

Cloud

let's move
the java world

Patterns

Nicolas De Loof - cloudbees



This talk is a mashup

- « PaaS design »

Michael Neale, CloudBees



- « Cloud Best practices »

Eric Bottard, VMWare

- my own ...

✘ Impossible d'afficher l'image. Votre ordinateur manque peut-être de mémoire pour ouvrir l'image ou l'image est endommagée. Redémarez l'ordinateur, puis ouvrez à nouveau le fichier. Si le x rouge est toujours affiché, vous devrez peut-être supprimer l'image avant de la réinsérer.

<me>



Support engineer



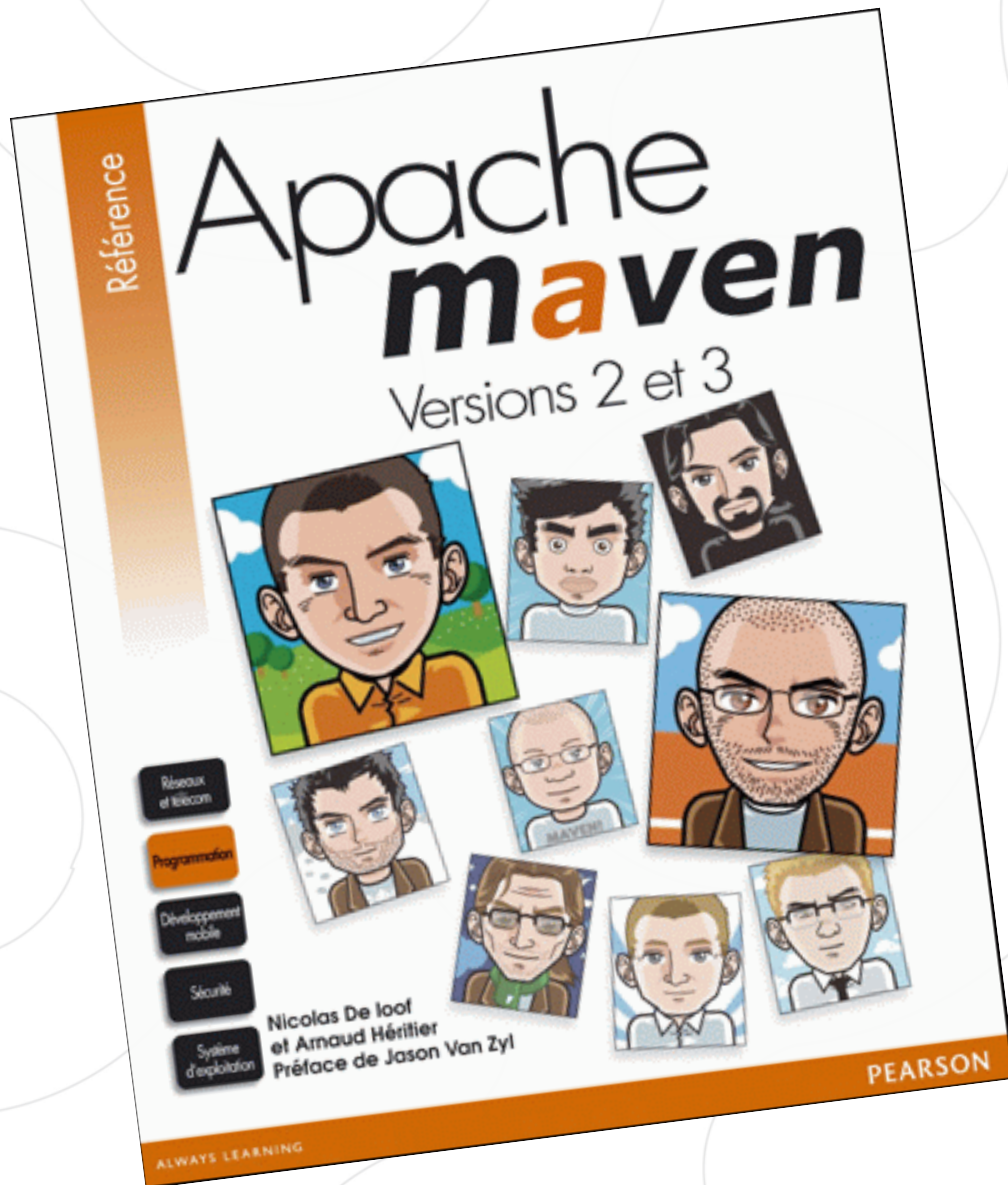
CloudBees[®]

