
CERN

Scaling Agile Infrastructure, Development and Change Management

Ben Jones, HEPiX Annecy 2014

When the project no longer fits into one meeting room, it's not just the infrastructure that has to scale.

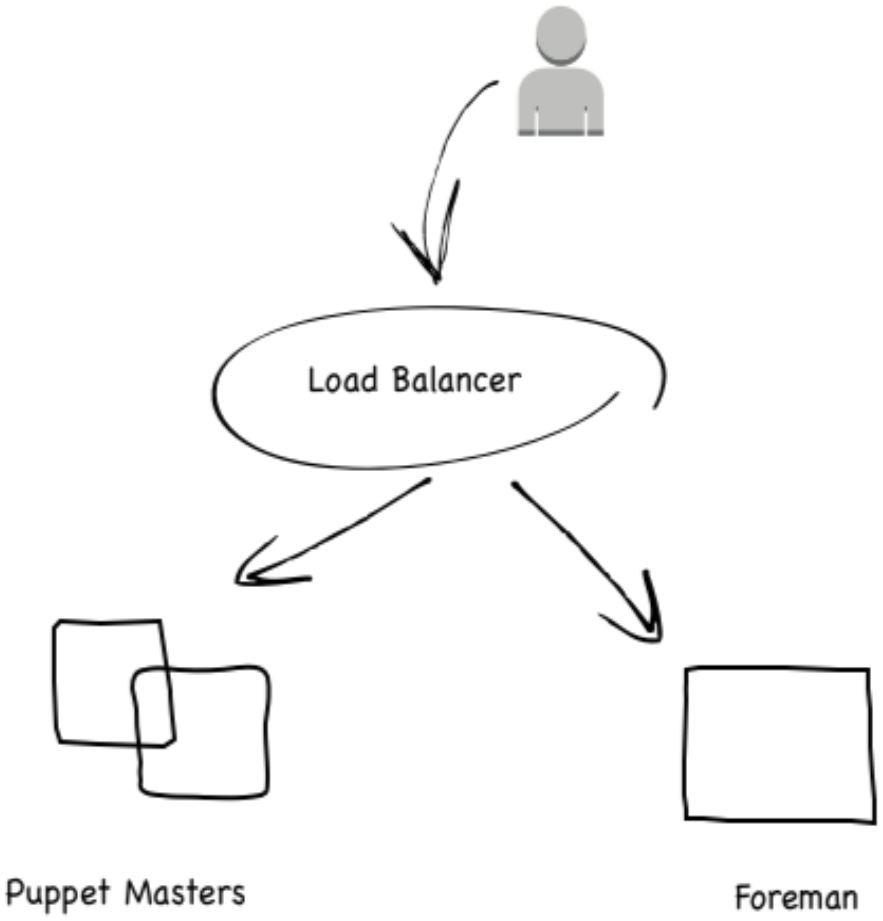
Scaling AI

- Infrastructure: getting to 9K nodes (++)
- Development: multiple groups making changes, “operational” vs “feature” changes
- Process: how to manage different requirements for the speed of change.
  - Move fast and don’t break things

Puppet choke points

- Compilations on masters
  - at scale, time to compile isn't just convenience for client, it's capacity of masters
- Submissions to puppetdb
  - replace_facts, replace_catalog, store_report
- ENC (foreman)

