31/10/2024

Embrace fleet reboots and make them boring

Everton Didone Foscarini
SRE - Edge - London
Cloudflare

Summary

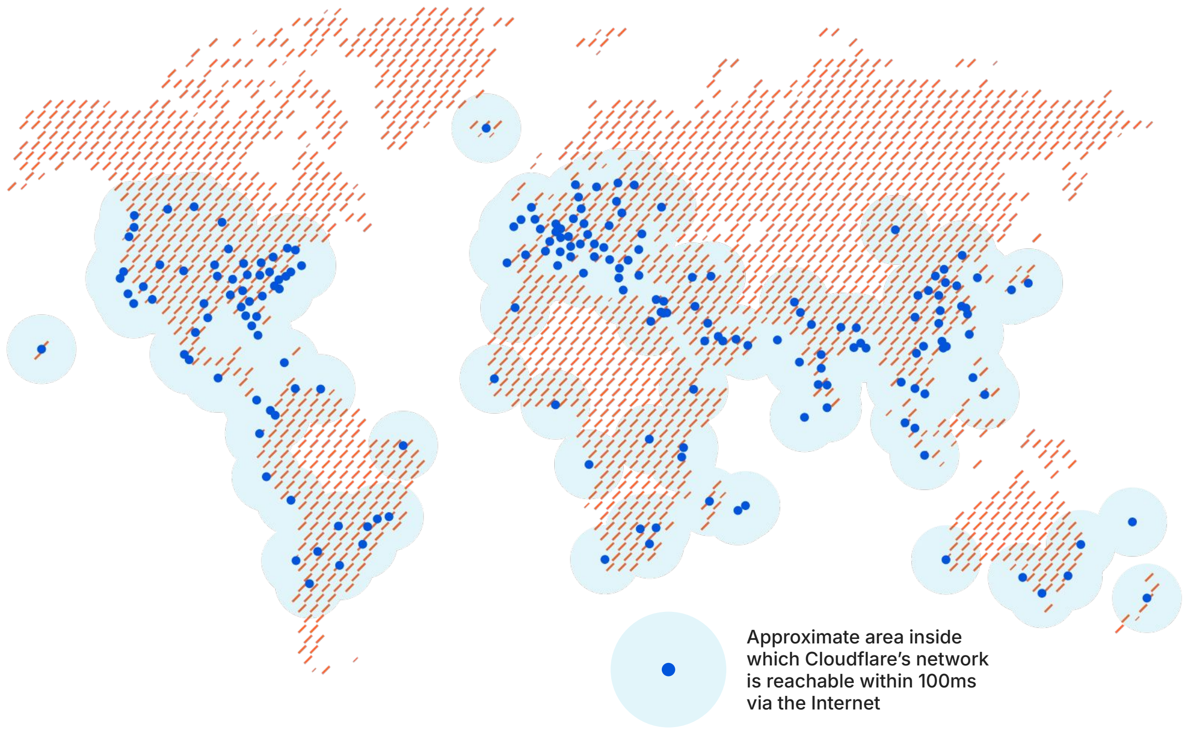
- whoami
- Cloudflare’s network
- Edge vs Core, Edge SRE team, Cloudflare Edge PoPs
- uptime
- Why we reboot servers
- How to reboot all Edge servers
- Tale of a kernel upgrade
- Automate ourselves
- How to reboot all Edge servers (the **boring** version)
- Another kernel upgrade
- Further improvements in Automation and processes
- Takeaways

whoami

- Working with GNU/Linux since 2003
    - Started with LTSP for 20 terminals in a Pentium 4 with 2GB of RAM
- 2008 - 2016 - UFRGS - IT on University in Brazil
    - Apache, nginx, LDAP, Xen, XenServer, dovecot, postfix and much more
- 2017 - Joined Cloudflare
    - Salt, nginx, consul, nomad, building, debugging and understanding our Edge services and clusters

@foscarini@mastodon.social

Cloudflare's network


Approximate area inside which Cloudflare's network is reachable within 100ms via the Internet

330

cities and 120 countries

158B

cyber threats blocked each day in Q2'24

60 million

HTTP requests per second on average in Q2'24

39 million

DNS queries per second in Q2'24

Cloudflare - Edge vs Core

Edge

- Hundreds of PoPs around the world
- Close to eyeballs
- Each PoP is an independent cluster
- Mostly stateless compute and DNS/proxying/caching/magic
- PoPs vary from handful to thousands of metals
- Bare metal

Core

- Three PoPs in HA setup, with a fourth for DR (search for Cloudflare Code Orange for more details)
- Control plane, customer dashboard and analytics
- Hundreds of services/microservices
- Postgres, K8S, Clickhouse, etc

Edge SRE team

- Follow the sun strategy, Edge SRE are working from 3 regions:
  - Europe
  - APJC
  - US
- Operation, automation and improvements of Edge platform and services
- We manage servers and oversee the platform that runs the services that eyeballs use
- Failures in Edge systems have high impact for customers
- Always looking for opportunities to improve automation and reliability

Cloudflare Edge PoPs (servers perspective)

Datacenter management (DM)

- Software required to operate the PoP
- Not in the direct request path of customer traffic
- NetBoot
- Salt master
- Prometheus

