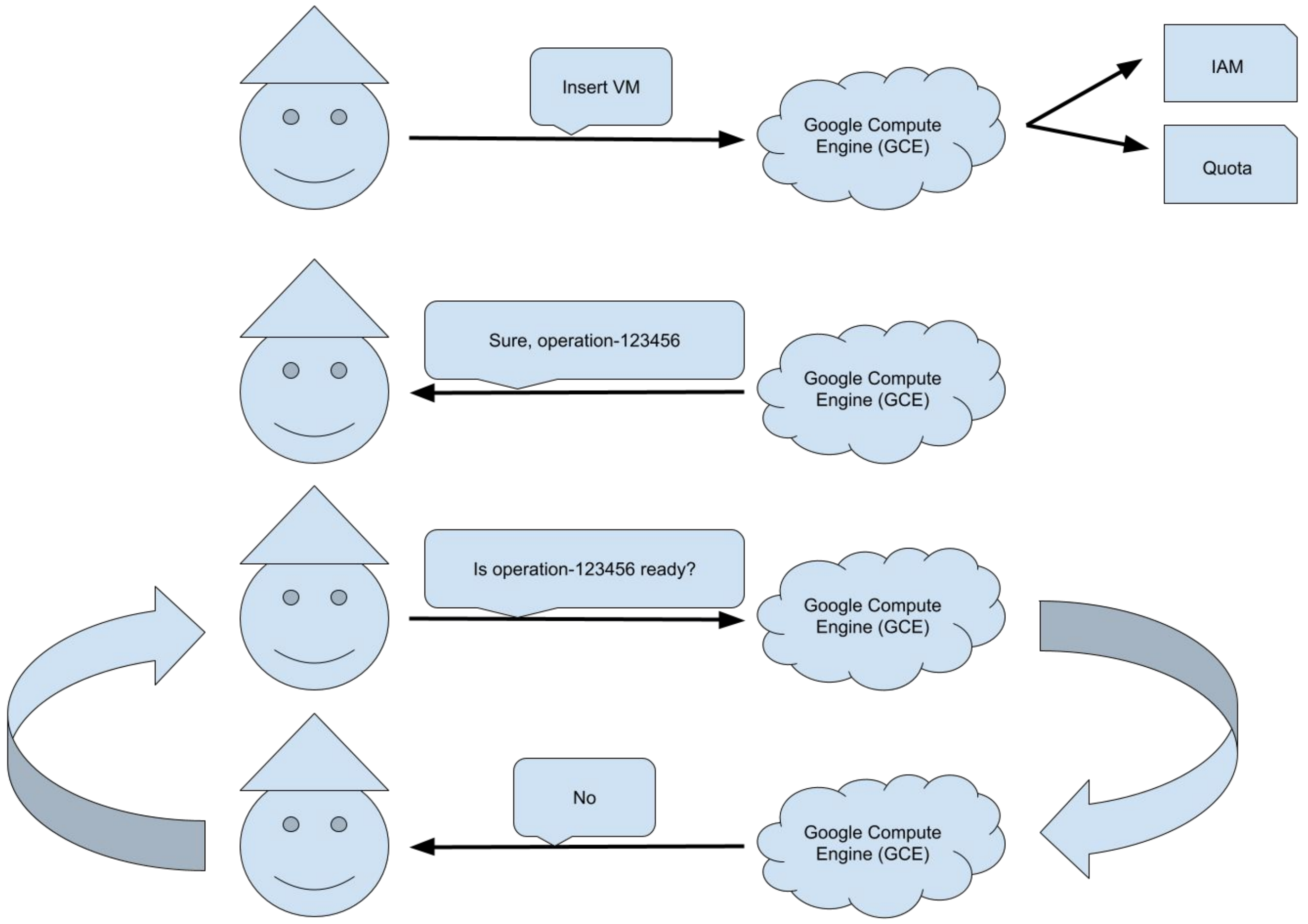
Google

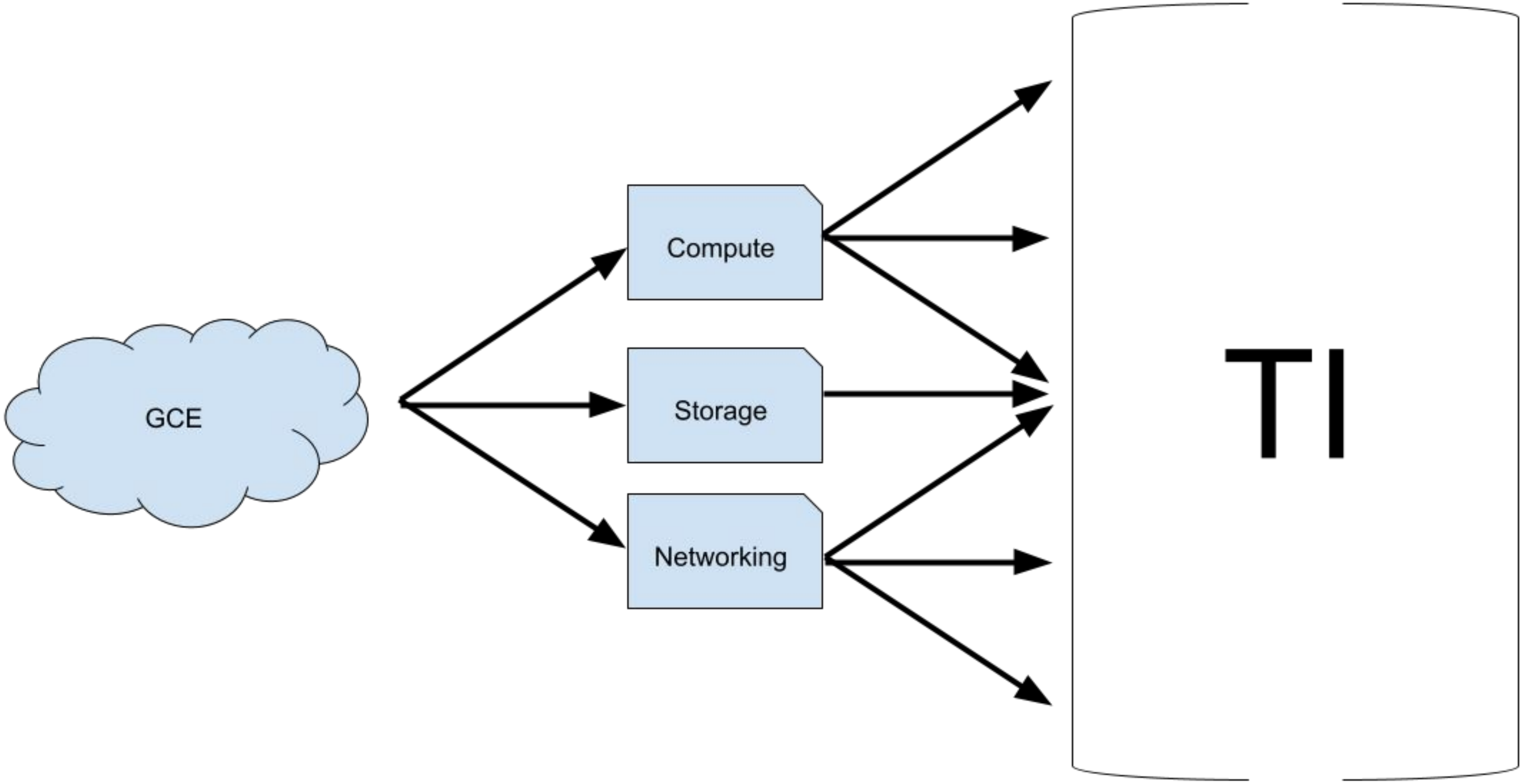
Going from 30 to 30 million SLOs

Alex Palcuie

sre.google

Site Reliability Engineering


Insert VM
Google Compute Engine (GCE)
IAM
Quota
Sure, operation-123456
Google Compute Engine (GCE)
Is operation-123456 ready?
Google Compute Engine (GCE)
No
Google Compute Engine (GCE)


