



# 5 years of FRRouting



RIPE 84, May 2022, Berlin

Donatas Abraitis  
NetDEF / OpenSourceRouting

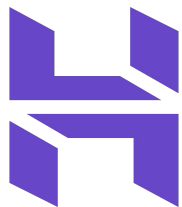


**netDEF**



# Who is this guy?

- Sysadmin, DevOps, SRE (since 2011)
- FRR maintainer (since 2019)
- SWE at NetDEF / OpenSourceRouting
- SRE at Hostinger



**netDEF**





# Releases

- FRR employs **X.Y.Z** versioning scheme
- 3 releases per-year
- Stages
  - Freeze
  - Stabilization
  - RC
  - Release
- 2-3 release managers and rotate per release
- No LTS releases
  - Backports in two latest releases
  - @Mergifyio helps here

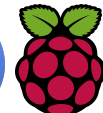
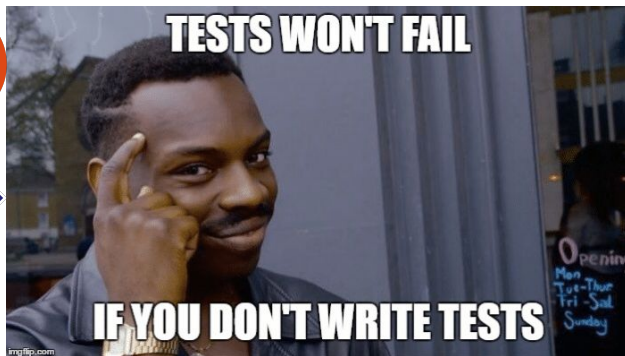
Release	2022-07-05
RC	2022-06-21
Stabilization branch	2022-06-07
Freeze	2022-05-24

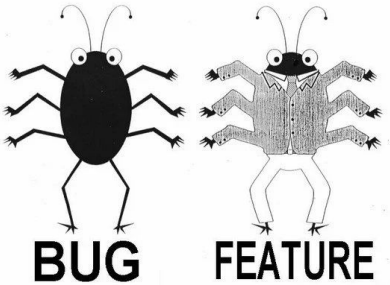
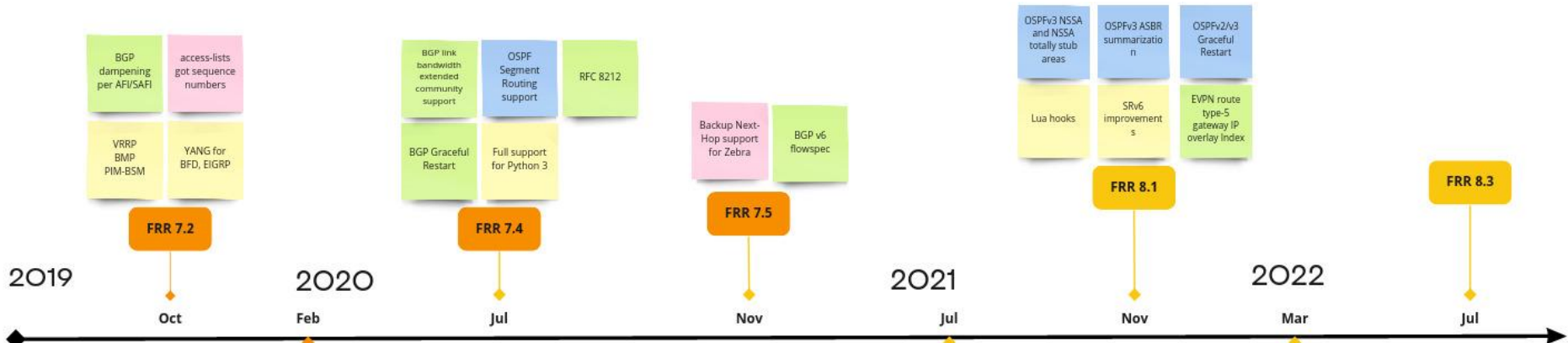




