From Monolith to Services at Scale

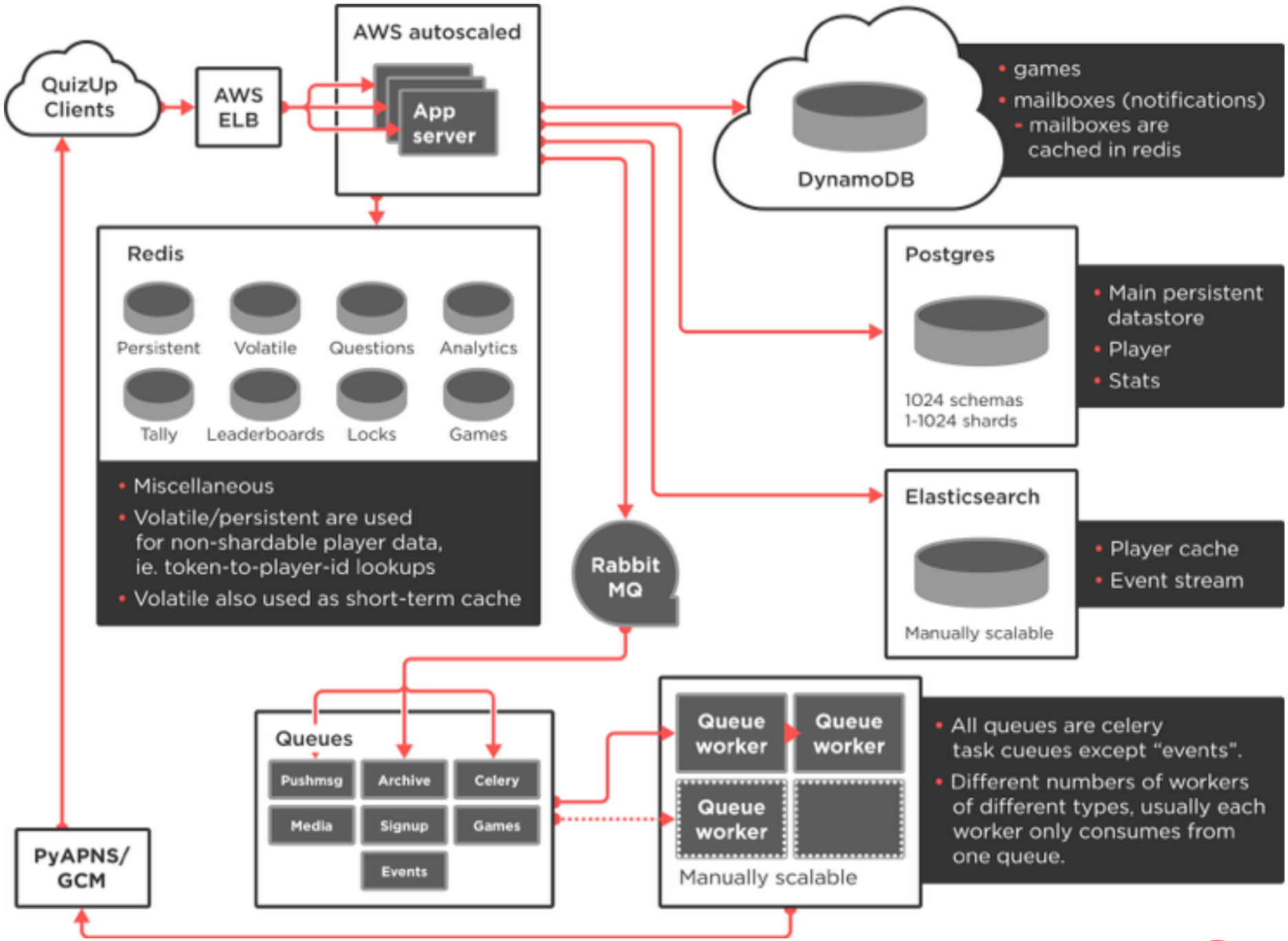
How QuizUp is making the (inevitable?) transition, one endpoint at a time

Steinn Eldjárn Sigurðarson

Reykjavík February 6th 2015

QuizUp

Original Architecture



QuizUp

App Server Monolith

QuizUp

Problems?

