System management with RPM and YADT
A Solution for Data Centers

Brussels | 2012-02-05 | Ralph Angenendt
Application Manager
So what is it?

- RPM
  - Well known packaging format
  - Easy to use (and package)
  - Built-in content verification
  - Complete toolchain
So what is it?

- YADT
  - An Augmented Deployment Tool
  - Central management of dependencies between
    - Services
    - Systems
    - Software Packages
RPM, huh?

Sure. Everything is packaged as an RPM

→ Our system software (RHEL – 100% RPM)
→ Software from the outside (think EPEL)
→ Our Applications
  → We wish
  → But we're getting there
But config?

- It comes in files
- RPM is good at handling files
- There are tools to get RPMs on a machine
- RPM can verify package contents
- Updates are easy
So you build RPMs for every machine?

Um. No.
So you build RPMs for every machine?

Well, sort of.
So you build RPMs for every machine?

We let machines do it.
„Config Subversion“

- All Configuration is kept in an SVN repository
  - Hierarchical
  - Supports a „Data Center“ layout
  - Is easy to understand
  - Typical unixy filesystem layout
Config „subversion“

- Goes from general to special
- On-Commit
  - RPM building
  - YUM repository generation
- Also works with dpkg and apt
  - If you write the code to support it
In general it looks like this

Overwrites

all/
loc/
type/
loctype/
host/
In general it looks like this

```
all/
  etc/
  data/
  VARIABLES/
loc/
type/
loctype/
host/
```
In general it looks like this

```
In general it looks like this

all/
loc/
tuv/
ber/
ham/

Overwrites

etc/
data/
VARIABLES/
type/
loctype/
host/
```
In general it looks like this

```
all/
loc/
type/
  web/
  app/
  etc/
  data/
  VARIABLES/
loctype/
host/
```

Overwrites
In general it looks like this

- all/
- loc/
- type/
- loctype/
- berweb/
- berapp/
- etc/
- data/
- VARIABLES/
- host/

Overwrites
In general it looks like this

```
all/
loc/
type/
loctype/
host/
  berweb01/
  berweb02/
  etc/
data/
  VARIABLES/
```
VARIABLES?

- VARIABLES/ contains – well – variables
  - Many hosts have a similar configuration
  - Best to configure that in a general way
    - All hosts use a proxy
    - Proxies in tuv, ber and in ham are different
Variables

all/etc/proxy.conf:
    [...]  
    proxy_port=3128
    proxy_host=@@@PROXY_HOST@@@

loc/tuv/VARIABLES/PROXY_HOST:
    tuvprx.example.com

loc/ber/VARIABLES/PROXY_HOST:
    berprx.example.com

loc/ham/VARIABLE/PROXY_HOST:
    hamprx.example.com
More specials

- There are two special Variables
  - **RPM_PROVIDES**
    - config-hostname (e.g. config-berweb01)
  - **RPM_REQUIRES**
    - tomcat, httpd, java-application

- **RPM_PROVIDES** is required during kickstart
- Content of **RPM_REQUIRES** pulls in all other needed RPMs for the host
Putting it all together

config-rpm-maker

substitutes
VARIABLES

builds
RPMs

creates
YUM-Repo
Putting it all together

- **yadt-config-rpm-maker**
  - Works as a post-commit hook in subversion
  - Written in python
  - Creates packages in parallel
  - Automatically determines which packages have to be rebuilt
  - Rebuilds the minimal set needed
  - Is open source (GPL)
  - Available from https://code.google.com/p/yadt/
Caveats

- RPM dislikes a few things
  - Mainly two packages owning the same file
  - Not every software has a config.d/
  - „Generic“ config mostly not usable
  - Installation tends to break, then
Caveats

Solution

- Write wrapper packages
- Those overwrite config via `%post`
  - Config now includes config.d/ (if possible)
- Write your own config.d/ structure
- Overwrite original config by piecing things from config.d/ together
Summary