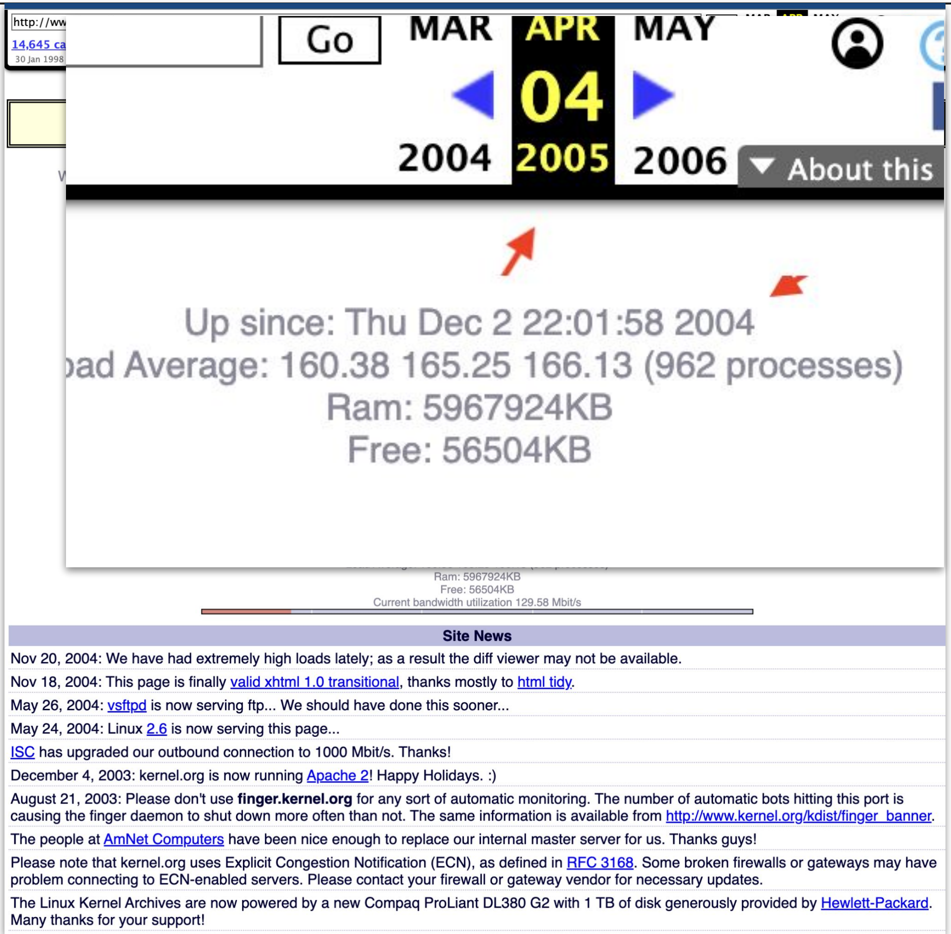
Metal

- How we call our edge compute nodes
- Customer facing software
- Handles requests from eyeballs
- Every metal is configured as the same
- Attracts traffic via ECMP
- UNIMOG - XDP load balancer for dynamic load balancing
- Consul for service discovery, along with internal APIs used by salt to configure services

High uptime used to be cool

- Back in 2005...

https://web.archive.org/web/20050404074539/http://www.kernel.org:80/



Up since: Thu Dec 2 22:01:58 2004
oad Average: 160.38 165.25 166.13 (962 processes)
Ram: 5967924KB
Free: 56504KB

Ram: 5967924KB
Free: 56504KB
Current bandwidth utilization 129.58 Mbit/s

Site News

Nov 20, 2004: We have had extremely high loads lately; as a result the diff viewer may not be available.

Nov 18, 2004: This page is finally valid xhtml 1.0 transitional, thanks mostly to html tidy.

May 26, 2004: vsftpd is now serving ftp... We should have done this sooner...

May 24, 2004: Linux 2.6 is now serving this page...

ISC has upgraded our outbound connection to 1000 Mbit/s. Thanks!

December 4, 2003: kernel.org is now running Apache 2! Happy Holidays. :)

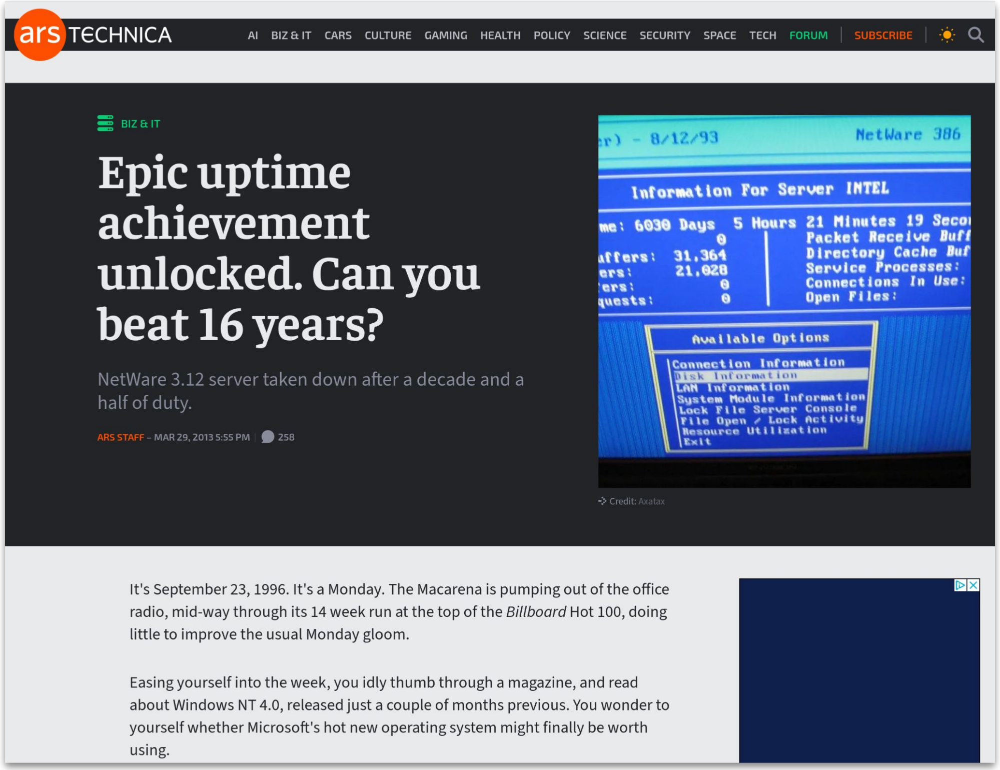
August 21, 2003: Please don't use **finger.kernel.org** for any sort of automatic monitoring. The number of automatic bots hitting this port is causing the finger daemon to shut down more often than not. The same information is available from http://www.kernel.org/kdist/finger_banner.

