https://medium.com/@amiremohamadi

https://virgool.io/@amiremohamadi

https://amiremohamadi.github.io

Me?!

- https://github.com/bpftrace/bpftrace/commits?author=amiremohamadi
- https://lore.kernel.org/lkml/?q=amiremohamadi
- https://ebpf.io/case-studies

eBPF Case Studies

Here are some of the organizations that are using eBPF in production. If you're using eBPF and aren't on this list, **please submit a pull request**.

Google

Google uses eBPF for security auditing, packet processing, and performance monitoring

VIDEO · TALK

NETFLIX

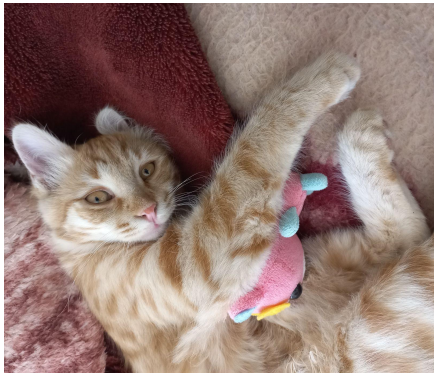
Netflix uses eBPF at scale for network insights

BLOG · VIDEO

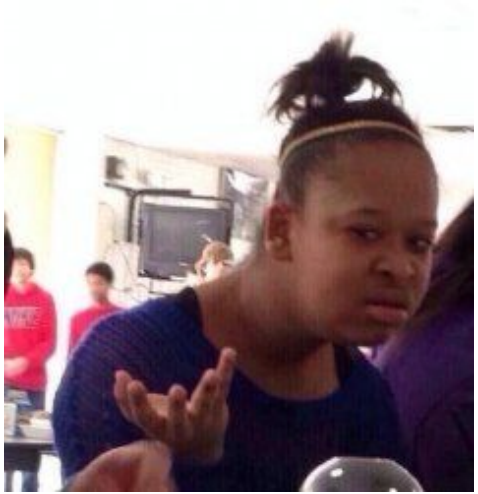
arvancloud

ArvanCloud uses eBPF for their CDN product

BLOG

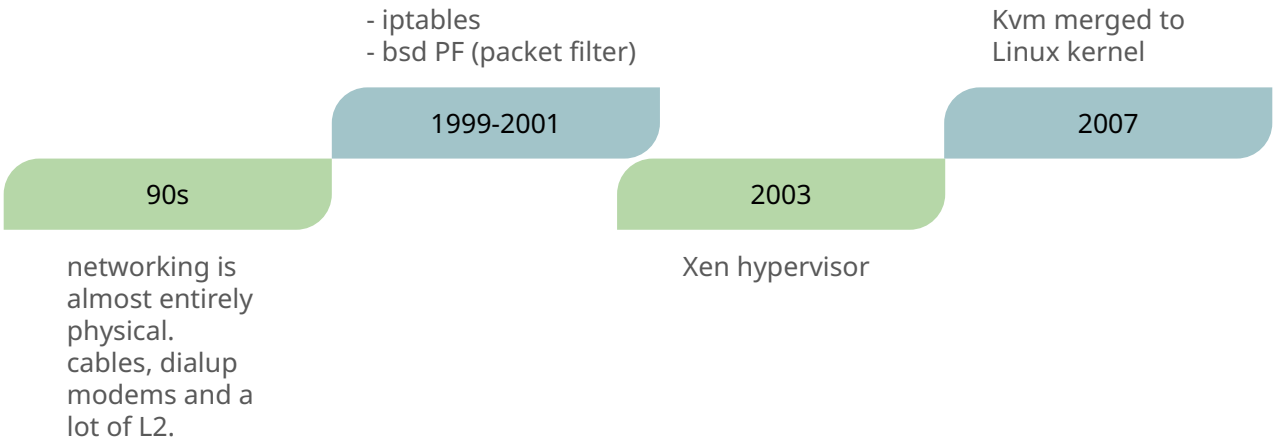


MOTHER
OF
5 CATS



WTF is eBPF?