# Continuous Integration

- `@frrbot` repo decorator (labels, styling suggestions, sanity checks for commits)
- Tests are mandatory for
  - All new features
  - Bug fixes that requires to be RFC compliance
- ~600 CI agents
  - ~70 Raspberry PI
  - CentOS 7, RedHat 8, Fedora 29
  - Debian 9/10/11, Ubuntu 16.04/18.04/20.04/22.04
  - FreeBSD 11/12, OpenBSD 7, NetBSD 9
  - Virtual Machines
  - Managed with Ansible
  - Build success rate: 50%
  - Average run: 130 minutes
  - Average time to fix a failure: 1 hour, 55 minutes
  - Average number of builds between fixes: 2.35 builds

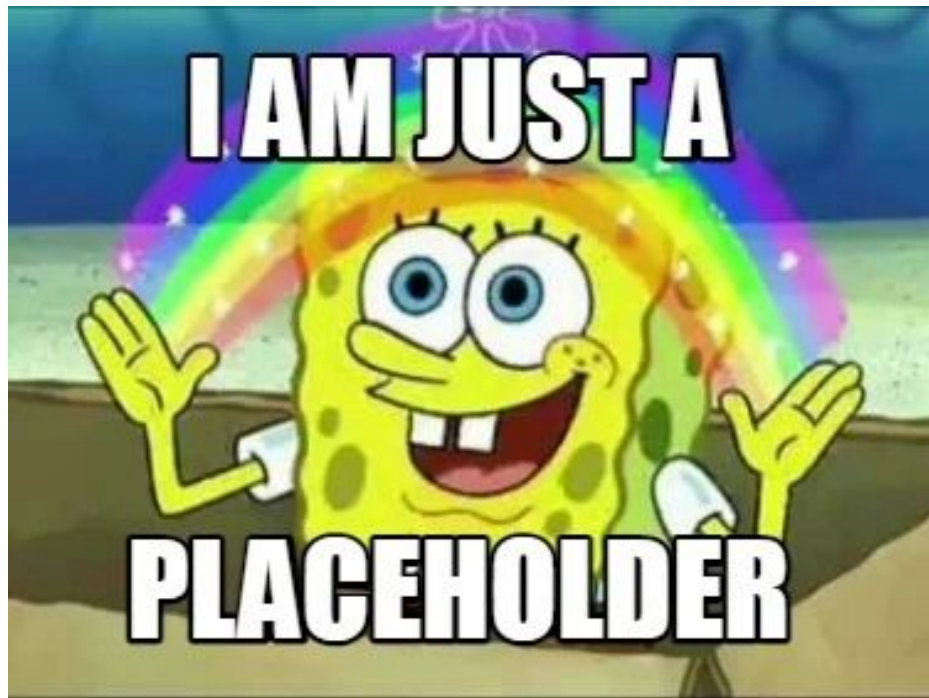






## Discover unknown knowns

- Lua hooks
- sharpd daemon
- Tracing in FRR
- BGP Long-lived Graceful Restart
- BGP suppress-duplicates





# Unknown feature: Lua hooks

```
$ cat /etc/frr/scripts/zebra.lua
local clients = {}
clients["172.16.13.1/32"] = { description = "client1" }
clients["172.16.13.2/32"] = { description = "client2" }

function on_rib_process_dplane_results(ctx)
    -- Log only newly added routes
    if ctx.zd_op ~= 1 then
        return {}
    end

    prefix = ctx.rinfo.zd_dest.network
    if clients[prefix] then
        log.info(clients[prefix].description)
    else
        log.warn("Unknown client")
    end
    return {}
end
```

```
zebra[162633]: [N9WAX-QFTTH] Unknown client
zebra[162633]: [JZMNV-MDH8J] New prefix 172.16.13.2/32, from client2
zebra[162633]: [VF3QY-FP1PP] Unknown client

spine(config)# zebra on-rib-process script zebra
Successfully added script clients for hook call on_rib_process_dplane_results
spine(config)#
```

- Since FRR 8.1
- Zebra RIB dataplane
  - Read-only
- BGP + route-map
- Lua 5.3
- *--enable-scripting*





# Unknown feature: Lua hooks (cont.)

```
router bgp 65000
 neighbor 192.168.0.1 remote-as external
 !
 address-family ipv4 unicast
  neighbor 192.168.0.1 route-map peer in
 exit-address-family
 !
 !
 route-map peer permit 10
  match script rmap
 exit
 !

$ vtysh -c 'show bgp ipv4 unicast neighbors 192.168.0.1 routes json' | jq
'.routes."172.16.1.0/24"[0].metric'
333
```

```
$ cat /etc/frr/scripts/rmap.lua
function route_match(prefix, attributes, peer,
  RM_FAILURE, RM_NOMATCH, RM_MATCH, RM_MATCH_AND_CHANGE)

  function on_match_and_change(prefix, attributes)
    attributes["metric"] = 333
    return {
      action = RM_MATCH_AND_CHANGE,
      attributes = attributes
    }
  end

  return on_match_and_change(prefix, attributes)
end
```

