

Automate Monitoring with Salt and Checkmk

—

Philipp Lemke

February / 2020

ConfigManagementCamp Ghent



Agenda

1. **About me**
2. **Project salt-checkmk**
3. **Salt & Checkmk Synergies**
4. **Setting up a monitoring environment from scratch**
 - 4.1. Install Checkmk
 - 4.2. Add your Salt-Minions automatically to checkmk
 - 4.3. Install checkmk Monitoring Agents via Salt
 - 4.4. Use Salt Grains to define Rules in checkmk
5. **Checkmk Notifications / Salt Event Bus / Reactors**
 - 5.1. ACL Definitions
 - 5.2. Install Notification Plugin & Define Rule
 - 5.3. Define Salt Reactor Config & Reactor
 - 5.4. Invoke Orchestration Runner
 - 5.5. Example(s)

About me

Philipp Lemke
Trier / Germany

@Agfa HealthCare
**Solution Architect AMS
Automation / Monitoring**
philipp.lemke@agfa.com
www.agfahealthcare.com



@Sideline
Freelancer / Consultant
pl@philipp-lemke.de
www.philipp-lemke.de
Github: PhilippLemke



Project salt-checkmk



<https://github.com/tribe29/salt-checkmk>

States / Modules

Execution- & State Modules
SLS Example States

Documentation

In Salt & Checkmk User
context

Formula

Content provided as Salt
Formula

Synergies

SALTSTACK



checkmk

- Intelligent Automation
- Python Environment
- Asset Data (Grains)

- Intelligent Monitoring
- Detailed Health Info
- Notifications

Both

- Compatible with most OS
- Written in Python
- Scalability:
Master-of-Masters Concept