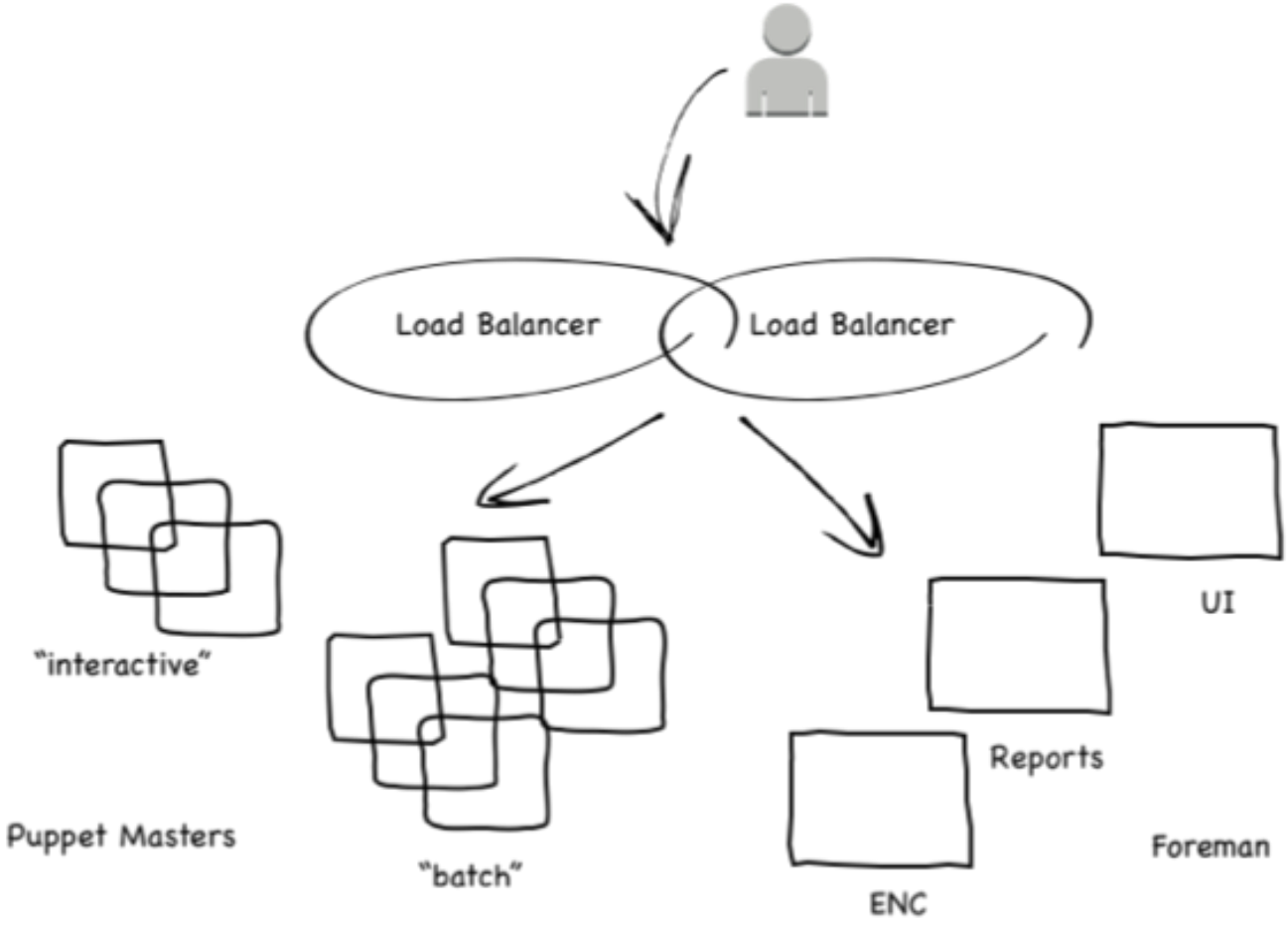
Simple Puppet Infrastructure



Problems with original infra

- Spikes in puppet compilation times make for unhappy users
  - Most automatic puppet runs do nothing, whilst people manually running puppet expect something to happen, and quickly
- Large foreman reports could overload nodes, impacting UI or ENC

