Chef@GSI revisited

Christopher Huhn, Dennis Klein
GSI Darmstadt
C.Huhn@gsi.de, D.Klein@gsi.de

Hepix Spring 2016, Desy Zeuthen

Hepix Vancouver 2011

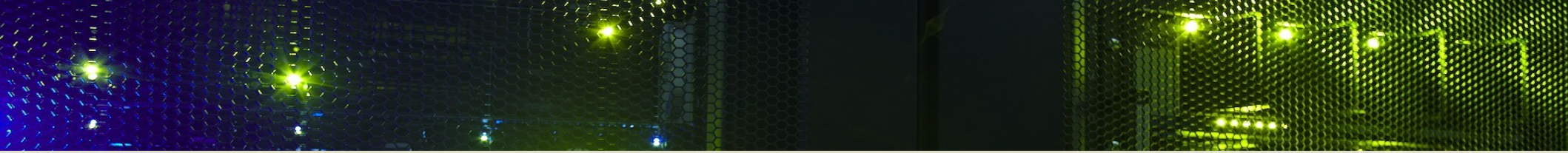
„Evolution of Configuration Management at GSI“

Status 2011

- All Linux nodes managed with Cfengine 2
  - Admin team: **3** + 2
  - Only basic setup, service config out-of-band (complex shell scripts)
- Scalability? Future requirements?
  - number of nodes
  - number of services
  - **overall complexity** of the infrastructure
  - **number of co-admins**
- Devops! Chef testbed on the then new batchfarm (Gridengine)

Wishlist 2011

- Ad-hoc (push) operations
- Extensible configuration language, API, abstraction level higher than 'editfiles'
- Completely create configuration for a complex service inside the CMS
  - Using cluster inventory information
- Automatically assign roles to emerging nodes
  - By hostname, subnet, static assignment
- Admin roles and multi-tenancy, partitioning
  - No global root for every admin
- Software engineering magic
  - Knowledge Management
    - Why has node X role Y?
    - Where does feature X come from?
  - Automated Documentation
  - Automatic quality assurance and tests:
    - Coverage, Unit tests, Integration tests, Continuous Integration
  - API, Interfaces with other systems (Issue tracker, Hardware DB, etc.)



Chef status 2016

- Chef is serious business
  - In use at Facebook, Yahoo, Etsy, …
  - (Client) runs on:
    - Unixoids (Linux, BSD, MacOS, AIX, …)
    - **Windows**
    - Cisco, Juniper, Arista, Mellanox
- Old and new competitors:
  - Puppet
  - Cfengine
  - Saltstack
  - Ansible
  - …
- Different but same?

Chef@GSI status

- Chef well established at GSI
  - 1500 nodes
  - 300 roles
  - 150 cookbooks
- **10 + 6 co-workers**
- Complex services completely built from scratch via cookbooks
  - No complex shell scripts, no ad-hoc administration
  - Configured from attributes and data bags

Chef@GSI status

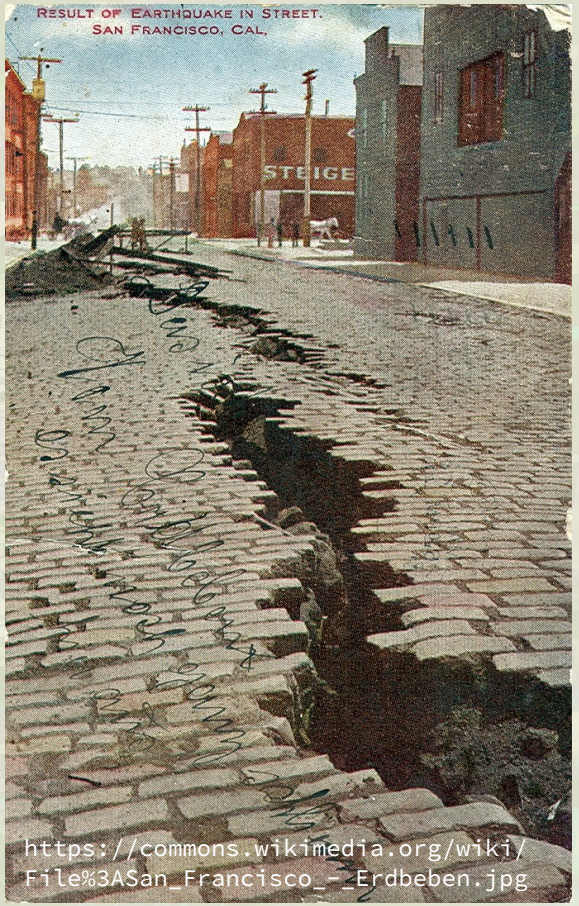
- Chef well established at GSI
  - 1500 nodes
  - 300 roles
  - 150 cookbooks
- **10 + 6 co-workers**
- Complex services completely built from scratch via cookbooks
  - No complex shell scripts, no ad-hoc administration
  - Configured from attributes and data bags

But also still 300 Cfengine nodes ?!

