



Quattor Update

Aquilon & the Quattor Community

Ian Collier, James Adams

STFC RAL Tier 1

ian.collier@stfc.ac.uk

HEPiX, Annecy, 22nd May 2014

Context

- RAL Tier 1 adopted Quattor in 2009
 - Required lifecycle management
 - Needed configuration management
 - Quattor combines 2 in mature toolset with active community
 - Appeared the best choice for us at that point.
 - Has improved almost beyond recognition since then
- Quattor has transformed our ability to effectively manage our site as scale increases.
- We like the consistency verification before deployment
- We benefit significantly from the collaboration - in proportion to what we put in.
- We are still feeling effects of some decisions made early on.



Aquilon

- Some History
- Aquilon
- Example
- Conclusion



How did we get here?

- 1st Generation — CDB
 - Pan code stored in CVS
 - Basic deployment workflow tooling
 - Global locking quickly caused scaling problems
 - Abandoned by the community, still used by CERN for legacy systems
- 2nd Generation — SCDB
 - Pan code stored in Subversion
 - Tagged deployment workflow based on ant and SVN repository hooks
 - Admins can check out, develop, check consistency of changes on local system
 - Global deploys cause scaling pain



(S)CDB

CDB & SCDB have similar principles

- Code → Compile → Commit → Deploy → Repeat

Both are essentially an environment for writing Pan

- Some layout guidelines
- Lack of enforced rules for structure of configuration leads to fragmentation, even within sites

Inputting lots of systems gets boring quickly

- Sites have built custom inventory databases
- Scripting only goes so far

But powerful enough to be good enough!



Motivation

2007: Morgan Stanley joined community

- Outgrown existing system
- Planning to deploy 20,000+ hosts
- (S)CDB wouldn't scale to this

Requirements

- Global builds not mandatory
- Large numbers of admins with different privileges
 - Routine operations as documented commands
 - Make changes without editing Pan code
- Ability to branch configuration for development and testing
 - Test changes without committing to a VCS
 - Deploy hosts from branches
- Provide structure for configuration



Something entirely new required

Aquilon

- 3rd generation configuration management database
- Builds upon concepts from previous CMDBs
 - But still a paradigm shift
 - Incorporates inventory
 - Provides structure
- Development mostly undertaken by Morgan Stanley



