# Scaling and monitoring your puppetserver for thousands of clients

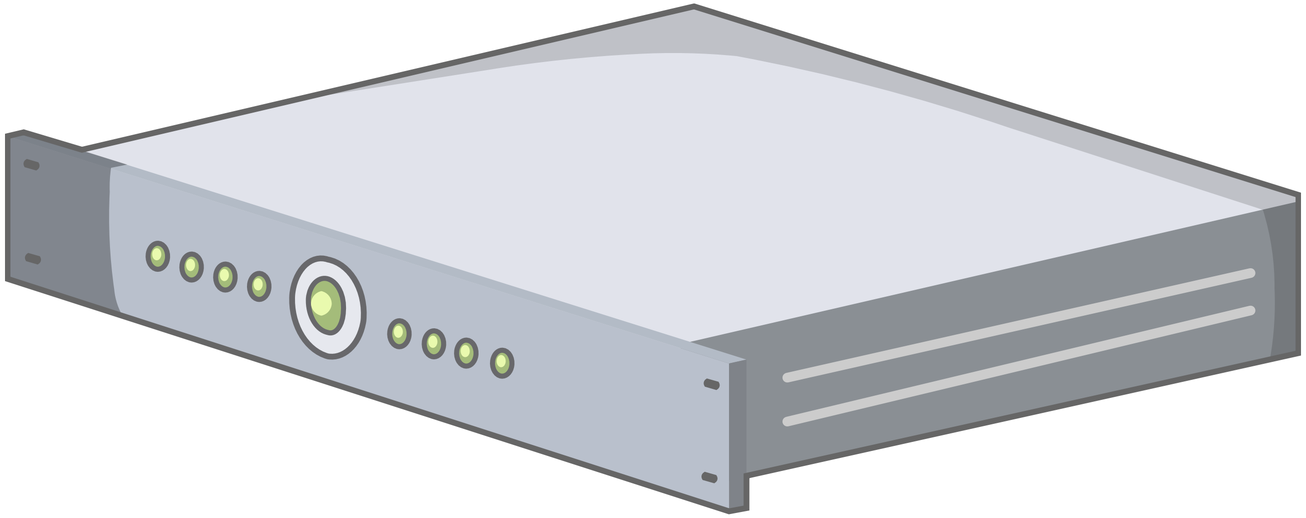
Includes all pitfalls!

## $ whoami

- Tim 'bastelfreak' Meusel
- DevOps Engineer at GoDaddy EMEA
- Puppet Contributor since 2012
- Merging stuff on Vox Pupuli since 2015
- Vox Pupuli PMC member