- **inefficient**:
  - deployment queues
  - request load variability
  - wasted infrastructure
- **scary**:
  - deployment mistakes = QuizUp is down!
  - long/slow deployments (20-50 app servers)

Solution?

(micro)services

(micro)service benefits

- efficient
  - homogeneous request load profiles
    = easy capacity planning
    = more efficient infrastructure
- logic isolation
- no deployment queues = faster iteration

(micro)service benefits

- flexibility
  - rewrite while maintaining ext. interfaces
  - route by path/client/version
  - legacy support = multiple services,
    not code branching all the time
- reliability (bulkheading, circuit-breaking)
- ... more

Pitfalls?

- discovery
- routing
- monitoring
- failure tracking
- “service ready”-checklist
- → **needs more complex infrastructure**

Solution Components: ZooRunner

- process wrapper
- can health check
- registers child in ZooKeeper:
  - zk://services/<child>
- dies on child death
- services are less tightly integrated with zookeeper
- more reliable than sidecar, more fragile too

Solution Components: NGiNX

- fast, reliable
- developer experience
- clean, friendly codebase
- custom modules:
  - accounting (metrics)
  - authentication (lua)

Solution Components: EIP Manager

- reliability of non-ELB solution?
- X AWS Elastic IPs (fixed)
- NGiNX run and registered via ZooRunner
- More routers than IPs
- Extra standby router claims IP if unused

Solution Components: Router Manager

- watches zk://routes/*
- zk://services/*
- routes are manually configured (for now)
- zk://routes/collections =

```
{"service": "topics", "session_required": false, "locations": ["/collections"],
"https": false, "default_server": “localhost:8888"}
```

- finds service nodes, generates NGiNX config for routing (location + upstreams)

Solution Components: Docker

- tools and services to reliably build and run Linux containers
- not just hype!
- feels like building a **huge** binary
- .. which is good!

Solution Components: Docker (cont.)

- standardized deliverables across stacks
- run unit tests inside production “binary”
- perfect for complex integration tests
- lighter than VMs
- portable between local machines, cloud and different providers!

Solution Components: Docker Registries

- Once built, dockers must be stored somewhere
- registry in each location (office, dev DC, prod DC)
- CI builds and pushes
  - all dockers tagged with githash
  - tagged „stable“ @ deploy time

