SaltStack

Orchestration & Application Deployment

Arnold Bechtoldt

Oberhausen, 07.11.2015

Arnold Bechtoldt

Systems Engineer @ inovex GmbH

- Platform Engineering
- System Automation & Development
- DevOps Support & Consulting
- Open Source Software Contributions

Agenda

1. Orchestration in General
2. SaltStack Orchestration Basics
3. Hands on: Demo
4. Q&A

Assets & Links

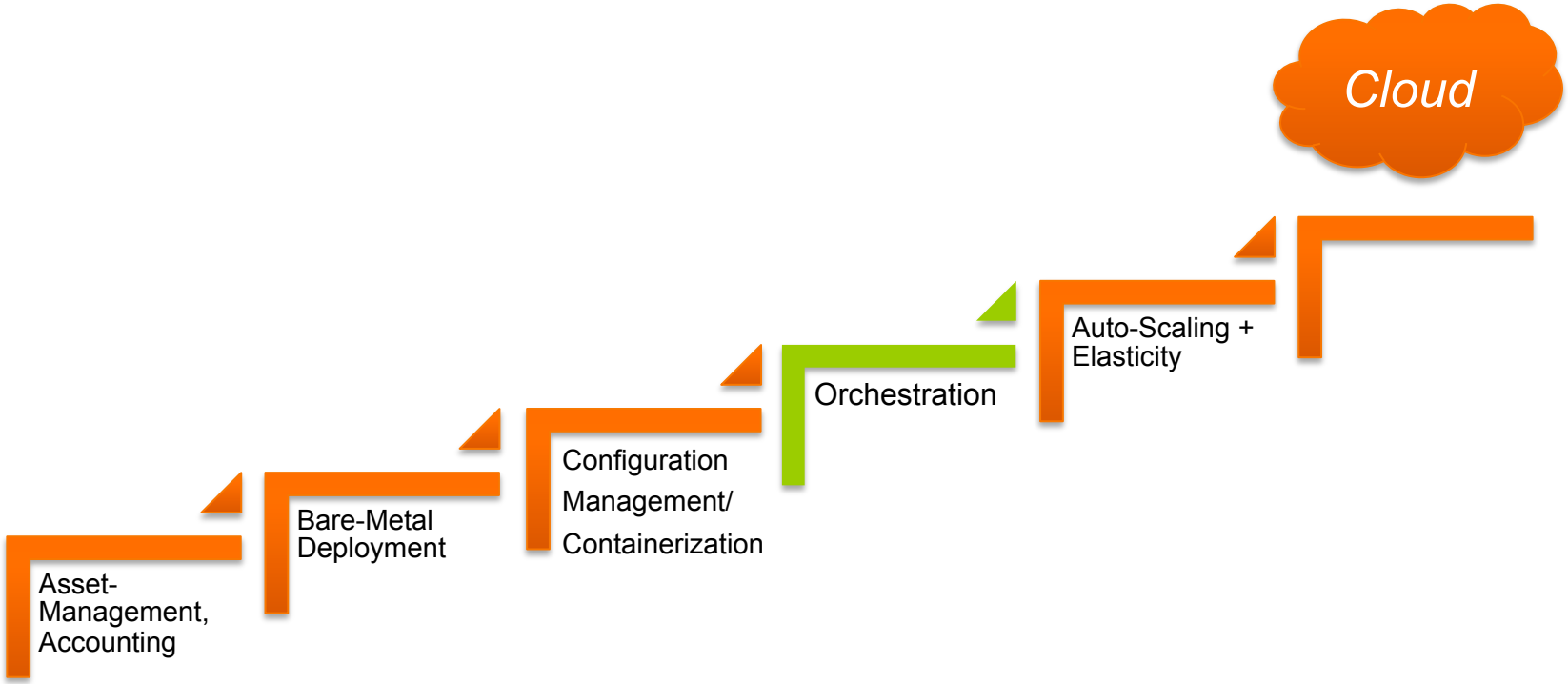
github.com/bechtoldt/talk-salt-orchestration

inovex.de

arnoldbechtoldt.com

Orchestration in General

Highway To Heaven



Who needs Orchestration?