- <http://docs.frrouting.org/en/latest/scripting.html>
- <http://docs.frrouting.org/en/latest/zebra.html#scripting>





# Unknown feature: sharpd

- Testing framework
  - No external software to generate millions of routes
- Verify
  - How fast the routes are processed?
  - How many resources do I need?
  - How does my router behave with millions of routes overall?
- Can send `opaque` data to Zebra

```
router bgp 65000
  address-family ipv4 unicast
    redistribute sharp
  exit-address-family
  !
exit

# sharp install routes 172.16.13.1 nexthop 192.168.0.1 254
# show ip route 172.16.13.0/24 longer-prefixes | include 172.16.13.1/32
D>* 172.16.13.1/32 [150/0] via 192.168.0.1, eth1, weight 1, 00:00:14

# sharp install routes 172.16.31.1 nexthop 192.168.0.2 1 opaque client1
# show ip route 172.16.31.1
Routing entry for 172.16.31.1/32
  Known via "sharp", distance 150, metric 0
  Last update 00:00:08 ago
    192.168.0.2, via eth1, weight 1
    Opaque Data: client1
Routing entry for 172.16.31.1/32
  Known via "bgp", distance 20, metric 0, best
  Last update 00:03:19 ago
    * 192.168.0.2, via eth1, weight 1
```

- <http://docs.frrouting.org/en/latest/sharp.html>



# Unknown feature: Tracing (LTTng)

```
$ vtysh -c 'show version' | grep --color lttng
  '--enable-systemd' '--enable-examplendir=/usr/share/doc/frr/examples/'
'--localstatedir=/var/run/frr' '--sbindir=/usr/lib/frr' '--sysconfdir=/etc/frr'
'--enable-vtysh' '--enable-isisd' '--enable-pimd' '--enable-watchfrr'
'--enable-ospfclient=yes' '--enable-ospfapi=yes' '--enable-multipath=64'
'--enable-user=frr' '--enable-group=frr' '--enable-vty-group=frrvty'
'--enable-configfile-mask=0640' '--enable-logfile-mask=0640' '--enable-rtadv'
'--enable-fpm' '--enable-ldpd' '--with-pkg-git-version'
'--with-pkg-extra-version=MyOwnFRRVersion' '--enable-rpki=yes'
'--enable-sharpd' '--enable-lttng=yes'

$ systemctl show frr | grep Envir
Environment=LD_PRELOAD=libltng-ust-fork.so # Required only for LTTng

$ lttng list --userspace | grep --color frr_bgp:process_update
  frr_bgp:process_update (loglevel: TRACE_INFO (6)) (type: tracepoint)

$ lttng create frr --output=/root/lttng-traces/frr
$ lttng enable-event --userspace frr_bgp:process_update
$ lttng start && sleep 10 ; lttng stop
```

```
$ lttng view
Trace directory: /root/lttng-traces/frr

frr_bgp:process_update: { cpu_id = 6 }, { peer = "192.168.0.1",
prefix = "172.16.1.0/24", addpath_id = 0, afi = 1, safi = 1,
attribute_ptr = 0x7FFD3537DB00 }
```

- Not
  - sysdig, strace, tcpdump, ...
- Linux Trace Toolkit Next-Generation
- Since FRR 8.0
- `--enable-lttng`

- <https://docs.frrouting.org/projects/dev-guide/en/latest/tracing.html>
- <https://lttng.org>



