



Xikolo Infrastructure

openHPI, openSAP and beyond

Linux Containers Seminar

Malte Swart
Hasso-Plattner Institute

Potsdam, Germany, May 6, 2015



Table of Contents

1. Tracing A Web Request
2. Tracing A Commit / A Release
3. Anything Else?
4. All Together
5. What's next?

Tracing A Web Request

External LoadBalancer

- ▶ terminates all external communication
- ▶ does HTTPS offloading (internal net is protected)

External LoadBalancer

- ▶ terminates all external communication
- ▶ does HTTPS offloading (internal net is protected)
- ▶ responsible for all here hosted domains: `open.hpi.de`,
`open.sap.com`, `codeocean.openhpi.de`, `blog.openhpi.de`
...
- ▶ based on domain forwarding to different sets of backend services (up to 10 per set/domain)

External LoadBalancer

- ▶ terminates all external communication
- ▶ does HTTPS offloading (internal net is protected)
- ▶ responsible for all here hosted domains: `open.hpi.de`,
`open.sap.com`, `codeocean.openhpi.de`, `blog.openhpi.de`
...
- ▶ based on domain forwarding to different sets of backend services (up to 10 per set/domain)
- ▶ (software) LoadBalancer for failover



By wiki.nginx.org

- ▶ User facing web server: NGinx



By wiki.nginx.org

- ▶ User facing web server: NGinx
- ▶ forwards special services like piwik (provided on subpath) to special VMs



By wiki.nginx.org

- ▶ User facing web server: NGinx
- ▶ forwards special services like piwik (provided on subpath) to special VMs
- ▶ serves application assets (images, javascript ...) directly
- ▶ provides basic cache header and compression options
- ▶ forwards remaining requests to application



From
wikipedia



From rails
bootstrap
app

- ▶ each worker processes requests serial (unicorn)
- ▶ current frontends: web, API and some very special purpose frontends
- ▶ main tasks of frontends: session management, authentication and authorization, template rendering + frontend stuff (html ...)
- ▶ data are requested from services via HTTP/JSON



By Peter Gilg

- ▶ stored within central Redis (key-value storage)
- ▶ cache responses for statically objects like user, course
- ▶ cache is invalidated on data change → should be up to date at every time



By haproxy.org



By wiki.nginx.org

- ▶ HAProxy for large instances like openHPI and openSAP
- ▶ monitors list of possible upstream continuously via health check
- ▶ redirects internal HTTP requests to one available backend
- ▶ for single instance: Nginx sites instead of load balancing



From
wikipedia



From rails
bootstrap
app

- ▶ again Ruby on Rails
- ▶ currently each backend server runs every service
- ▶ provides JSON api for frontends and other services



Provided by The PostgreSQL Community Association

- ▶ every service has individual database, could use their matching database tools
- ▶ mostly PostgreSQL 9.1
- ▶ but e.g. ElasticSearch, too



- ▶ Asynchronous communication between services
- ▶ Services publishing messages like new user created
- ▶ other services subscribe to all events they need



Simple, efficient background processing for Ruby.

From sidekiq.org

- ▶ Background processing

Tracing A Commit / A Release



Provided by Gitlab

- ▶ pushed into central gitlab



Provided by JetBrains

It gets a little bit complicated!

In general: Teamcity monitors Gitlab for new changes

Unit Tests

- ▶ Independent per service: all external communication is stubbed / disabled
- ▶ Ensure that service fulfils request specification

Integration Tests

- ▶ Goal: Testing all together
- ▶ checking out all services in its version
- ▶ running all services in productive-ish configuration
- ▶ executes selenium tests to check whether the interaction between services is working

Packaging

- ▶ Again independent per service
- ▶ building deb packages from source code
- ▶ skips optional things like test directory
- ▶ embedding all needed gems for service
- ▶ compiles assets (if needed)
- ▶ generates upstart scripts for all components (unicorn, sidekiq ...)

Release

- ▶ chooses a revision combination that resulted in green integration
- ▶ is executed automatically for opensap staging / manual for all other instances
- ▶ publishes packages into instance specific Debian repository (packages are reused)



Provided by PuppetLabs Media Kit

- ▶ runs on every productive VM
- ▶ upgrades Xikolo packages with new ones from repository
- ▶ executes database migrations
- ▶ injects production specific configurations (database credentials, logging settings ...)
- ▶ manages Xikolo configuration and service endpoint mapping

Anything Else?

User Content

- ▶ Beside database records primary files
- ▶ shared between nodes via central NFS

E-Mail

Central e-mail relay

Logging

Central Syslog Server

All Together

Basic Structure

- ▶ Frontend-LoadBalancer
- ▶ Compute nodes with VMs
- ▶ database server with PostgreSQL, Redis, RabbitMQ, ElasticSearch
- ▶ one infrastructure server one with syslog, exim, nfs-server

What's next?

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)
- ▶ Increase maintainability (current 3 server can not be upgraded/adjusted without downtime)

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)
- ▶ Increase maintainability (current 3 server can not be upgraded/adjusted without downtime)
- ▶ Convert pull component of deployment process (puppet) into push process

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)
- ▶ Increase maintainability (current 3 server can not be upgraded/adjusted without downtime)
- ▶ Convert pull component of deployment process (puppet) into push process
- ▶ Make instances more independent with regard of more special configuration / service list

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)
- ▶ Increase maintainability (current 3 server can not be upgraded/adjusted without downtime)
- ▶ Convert pull component of deployment process (puppet) into push process
- ▶ Make instances more independent with regard of more special configuration / service list
- ▶ More precise scaling (more specific per service or service process like unicorn, msgr, sidekiq)

Mid/long-term improvements

- ▶ Increase reliability (current 3 server must not fail)
- ▶ Increase maintainability (current 3 server can not be upgraded/adjusted without downtime)
- ▶ Convert pull component of deployment process (puppet) into push process
- ▶ Make instances more independent with regard of more special configuration / service list
- ▶ More precise scaling (more specific per service or service process like unicorn, msgr, sidekiq)
- ▶ Replace central file system (NFS) with distributed service approach