Distributed Topology Support

Master-of-Masters

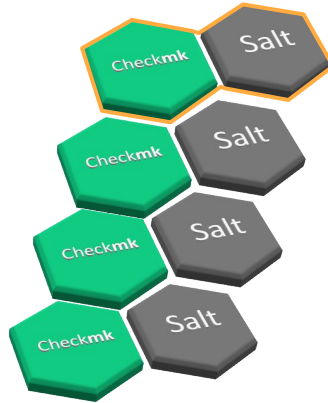
Checkmk Master



Central Data Center

Salt Syndic

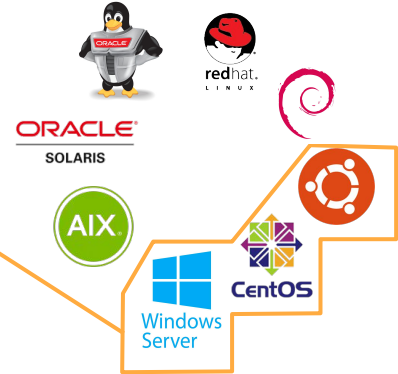
Checkmk Slave



Remote Sites / Data Centers

Minions

Hosts

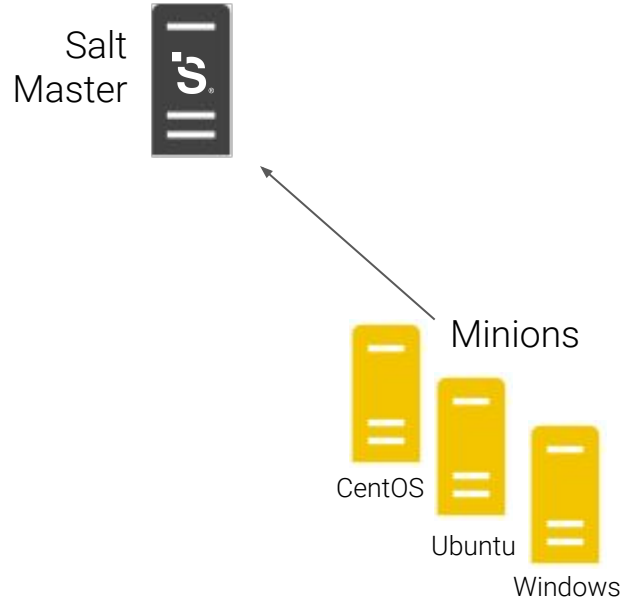


Setting up Checkmk & Minion Monitoring



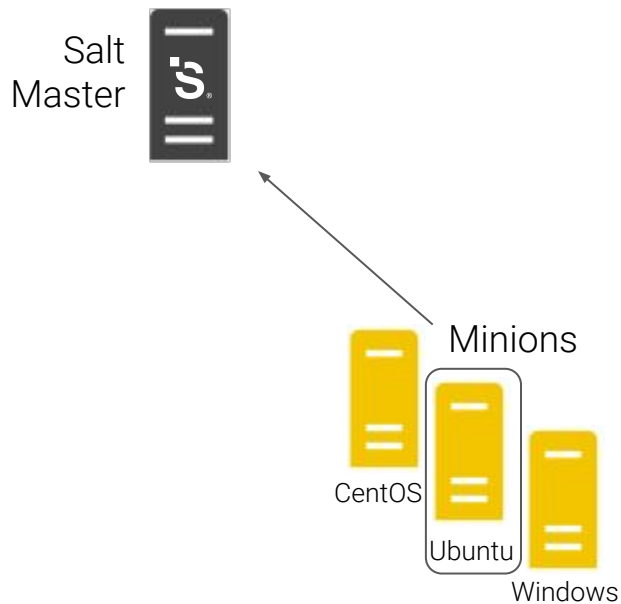
Starting Point

Salt Environment with Master & Minions



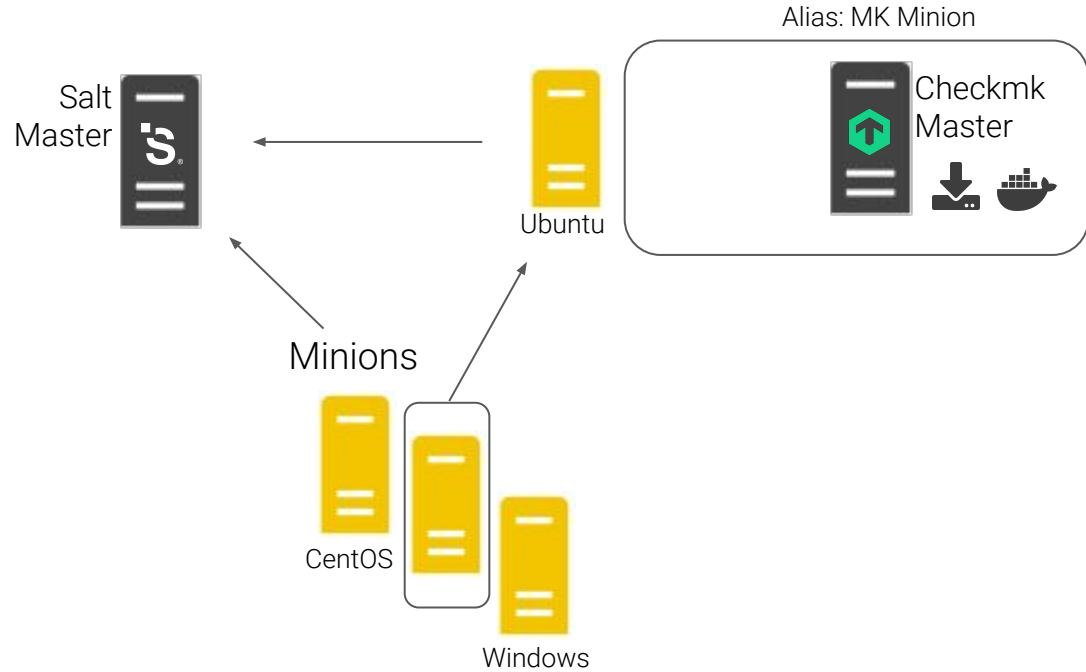
Prepare CheckMK setup

Designate one Minion to setup checkmk



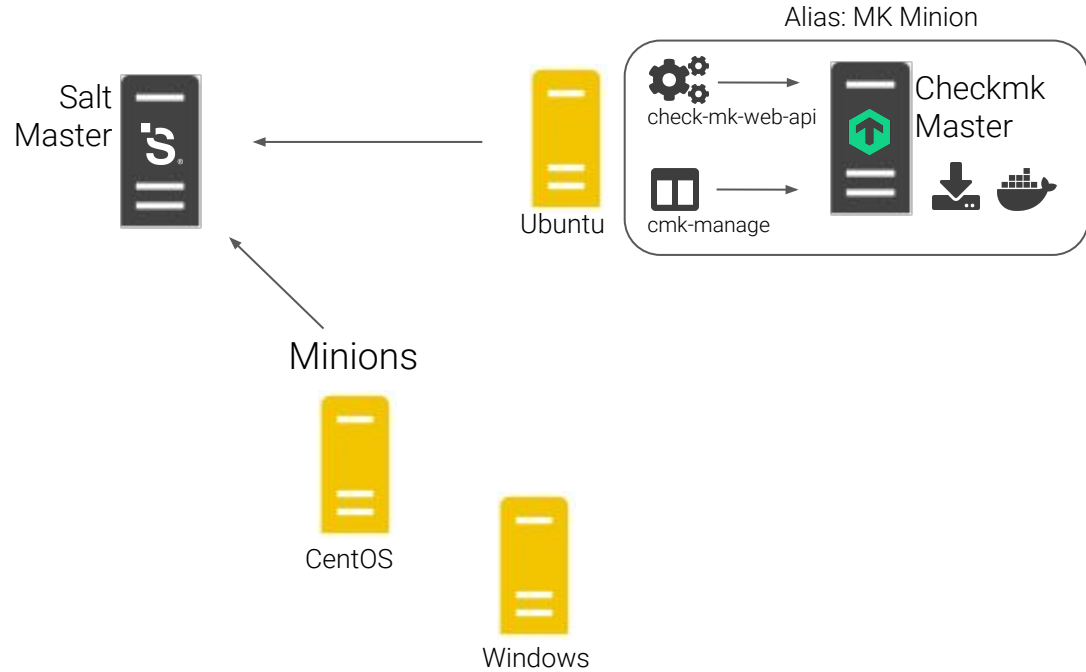
Setup CheckMK

Install a checkmk Master on that Minion



Manage CheckMK

Management by Execution and State Modules



Demo

The screenshot displays the Checkmk web interface in a browser window and a terminal window below it.

Browser Window:

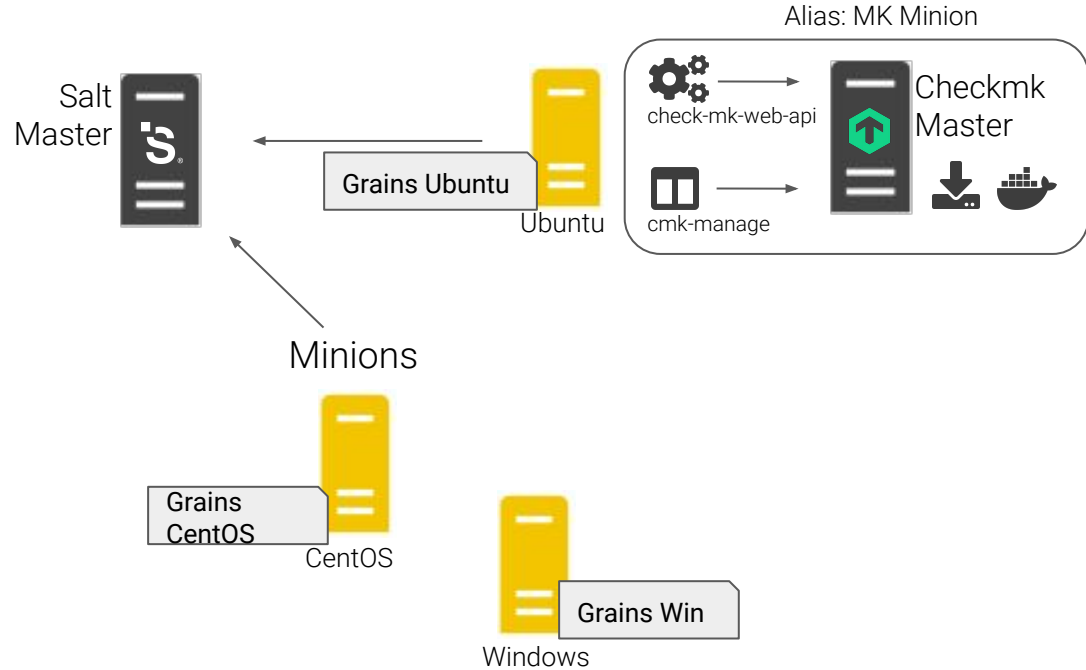
- Address bar: `localhost:8080/cm/check_mk/index.py?start_url=%2Fcmk%2Fcheck_mk%2Fwato.py%3Fed...`
- Page title: `checkmk`
- Left sidebar (WATO - CONFIGURATION):
 - Main Menu
 - Monitoring Agents
 - Hosts
 - Tags
 - Global Settings
 - Host & Service Parameters
 - Manual Checks
 - Check Plugins
 - Host & Service Groups
 - Users
 - Roles & Permissions
 - Contact Groups
 - Notifications
 - Time Periods
- Right pane: `Edit user automation` (cmkadmin (admin) 11:57)
 - repeat: (optional)
 - Enforce change: ☐ Change password at next login or access
 - Automation secret for machine accounts: `09548093-bf80-426c-b073-88f16fb080a5`
 - ☐ disable the login to this account
 - Use the global configuration (dropdown)
 - ☒ Administrator
 - ☐ Guest user
 - ☐ Normal monitoring user
 - CONTACT GROUPS
 - NOTIFICATIONS
 - PERSONAL SETTINGS

Terminal Window:

```
pl@arcticskate: ~
cmk_user=automation cmk_secret=09548093-bf80-426c-b073-88f16fb080a5
mk-minion:
-----
automation:
-----
alias:
-----
  Check MK Automation - used for calling web services
automation secret:
  09548093-bf80-426c-b073-88f16fb080a5
contactgroups:
email:
enforce_pw_change:
  False
language:
  en
last_pw_change:
  1570470963
locked:
  False
notifications_enabled:
```

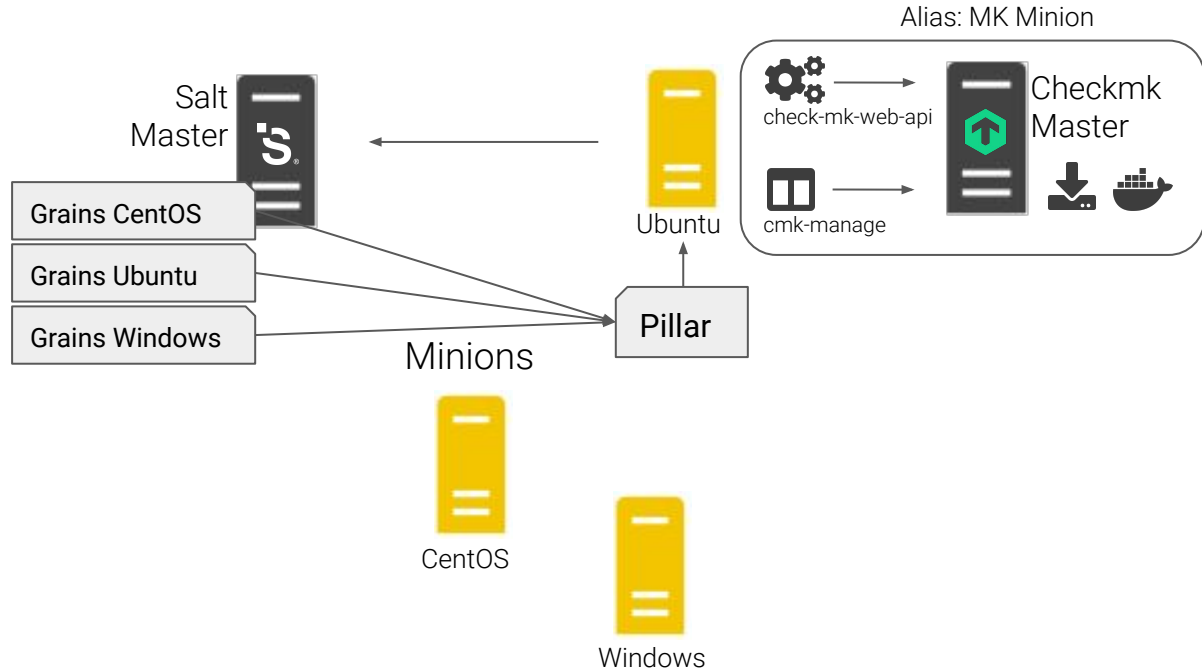
Add Minions to Checkmk

Gather Grains from Minions



Add Minions to Checkmk

Provide "MK Minion" Grains as Pillar



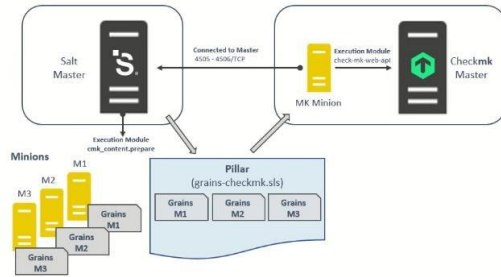
Apply State **add-minions** on MK Minion



Demo

```
base:
  '<replace with your MK Minion ID>':
    - cmk-base
    - grains_checkmk
```

Gather Grains from Minions



Gather grains from all minions that you plan to add to the monitoring site. Executed on your Salt-Master

```
salt mk-minion cmk_content.prepare pillar_file=/srv/pillar/grains_checkmk.sls target="*" tg
```

Remark: In general I prefer salt-mine to get this job done, but `cmk_content.prepare` will also work fine in complex master-of-master setups.