## starting point

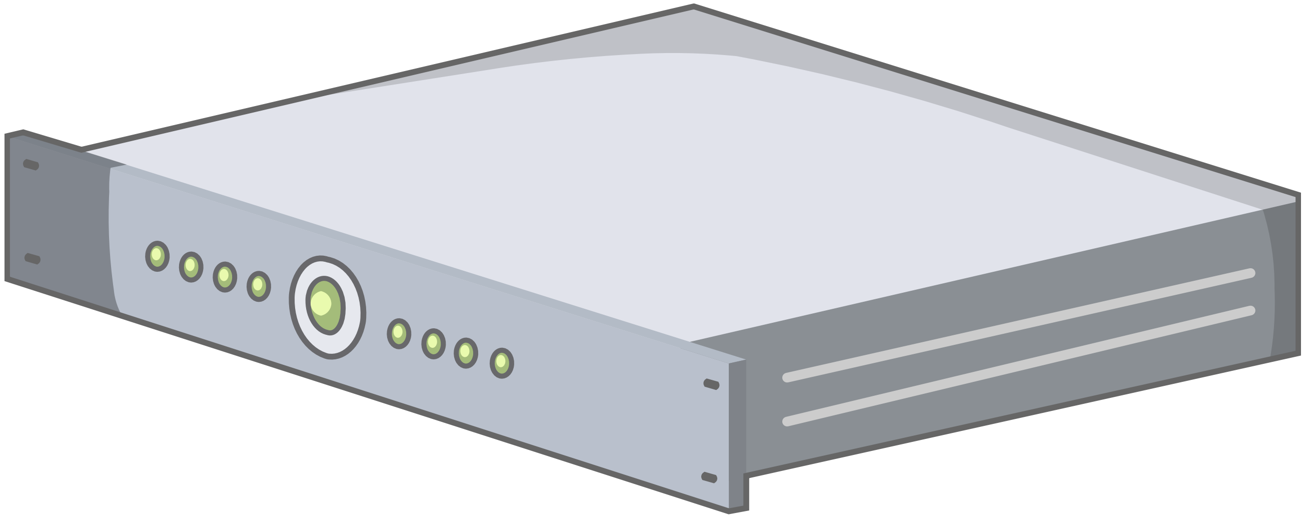
Puppetserver



## starting point

Puppetserver

PuppetDB

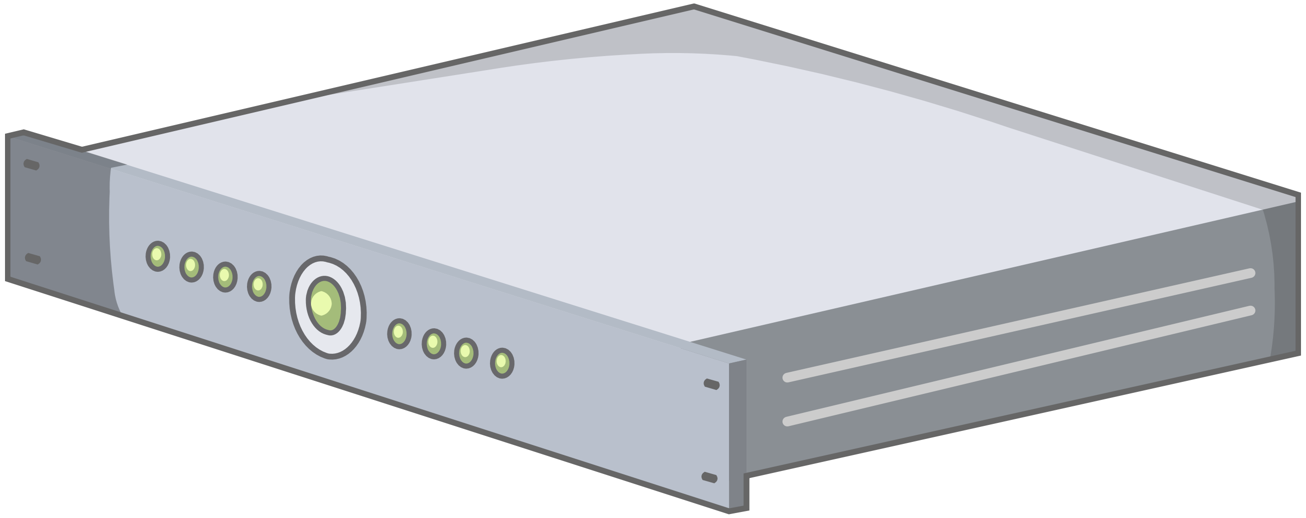


## starting point

Puppetserver

PuppetDB

Postgresql



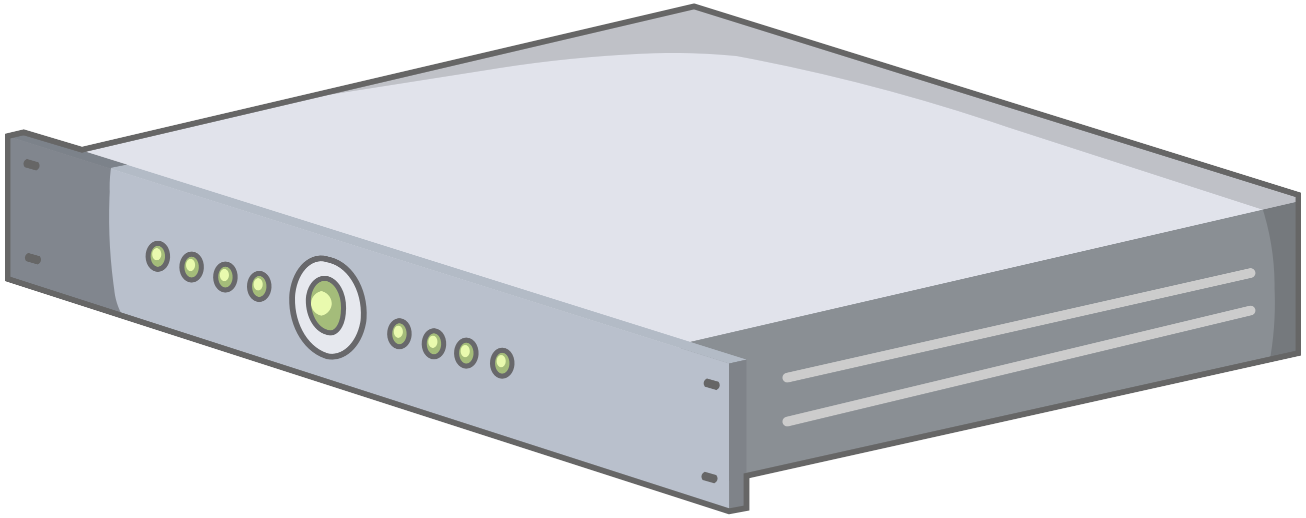
## starting point

Puppetserver

PuppetDB

Postgresql

PuppetBoard



## starting point

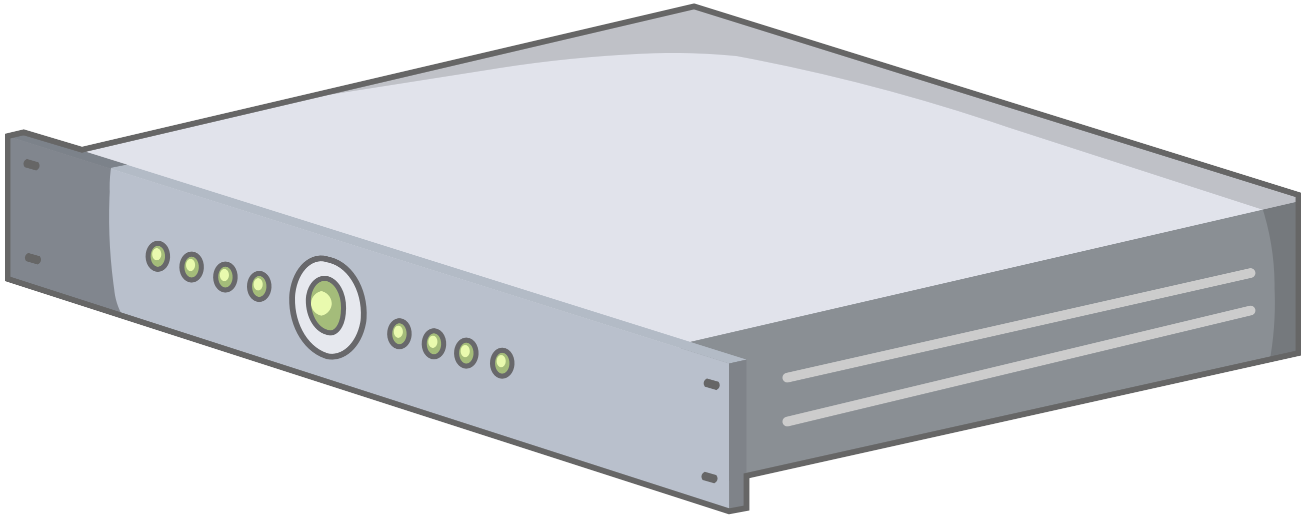
Puppetserver

PuppetDB

Postgresql

PuppetBoard

Foreman



## starting point

Puppetserver

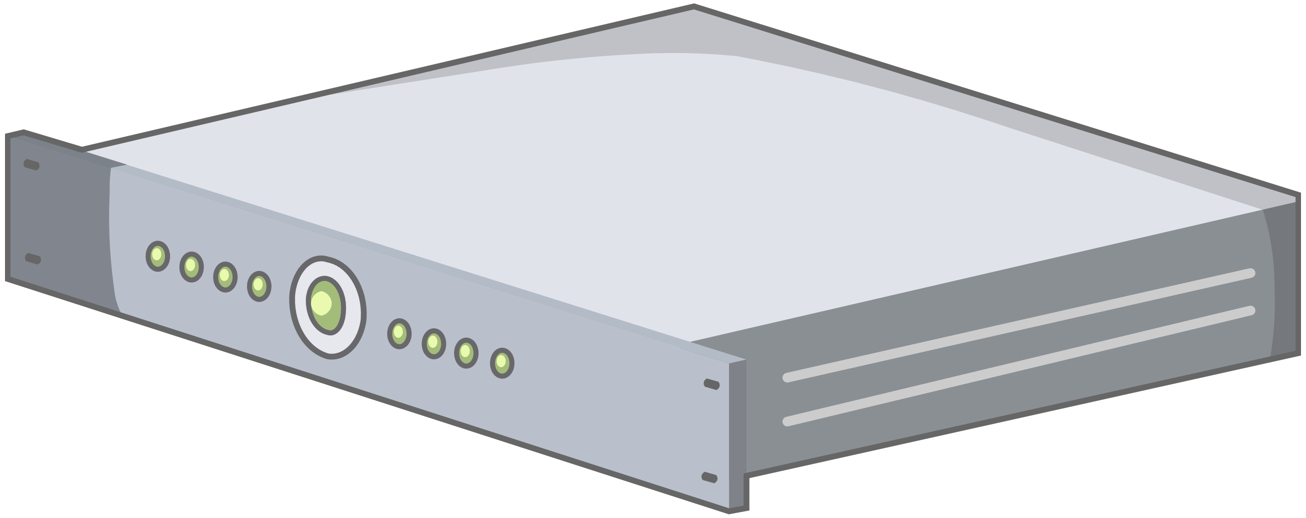
PuppetDB

Postgresql

PuppetBoard

Foreman

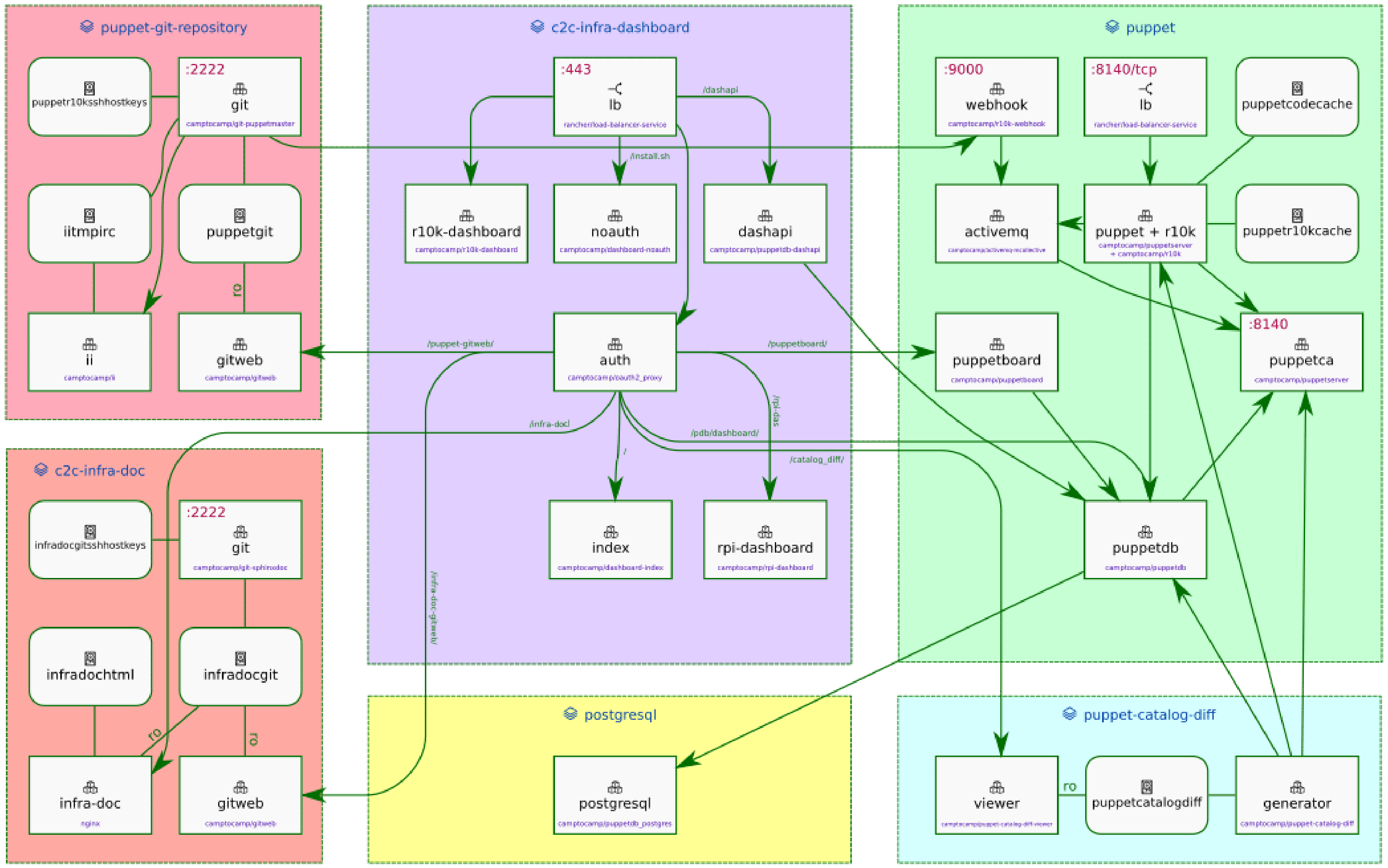
gnatsd

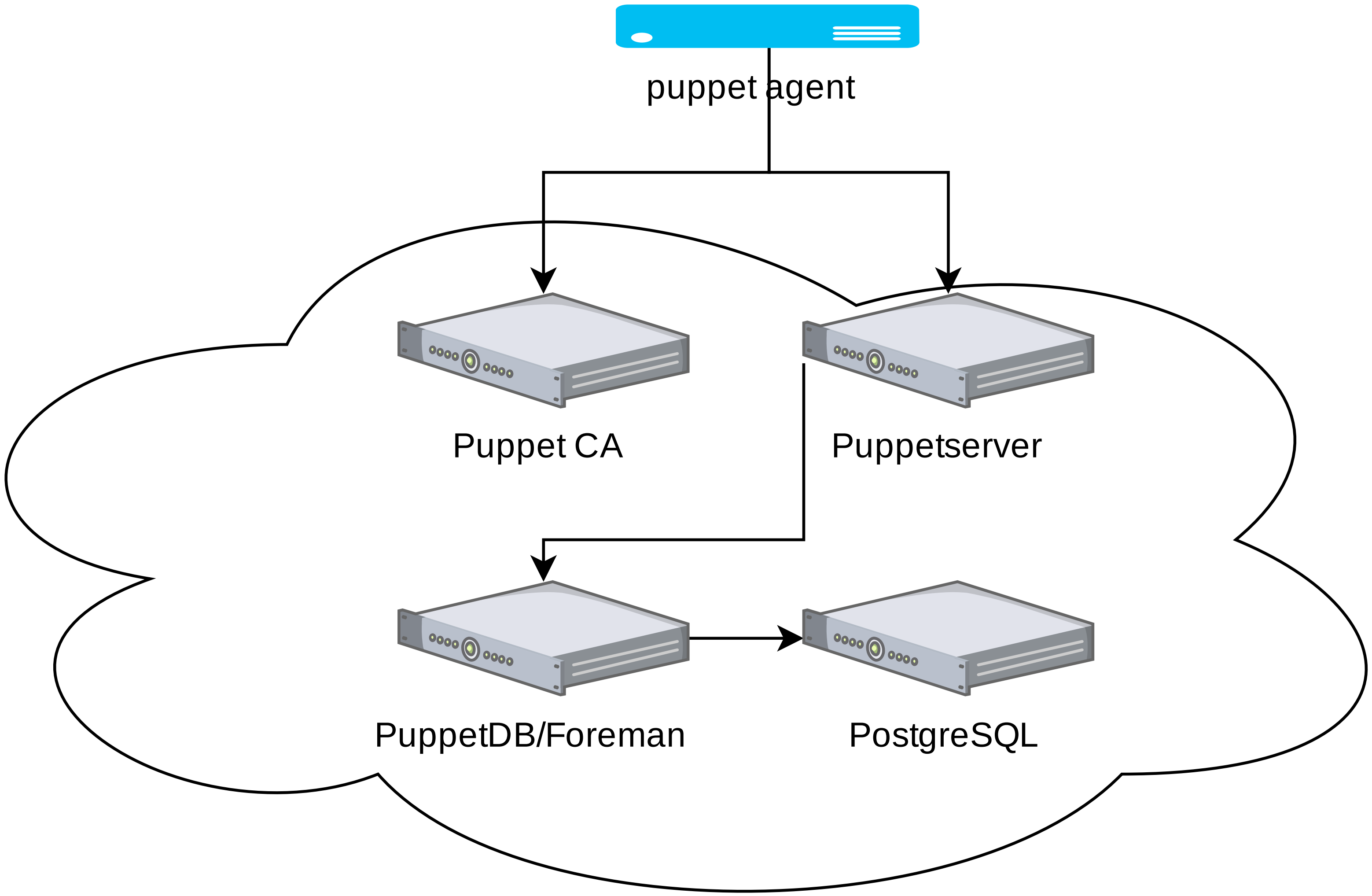


Coworker: Puppet agents are slow, can you please check the monitoring?

Coworker: Puppet agents are slow, can you please check the monitoring?