The Java™ PaaS Company



censuré

Maven & Jenkins commiter



BreizhJUG leader



Breizh C@mp
Mix de technologies

</me>



End Users

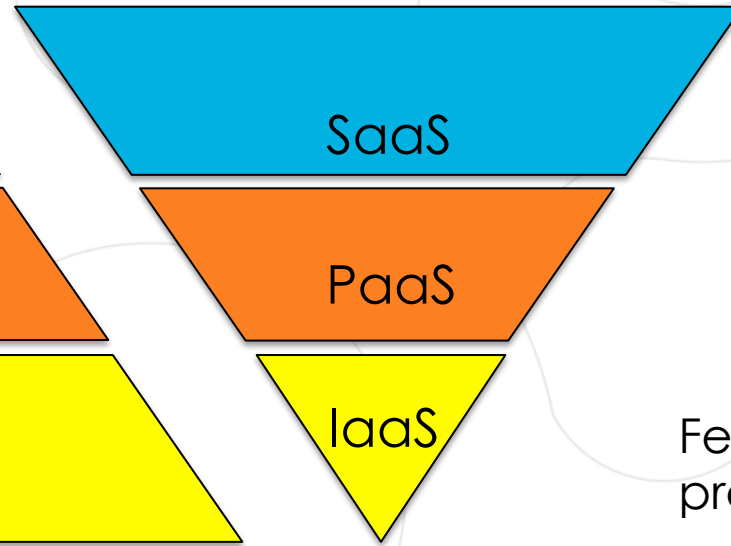
SaaS

Application
Developers

PaaS

Infrastructure
Architects

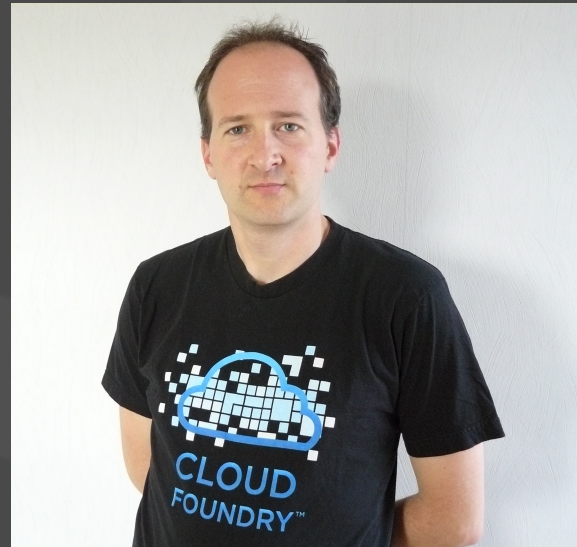
IaaS



Thousands
Applications

A Dozen
platforms

Few world-scale
providers



jeecon

Cloud

easy

On premises

refactor

May be difficult
for legacy
apps

Green field applications

- Can select **modern** solutions
most (not all) frameworks are well
designed for **Cloud**
 - Started on cloud,
discovers and **adapts** to constraints
- mix-it.fr
- cfp.breizhcamp.org

« Classic » applications

- Not such **modern** solutions
common frameworks may not fit cloud
constraints
- Started on premises, single server
hit cloud constraints

The Devoxx apps

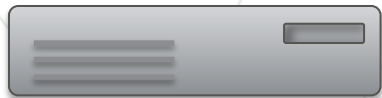
- Call for Paper and Registration
- High traffic two months a year
- Self hosted (parleys.com infra)
moved to CloudBees PaaS
- Wicket, Spring, MySQL
No hype, like 99% java applications

Scale out



Scale up

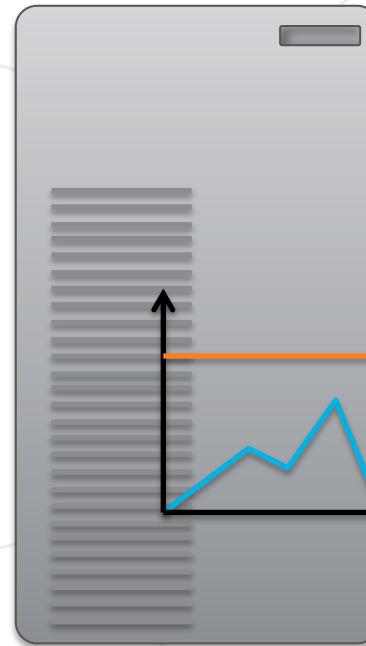
- Still possible, but will hit a **limit**
- Not optimal



M1 small



M1 large



M3 double extra large

M1 extra large

- 64 bits
- 15 Gb memory
- 4 Vcore, 2 cpu unit (~2.5Gh)
- 1.6 Tb HD

aka « my personnal laptop »

Scale out

Design for **clustering**

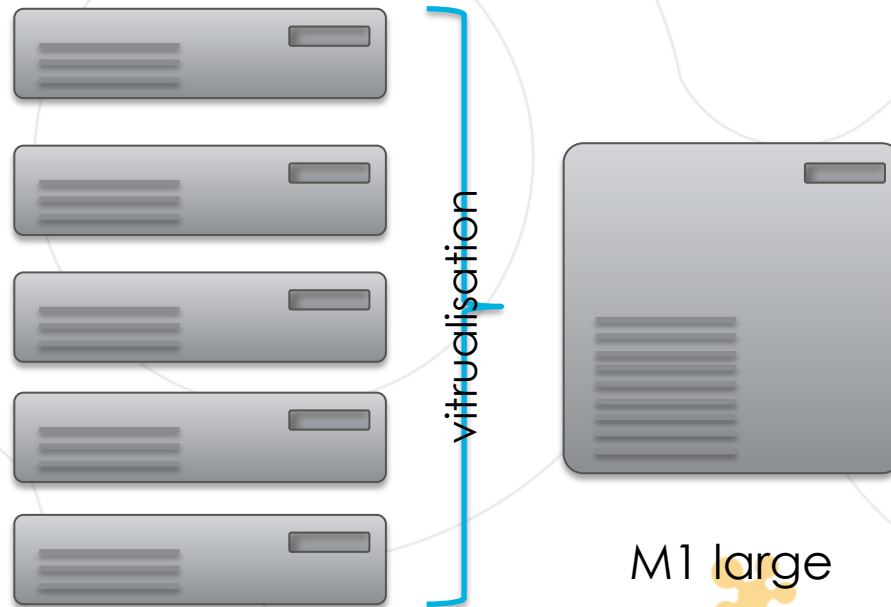
Adapt resources to actual **load**



Pay only for actual **use**

What a PaaS actually does ...

- slice a server into cells
- Multi-tenant app hosting



Multi-tenancy

- Hardware level (IaaS)
- OS level (hypervisor)
- OS virtualization (cgroups, LXC)
- Middleware ?

Java 9 to be multi-tenant ?

be State
Less



... if you can

RESTFul, Stateless

- Linear horizontal scaling

But ...