- Lazy Admins: *„Don‘t repeat yourself“* (DRY)
- Busy Admins: *„I have a lot of other problems and [...]“*
- Small Teams: *„Ask X, but she/he is out of office until next week“*

Orchestration...

- ... uses *remote execution* to distribute system commands
- ... should be more than just executing commands (most of the tools end here)
- ... should be able to respect internal & external dependencies/relations
- ... should happen deterministic, imperative and *fast*
- ... must [RFC2119] be *easy* to understand/learn (YMMV)

A not-so-cool Example

```
#!/bin/bash -e
for node in $(mco find --np -C roles::node)
do
  echo "restarting ssh on $node"
  mco service --np sshd restart -I $node >/dev/null
  sleep 2
  mco rpc service status service=sshd -I $node -j | \
    jgrep data.status=running -s data.status >/dev/null
  echo "ssh is up on $node"
done
```

MCollective + Shell Script

For instance, if you use a single-purpose deployment tool [...] to trigger some legacy configuration management, that might be called “orchestrating X with Y”. In reality, it’s usually just replacing a non-scaling [...] server solution X with a stand-alone implementation of Y.

http://www.ansible.com/blog/orchestration-you-keep-using-that-word


SaltStack Orchestration Basics

SaltStack Orchestration

SaltStack implements the same techiniques that other tools do:

- send commands to servers
- send commands to servers that have different operating systems
- send commands only to a subset of servers
- run command A, then B, then C

Some people call this orchestration and wrap this shell-oneliner with thousands LoC.

SaltStack Orchestration

What we really want to have:

- scale accross thousands of servers
- easy configuration (less software programming)
- an interface to implement internal/external relationships (pre/post tasks)
- fully automated workflows/procedures (no manual interaction)

SaltStack Orchestration

What we really want to have:

- scale accross thousands of servers

  SOA – Message Bus Architecture

- easy configuration (less software programming)

  YAML – JSON – Python DSL

- an interface to implement internal/external relationships (pre/post tasks)

  State – Execution – Pillar – Returner – Beacon – Cloud (~ 600 modules)

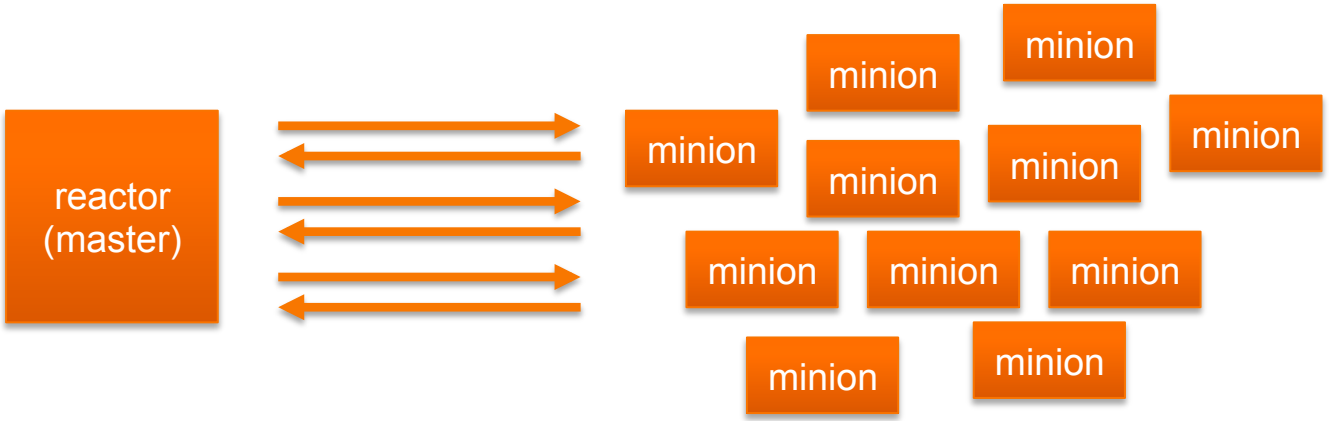
- fully automated workflows/procedures (no manual interaction)