Uhm, which monitoring?


puppet-git-repository
puppetr10ksshhostkeys
:2222
git
camptocamp/git-puppetmaster
iitmpirc
puppetgit
ro
ii
camptocamp/ii
gitweb
camptocamp/gitweb
c2c-infra-dashboard
:443
lb
rancher/load-balancer-service
/dashapi
/install.sh
r10k-dashboard
camptocamp/r10k-dashboard
noauth
camptocamp/dashboard-noauth
dashapi
camptocamp/puppetdb-dashapi
/puppet-gitweb/
auth
camptocamp/oauth2_proxy
/puppetboard/
/rpi-das
/infra-doc/
/pdb/dashboard/
/
/catalog_diff/
index
camptocamp/dashboard-index
rpi-dashboard
camptocamp/rpi-dashboard
/infra-doc-gitweb/
puppet
:9000
webhook
camptocamp/r10k-webhook
:8140/tcp
lb
rancher/load-balancer-service
puppetcodecache
activemq
camptocamp/activemq-mcollective
puppet + r10k
camptocamp/puppetserver + camptocamp/r10k
puppetr10kcache
puppetboard
camptocamp/puppetboard
:8140
puppetca
camptocamp/puppetserver
puppetdb
camptocamp/puppetdb
c2c-infra-doc
infradocgitsshhostkeys
:2222
git
camptocamp/git-sphinxdoc
infradochtml
infradocgit
ro
ro
infra-doc
nginx
gitweb
camptocamp/gitweb
postgresql
postgresql
camptocamp/puppetdb_postgres
puppet-catalog-diff
viewer
camptocamp/puppet-catalog-diff-viewer
ro
puppetcatalogdiff
generator
camptocamp/puppet-catalog-diff



This is the cloud. The cloud is a half rack in your favourite datacenter

## Improvements

### Puppetserver

- Uses jruby within a JVM
- One jruby instance compiles one catalog at a time
- More instances => more catalogs per minute
- Use theforeman/puppet to configure puppeterver

```
---
puppet::server_max_active_instances: %{facts.processors.count}
```