GCE
Compute
Storage
Networking
TI

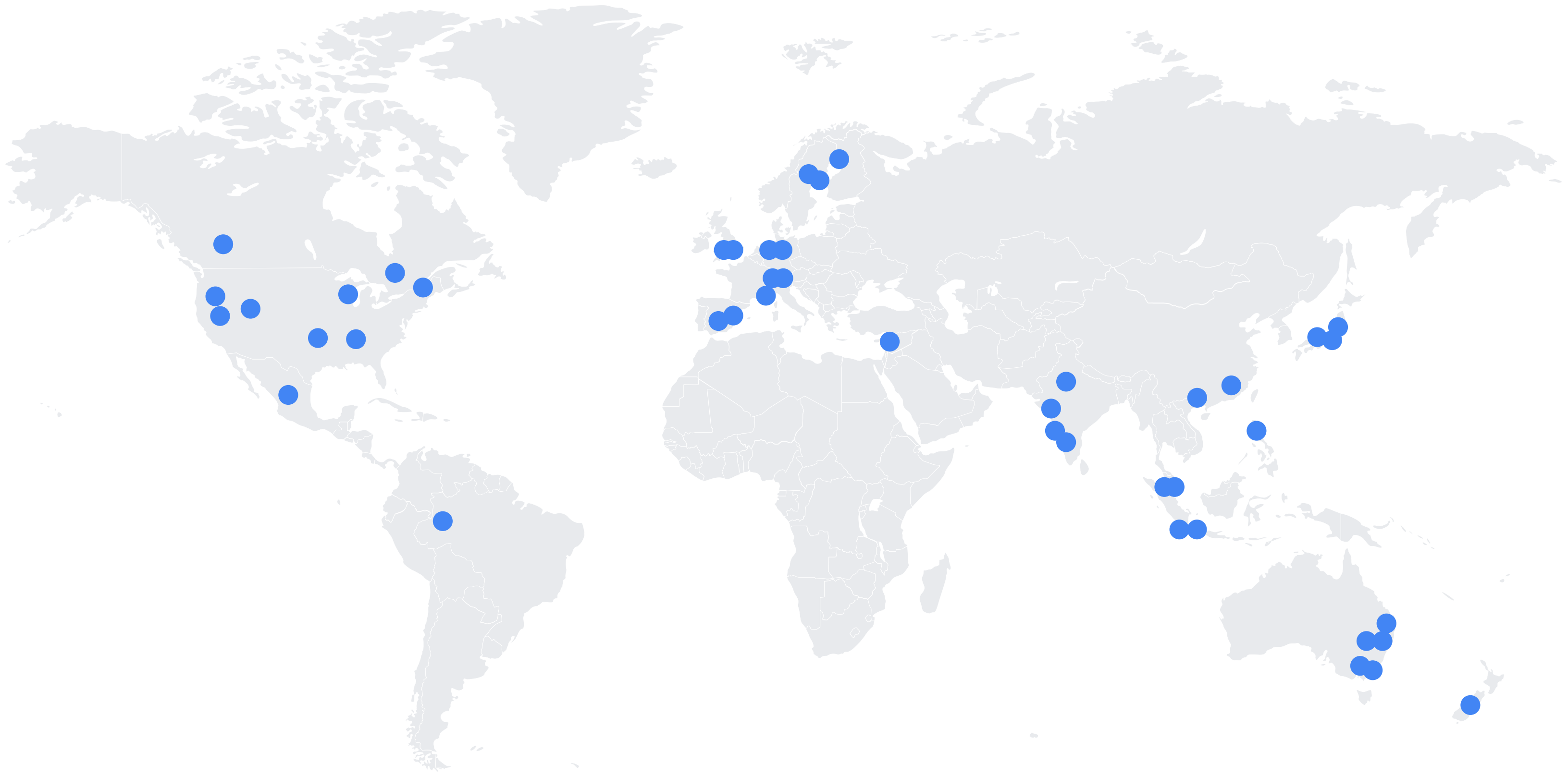
GCE Control Plane

Resources

- Instance
- Instance Group Manager
- Disk
- Snapshot
- Image
- Autoscaler
- Network
- Subnetwork
- Address
- Forwarding Rule
- Firewall
- ...

Methods

- Insert
- Get
- List
- Aggregated List
- Delete
- Patch
- ...



Google Cloud

Service Level **Indicator** (SLI)

Service Level **Objective** (SLO)

Service Level **Agreement** (SLA)

target availability = good requests / total requests

99.95% = 9,995 good requests / 10,000 requests

Latency SLOs "tricks"

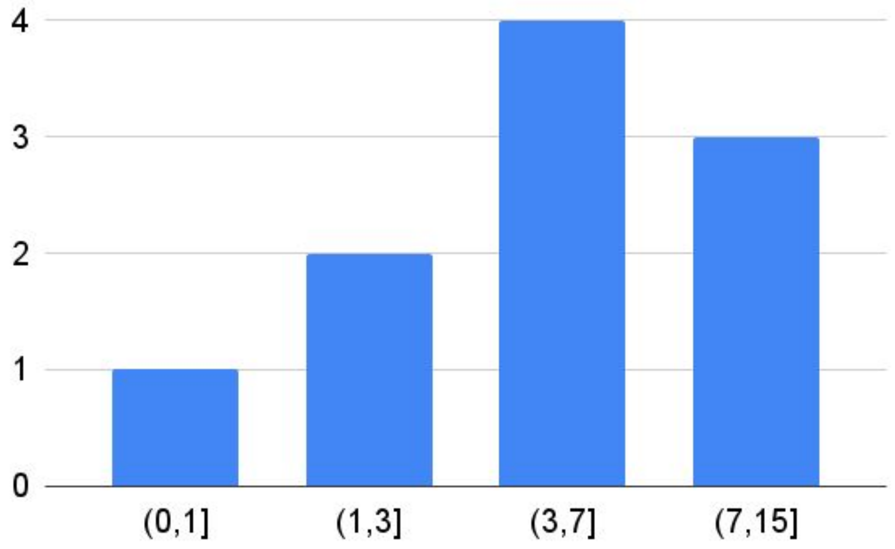
P90 compute.instances.get <= 10 seconds