First impressions

Git as VCS for Pan code

- Finally! Proper branching and merging

Broker daemon running system

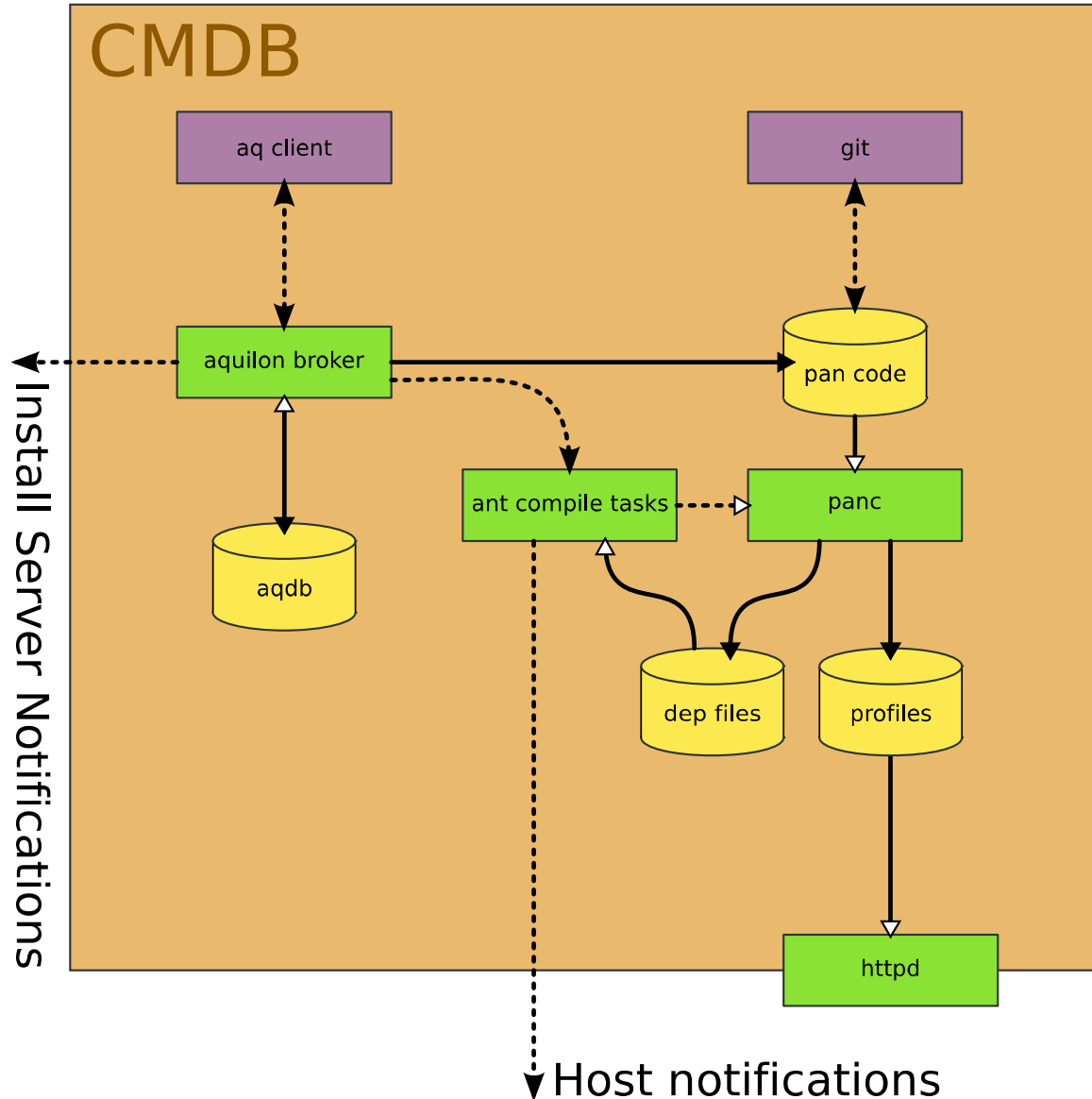
- Owns parts of configuration
- Role based permissions

CLI for interaction with broker

- Make configuration changes
- Request git branches



Architecture



Broker

Source of **all** power

- Provides workflow engine
- Writes Pan code for objects and relationships
- Owns blessed Git repository
- Users request branches and work on clones (sandboxes)
- Allows hosts to be built from sandboxes

Pure Python

SQLAlchemy as object-relational mapper, objects in RDBMS*

REST-ish API for client

- /host/www.example.com
- /find/host?personality=webserver

*Many will work, but only PostgreSQL and Oracle are supported.



Science & Technology
Facilities Council

Sandboxes

Production configuration in the prod domain
Branched into sandboxes for development

```
aq add sandbox  
--sandbox new-awesomeness
```

Creates branch in the broker owned repository
Auto-cloned to user's home directory by client



Sandboxes

Published for review by others

```
aq publish --sandbox new-awesomeness
```

Deployed (merged) back into prod when ready

```
aq deploy  
  --source ianc/new-awesomeness  
  --target prod
```



Objects

Aquilon provides objects for modelling inventory, high level configuration and the relationships between them.

- Inventory
 - Location** Buildings, Rooms, Racks, Desks...
 - Hardware** Machines, NICs, Drives, CPUs...
 - Network** Switches, Routers, Subnets, Gateways...
- Configuration
 - Feature** Re-usable block of Pan code configuring something
 - Personality** A collection of Features
 - Host** Machine, FQDN, IP, Personality & OS

Each object has a corresponding `add`, `del`, and `update` command.



Services and Mappings

Services

- Model the concept of a service
- Particular instances of services
- Track servers and clients

Service maps

- Rules defining which hosts use which instance of which service
- Rules can be defined based on:
 - Organisation
 - Physical Location
 - Network IP address



Example

You have two clusters *arrow* and *angel*:

- Both have different types of compute node.
- Each has an NFS server based on the same personality.
- Each is in a separate subnet.



