

faster core (*python) deploys going down the rabbit hole

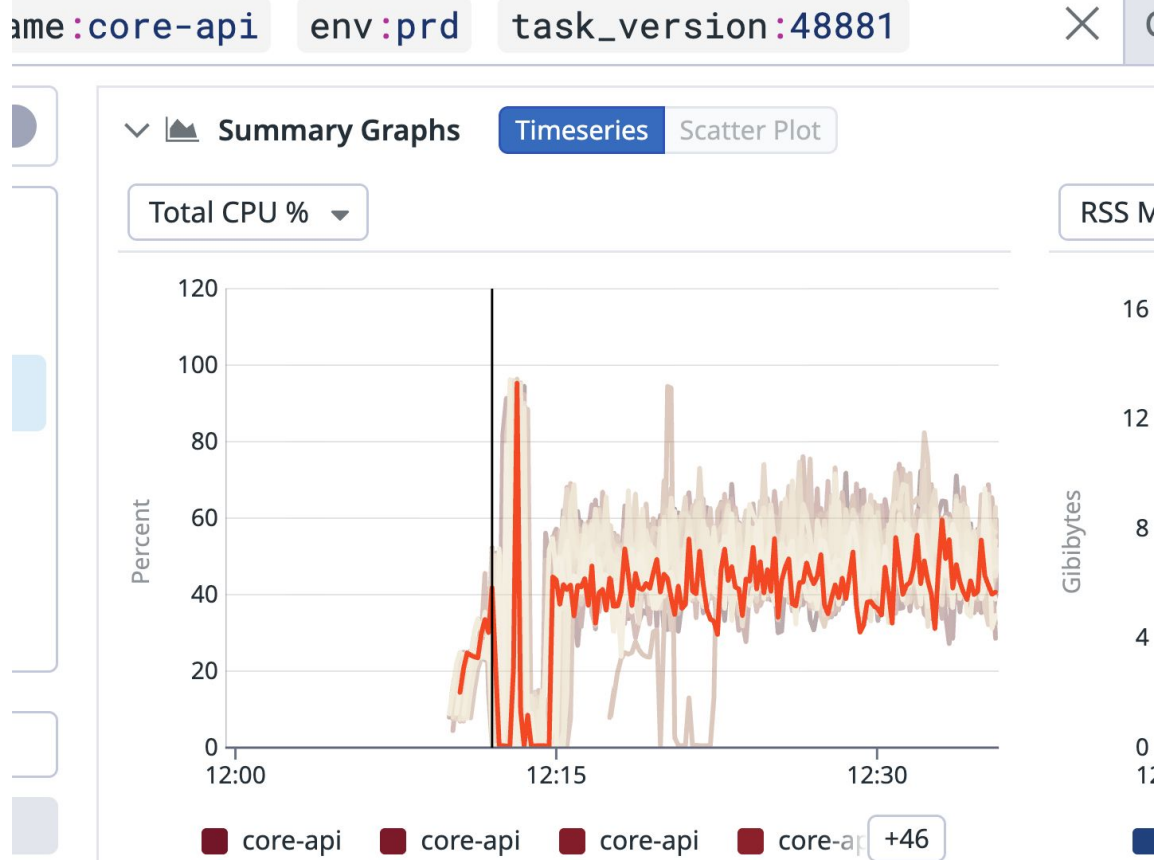
Umar Yousafzai | 01/15/2026

uyousafz [at] uwaterloo [dot] ca



What is #container-boot-cpu-spike?

who is cpu?



Was it too many requests?

the plot thickens...just ride it out



Stephen Snow VIP 09:58

Ah - so the thing I wanted to confirm was true, in that the exptl workers (which **aren't serving any traffic as of yesterday afternoon**) still exhibit the same system CPU spike as other workers.



incident APP 18:33

replied to [a thread](#)



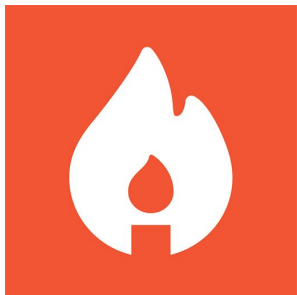
@ssnow_ramp merged a Pull Request:

[delay successful health checks by configured amount](#)

Repo: [ramp/core](#) | Author: [ssnow_ramp](#)



job done?
no



sev0/1/2/



Stephen Snow VIP 15:18

i will execute the rollback



chill for 2 extra mins

game plan w/ dev-deploys

where are my cpu cycles?

