User Margin in Per-Usage billing setup

APIFY

What this will be about

- Matej Sestak
  - Doing Data at Apify
  - Background in CS and ML
  - Transitioned into Data engineering/analytics

- This talk will give you insight into:
  - Why you should care about your product margin
  - What pitfalls does usage-base billing present
  - How did we went about solving this problem at Apify

Apify

Cloud platform that provides specialized infrastructure and services for web scraping and browser automation

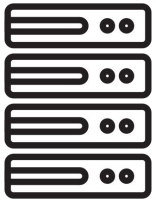
crawlee

APIFY

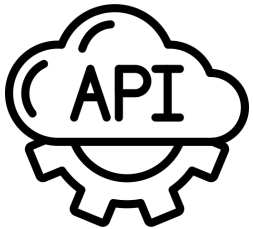
Apify in Numbers

>1PB of data processed a month

>15M jobs automated a month

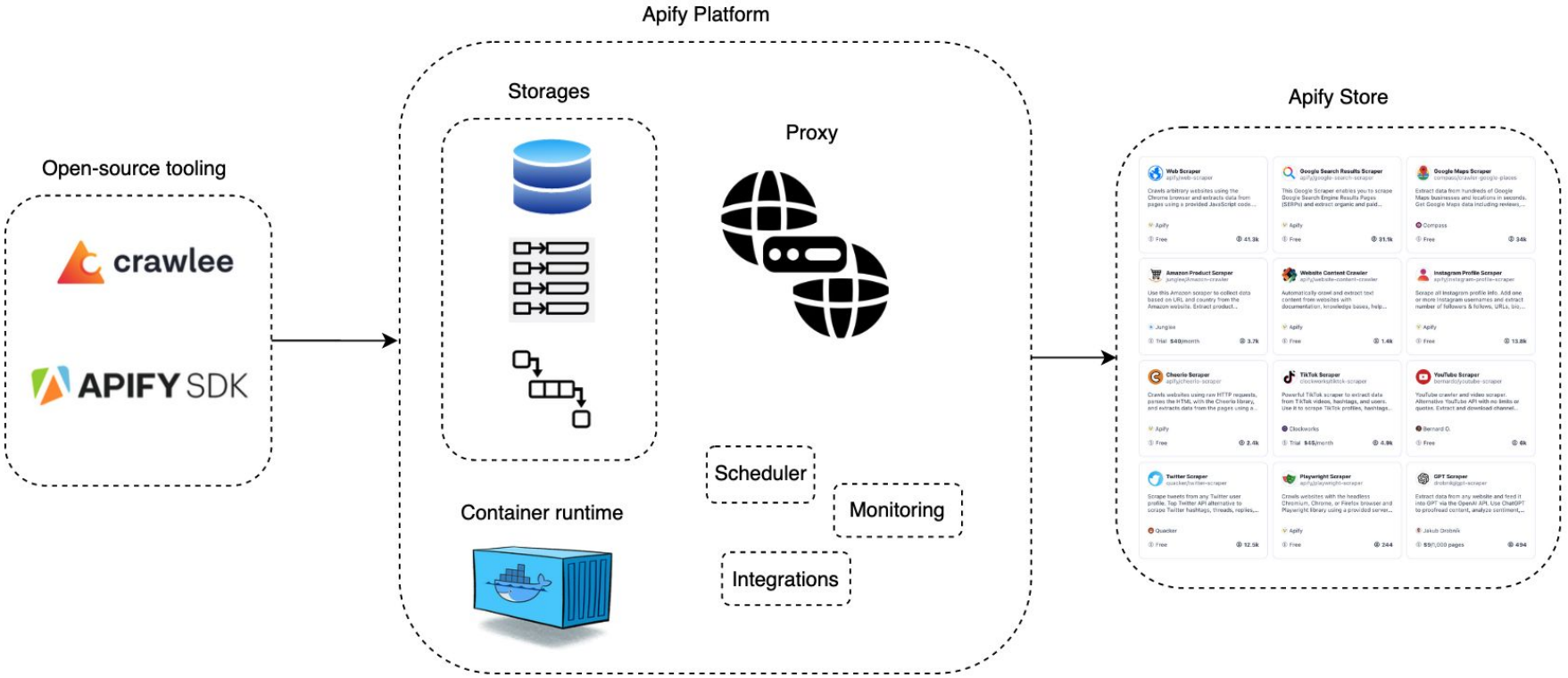


Up to 10TB of RAM used over 100s of EC2 instances



Up to 500k of API request handled per minute

Apify product offering



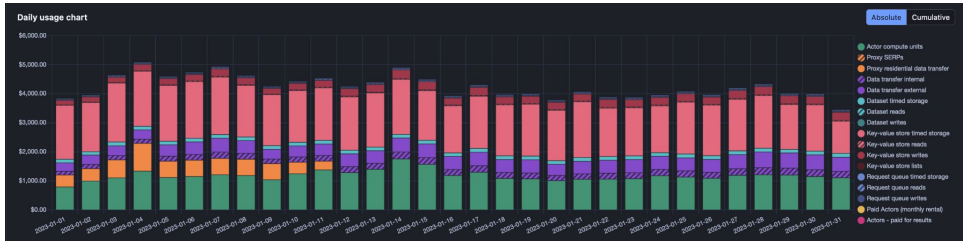
Apify product offering - pricing

Actor runs

Compute units (CU) ⓘ 1 GB of RAM / hour	**10 CUs** included then **$0.4** / CU	**100 CUs** included then **$0.4** / CU	**500 CUs** included then **$0.3** / CU	**1,000 CUs** included then **$0.25** / CU	**Custom**
RAM ⓘ	**8 GB**	**32 GB**	**128 GB**	**256 GB**	**Unlimited**
Max concurrent runs ⓘ	**25**	**32**	**128**	**256**	**Unlimited**
Rented Actors ⓘ	**Limited** ⓘ	Deducted from **prepaid usage**	Deducted from **prepaid usage**	Deducted from **prepaid usage**	**Custom**

Proxy

Residential proxies ⓘ	**$13** / GB	**$13** / GB	**$11** / GB	**$10** / GB	**Custom**