Request No	Latency Seconds	Percentile
1	1	P10
2	2	P20
3	3	P30
4	4	P40
5	5	P50
6	6	P60
7	7	P70
8	8	P80
9	9	P90

Latency SLOs "tricks"

P90 compute.instances.get <= 10 seconds

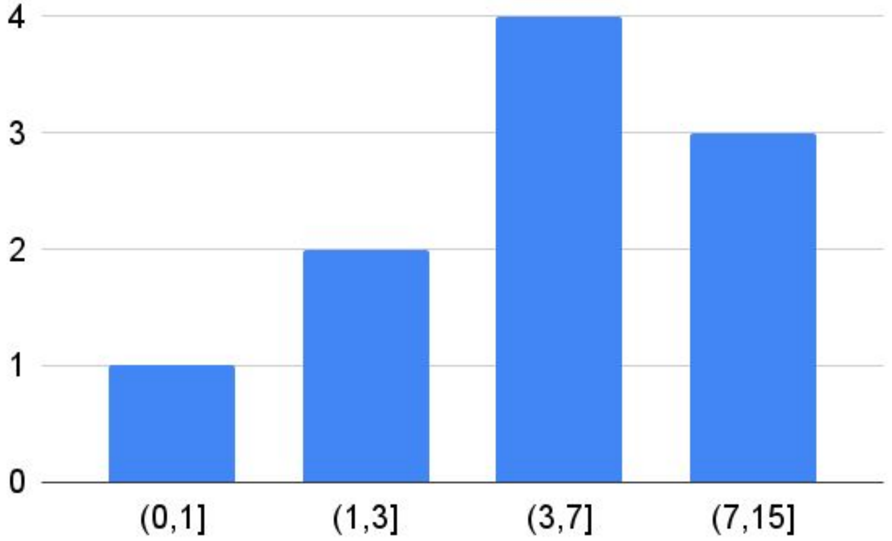
Request No	Latency Seconds	Percentile
1	1	P10
2	2	P20
3	3	P30
4	4	P40
5	5	P50
6	6	P60
7	7	P70
8	8	P80
9	9	P90



Latency SLOs "tricks"

P90 compute.instances.get <= 10 seconds

Request No	Latency Seconds	Percentile
1	1	P10
2	2	P20
3	3	P30
4	4	P40
5	5	P50
6	6	P60
7	7	P70
8	8	P80
9	**14**	P90



target = fast requests / total requests

fast request is a request within target latency

For P90 set target to 90%

Latency SLOs "tricks"

