CONF42

Journey Beyond: AWS Quest for Excellence

Hlotenko Dmytro, Cloud Engineer
APA-Tech, March 2023

APA TECH

Dmytro Hlotenko (he/him)

- Cloud Engineer of APA-IT or “Mr. Amazon”
- AWS Community Builder – Cloud Operations
- AWS User Group Vienna Co-Leader
- Ukrainian IT Geek and Enthusiast
- M.Sc. in Telecom w/h (SUITT – Odesa, Ukraine, 2022)
- B.Sc in Business Management (SUITT, 2021)
- In IT since 2014, has a background in system engineering,networks, programming, operations, game design, e.t.c.
- Started with AWS in 2020, with a full focus since 2022
- 4x AWS Certified – hunting for the “Golden Jacket”
- Motorsport fan and photography hobbyist

aws sts get-caller-identity

Tonight!

- APA-IT and what do we do in AWS.
- What is the MediaContact-Plus – APA-OTS.
- Welcome onboard! But how?
  - How to know your workload.
  - EVO. Analysis and strategic re-architecture.
  - AWS SES as a foundation for the application.
  - The Swiss Army Knife
- It’s running! Or not? Ballade about running your stuff.
  - Why the easiest way may not be the optimal way.
  - Rightsizing, scaling, and the Red Hats!
  - Databases, surprises and finding the approach.
  - Has the Graviton Gravity?

1

APA-IT and what do we do in AWS.

APA-Tech is powering media production of Austria

Way more than just the press agency’s IT department.

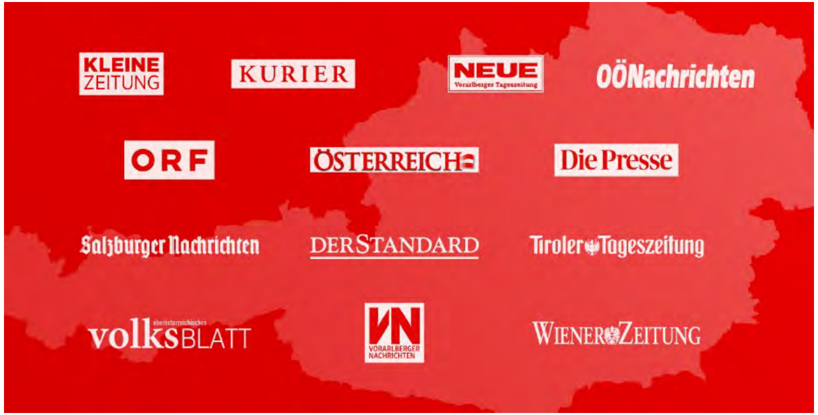
In-house:

APA

APA NEWS

APA COMM

External:



And even more, you won’t even suspect!

APA-Tech is powering media production of Austria

Way more than just the press agency’s IT department.

- AWS is the **only cloud** in APA-Tech with production-grade workloads since the takedown of Stratoscale Cloud.
- Which **opportunities** does the AWS bring to us?
  - Cloud Integrations and expansions for applications
  - Powers journalist mobile applications
  - ETL, Data Processing and Analytics
  - Disaster Recovery and critical system redundancy
  - Runs critical journalist applications
  - Media Publishing and Processing

2

What is the MediaContact-Plus – APA-OTS.

What is the APA-OTS & MediaContact-Plus

- **APA-OTS** makes your content visible.
  - 10k+ Journalists, 800 Press Offices, international transmission without the limits.
- **MediaContact-Plus** bridges your company with contacts all around the world.
  - Gives you time to concentrate on the content of your communication work.
  - Setup, distribute, and reach the top-grade contacts with just a few clicks.
  - Delivery problems are left in the past.
- **Rest is handled by us.**

What is the APA-OTS & MediaContact-Plus

- 50,000 **Journalists** and **blogger** contacts
- **97% Delivery Rate** – powered by AWS SES
  - Over 750k sends per month
  - Over 75,000 user sessions per month
- **100% resonance measurement** powered by APA-OTS PR-Desk
- Joint development of **APA-Tech** and external company
- **Deeply integrated** with other APA systems
- Powered by **AWS** since 2021. Runs exclusively on AWS.

3

AWS SES

AWS SES as a foundation for the application

Are we in contact?

- What I **love** about it:
  - Reliable if configured in a proper way.
  - It’s easy to set up and it just works.
- What I **hate** about it:
  - Black magic behind the complaint rate.
  - Deliverability is hard to investigate.
  - It’s too simple.

4

Welcome onboard! But how?

How to know your workload.

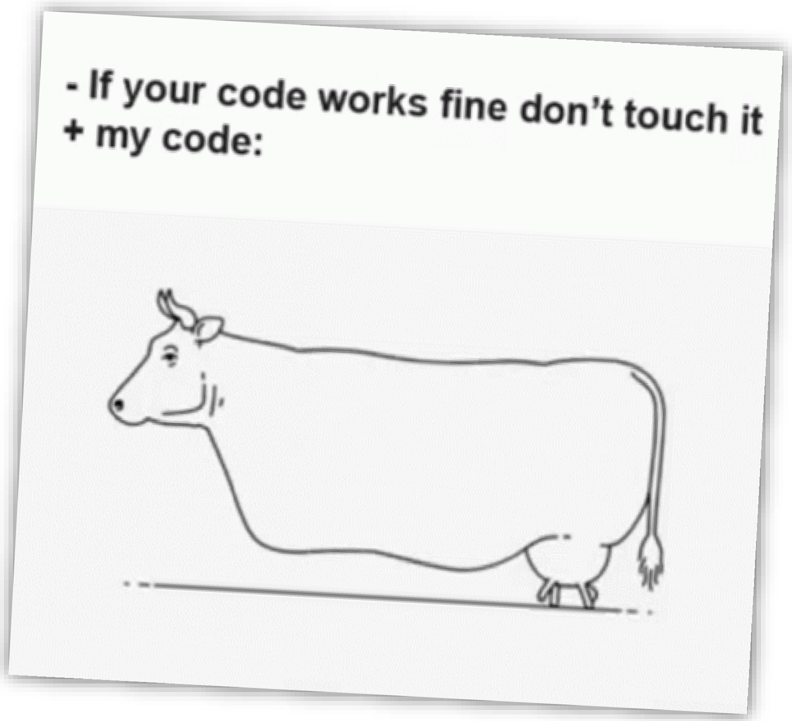
You have got the workload in your hands…

Your actions:

a) Leave everything as it is

b) Drink coffee and relax

c) Start the analysis!



- If your code works fine don’t touch it
+ my code:

How to know your workload.