Puppet Infrastructure split by traffic type

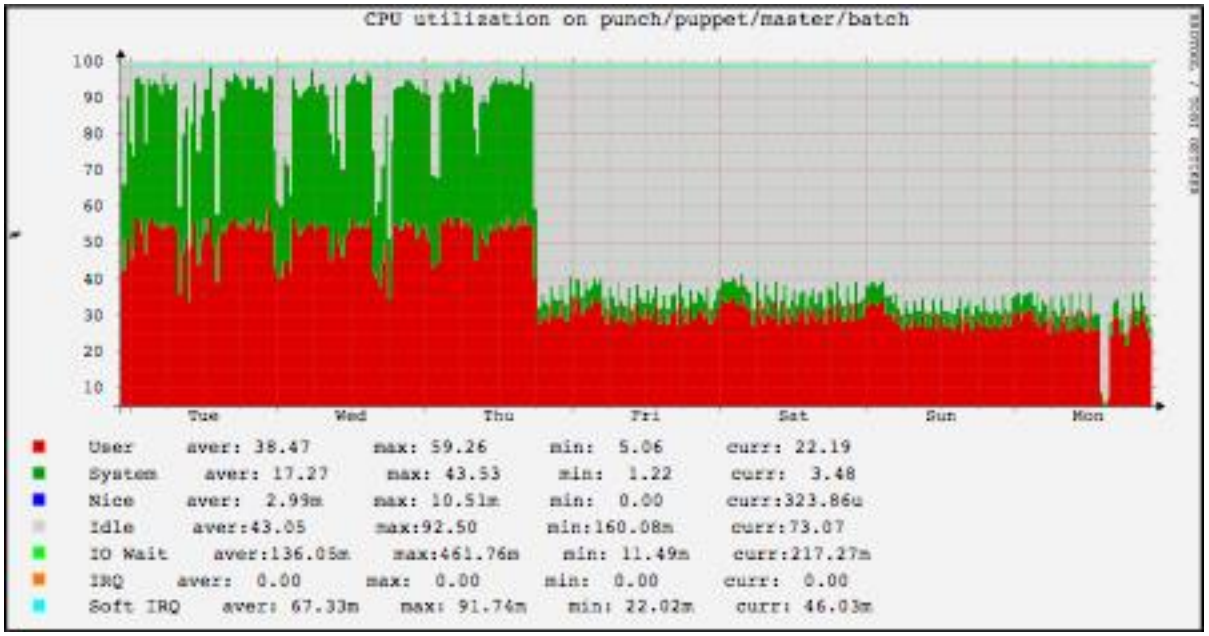


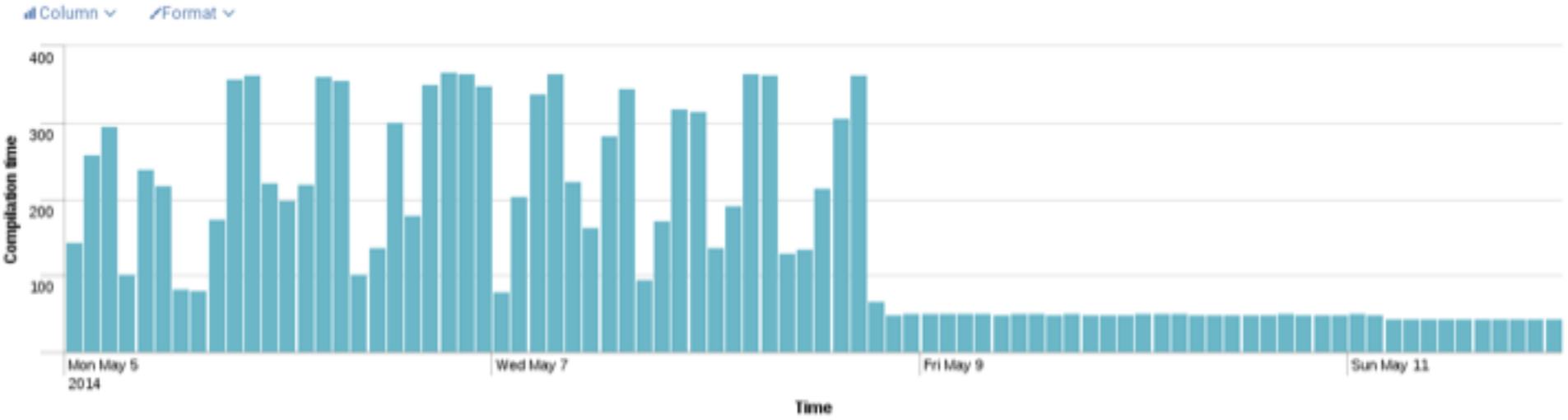
PuppetDB catalog duplication

- Every puppet run the catalog is stored in puppetdb
- In theory, should check if something has changed before storing new – wasn't working
- Catalog duplication was often < 5%
- puppetdb 1.6 fixed performance issues with catalog duplication and fact submission

Puppet file stats

- Shared NFS for manifests of multiple puppet masters
- Puppet masters were very busy, with timeouts at peak
- strace showed huge numbers of (unsuccessful) stats, with 82% dedicated to resolving “types”
- Patch backported from puppet 3.5, stat ops for default compilation from 8331 to 1381


CPU utilization on punch/puppet/master/batch
100
90
80
70
60
50
40
30
20
10
Tue
Wed
Thu
Fri
Sat
Sun
Mon
User aver: 38.47 max: 59.26 min: 5.06 curr: 22.19
System aver: 17.27 max: 43.53 min: 1.22 curr: 3.48
Nice aver: 2.99m max: 10.51m min: 0.00 curr:323.86u
Idle aver:43.05 max:92.50 min:160.08m curr:73.07
IO Wait aver:136.05m max:461.76m min: 11.49m curr:217.27m
IRQ aver: 0.00 max: 0.00 min: 0.00 curr: 0.00
Soft IRQ aver: 67.33m max: 91.74m min: 22.02m curr: 46.03m


Column
Format
400
300
200
100
Compilation time
Mon May 5
2014
Wed May 7
Fri May 9
Sun May 11
Time

Upstream fixes problems

- Some shaping of service necessary, but…
- Big performance headaches have been fixed by upstream
- …though we've had to be prepared to use patches / trunk