Migration

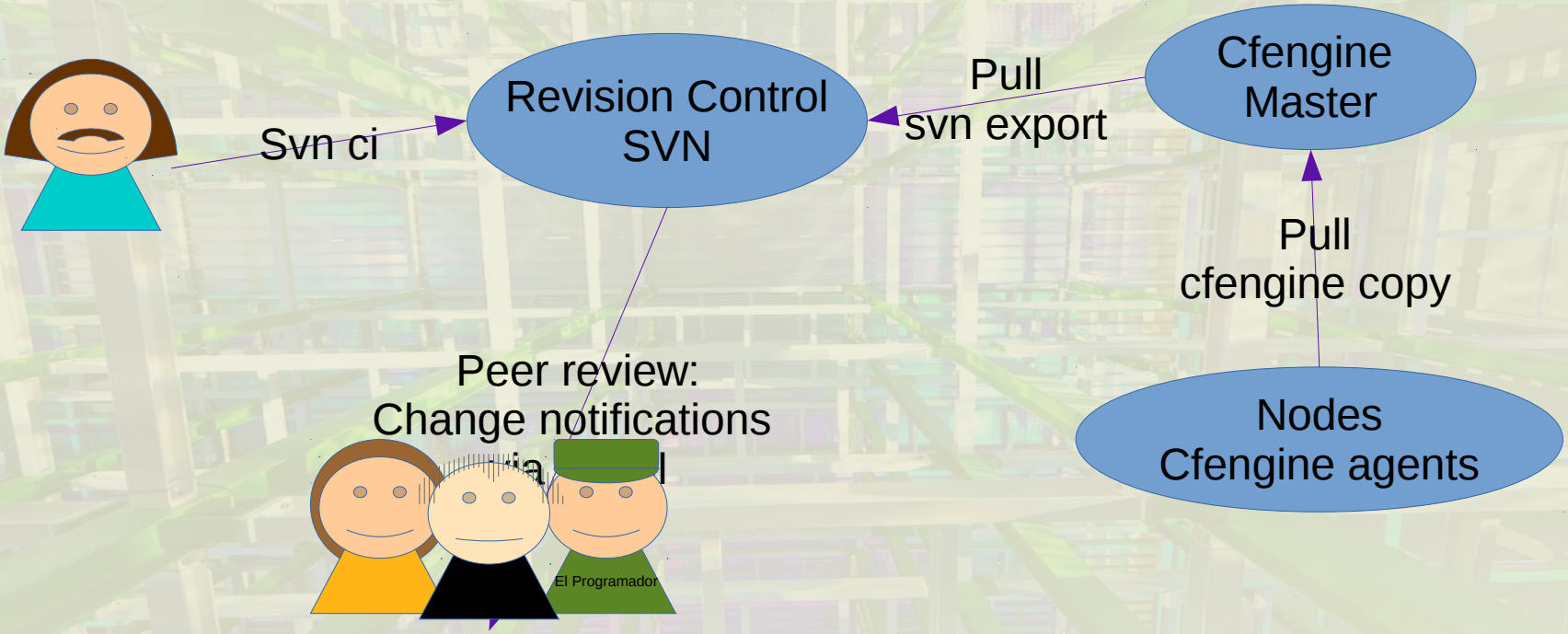
- Cfengine → Chef was also SVN → Git
  - No sustainable central git service at GSI
    - Gitorious + gitolite/gitweb + gitlab
  - git more complicated, laboring for small changes
  - ~~git advantages not visible in daily work~~
    SVN users still don't understand distributed VCS
- Higher abstraction level needs more coordination

- Vocabulary : googling „Chef“ etc has lots of unwanted results
  - Add “opscode” to the search terms



SVN/Cfengine workflow

`svn up` → edit → `svn commit` → check results



Git/Chef workflow

- Cookbook versioning vs. git tags

  `git pull -r`

  → increase CB version in **`metadata.rb`**

  → edit, test?