# Unknown feature: Tracing (SystemTap)

```
$ vtysh -c 'show version' | grep --color usdt
  '--enable-systemd' '--enable-examplendir=/usr/share/doc/frr/examples/'
 '--localstatedir=/var/run/frr' '--sbindir=/usr/lib/frr' '--sysconfdir=/etc/frr'
 '--enable-vtysh' '--enable-isis' '--enable-pimd' '--enable-watchfrr'
 '--enable-ospfclient=yes' '--enable-ospfapi=yes' '--enable-multipath=64'
 '--enable-user=frr' '--enable-group=frr' '--enable-vty-group=frrvty'
 '--enable-configfile-mask=0640' '--enable-logfile-mask=0640' '--enable-rtadv'
 '--enable-fpm' '--enable-ldpd' '--with-pkg-git-version'
 '--with-pkg-extra-version=MyOwnFRRVersion' '--enable-rpki=yes'
 '--enable-sharpd' '--enable-usdt=yes'

$ readelf -n /usr/lib/frr/bgpd | grep process_update -A2
Name: process_update
Location: 0x00000000014b2b8, Base: 0x00000000029ad81, Semaphore:
0x0000000000000000
Arguments: 8@%r14 8@%r15 4@-616(%rbp) 4@%r12d 4@%ebx 8@-624(%rbp)

$ stap -L 'process("/usr/lib/frr/bgpd").mark("*")' | wc -l
28

$ stap -L 'process("/usr/lib/frr/bgpd").function("bgp_create@bgpd/bgpd.c:3069")'
process("/usr/lib/frr/bgpd").function("bgp_create@bgpd/bgpd.c:3069")
$inst_type:enum bgp_instance_type $name:char const* $as:as_t*
```

```
$ cat process_updates.stp
probe process("/usr/lib/frr/bgpd").mark("process_update")
{
  aspath = @cast($arg6, "attr")->aspath;
  printf("> %s via %s (%s)\n",
        user_string($arg2),
        user_string(@cast($arg1, "peer")->host),
        user_string(@cast(aspath, "aspath")->str));
}

$ stap -m process_updates -p4 process_updates.stp
$ staprun process_updates.ko
> 172.16.1.0/24 via 192.168.0.1 (65001)
> 172.16.0.0/24 via 192.168.0.1 (65001)
```

- User-Space probing (uprobes)
- Kernel hacking (with Guru mode)
- `--enable-usdt`
- Kernel module
- Can't use with LTTng

- <https://docs.frrouting.org/projects/dev-guide/en/latest/tracing.html>
- <https://sourceware.org/systemtap>



# Unknown feature: BGP Long-lived Graceful Restart

```
router bgp 65000
  bgp long-lived-graceful-restart stale-time 30
  bgp graceful-restart restart-time 5
  bgp graceful-restart
!
BGP routing table entry for 192.168.100.1/32, version 103
Paths: (2 available, best #2, table default)
  Advertised to non peer-group peers:
    donatas-pc(192.168.10.17)
  65000 65066, (stale)
    192.168.0.2(spine1-debian-11) from spine1-debian-11(192.168.0.2) (0.0.0.0)
    Origin incomplete, metric 0, localpref 190, valid, external
    Community: llgr-stale
    Last update: Thu Mar 31 10:37:39 2022
    Time until Long-lived stale route deleted: 28
  65003
    192.168.10.17(donatas-pc) from donatas-pc(192.168.10.17) (192.168.100.1)
    Origin incomplete, metric 0, valid, external, best (First path received)
    Last update: Thu Mar 31 10:22:48 2022
```

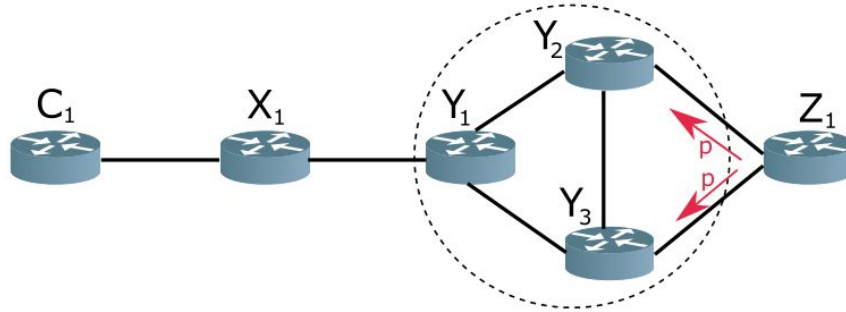
- `bgp graceful-restart restart-time (0-4095)`  
(~1.1h)
- Communities
  - Routes with LLGR\_STALE are less-preferred
  - Routes with NO\_LLGR are flushed immediately
- Since FRR 8.2