Datacenter proxies ⓘ	**5 IPs** included	**30 IPs** included then **$1** / IP	**200 IPs** included then **$0.8** / IP	**500 IPs** included then **$0.6** / IP	**Custom**
SERPs proxy ⓘ	**$3** / 1,000 SERPs	**$3** / 1,000 SERPs	**$2.5** / 1,000 SERPs	**$1.9** / 1,000 SERPs	**Custom**



Storages and Data transfer

Storage

Dataset					
Timed storage 1,000 GB-hours ⓘ	**$1.00**	**$1.00**	**$0.90**	**$0.80**	**Custom**
1,000 reads	**$0.0004**	**$0.0004**	**$0.00036**	**$0.00032**	**Custom**
1,000 writes	**$0.005**	**$0.005**	**$0.0045**	**$0.004**	**Custom**
Key-value store					
Timed storage 1,000 GB-hours ⓘ	**$1.00**	**$1.00**	**$0.90**	**$0.80**	**Custom**
1,000 reads	**$0.005**	**$0.005**	**$0.0045**	**$0.004**	**Custom**
1,000 writes	**$0.05**	**$0.05**	**$0.045**	**$0.04**	**Custom**
1,000 lists	**$0.05**	**$0.05**	**$0.045**	**$0.04**	**Custom**
Request queue					
Timed storage 1,000 GB-hours ⓘ	**$4.00**	**$4.00**	**$3.60**	**$3.20**	**Custom**
1,000 reads	**$0.004**	**$0.004**	**$0.0036**	**$0.0032**	**Custom**
1,000 writes	**$0.02**	**$0.02**	**$0.018**	**$0.016**	**Custom**

Data transfer

External / GB ⓘ	**$0.20**	**$0.20**	**$0.19**	**$0.18**	**Custom**
Internal / GB ⓘ	**$0.05**	**$0.05**	**$0.045**	**$0.04**	**Custom**

Built on top of AWS

Sales: What is the actual cost of 1 write to request queue?

Why we need to know our margin?

Marek Trunkát
CTO

"Are we measuring our users usage correctly?"

Jan Čurn
Co-founder & CEO

"Which part of business are working and which are not?"