The people at AmNet Computers have been nice enough to replace our internal master server for us. Thanks guys!

Please note that kernel.org uses Explicit Congestion Notification (ECN), as defined in RFC 3168. Some broken firewalls or gateways may have problem connecting to ECN-enabled servers. Please contact your firewall or gateway vendor for necessary updates.

The Linux Kernel Archives are now powered by a new Compaq ProLiant DL380 G2 with 1 TB of disk generously provided by Hewlett-Packard. Many thanks for your support!

High uptime used to be cool



ars TECHNICA
AI BIZ & IT CARS CULTURE GAMING HEALTH POLICY SCIENCE SECURITY SPACE TECH FORUM SUBSCRIBE
BIZ & IT

Epic uptime achievement unlocked. Can you beat 16 years?

NetWare 3.12 server taken down after a decade and a half of duty.

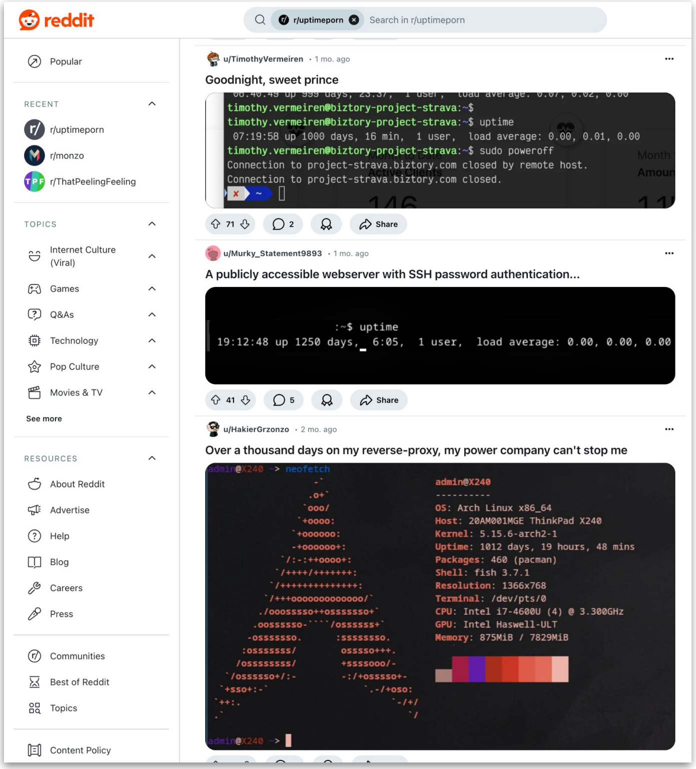
ARS STAFF – MAR 29, 2013 5:55 PM | 258

```
r) - 8/12/93                                  NetWare 386
          Information For Server  INTEL
me: 6030 Days  5 Hours 21 Minutes 19 Seco
                    0          Packet Receive Buff
affers:   31,364               Directory Cache Buf
ers:      21,028               Service Processes:
ers:           0               Connections In Use:
quests:        0               Open Files:

            Available Options
        Connection Information
        Disk Information
        LAN Information
        System Module Information
        Lock File Server Console
        File Open / Lock Activity
        Resource Utilization
        Exit
```

Credit: Axatax

It's September 23, 1996. It's a Monday. The Macarena is pumping out of the office radio, mid-way through its 14 week run at the top of the *Billboard* Hot 100, doing little to improve the usual Monday gloom.

Easing yourself into the week, you idly thumb through a magazine, and read about Windows NT 4.0, released just a couple of months previous. You wonder to yourself whether Microsoft's hot new operating system might finally be worth using.



u/TimothyVermeiren · 1 mo. ago

Goodnight, sweet prince

```
 08:40:49 up 999 days, 23:37,  1 user,  load average: 0.07, 0.02, 0.00
timothy.vermeiren@biztory-project-strava:~$
timothy.vermeiren@biztory-project-strava:~$ uptime
 07:19:58 up 1000 days, 16 min,  1 user,  load average: 0.00, 0.01, 0.00
timothy.vermeiren@biztory-project-strava:~$ sudo poweroff
Connection to project-strava.biztory.com closed by remote host.
Connection to project-strava.biztory.com closed.
 ✗  ~
```

71 2 Share

u/Murky_Statement9893 · 1 mo. ago

A publicly accessible webserver with SSH password authentication...

```
                    :~$ uptime
 19:12:48 up 1250 days,  6:05,  1 user,  load average: 0.00, 0.00, 0.00
```