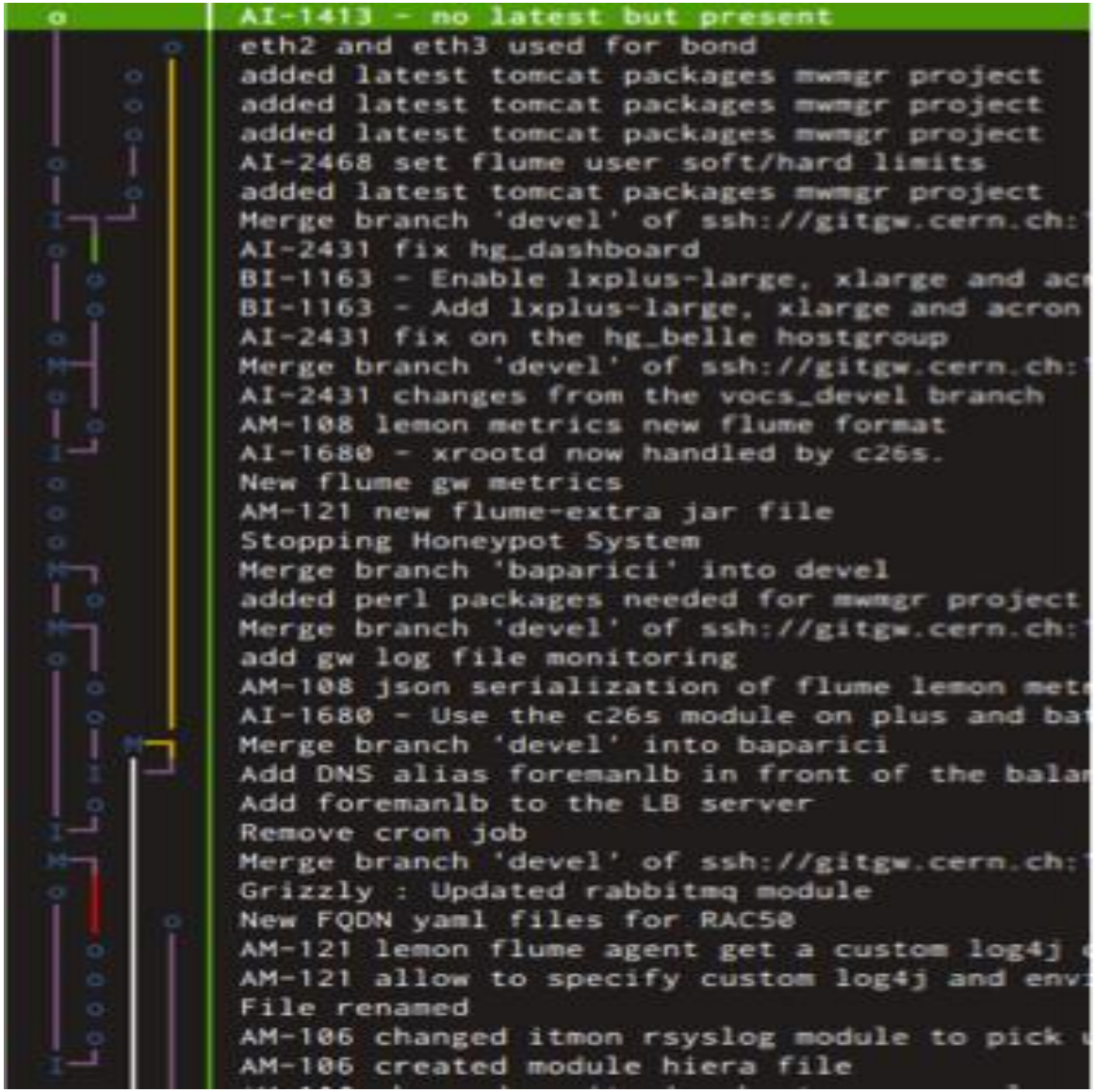
Original Dev practices too simple

- Puppet modules are a tree on masters, so initial plan was to treat them as single project
- One git repo, branches of “production” (master) and “dev” map to puppet environments
- Can’t merge dev -> prod without freezing
- Used cherry-pick to promote changes

Easy cherry-pick

```
o [devel] configure elasticsearch endpoint for kibana
o remove lb configuration and flume from teststack
o Adding virtual host for ermis
o condor faster discovery
o Do some clean-up in my hostgroup and create a new on
o condor hg, preemption disabled
o Adding template folder to my hostgroup
o Creating hostgroup for ermis.
o hg_vobox new variable
o extract all fields from nova api requests
o add regex nova api for flume
o kibana configuration data
o add standalone kibana configuration
o Added landb set integration
o Fixed error with defined/undef
o More complete configuration for myproxy
o Updated hiera following changes in myproxy hg/module
o Reorganized myproxy module
o use sssd_filter_users so that values defined in diff
o Add new its6 koji repository
o hg_bi rsyslog hiera server
o remove useless notify statment
o AI-2281 - Open ssh ports so aiadm can access.
o remove setting of rules from puppet on request of se
o add teststak - the openstack test instance
```