You have got the workload in your hands…

Your actions:

~~a) Leave everything as it is~~

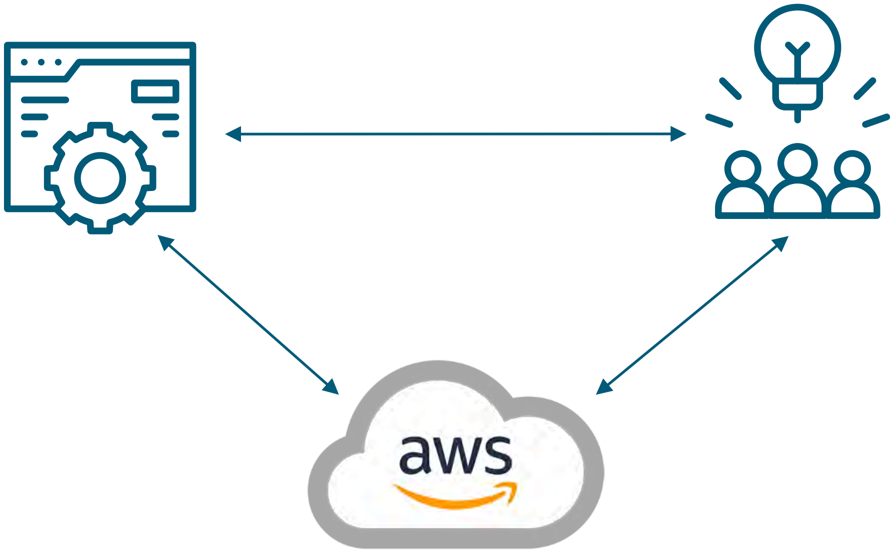
~~b) Drink coffee and relax~~

c) Start the analysis!

Because it’s our work!

How to know your workload.

Achieve the unity. Involve. Observe. Identify before it hits back.



How to know your workload.

You already have it in your hands.

- **Analyse** the decisions and approaches – soft skills (sorry, might be an exclusion)
- **Observe** the needs and behavior – CloudWatch Enhanced Monitoring & Data
- **Analyse the data** – CloudWatch Insights & Dashboards
- **Simulate** the breakages - AWS Fault Injection Service
- **Find the limits** – Apache JMeter & Selenium & Robot Framework
- Keep an eye on **performance** – analyze the data from the limits
- Don't forget the **monitoring** – CheckMK Agent & CloudWatch

How to know your workload.

It’s all about the people.

- **Collect** the feedback from the users – reach to the missed
- **Setup** the contact with the team – you can’t handle it alone
- **Find** the reasons behind the decisions – because for some reason it was made
- **Keep track** of the timeline and plans – don’t overengineer
- **Know** your user and use case – focus on the delivery of the core functions

How to know your workload.

It’s not on your local PC.

- Go through the **architecture** – know with what you are dealing
- **Don’t reinvent** the wheel…
- … Do it in the **AWS** way and use the cloud services
- **Automate** the routine – because time is gold
- Find the **weak spots** – customer must be happy
- **Save** where it’s possible – on-demand is not a choice
- Don’t forget about the **security** – just use the mind and services

5

EVO

EVO

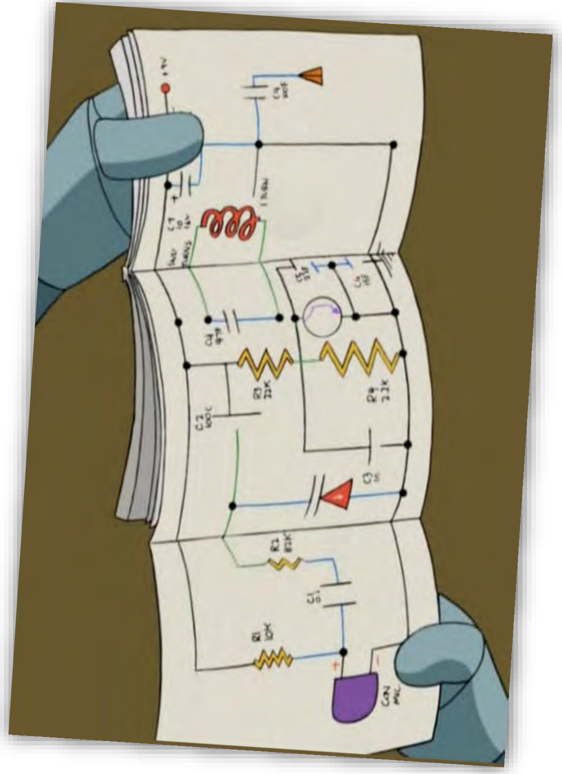
Target – keep the transparent processes to which developers are already used

Goal – minimize the blind spots and improve efficiency in all aspects

EVO

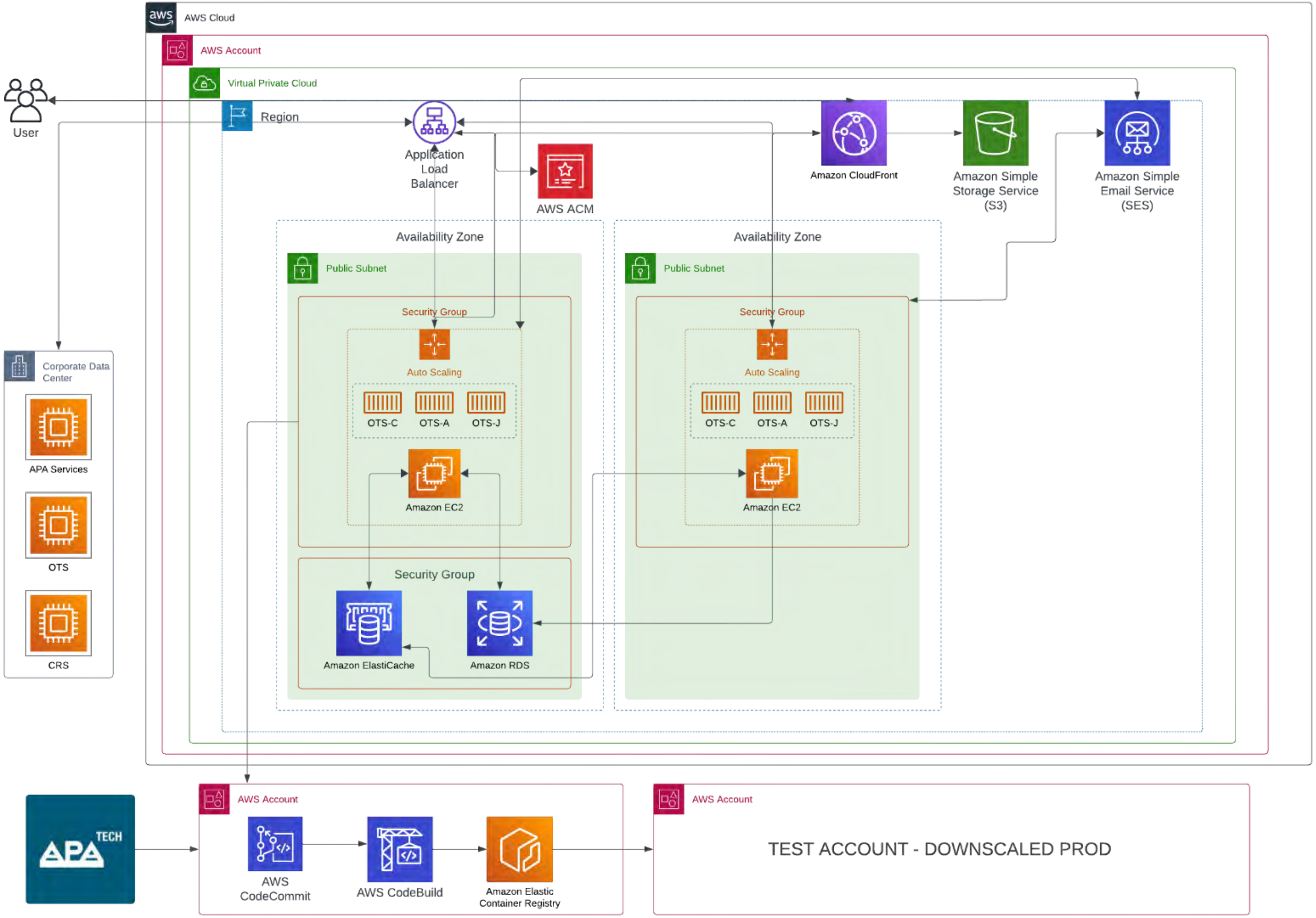
“But it works?!”