Václav Růt
Head of Delivery

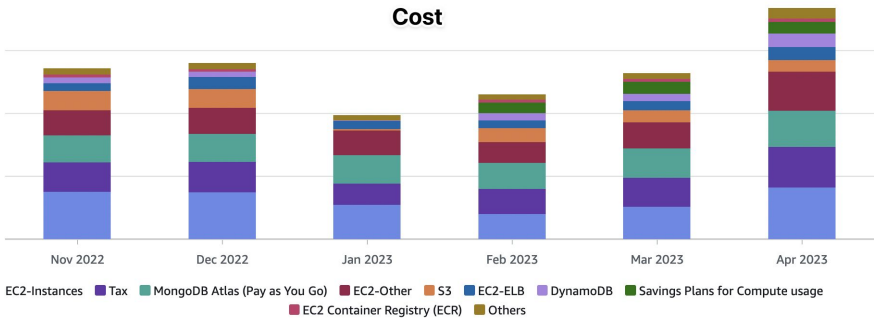
"Are we pricing our enterprise clients correctly?"

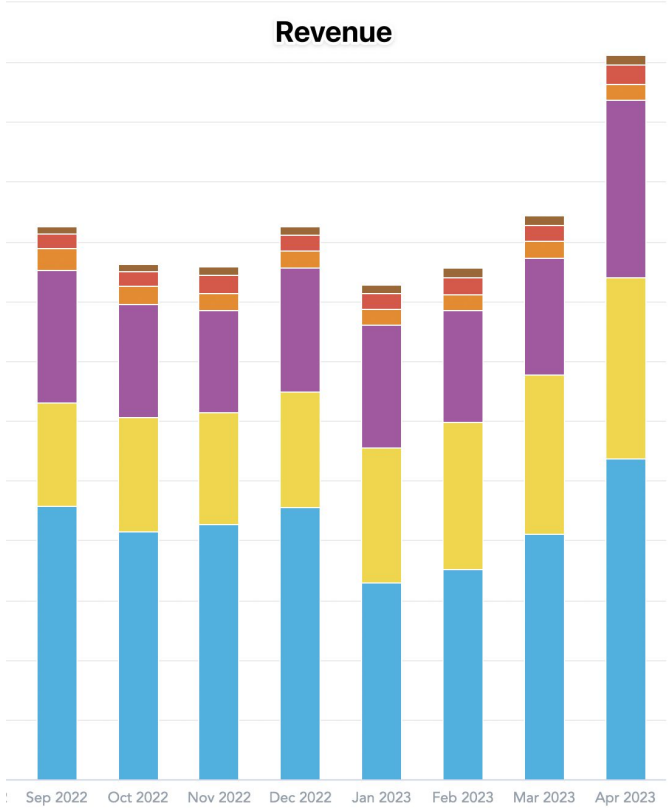
Jan Ženíšek
Head of Product

"How should we price our services?"

Company level margin

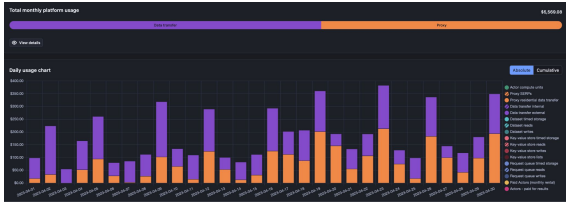
- Check *Cost Explorer* to get our cost
- Get revenue from finance
- (Revenue - Cost) / Revenue = Margin
- Enough for general price setting, optimization