1. confirm behaviour in dev-deploy via top
2. look at post-fork background callbacks for core
3. get the big guns out with perf & py-spy

```
top - 14:27:04 up 20 min,  0 users,  load average: 0.38, 0.77, 0.92
Tasks: 14 total, 1 running, 13 sleep, 0 d-sleep, 0 stopped, 0 zombie
%Cpu0  :  0.0/1.3   1[          ]   %Cpu1  :  0.7/0.7   1[          ]
%Cpu2  :  0.7/1.3   2[          ]   %Cpu3  :  0.0/1.3   1[          ]
%Cpu4  :  0.7/0.7   1[          ]   %Cpu5  :  1.3/1.3   3[          ]
%Cpu6  :  2.6/1.3   4[          ]   %Cpu7  :  1.4/0.7   2[          ]
GiB Mem : 38.4/62.0  [          ]   GiB Swap: 0.0/0.0  [          ]
```

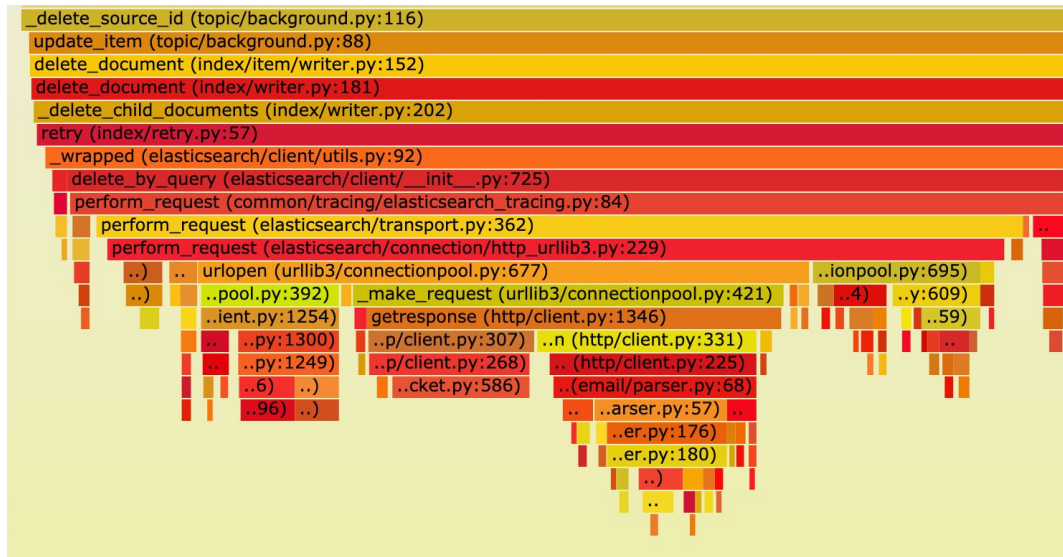
PID	USER	%MEM	TIME+	COMMAND
34	appuser	4.8	3:16.44	gunicorn: master [suma.wsgi:app]
1	appuser	0.0	0:00.03	make run_gunicorn
13	root	0.0	0:00.07	/managed-agents/execute-command/amazon+
28	appuser	0.0	0:00.00	/bin/sh -c set -x; \ reload_arg=""; \ +
32	appuser	0.0	0:00.00	nginx: master process nginx -c /etc/ng+
33	appuser	0.0	0:00.26	nginx: worker process
39	root	0.0	0:00.14	/managed-agents/execute-command/ssm-ag+
217	appuser	4.7	0:01.70	gunicorn: worker [suma.wsgi:app]
232	appuser	4.7	0:01.89	gunicorn: worker [suma.wsgi:app]
254	appuser	4.7	0:02.21	gunicorn: worker [suma.wsgi:app]
270	appuser	4.7	0:02.53	gunicorn: worker [suma.wsgi:app]
963	root	0.0	0:00.65	/managed-agents/execute-command/ssm-se+
976	root	0.0	0:00.00	bash