  Event System – Reactor – Runner – APIs

SaltStack Terminology

State	A resource should look like {…}
Pillar	Database/CMDB (Files, RDBMS, NoSQL)
Execution	Execute command X on a server. Used by other Salt components.
Returner	Log store of job results (DB, Elasticsearch, Syslog, Monitoring)
Beacon	Special event triggers (inotify, load, procs, network/user activity)
Reactor	Reacts to events by triggering subsequent procedures (execution modules)
Cloud	IaaS provisioning (EC2, OpenStack, Digital Ocean, Linode, GCE, VMware)
…	

Event-Driven System Automation



salt/key	salt/job/new	node/stonith/new	update/application
salt/auth	salt/job/return	scale/out	loadbalancer/config/update
salt/minion/start	salt/presence/present	monitoring/hosts/new	firewall/config/update
salt/minion/stop	salt/presence/change	monitoring/hosts/remove	coffee/new

Event Structure

salt/job/20151104191820394966/new {

```
"_stamp": "2015-11-04T18:18:20.512126",
"arg": [
    ”orchestration.bootstrap",
    {
        "__kwarg__": true,
        "test": false
    }
],
"fun": "state.sls",
"jid": "20151104191820394966",
"minions": [
    ”mw42"
],
"tgt": ”mw42",
"tgt_type": "glob",
"user": ”root”
```

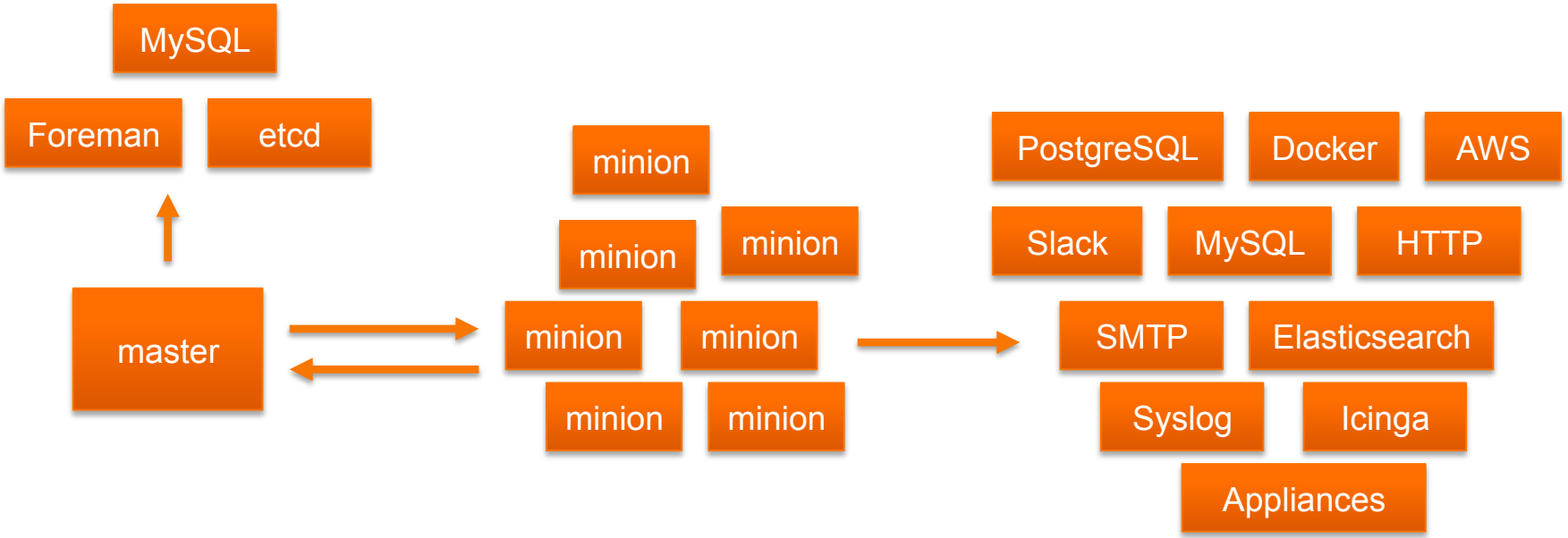
}

custom/minion/haste_server_started {

```
"_stamp": "2015-11-04T18:33:54.650568",
"cmd": "_minion_event",
"data": {
    ”custom": {
        "onchanges": [],
        "foo": "bar",
        "num": 42,
    },
    "sfun": "wait"
},
"id": "mw2",
"pretag": null,
"tag": "custom/minion/haste_server_started"
```