- How can we avoid breaking steady processes?
- Can we run more **cost-efficiently**?
- Are we **well fit** for our application?
- Are we **right-sized**?
- Is the setup **reliable**?
- Do we have enough **monitoring coverage**?
- Is it **secure** enough?
- I wanted to get rid of the operational **overhead**.



EVO

School album



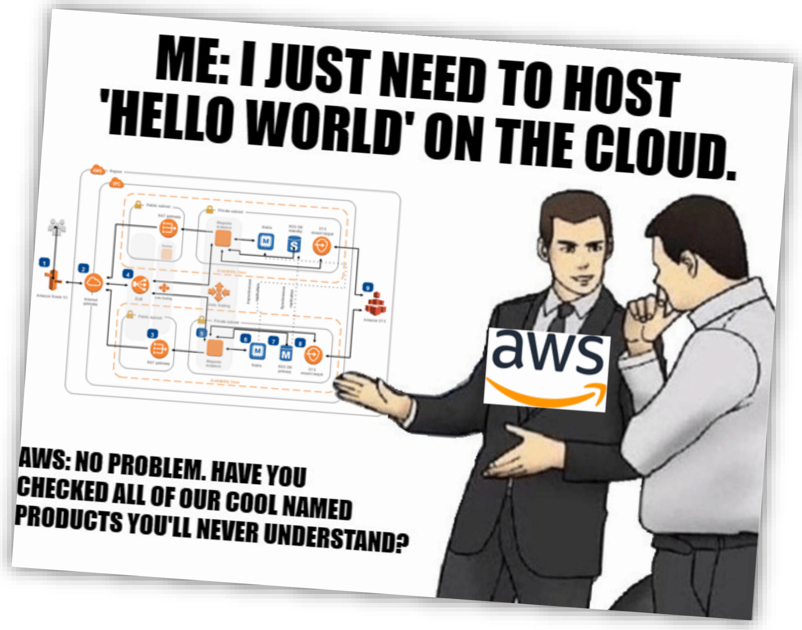
EVO

“Will you still love me when I'm no longer young and beautiful?”

This is a typical AWS deployment! It works? Yes.

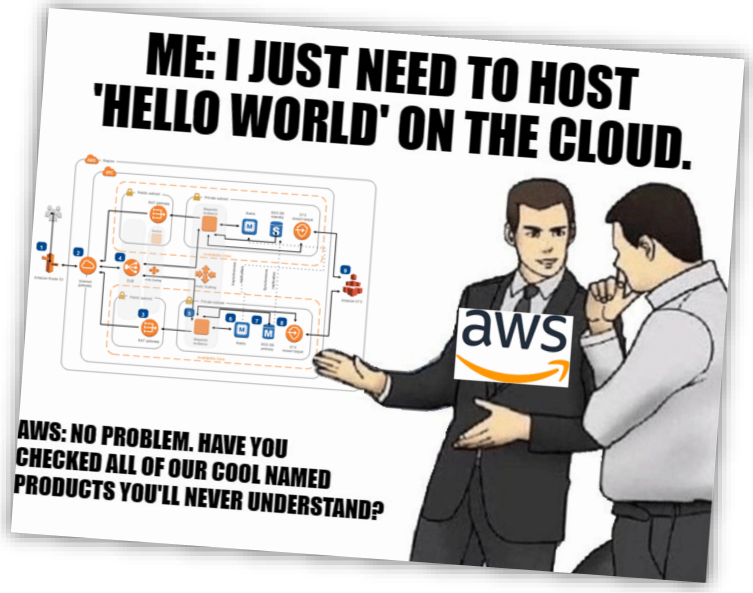
Did I like it? No.

1. We just run our stuff on the On-Demand EC2 (we are rich!)
2. Single RDS is a single point of failure.
3. No automation. At all.
4. Huge node provision time.
5. Lack of monitoring, and bad incident response.
6. Some small? security things.
7. It’s click ops…


ME: I JUST NEED TO HOST 'HELLO WORLD' ON THE CLOUD.
aws
AWS: NO PROBLEM. HAVE YOU CHECKED ALL OF OUR COOL NAMED PRODUCTS YOU'LL NEVER UNDERSTAND?