Callback	Purpose
initialize_db_extensions()	Initialize DB after fork
TemporalExtensionLoader.load()	Initialize Temporal SDK
_init_ld_relay_proxy_client()	Init LaunchDarkly relay proxy
start_launchdarkly_long_cache_refresh()	Start LD cache refresh
apply_network_egress_patches()	Apply network egress patches
enable_env_secret_tracking()	Enable secret tracking
_init_profiler (dd_profiling)	Initialize Datadog profiler

py-spy

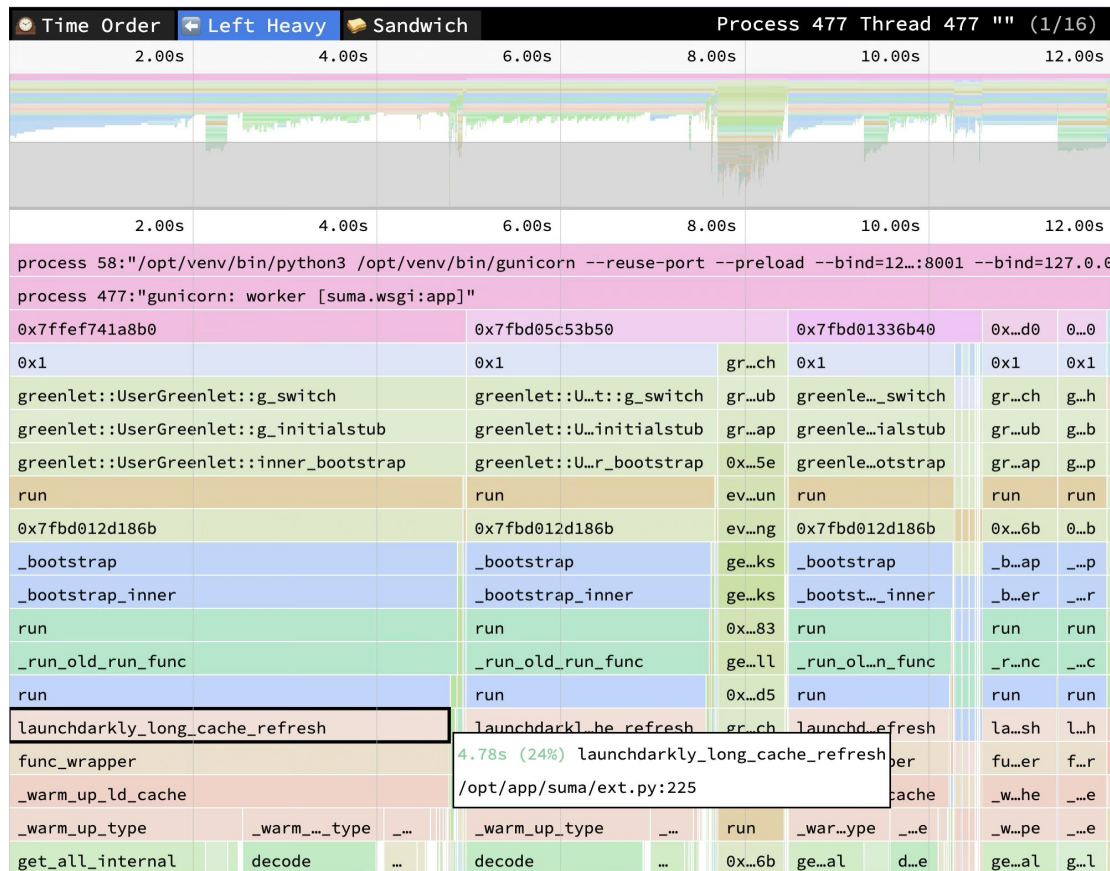
it's lit

- spies on a python app by reading it's system memory for call stack
- result is a flame graph
- requires SYS_PTRACE enabled (so one proc can spy on another)