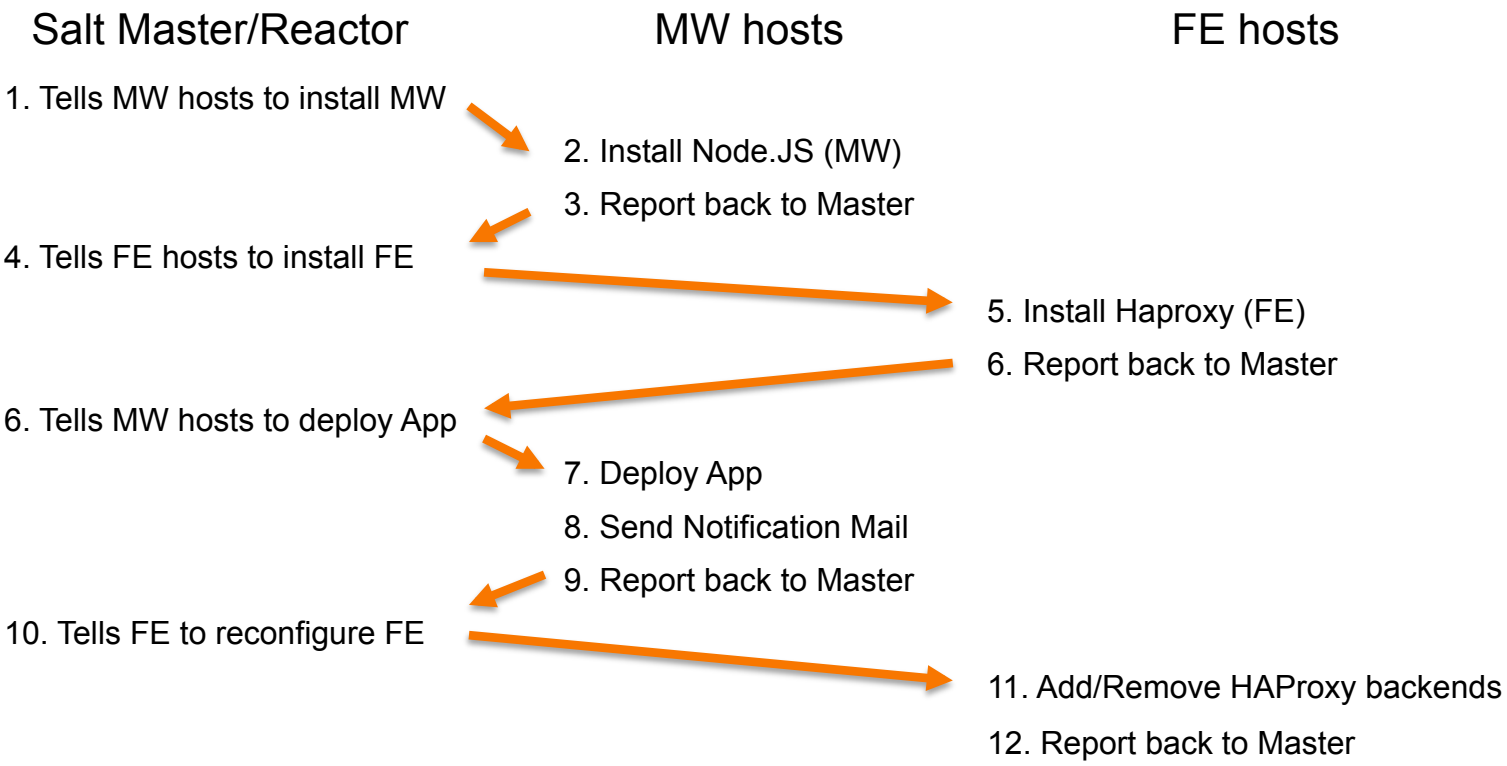
}

Event-Driven System Automation

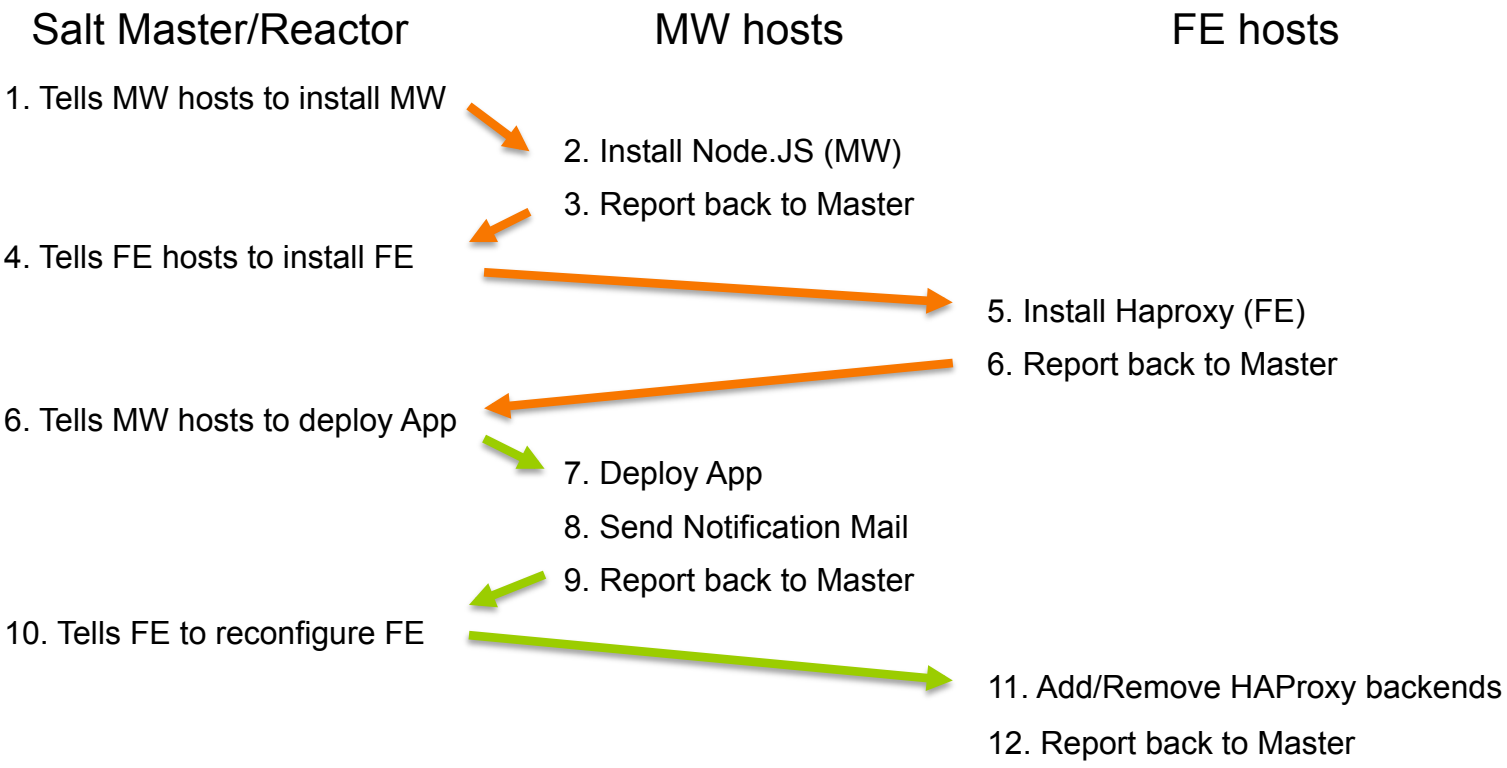


Hands on: Demo

Demo Concept



Demo Concept



Assets & Links

github.com/bechtoldt/talk-salt-orchestration

sh.arbe.io/cloud-provision

youtu.be/9MzeK4u4pkM (demo)


Q&A

Kolleginnen und Kollegen gesucht!

- Application Development
- Business Development
- Consulting
- Data Management & Analytics
- IT Engineering & Operations

- Hamburg
- Karlsruhe
- Köln
- München
- Pforzheim

inovex.de/jobs

Thank You!

Arnold Bechtoldt
IT Engineering & Operations

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe - Germany

arnold.bechtoldt@inovex.de

inovex.de

+ArnoldBechtoldtGER

github.com/bechtoldt

arbe.io

youtube.com/
inovexGmbH