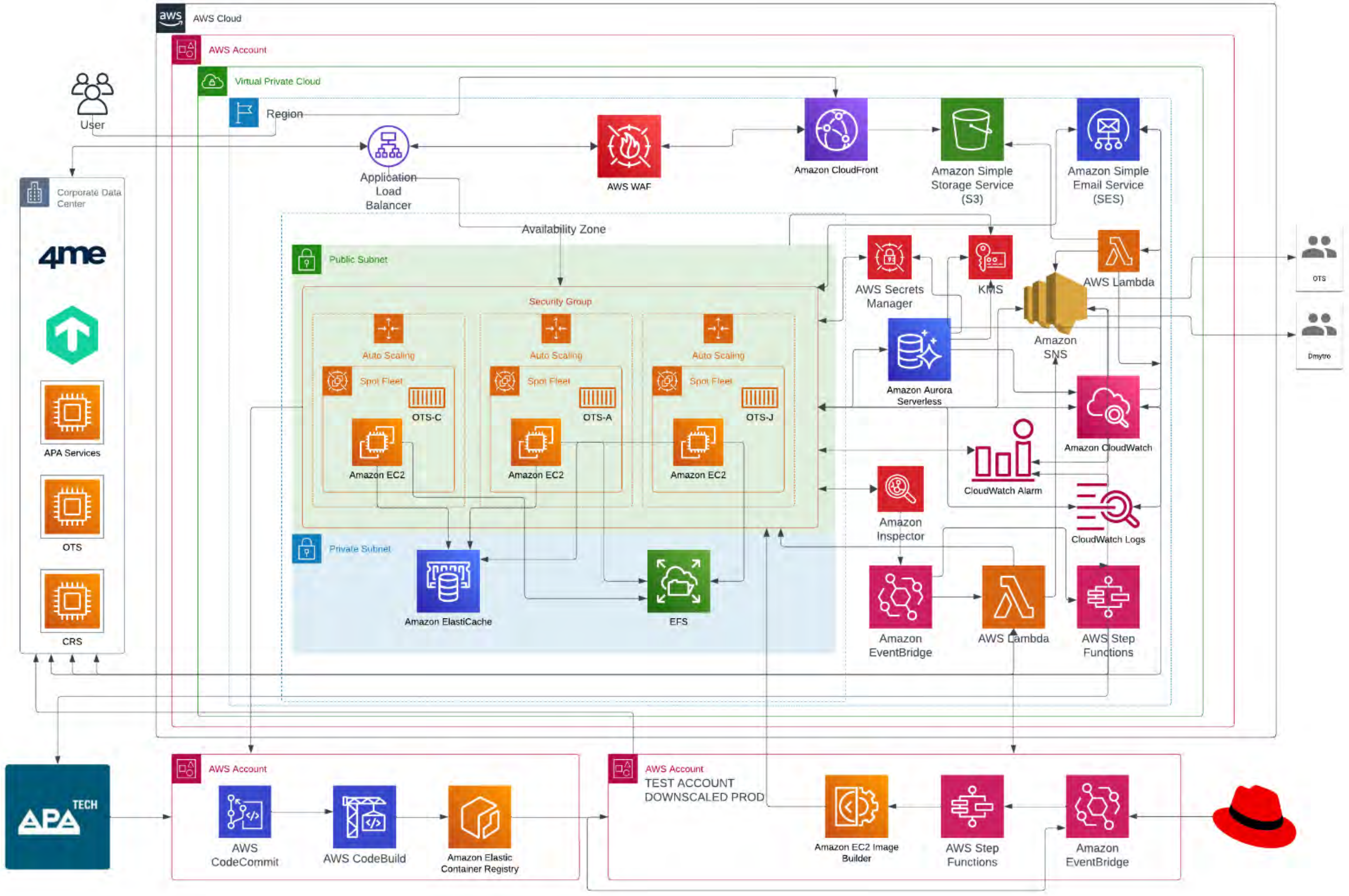
EVO

1. No WAF – no bot and data scrapping protection (essential in our case)
2. If you have ALB or CloudFront – you have to just activate it, no changes are necessary
3. WAF covers the late patching with itself
4. Secrets Manager is a must-have for the storage
5. Inspector is very easy to use for security assessment
6. Config intercepts bad configurations and resolves mess in the deployments
7. GuardDuty is optional


ME: I JUST NEED TO HOST 'HELLO WORLD' ON THE CLOUD.
aws
AWS: NO PROBLEM. HAVE YOU CHECKED ALL OF OUR COOL NAMED PRODUCTS YOU'LL NEVER UNDERSTAND?

EVO

A bit of magic



EVO

“He used to call me poison”

- We improved the **performance** in times staying at almost the same costs
- We got **monitoring** coverage from the AWS up to the frontend
- Absolutely **automated** ITIL process-compliant deployment
- ClickOps was **converted** into the IaC with CDK-powered automation
- Improved **resiliency** and security
- Automated control **of SES events** and reporting to the customer care team
- And lot’s of other small **improvements**

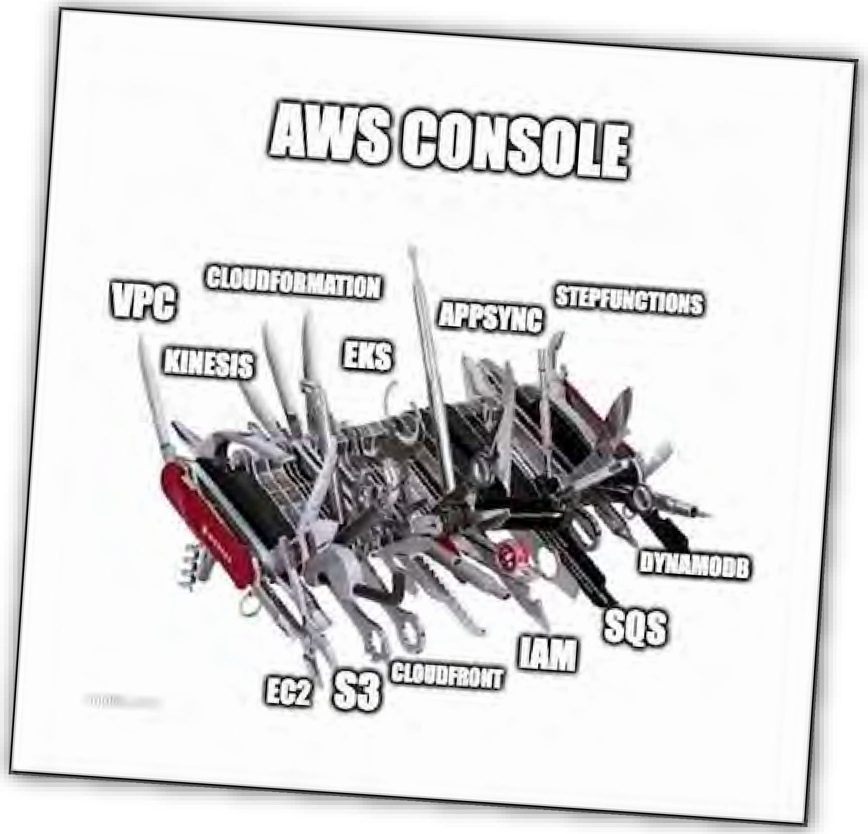
6

The Swiss Army Knife

The Swiss Army Knife

Just make it yours.

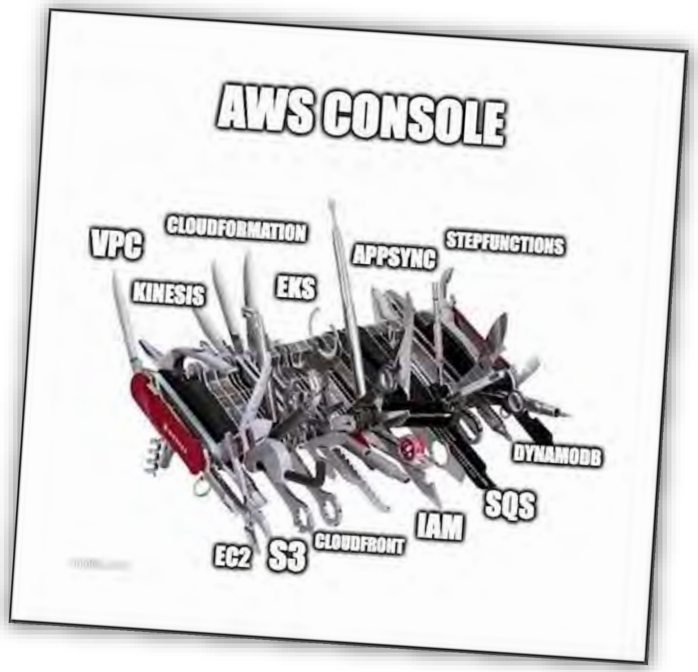
?



The Swiss Army Knife

Just make it yours.

- Any AWS service + this gives you coverage for the unusual cases:
  - AWS Lambda
  - AWS EventBridge
  - AWS SNS
  - AWS CW & Logs
- Expanded by:
  - Step Functions
  - DynamoDB
  - S3



The Swiss Army Knife

AWS Fault Injection Service

- Allows to **break parts** of the setup granularly
- Allows to setup **recovery operations**
- **Game changer** for observability coverage
- **Removes** blind spots for resilience
- Automates **routine operations**
- **Expandable**!
- It’s even cooler since this is
  a **part of the Resilience Hub** now!

