filename: '/etc/salt/pki/master/ssh/salt-ssh.rsa'
```

Saltify Existing Machines 2/2

Mapfile

```
salt-machine:
  - first-machine:
      ssh_host: 1.2.3.4
  - second-machine:
      ssh_host: 1.2.3.5
  - third-machine:
      ssh_host: 1.2.3.6
```

Thank You!

For more information, refer to info@b1-systems.de
or +49 (0)8457 - 931096