Test / Show pillar data

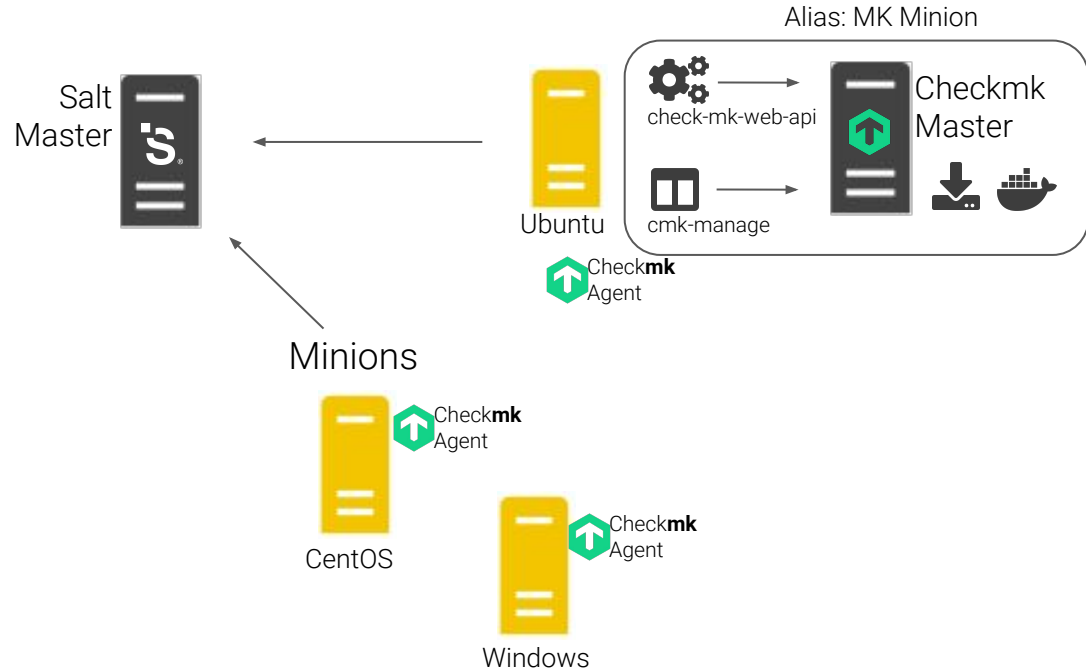
```

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL
[3: Shell: checkmk/ched + [ ] [ ] [ ] [ ]
pi@artciskate:~$ cp /srv/formulas/salt-checkmk/pillar.example /srv/pillar/cmik-base.sls
pi@artciskate:~$ vi /srv/pillar/cmik-base.sls
pi@artciskate:~$ vi /srv/pillar/top.sls
pi@artciskate:~$

```


Install CheckMK Agents

Install CheckMK Agent via Salt



Install CheckMK Agents

Use the salt **pkg** Module / State :-)

Install CheckMK Agents

Remark for Windows-Minions

Before Checkmk 1.6 -> Check productversion of Agent

msiinfo export check_mk_agent.msi Property

Property	Value
s72	l0
Property	Property
UpgradeCode	{8548B2C1-F4AA-4C5C-89E4-8FB0BDFB9EE4}
INSTALLSERVICE	1
WixUIRMOption	UserM
WIXUI_INSTALLDIR	INSTALLDIR
ALLUSERS	1
ARPNO MODIFY	yes
REINSTALLMODE	amus
Manufacturer	Mathias Kettner GmbH
ProductCode	{5847CFEA-6802-11E9-92CC-0CC47ADF0684}
ProductLanguage	1033
ProductName	Check MK Agent
ProductVersion	1.5.0.3270
DefaultUIFont	WixUI_Font_Normal
WixUI_Mode	InstallDir
ErrorDialog	ErrorDlg
SecureCustomProperties	ARPNO MODIFY;EXISTINGINSTALLDIR;

Install CheckMK Agents

The Key in salt-winrepo must match with ProductVersion in MSI-Package

```
check-mk-agent:
  '1.6.0p3':
    full_name: 'Check MK Agent 1.6'
    installer: 'salt://blobs/check_mk/1.6.0p3/share/check_mk/agents/windows/check_mk_agent.msi'
    install_flags: '/qn /norestart'
    uninstaller: 'salt://blobs/check_mk/1.6.0p3/share/check_mk/agents/windows/check_mk_agent.msi'
    uninstall_flags: '/qn /norestart'
    msiexec: True
    locale: en_US
    reboot: False

  '1.5.0.3270':
    full_name: 'Check MK Agent 1.5'
    installer: 'salt://blobs/check_mk/1.5.0p16/share/check_mk/agents/windows/check_mk_agent.msi'
    install_flags: '/qn /norestart'
    uninstaller: 'salt://blobs/check_mk/1.5.0p16/share/check_mk/agents/windows/check_mk_agent.msi'
    uninstall_flags: '/qn /norestart'
    msiexec: True
    locale: en_US
    reboot: False
```