P90 compute.instances.get <= 10 seconds

Request No	Latency Seconds	Percentile
1	1	P10
2	2	P20
3	3	P30
4	4	P40
5	5	P50
6	6	P60
7	7	P70
8	8	P80
9	**14**	P90

8 / 9 = 88% "availability"

Request No	Latency Seconds	Percentile
1	1	P10
2	2	P20
3	3	P30
4	4	P40
5	5	P50
6	6	P60
7	7	P70
8	8	P80
9	9	P90

9 / 9 = 100% "availability"

Latency SLO tricks

API	Target P90
compute.instances.get	10s
compute.instances.list	30s
compute.instances.insert	60s

API	Latency	Status
compute.instances.get	5	✅
compute.instances.get	3	✅
compute.instances.get	9	✅
compute.instances.list	25	✅
compute.instances.insert	55	✅
compute.instances.insert	40	✅
compute.instances.get	15	❌
compute.instances.get	2	✅
compute.instances.get	4	✅
compute.instances.get	4	✅

9 / 10 = 90% "availability"

The original ~30 SLOs

us-central1
availability
typical latency
tail latency

us-central1-a
availability
typical latency
tail latency

us-central1-b
availability
typical latency
tail latency

us-central1-c
availability
typical latency
tail latency

europe-west1
availability
typical latency
tail latency

europe-west1-a
availability
typical latency
tail latency

europe-west1-b
availability
typical latency
tail latency

europe-west1-c
availability
typical latency
tail latency

asia-east1
availability
typical latency
tail latency

asia-east1-a
availability
typical latency
tail latency

asia-east1-b
availability
typical latency
tail latency

asia-east1-c
availability
typical latency
tail latency

API	SLO	us-central1-a	us-central1-b	us-central1-c	us-central1-f	us-east1-b	us-east1-c	us-east1-d	europe-west1-b	europe-west1-c	europe-west1-d	us-west1-a	us-west1-b	us-west1-c	europe-west4-a	europe-west4-b	europe-west4-c	us-east4-a	us-east4-b	us-east4-c	europe-west3-a	europe-west3-b	europe-west3-c	europe-west2-a	europe-west2-b	europe-west2-c	asia-east1-a	asia-east1-b	asia-east1-c	asia-southeast1-a	...
compute.instanceGroupManagers.listManagedInstances	availability																														
compute.instanceGroupManagers.listManagedInstances	tail_latency																														
compute.instanceGroupManagers.listManagedInstances	typical_latency																														
compute.instanceGroupManagers.list	availability																														
compute.instanceGroupManagers.list	tail_latency																														
compute.instanceGroupManagers.list	typical_latency																														
compute.instances.list	availability																														
compute.instances.list	tail_latency																														
compute.instances.list	typical_latency																														
compute.disks.list	availability																														
compute.disks.list	tail_latency																														
compute.disks.list	typical_latency																														
compute.instanceGroupManagers.get	availability																														
compute.instanceGroupManagers.get	tail_latency																														
...																															

GCE Complexity Growth

2016

- 43 Resources
- 97 API methods
- 9 regions
- 20 zones

2021

- 81 Resources
- 423 API methods
- 33 regions
- 96 zones

“They are huge. They are like a giant which lumbers around while you are a gnat. You are nothing to them.

This becomes obvious when talking about some problem you experienced at the hands of their system. The whole time, their dashboard stayed green because from their point of view, they had tremendous availability. We're talking 99.999% here! Totally legit!”

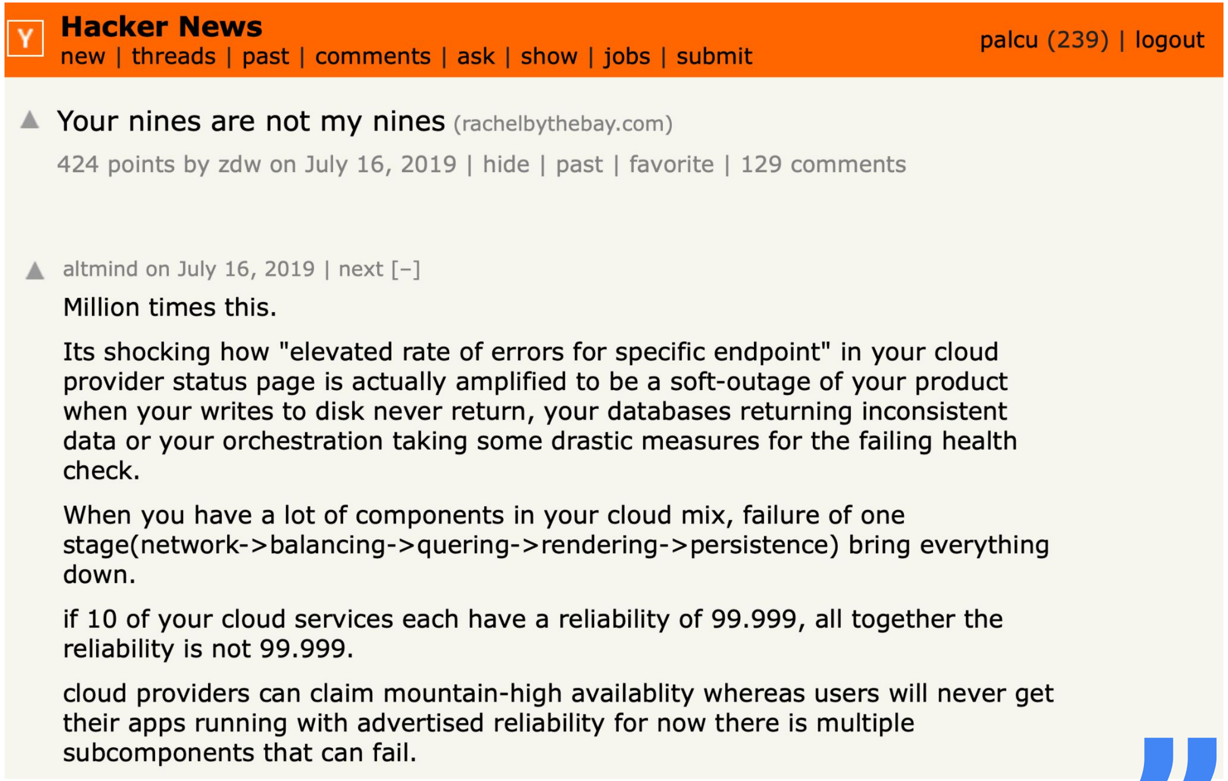
Rachel Kroll
https://rachelbythebay.com/w/2019/07/15/giant/