• <http://docs.frrouting.org/en/latest/bgp.html#graceful-restart>

• <http://docs.frrouting.org/en/latest/bgp.html#long-lived-graceful-restart>



# Unknown feature: `bgp suppress-duplicates`



```
bgp community-list standard c1 seq 1 permit 65004:2
bgp community-list standard c1 seq 2 permit 65004:3
!
route-map c1 permit 10
  set comm-list c1 delete
!
```

- X1 receives paths with communities
  - 65004:2
  - 65004:3
- Disable the link Y1-Y2
- Communities are stripped at egress from X1 towards C1
- Updates are forced for
  - Graceful Restart
  - Route Refresh
- Since FRR 8.0
- Enabled by default

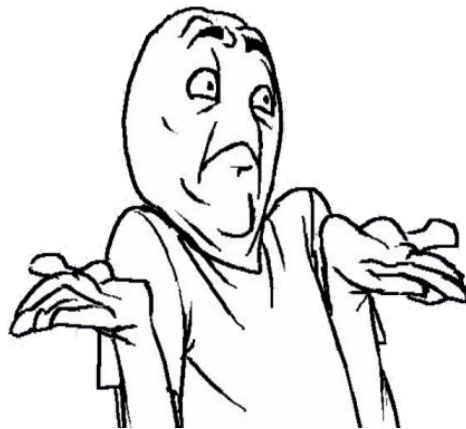
- <http://docs.frrouting.org/en/latest/bgp.html#suppress-duplicate-updates>
- <https://www.cmand.org/communityexploration>



# How to start contributing?

## How did I start?

- 4 years ago
- Never been a programmer
- Motivation to work on huge open-source projects
  - Took open issues and tried to solve
  - The first Pull Request I created was to implement route-map (set) for default-originate



```
router bgp 65000
  neighbor 2001:db8::1 remote-as external
  !
  address-family ipv6 unicast
    neighbor 2001:db8::1 default-originate route-map default
  exit-address-family
  !
  route-map default permit 10
    set community 65000:900
  !
```



# How to start contributing? (cont.)

TL;DR: I have a fix, how to propose it?

- Create a fork of [frrouting/frr](https://github.com/frrouting/frr) repository to your own account
  - Create a separate branch based on the master branch
    - `git checkout -b fix/bgpd_nhlen`
  - Make changes and stage
    - `git add`
  - Apply code formatting
    - `git clang-format`
  - Run unit tests
    - `make check`
    - `make dist`
  - Commit changes
    - Documentation is a **MUST** if the changes introduce a new CLI (or deprecate)
    - Topology tests are **REQUIRED** as well for new functionality
    - Do not forget to properly signed off for every commit
      - `git commit -s`
    - Every commit should have a decent description
    - Add a prefix for every commit to distinguish a daemon (`bgpd: add`, `pimd: fix`, `ospfd: remove`)
  - Take off
    - `git push`
- <http://docs.frrouting.org/projects/dev-guide/en/latest/workflow.html>

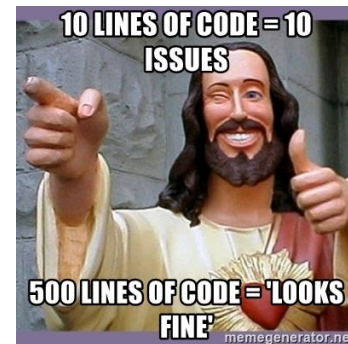




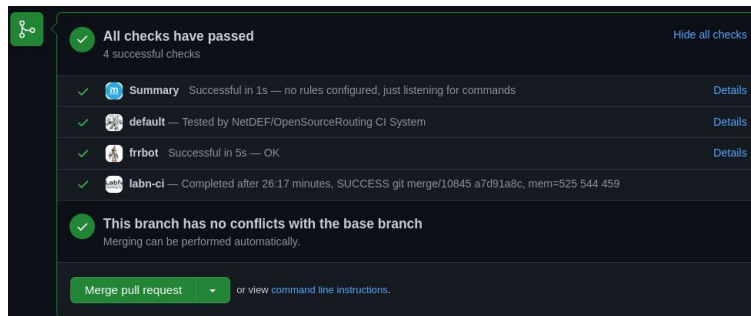
# How to start contributing? (cont.)