Not so easy


AI-1413 - no latest but present
eth2 and eth3 used for bond
added latest tomcat packages mwmgr project
added latest tomcat packages mwmgr project
added latest tomcat packages mwmgr project
AI-2468 set flume user soft/hard limits
added latest tomcat packages mwmgr project
Merge branch 'devel' of ssh://gitgw.cern.ch:
AI-2431 fix hg_dashboard
BI-1163 - Enable lxplus-large, xlarge and ac
BI-1163 - Add lxplus-large, xlarge and acron
AI-2431 fix on the hg_belle hostgroup
Merge branch 'devel' of ssh://gitgw.cern.ch:
AI-2431 changes from the vocs_devel branch
AM-108 lemon metrics new flume format
AI-1680 - xrootd now handled by c26s.
New flume gw metrics
AM-121 new flume-extra jar file
Stopping Honeypot System
Merge branch 'baparici' into devel
added perl packages needed for mwmgr project
Merge branch 'devel' of ssh://gitgw.cern.ch:
add gw log file monitoring
AM-108 json serialization of flume lemon met
AI-1680 - Use the c26s module on plus and ba
Merge branch 'devel' into baparici
Add DNS alias foremanlb in front of the bala
Add foremanlb to the LB server
Remove cron job
Merge branch 'devel' of ssh://gitgw.cern.ch:
Grizzly : Updated rabbitmq module
New FQDN yaml files for RAC50
AM-121 lemon flume agent get a custom log4j
AM-121 allow to specify custom log4j and env
File renamed
AM-106 changed itmon rsyslog module to pick
AM-106 created module hiera file

Now: modules are repos

- Each module is its own repository
- Hostgroup / Module split for services / reusable code
- Means that Service Managers and Module Maintainers can move at own pace
- the technical challenge was to create the single tree of puppet manifests for the puppet masters
- We'd hoped that puppet-librarian would do this

jens

- In the end we had to write our own librarian
- Puppet environments are collections of module / hostgroup branches
- “Golden” environments: “production”, “qa”, and user configurable environments

```
$ cat production.yaml
---
default: master
notifications: puppet-admins
```

```
$ cat ostest.yaml
---
default: master
notifications: os-tweakers
overrides:
  hostgroups:
    grizzly: ostest
  modules:
    openstack: ostest
```

jens

- Environments are created based on default branches and overrides
- jens symlinks to correct unpacked branch of each module

```
$ pwd
/mnt/puppetnfsdir/environments/ostest

$ readlink modules/openstack
../../../clone/modules/openstack/ostest/code

$ readlink hostgroups/hg_grizzly
../../../clone/hostgroups/grizzly/ostest/code

$ readlink modules/base
../../../clone/modules/base/master/code
```

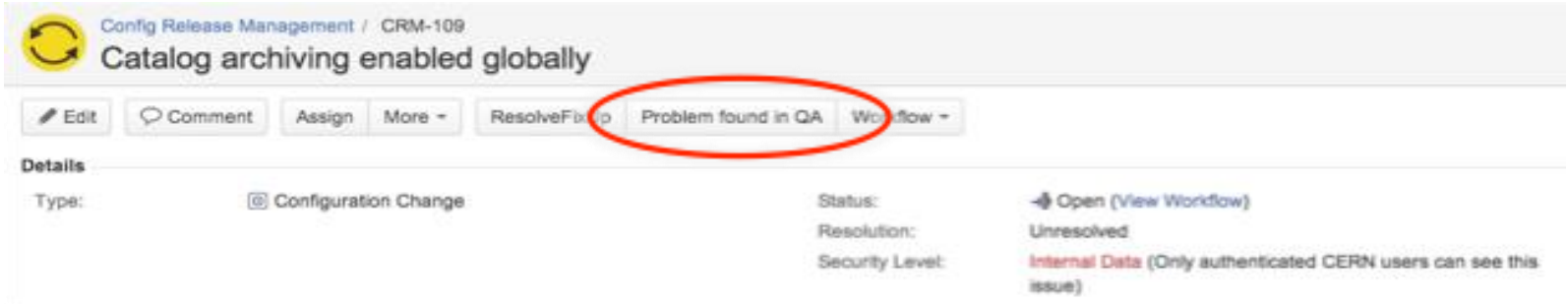
Infrastructure is code

- Each module and hostgroup is a git repository, but it drives configuration
- It's code, treat it like code, run it like a software project
- A running service is configured by many modules, with different groups developing them
- Need to manage risk and throughput
- Throughput and stability isn't a 0-sum game

Strong QA process

- Mandatory process for “shared” modules
  - recommended for non-shared
  - module maintainers expected to maintain QA & master branches
  - service managers expected to help with QA node coverage
  - changes are QA’d for >= 1 week
  - anyone can press the “stop” button.

