



# Building Reliability with Spokes

Patrick Reynolds (@piki)  
and the Git Infrastructure team

# Git Infrastructure



Proxies and RPC

- **HTTPS, SSH, SVN**
- **Internal API**



Git file servers



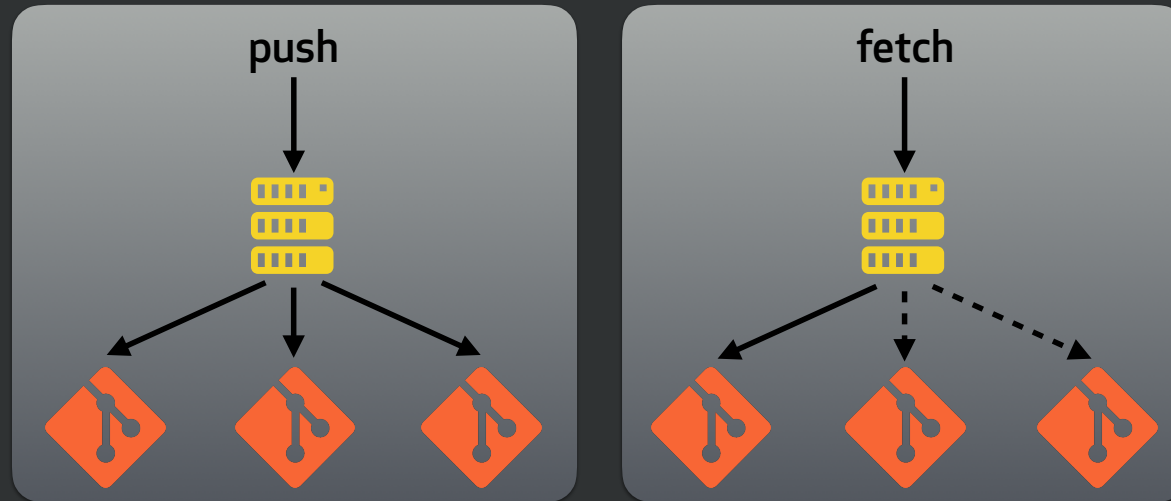
- Run a Git service
- Make it scale
- Make it reliable at scale

# Goals



- One server down = no impact
- Two servers down = minimal impact
- Balance disk space and CPU load
- Automatically resync and repair
- Horizontal scaling
- Maintenance operations = no impact
- GitHub Enterprise, too

# Spokes



- Git-level replication
- Three copies of every repository, wiki, and gist
  - One copy = read-only
  - Two copies = fully functional

# Surviving failures



- **Goal:** one server down = no impact

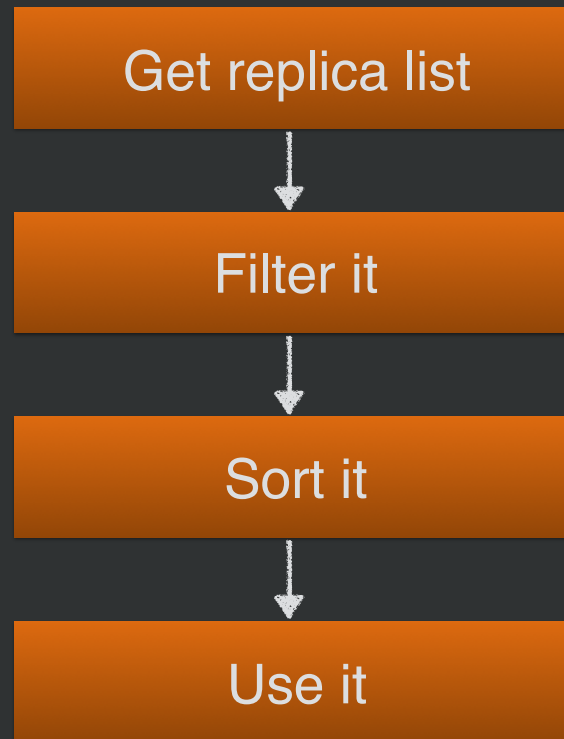
15:28 CST    **Everything operating normally.**

15:20 CST    **We're continuing to investigate the issue. Less than 5 percent of repositories are affected.**

15:03 CST    **We're troubleshooting an issue affecting a small number of repositories.**