41 5 Share

u/HakierGrzonzo · 2 mo. ago

Over a thousand days on my reverse-proxy, my power company can't stop me

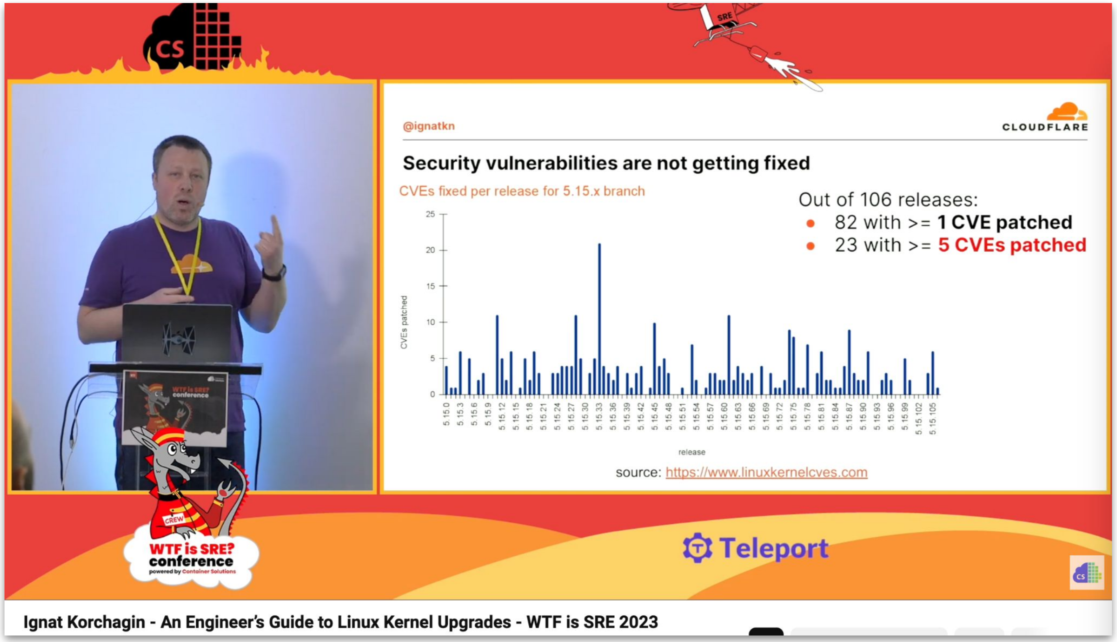
```
admin@X240 -> neofetch
                              admin@X240
                              ----------
                              OS: Arch Linux x86_64
                              Host: 20AM001MGE ThinkPad X240
                              Kernel: 5.15.6-arch2-1
                              Uptime: 1012 days, 19 hours, 48 mins
                              Packages: 460 (pacman)
                              Shell: fish 3.7.1
                              Resolution: 1366x768
                              Terminal: /dev/pts/0
                              CPU: Intel i7-4600U (4) @ 3.300GHz
                              GPU: Intel Haswell-ULT
                              Memory: 875MiB / 7829MiB

admin@X240 ->
```

Why we reboot servers?

- Linux kernel updates
- Distribution upgrades
    - Debian Trixie is coming, get ready
- Delivering infrastructure updates and changes that would break a running system
    - Change disk layout
    - Port binding or IP addressing deployments
    - Reconfiguring ECMP parameters with router
    - Complex feature changes that could cause services to restart or drop eyeball requests
- Firmware upgrades
- Libraries sometimes are updated from Debian security, but not every service is restarted, so vulnerable code may be still running
    - PCI DSS requires that organizations install critical or high-risk security patches within 30 days of release
    - Periodic reboots are a way of enforcing compliance

Why we reboot servers?

- Linux kernel updates
  - Each new kernel version comes with dozens of bug fixes
  - Most minor releases fix CVEs too
  - Can also include performance improvements
- Cloudflare's Linux team automated kernel updates, so a newer kernel is always available for next reboot
- Check Ignat's talk for more reasons: **An Engineer's Guide to Linux Kernel Upgrades**



Security vulnerabilities are not getting fixed

CVEs fixed per release for 5.15.x branch

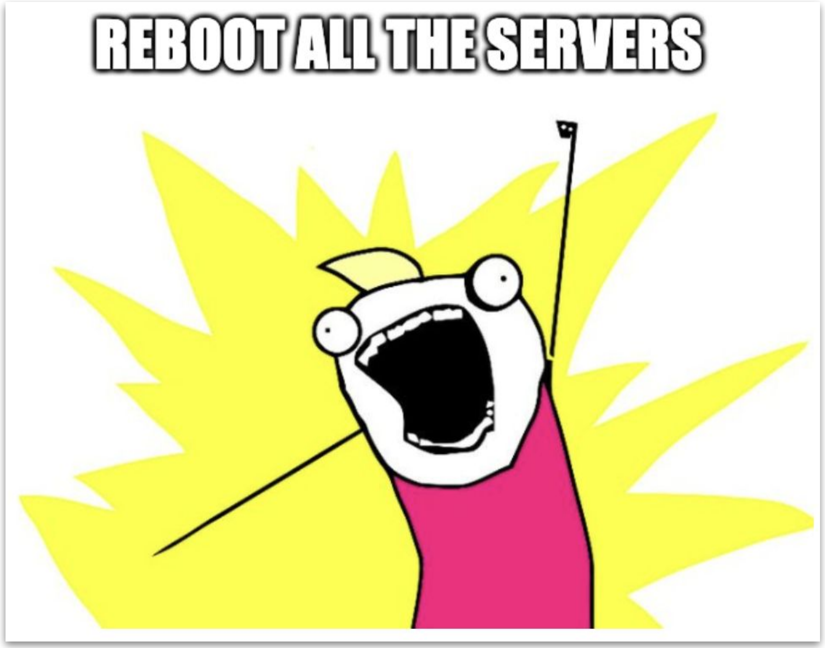
Out of 106 releases:
- 82 with >= **1 CVE patched**
- 23 with >= **5 CVEs patched**

source: https://www.linuxkernelcves.com

Ignat Korchagin - An Engineer's Guide to Linux Kernel Upgrades - WTF is SRE 2023

How to reboot all Edge servers? The exciting way until 2018:

- Disable PoP (drain anycast)
- Execute a few tasks
    - sometimes in hosts
    - sometimes re-provisioning commands
    - sometimes using salt
    - sometimes a code change in git
    - sometimes multiple of above
- Reboot all servers
- Wait for servers to boot
- Review servers that failed to boot, try to debug them.
- Rush, eyeballs in the region are waking up and neighbor PoPs are getting overloaded!
- Enable services
- Execute healthchecks
- Enable PoP
- Repeat for next PoP

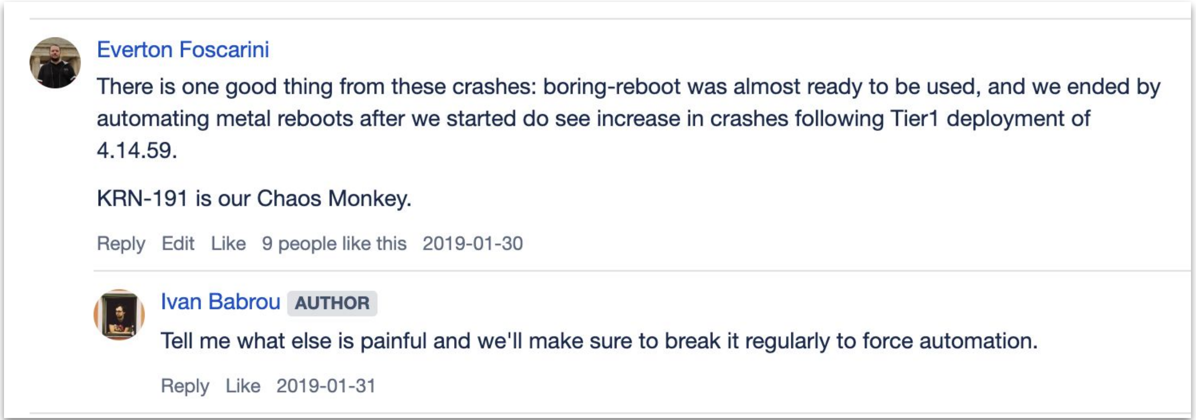


Back in 2018, with a bit more than a **hundred PoPs** the process was taking between **20 and 40 days** to complete, depending on complexity of included changes

Repeat 3x or 4x per year

Don’t want to reboot servers? Kernel can crash too

- Release of Linux 4.14.58/59 (in 2018-07) along with a microcode update via BIOS caused a high rate of server crashes
  - Peaked at 20 crashes per 8h shift
  - Metal recovery still required manual actions
- Prompted for immediate automation of metal enable task
  - automation was already being developed and was deployed after 2 weeks, when crash rate increased
- Multiple kernel versions tested to identify the offending bug, probably something related to retpoline mitigations
- Impact reduced in Linux 4.19



Everton Foscarini

There is one good thing from these crashes: boring-reboot was almost ready to be used, and we ended by automating metal reboots after we started do see increase in crashes following Tier1 deployment of 4.14.59.

KRN-191 is our Chaos Monkey.

Reply Edit Like 9 people like this 2019-01-30

Ivan Babrou AUTHOR

Tell me what else is painful and we'll make sure to break it regularly to force automation.

Reply Like 2019-01-31

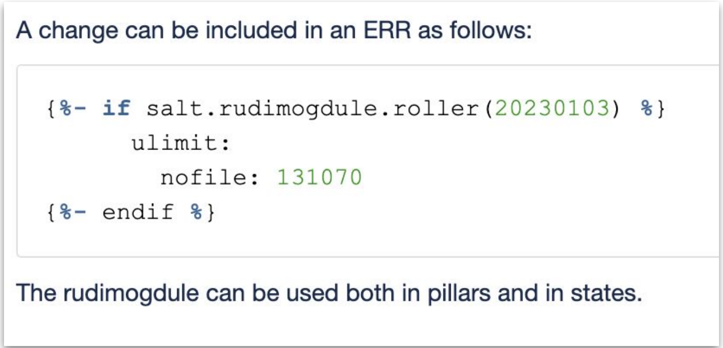
Automate ourselves - metal-enabler-service.sh

- Servers rebooted always required manual actions to enable, due to multiple issues:
  - Technical debt due to software used for data replication
    - Prone to data corruption on crash
    - Unable to self heal
    - Quicksilver replaced it a few months before (https://blog.cloudflare.com/moving-quicksilver-into-production/)
  - Lack of automated healthchecks
    - There was no previous need for creating them
    - Heavy reliance on alerts to expose failures
- Conjunction of factors allowed automation to be finally created and enabled
  - Servers rebooted or crashed now would self-enable
  - Exposed new failure modes that required tuning to enabling traffic slowly, to warm up internal cache
- Also created script to drain traffic orderly and disable services
- Unlocked further automation of reboots

Automate ourselves - rudimog (by Sami Kerola)

- Rudimentary release coordinator
  - Scheduled reboots over a 29 day window for every server
  - Reboots triggered in systemd timer per server

- Implemented method for deploying configuration changes over reboots
  - Server state is stored in UEFI variable, similar to a release version
  - Before reboot
    - changes applied from a known state, with customer-facing services stopped
    - Could wipe disks and partitioning as state was in UEFI
    - Move data between partitions
  - After reboot
    - Salt is aware of server state and can enable new code paths
    - Healthcheck can validate the expected changes were applied


A change can be included in an ERR as follows:

```
{%- if salt.rudimogdule.roller(20230103) %}
      ulimit:
        nofile: 131070
{%- endif %}
```

The rudimogdule can be used both in pillars and in states.