TL;DR: I have a fix, how to propose it? (cont.)

- Create Pull Request on Github
- Wait for CI to kick in
  - 50% success ratio
  - `ci:rerun` failed `test_ospfv3_single_area`
- Once it's **green**, usually people start reviewing
  - Pull Requests marked as **Draft** as skipped from reviews
  - It can take even weeks if the changes require further discussions or changes
  - Technical community call on every Tuesday
- Use FRRouting Slack to ask for help
- Less-known mailing lists also exist



- <https://frrouting.org/community>
- <https://lists.frrouting.org>

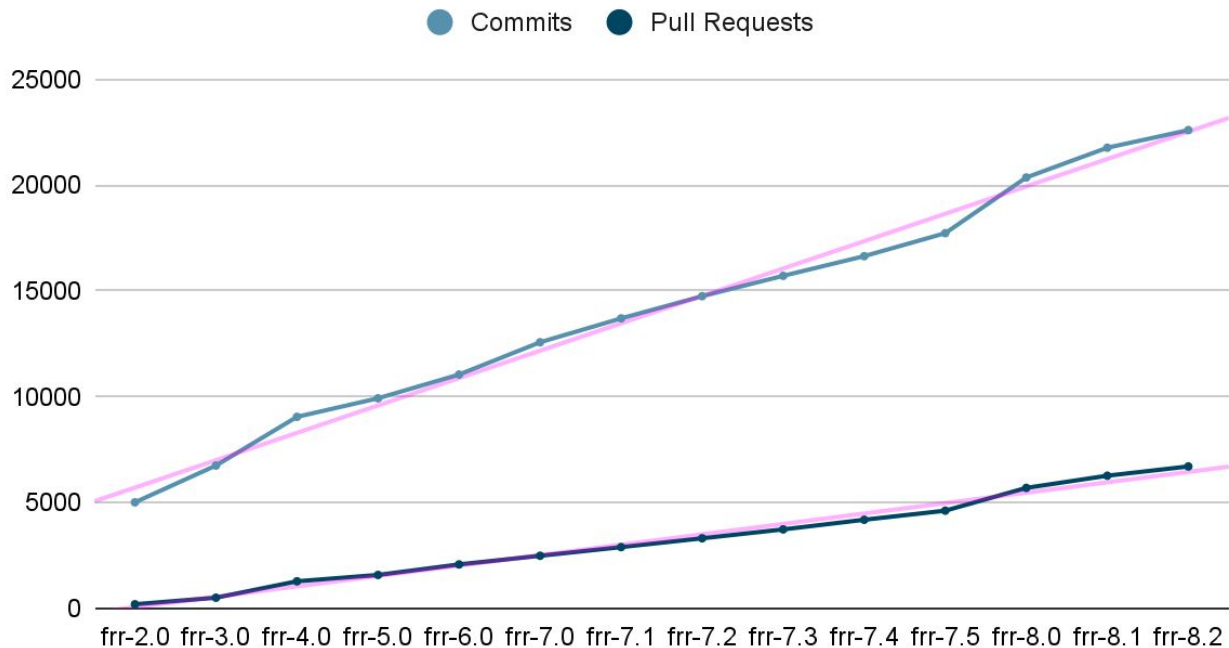






# Show me the numbers

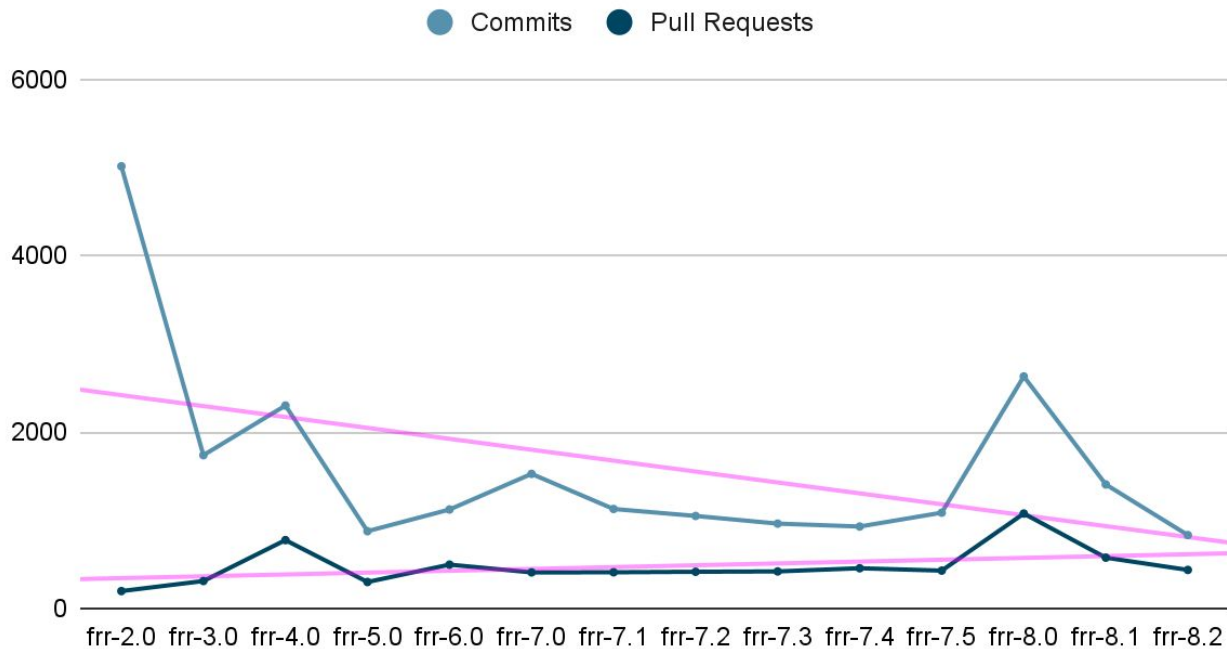
Commits/Pull Requests (overall)





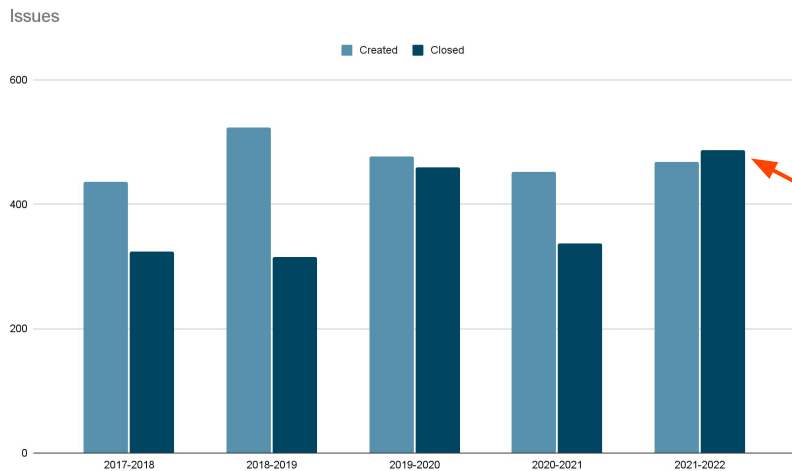
# Show me the numbers

Commits/Pull Requests (per release)