- Application - User « conversation » has a state
- Stateless apps mostly use caches then **cache** needs to be **distributed**

Stateless, really ?

- Beware your frameworks !

Grails

security plugin

Spring-security

HttpSession

Does Stateless really exist ?

- Client side state with **browser cookies**
→ higher network traffic, security
- Use **http session** (servlet frameworks)
→ memcache session replication
→ sticky session
- Use a **central service** (DB)
→ SPoF, DB scalability

beware LOCK-IN

Prefer portable API



Standards

- Use standard, portable APIs
(aka *Java EE, the good parts*)
- Set runtime configuration via
env variables / system properties
- at least use some abstraction to
insulate vendor-specific code

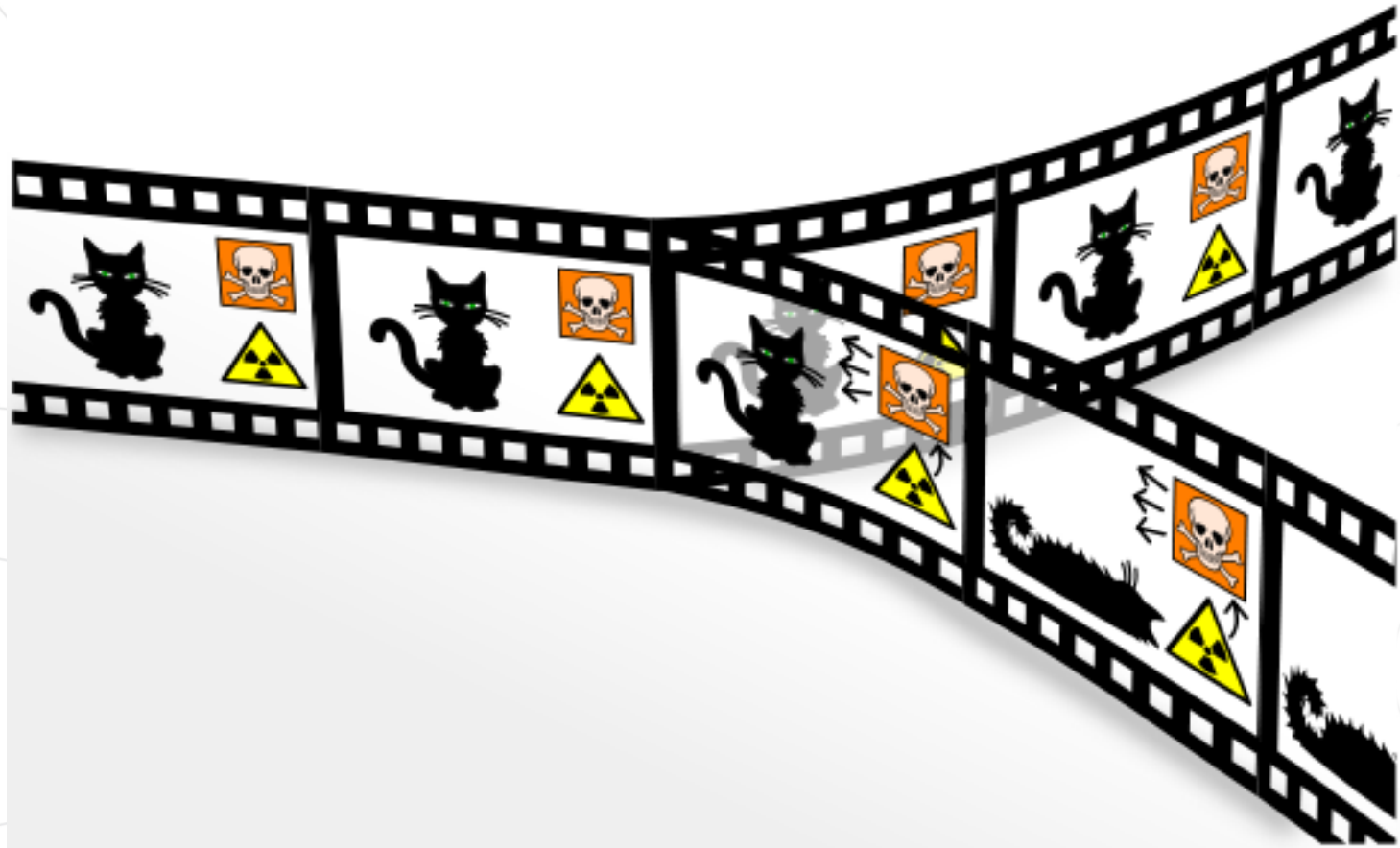
Some Standards

- Java EE
- Java Servlet
- JVM
- LAMP
- Node.js
- RVM
- ..