## Improvements

### Puppetserver

- Puppetserver can cache code by loading it from disk to ram
- + Minimal decreased compilation time for each catalog
- - You should clean the cache after each deploy

```
---
puppet::server_environment_class_cache_enabled: true
```

- cleaning the cache:

```
 print hostcert) \
stprivkey) \
 cacert) \

server):8140/puppet-admin-api/v1/environment-cache?environment=productio
```

@bastelsblog for @voxpupuliorg

## Improvements

### Puppetserver

- theforeman/puppet creates the `development` and `production` environment
- r10k purges unknown environments
- theforeman/puppet restarts puppetserver if it creates an environment
- I have no git branches named `development` nor `production` in my control repo...
- Each environment deploy lead to a restarted puppetserver for weeks

```
---
# don't create development/production env
puppet::server_environments: []
# don't create /etc/puppetlabs/code/environments/common
puppet::server_common_modules_path: ''
```

@bastelsblog for @voxpupuliorg

## Improvements

### Puppetserver

- JVM has a configureable minimal and maximal amount of memory to allocate
- Memory is shared across all jruby instances (and other threads)
- Puppet docs suggest that minimal=maximal memory
  - That is based on Java 6 docs, so probably outdated
- Required memory per instance depends entirely on the codebase (modules)
- 2GB seem to work out fine for my setup

```
# How do I do this with hiera?
$cpu_count_twice = $facts['processors']['count'] * 2
$cpu_count = $facts['processors']['count'] * 1
class{'puppet':
  server_jvm_min_heap_size => "${cpu_count}G",
  server_jvm_max_heap_size => "${cpu_count_twice}G",
}
```

## Improvements

Puppetserver

Foreman

- foreman supports caching out of the box
- we use saz/memcached to provision memcached

```
# 50GB of cache
memcached::max_memory: 51200
foreman::plugin::memcache::hosts:
  - 127.0.0.1
```

```
include memcached
include foreman::plugin::memcache
```

## Improvements

Puppetserver

Foreman

- `passenger-status` says it only runs with a single process..
- theforeman/foreman uses puppetlabs/apache to configure passenger

```
--
pache::mod::passenger::passenger_max_pool_size: %{facts.processors.count
pache::mod::passenger::passenger_min_instances: %{facts.processors.count
```

## Improvements

Puppetserver

Foreman

- We use theforeman/foreman to manage foreman
- Tuned puppetserver results in more requests to foreman

```yaml
---
# default is 5
foreman::db_pool: 20
foreman::keepalive: true
foreman::max_keepalive_requests: 1000
foreman::keepalive_timeout: 180
```

## Improvements

Puppetserver

Foreman

PostgreSQL

- This deserves a dedicated conference (there actually is)
- Attend the PostgresConf or the PostgreSQL DevRoom at FOSDEM
- Ask people in #postgresql on freenode
- Use at least postgres 10 and rely on the upstream repos if possible
- Don't use harddrive, SATA/SAS SSDs and NVMe SSDs are the way to go
- Execute pgtune

```
pgtune -i /var/lib/pgsql/10/data/postgresql.conf -o postgresql5.conf
```

```
postgresql::server::config_entry{'max_connections':
  value => 400,
}
```

## Improvements

Puppetserver

Foreman

PostgreSQL

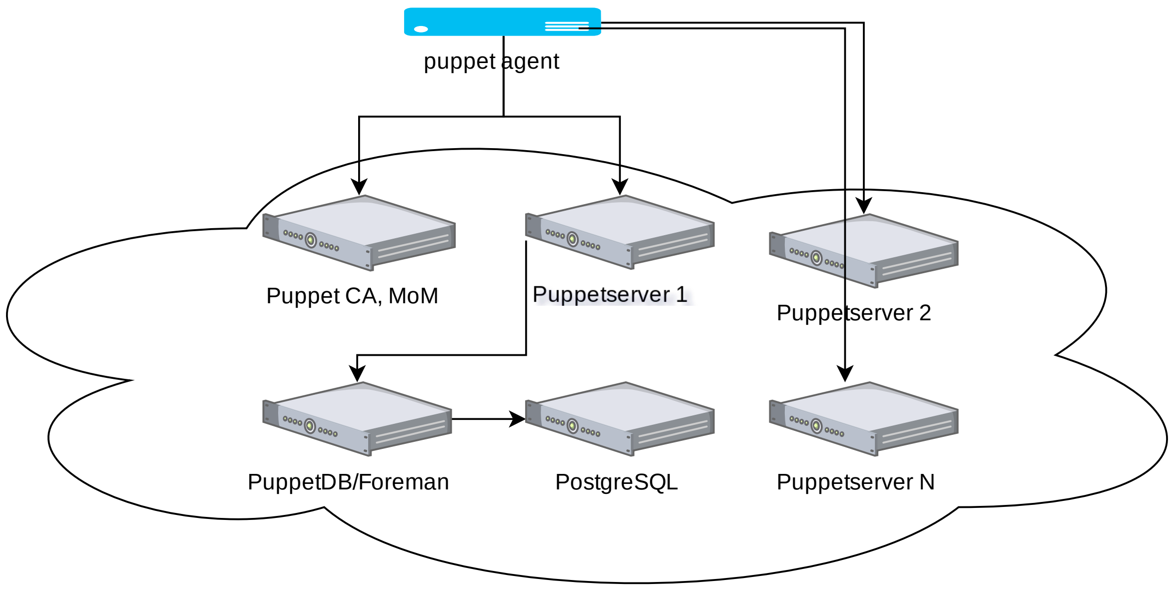
**PuppetDB**

- Simply JVM service that scales good with the amount of threads
- It's a RESTful service that stores data in PostgreSQL

```
puppetdb::server::java_args:
  '-Xmx': '8192m'
  '-Xms': '2048m'
puppetdb::server::node_ttl: '14d',
puppetdb::server::node_purge_ttl: '14d',
puppetdb::server::report_ttl: '999d'
# default is 50
puppetdb::server::max_threads: 100
# default is processorcount / 2
puppetdb::server::command_threads: %{facts.processors.count}
# default is 4, have your database in mind
puppetdb::server::concurrent_writes: 8
puppetdb::server::automatic_dlo_cleanup: true
```