# Show me the numbers



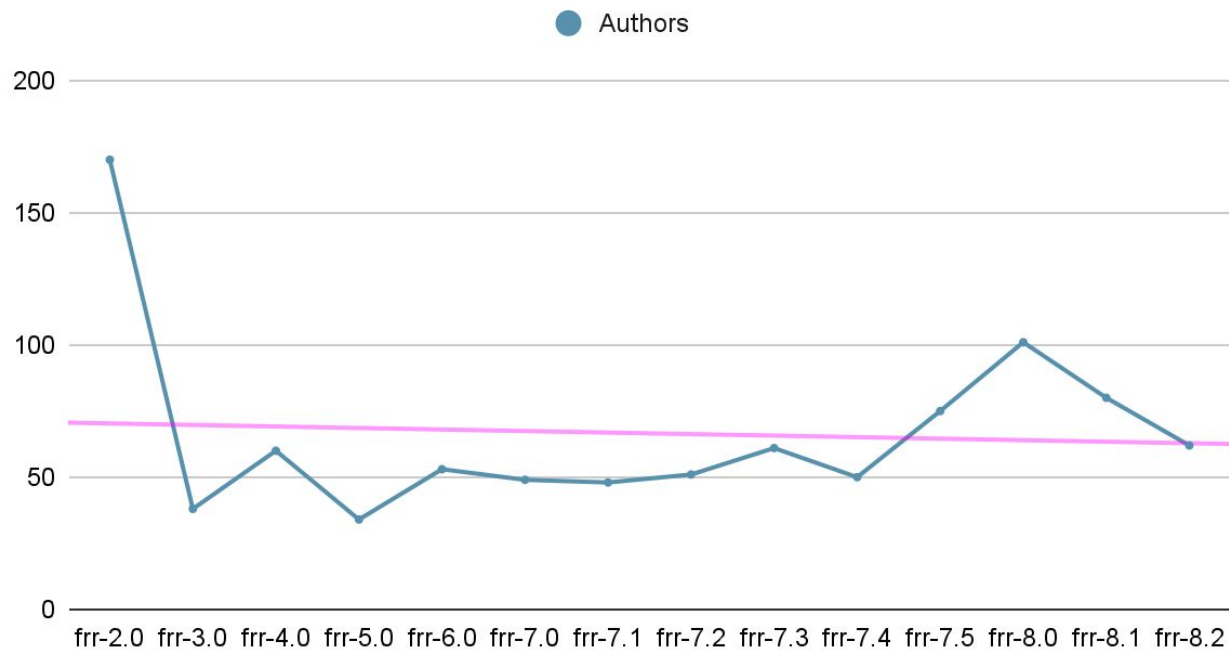
## Pull Requests





# Show me the numbers

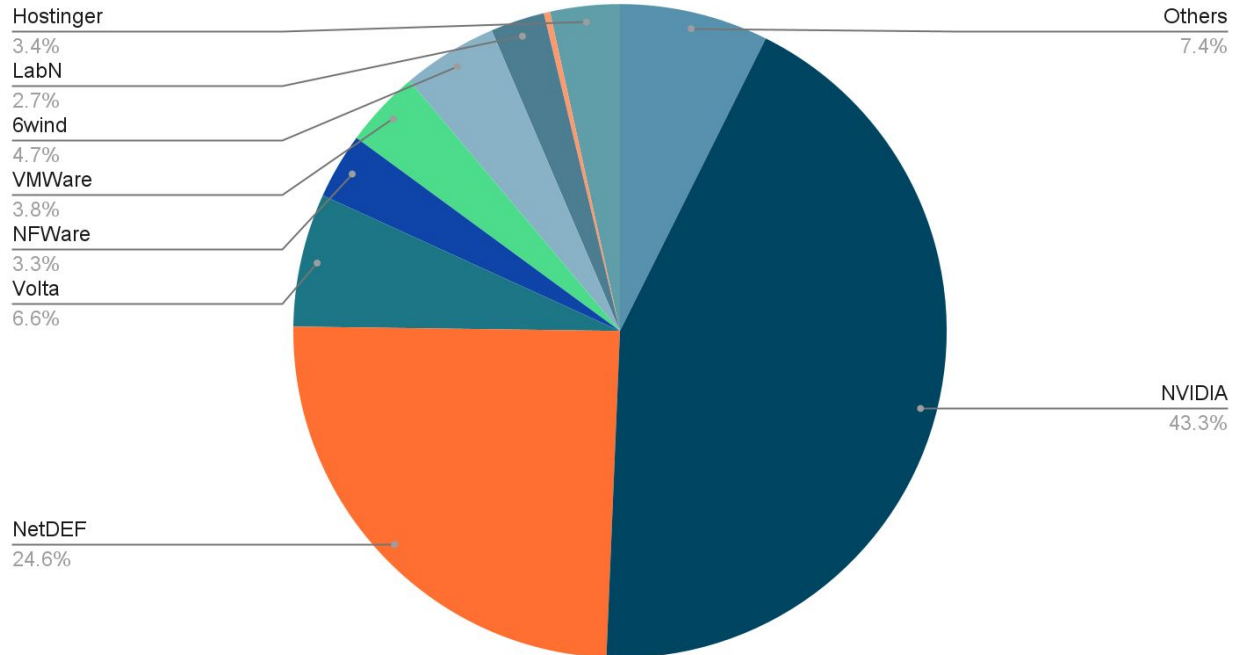
Authors (per release)





# Show me the numbers

The percentage of commits per company (since frr-2.0)





# STOP WAR



@abradona



/ton31337

Donatas Abraitis



**netDEF**