HISTORY


- iptables
- bsd PF (packet filter)
Kvm merged to
Linux kernel
1999-2001
2007
90s
2003
networking is
almost entirely
physical.
cables, dialup
modems and a
lot of L2.
Xen hypervisor

HISTORY

Still lots of iptables

First commit into kubernates

2014

2013

Hello Docker! Networking is mostly inherited from VM era

HISTORY


SPEAKEASY
PRODUCTIONS
eBPF
UNLOCKING
THE KERNEL
PREMIERING NOV 8TH
AT KUBECON + CLOUDNATIVECON CHICAGO
eBPF FOUNDATION
ISOVALENT
intel.

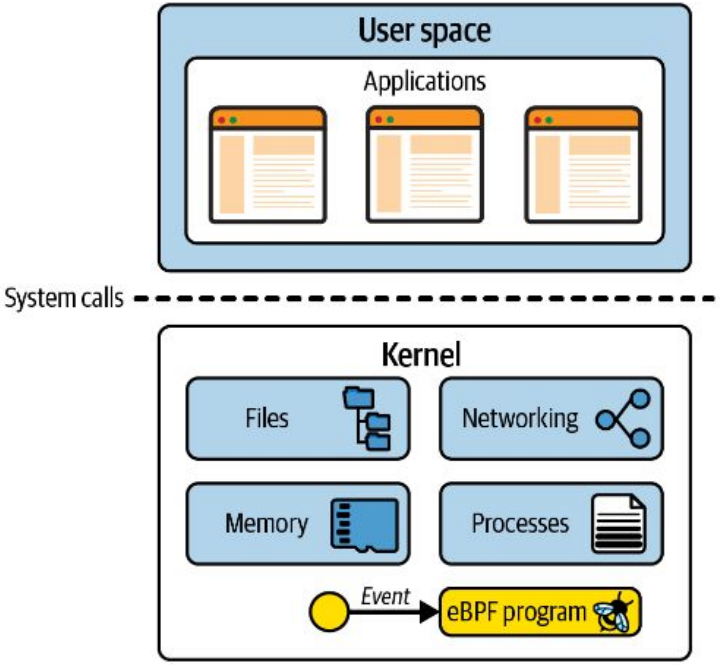
alexei starovoitov sent a patch improving the existing BPF infrastructure in the kernel and as a result BPF -> eBPF

netdev.vger.kernel.org archive mirror

search help / color / mirror / Atom feed

```
From: Daniel Borkmann <dborkman@redhat.com>
To: davem@davemloft.net
Cc: ast@plumgrid.com, netdev@vger.kernel.org,
        Hagen Paul Pfeifer <hagen@jauu.net>,
        Kees Cook <keescook@chromium.org>, Paul Moore <pmoore@redhat.com>,
        Ingo Molnar <mingo@kernel.org>,
        "H. Peter Anvin" <hpa@linux.intel.com>,
        linux-kernel@vger.kernel.org
Subject: [PATCH net-next 8/9] net: filter: rework/optimize internal BPF interpreter's instruction set
Date: Fri, 21 Mar 2014 13:20:17 +0100   [thread overview]
Message-ID: <1395404418-25376-9-git-send-email-dborkman@redhat.com> (raw)
In-Reply-To: <1395404418-25376-1-git-send-email-dborkman@redhat.com>

From: Alexei Starovoitov <ast@plumgrid.com>

This patch replaces/reworks the kernel-internel BPF interpreter with
an optimized BPF instruction set format that is modelled closer to
mimic native instruction sets and is designed to be JITed with one to
one mapping. Thus, the new interpreter is noticeably faster than the
current implementation of sk_run_filter(); mainly for two reasons:
```

Kernel


User space
Applications
System calls
Kernel
Files
Networking
Memory
Processes
Event
eBPF program