Demo

Install checkmk Agent via Salt

Windows

Refresh package database on Windows Minions to publish the agent package from formula

```
salt -G 'os:windows' pkg.refresh_db
```

Install checkmk Agent on your Minions

```
salt '*' state.apply cnk-manage.deploy-cnk-agent
```

```

OUTPUT  DEB CONSOLE  TERMINAL
[3: Shell: checkmk/check▼ + □ ▢ ✕
-----
failed:
0
success:
1
total:
1
pi@articskate:~$ salt '*' state.apply cmk-manage.deploy-cmk-agent
mk-minion:
-----
ID: check-mk-agent
Function: pkg.installed
Result: True
Comment: The following packages were installed/updated: check-mk-agent
Started: 15:46:55.927637
Duration: 4197.749 ms
Changes:

-----
check-mk-agent:

new:
1.6.0p3-1

old:

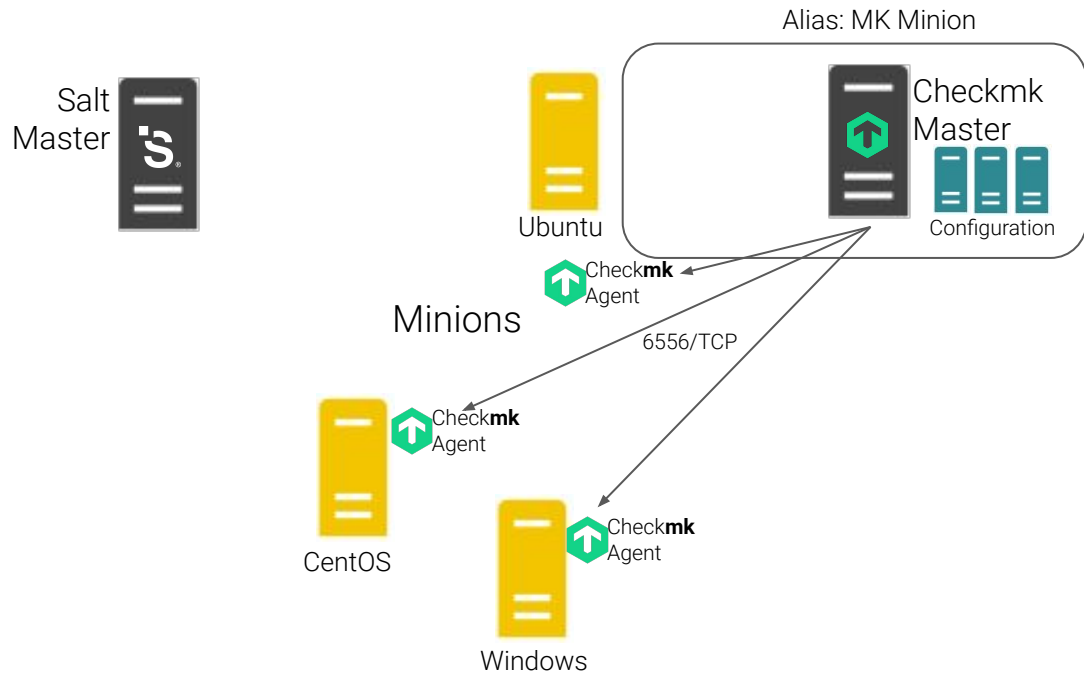
Summary for mk-minion
-----
Succeeded: 1 (changed=1)
Failed: 0
-- 1 -----
Total states run: 1
Total run time: 4.198 s
ubuntu-srv1:

-----
ID: check-mk-agent
Function: pkg.installed
Result: True
Comment: The following packages were installed/updated: check-mk-agent
Started: 15:46:56.444893
Duration: 6078.522 ms

```



Monitor your Minions



Demo

checkmkRaw 1.6.0p3

TACTICAL OVERVIEW

Hosts	Problems	Unhandled
0	0	0
Services	Problems	Unhandled
0	0	0
Events	Problems	Unhandled
0	0	0

QUICKSEARCH

BOOKMARKS

ADD BOOKMARKEDIT

VIEWS

WATO - CONFIGURATION

- Main Menu
- Monitoring Agents
- Hosts
- Tags
- Global Settings
- Host & Service Parameters
- Manual Checks
- Check Plugins
- Host & Service Groups
- Users
- Roles & Permissions
- Contact Groups
- Notifications
- Time Periods
- Event Console

© TRIBEZ29 GMBH

Checkmk Local site cmk - / x +

← → ↻ ⌂ ⓘ Not secure | 192.168.56.1:8080/cmk/check_mk/index.py?start_url=%2Fcmk%2Fcheck_mk%2Fwato.py%3Ffolder%3D%26mod...

Activate pending changes9 rows cmkadmin (admin) 22:49

Main MenuDiscard Changes!Site ConfigurationAudit Log

Activating...

▼ ACTIVATE PENDING CHANGES

Comment (optional)

Activate foreign changes

☒ Activate changes of other users

There are some changes made by your colleagues that you will activate if you proceed. You need to enable the checkbox above to confirm the activation of these changes.