Beware File System



Cloud uses Schrödinger FS

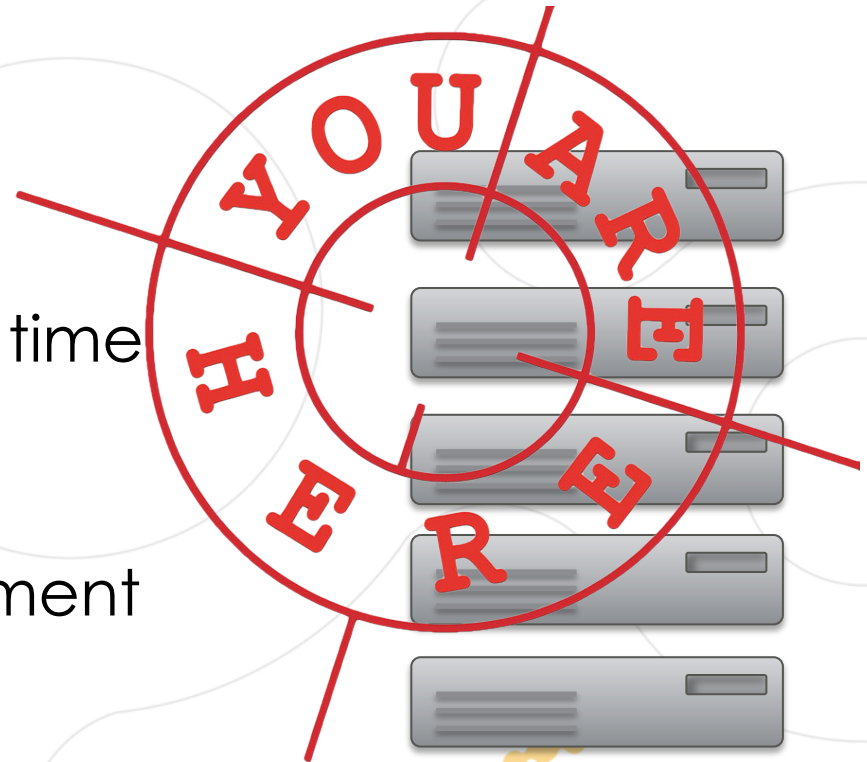


Looks like it's **alive**, but it's **not**

PaaS == pool of servers

Your host at this time

Your host after (re)deployment



FileSystem is ephemeral (and not distributed)

Use store engine (S3, DB Blob, ...)

Use fileSystem wrappers (s3fs)

Use ephemeral as a cache

Singletons ... aren't



Sample: Quartz Job Scheduler

Job will be triggered on **all** nodes !

→ Use Persistent (JDBC) Job Store

→ Use dedicated cron service



Latency

matters

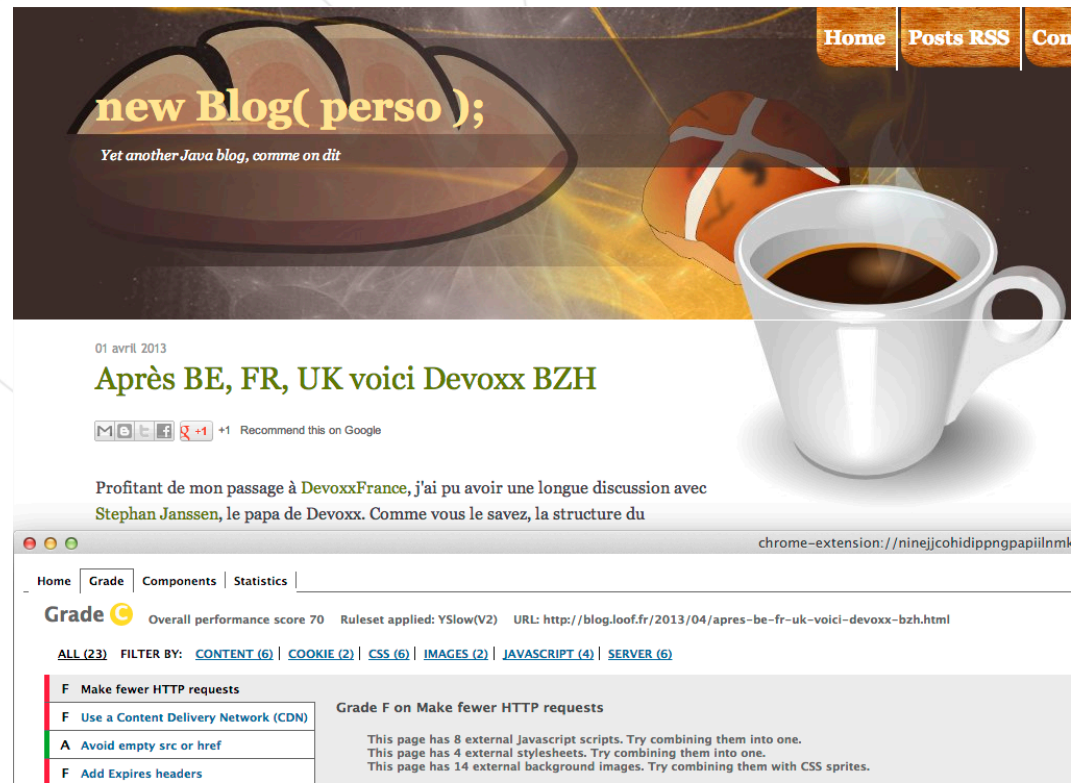




● AWS Regions ○ Amazon Edge Locations (CloudFront & Route 53)

Measure

- Chrome DEV Tools
- Google Pagespeed
- YSlow



The screenshot shows a web browser window displaying a blog post. The blog post has a header with the text "new Blog(perso);" and a sub-header "Yet another Java blog, comme on dit". The main content of the post is "Après BE, FR, UK voici Devovx BZH". Below the post content, there are social media sharing icons and a "Recommend this on Google" button. The browser's address bar shows the URL "http://blog.loof.fr/2013/04/apres-be-fr-uk-voici-devovx-bzh.html".

Overlaid on the bottom of the browser window is a performance analysis tool (YSlow) showing a "Grade C" overall performance score. The tool lists several performance issues:

- F** Make fewer HTTP requests
- F** Use a Content Delivery Network (CDN)
- A** Avoid empty src or href
- F** Add Expires headers

The tool also provides additional information: "This page has 8 external Javascript scripts. Try combining them into one." and "This page has 4 external stylesheets. Try combining them into one." and "This page has 14 external background images. Try combining them with CSS sprites."

