

Saltstack



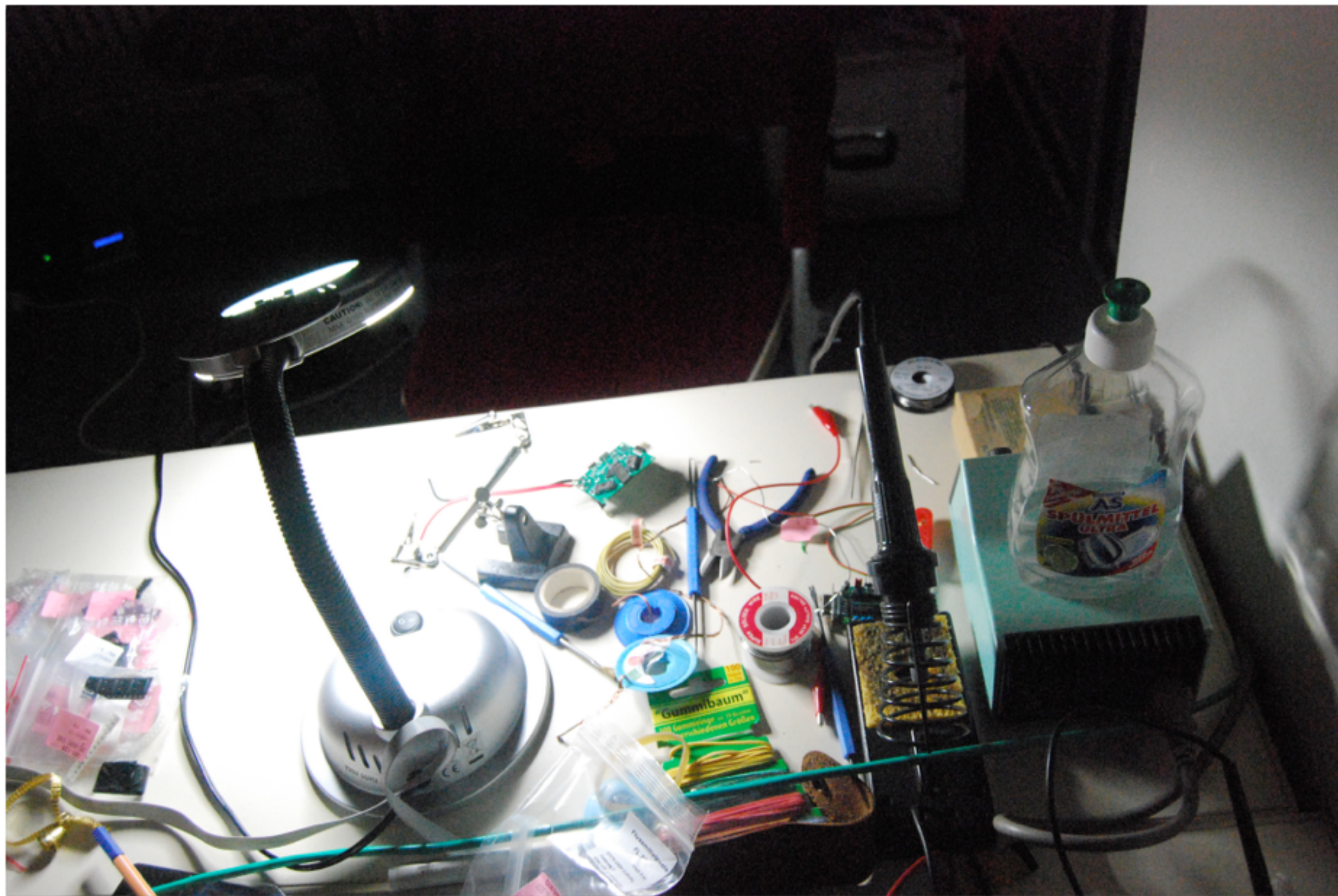
Thorsten Kramm | Berlin Juni 2015

SLAC Secure Linux
Administration
Conference 2015
24.-26. Juni 2015 | Berlin



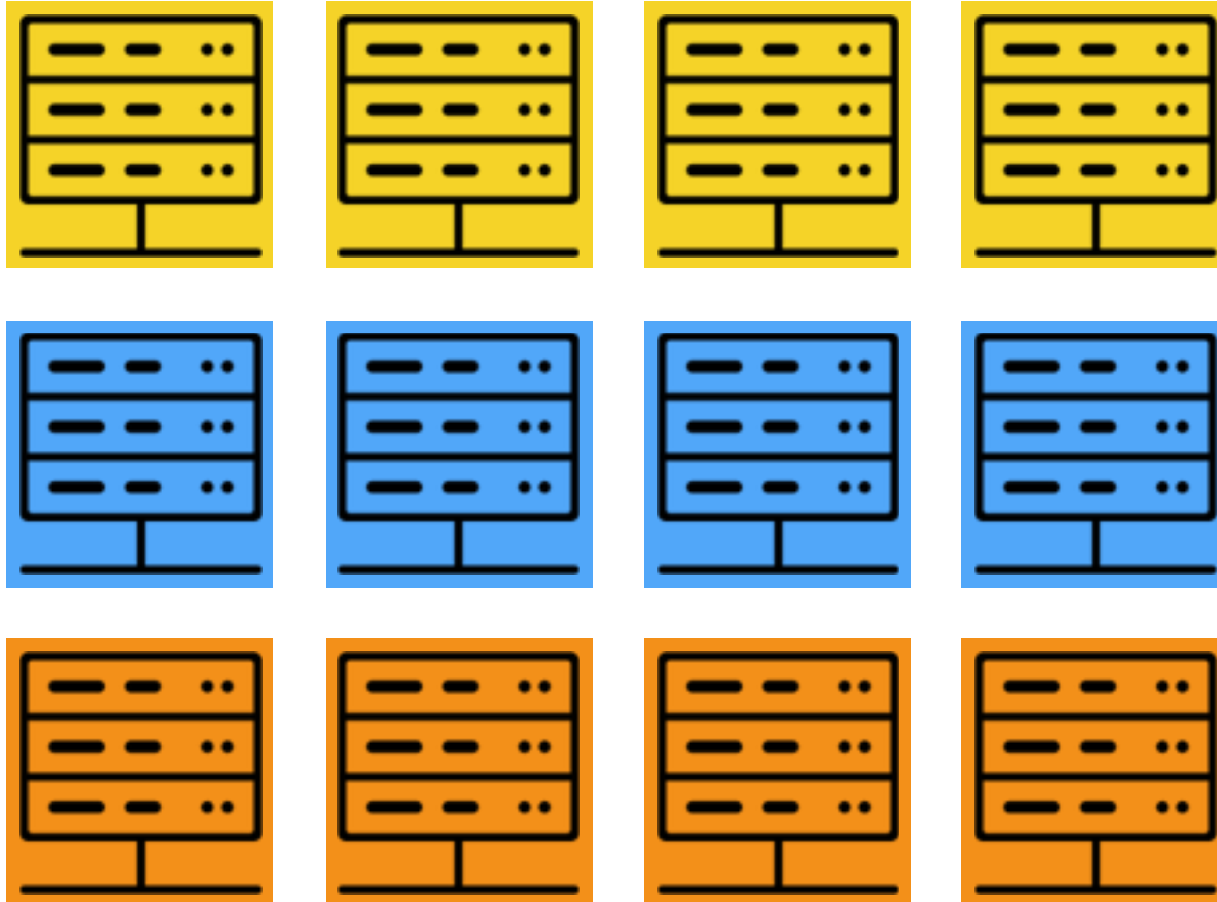








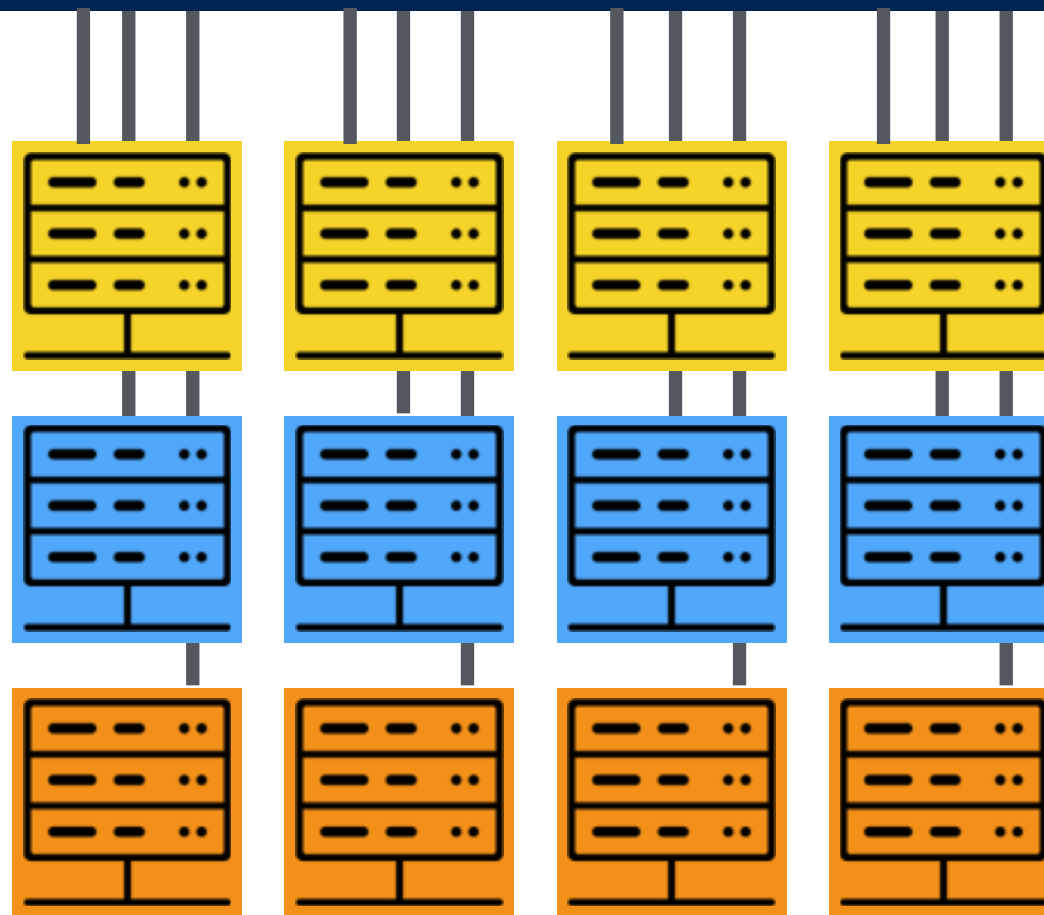
Computersysteme



Einheit
Ordnung
Automatisierung

Was macht Saltstack?

Eine `root@salt-master:# full-control` alle.



Was macht Saltstack?

```
root@salt-master:# full-controll
```

Einer beherrscht sie alle.

- ✓ Befehle parallel ausführen.
- ✓ Systeme und Software installieren.
- ✓ Systemstatus sicher stellen.
- ✓ Konfigurationen erzeugen und bearbeiten.



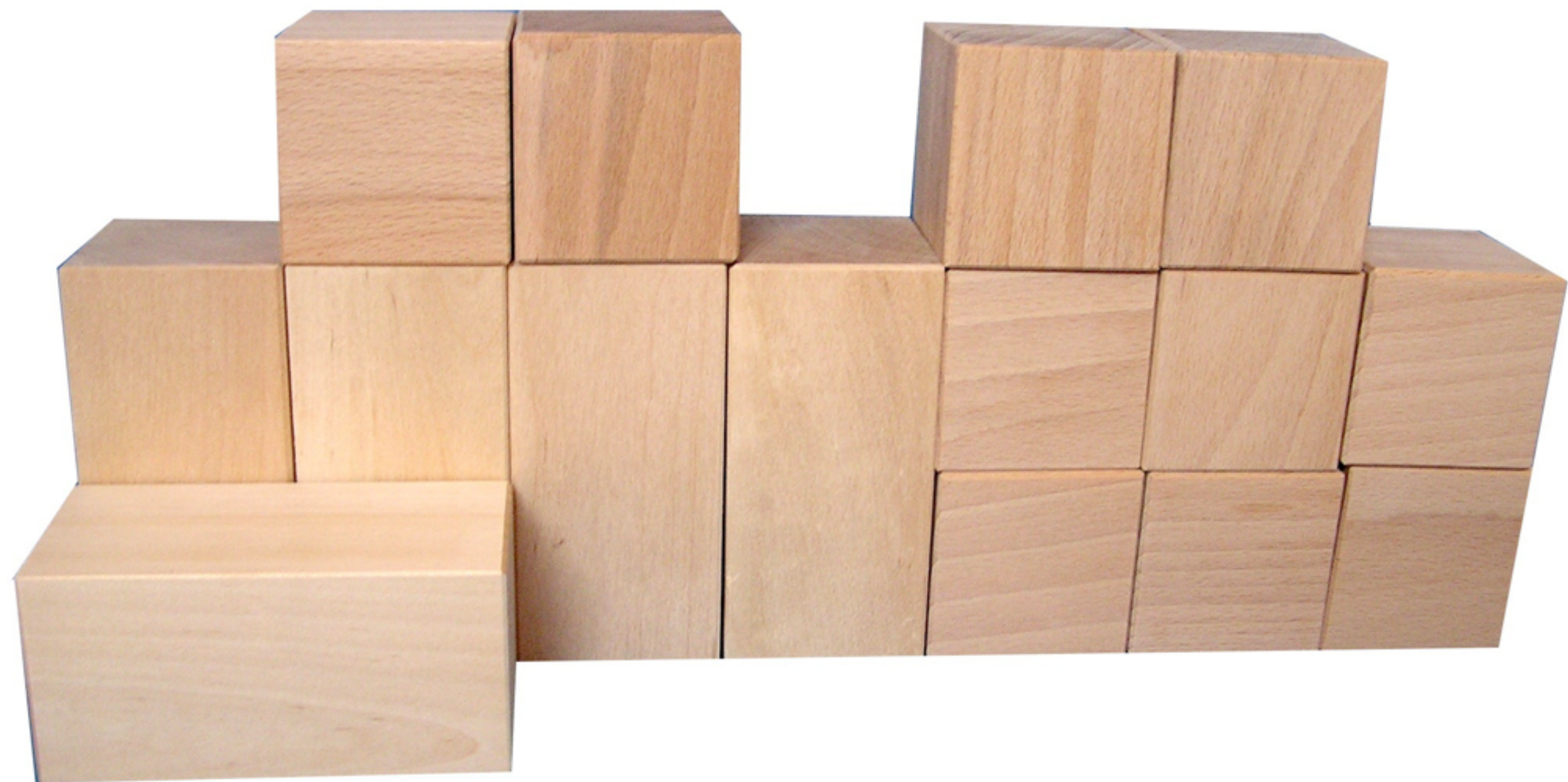
Warum Saltstack?

... es gibt doch schon

Puppet
cfEngine
Chef
ansible

...







Saltstack ist einfach

...

... aber nicht
beschränkt!

Salt-Master installieren

- ✓ Hinzufügen des Saltstack Software Repositories

```
# /etc/apt/sources.list.d/saltstack.list  
deb http://debian.saltstack.com/debian jessie-saltstack main
```

- ✓ Importieren des Schlüssels

```
wget -q -O- "http://debian.saltstack.com/debian-salt-team-joehealy.gpg.key" | apt-key add -
```

- ✓ Master installieren

```
apt-get update && apt-get install salt-master
```



Minions installieren

- ✓ Füge DNS- oder Hosts-Eintrag hinzu

```
root@minion2:~# cat /etc/hosts
127.0.0.1          localhost
127.0.1.1          minion2.lab4.org minion2
10.129.163.163     salt.lab4.org salt
```