Define Services

Define a nfs service with an instance for each cluster.

```
aq add service
    --service nfs
    --instance arrow
aq add service
    --service nfs
    --instance angel
```



Bind Servers

Bind a server to each nfs instance.

```
aq bind server  
  --service nfs  
  --instance arrow  
  --hostname snake.example.com
```

```
aq bind server  
  --service nfs  
  --instance angel  
  --hostname clockwork.example.com
```



Add Requirements

Add nfs requirement to both compute node personalities.

```
aq add required service  
    --service nfs  
    --archetype linux  
    --personality gpu-cluster-node
```

```
aq add required service  
    --service nfs  
    --archetype linux  
    --personality phi-cluster-node
```



Map Services

Map service nfs based on network subnet.

```
aq map service
    --service cluster-nfs
    --instance arrow
    --networkip 172.16.7.0
```

```
aq map service
    --service cluster-nfs
    --instance angel
    --networkip 172.16.12.0
```



RAL Experience

First site to try Aquilon outside Morgan Stanley

- Lots of work required to generalise
- Began 3-4 years ago

Just began running production services

- 250 hosts total
- Alongside SCDB

Using SCDB feels painful by comparison

- Full migration under way (all WNs by end of summer)



Aquilon Summary

- The third generation CMDB for Quattor
- Integrated inventory information
- Provides a framework for configuration code
- Broker is source of ultimate power
- Just beginning to realise possibilities for automation



Community

- Highlights
- Github
- Releases & packaging
- YUM
- New components
- Workshops etc.



Highlights in last year

- Yum based package management
 - (almost) all sites migrating
- Regular releases
 - integration/testing infrastructure
- Aquilon entering production outside Morgan Stanley



Github

- Everything now on GitHub (previously mostly but not exclusively sourceforge)
- Has transformed collaboration
 - much more effective
 - Perhaps the most significant change in last 2 years
- Allowing more coherent overview of toolset
 - More easily able to see obsolete/deprecated components
 - 32 unused components removed just this week!
- Better review and scrutiny of contributions
- Workflows familiar to git users



Release & packaging

- Regular releases
 - Every two-three months
 - Release manager currently at RAL (James Adams)
 - Release includes components and deployment server
 - Core template library about to be integrated (next release)
 - Solaris packages not integrated yet
- Packaging automated using Jenkins
 - Jenkins also tests pull requests
- Process was just beginning a year ago
 - Significant effort to establish
 - Running smoothly & paying dividends



YUM Package Management

- Management of packages & dependencies has been the single real 'pain point' with Quattor.
- YUM no fully supported in most used, and latest releases
 - YUM has matured – usable without losing rollback.
 - Most sites migrating – some complete
 - Resulting in accelerating effort on rest of toolset as less time sunk into package management
 - Latest middleware templates (EMI-3) only support yum – significantly less work
 - Profiles are smaller and compile faster
- Backward compatible – can apply and then remove now redundant package specifications afterwards.



New components

ncm-puppet

- Allows puppet modules to be driven by quattor using site information stored in quattor CMDB

Why?

- Would like to benefit from wider community configuration efforts where it makes sense
- Do not want to duplicate effort unnecessarily
- Prototype working with DPM modules



New components

ncm-metaconfig

- Writes structured configuration file formats
- Allows definition of schema ‘on the fly’ without needing to write or package and distribute perl components
 - Maintains configuration validation before deployment
- Can replace the (many) components that just write configuration files
- Project underway to replace simpler components



Workshops & Community

- Community is small but very engaged.
 - Still around 50 sites
 - About 15 regular contributors/reviewers on github in last month (based on notifications)
- Several active mailing lists
 - Quattor, quattor-devel, quattor-grid(middleware components), quattor-aquilon
- Short weekly developer standup meeting, adhoc meetings as required. (Ugent at RAL this week)
- Twice yearly workshops
 - 17th Quattor Workshop in Amsterdam in March



Summary

- Quattor toolset is in very good shape
- Yum package management
- Regular release process
- Next generation Aquilon configuration database & broker entering wider production
- Github facilitating stronger collaboration
- Active/engaged community
- (Significate) Investment in processes & tools is repayed by more time spent on added-value development by developers
- Continues to meet our requirements well



Questions?



Science & Technology
Facilities Council