“Meanwhile, you were having a really bad day. Nothing was working. Your business was in shambles. Your customers were at your throat yelling for action, and all you could do is point at the vendor. What happened?

Well, this is the point where you find out that their "99.999%" availability is for their entire system. They see that, and they're good. It's not a problem! Everything is fine.

You are the bug on the windscreen of the locomotive. The train has no idea you were ever there.”

Rachel Kroll
https://rachelbythebay.com/w/2019/07/15/giant/


Hacker News
new | threads | past | comments | ask | show | jobs | submit
palcu (239) | logout

▲ Your nines are not my nines (rachelbythebay.com)
424 points by zdw on July 16, 2019 | hide | past | favorite | 129 comments

▲ altmind on July 16, 2019 | next [–]

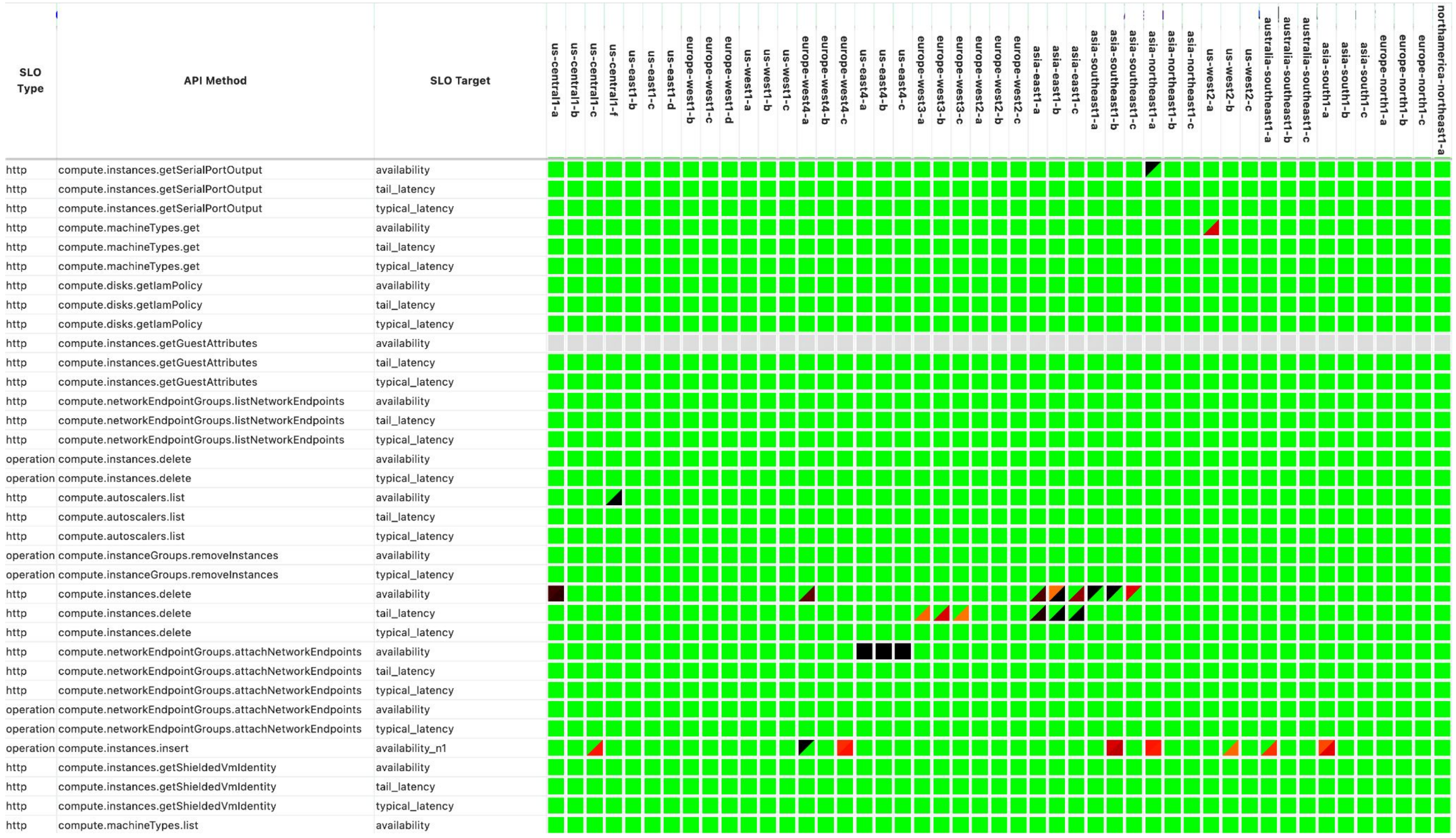
Million times this.