- ✓ Minion installieren

```
apt-get update && apt-get install salt-minion
```

„Master, ich will der folgen“

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt-key -L
Accepted Keys:
Unaccepted Keys:
minion1.lab4.org
minion2.lab4.org
minion3.lab4.org
Rejected Keys:
root@salt-master:~#
```

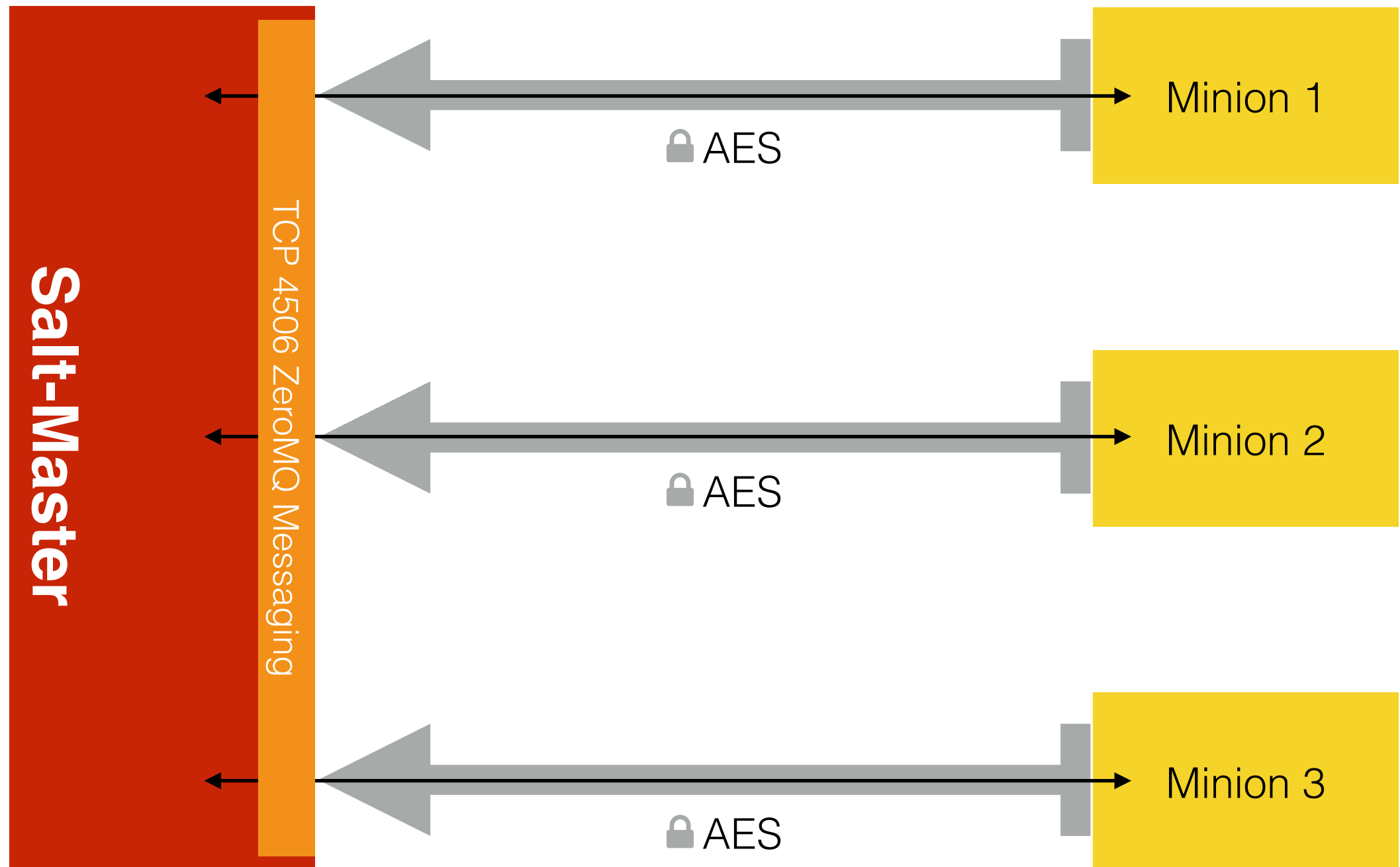

„Du gehörst mir!“

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt-key -L
Accepted Keys:
Unaccepted Keys:
minion1.lab4.org
minion2.lab4.org
minion3.lab4.org
Rejected Keys:
root@salt-master:~# salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
minion1.lab4.org
minion2.lab4.org
minion3.lab4.org
Proceed? [n/Y] Y
Key for minion minion1.lab4.org accepted.
Key for minion minion2.lab4.org accepted.
Key for minion minion3.lab4.org accepted.
root@salt-master:~#
```

„Seit ihr alle da?“

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt * test.ping
minion2.lab4.org:
  True
minion3.lab4.org:
  True
minion1.lab4.org:
  True
root@salt-master:~#
```


hinter den Kulissen



bidirektionale Kommunikation

Befehlen - Salt Module

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt * cmd.run "lsb_release -d"
minion2.lab4.org:
  Description:      Debian GNU/Linux 7.6 (wheezy)
minion3.lab4.org:
  Description:      CentOS release 6.5 (Final)
minion1.lab4.org:
  Description:      Ubuntu 14.04.1 LTS
root@salt-master:~#
```

Ein Befehl - mehrere „Sprachen“

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt * pkg.upgrade
minion2.lab4.org:
-----
acpi-support-base:
-----
new:
  0.140-5+deb7u2
old:
  0.140-5+deb7u1
krb5-locales:
-----
new:
  1.10.1+dfsg-5+deb7u2
old:
  1.10.1+dfsg-5+deb7u1
libgssapi-krb5-2:
-----
new:
  1.10.1+dfsg-5+deb7u2
old:
  1.10.1+dfsg-5+deb7u1
libk5crypto3:
```

Debian/ Ubuntu:
`apt-get update && apt-get -y upgrade`

CentOS/ RedHat:
`yum update`

Beispiel: Benutzer anlegen

- ✓ Benutzer auf allen Systemen anlegen

```
salt * user.add thorsten
```

- ✓ Einen öffentlichen SSH-Schlüssel hinterlegen

```
salt '*' ssh.set_auth_key thorsten AAAAB3NzaC1y...p8cow==  
enc='rsa' comment='kramm@lin4.de'
```

- ✓ Standard-Shell ändern