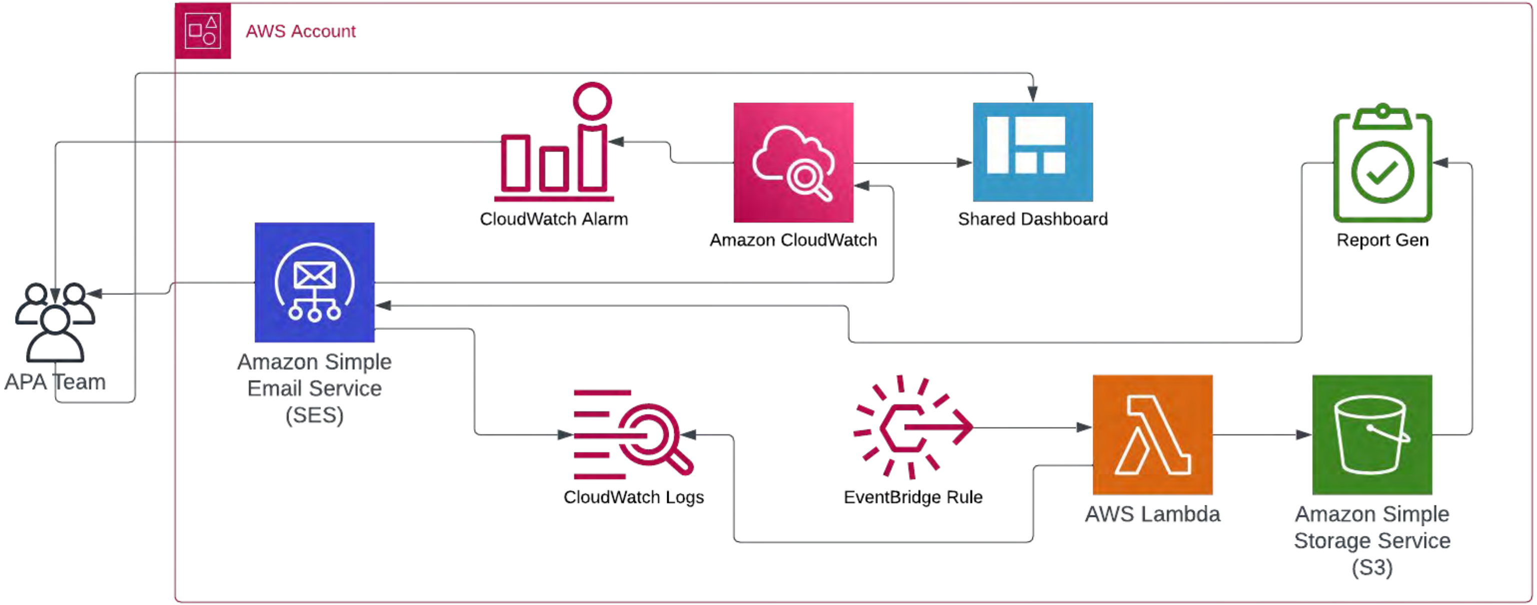
The Swiss Army Knife

AWS RDS Performance Insights

- Life saver for **performance troubleshooting** and rightsizing
- **No operational overhead** and easy to setup
- Was a **key finder** for a critical issue
- Source of valuable information for **optimization**
- Allows to incorporate **potential issues**
- **Cheap**!

The Swiss Army Knife

Example of AWS SES extension



7

It’s running! Or not?
Ballade about running your stuff.

Why the easiest way may not be the optimal way.

Gentlemen, a short view back to the past...

- Yes, **most** managed AWS services heavily reduce operational overhead…
- But you have to use them in a clever way:
    - There is a **balance** between your job and the price of the AWS service
    - You may **lack control** over the things you need
    - It may **not fit** into the enterprise processes

But still, **most of them are cool**.

Why the easiest way may not be the optimal way.

Let’s run a **minimum configuration** of the compute part:

- A few bare EC2 with a user data script in ASG (no discounts): 100%
  - EC2 Savings Plan – 62%
  - EC2 Spot Fleet – 51%
- AWS Fargate with the same vCPU/MEM capacity – 473%
- AWS EKS EC2 workers – 140%

Why the easiest way may not be the optimal way.

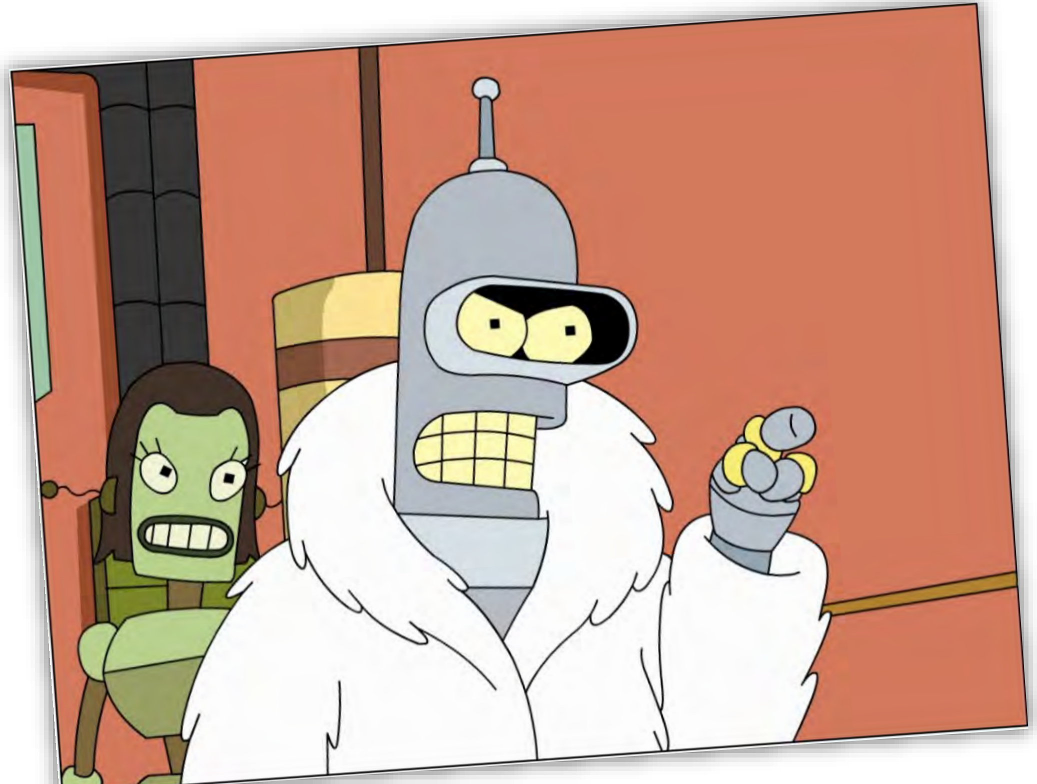
Nice websites!

Kubernetes instance calculator

Fargate Pricing Calculator

Why the easiest way may not be the optimal way.

I’m going to build my own system with spot instances and automation!