Its shocking how "elevated rate of errors for specific endpoint" in your cloud provider status page is actually amplified to be a soft-outage of your product when your writes to disk never return, your databases returning inconsistent data or your orchestration taking some drastic measures for the failing health check.

When you have a lot of components in your cloud mix, failure of one stage(network->balancing->quering->rendering->persistence) bring everything down.

if 10 of your cloud services each have a reliability of 99.999, all together the reliability is not 99.999.

cloud providers can claim mountain-high availablity whereas users will never get their apps running with advertised reliability for now there is multiple subcomponents that can fail.



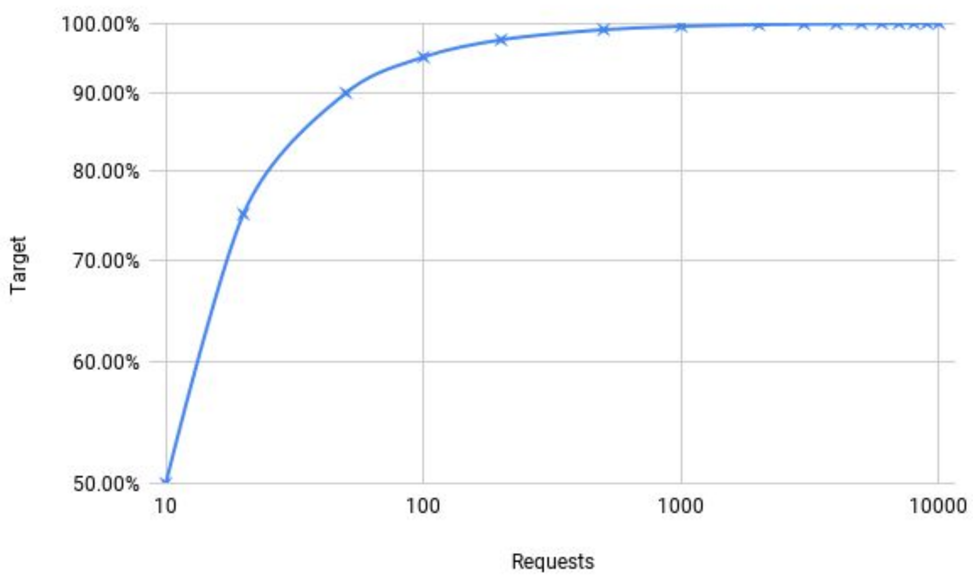
SLO Type	API Method	SLO Target	us-central1-a	us-central1-b	us-central1-c	us-central1-f	us-east1-b	us-east1-c	us-east1-d	europe-west1-b	europe-west1-c	europe-west1-d	us-west1-a	us-west1-b	us-west1-c	europe-west4-a	europe-west4-b	europe-west4-c	us-east4-a	us-east4-b	us-east4-c	europe-west3-a	europe-west3-b	europe-west3-c	europe-west2-a	europe-west2-b	europe-west2-c	asia-east1-a	asia-east1-b	asia-east1-c	asia-southeast1-a	asia-southeast1-b	asia-southeast1-c	asia-northeast1-a	asia-northeast1-b	asia-northeast1-c	us-west2-a	us-west2-b	us-west2-c	australia-southeast1-a	australia-southeast1-b	australia-southeast1-c	asia-south1-a	asia-south1-b	asia-south1-c	europe-north1-a	europe-north1-b	europe-north1-c	northamerica-northeast1-a
http	compute.instances.getSerialPortOutput	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getSerialPortOutput	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getSerialPortOutput	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.machineTypes.get	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.machineTypes.get	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.machineTypes.get	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.disks.getIamPolicy	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.disks.getIamPolicy	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.disks.getIamPolicy	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getGuestAttributes	availability	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
http	compute.instances.getGuestAttributes	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getGuestAttributes	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.listNetworkEndpoints	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.listNetworkEndpoints	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.listNetworkEndpoints	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.instances.delete	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.instances.delete	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.autoscalers.list	availability	🟩	🟩	🟩	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.autoscalers.list	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.autoscalers.list	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.instanceGroups.removeInstances	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.instanceGroups.removeInstances	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.delete	availability	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	🟧	🟥	⬛	⬛	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.delete	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟧	🟥	🟧	🟩	🟩	🟩	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.delete	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.attachNetworkEndpoints	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.attachNetworkEndpoints	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.networkEndpointGroups.attachNetworkEndpoints	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.networkEndpointGroups.attachNetworkEndpoints	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.networkEndpointGroups.attachNetworkEndpoints	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
operation	compute.instances.insert	availability_n1	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟩	🟩	🟩	🟧	🟩	🟥	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getShieldedVmIdentity	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getShieldedVmIdentity	tail_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.instances.getShieldedVmIdentity	typical_latency	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
http	compute.machineTypes.list	availability	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