```
salt '*' user.chshell thorsten /bin/bash
```

Full list of builtin executio...

docs.saltstack.com/en/latest/ref/modules/all/

Suchen

Table of ContentsGlossarypreviousnextall salt modulesindexGet Started

2014.7.6

2015.5.2 ✓

Develop

Announcements

Logging

External Logging Handlers

Salt File Server

Full list of builtin fileserver modules

Salt code and internals

Full list of builtin execution modules

salt.modules.pkg

salt.modules.aliases

salt.modules.alternatives

salt.modules.apache

salt.modules.aptpkg

salt.modules.archive

salt.modules.artifactory

salt.modules.at

salt.modules.augeas_cfg

salt.modules.aws_sqs

salt.modules.blockdev

YouTube

Twitter

Mail

GitHub

Discord

Reddit

Stack Overflow

Slack

289 Module

FULL LIST OF BUILTIN EXECUTION MODULES

Virtual modules

• salt.modules.pkg

aliases	Manage the information in the aliases file
alternatives	Support for Alternatives system
apache	Support for Apache
aptpkg	Support for APT (Advanced Packaging Tool)
archive	A module to wrap (non-Windows) archive calls
artifactory	Module for fetching artifacts from Artifactory
at	Wrapper module for at(1)
augeas_cfg	Manages configuration files via augeas
aws_sqs	Support for the Amazon Simple Queue Service.
blockdev	Module for managing block devices
bluez	Support for Bluetooth (using BlueZ in Linux).
boto_asg	Connection module for Amazon Autoscale Groups
boto_cfn	Connection module for Amazon Cloud Formation

Salt Modules

Salt-Master

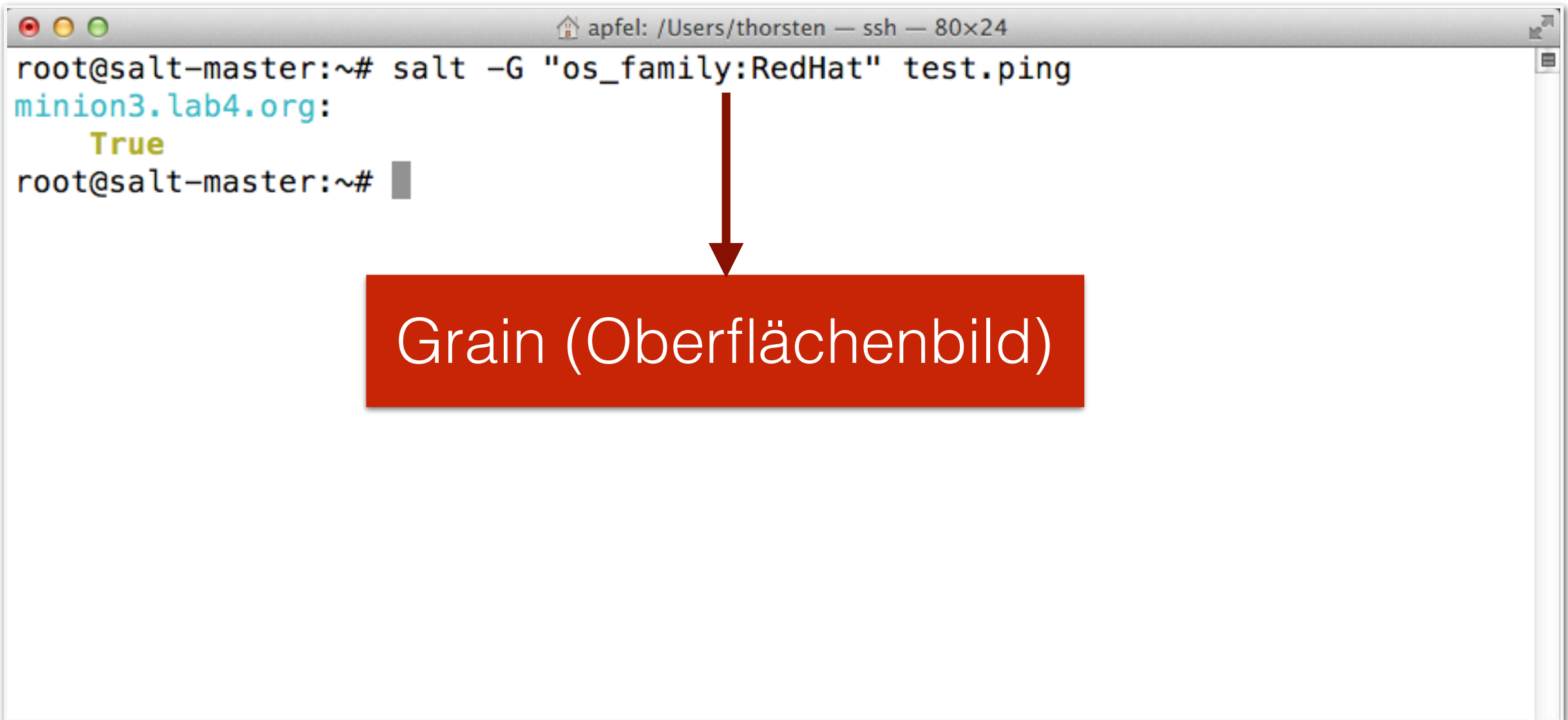
Abstraktionsschicht mit einheitlichen Kommandos

- ✓ Arch Linux
- ✓ Debian
- ✓ Fedora
- ✓ Gentoo

- ✓ RHEL & Clones
- ✓ Solaris
- ✓ Ubuntu
- ✓ SUSE

- ✓ Windows
- ✓ OS X
- ✓ FreeBSD

Minions adressieren



```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt -G "os_family:RedHat" test.ping
minion3.lab4.org:
  True
root@salt-master:~#
```

Grain (Oberflächenbild)

unter der Oberfläche

```
apfel: /Users/thorsten — ssh — 80x24
root@salt-master:~# salt minion1.lab4.org grains.items
minion1.lab4.org:
  biosreleasedate: 01/01/2011
  biosversion: Bochs
  cpu_flags: fpu de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pse36 clf
  lush mmx fxsr sse sse2 syscall nx lm rep_good nopl pni vmx cx16 popcnt hypervisor lah_f_lm
  cpu_model: QEMU Virtual CPU version 1.4.0
  cpuarch: x86_64
  defaultencoding: UTF-8
  defaultlanguage: en_US
  domain: lab4.org
  fqdn: minion1.lab4.org
  fqdn_ip4:
    127.0.1.1
  fqdn_ip6:
  gpus:
    {'model': 'Device 1111', 'vendor': 'unknown'}
  host: minion1
  hwaddr_interfaces: {'lo': '00:00:00:00:00:00', 'eth1': '04:01:24:df:cd:02', 'eth0': '04:01:24:df:cd:01'}
  id: minion1.lab4.org
  ip_interfaces: {'lo': ['127.0.0.1'], 'eth1': ['10.129.163.167'], 'eth0': ['178.62.166.94']}
```

States definieren

/etc/salt/master

```
apfel: /Users/thorsten — ssh — 80x21
# roots cannot match, otherwise the downloaded files will not be able to be
# reliably ensured. A base environment is required to house the top file.
# Example:
# file_roots:
#   base:
#     - /srv/salt/
#   dev:
#     - /srv/salt/dev/services
#     - /srv/salt/dev/states
#   prod:
#     - /srv/salt/prod/services
#     - /srv/salt/prod/states

file_roots:
  base:
    - /srv/salt

# The hash_type is the hash to use when discovering the hash of a file on
# the master server. The default is md5, but sha1, sha224, sha256, sha384
# and sha512 are also supported.
```


Die Kommandobrücke: top.sls

`/srv/salt/top.sls`

```
apfel: /Users/thorsten — ssh — 80x21
#
# This is the top state file
#
base:
  # Apply the following states to all minions
  # All states are defined in a seperate file
  # i.e. sate "foo" is defined in file "foo.sls"

  # Apply to all Linux hosts
  'kernel:Linux':
    - match: grain
    - create_user

~
~
~
~
"top.sls" 16L, 271C 1,1 All
```

Benutzer verwalten

```
/srv/salt/create_user.sls
```

[illegible]

Push & Pull

Minions holen sich alle States ab:

```
root@minion2:~# salt-call state.highstate
```

Master schickt alle States (top.sls):

```
root@master:~# salt minion1.lab4.org state.highstate
```

Master schickt einen State:

```
root@master:~# salt minion1.lab4.org state.sls create_user
```

Pull

Push

apfel: /Users/thorsten — ssh — 80x21

root@salt-master:/srv/salt#



dynamische Dateien mit Jinja

etc-hosts.sls

```
{% set domain = 'example.com' %}  
gw:  
  host.present:  
    - ip: 192.168.152.10  
    - names:  
      - gw  
      - gw.{{domain}}  
  
mail:  
  host.present:  
    - ip: 192.168.152.7  
    - names:  
      - mail  
      - mail.{{domain}}
```

A diagram consisting of a vertical red line on the right side of the code block. Two horizontal red arrows point from this line to the Jinja templated strings in the code: one points to `gw.{{domain}}` in the `gw` section, and the other points to `mail.{{domain}}` in the `mail` section. This illustrates how the `domain` variable is used to dynamically generate hostnames.

dynamische Dateien mit Jinja

users.sls

```
{% for usr in 'moe','larry','curly' %}  
{{ usr }}:  
  group:  
    - present  
  user:  
    - present  
    - gid_from_name: True  
    - require:  
      - group: {{ usr }}  
{% endfor %}
```


dynamische Dateien mit Jinja

apache.sls

```
apache:
  pkg.installed:
    {% if grains['os'] == 'RedHat' %}
    - name: httpd
    {% elif grains['os'] == 'Ubuntu' %}
    - name: apache2
    {% endif %}
```

Dateien zentral verwalten

burp-backup.sls

```
burp:
  pkg:
    - installed

/etc/burp/burp.conf:
  file.managed:
    - source: salt://files/all/etc/burp/burp.conf
    - mode: 0600
    - user: root
    - group: root
    - require: burp
```

Dateien kopieren & bearbeiten

burp-backup.sls

```
burp:
  pkg:
    - installed
/etc/burp/cname.conf:
  file.managed:
    - source: salt://burp-backup/files/cname.conf
    - template: jinja
    - mode: 0600
    - user: root
    - group: root
    - require: burp
```



```
cname = {{ grains['fqdn'] }}
```

Pillars: der Datenspeicher

- ✓ Aktivieren Sie Pillars in /etc/salt/master:

```
pillar_roots:  
  base:  
    - /srv/pillar
```

- ✓ Legen Sie eine top.sls

```
base:  
  '*':  
    - static  
    - packages
```


Pillars definieren Variablen

dynamisch