Activate affectedActivate selected

Activation status

ACTIONS	SITE	STATUS	VERSION	CHANGES	PROGRESS	DETAILS
 	Local site cmk	online	1.6.0p3	9	Activating	Started at: 22:49:52. Approximately finishes in 2.8 seconds

Pending changes

OBJECT	TIME	USER	AFFECTED SITES	CHANGE
	2019-11-13 22:49:04	cmkadmin	All sites	Did service discovery on host win10vm: 0 added, 0 removed, 19 kept, 19 total services and 0 host labels added, 0 host labels total
	2019-11-13 22:49:04	cmkadmin	All sites	Did service discovery on host ubuntu-srv1: 0 added, 0 removed, 29 kept, 29 total services and 0 host labels added, 1 host labels total
	2019-11-13 22:49:04	cmkadmin	All sites	Did service discovery on host mk-minion: 0 added, 0 removed, 25 kept, 25 total services and 0 host labels added, 1 host labels total
	2019-11-13 22:49:04	cmkadmin	All sites	Did service discovery on host centosvm1: 0 added, 0 removed, 19 kept, 19 total services and 0 host labels added, 0 host labels total
	2019-11-13 22:48:00	automation	All sites	Created new host ubuntu-srv1

Salt Grains in checkmk

▼ CONDITIONS

Condition type Explicit conditions ▼

Folder Main directory ▼

Host tags

 Salt Grains / manufacturer	is ▼	IBM ▼
 Salt Grains / osfinger	is ▼	AIX-7 ▼
 Salt Grains / productname	is ▼	8286-42A ▼

..... ▼ [Add tag condition](#)

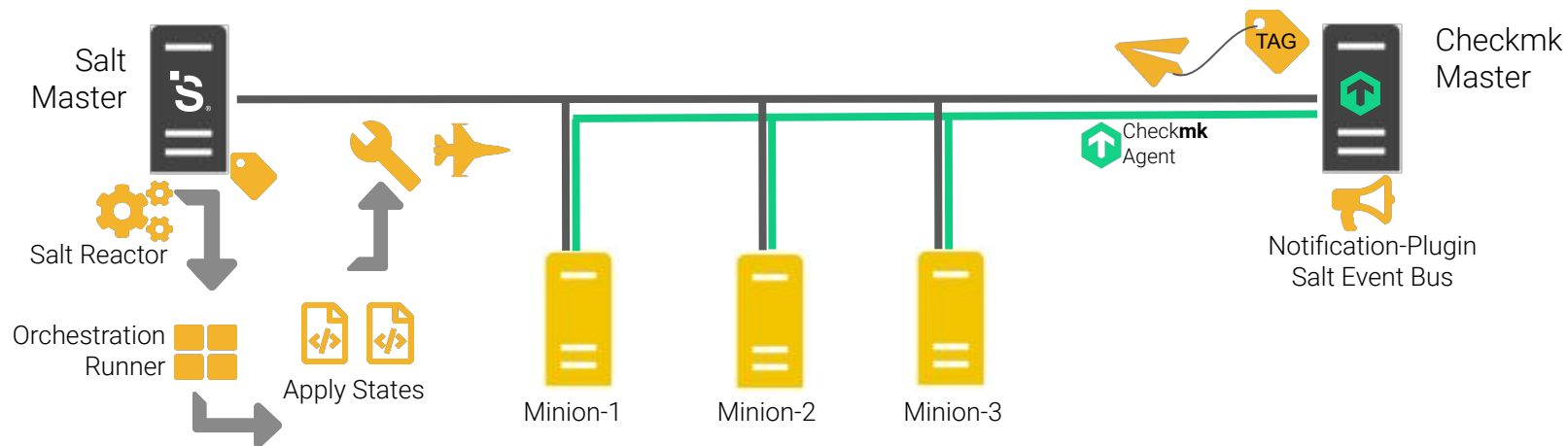
 Salt Grains / manufacturer	is ▼	HPE ▼
 Salt Grains / productname	is ▼	ProLiant DL380 Gen9 ▼
 Salt Grains / kernel	is ▼	Linux ▼

Notifications / Events / Reactors

to Support the Operation-Team



Notifications / Events / Reactors



Why should someone do that?

There are still methods in place to do this job:

Salt Beacons

Checkmk Alert Handlers

There are some disadvantages

- **Salt Beacons** let you use the Event-Bus to monitor stuff at your minions but it's not a replacement for an entire monitoring solution
 - User & Threshold Management
 - 1000+ "out of the box" Checks ...
- **Checkmk Alert Handlers** let you use scripts to react on monitoring events.
Not a replacement for a Data Center Automation Solution
 - No -> States, Orchestration, Requisites, ...

Benefits of this method

- **Use the strengths of each solution**
- **Salt: Automation Framework**
 - File Roots, Pillars, Grains, ...
 - Use existing State Files
 - Create relationships between states (Salt requisite system)
 - Possibility to Orchestrate thinks “cross host borders”
- **Checkmk: Monitoring Solution**
 - A huge bunch of “out of the box” checks
 - Automatic Service Discovery
 - Sophisticated Rule Management

Steps for Implementation

- Define Salt ACL for Monitoring Site
- “Install” Notification Plugin & Define Rule
- Define Salt Reactor Config & Reactor
- Invoke Orchestration Runner

Define ACL for Monitoring Site

@Salt Master

```
publisher_acl:  
  site1:  
    - event.send
```

OMD site name = User

```
OMD[site1]:~$ salt mk-minion test.ping  
Authorization error occurred.  
OMD[site1]:~$ salt mk-minion event.send test/notification data=foo  
mk-minion:  
  True
```

@Salt Minion

<https://docs.saltstack.com/en/latest/ref/configuration/nonroot.html>

Notification Plugin

~/local/share/check_mk/notifications/salt-eventbus

```
#!/usr/bin/python
# Salt Event Bus
import sys
import os

if __name__ == "__main__":
    sys.path.append('%s/lib/python' % os.environ['HOME'])
    from cmk.notification_plugins import utils

    # get notification context as dict
    context = utils.collect_context()

    # import salt client and fire event to master
    import salt.client
    caller = salt.client.Caller()
    caller.cmd('event.fire_master', context, 'checkmk/notification')
```