Patching the Kernel

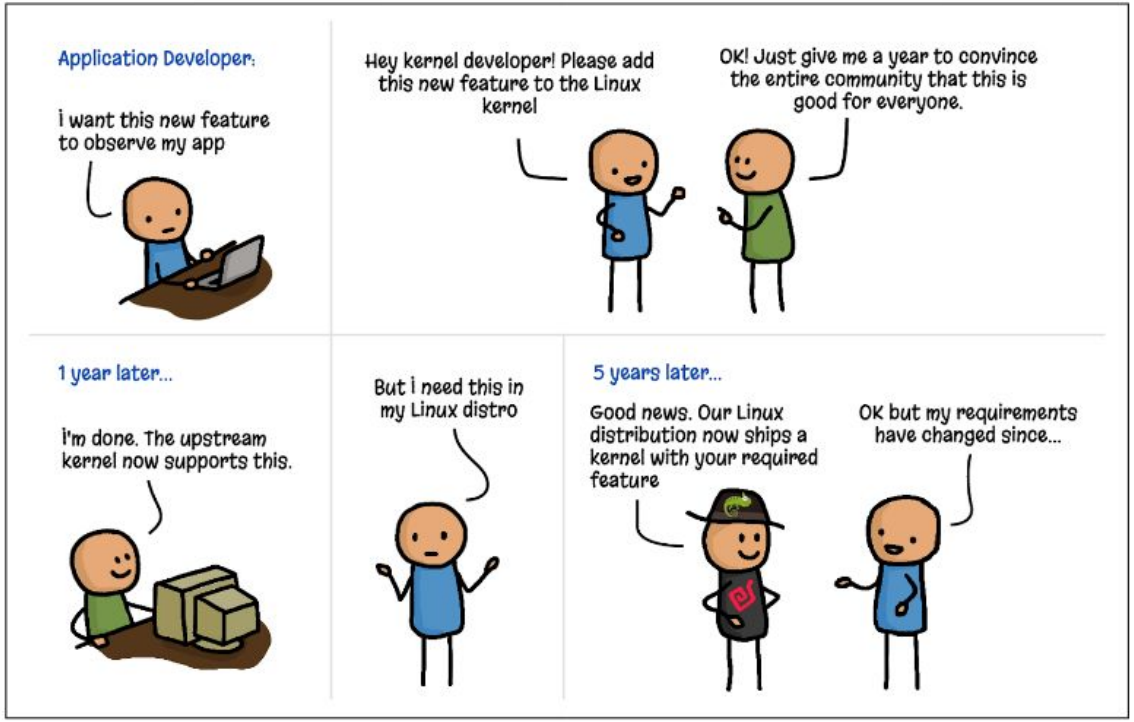
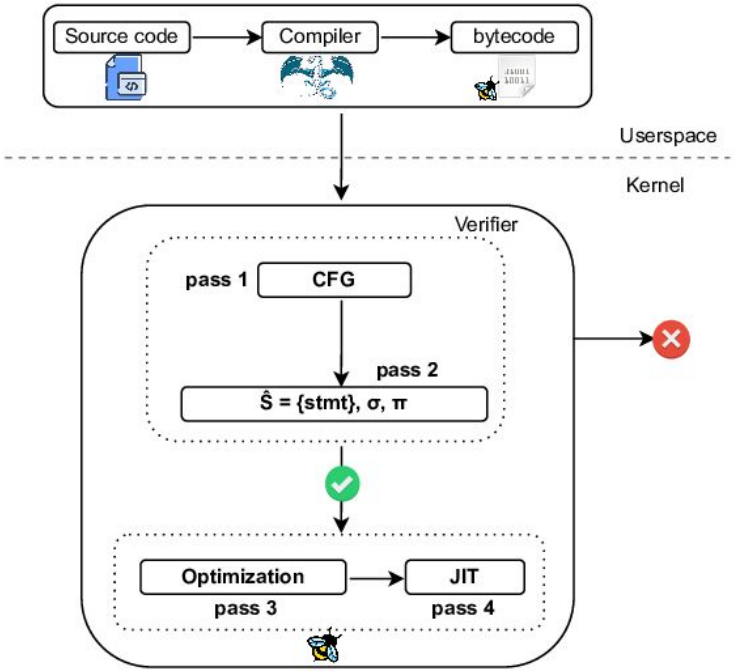


Figure 1-2. Adding features to the kernel (cartoon by Vadim Shchekoldin, Isovalent)