  `git commit -m …`
  `git push`
  `knife cookbook upload`

  → deploy, check results

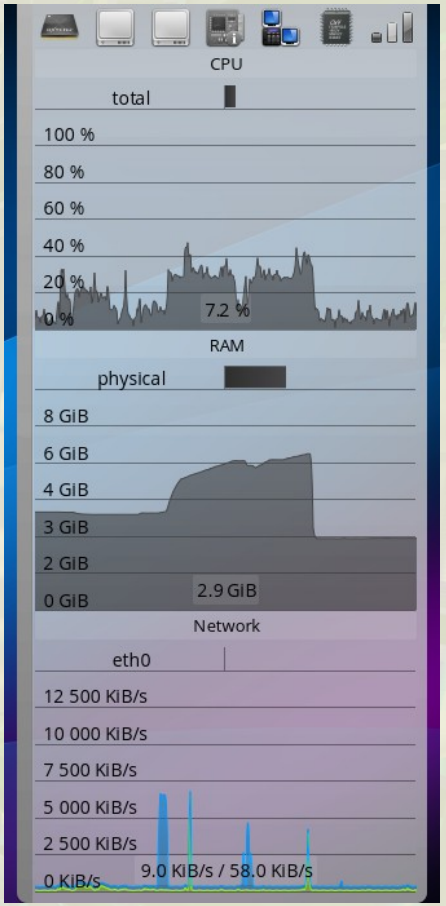
  `git tag`
  `git push –tags`

```
git tag -a `sed -ne
      "s/^version.*[\"\']\([0-9.]\+\[\"\']/\1/p"
       metadata.rb` \
   -m "`git log --format=%s -n 1`"
```

- How to prevent:
  - conflicting edits?
  - Uploads from dirty working copies
  - Git push –force
- Needs further automation!

Inventory

- Ohai plugins
- `knife exec` reporting
  - `nodes.all` may be fatal
- Interactions:
  - Example: Wiki, monitoring, sanity checks
- Node information for config rendering
  - Used for (old) Gridengine cluster config
  - Not used for (new) Slurm config
    - Configuration managed manually and distributed via NFS



Chef Lessons

- Ignore node objects:
  - Don't assigned attributes to nodes manually
    - Only `run_list` + automatic (ohai)
  - Not in revision control
  - Use **roles**
    - Rule of thumb: exactly one role per node?
- Community cookbooks
  - Your mileage may vary
  - Server and (trivial) Client in config in the same cookbook
  - Often focused on installation tasks
  - Multitude of cookbooks for the same service
    - Focused on different facets
  - Apache cookbook and web_app resource

sys cookbook

- GSI's workhorse
- https://github.com/GSI-HPC/sys-chef-cookbook
- Combined client config for various system services in a single cookbook
  - accounts, apt, autofs, cgroups, chef, env, ferm, ipmi, krb5, ldap, mail, network, pam, rsyslog, snmp, ssh, sudo, systemd, time, ...
- Controlled by defining attributes
  - Otherwise does nothing
  - Does not break stuff by accident

Distributing files and software

- Deploy resource etc
  - Where did that file come from?
  - curl | sudo bash ???
- Cleanup of garbage/old versions?
- cookbook_files splatters files over cookbooks
- No bidirectional file transfer (node → chefserver, node → node)

Distributing files and software

- Alternative: **build Debian packages**
  - Features for free:
    - Where did that file come from? `dpkg -S file`
    - Cleanup: `apt-get remove (--purge)`
    - Controlled origin secured by crypto, reproducible install
    - file checksums for alteration checks:
      - `debsums`
  - Used for proprietary software, admin scripts, document templates, ...