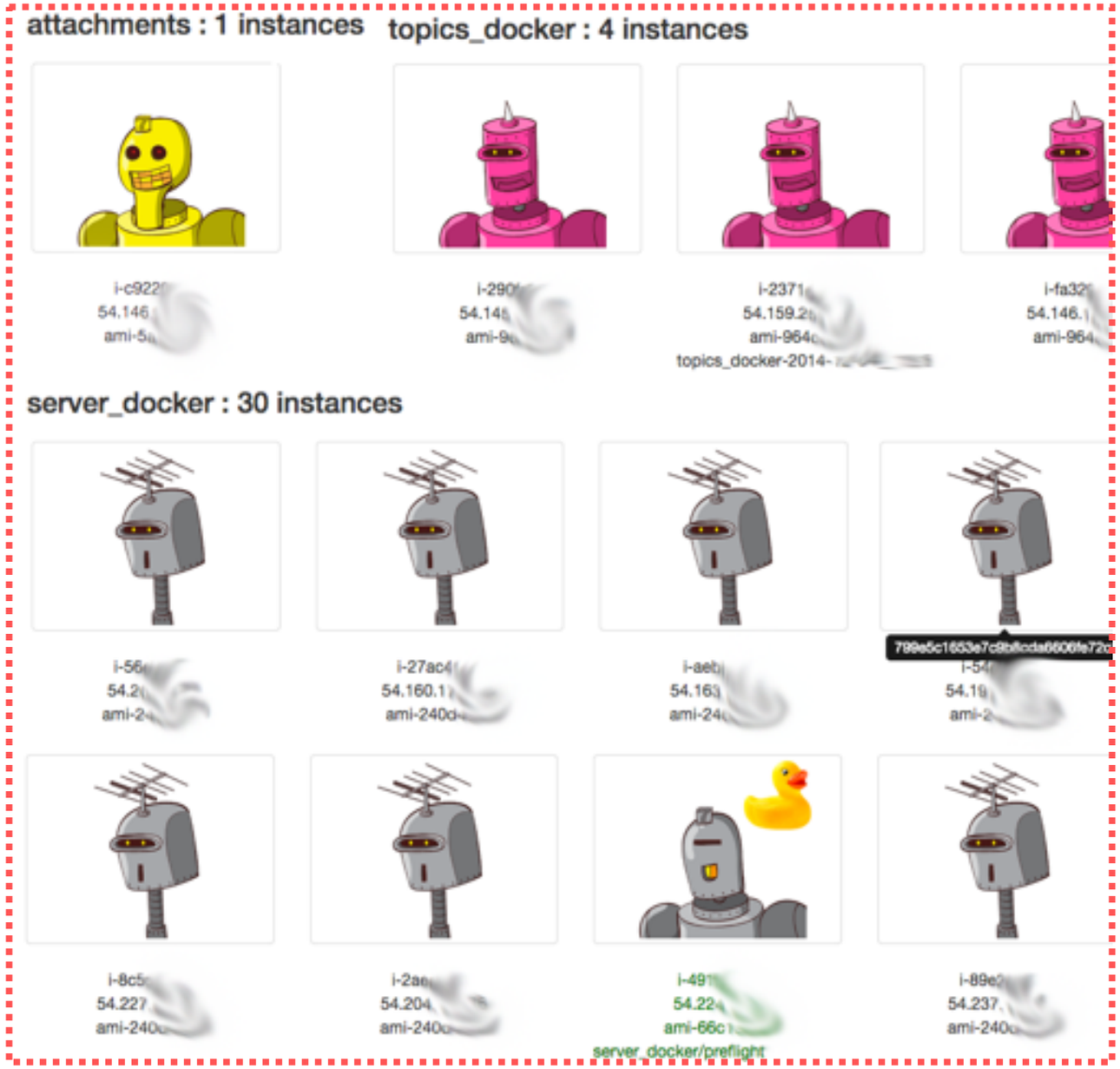
Solution Components: Harbourmaster

- multiple services
- multiple images
- multiple commits
- what's where?
- lists now, perhaps more in the future

```
(venv)ses: ~/w/harbourmaster (master) $ hbm list server
** Listing project: server
** Using registry: https://dockistry.production.quizup.com/
project    latest_githash    author                      timestamp             has_docker    is_deployed  deployed_docker
---------  ----------------  --------------------------  --------------------  ------------  -------------  -------------------------
server     799e5c1           gunnar.kristjans@gmail.com  2015-01-29T16:11:29Z             1              1  799e5c1 (image: 4c7340f)
```

Solution Components: “Robots”