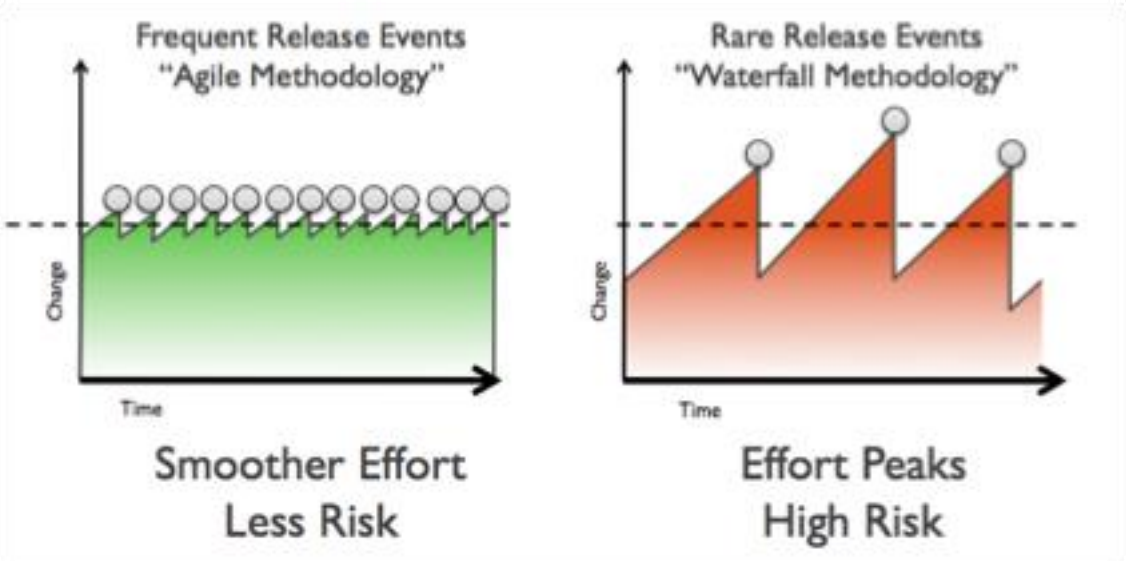
QA process



- Currently enforced only by convention and visibility
- Emergency workflow possible, with more visibility

Rate of change

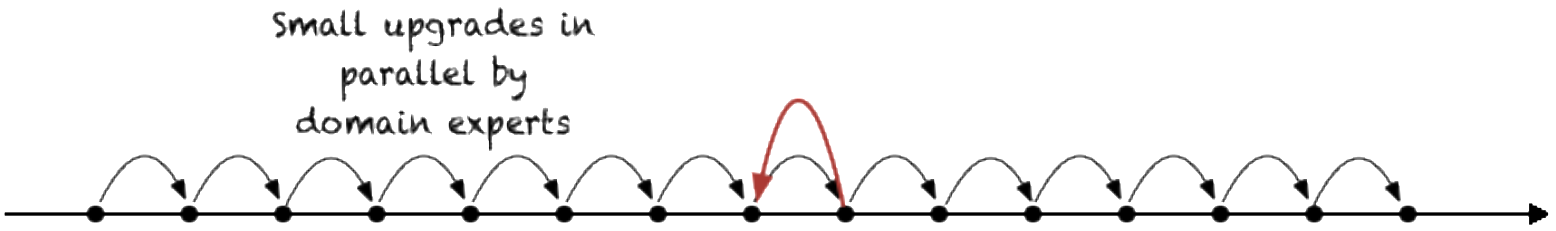
- By default changes flow individually into QA
- Changes flow individually into Prod after successful QA
- Production is always moving



