babiel

The Good, The Bad and The Ruby

Upgrading to Puppet 8

babiel

Disclaimer

- There will be some criticism, but:
- Puppet/OpenVox <3
- Still the best tool in the space

babiel

Our context

- 25 Puppet users
- 1300 Linux servers
- heterogenous - many different roles and variations
- 80 external modules in Puppetfile
- Upgrade performed in June 2024

babiel

When do we have to update?

Puppet version	Puppet Server version	PuppetDB version	Associated PE version	Projected EOL date
8.7.0	8.6.1	8.6.0	2023.x	Superseded by next developmental release.
7.31.0	7.17.1	7.19.0	2021.7.x	Superseded by next developmental release.
6.28.0	6.20.0	6.22	2019.8.x	February 2023

babiel

Turns out: now

Puppet version	Puppet Server version	PuppetDB version	Associated PE version	Projected EOL date
8.10.0	8.7.0	8.8.0	2023.x	Superseded by next developmental release.
7.34.0	7.17.3	7.20.0	2021.7.x	In overlap support. EOL February 2025.
6.28.0	6.20.0	6.22	2019.8.x	February 2023

babiel

What changed?

- Default settings
  - strict variables = true
  - exclude unchanged resources = true
  - strict = error
  - include legacy facts = false
- Ruby 3.2