Improve

- Use HTTP cache **headers**
- Use **unique path** per deployment hash, or just `?version=`
- Use a **CDN**

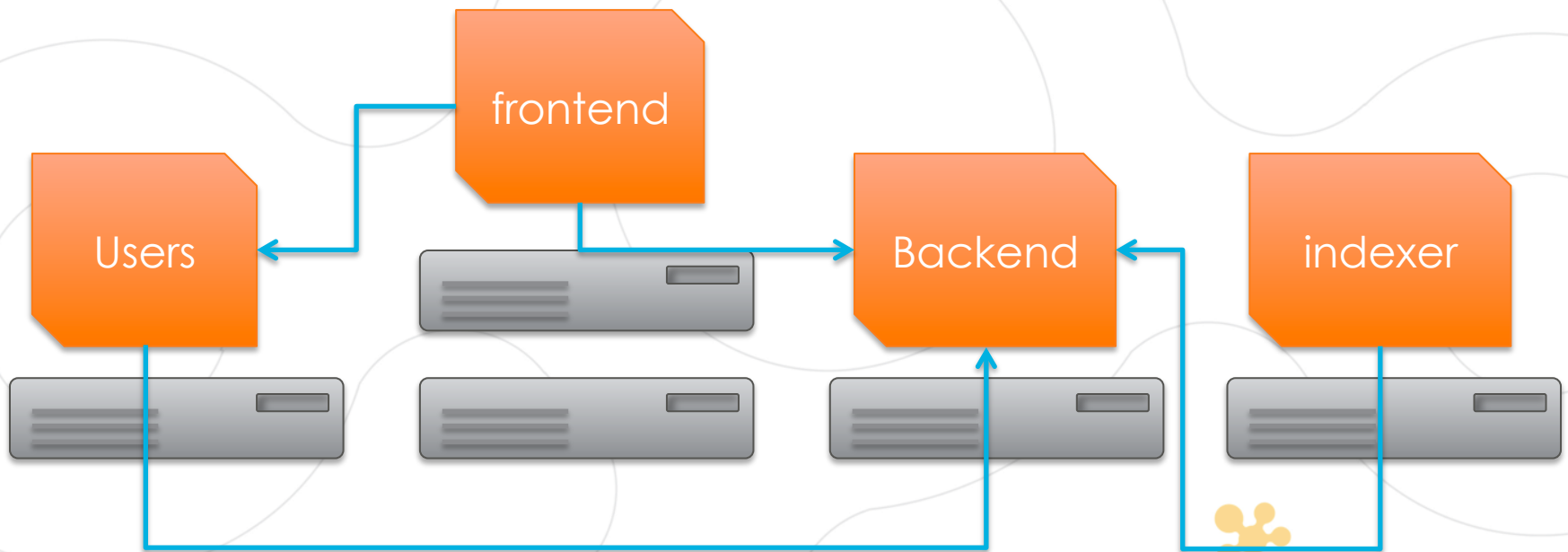
SOA

(the right way)



Small is beautiful

- small, specialized, elastic services
- Communicate with REST on HTTP/MQ



Consume Services



*aaS ecosystem

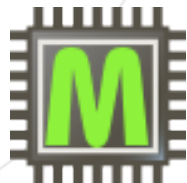
- *aaS is about **service**, not **software**
- **Integrate** services, don't try to setup your own **infrastructure**
- AWS, the place to be for *aaS



CLOUDANT

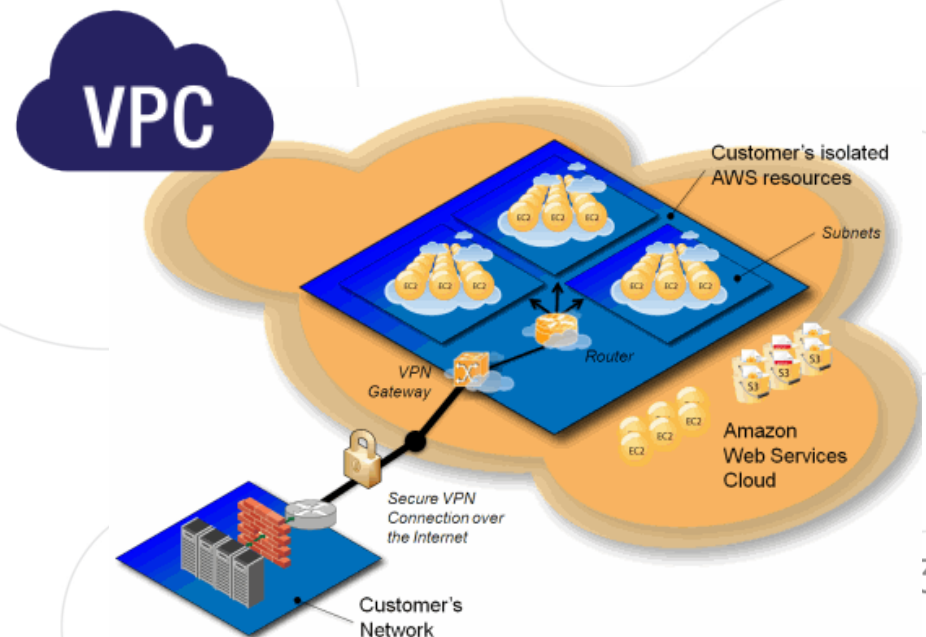


SendGrid
Email Delivery. Simplified.