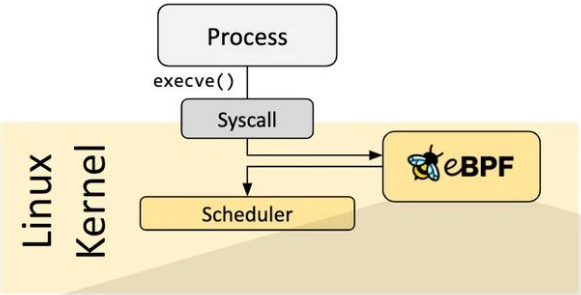
Kernel Modules

```
[   27.509349] CR2: 0000000000000000 CR3: 0000000119506000 CR4: 0000000000060670
[   27.509938] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[   27.510500] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
[   27.511080] Stack:
[   27.511518]  ffffffffa702c597 0000000000000002 ffffffffffffffffb ffffbca4008b3
f08
[   27.513069]  0000000002375008 ffffffffa702c9db ffff9725965ac000 ffffffffa6e7a
c00
[   27.514603]  0000000000000002 ffff972596333f00 ffffffffa6e0b3b0 ffff972596333
f00
[   27.516136] Call Trace:
[   27.516569]  [<ffffffffa702c597>] ? __handle_sysrq+0xf7/0x150
[   27.517063]  [<ffffffffa702c9db>] ? write_sysrq_trigger+0x2b/0x30
[   27.517565]  [<ffffffffa6e7ac00>] ? proc_reg_write+0x40/0x70
[   27.518057]  [<ffffffffa6e0b3b0>] ? vfs_write+0xb0/0x190
[   27.518540]  [<ffffffffa6e0c7f2>] ? SyS_write+0x52/0xc0
[   27.519037]  [<ffffffffa6c03b7d>] ? do_syscall_64+0x8d/0xf0
[   27.519530]  [<ffffffffa721924e>] ? entry_SYSCALL_64_after_swapgs+0x58/0xc6
[   27.520063] Code: 41 5c 41 5d 41 5e 41 5f e9 3c 08 cf ff 66 2e 0f 1f 84 00 00
 00 00 00 66 90 0f 1f 44 00 00 c7 05 29 5e a8 00 01 00 00 00 0f ae f8 <c6> 04 25
 00 00 00 00 01 c3 0f 1f 44 00 00 0f 1f 44 00 00 53 8d
[   27.533456] RIP  [<ffffffffa702be62>] sysrq_handle_crash+0x12/0x20
[   27.534112]  RSP <ffffbca4008b3e78>
[   27.534538] CR2: 0000000000000000
```