* Name is a word play with UNIMOG, XDP based load balancer:
https://blog.cloudflare.com/unimog-cloudflares-edge-load-balancer/

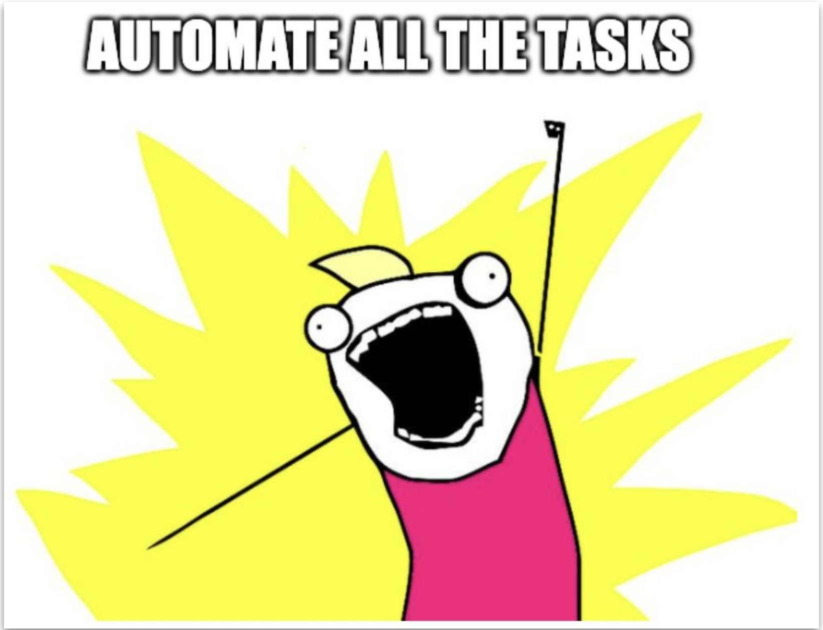
Automate ourselves - boring-reboot-server

- Create a reboot coordinator service per PoP
- Servers that need reboot join a queue and wait for authorization
- Reboots are allowed on off-peak hours
- Servers disable services and reboot themselves
- Restricts how many metals can reboot concurrently
- Can check metrics and further limit reboots due to known failure modes (ECMP group size)
- Tracks server state, and report which ones are unable to recover
- Too many reboot failures block reboots prompting for investigation and restricting impact

How to reboot all Edge servers? The boring way by 2019:

- ~~Disable PoP (drain anycast)~~ - not needed
- ~~Execute a few tasks~~ rudimog
  - ~~sometimes in hosts~~
  - ~~sometimes re-provisioning commands~~
  - ~~sometimes using salt~~
  - ~~sometimes a code change in git~~
  - ~~sometimes multiple of above~~
- ~~Reboot all servers~~ boring-reboot-server
- ~~Wait for servers to boot~~
- Review servers that failed to boot, try to debug them. reboot-rescuer
- ~~Rush, eyeballs in the region are waking up and other PoPs are getting overloaded!~~
- ~~Enable services~~ metal-enabler-service.sh
- ~~Execute healthchecks~~
- ~~Enable PoP~~
- ~~Repeat for next PoP~~ - every PoP is running in parallel, during maintenance window

Edge Reboot Releases became a quarterly task, with dates defined in advance, and teams and Edge SRE could include changes using rudimog to be deployed automatically

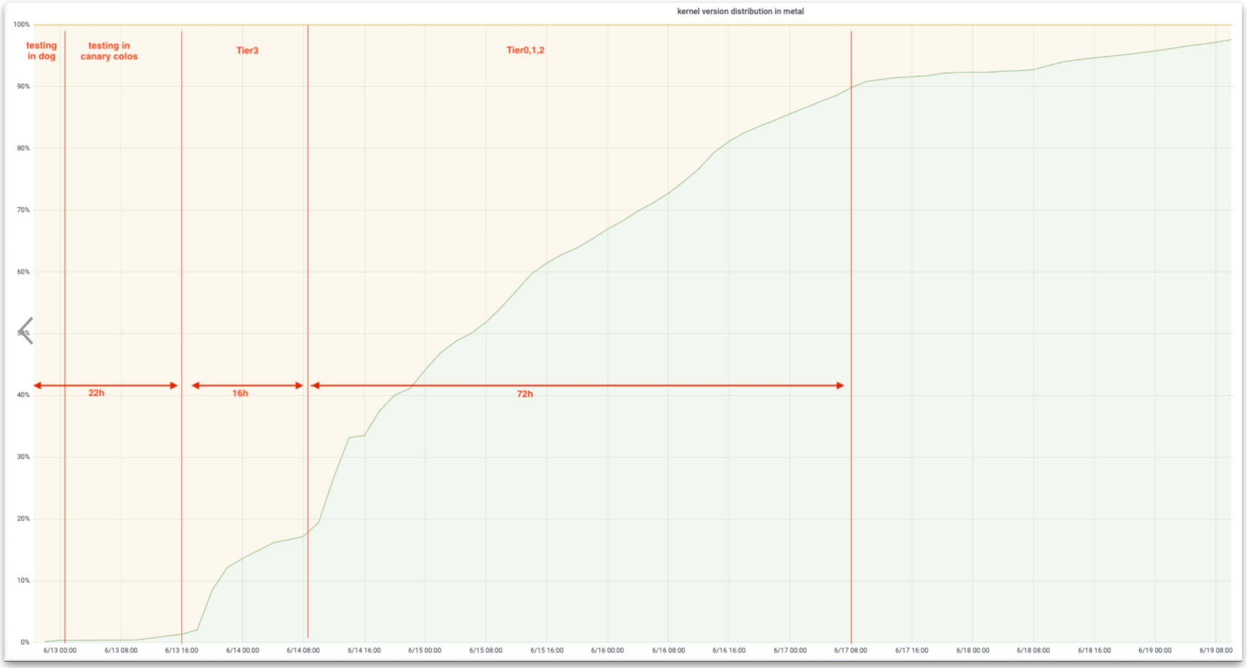


Emergency kernel upgrade - 2019-06

- Edge Reboot Release was automated for ~ 1 year
- Normal reboot cycle would take 29 days to complete
- CVE-2019-11478 - Linux SACK vulnerabilities - gotta update fast
- Memories from 2018-07 crashes are still vivid