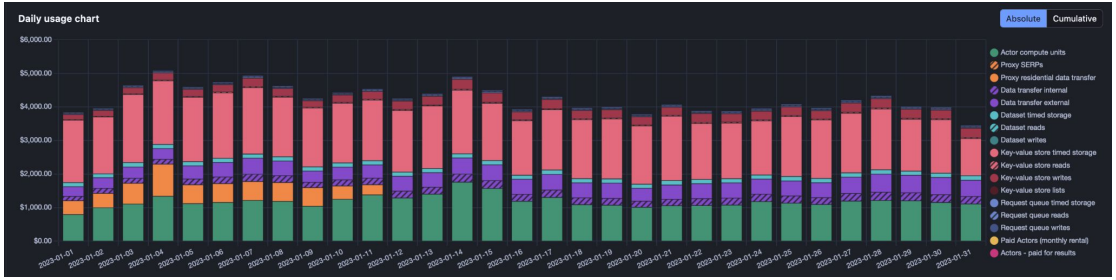


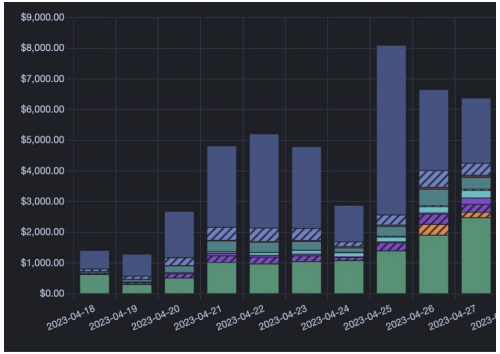


With wide product offering come difficulties

- Different customers have different usage patterns of our services
  - Compute heavy & low API rate
  - High API requests & low compute
  - Only proxy
  - High Data Transfer - media downloading





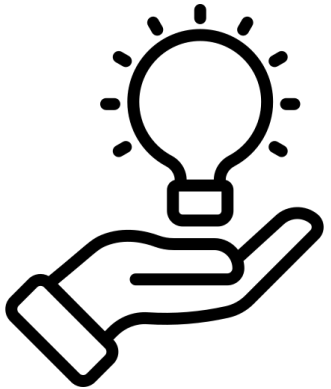


Cont.

- Different margin on different services
  - We were heavily optimizing last year, which led to changes in per service margins
- Everyone is cutting cost (including us)
  - We need to know, what services to focus on next
- We need to know, what is our margin on individual customer so we may offer them the best pricing while staying financially healthy
  - Discounts, yearly contracts incentives, ...

There are multiple solutions we could use

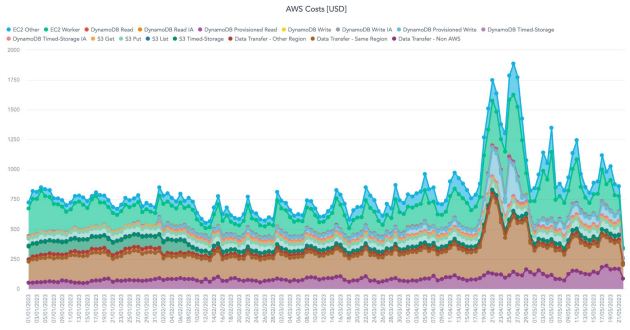
- Take company margin use it for all customers
  - Error-prone
- “On-premise” solution - each customer have dedicated account in AWS
  - 1700+ customers + 6000 active free users
  - Resource sharing - we use 64GB RAM worker machines, EKS, DBs, ...
- Use some a bit of data engineering and compute it correctly for each user

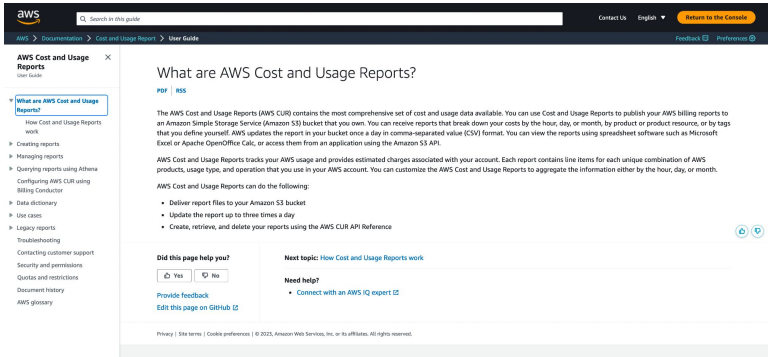


AWS Cost and Usage Report

> “The AWS Cost and Usage Reports (AWS CUR) contains the most comprehensive set of cost and usage data available. “