eBPF Verifier

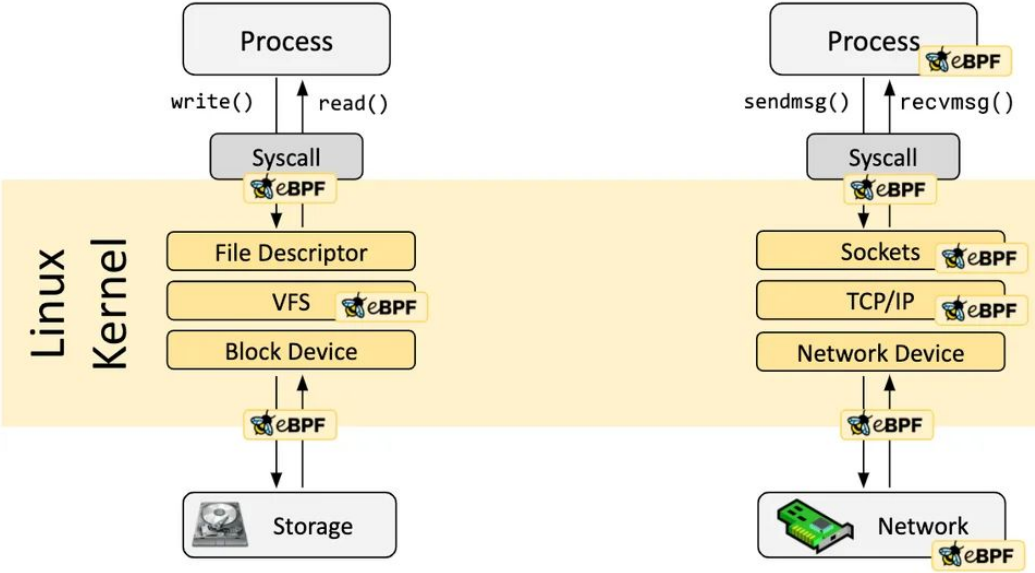


eBPF

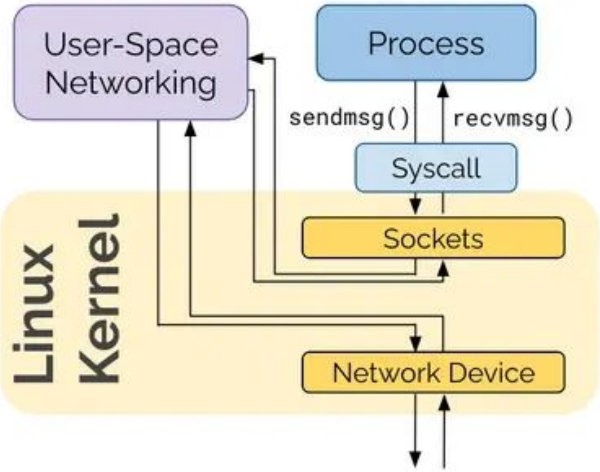


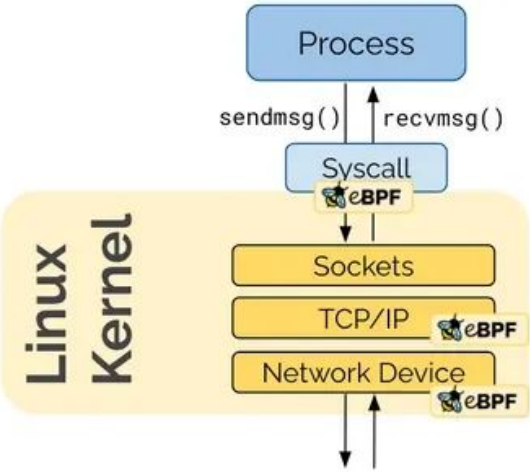

```
int syscall__ret_execve(struct pt_regs *ctx)
{
        struct comm_event event = {
                .pid = bpf_get_current_pid_tgid() >> 32,
                .type = TYPE_RETURN,
        };

        bpf_get_current_comm(&event.comm, sizeof(event.comm));
        comm_events.perf_submit(ctx, &event, sizeof(event));

        return 0;
}
```