babiel

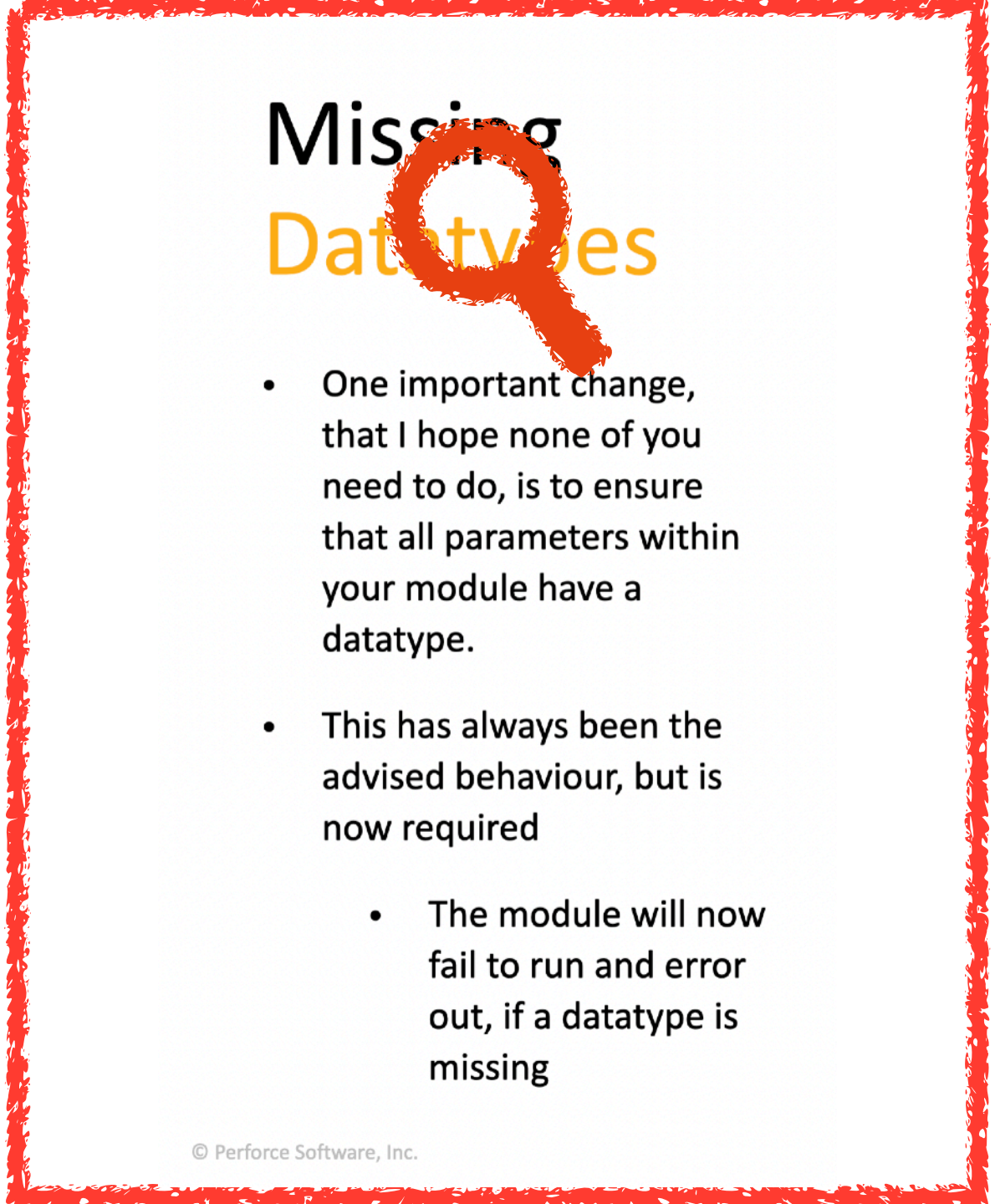
Already prepared

- Some settings already set
- strict variables = true
- exclude unchanged resources = true
  - why would reports ever contain unchanged resources?
  - not aware of any tool that shows them
  - previously changed to reduce risk of PuppetDB overload

babiel

strict = error

- Luckily it's only a new default
- Because you cannot reasonably use it
- Because it turns any deprecations fatal
- That includes deprecations raised by Puppet modules, not just Puppet itself
  - disable_warnings=deprecations does not help
  - Github issue: The deprecation function and Puppet 8 are not compatible #1391
