



Elastic Cloud Architecture

Morten Ingebrigtsen, Igor Kupczyński

Agenda



- Introduction & demo
- Dive into architecture
- Use case: recommendations with Graph
- Q&A





Introduction



Demo



Architecture

Architecture — key features

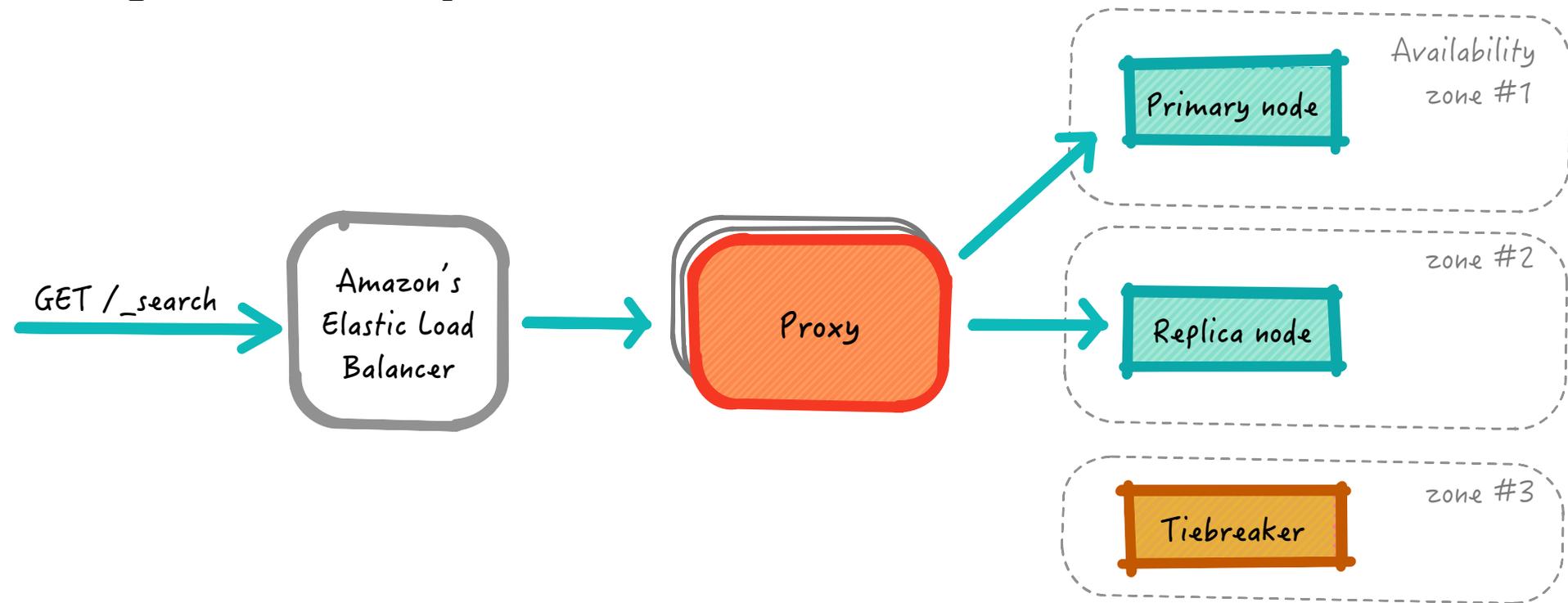
- Services
- AWS infrastructure
 - Regions and availability zones
 - EC2 — virtual servers (i2) 
 - Elastic Load Balancer — front proxy / load balancer 
 - S3 — backup and binary storage 
 - RDS — PostgreSQL 
- Docker all-the-things
- Zookeeper — state of the system + coordination