eBPF

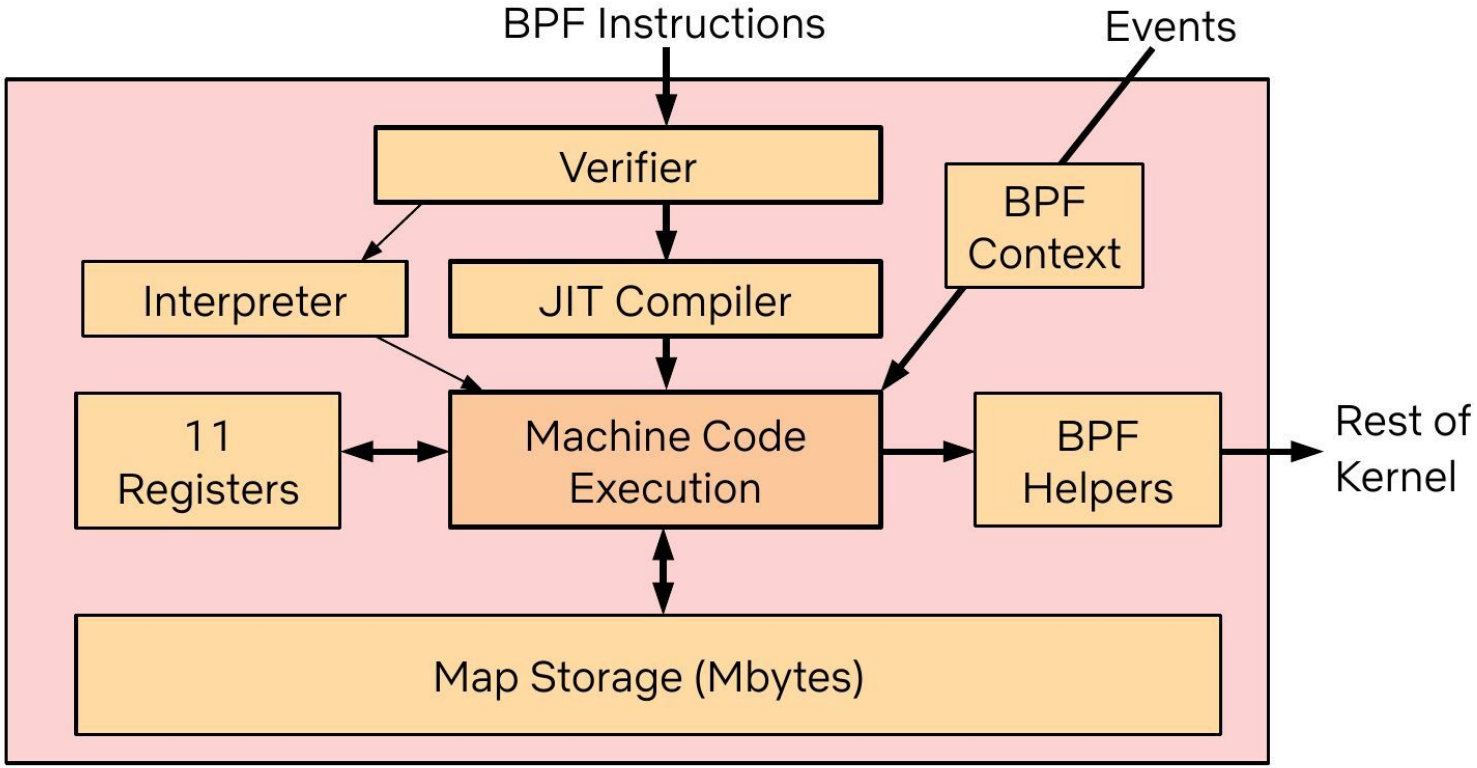

Process
write()
read()
Syscall
eBPF
Linux Kernel
File Descriptor
VFS
eBPF
Block Device
eBPF
Storage
Process
eBPF
sendmsg()
recvmsg()
Syscall
eBPF
Sockets
eBPF
TCP/IP
eBPF
Network Device
eBPF
Network
eBPF

eBPF


User-Space Networking
Process
sendmsg()
recvmsg()
Syscall
Sockets
Network Device
Linux Kernel


Process
sendmsg()
recvmsg()
Syscall
eBPF
Sockets
TCP/IP
eBPF
Network Device
eBPF
Linux Kernel

BPF Internals

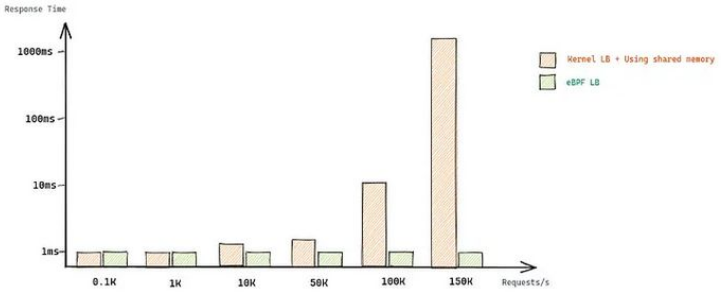


Use cases

We removed Shared-Memory by building an eBPF Load-Balancer!

Amir Mohamadi · Follow
4 min read · Feb 21, 2023

20



Benchmark: 1. Shared Memory 2. eBPF Load Balancer

Scan to read

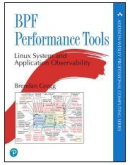
Use cases

bpftrace

XDP

bpfconf

BPF microconference



& Facebook Katran, Google KRSI, Netflix flowsrus,
and many more

Yes, it’s Turing Complete!

amiremohamadi clean up the maps de63421 · 7 months ago

LICENSE	add LICENSE	7 months ago
README.md	add README.md	7 months ago
snake.bt	clean up the maps	7 months ago

README GPL-3.0 license

bpfsnake

a bpftrace implementation of snake game.