99.95% reliability

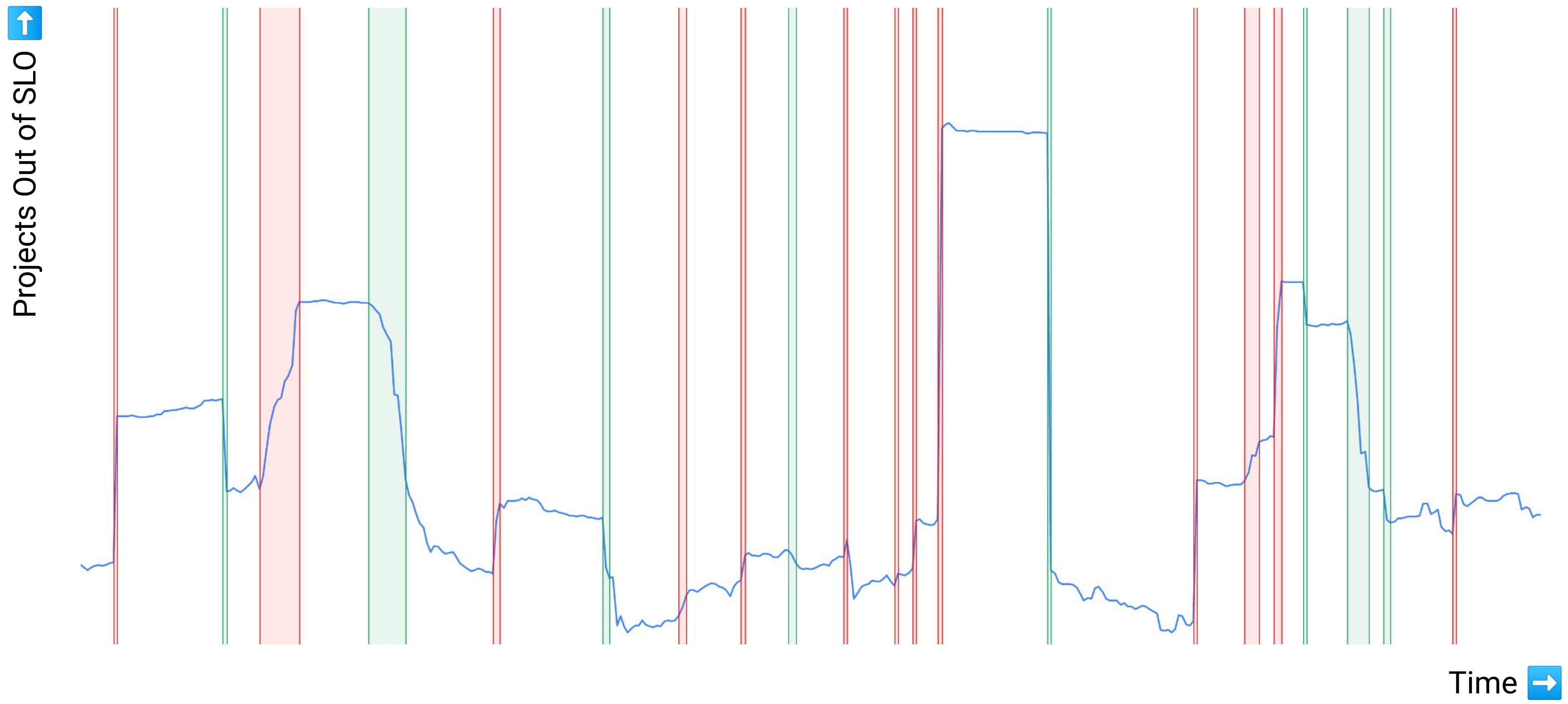
10,000 requests - **5** errors
20,000 requests - **10** errors
40,000 requests - **20** errors