Emergency kernel upgrade - 2019-06

- CVE-2019-11478 - Linux SACK vulnerabilities - gotta update fast
- Memories from 2018-07 crashes are still vivid
- Staggered release in first ~40h for validation
- 90% of Edge rebooted in 72h window



boring-reboot-server improvements over time

- Inclusion of more server roles (by **Thomas Lefebvre**)
  - Storage nodes with persistent state
  - Deny reboots if may cause data loss or too many shards offline (by **Rob Dinh**)
  - Multiple reboot queues per role

- Resource constraints
  - Cannot reboot more than a certain % of node role at a time

- Non-deferrable reboots (by **Sergei Klochkov**)
  - Automated reboot for disk failures and other hard failures types

- Dynamic maintenance windows based on server load (by **Opeyemi Onikute**)
  - Maintenance window adjusted automatically based on historical CPU usage

- Reboot-Rescuer - (by **Kasinath Kottukkal** and **Jaryl Chng**)
  - Multiple attempts to bring metals back to life
  - Hardware will always disappoint you

Clustering improvements

- services-enable and service-disable scripts with preflight/healthchecks to increase confidence

- Improvements on how to signal that a node is being drained for other metals in the cluster
  - Consul maintenance status
  - Also exported via prometheus metric

- Implemented a separated systemd target for public-facing services
  - *metal-public-services.target* (by **Nick Rhodes**)
  - Idea of on/off switch for all services, instead of listing all in the script or service dependencies
  - Servers boot to *multi-user.target*, *metal-public-services.target* is enabled after healthchecks pass

- Culture shift from using Salt to start services, to enable services in target
  - Reduce chances of unhealthy servers to join a cluster

Processes improvements

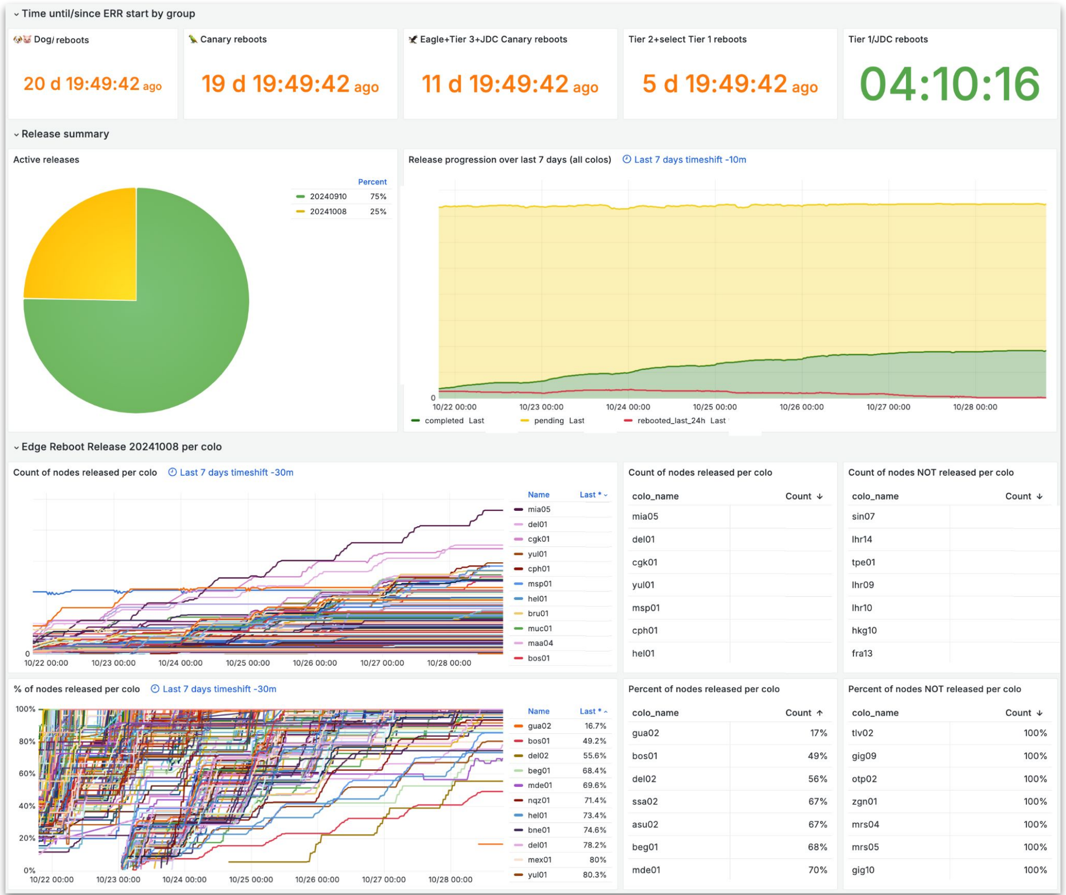
- Edge Reboot Releases happen every 28 days
  - 20240618, 20240716, 20240813, 20240905, 20241008
  - Release start is staggered
  - Edge SRE can pause reboots and help to revert or debug issues
  - High level monitoring over grafana

- Servers in PoPs reserved for dogfooding reboot every 5 days to catch regressions
  - Dogfooding and canary PoPs auto-reboot for firmware and kernel upgrades as soon as available for validation