Scan to play

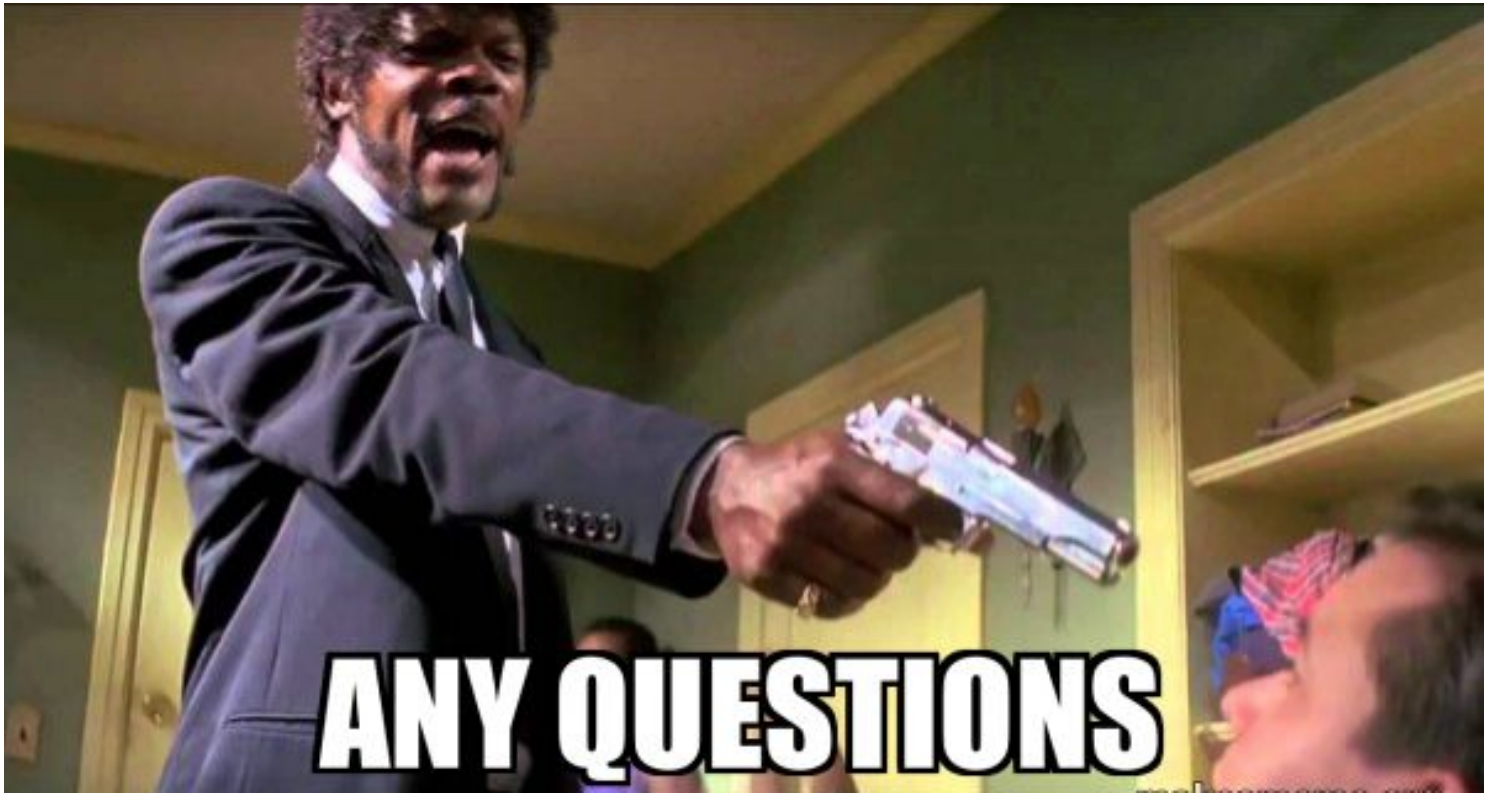
Wrap up!

Who to follow?

Liz Rice

Brendan Gregg

Bill Mulligan


ANY QUESTIONS