Private cloud is non-sense

- Do you produce *your own electricity* ?
- Security is about humans, not firewalls



Design for Failure



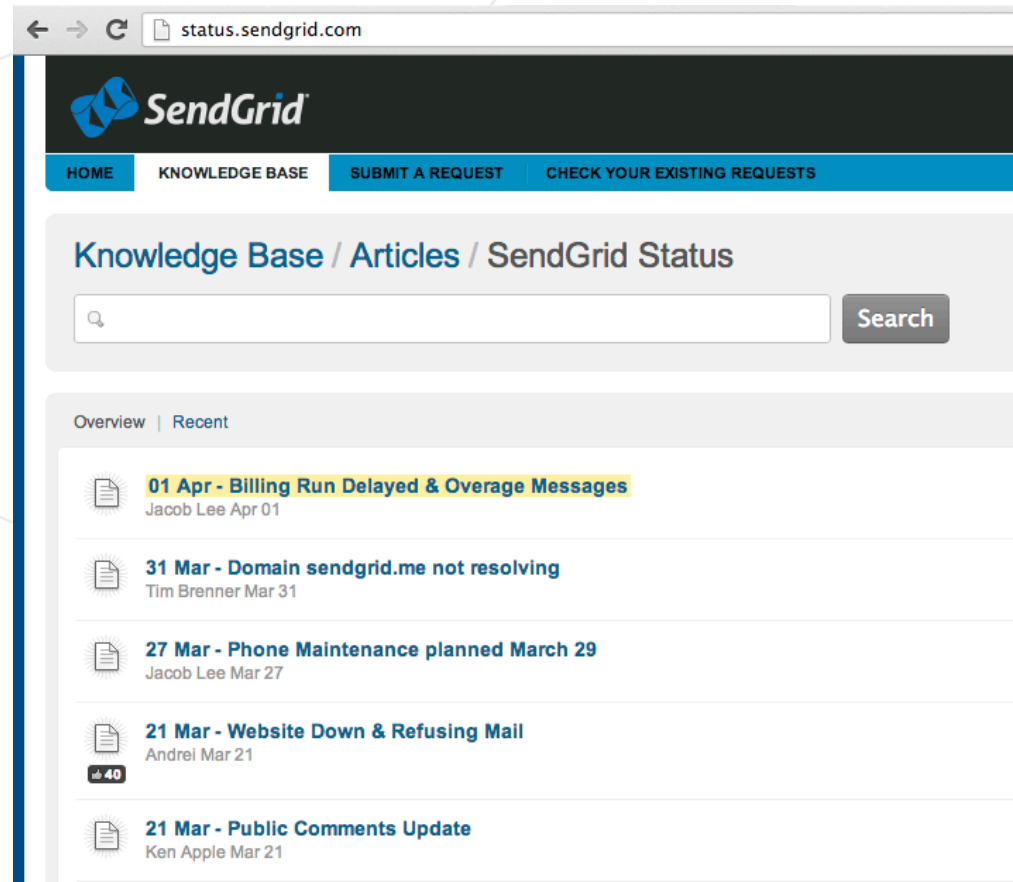
It ~~May~~ Will fail

The screenshot shows a dashboard titled "CloudBees" with a navigation menu icon in the top right. The main content area features a large status card on the left and three smaller status cards on the right. Each card includes a thermometer icon, a title, a status message, and a timestamp.

Component	Status	Timestamp
CloudBees Platform Status	All systems operating correctly	13:30 UTC - Mon, Mar 25 - 20 days ago
Authentication	All systems operating correctly	13:30 UTC - Mon, Mar 25 - 20 days ago
DEV@cloud	All systems operating correctly	15:57 UTC - Mon, Mar 25 - 20 days ago
RUN@cloud	All systems operating correctly	13:30 UTC - Mon, Mar 25 - 20 days ago

Beware resource

- Don't hang the app when resource fails
- Run asynch
- Use MQ



The screenshot shows a web browser window with the URL `status.sendgrid.com`. The page features the SendGrid logo and a navigation bar with links for HOME, KNOWLEDGE BASE, SUBMIT A REQUEST, and CHECK YOUR EXISTING REQUESTS. The main content area is titled "Knowledge Base / Articles / SendGrid Status" and includes a search bar. Below the search bar, there are tabs for "Overview" and "Recent". A list of recent status articles is displayed, each with a document icon, a title, and the author's name and date. The first article is highlighted in yellow.

Article Title	Author	Date
01 Apr - Billing Run Delayed & Overage Messages	Jacob Lee	Apr 01
31 Mar - Domain sendgrid.me not resolving	Tim Brenner	Mar 31
27 Mar - Phone Maintenance planned March 29	Jacob Lee	Mar 27
21 Mar - Website Down & Refusing Mail	Andrei	Mar 21
21 Mar - Public Comments Update	Ken Apple	Mar 21

is Cloud
Safer

anyway



Some metrics

- In 2012, CloudBees suffered 2 major outages 20 then 10 minutes

→ 99,99% (What's your actual availability rate ?)

Cloud outages are **visible**

Disaster recovery

- All deployed artifact → S3
 - DB on EBS, then daily → S3
- + your own backup strategy

<http://wiki.cloudbees.com/bin/view/Documentation/BackupPolicies>

Need **more** ?

- Multi-**zone** High-Availability
- Mutli-**region** redundancy
 - data sync to handle network latency
 - Short TTL DNS

No « **turn key** » 

Ops for Cloud



Infra is managed ... not app

- Need to instrument and monitor



Ops for Cloud apps



One team,
One goal,
One platform

Cloud is the best place to embrace DevOps

	Traditional	Cloud
Environment	DEV / INT / PROD	identical
Delivery	Mostly manual	full automation API based
DEV Process	Fire and forget	Continuous delivery
Team	Dev vs Ops vs QA	DevOps

Continuous

Integration

delivery

deployment  geecon

Continuous ...