Create Notification Rule



Notifications

Rules for the notification of contacts about host and service problems

▼ RULE PROPERTIES

Description

send notification event

Comment

Documentation URL

Rule activation

☐ do not apply this rule

Overriding by users

☐ allow users to deactivate this notification

▼ NOTIFICATION METHOD

Notification Method

Salt Event Bus ▼

Call with the following parameters: ▼

Notification Bulking



Example Event

salt-run state.event pretty=True

```
checkmk/notification {
  "_stamp": "2020-01-28T09:23:57.672410",
  "cmd": "_minion_event",
  "data": {
    "CONTACTALIAS": "salt automation user",
    "CONTACTNAME": "salt",
    "HOSTADDRESS": "192.168.56.4",
    "HOSTNAME": "centosvm1",
    "HOSTSTATE": "UP",
    "HOSTSTATEID": "0",
    "HOSTTAGS": "/wato/ auto-piggyback cmk-agent ip-v4 ip-v4-only lan no-snmp prod site:site1 tcp",
    "OMD_SITE": "site1",
    "SERVICECHECKCOMMAND": "check_mk-cpu.loads",
    "SERVICEDESC": "CPU load",
    "SERVICEOUTPUT": "OK - 15 min load: 0.05",
    "SERVICEPERFDATA": "load1=0;5;10;0;1 load5=0.02;5;10;0;1 load15=0.05;5;10;0;1",
    "SERVICESTATE": "OK",
    "SHORTDATETIME": "2020-01-28 10:23:51",
    "SVC_SL": "",
    "WHAT": "SERVICE"
  },
  },
  "id": "mk-minion",
  "tag": "checkmk/notification"
}
```

CMK Context

Event TAG

Reactor Config @ Salt Master

Event TAG

```
reactor:  
  - 'checkmk/notification':  
    - /srv/reactor/checkmk.sls
```

Reactor

```
{% if data['data']['SERVICECHECKCOMMAND'] == 'check_mk-df' and data['data']['SERVICESHORTSTATE'] == 'WARN' %}
```

```
  invoke_orch_lvm_ext:
    runner.state.orchestrate:
      - args:
        - mods: orch.lvm_ext
        - pillar:
            event_data: {{ data|json}}
      {% endif %}
```

```
{% if data['data']['SERVICECHECKCOMMAND'] == 'check_mk-oracle_tablespaces' and data['data']['SERVICESHORTSTATE'] == 'WARN' %}
```

```
  invoke_orch_oracle_tablespaces_ext:
    runner.state.orchestrate:
      - args:
        - mods: orch.oracle_tablespaces_ext
        - pillar:
            event_data: {{ data|json}}
      {% endif %}
```

Orchestration

```
{% set data = salt.pillar.get('event_data') %}  
{% set volume=data['data']['SERVICEDESC'].split()[1] %}
```

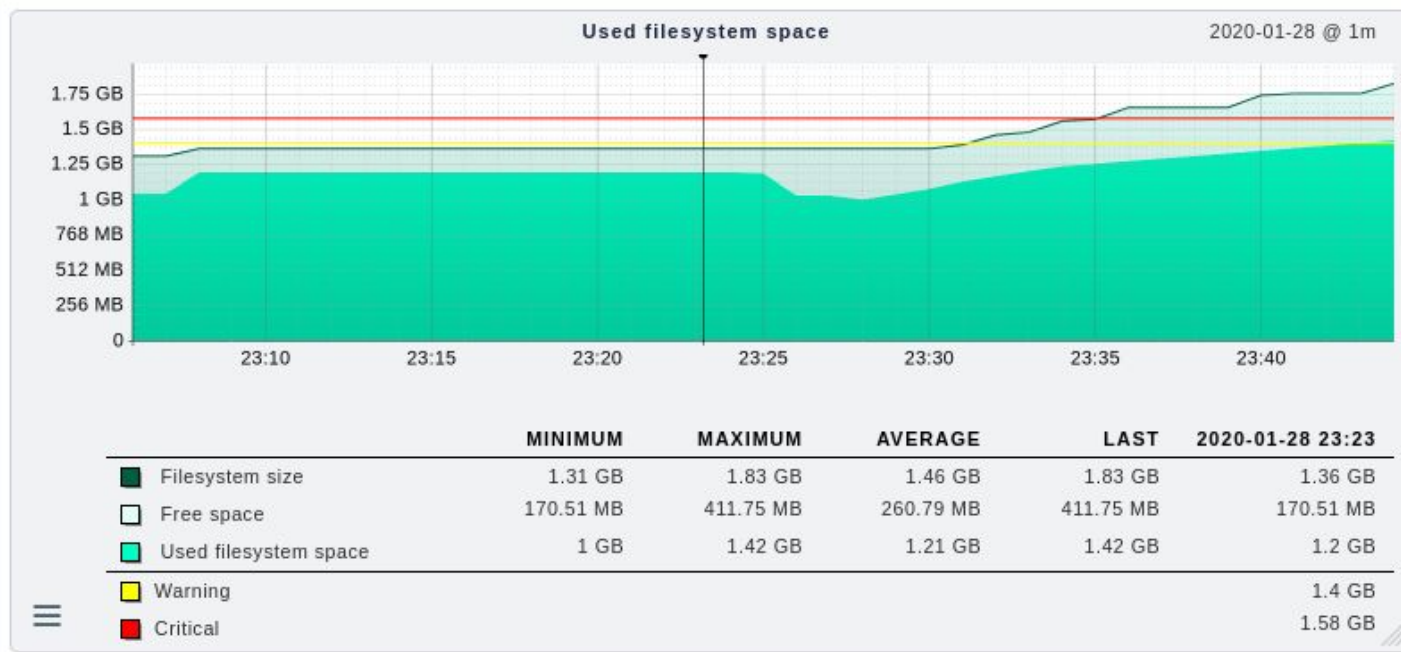
```
extend_lvm_volume:  
  salt.state:  
    - tgt: {{ data['data']['HOSTNAME'] }}  
    - sls:  
      - volume-mgmt.lvm_ext  
    - kwarg:  
      pillar:  
        volume: {{ volume }}
```