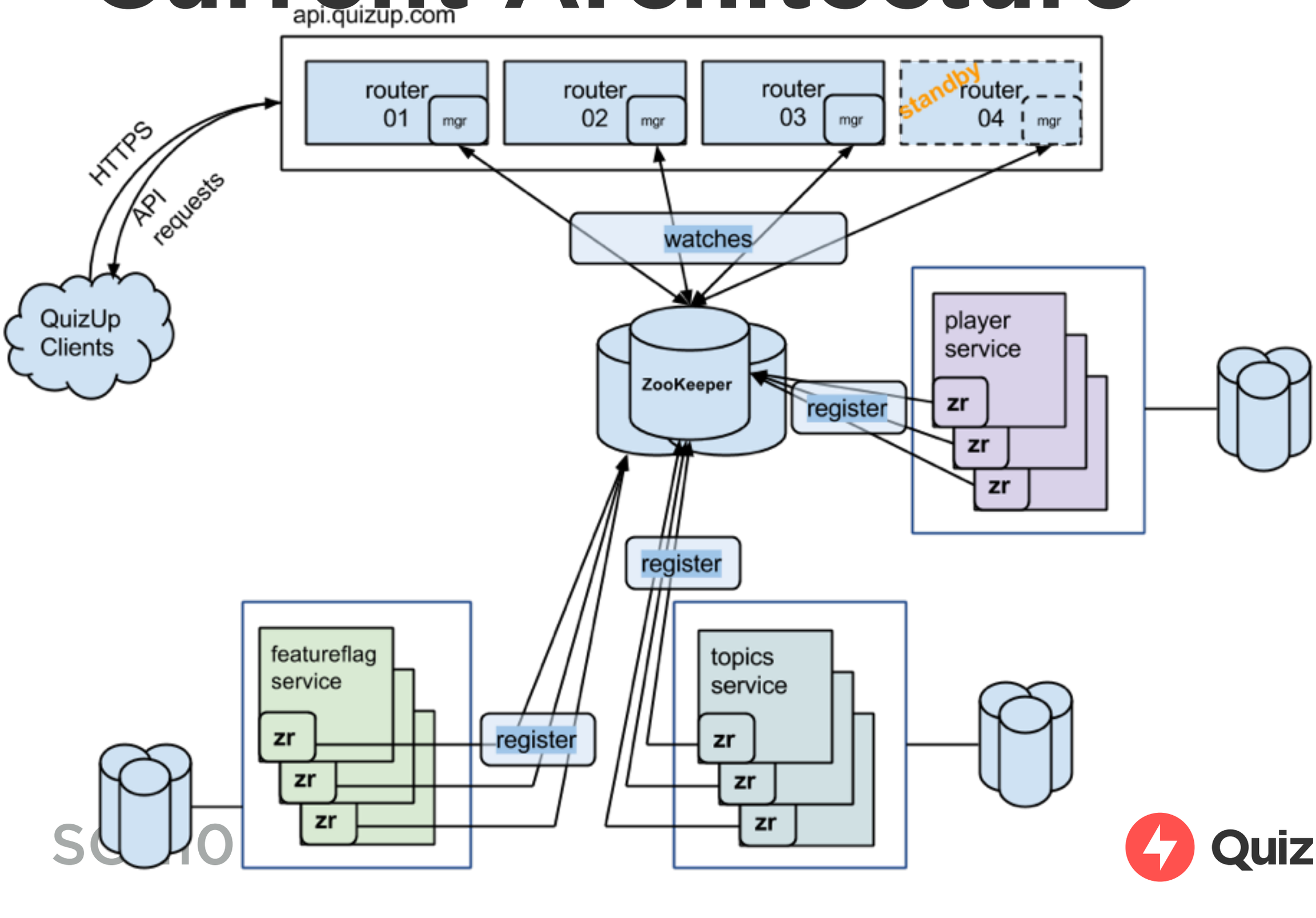
- multiple services
- multiple docker hosts
- multiple revisions
- … hard to spot inconsistencies?