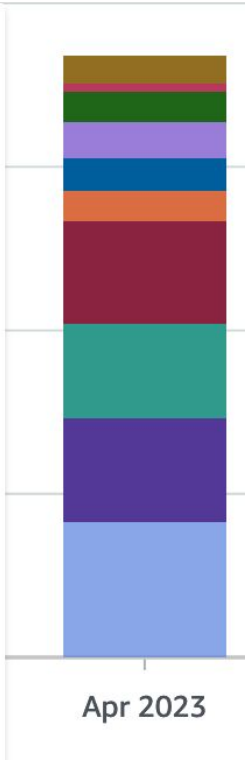
- Hourly granularity
- All the data you can get (200+ columns)
    - Usage, price, billing, discounts, product, reservation, …
- Allocation tags





Our cost structure

Apr 2023	
EC2-Instances	**$16,502.35**
Tax	$12,764.15
EC2-Other	$12,424.50
MongoDB Atlas (Pay as You Go)	$11,585.00
DynamoDB	$4,352.12
EC2-ELB	$4,044.06
S3	$3,760.22
Savings Plans for Compute usage	$3,749.39
EC2 Container Registry (ECR)	$937.18
Others	$3,426.83
Total costs	**$73,545.80**



logdna

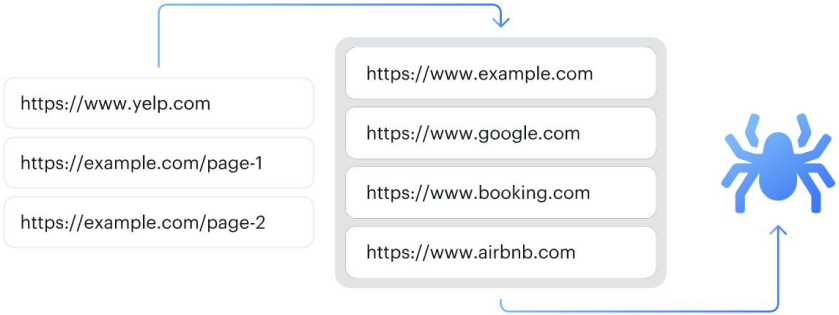
bright data

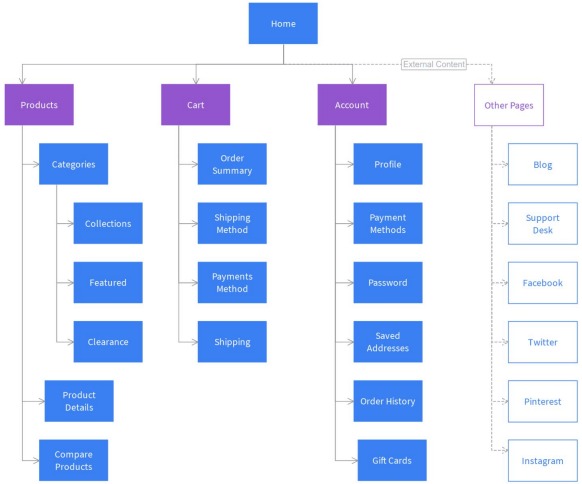
new relic

netnut

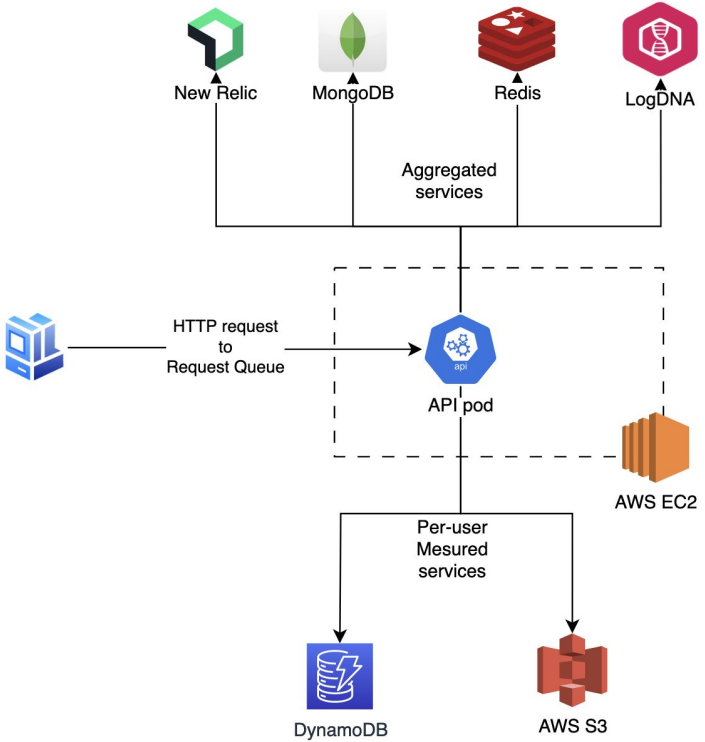
Case study: Apify Request queue

- Enqueue and retrieve requests such as URLs with an HTTP method and other parameters
- Combination of queue and database