Result

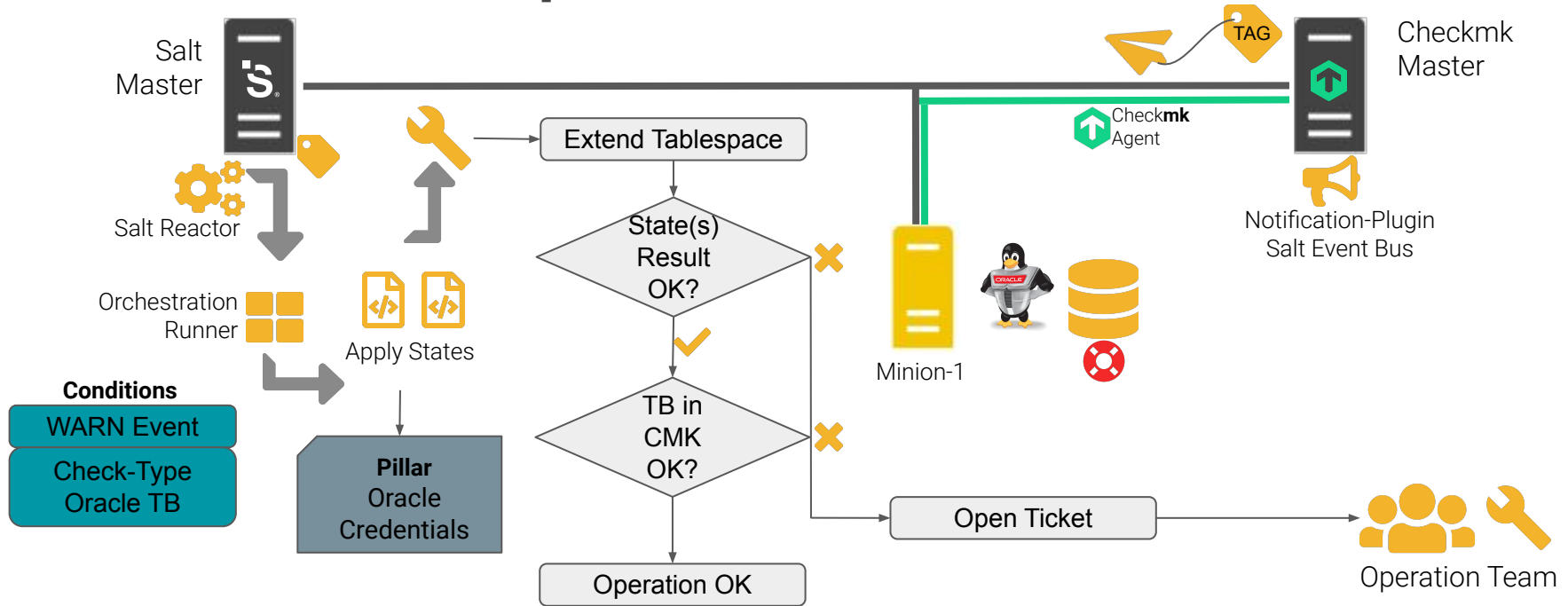
```
"salt_|-extend_lvm_volume_|-extend_lvm_volume_|-state": {
  "__id__": "extend_lvm_volume",
  "__jid__": "20200128230640495189",
  "__run_num__": 0,
  "__sls__": "orch.lvm_ext",
  "changes": {
    "out": "highstate",
    "ret": {
      "centosvm1": {
        "cmd_|-extend-lvm-volume_|-lvextend -L +100M $(mount|grep /srv/data1|awk '{print $1}')" && xfs_growfs /srv/data1|-run":
          "__id__": "extend-lvm-volume",
          "__run_num__": 0,
          "__sls__": "volume-mgmt.lvm_ext",
          "changes": {
            "pid": 13470,
            "retcode": 0,
            "stderr": "",
            "stdout": "  Size of logical volume vg00/lv1 changed from <1.27 GiB (325 extents) to <1.37 GiB (350 extents).\n
dev/mapper/vg00-lv1   isize=512    agcount=11, agsize=32000 blks\n                      =                  sectsz=512   attr=2, projid32bit=1\n                      bsize=4096   blocks=332800, inapct=25\n                      =                  sunit=0      swidth=0 blks\n                      bsize=4096   blocks=855, version=2\n                      =                  sectsz=512   sunit=0 blks, lazy-count=1\n                      from 332800 to 358400\n                      realtime=none\n\n",
            "comment": "Command \"lvextend -L +100M $(mount|grep /srv/data1|awk '{print $1}')} && xfs_growfs /srv/data1\" run",
            "duration": 260.752,
            "name": "lvextend -L +100M $(mount|grep /srv/data1|awk '{print $1}')} && xfs_growfs /srv/data1",
            "result": true,
            "start_time": "17:06:40.712631"
          },
        },
      },
    },
  },
  "comment": "States ran successfully. Updating centosvm1.",
  "duration": 621.618,
  "name": "extend_lvm_volume",
  "result": true,
  "start_time": "23:06:40.464613"
},
"outputter": "highstate",
"retcode": 0
},
"success": true,
"user": "Reactor"
```

Result

Local site site1, centosvm1, Filesystem /srv/data1



Workflow Example



Thank You

AGFA 
HealthCare
www.agfahealthcare.com


pl@philipp-lemke.de
www.philipp-lemke.de

Project salt-checkmk
 <https://github.com/tribe29/salt-checkmk>