- Edge Reboot Releases are automatically started and managed
  - Engineers are encouraged to self-serve on reboot changes, docs and validation methods available
  - No more need from approval from SRE to deploy something over reboots

- Alerts:
  - too many nodes fail to re-enable
    - Likely failure on building system state or services failing to start
  - too many nodes have uptime too high
    - Something is blocking the reboot queue, we investigate the regression

Embrace fleet reboots and make them boring



Time until/since ERR start by group

Dog/ reboots	Canary reboots	Eagle+Tier 3+JDC Canary reboots	Tier 2+select Tier 1 reboots	Tier 1/JDC reboots
20 d 19:49:42 ago	19 d 19:49:42 ago	11 d 19:49:42 ago	5 d 19:49:42 ago	04:10:16

Release summary

Active releases

	Percent
🟩 20240910	75%
🟨 20241008	25%

Release progression over last 7 days (all colos) Last 7 days timeshift -10m

Edge Reboot Release 20241008 per colo

Count of nodes released per colo Last 7 days timeshift -30m

Name	Last *
mia05	
del01	
cgk01	
yul01	
cph01	
msp01	
hel01	
bru01	
muc01	
maa04	
bos01	

Count of nodes released per colo

colo_name	Count ↓
mia05	
del01	
cgk01	
yul01	
msp01	
cph01	
hel01	

Count of nodes NOT released per colo

colo_name	Count ↓
sin07	
lhr14	
tpe01	
lhr09	
lhr10	
hkg10	
fra13	

% of nodes released per colo Last 7 days timeshift -30m

Name	Last *
gua02	16.7%
bos01	49.2%
del02	55.6%
beg01	68.4%
mde01	69.6%
nqz01	71.4%
hel01	73.4%
bne01	74.6%
del01	78.2%
mex01	80%
yul01	80.3%

Percent of nodes released per colo

colo_name	Count ↑
gua02	17%
bos01	49%
del02	56%
ssa02	67%
asu02	67%
beg01	68%
mde01	70%

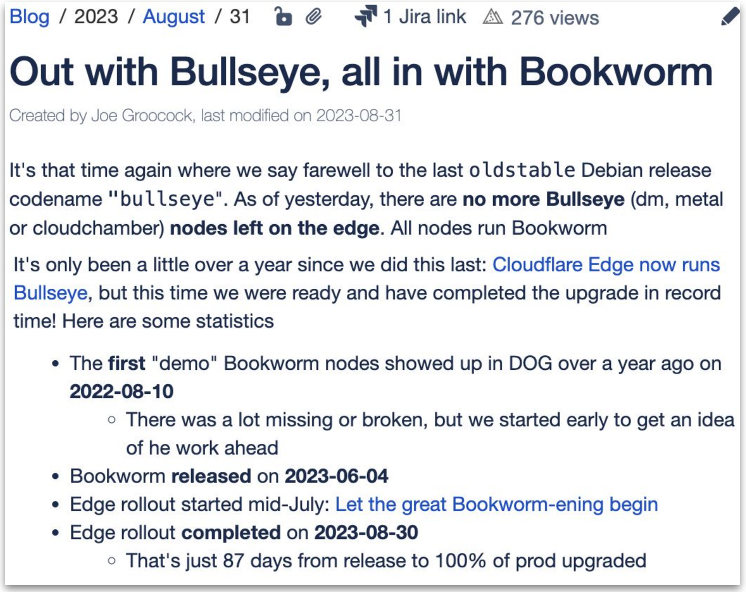
Percent of nodes NOT released per colo

colo_name	Count ↓
tlv02	100%
gig09	100%
otp02	100%
zgn01	100%
mrs04	100%
mrs05	100%
gig10	100%

Takeaways

- Identify major toil tasks and invest to automate them as soon as possible
  - Rough estimation that in 2018 manual reboots kept busy 1 SRE per region for a month
  - 480h of work per reboot release
  - This time commitment had to be approved by SRE director

- Reboot automation allows releasing difficult things with reduced effort and risk
  - Always up to date with Linux kernel
  - Major distribution upgrades are as hard as validating they work in dogfooding/canary PoPs
  - Reduced risk of impact as servers are drained during impactful changes



Blog / 2023 / August / 31 1 Jira link 276 views

Out with Bullseye, all in with Bookworm

Created by Joe Groocock, last modified on 2023-08-31

It's that time again where we say farewell to the last `oldstable` Debian release codename "`bullseye`". As of yesterday, there are **no more Bullseye** (dm, metal or cloudchamber) **nodes left on the edge**. All nodes run Bookworm

It's only been a little over a year since we did this last: Cloudflare Edge now runs Bullseye, but this time we were ready and have completed the upgrade in record time! Here are some statistics

- The **first** "demo" Bookworm nodes showed up in DOG over a year ago on **2022-08-10**
  - There was a lot missing or broken, but we started early to get an idea of he work ahead
- Bookworm **released** on **2023-06-04**
- Edge rollout started mid-July: Let the great Bookworm-ening begin
- Edge rollout **completed** on **2023-08-30**
  - That's just 87 days from release to 100% of prod upgraded

Takeaways

- Create methods of gracefully draining services from clusters to avoid customer impact
  - Specially important when you reboot every host every month.
- Design systems prepared for eventual reboots
  - Then you are ready for random crashes too
- Create preflight/healthchecks and end-to-end tests
  - Allow self-healing when things go wrong on reboot
  - Broken server can disable itself automatically if healthchecks fail
- Reboot all the time to catch regressions
  - Regressions are frequent in systems that are always changing
  - Elect dogfooding/canary nodes that will get updates early and reboot them often

Thank you

Questions?

Everton Didone Foscarini
SRE - Edge - London
Cloudflare

@foscarini@mastodon.social

work with me!