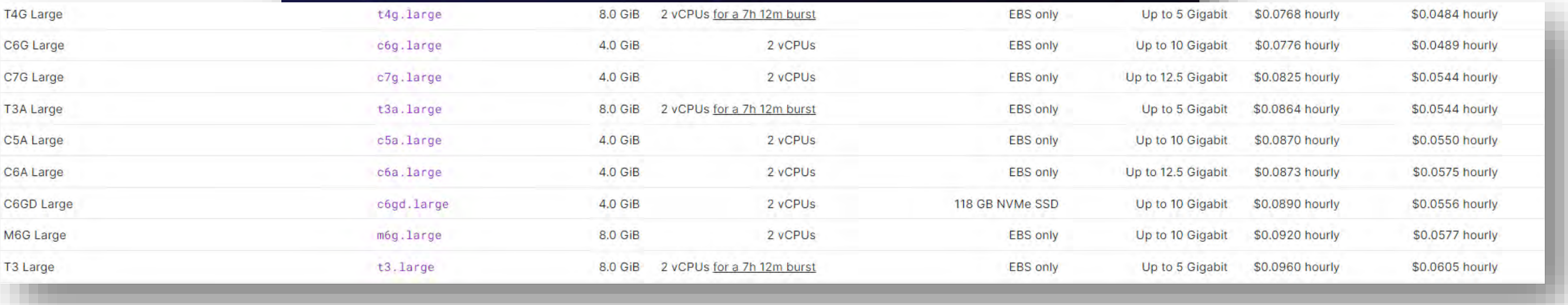
Why the easiest way may not be the optimal way.

EC2 is still an option if you have some specific requirements.

- We wanted to **keep control** of the host
- I didn’t want to rearchitect the base with no **significant benefits**
- **Load and consumption** are not symmetric:
  - One pod may overfill the worker
  - Or be killed by the limits
  - Or the host is overprovisioned (this is why I performed the load testing)
  - Specific of MCP worker is high memory and storage usage, but CPU is idling
- Of course, we are lacking **some management**. But do we want to spend extra 70$ if the job is done in another (cheaper) way?

Has the Graviton Gravity?

The Gemini



T4G Large	t4g.large	8.0 GiB	2 vCPUs for a 7h 12m burst	EBS only	Up to 5 Gigabit	$0.0768 hourly	$0.0484 hourly
C6G Large	c6g.large	4.0 GiB	2 vCPUs	EBS only	Up to 10 Gigabit	$0.0776 hourly	$0.0489 hourly
C7G Large	c7g.large	4.0 GiB	2 vCPUs	EBS only	Up to 12.5 Gigabit	$0.0825 hourly	$0.0544 hourly
T3A Large	t3a.large	8.0 GiB	2 vCPUs for a 7h 12m burst	EBS only	Up to 5 Gigabit	$0.0864 hourly	$0.0544 hourly
C5A Large	c5a.large	4.0 GiB	2 vCPUs	EBS only	Up to 10 Gigabit	$0.0870 hourly	$0.0550 hourly
C6A Large	c6a.large	4.0 GiB	2 vCPUs	EBS only	Up to 12.5 Gigabit	$0.0873 hourly	$0.0575 hourly
C6GD Large	c6gd.large	4.0 GiB	2 vCPUs	118 GB NVMe SSD	Up to 10 Gigabit	$0.0890 hourly	$0.0556 hourly
M6G Large	m6g.large	8.0 GiB	2 vCPUs	EBS only	Up to 10 Gigabit	$0.0920 hourly	$0.0577 hourly
T3 Large	t3.large	8.0 GiB	2 vCPUs for a 7h 12m burst	EBS only	Up to 5 Gigabit	$0.0960 hourly	$0.0605 hourly

Has the Graviton Gravity?

The Gemini

```
exec user process caused: exec format error
```

Has the Graviton Gravity?

The Gemini

dockerx

Jib

Has the Graviton Gravity?

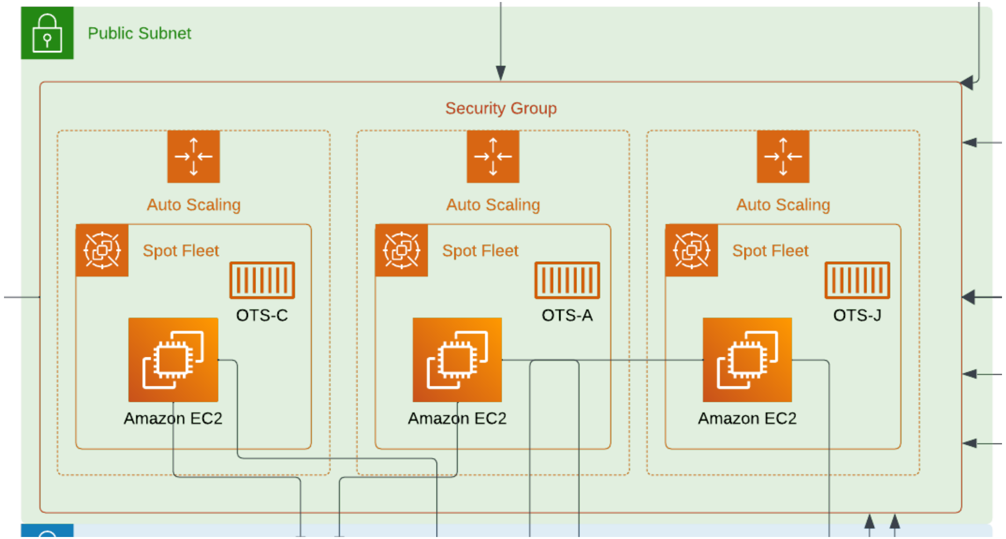
The Gemini

- Just rebuilt your container. Maybe swap the base image.
- It will require extensive testing and evaluation
  - Some of the monitoring solutions are not completely supported
  - Some of specific things may get broken
  - You might see a performance hit if you rely on SSE or AVX instructions
- Motivation is still unclear.
- It will not do things for you.
- There are other ways to save money.

Rightsizing, scaling, and the Red Hats!

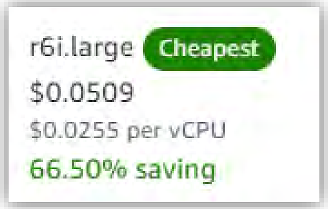
Granular scaling of the worker nodes according to the CPU and custom metrics

This is why AWS offers specialized instance types!



Rightsizing, scaling, and the Red Hats!

- I absolutely love Spot Instances! (get more for the less)
- Termination factor is not scary
- Spot fleet saves your nerves
- You can win over 60% of the price
- + Compute and EC2 Savings Plans are your friends!