- Send every request to the best server

# Picking the best server



# Failure detector



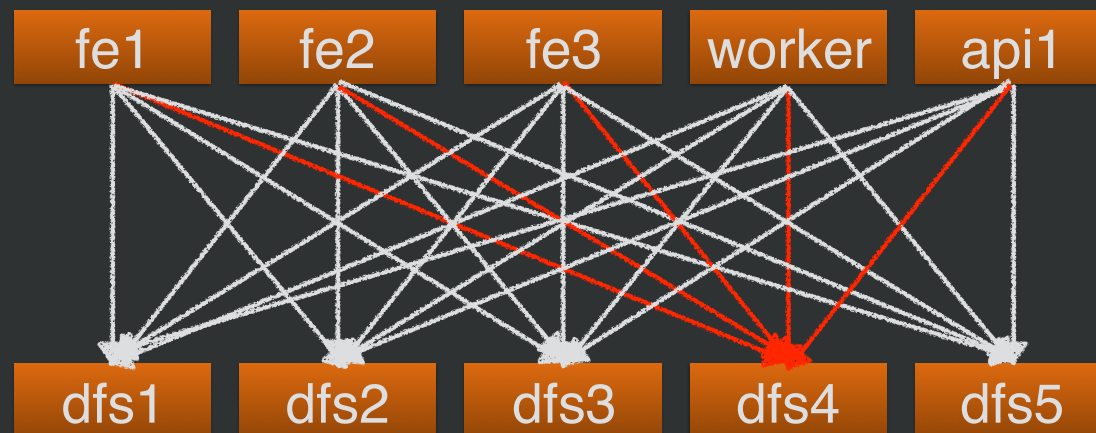
- Watch real traffic
  - Three failures in a row = offline
  - One success = back online
- Heartbeats
- Quick failover



# Failure detector



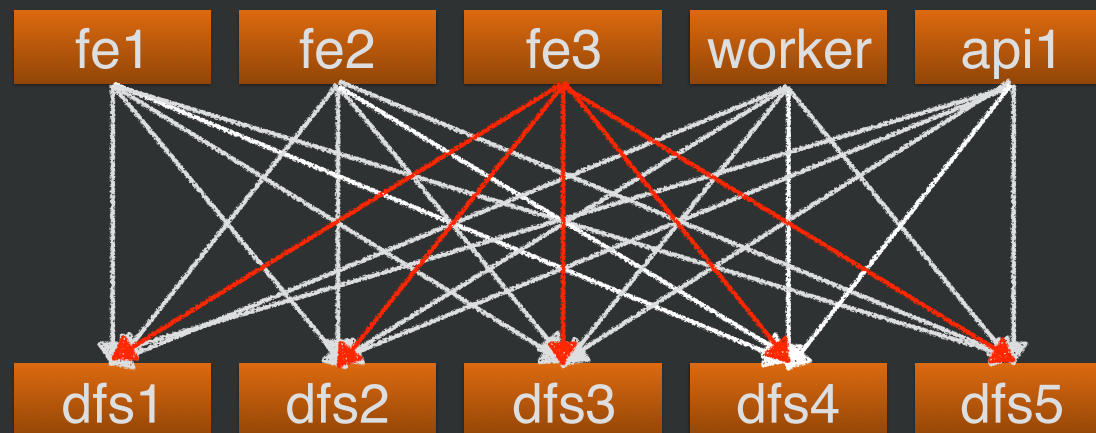
- Per-client list of offline servers



# Failure detector



- Per-client list of offline servers








piki@github-staff1-cp1-prd:~\$ ./find-fewest-clones



🔗 Set up a server | 📄 Add an app or code integration | ➡ Invite others to this private channel

Today

-  piki
-  piki
-  piki
-  piki
-  habot
-  peng

habot

# No missed operations



- **Goal:** one server down = no impact
  - Reads pick a healthy server, fail over
  - Writes go to all healthy servers
- **Goal:** two servers down = minimal impact
  - No impact if in the same rack
  - $\leq 0.01\%$  of repositories are read-only



- **Goal:** balance CPU load
  - Treat busy servers as offline

# Recovery



- What happens after downtime?
- What happens after failure?



# Goals and repairs



- Goals
  - Properties that are supposed to be true
  - Expressed as SQL queries
- Repairs
  - How to make each property true, if it isn't
  - Expressed as Resque jobs

# Goals and repairs





# Goal-based programming



- Decoupled in code
- Decoupled in time
- Reusable



- **Goal:** balance disk space
  - Set balanced disks as a repair goal
  - Move repository replicas as needed
- **Goal:** resync and repair
  - Set unanimous checksums as a repair goal
  - `git fetch` + `rsync` as needed

# Maintenance operations



- **Goal:** maintenance operations = no impact
  - Reboots work like any other server crash
  - Removing a server works like a failure
  - Can we be more graceful?
    - Quiescing and evacuating

# Horizontal scaling



- **Goal:** horizontal scaling
  - Need more CPU or disk space?
  - Add a server
  - There is no step 2



- **Goal:** GitHub Enterprise, too
  - Spokes = file servers for Enterprise Cluster



- Replicate everything
- Deemphasize individual servers
- Application traffic > heartbeats
- Goals > failure modes
- Learn more: [githubengineering.com](https://githubengineering.com)



</talk>