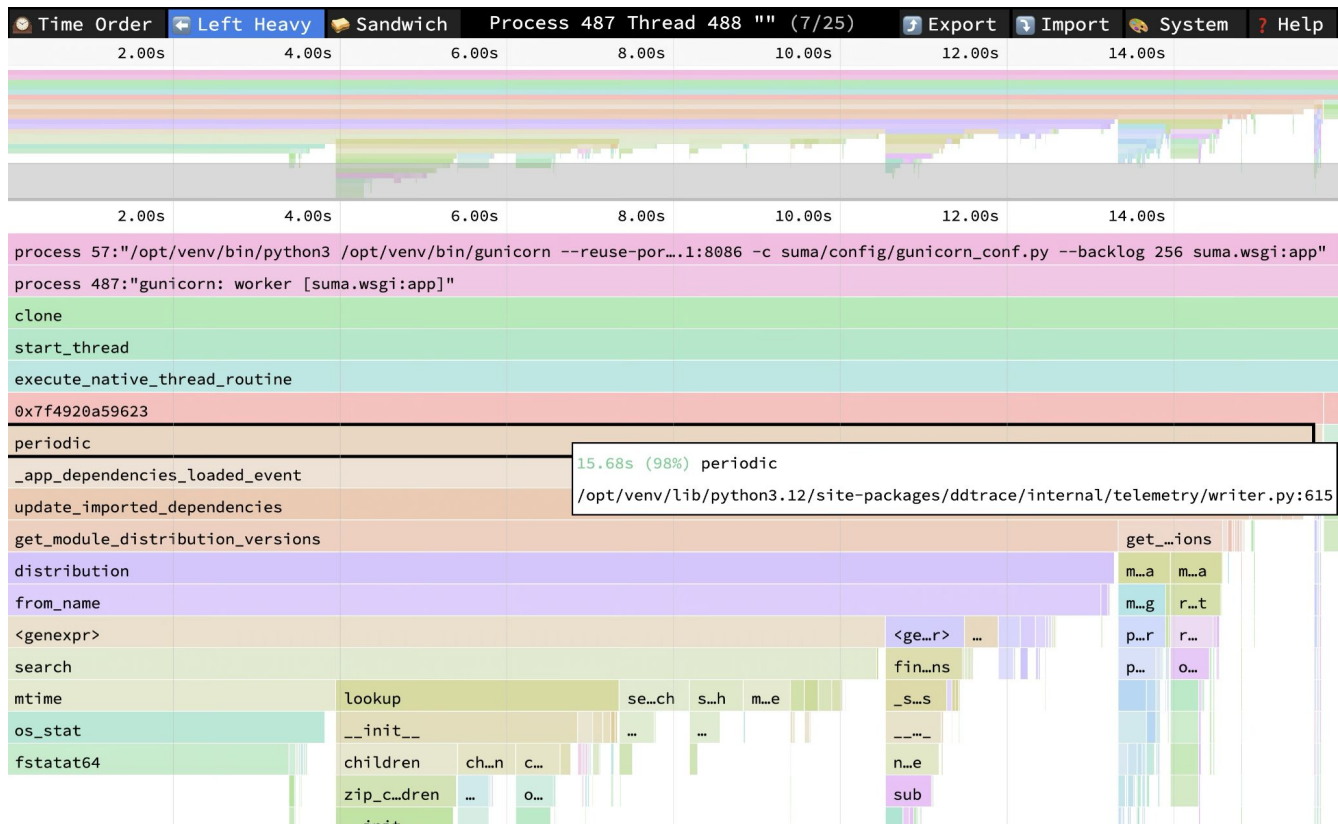
py-spy manual attempt #1: ld_long_cache_refresh