```
# /srv/pillar/packages.sls
{% if grains['os'] == 'RedHat' %}
apache: httpd
git: git
{% elif grains['os'] == 'Debian' %}
apache: apache2
git: git-core
```

Pillars in States verwenden

```
# /srv/sates/linux/apache.sls
apache:
  pkg.installed:
    - name: {{ pillar['apache'] }}
```

→ apt-get install apache2

→ yum install httpd

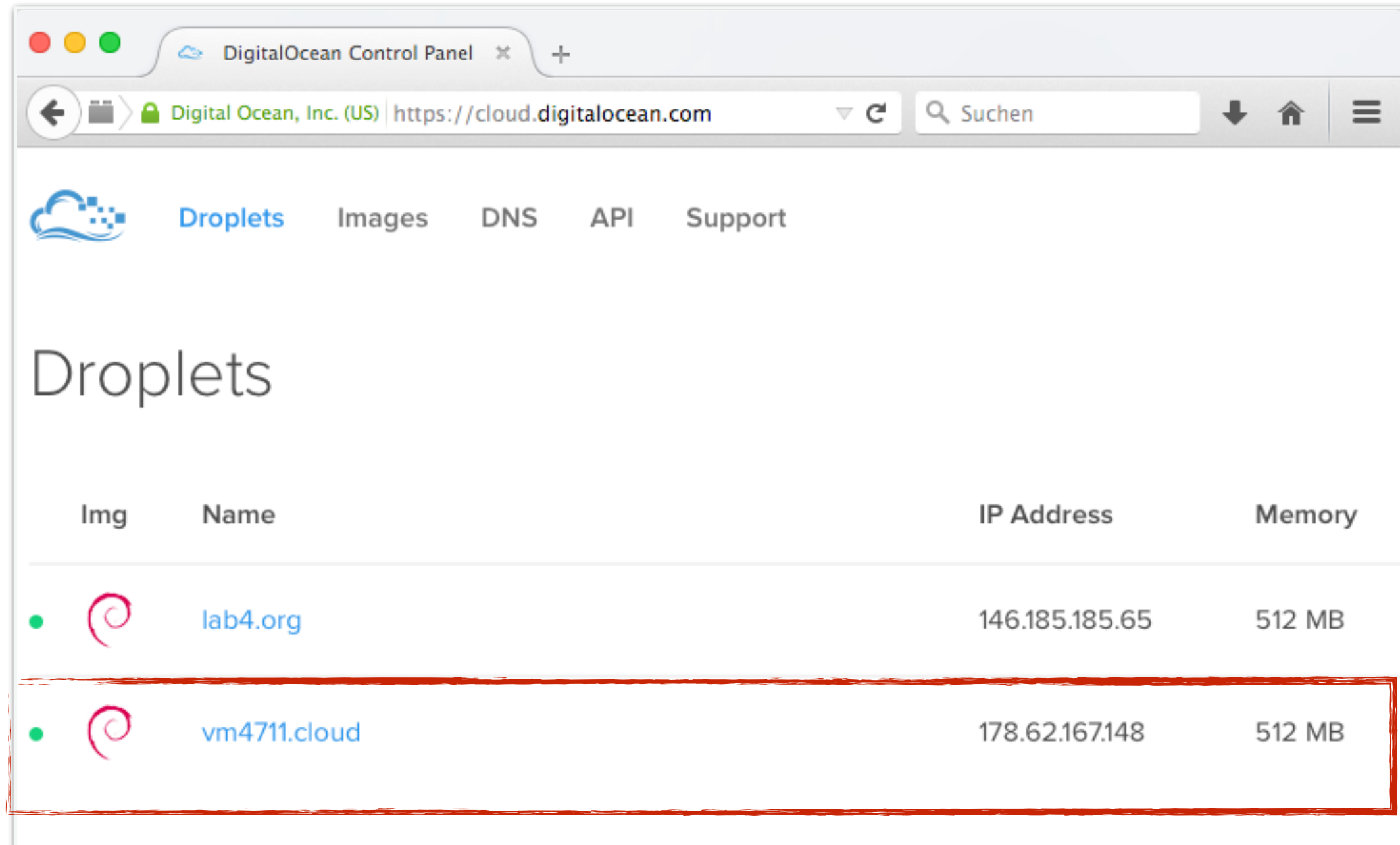
Virtuelle Maschinen und die Cloud



Thorsten Kramm | Berlin Juni 2015

SLAC Secure Linux
Administration
Conference 2015
24.-26. Juni 2015 | Berlin

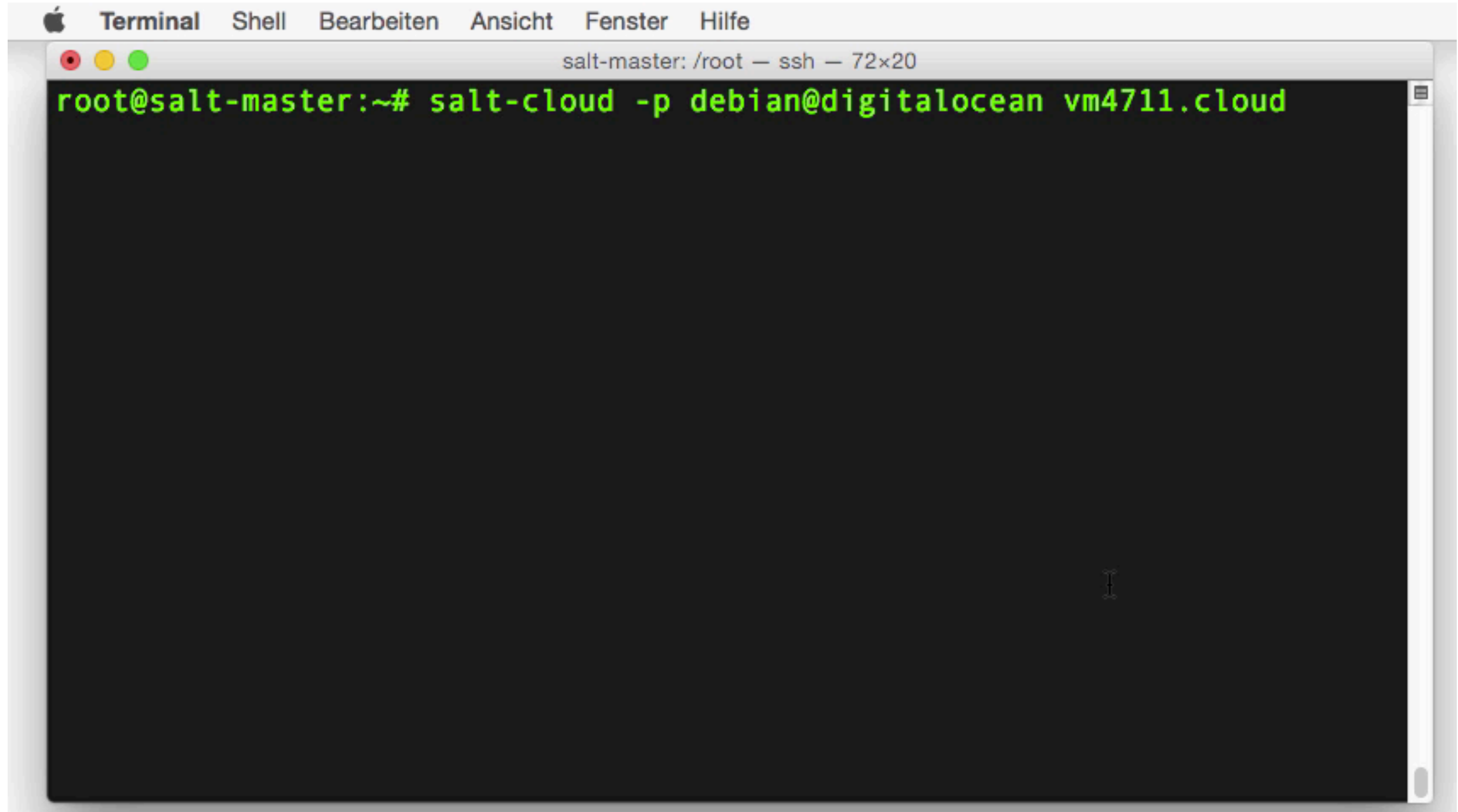
voilà!



The screenshot shows the DigitalOcean Control Panel in a web browser. The browser's address bar displays 'https://cloud.digitalocean.com'. The page has a navigation bar with links for 'Droplets', 'Images', 'DNS', 'API', and 'Support'. The main heading is 'Droplets'. Below it is a table with columns: 'Img', 'Name', 'IP Address', and 'Memory'. There are two rows of droplets. The first row shows a droplet with a green status dot, a red spiral icon, the name 'lab4.org', IP address '146.185.185.65', and '512 MB' of memory. The second row, which is highlighted with a red rectangular box, shows a droplet with a green status dot, a red spiral icon, the name 'vm4711.cloud', IP address '178.62.167.148', and '512 MB' of memory.

Img	Name	IP Address	Memory
● 	lab4.org	146.185.185.65	512 MB
● 	vm4711.cloud	178.62.167.148	512 MB

Hinter den Wolken



A screenshot of a macOS Terminal window. The title bar shows 'Terminal' and standard window controls. The menu bar includes 'Terminal', 'Shell', 'Bearbeiten', 'Ansicht', 'Fenster', and 'Hilfe'. The status bar indicates 'salt-master: /root — ssh — 72x20'. The terminal content shows a green prompt 'root@salt-master:~#' followed by the command 'salt-cloud -p debian@digitalocean vm4711.cloud' entered in green text. The rest of the terminal area is black with a cursor visible.