- Complete config is in one package
- Config pulls in „complete machine“
- Tool chain allows easy verification
- Tool chain is well known
- Package format is
  - Well known
  - Rather easy (from an „RPM person“ view)
- Config is precalculated before copying
Nexus Yum Plugin available from https://code.google.com/p/nexus-yum-plugin/
YADT

- Knows your Data Center
  - Allows you to model your DC
  - YAML-based description of
    - Services
    - Applications
    - Hosts
- Knows about dependencies between
  - Packages
  - Services
  - Systems
Configuration

Target definition in file „target“:

```
name: probau
log-dir: logs

hosts:
- hambau*.example.com
- berbau*.exampe.com
```
Service definition in file „yadt.services“:

- service1:
  needs_services: [service2]

- service2:
  needs_services: [service3]

- service3:
Configuration

Notations:

- `service://hostname/servicename`
- `host://hostname/`
- `artefact://hostname/packagename/version`
- `yadt status service://hostname1/httpd`
- `yadt ignore host://{host2|host33}`
- `yadt lock -m host://hostname3`
- `yadt updateartefact artefact:// [host1..host15]/yadt-client`
YADT - the smallest unit

- **tomcat**: Depends on **tomcat config**
  - Restarts on update

- **yadt.services**:
  - **tomcat**:
    - **Target**:
      - hosts:
        - foo.example.com
YADT - simple dependencies

- httpd:
  - needs_services: [tomcat]

- tomcat:

```
yadt.services:
- httpd:
  needs_services: [tomcat]
- tomcat:
```
YADT - adding external services

yadt.services:

- loadbalancer:
  needs_services: [httpd]
  class: LoadbalancerService
  loadbalancer_clusters: [pro-fe]
  pool: test
  port: 80
  status_max_tries: 2

- httpd:
  needs_services: [tomcat]

- tomcat:
External services

- YADT has a service layer
  - Python module
  - Can also execute scripts
  - Loadbalancer:
    - Uses the F5 Big IP python api
    - Can disable/enable hosts
  - We also use it for making Nagios go quiet
  - Not yet open source
    - Needs to be generalized
YADT - adding services on other systems

```
yadt.services:
- loadbalancer:
  needs_services: [httpd]
  class: LoadbalancerService
  loadbalancer_clusters: [pro-fe]
  pool: test
  [...]
- tomcat:
  needs_services: [app]

yadt.services:
- app:
```

Diagram:
- LB
  - httpd
    - config
  - tomcat
    - config
- app
  - config
YADT - more complex modeling

httpd

httpd config

tomcat

tomcat config

app

app config

target: hosts:
- host[01..02]

Slide 35 | System management with RPM & YADT | Ralph Angenendt
YADT - chunks and wave deployment

1. 
2. 
3.
Interface
Conclusion

- RPM configuration works astonishingly well
- Though needs work around caveats
- Easy to maintain (for everyone, just change config)
- Distribution via yum repositories
- One RPM pulls up a complete machine
- Need to „resetup“?
  - Remove config-rpm
  - Reinstall config-rpm
Conclusion

- YADT is a work in progress – but it works reliably
  - For many machines, it can get slow
    - Especially when nagios / loadbalancer are included
  - Services layer not yet open sourced
  - Easy configuration
  - Needs package based distribution system
  - We use it on a daily basis
Outlook

- New yadt-shell (the interface) on the way
- Work has started to parallelize yadt
  - Against slowness
  - Let's you do „server, rack, datacenter“ scenario
  - You can determine fault tolerance
- Would be nice to have a working Demo system =:(
The End (finally!)

YADT
https://code.google.com/p/yadt/

Yadt-rpm-config-maker
https://code.google.com/p/yadt/

Nexus YUM plugin
https://code.google.com/p/nexus-yum-plugin/
Thank you very much! Please contact me for further questions and discussions.

Kontakt:
Immobilien Scout GmbH
Andreasstraße 10
10243 Berlin

Fon: +49 30 243 01-1036
Email: ralph.angenendt@immobilienscout24.de
URL: www.immobilienscout24.de