SOL9

QuizUp

Current Architecture



QuizUp

Current Architecture

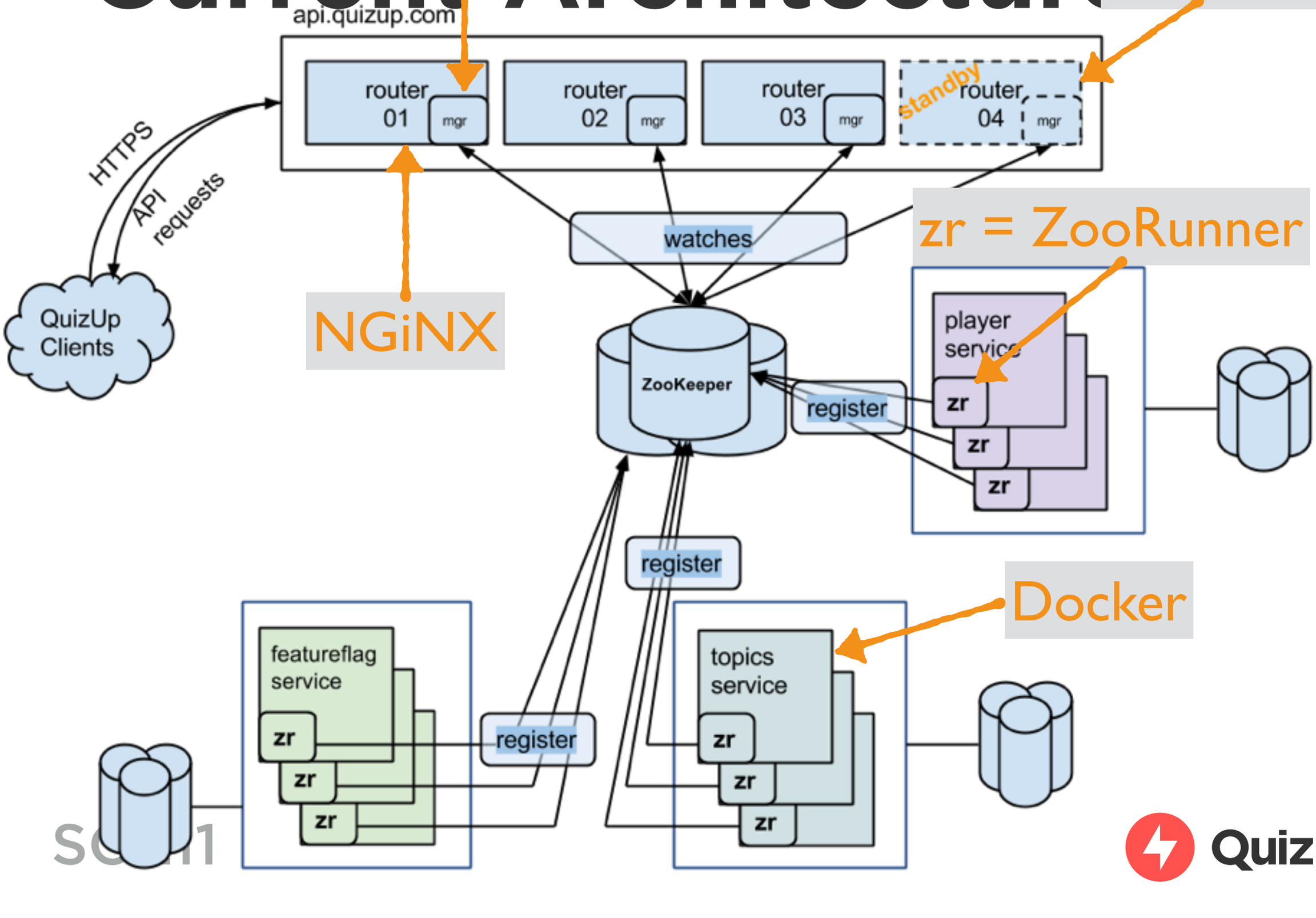
Router Manager

EIP Manager

zr = ZooRunner

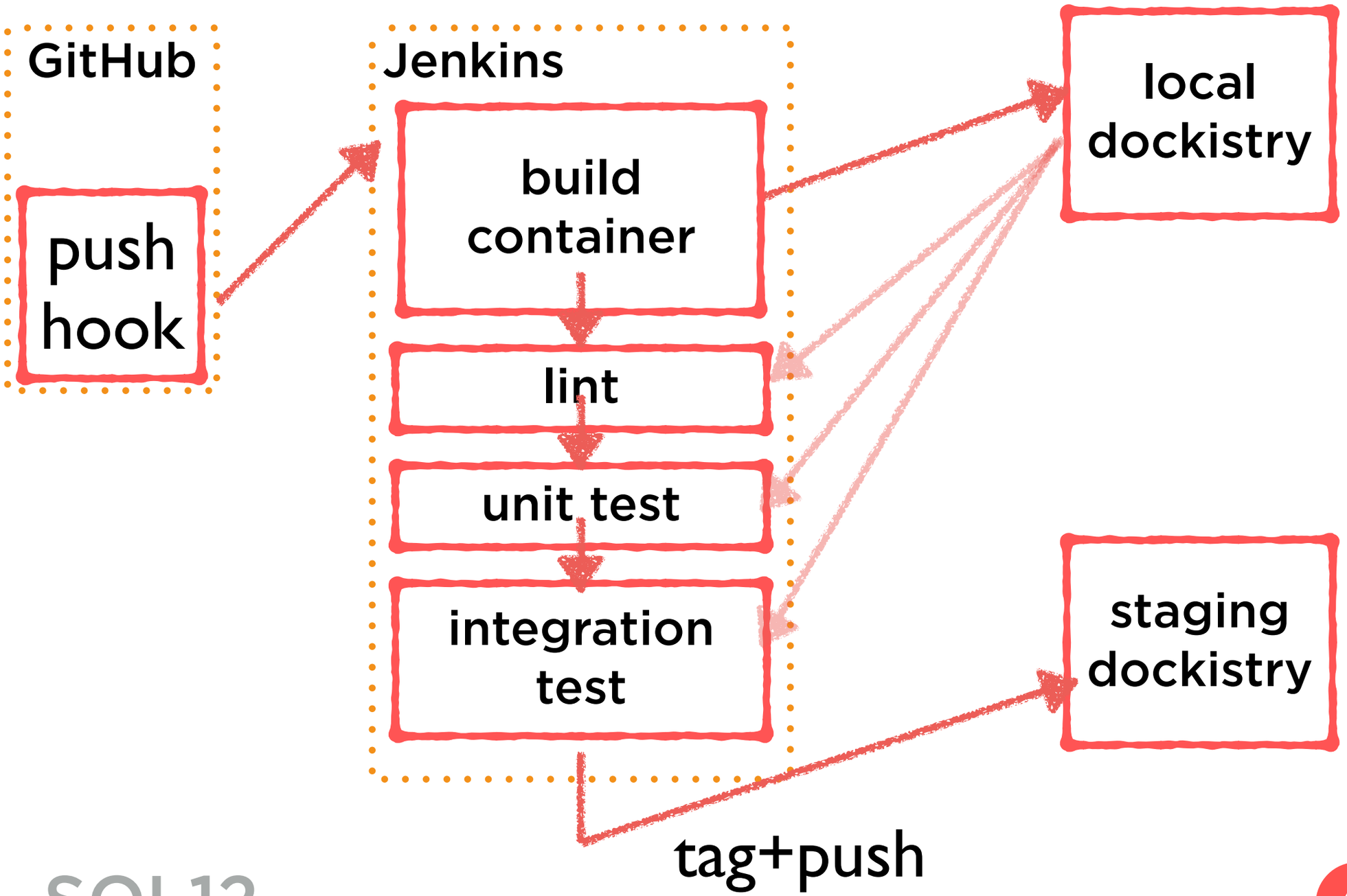
NGiNX

Docker



QuizUp

CI Pipeline



Benefits? Next steps?

- + team autonomy
- + development speed
- + performance

- 10+ services, 5+ in development

- ? central eventbus / message queue
- ? standardize stacks

QuizUp

Lessons learned? (so far!)

- it’s hard to avoid re-inventing the wheel
- gradual changes are key
- small, simple components
- keep watch of new developments
- productionization checklist

Thank You!

questions?

Reykjavík February 6th 2015

QuizUp