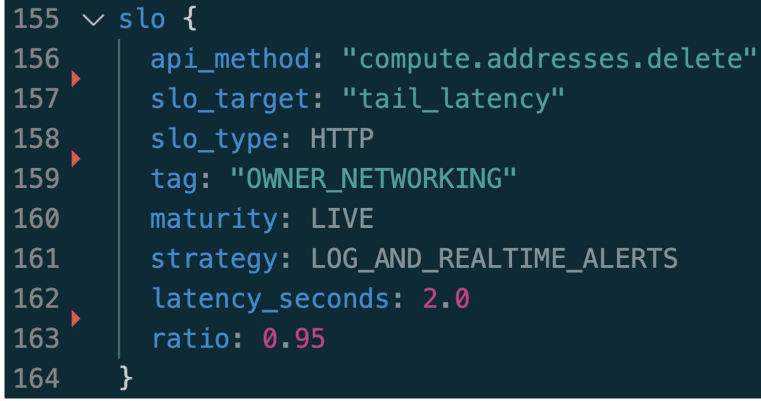
1,000 requests - **1** error

The rule of 5 errors

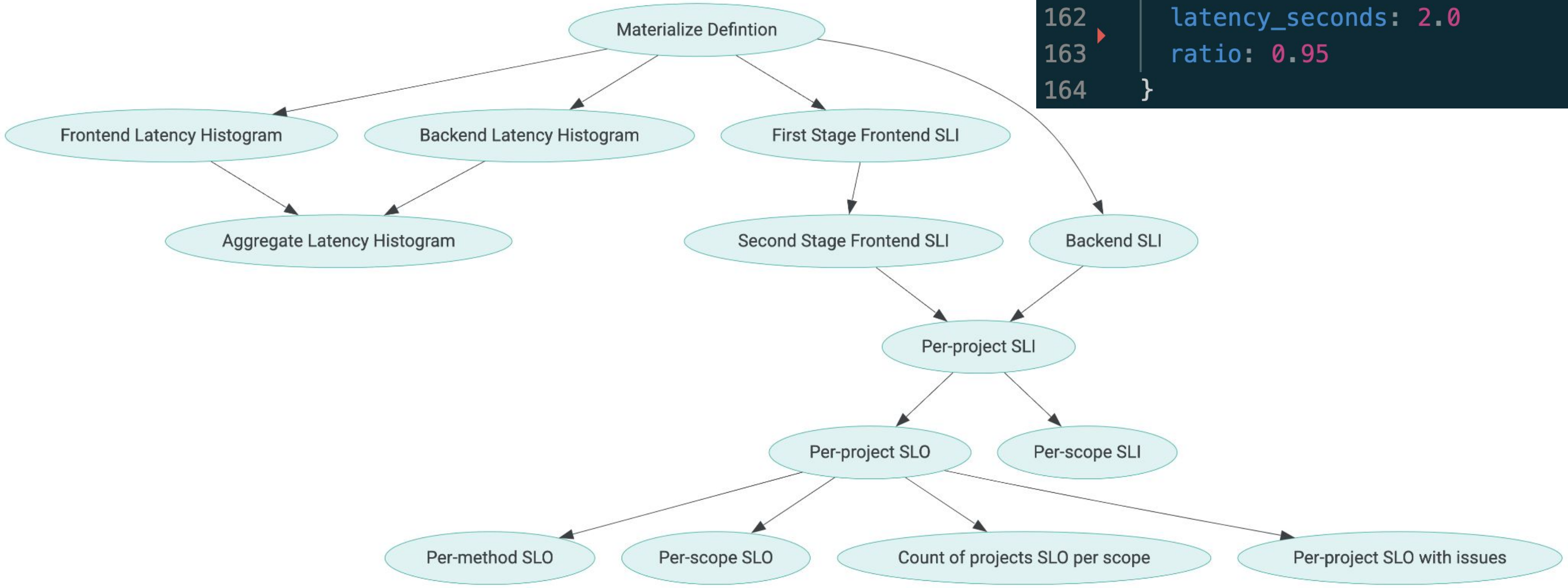


Requests	Actual errors	Target	Errors	Success
10	50.00%	50.00%	5	5
20	25.00%	75.00%	5	15
50	10.00%	90.00%	5	45
100	5.00%	95.00%	5	95
200	2.50%	97.50%	5	195
500	1.00%	99.00%	5	495
1000	0.50%	99.50%	5	995
2000	0.25%	99.75%	5	1995
3000	0.17%	99.83%	5	2995
4000	0.13%	99.88%	5	3995
5000	0.10%	99.90%	5	4995
6000	0.08%	99.92%	5	5995
7000	0.07%	99.93%	5	6995
8000	0.06%	99.94%	5	7995
9000	0.06%	99.94%	5	8995
10000	0.05%	99.95%	5	9995


⬆️ Projects Out of SLO
Time ➡️



```
slo {
  api_method: "compute.addresses.delete"
  slo_target: "tail_latency"
  slo_type: HTTP
  tag: "OWNER_NETWORKING"
  maturity: LIVE
  strategy: LOG_AND_REALTIME_ALERTS
  latency_seconds: 2.0
  ratio: 0.95
}
```


Materialize Defintion
Frontend Latency Histogram
Backend Latency Histogram
First Stage Frontend SLI
Aggregate Latency Histogram
Second Stage Frontend SLI
Backend SLI
Per-project SLI
Per-project SLO
Per-scope SLI
Per-method SLO
Per-scope SLO
Count of projects SLO per scope
Per-project SLO with issues

Worst SLOs for which we burnt the budget and we don't have a bug

Scope Type	Scope Name	Api Method	Slo Type	Slo Target	Ratio Of Slo Used	Window Length Days	Bad Requests	Total Requests	Out Of Slo Projects	SLO Link
region	us-central1	compute.regionInstances.recommendLocations	http	availability	2.18	30	18,000	16,000,000	502	Drilldown...
zone	us-central1-a	compute.instances.insert	operation	availability_n1	4.02	30	110,000	40,000,000	323	Drilldown...
global	global	compute.projects.setCommonInstanceMetadata	http	availability	1.08	30	7,000	13,000,000	312	Drilldown...
global	global	compute.networks.addPeering	http	availability	1.14	30	10,000	16,000,000	303	Drilldown...
zone	us-central1-f	compute.instances.getShieldedVmIdentity	http	availability	1.29	30	10,000	12,000,000	288	Drilldown...
region	us-west1	compute.regionInstanceGroupManagers.insert	http	availability	1.45	30	2,000	2,800,000	175	Drilldown...

Worst SLOs for which we are still in budget, but a lot of projects are not

Scope Type	Scope Name	Api Method	Slo Type	Slo Target	Ratio Of Slo Used	Window Length Days	Bad Requests	Total Requests	Out Of Slo Projects	SLO Link
zone	us-central1-a	compute.instanceGroupManagers.get	http	availability	0.21	30	1,400	2,000,000	289	Drilldown...
zone	us-central1-b	compute.instances.insert	operation	availability_n1	0.15	30	4,000	23,000,000	286	Drilldown...
zone	us-central1-a	compute.instances.get	http	availability	0.18	30	40,000,000	100,000,000	219	Drilldown...
region	us-central1	compute.addresses.insert	http	availability	0.46	30	10,000,000	30,000,000	204	Drilldown...
zone	us-central1-a	compute.instanceGroupManagers.list	http	availability	0.17	30	1,000,000	100,000,000	204	Drilldown...
global	global	compute.autoscalers.aggregatedList	http	availability	0.16	30	4,000	2,000,000,000	180	Drilldown...

Worst offending bugs

Out Of Slo Projects	Bug Link	Api Method	Slo Type	Slo Target	Out of Slo Scopes
804	204726573	compute.instances.getGuestAttributes	http	availability	120
198	187519918	compute.disks.insert	http	availability	120
184	185914369	compute.instances.attachDisk	http	availability	120
136	174667773	compute.forwardingRules.insert	http	availability	40
131	185914369	compute.instances.detachDisk	http	availability	120

Thank you!