## Scaling

### The Idea



This is the cloud. The cloud is one rack in your favourite datacenter

## Scaling

The Idea

tk-jetty9

Puppetserver docs:

- `selector-threads`: This sets the number of selectors that the webserver will dedicate to processing events on connected sockets for unencrypted HTTPS traffic. No known upper limit
- `ssl-selector-threads`: same as `selector-threads`, just for HTTPS. "Defaults to the number of virtual cores on the host divided by 2, with a minimum of 1 and maximum of 4"

## Scaling

The Idea

tk-jetty9

© knowyourmeme.com

- Deploy nginx on each Puppetserver server to terminate TLS
  - Increase default threadcount from 1 to $more...
- Bind puppetserver to localhost with http
  - And don't require TLS client certificates
- Setup consul for dynamic loadbalancing across all puppetservers
- Soonish available at github.com/bastelfreak/puppetcontrolrepo

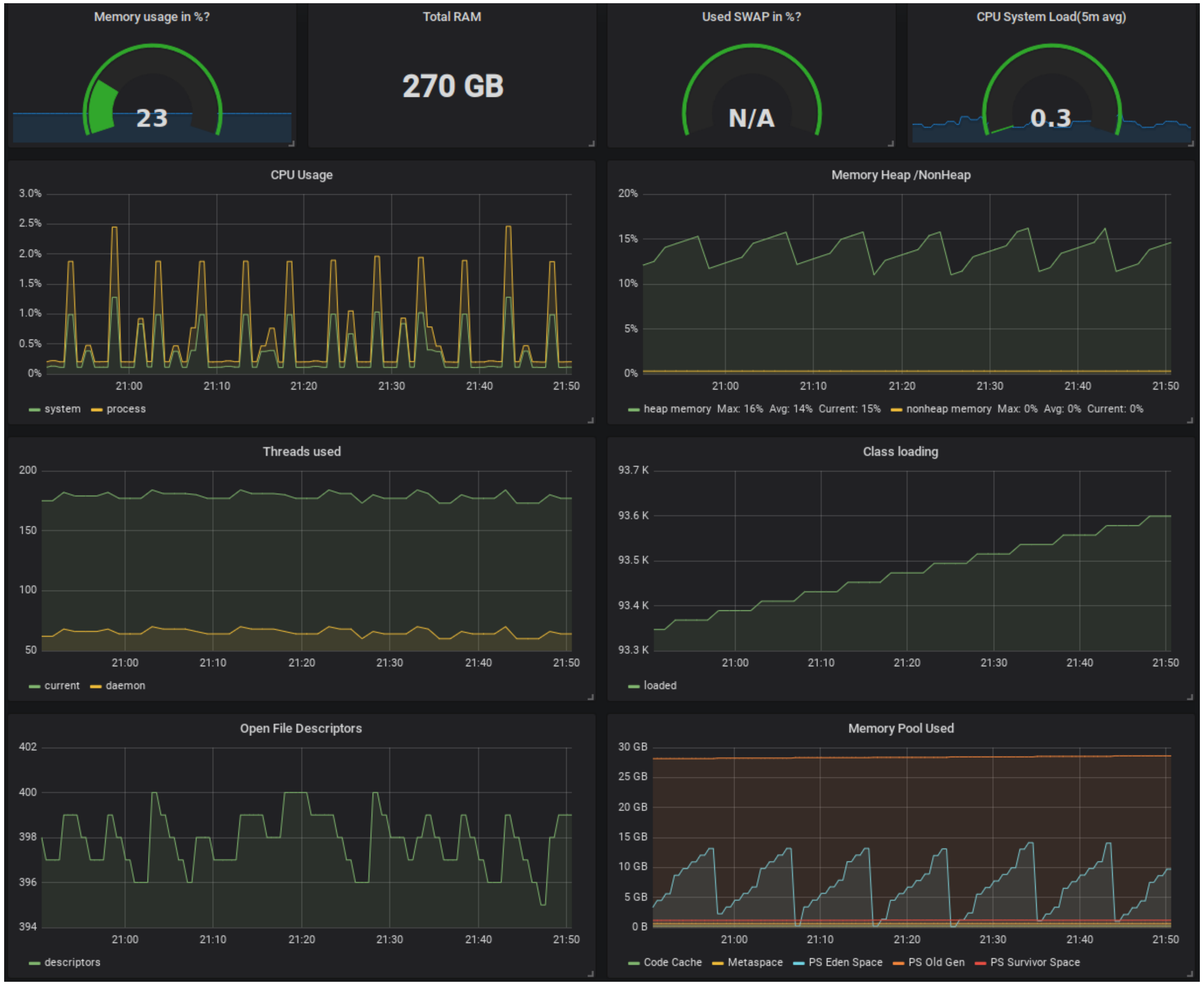
## Metrics

- Puppetserver can expose graphite
  - A graphite stack looks very complicated
- Puppetserver has a metrics API and exposes JMX data via Jolokia
- We can load a prometheus exporter into the JVM to write metrics into a prometheus instance

```
puppet::server_jvm_extra_args:
  - '-javaagent:prometheus.jar=127.0.0.1:9020:config.yaml'
```