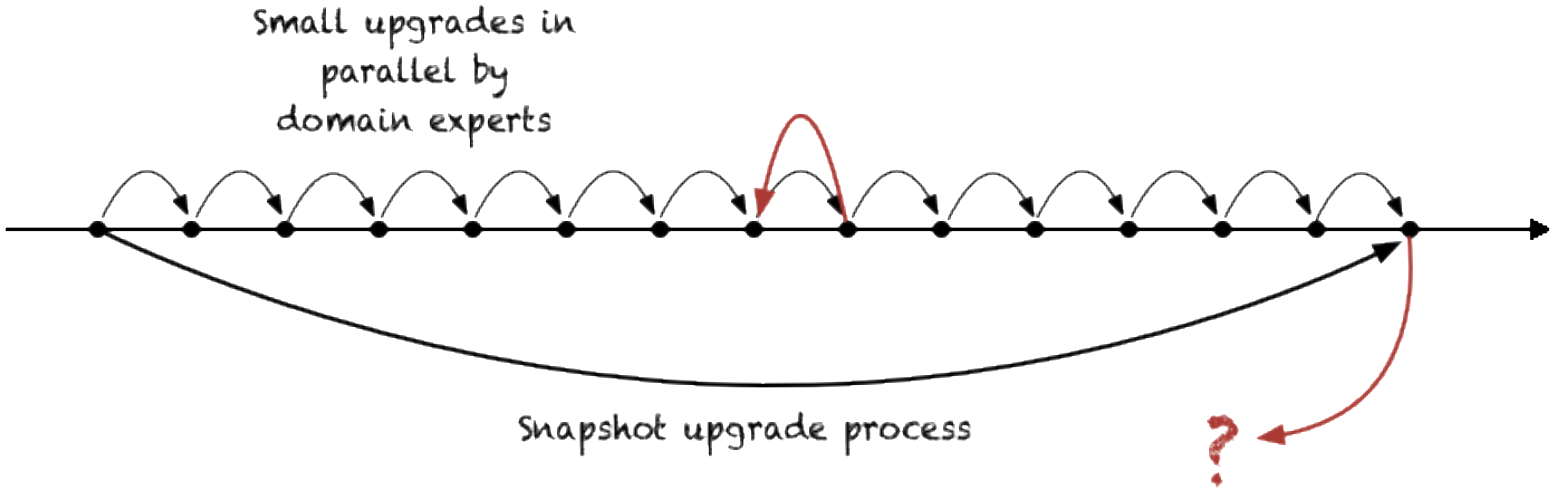
DBA's corner when mentioning automatic updates

Delivery != deployment

- Continuous delivery doesn't have to mean continuous deployment
- Whilst we believe that risk increases with time and number of changes, it's for services to determine best policy
- Snapshots of configuration: jens pointing to commits rather than branches for overrides
- yum repo snapshots
- Service Managers can "freeze" and upgrade in their own time



"flow" upgrade model



"snapshot" upgrade model

Canaries

- To manage rate of change, essential to detect failures
- “Canary” machines are exposed to changes sooner than other machines in service
- Can use “QA” or delayed production

Continuous Integration

- Still manual steps that could be automated
- Most changes are feature -> QA -> master
- Creating jenkins tests fro modules, and some functional tests
- Build pipeline to take feature branch and merge to QA, then production
- Make it easier to run with tests than without

Summary

- Upstream works: we're not alone with scale
- Change is inevitable; suffering is optional
  - Important to have levers for service managers to configure rate of change
- Let's stop doing the machines job for them
  - automate tests & build pipelines
- Things I didn't cover: software version drift, run book automation, infrastructure data


WHAT?
flickr: eddiedangerous