Lets talk numbers

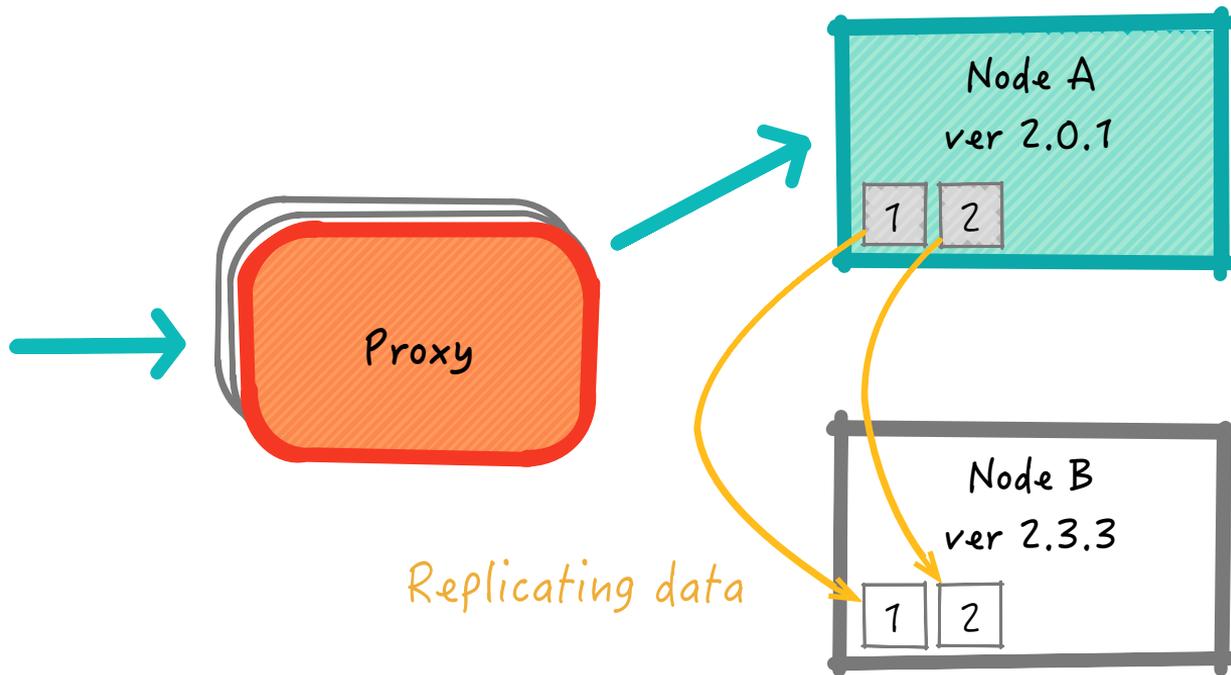
- 7 AWS regions
- **Thousands** of active elasticsearch clusters
- **Billions** of elasticsearch requests per day

- **15** team members
 - both SREs and devs, with some overlap in responsibilities
 - it started with 4