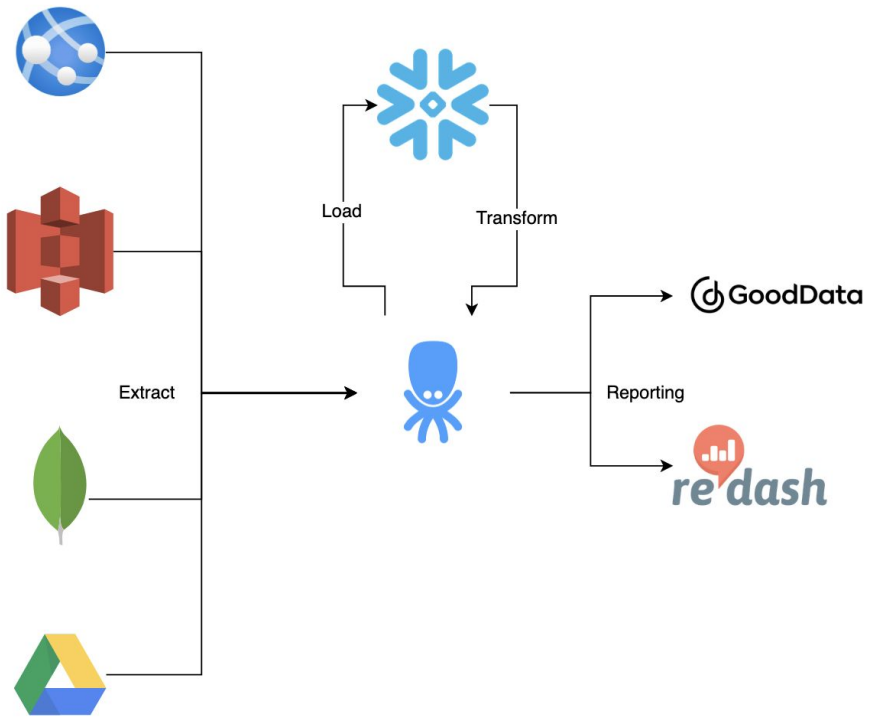
What is used when making a call to RQ



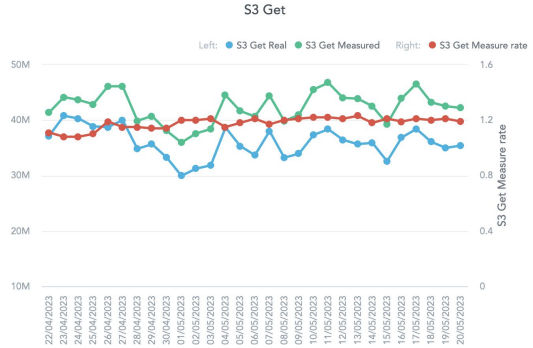

```
{
  "_id": "ppBFjmkrMaGdENQWp-2023-01-20",
  "date": {"$date": "2023-01-20T00:00:00.000Z"},
  "snapshots": [...],
  "stats": {
    "datasetS3GbyteHours": 119089.66501246877,
    "keyValueStoreS3GbyteHours": 1737334.8674574546,
    "requestQueueDynamodbGbyteHours": 4400.177836513421,
    "requestQueueS3GbyteHours": 1311.6438481732437
  },
  "userId": "ppBFjmkrMaGdENQWp"
}
```