```
root@salt-master:~# salt-cloud -p debian@digitalocean vm4711.cloud
```


Unterstützte Plattformen

Public

- ✓ Aliyun
- ✓ Azure
- ✓ DigitalOcean
- ✓ EC2
- ✓ GoGrid
- ✓ Google Compute
- ✓ HP Cloud
- ✓ Joyent
- ✓ Linode
- ✓ Rackspace
- ✓ SoftLayer
- ✓ Vexxhost
- ✓ Scaleway

Privat

- ✓ OpenStack
- ✓ Parallels
- ✓ Proxmox
- ✓ VMWare
- ✓ LXC

Wolken konfigurieren

- ☑ Zugangsdaten zu einem Cloud-Provider eintragen

```
# /etc/salt/cloud.providers.d/digital_ocean.conf
digital_ocean_cfg:
  provider: digital_ocean
  personal_access_token: XYZ
  ssh_key_file: /root/.ssh/id_rsa
  ssh_key_names: my_salt_master,Thorsten Kramm
  location: ams2
```

Wolken konfigurieren

- ✓ VM-Templates auf Angebote „mappen“

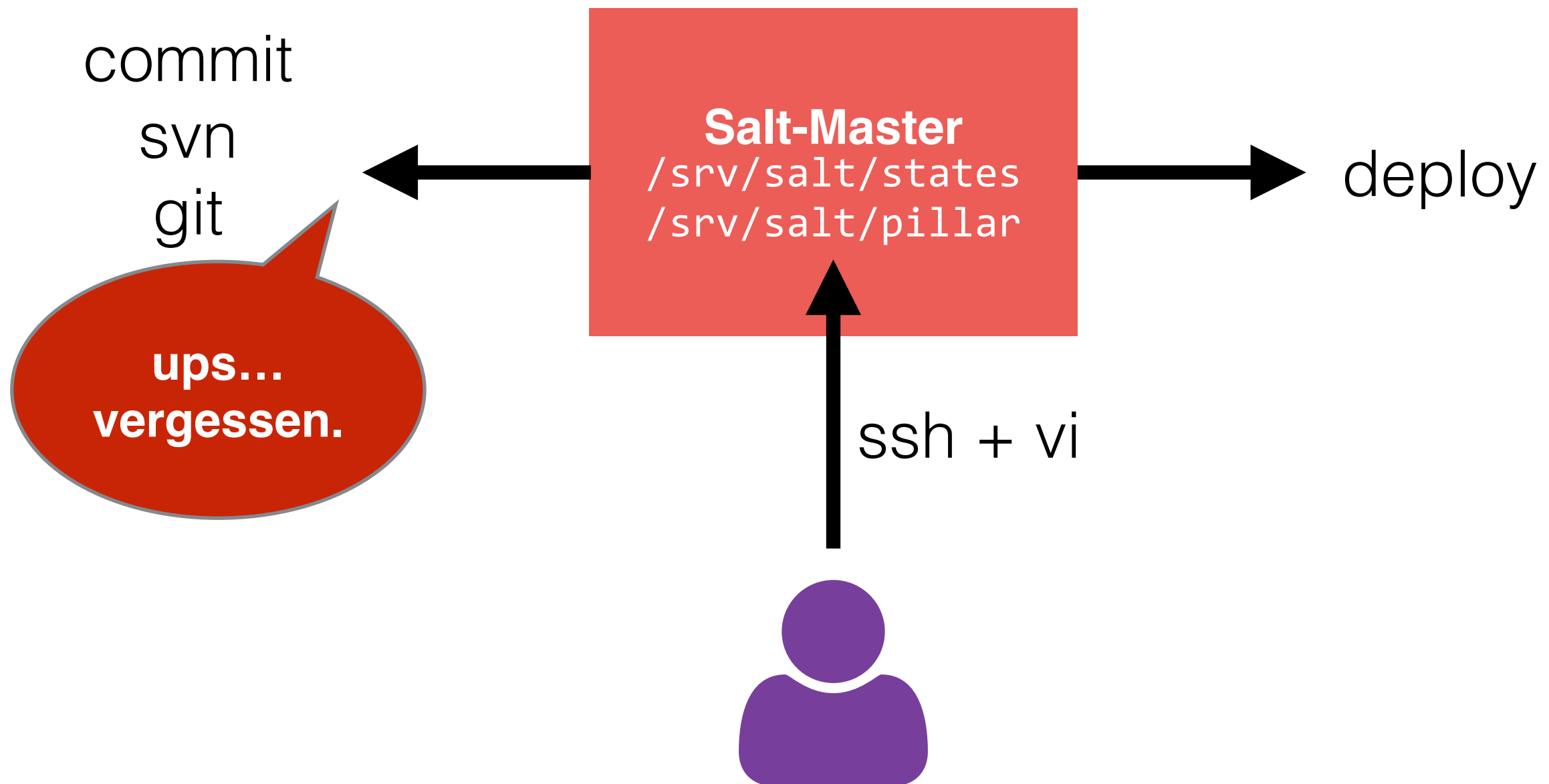
```
# /etc/salt/cloud.profiles.d/digital_ocean.conf
debian@digitalocean:
  provider: digital_ocean_cfg
  image: debian-8-x64