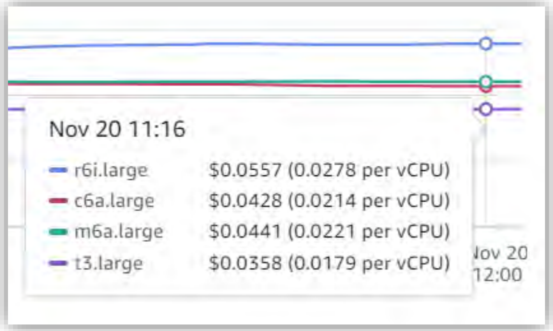


r6i.large **Cheapest**

$0.0509

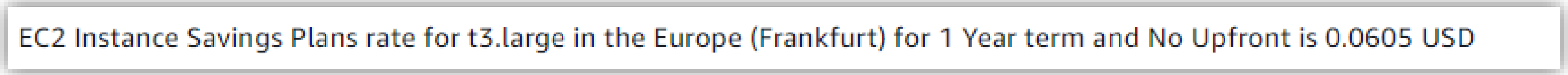
$0.0255 per vCPU

66.50% saving

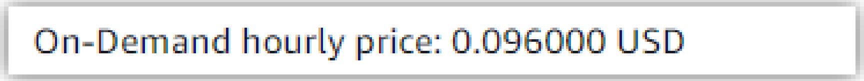


Nov 20 11:16

▬ r6i.large	$0.0557 (0.0278 per vCPU)
▬ c6a.large	$0.0428 (0.0214 per vCPU)
▬ m6a.large	$0.0441 (0.0221 per vCPU)
▬ t3.large	$0.0358 (0.0179 per vCPU)

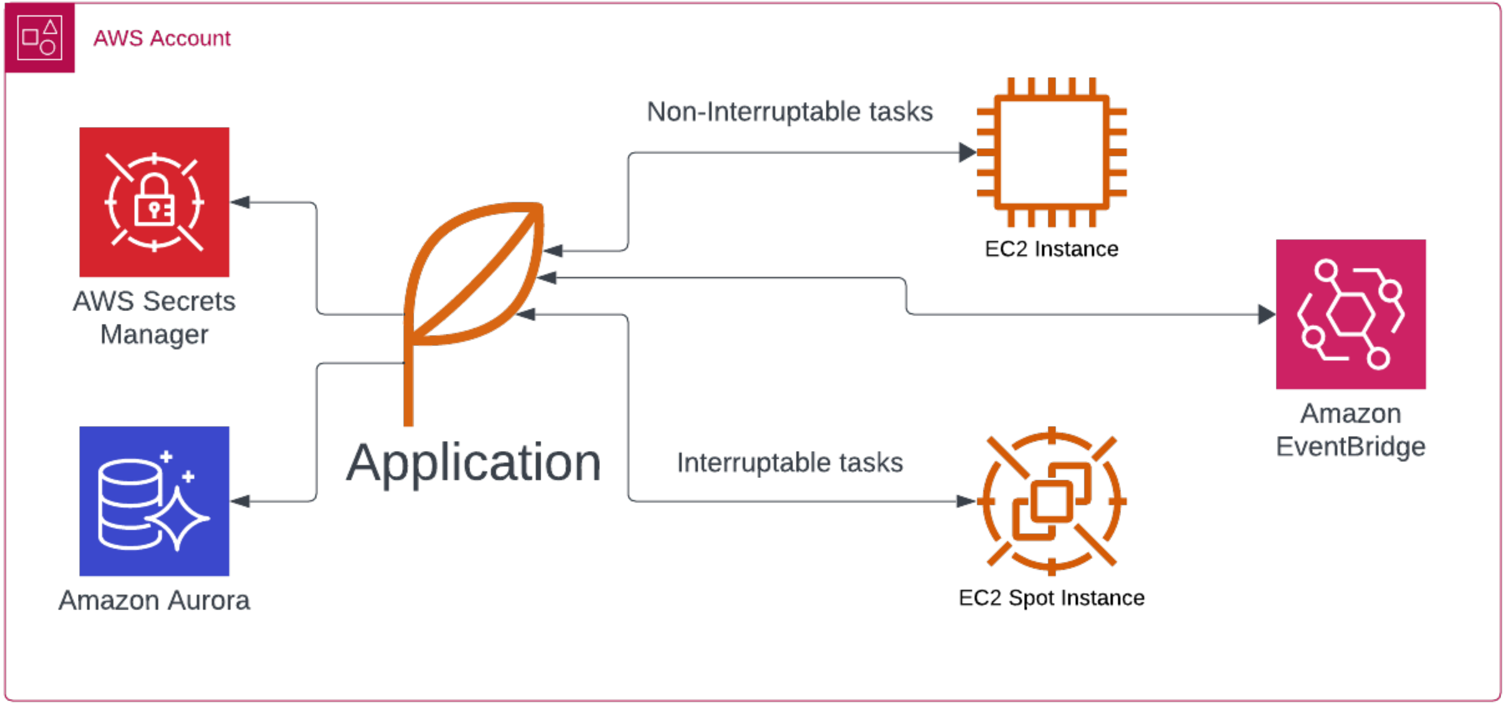


EC2 Instance Savings Plans rate for t3.large in the Europe (Frankfurt) for 1 Year term and No Upfront is 0.0605 USD



On-Demand hourly price: 0.096000 USD

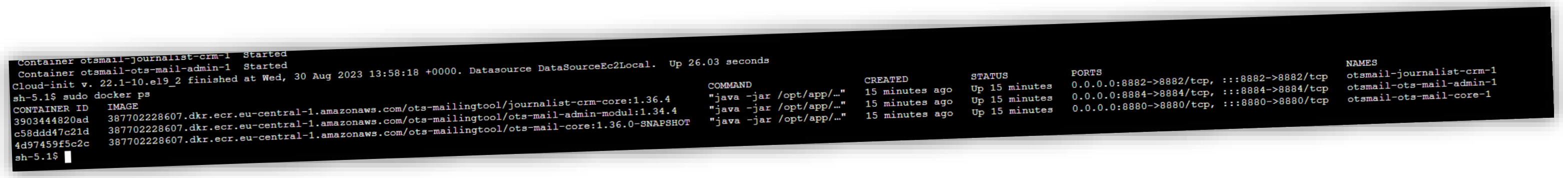
Rightsizing, scaling, and the Red Hats!



Why the easiest way may not be the optimal way.

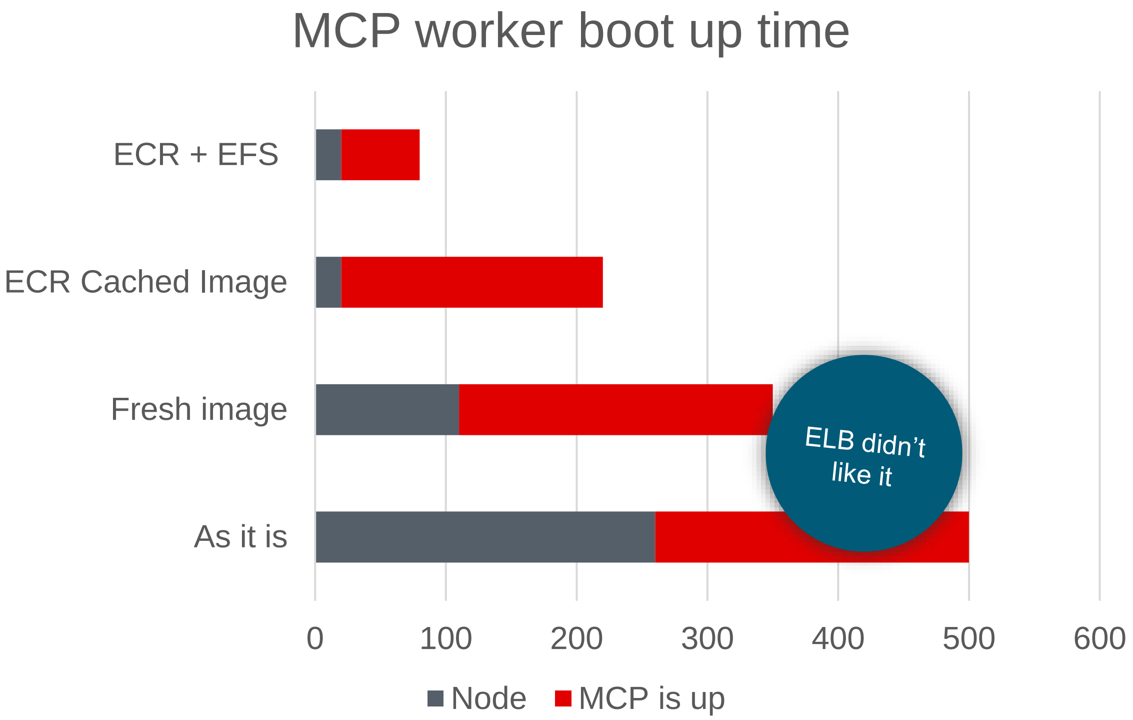
Do the problems come with scale? Yes.