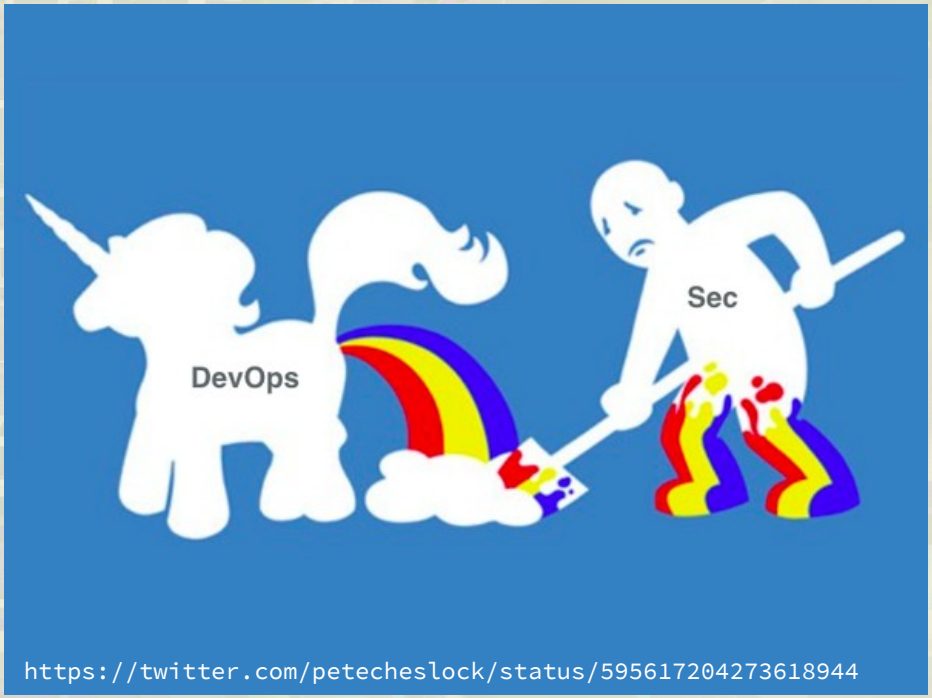
Chef annoyances

- Robustness
  - Chef-client is fail-fast (== you shoot yourself in the foot)
- Attribute hell
  - Attribute deep merging sometimes has “interesting” results
  - Recipe evaluation/execution order too
  - Cumbersome to debug

http://fav.me/d2xcg1w

Chef Security

- Unsecured private keys on the file system
  - SSH agent plugin for knife (→ D.Klein)
- Nodes have access to all other nodes' attributes (in the same organization)
- Secret plugin in sys cookbook (end-to-end encryption between nodes/clients, → B.Neuburger)
- Updatability of Omnibus-Chef-Server packages?
  - Contains private Init (runit), Perl, Python, Ruby, JRE, OpenSSL (!), PostgreSQL server, Erlang, …
  - > 74k files, > 450 MB
- Trustworthiness of supermarket.chef.io/github/rubygems ... ?
  - Cf .http://blog.rubygems.org/2016/04/06/gem-replacement-vulnerability-and-mitigation.html



https://twitter.com/petecheslock/status/595617204273618944

ACLs, roles, multi-tenancy

- Yes but no?
- Chef Server ACL system still object to change
- ACL management difficult to organize
- Better control via Git and moderated uploads?
- Alice T2 service nodes:
  - HPC: system cookbooks
  - Grid admins: service cookbooks
  - Shared role definitions
- central configuration management **service**

Testing/CI/Quality Assurance

- Lint
  - `rubocop`
  - `foodcritic`
- Testsuites
  - `rspec`, `chefspec`
  - `kitchen.ci`, `serverspec`
- Documentation generation
  - Chef plugin for `yard`

- Writing test code is substantial work
  - `rspec` is a mighty beast
  - Test code ratio 1:1 to 1:5 recommended in TDD!

House keeping

- Configuration space pollution
- Cleanup of unused/stale objects
- Coverage check?
  - Not only desirable for unit tests
- Code duplication check?

https://commons.wikimedia.org/wiki/File%3A(Six_sailors_sweeping_sidewalks_at_Naval_Training_Center%2C_San_Diego%2C_California.)_-_NARA_-_295581.jpg

~~Wishlist 2011~~ Status 2016

- **Extensible configuration language, API**
- **Create a configuration for a central service from inventory information**
- **Automatically assign roles to emerging nodes**
  - E.g. via custom cron'd knife exec Script
- **Perform ad-hoc state changes**
  - `knife ssh ...`
- **Admin roles** and **multi-tenancy, partitioning**
- **Knowledge Management and Quality Assurance**
  - ***Automatic quality assurance and tests:***
    - ***Unit test, integration test***
      Coverage checks,
      Continuous Integration
  - ***Automated documentation generation***
  - **Interface with other systems (Issue tracker, Hardware DB, etc.)**

There is a theory ...

"... which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened."

Douglas Adams