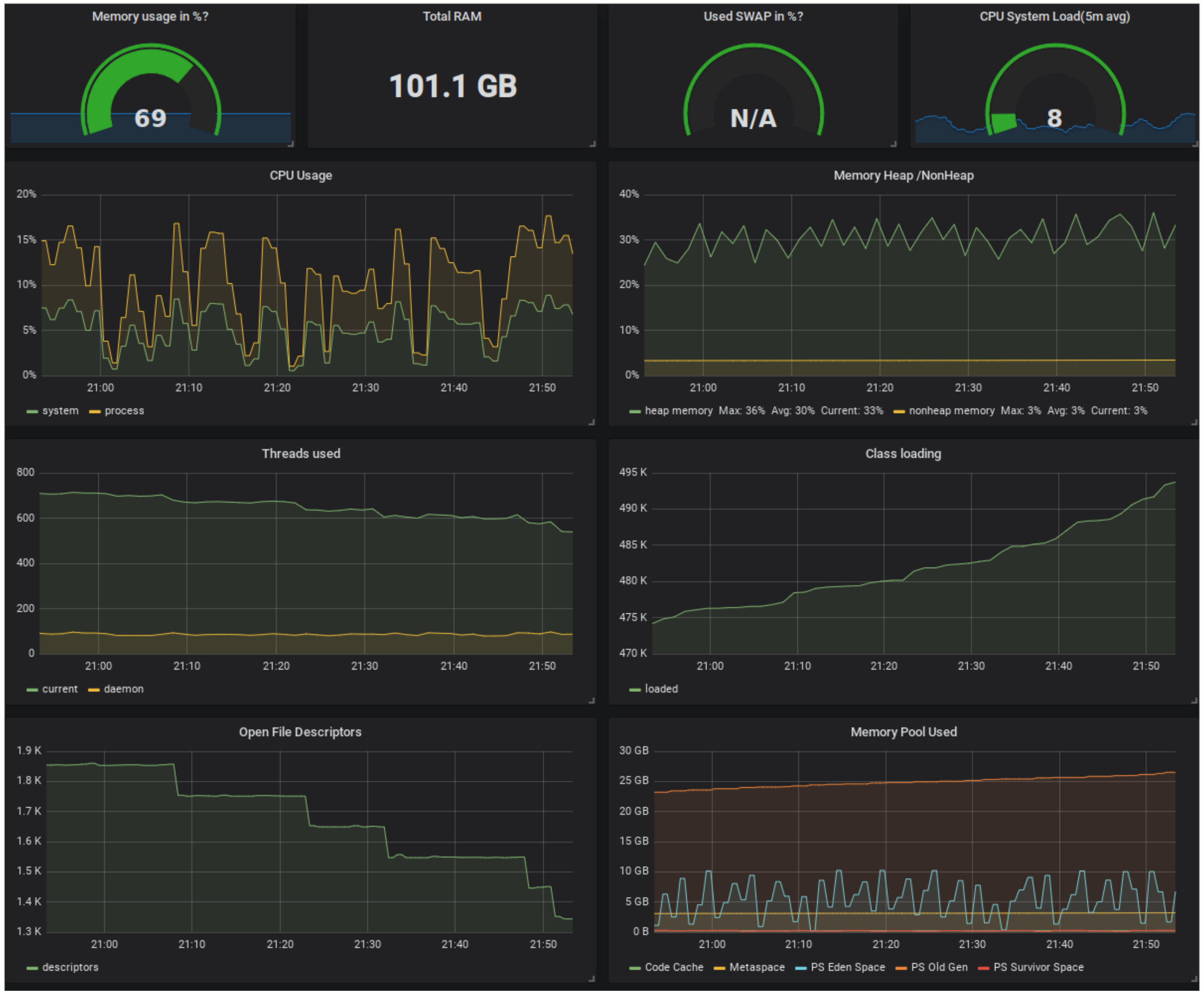
## Metrics

### Generic JMX



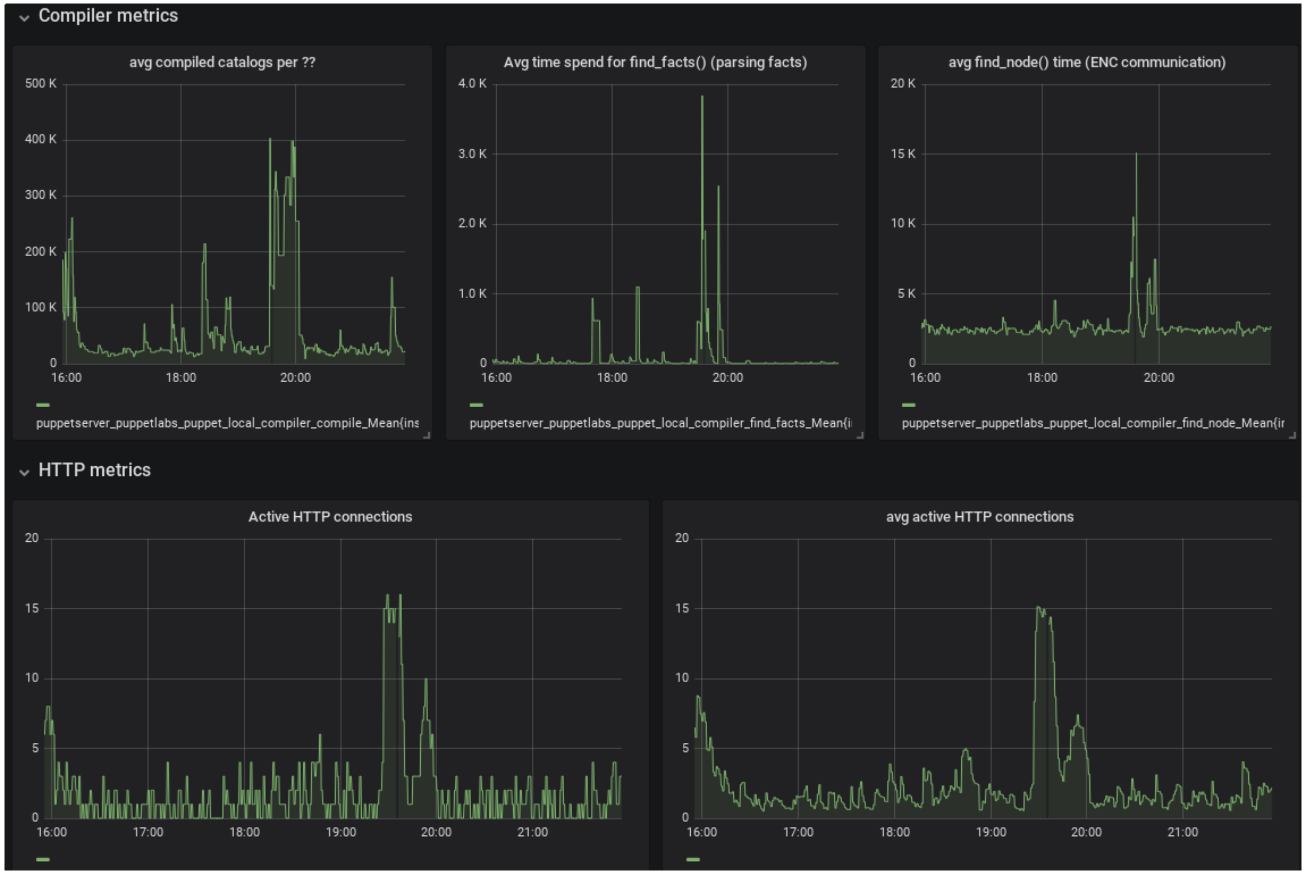
# Metrics

## Generic JMX



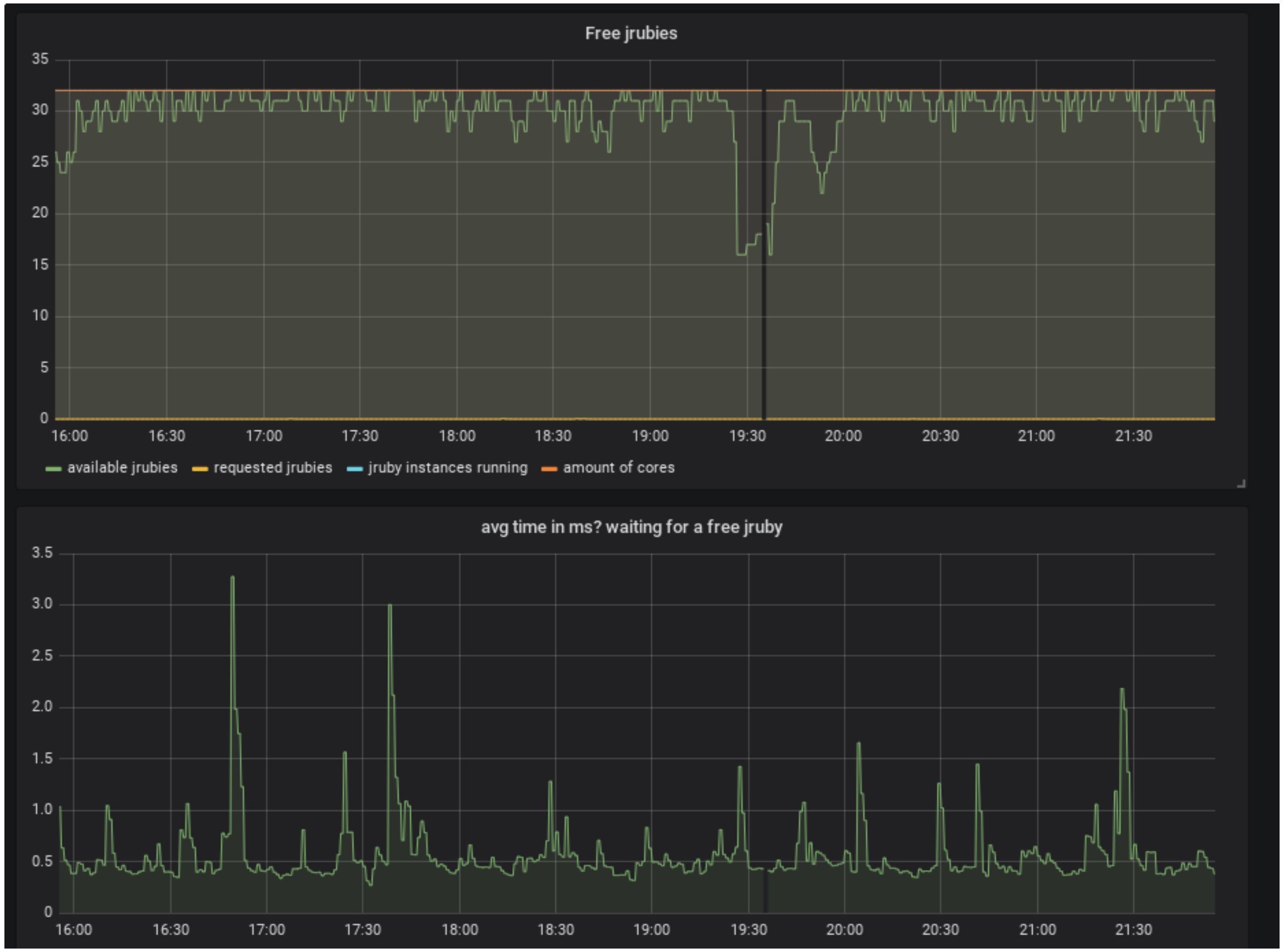
# Metrics

## Puppetserver


Compiler metrics
avg compiled catalogs per ??
Avg time spend for find_facts() (parsing facts)
avg find_node() time (ENC communication)
puppetserver_puppetlabs_puppet_local_compiler_compile_Mean{ins
puppetserver_puppetlabs_puppet_local_compiler_find_facts_Mean{i
puppetserver_puppetlabs_puppet_local_compiler_find_node_Mean{ir
HTTP metrics
Active HTTP connections
avg active HTTP connections

# Metrics

## Puppetserver



## Summary

Scaling a Puppetserver stack

- It's a complex distributed system with many tunables and pitfalls
- Start with proper monitoring instead of guessing
- Best practice controlrepo with all tuneables, explanations, unit/acceptance tests
- Contact: tim@bastelfreak.de or bastelfreak on freenode
- Collection of related talks

## Thanks for your attention!