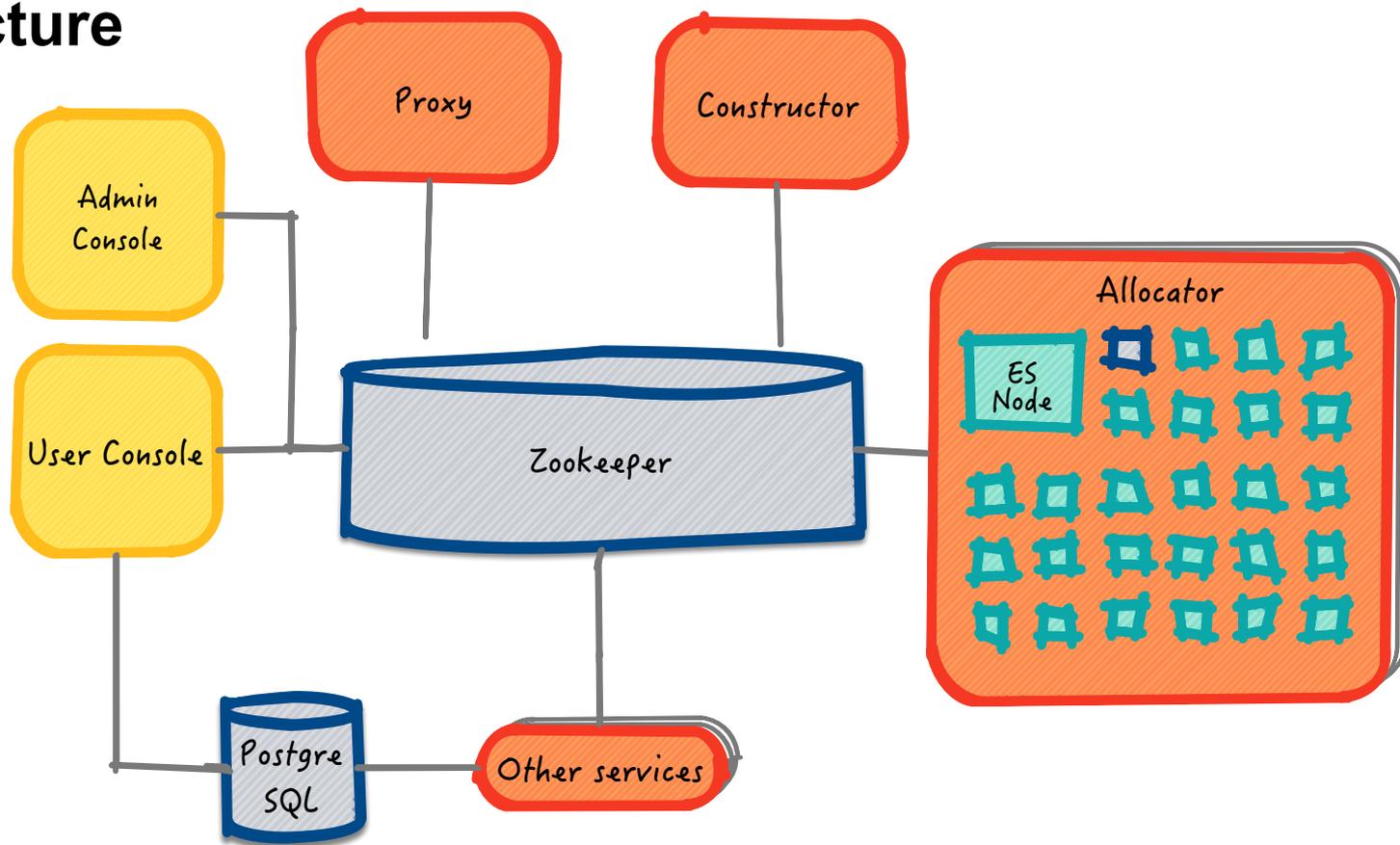
High availability



No-downtime scaling & upgrades



Architecture

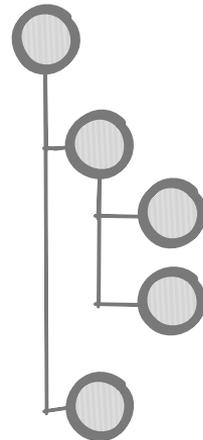


Zookeeper



- Distributed data store for coordination and storing the global state
- Filesystem tree-like structure — znodes
- **CAP** — consistent in the event of network partitions
- Fast **reads**, but only simple queries are possible
- **Writes** need ACK from a quorum, so slower; data should fit in memory
- Writes are linear — all clients see them in the same order
- Possible to set **watches**

- Elastic Cloud — zookeeper stores the state of the whole system





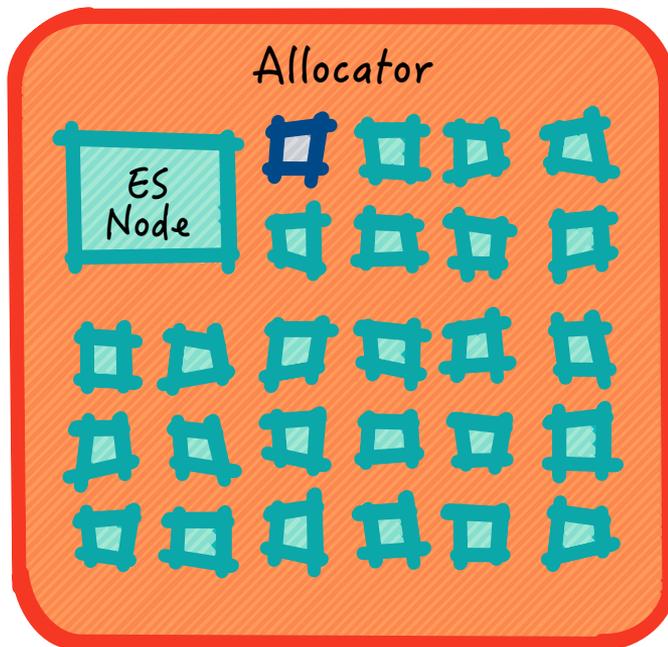
***Use Zookeeper.** It's mature, well-designed, and battle-tested. Because the consequences of its connection model and linearizability properties are subtle, you should, wherever possible, **take advantage of tested recipes and client libraries like Curator**, which do their best to correctly handle the complex state transitions associated with session and connection loss.*

<https://aphyr.com/posts/291-jepsen-zookeeper>

Kyle Kingsbury a.k.a. **Aphyr**

Allocator & elasticsearch nodes

- Manages elasticsearch nodes
- All allocators in a region form an allocator pool
- Advertises its capacity and free resources in zookeeper
- Each node is a docker container
 - Guarantees memory, CPU, IOPS, disk space
 - Isolates the nodes



Constructor



- The brains behind **allocator**
- Calculates what needs to be changed when a cluster is added or reconfigured
 - Monitors new requests from user & admin console
 - Writes to a znode
 - Allocator watches it and acts
- Assigns new clusters to a proper allocator

Security

- Stunnels between containers
- znode ACLs (each container can access only part of the state important for it)
- Dockerized elasticsearch nodes
- Oauth & Time-based One Time Passwords in the admin console
- Shield (Elastic Security) for all the clusters
- Dedicated service to manage secrets in the cloud
- Backups every 30 mins, retention 48 hours
 - Isolated S3 bucket



Use case: Graph



Go to <https://cloud.elastic.co> and start your trial today!