  size: 512MB
  location: ams2
  private_networking: False
```

Versionskontrolle mit git

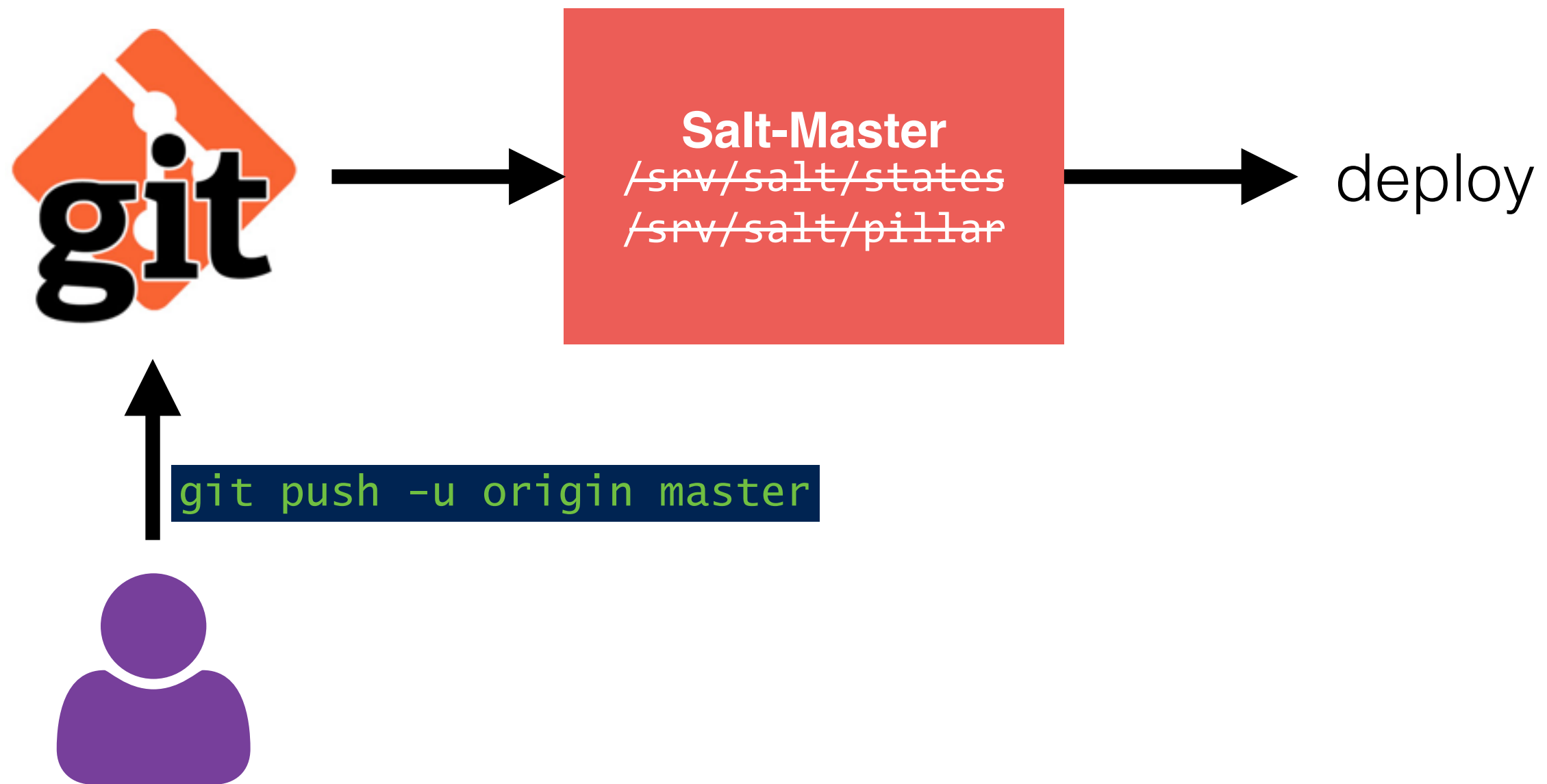
Warum man ein VCS nutzen sollte:

- ✓ Änderungen protokollieren
- ✓ Arbeiten im Team ermöglichen
- ✓ Rollouts kontrollieren
- ✓ Qualitätssicherung „vorschalten“

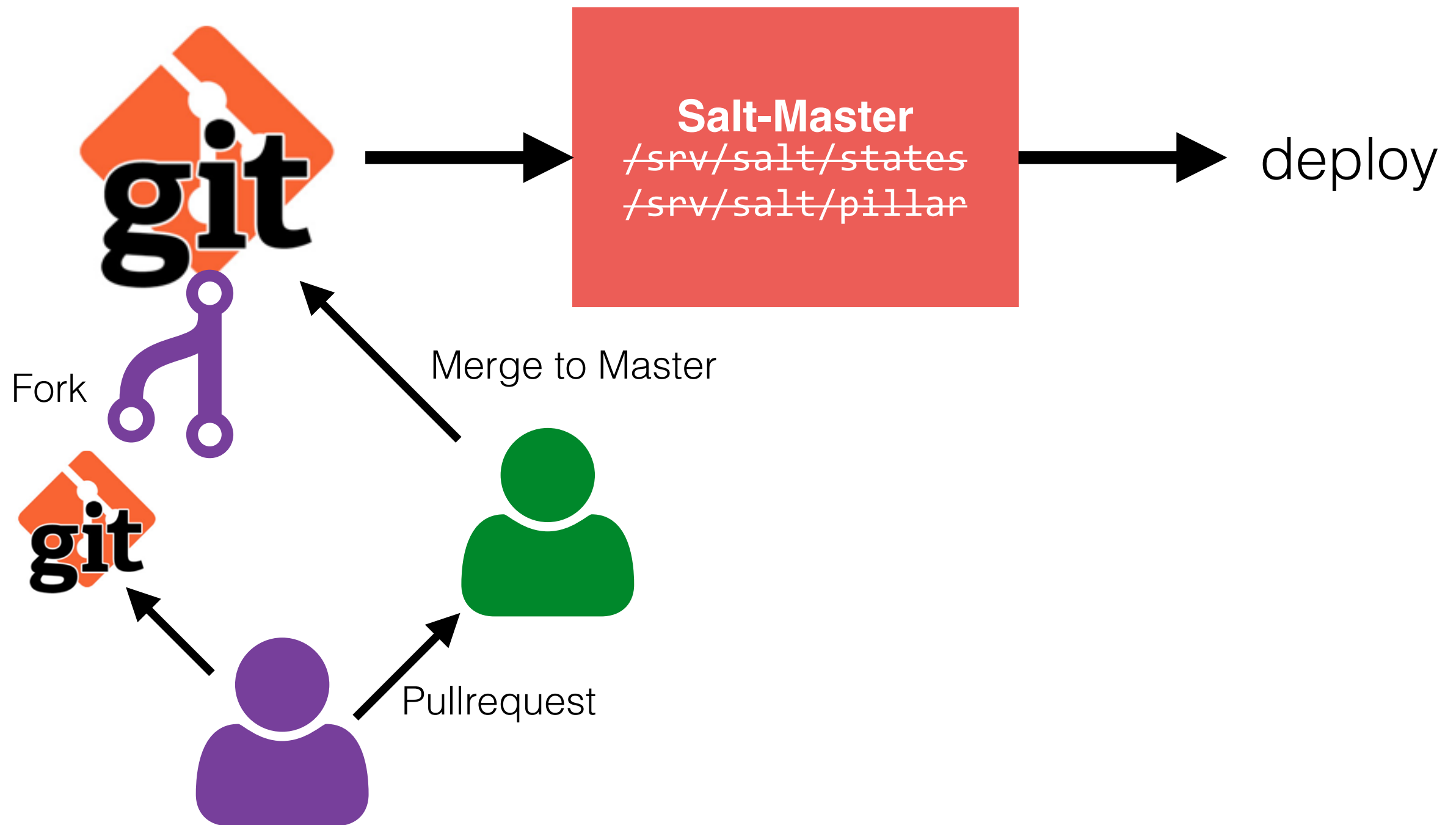
Arbeiten mit VCS Version 1



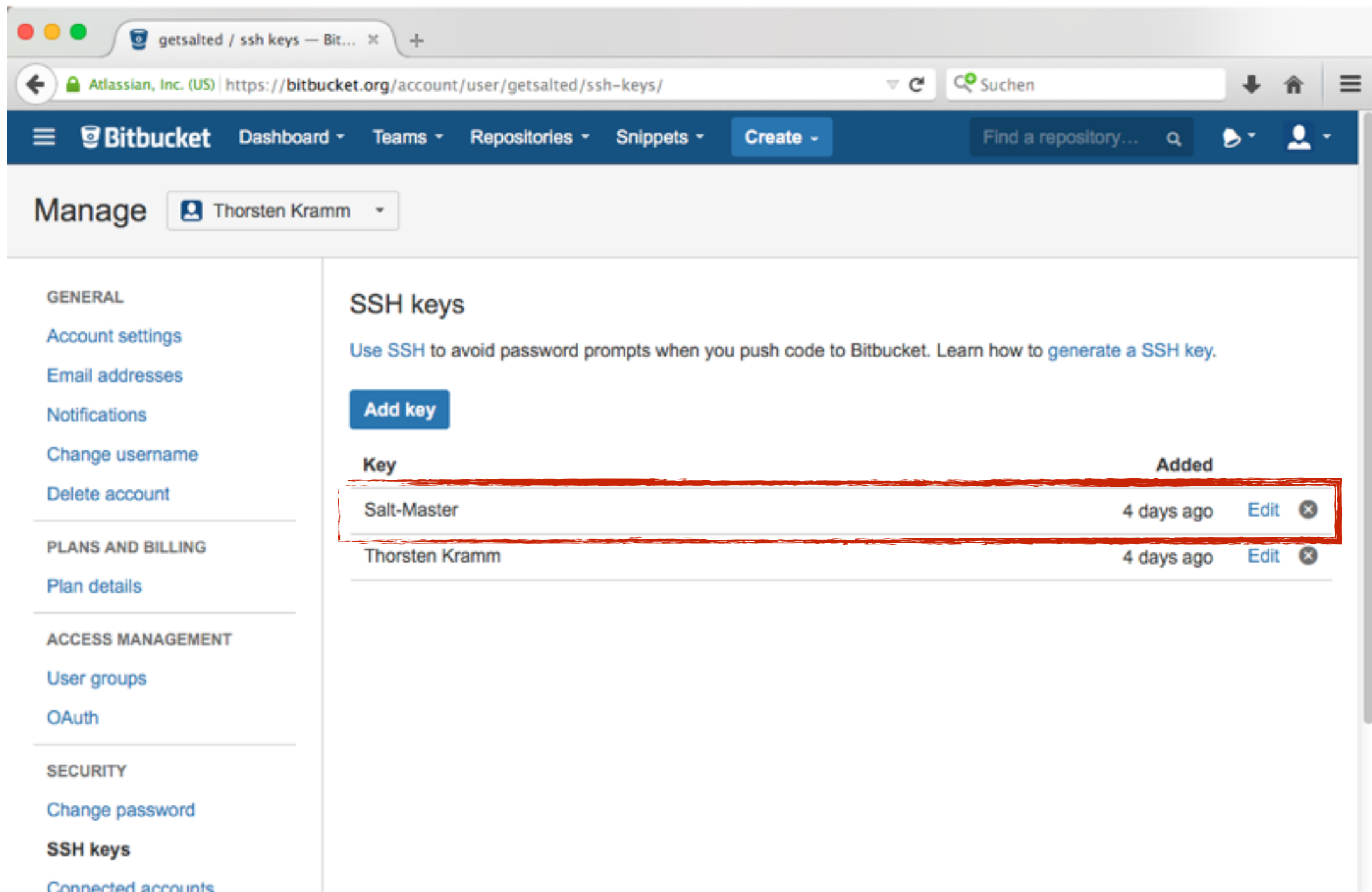
Arbeiten mit VCS Version 2



Arbeiten mit VCS Version 3



Salt-Gitfs



The screenshot shows the Bitbucket web interface for user 'getsalted'. The page title is 'SSH keys'. A red rectangle highlights the first entry in the SSH keys table, which is named 'Salt-Master' and was added '4 days ago'. The table has columns for 'Key', 'Added', and actions ('Edit' and 'Delete').

Key	Added	
Salt-Master	4 days ago	Edit ×
Thorsten Kramm	4 days ago	Edit ×

Salt-States & Gitfs