Integration

- Git push
- Build
- Test

Deployment

- Git push
- Build
- Test
- Deploy
- Production

Delivery

- Git push
- Build
- Test
- Ready for production
- Production

manage

Resources as code



Manage Database

you can connect tools to your DB

The screenshot shows the MySQL Workbench interface. The SQL Editor window is titled "SQL Editor (devoxx-cfp-production)" and contains the following SQL query:

```
select * from presentation where title = "cloud patterns"
```

The results pane shows a table with the following columns: id, created_on, modified_on, uuid, version, and audience. The table is currently empty, with all cells containing NULL.

The Action Output pane shows the following results:

	Time	Action	Response	Duration / Fetch Time
✓ 1	15:12:48	select * from presentation where title = "clo...	0 row(s) returned	0.202 sec / 0.001 sec
✓ 2	15:13:05	select * from presentation where title like "c...	0 row(s) returned	0.406 sec / 0.000 sec
✓ 3	15:13:16	select * from presentation where title like "*...	0 row(s) returned	0.596 sec / 0.000 sec

The status bar at the bottom indicates "Query Completed".

But how to ...

- Replicate environment
- Manage DB upgrades
- Tag versions (and DB schema)
- ..

All in SCM

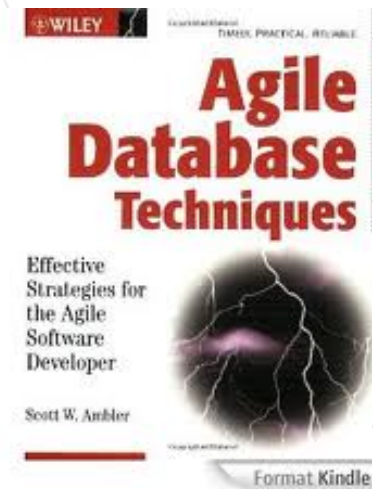
- Application configuration
- Runtime properties
- Resources binding
- Database scripts...