- set strict=warning and move on



babiel

Are data types required?

- According to yesterday's talk, data types for parameters are required in Puppet 8
- This is not in the upgrade docs or release notes, or the language docs
- I also could not reproduce it
- (it is a good practice that you can enforce with a Vox Pupuli puppet-lint plugin)

babiel

AGENDA

Testing Puppet changes

babiel

The problem with RSpec

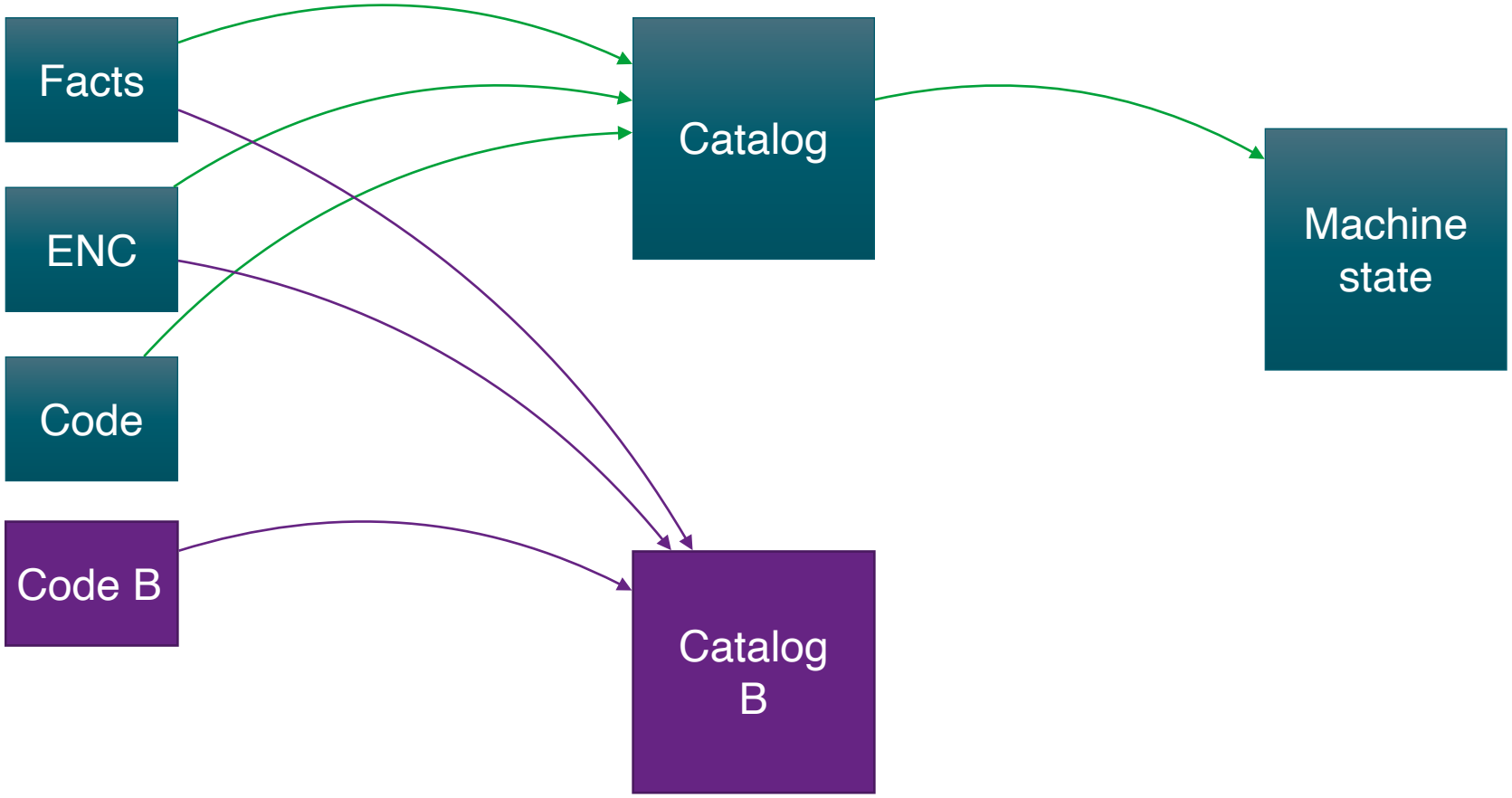
- not Ruby developers
  - and RSpec is even weirder
- Often tautological, tests just re-state the Puppet code
- Heterogeneous landscape
  - need to check all machines to make sure
- Have some tests for critical *logic*