s/o speedscope



py-spy auto attempt #4: Id_cache_refresh disabled

s/o speedscope click me



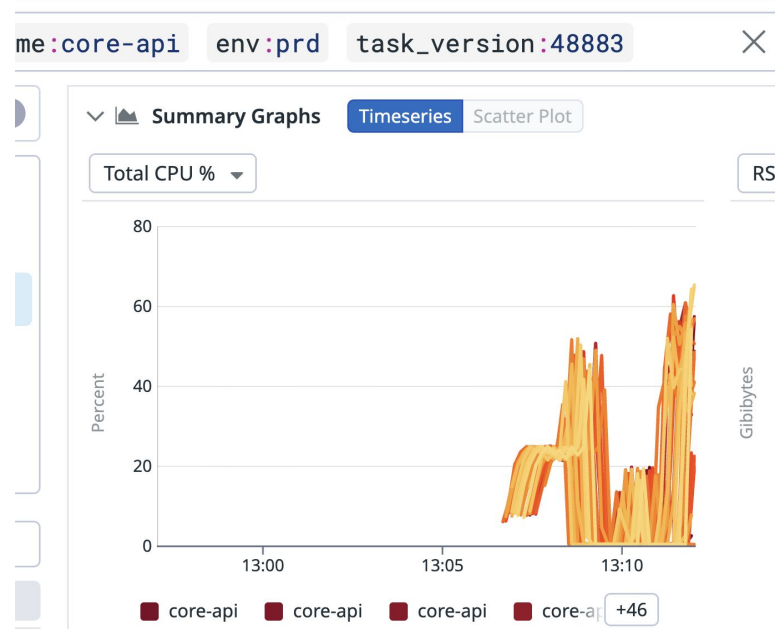
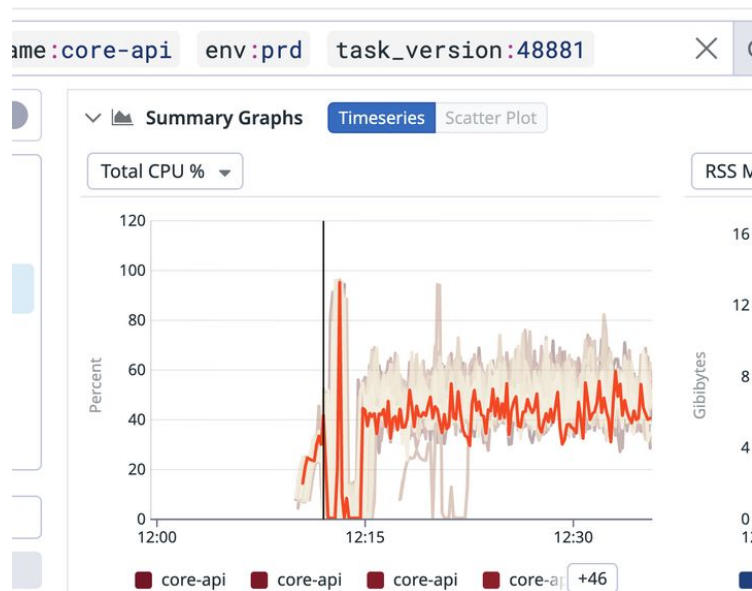
wtf is ddtrace-telemetry?

why datadog why

- “diagnostic telemetry data about ddtrace”... **useless**
- disabled via env vars

```
57 @cached(maxsize=1024)
58 v def get_module_distribution_versions(module_name: str) -> t.Optional[t.Tuple[str, str]]:
59     if not module_name:
60         return None
61
62     names: t.List[str] = []
63     pkgs = get_package_distributions()
64     dist_map = get_distributions()
65     while names == []:
66         # First try to resolve the module name from package distributions
67         version = dist_map.get(module_name)
68         if version:
69             return (module_name, version)
70         # Since we've failed to resolve, try to resolve the parent package
71         names = pkgs.get(module_name, [])
72         if not names:
73             p = module_name.rfind(".")
74             if p > 0:
75                 module_name = module_name[:p]
76             else:
77                 break
78     if len(names) != 1:
79         # either it was not resolved due to multiple packages with the same name
80         # or it's a multipurpose package (like '__pycache__')
81         return None
82     return (names[0], get_version_for_package(names[0]))
```

result?
job done



result?

happy people

 **Stephen Snow** VIP 08:48
good timing on this @umar, every minute counts for getting our containers healthy now due to #_inc-10-20-inc_41178-production-scaling-impaired-by-aws-capacity-constraint



Matt Shalhoub Nov 21st, 2025 at 14:42

love a fast rollback



Arman Vaziri Nov 26th, 2025 at 13:23

Wait Arman just to double check you wanted to also rollout a delay to growth bc of the same cpu spike issue?

yeah i had a pr almost ready before i remembered about this channel



<https://github.com/ramp/growth/pull/5658>

hardest part?

exfiltrating data from aws, thx chatgpt

can i use the boto3 python client from the container to send the file somewhere else?? please help im losing my mind over this

whats the quickest way to get the profiling file off of the container

FIX THIS IT STILL DOESNT WORK

```
~ $  
~ $ aws ecs execute-command \  
> --region us-east-1 \  
> --cluster dev-sum-useast1-common \  
> --task f768c5ee108d41b4806dfd35f5a61607 \  
> --container core-api \  
> --interactive \  
> --command "bash -lc 'base64 -w0 /opt/app/hurricane.json  
2>/dev/null || base64 /opt/app/hurricane.json | tr -d \"\\n\\\"\" \"\" \  
> 2>/dev/null | tr -d '\\r\\n' | tr -d -c 'A-Za-z0-9+/' | base64 -d >  
hurricane.json"
```

base64: invalid input

```
~ $
```

Thank you!

- ssnow, joe & anthony for words of wisdom
- kevinhu for sitting next to me while i debugged
- jamie wong for building speedscope