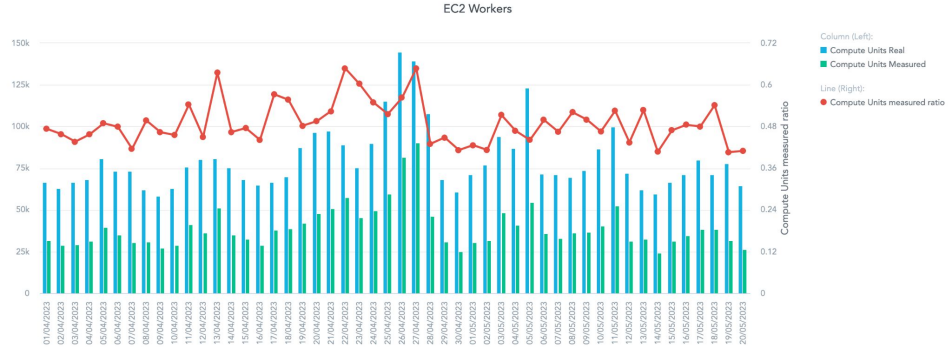
```
{
  "_id": "ppBFjmkrMaGdENQWp--2023-01-20",
  "date": {"$date": "2023-01-20T00:00:00.000Z"},
  "stats": {
    "deleteCount": 0,
    "dynamodbReadUnits": 775637,
    "dynamodbWriteUnits": 3367051,
    "headItemReadCount": 2465383,
    "readCount": 801590,
    "s3DeleteCount": 0,
    "s3GetCount": 727,
    "s3ListCount": 0,
    "s3PutCount": 1409,
    "writeCount": 2175410,
    "hardRemovalS3DeleteCount": 0,
    "hardRemovalS3ListCount": 1671
  },
  "userId": "ppBFjmkrMaGdENQWp"
}
```