octocatalog-diff

- The catalog is what Puppet will apply to machines
- Diffing catalogs gives us a very good idea of expected changes
- Process
  - Make changes
  - compile catalogs for all machines before and after
  - diff catalogs

babiel

Catalog



babiel

AGENDA

legacy facts

babiel

What are legacy facts

- From a time long ago, when Puppet did not have structured facts
- for example: ipaddress_eth0 -> networking.interfaces.eth0.ip
- Some are less obvious
  - hostname -> networking.hostname

babiel

Detecting legacy facts

- puppet-lint check
- but it misses:
  - ERB templates
  - EPP templates
  - Ruby code
  - Hiera config

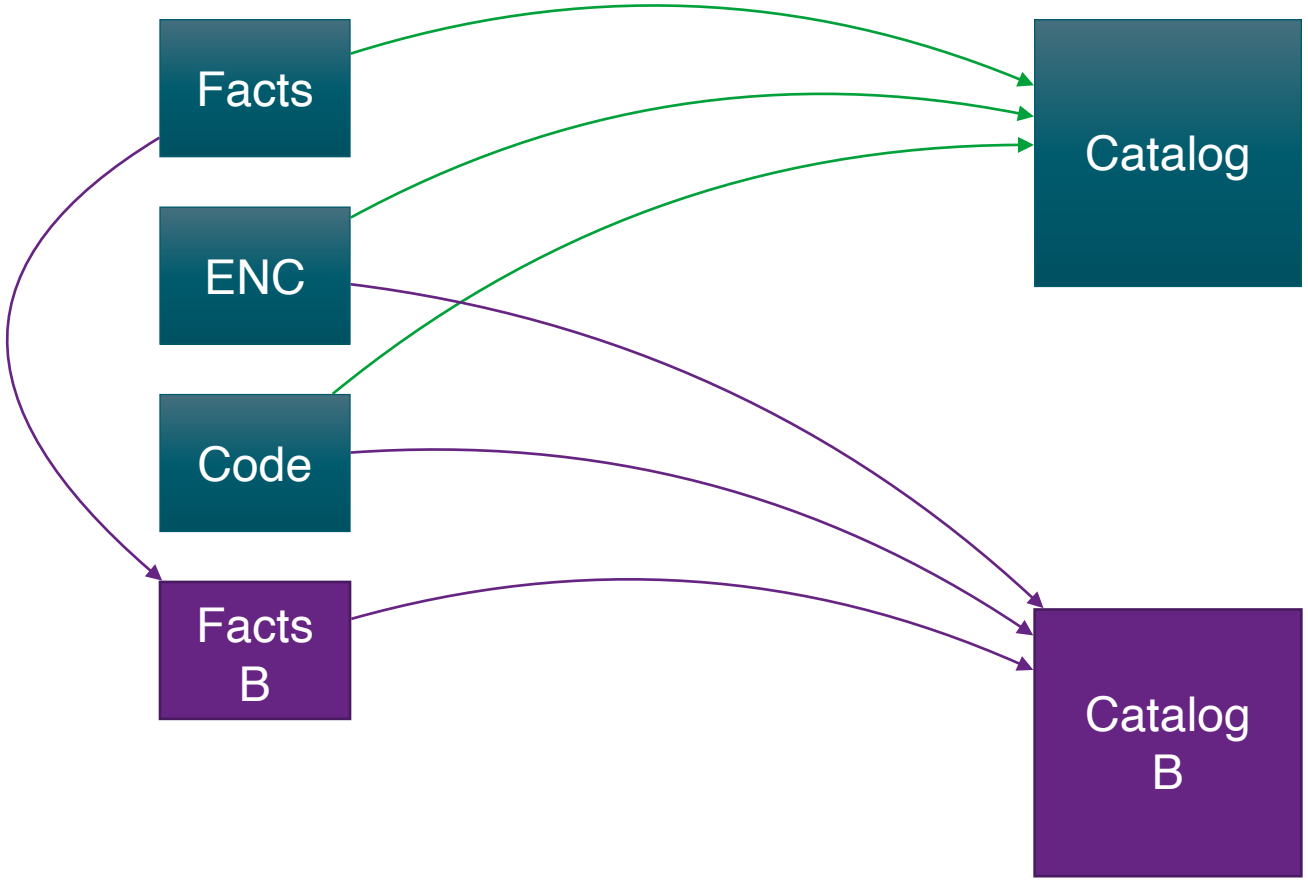
babiel

Testing without legacy facts

- octocatalog-diff can compare not just between code revisions
- but also between different settings, facts or ENC input
- but legacy facts is an agent-side setting
  - don't want to set it yet
- instead: load facts JSON, strip out legacy facts, feed into octocatalog-diff for comparison

babiel

Diffing legacy facts



github.com/babiel/strip-legacy-facts

- Facter has a YAML file describing all facts
  - including which are legacy (hidden = true)
- Download all facts from PuppetDB
- Write a Go program
  - that reads facts JSON
  - and prints it without the legacy facts
- or: run facter on nodes with hidden facts disabled

diffing

```
./bin/octocatalog-diff \
--from-fact-file fact-diff/legacy-facts/__HOST__.json \
--to-fact-file fact-diff/facts/__HOST__.json \
-n __HOST__
```

babiel

Disabling legacy facts on Puppet 7

- Good practice: split a big rollout into multiple smaller ones
- We can set include_legacy_facts=false on Puppet 7

babiel

Rollout to just 5% of agents

```
ini_setting { 'puppet.conf include_legacy_facts':
  ensure  => present,
  path    => "${facts['puppet_confdir']}/puppet.conf",
  section => 'agent',
  setting => 'include_legacy_facts',
  value   => !(fqdn_rand(100) < 5),
}
```

babiel

Result

- Mostly had to change our own code
- Only sent pull requests for 4 external modules

babiel

AGENDA

Ruby 3.2

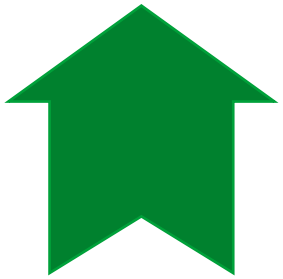
babiel

Ruby 3.2

- A bunch of changes, but only one that really affected Puppet:
  - Several duplicate methods were removed

File.exists?

File.exist?