App deployment is 100% **reproducible**

Use agile DB tools



LIQUI  BASE



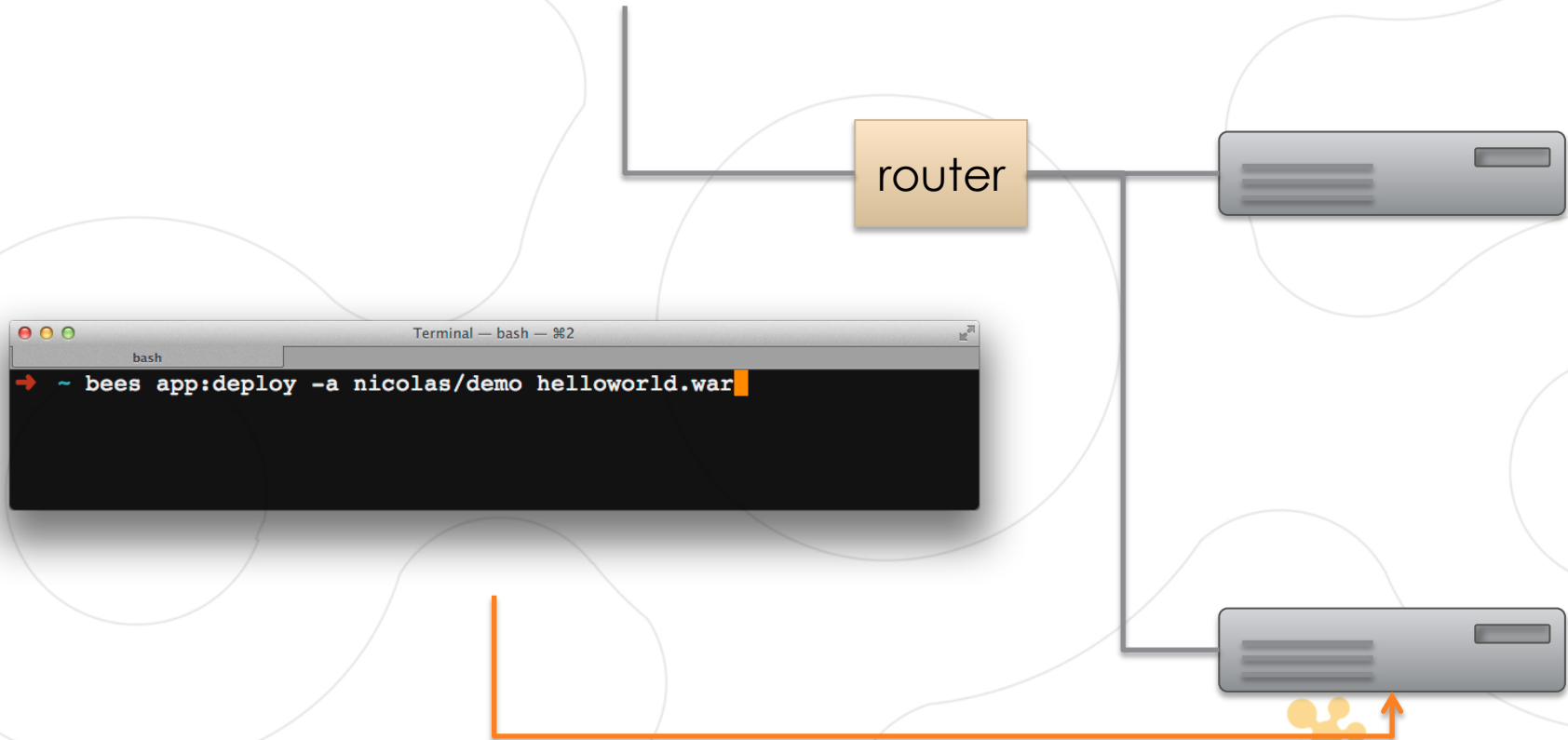
concurrent deployment

0 downtime

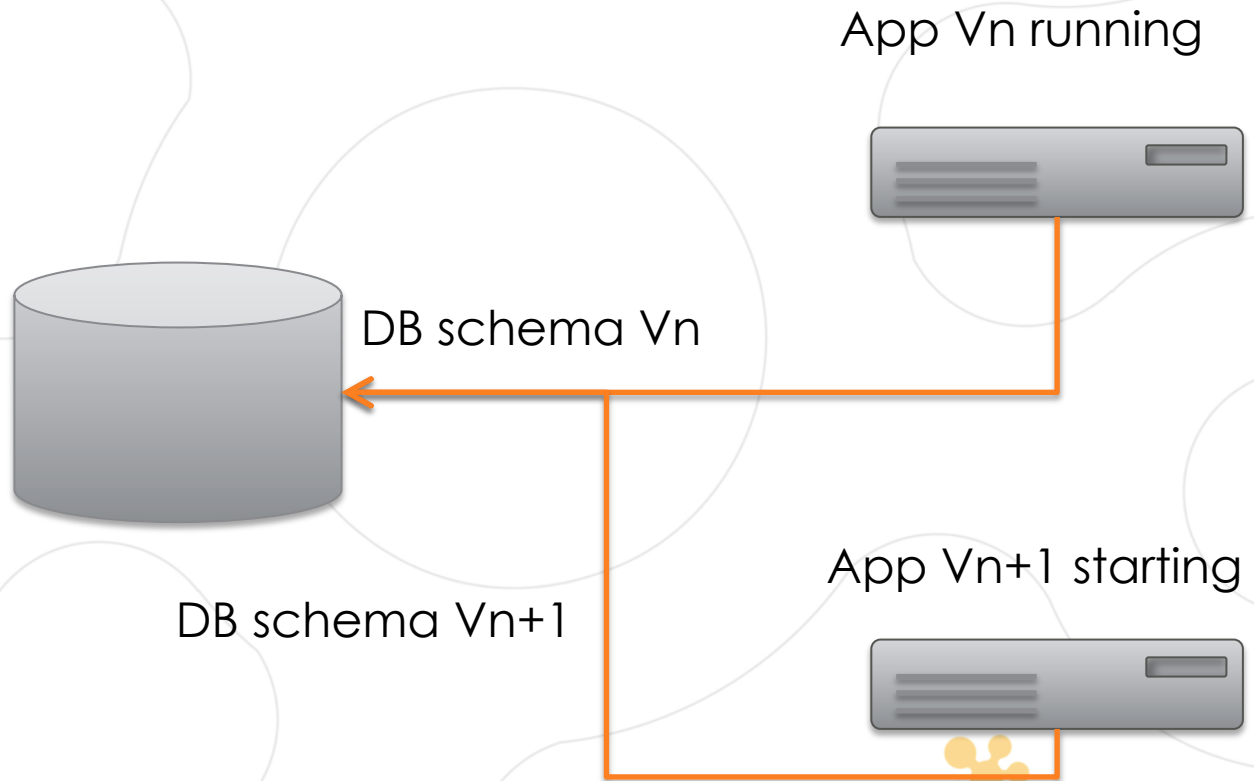


0 downtime

<http://demo.nicolas.cloudbees.net>



Beware resources migration



- V_{n+1} schema to be V_n compatible
- V_{n+2} can do some cleanup
i.e. « **@deprecated** » for DB

Or ...

temporary deploy a « maintenance » page



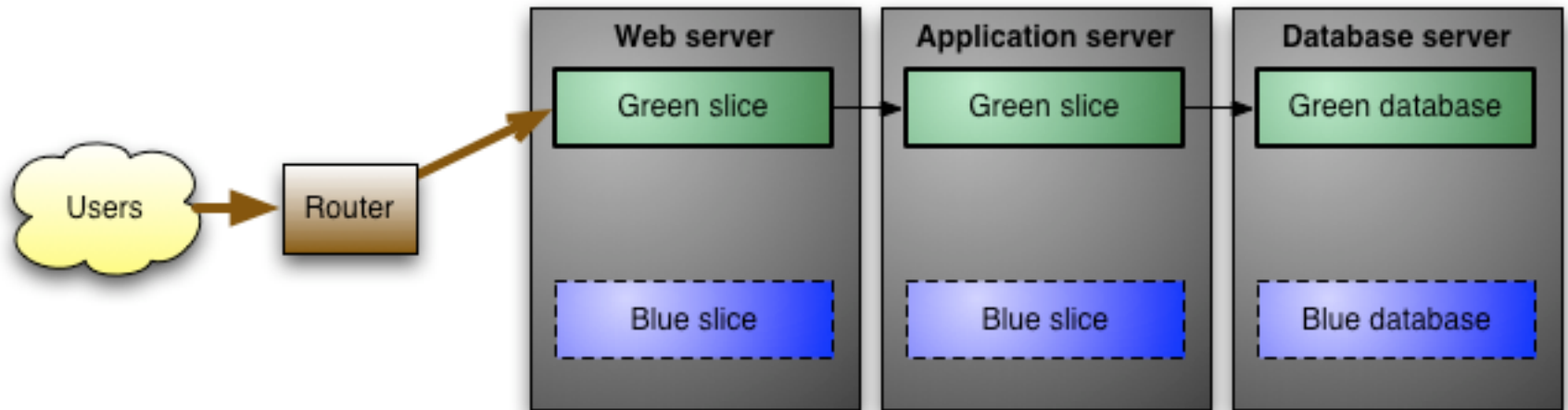
Deployment Is not an event

anymore



Green / Blue

<http://martinfowler.com/bliki/BlueGreenDeployment.html>



```
Terminal — bash — %2
bash
➔ ~ bees app:proxy:update -a acme-maintenance -al www.acme.com
```

thank
YOU

Questions ?