```
# /etc/salt/master
fileserver_backend:
  - git

gitfs_remotes:
  - git@bitbucket.org:getsalted/salt-states.git:
    - user: getsalted
    - pubkey: /root/.ssh/id_rsa.pub
    - privkey: /root/.ssh/id_rsa
    - root: states
```

Salt-Pillar & Gitfs

```
# /etc/salt/master
pillar_roots:
  base:
    - /srv/salt/pillar-empty

ext_pillar:
  - git: master https://user:pass@bitbucket.org/getsalted/salt-
    states.git root=pillar
```

Authentifizierung per SSH-Key leider noch nicht möglich.

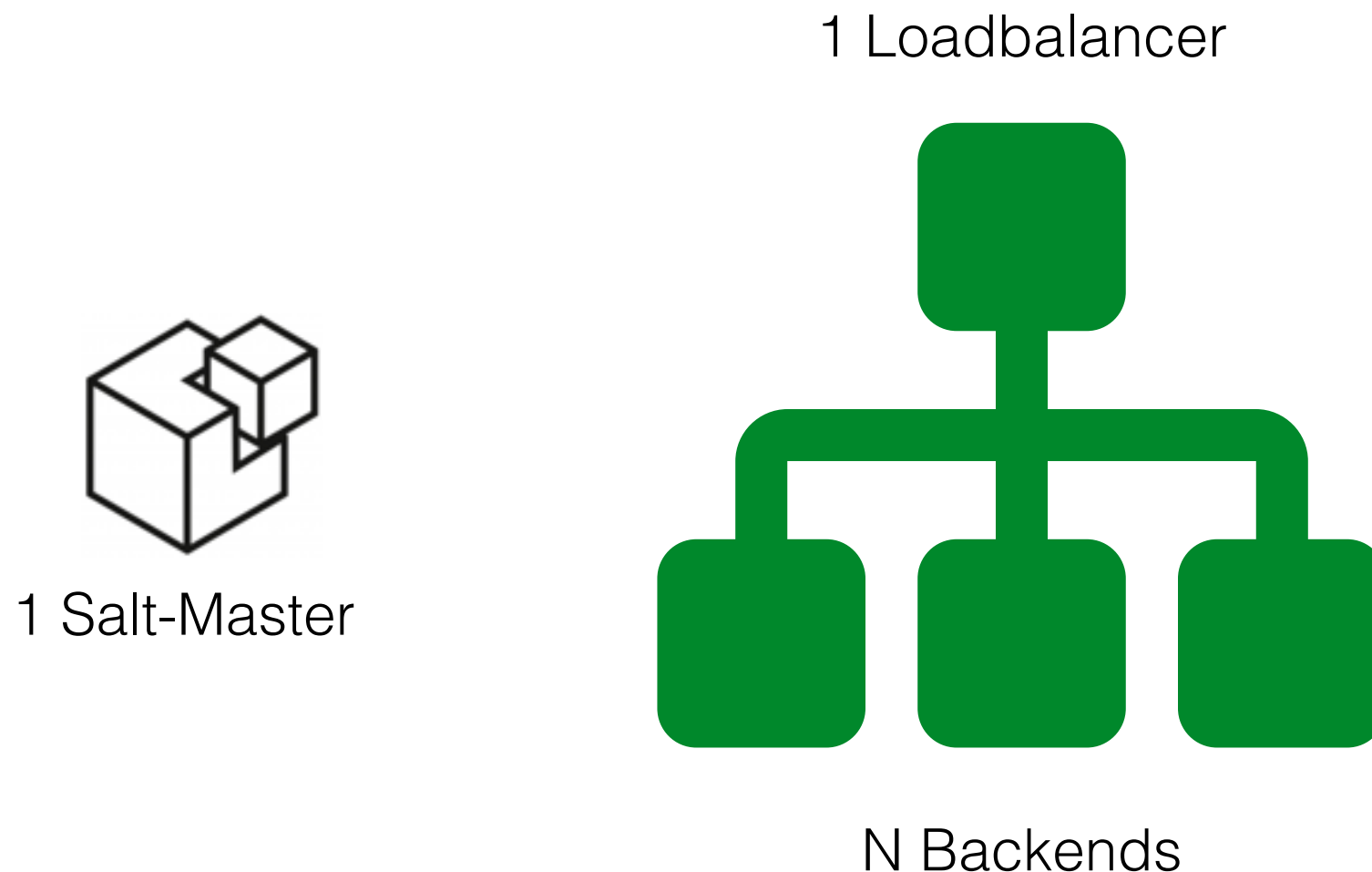
dynamische Pillars



```
# /etc/salt/master
my_etcd_config:
  etcd.host: 127.0.0.1
  etcd.port: 2379

ext_pillar:
  - etcd: my_etcd_config
```

Beispiel: Webcluster



- ✓ Last der Backends überwachen

```
salt-master:~# salt '*.backends.local' status.loadavg
```

Schwitz! Die Last ist zu hoch.

- ✓ Neue VM erzeugen

```
salt-master:~# salt-cloud -p lxc-ubuntu 4.backends.local
```

- ✓ Mache die neue VM zum Backend

```
salt-master:~# salt 4.backends.local state.sls web-backend
```

- ✓ Neues Backend klinkt sich in Cluster ein

```
4.backends:~# curl http://salt:2379/v2/keys/backends/\
$(hostname -f) -XPUT -d value="online"
```

- ✓ Master aktualisiert die Loadbalancer-Konfiguration

```
salt-master:~# salt loadbalancer.local state.sls nginx
```

loadbalancer.conf

```
http {
    upstream myapp1 {
        {% for key, value in pillar['backends'].iteritems() %}
        server {{key}};
        {% end for %}
    }
}
```

nginx.sls

```
nginx:
  service.running:
    - enable: True
    - reload: True
    - watch:
      - file: loadbalancer.conf
```

Thorsten Kramm

kramm@lin4.de