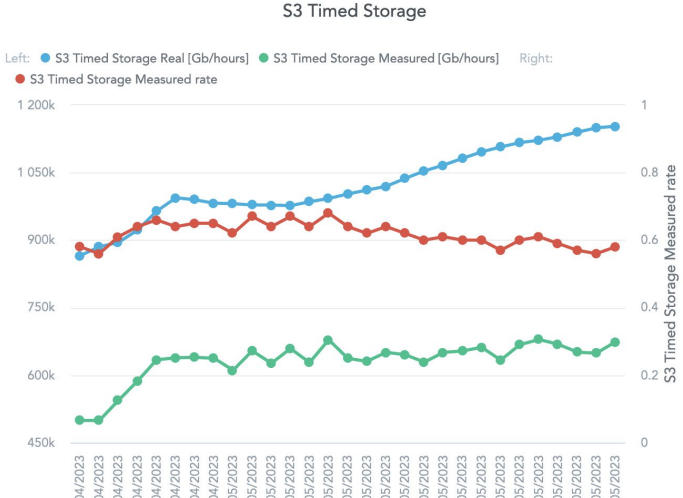
Data pipeline

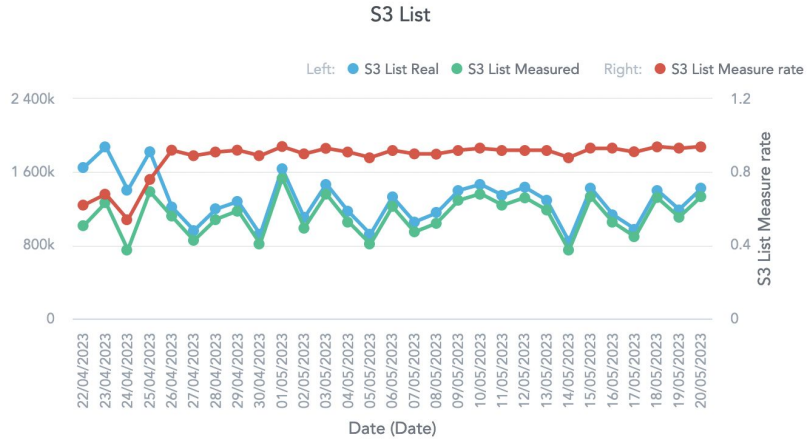


Usage measuring comparison

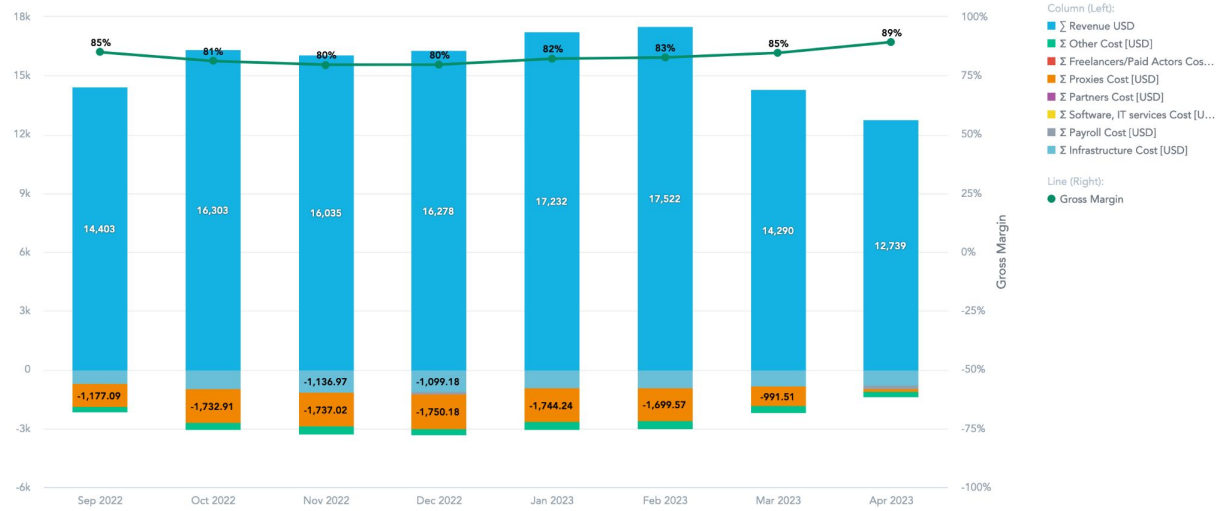








Margin per user



Month	User Id	∑ Revenue USD	Σ Cost of Revenue [USD]	Gross Profit USD	Σ Infrastructure Cost [USD]	Σ Proxies Cost [USD]	Σ Payroll Cost [USD]	Σ Software, IT services [USD]	Σ Partners Cost [USD]	Σ Freelancers Cost [USD]	Σ Paid Actors Cost [USD]	Σ Other Cost [USD]	Gross Margin
Feb 2023	5njhzC6bd87wpC39v	$49.00	$336.74	-$287.74	$4.06	$1.38	$330.12	$0.02	$0.00	$0.00	$0.00	$1.16	-587.22%
Mar 2023	5njhzC6bd87wpC39v	$106.31	$356.88	-$250.57	$9.27	$5.46	$339.45	$0.01	$0.00	$0.00	$0.00	$2.69	-235.70%
Apr 2023	5njhzC6bd87wpC39v	$51,763.09	$13,041.77	$38,721.32	$7,131.95	$1,778.97	$2,988.94	$0.01	$0.00	$0.00	$45.00	$1,096.90	74.80%

Month	User Id	SUM Infrastructure Cost [USD]	Σ AWS Cost [USD]	Non AWS Infrastructer Cost [USD]	AWS Data Transfer Cost [USD]	AWS EC2 API Cost [USD]	AWS EC2 Proxy Cost [USD]	AWS EC2 Worker Cost [USD]	AWS Other Cost [USD]	AWS Storage Cost [USD]
Feb 2023	pqQB4FWGoS8nsfMoo	$889.90	$570.60	$319.30	$101.93	$39.12	$71.49	$104.79	$27.97	$225.30

Our outcomes from the project

- Engineering:
  - Cost saving - we identified services, whose optimization will bring the biggest benefits to us
- Sales:
  - Deal re-negotiating - sales team knows how low they can set prices to keep our target profitability
- Product:
  - New pricing - introducing new platform pricing with knowing unit margin
- Project Delivery:
  - Better project monitoring

Sales: What is the actual cost of 1 write to request queue?

Sales: What is the actual cost of 1 write to request queue?

S3 + DynamoDB + API server + MongoDB + Observability = $0.0000039894 / write

With our base pricing ($0.00002), this gives us a margin of ~80%

Thank you