Lots of facts

More linter trouble

- Rubocop is a Ruby linter
- That can detect deprecated methods
- Unfortunately, it did not know about FileTest
- We used a more advanced tool:
  - `grep -r -E '(File|Dir|FileTest).exists\?'`

babiel

Results

- We paused our first attempt to upgrade to Puppet 8 (in September 2023)
  - because so many modules needed patches
- When we resumed, it was luckily mostly resolved
- Turns out: Puppet 8.3.1 re-added those methods.
  - I learned this yesterday accidentally from the release notes
  - Not mentioned in the Upgrade Guide
  - also not for FileTest

Function uriescape

- Error: Could not call 'find' on 'catalog': Evaluation Error: Error while evaluating a Function Call, Puppet: This function is not available in Puppet 8. URI.escape no longer exists as of Ruby 3+.
- ok - so what's the replacement? not mentioned in Puppet's upgrade guide either
- Luckily module had an update
- Apparent replacement:
  - `inline_template('<%= URI::DEFAULT_PARSER.escape(@foo) %>')`

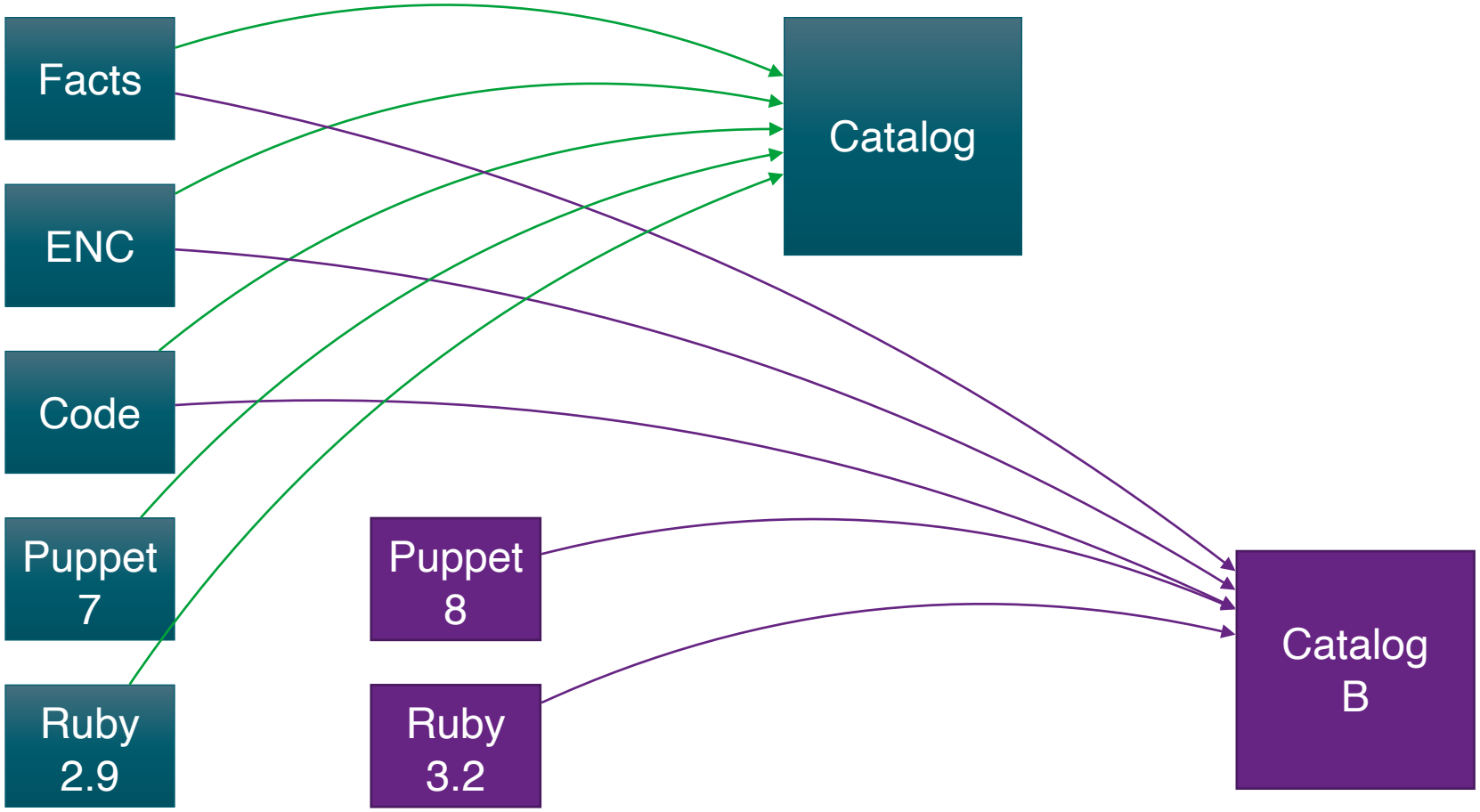
babiel

AGENDA

Diffing Puppet versions

babiel

Diffing Puppet versions



babiel

Setup second Puppet control repo

```
GITPATH_PUPPET=$PWD
git worktree add ../puppet-control@puppet8 puppet8-branch
cd ../puppet-control@puppet8
GITPATH_PUPPET_NEW=$PWD
# edit .ruby-version or whatever to set Ruby version
# edit Gemfile for Puppet version
bundle update puppet
bundle install --binstubs
./bin/r10k puppetfile install --verbose --force
```

babiel

Diff between

```
# modify .octocatalog-diff.cfg.rb
# settings[:pass_env_vars] = %w(LANG LC_CTYPE SSH_AUTH_SOCK)
# remove settings[:cached_master_dir]

./bin/octocatalog-diff --quiet \
  --from-puppet-binary ${GITPATH_PUPPET}/bin/puppet \
  --to-puppet-binary ${GITPATH_PUPPET_NEW}/bin/puppet \
  --bootstrapped-from-dir=${GITPATH_PUPPET} \
  --bootstrapped-to-dir=${GITPATH_PUPPET_NEW} \
  -n __HOSTNAME__
```

babiel

AGENDA

Rolling out

babiel

Our Puppet Server machines

- PuppetDB and its PostgreSQL
- One Puppet Server for compiling catalogs
- Another Puppet Server only for Puppet CA
- Separates the complicated and heavily-loaded part from the stateful parts
- Additionally we have Foreman

babiel

How to update?

- What's the upgrade path?
- How do you update incrementally?
- We know that a Puppet 8 server works with Puppet 7 agents
- But what about PuppetDB?

babiel

“The order in which you upgrade components is important. Always upgrade Puppet Server and PuppetDB simultaneously, including the puppetdb-termini package on Puppet Server nodes, and always upgrade them before you upgrade agent nodes. Do not run different major versions on your Puppet primary servers (including Server) and PuppetDB nodes.”

babiel

“If you upgrade PuppetDB without upgrading the PuppetDB-termini, your Puppet deployment should continue to function identically, with no loss of functionality. However, you may not be able to take advantage of new PuppetDB features until you upgrade the PuppetDB-termini.”

babiel

Our rollout

1. Puppet CA
2. PuppetDB
3. Puppet compiler
4. Agents

babiel

1. Puppet CA

- Verify that Puppet 8 generally works
- that agents can talk to it
- that Foreman can talk to it
- if it failed, we could have not signed new agents, but configuration would have been fine
- no problems

babiel

2. PuppetDB

- This one has the most state, therefore would be most annoying to roll back
- But we expected no problems, and had none
- Puppet Server 7 could still report to it fine

babiel

3. Puppet compiler

- Highest risk because configurations might change
- Could set up a new compiler machine and incrementally move agents to it
- But decided to trust our previous testing
- No problems

babiel

4. Agents

- Incrementally update machines to new packages
- No problems

babiel

AGENDA

Summary

babiel

Summary: Documentation

- Guide on required code changes mostly good
- Confusion on EOL dates or rollout procedure

Summary: Ecosystem

- If a module is not by Puppet Labs or Vox Pupuli, it is likely unmaintained
- octocatalog-diff apparently unmaintained
  - Our pull request: Fixes for Ruby 3.2 #302

babiel

Summary: Programming

- Puppet code is code, but the tools for it are still lacking compared to other programming languages
  - Incomplete linter coverage
  - No native test framework
- Ruby feels more like a liability than a feature

babiel

Last slide

- If you can read German, check out our blog: https://tech.babiel.com/
- If you live in Germany, check out our jobs: https://babiel.jobs
- Time for Questions