- **Response time** is critical for the scaling:
  We can’t afford to affect service response time meanwhile worker starts up
- ASG also helps here:
  - **Scheduled scaling**
  - **Warm pool**
- And an absolute life-saver is the **EC2 Image Builder**!
- … and also cache on **EFS** helps



```
 Container otsmail-journalist-crm-1  Started
 Container otsmail-ots-mail-admin-1  Started
Cloud-init v. 22.1-10.el9_2 finished at Wed, 30 Aug 2023 13:58:18 +0000. Datasource DataSourceEc2Local.  Up 26.03 seconds
sh-5.1$ sudo docker ps
CONTAINER ID   IMAGE                                                                                       COMMAND                  CREATED          STATUS          PORTS                                        NAMES
3903444820ad   387702228607.dkr.ecr.eu-central-1.amazonaws.com/ots-mailingtool/journalist-crm-core:1.36.4   "java -jar /opt/app/…"   15 minutes ago   Up 15 minutes   0.0.0.0:8882->8882/tcp, :::8882->8882/tcp   otsmail-journalist-crm-1
c58ddd47c21d   387702228607.dkr.ecr.eu-central-1.amazonaws.com/ots-mailingtool/ots-mail-admin-modul:1.34.4  "java -jar /opt/app/…"   15 minutes ago   Up 15 minutes   0.0.0.0:8884->8884/tcp, :::8884->8884/tcp   otsmail-ots-mail-admin-1
4d97459f5c2c   387702228607.dkr.ecr.eu-central-1.amazonaws.com/ots-mailingtool/ots-mail-core:1.36.0-SNAPSHOT "java -jar /opt/app/…"   15 minutes ago   Up 15 minutes   0.0.0.0:8880->8880/tcp, :::8880->8880/tcp   otsmail-ots-mail-core-1
sh-5.1$ █
```

Rightsizing, scaling, and the Red Hats!



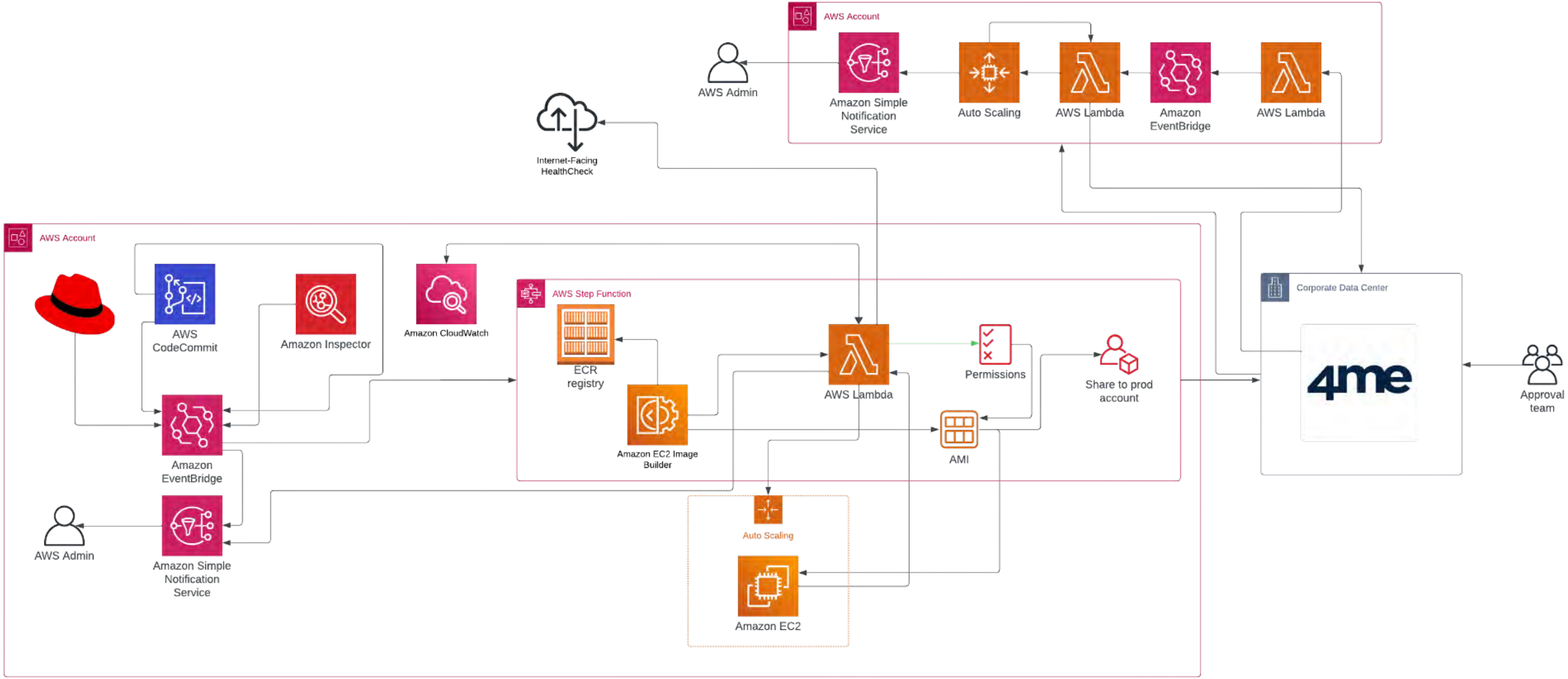
ELB didn't like it

Why the easiest way may not be the optimal way.

Could you give me your Red Hat?

- If your enterprise has a Red Hat **subscription**, you can also use them on AWS
- You can use BYOL AMI of RHEL and **don’t pay** the hourly rate
- **Easy to setup**
  - You provision necessary roles for the Red Hat in your account
  - You can also use StackSets in your organization
  - And then just verify the account at the Red Hat console
- **Bonus**: you also get management with Red Hat Sattelite through the RHHCC
- Then, just run EC2 from the **private BYOL** image…
- … but default setup is **bad for ASG**

Rightsizing, scaling, and the Red Hats!

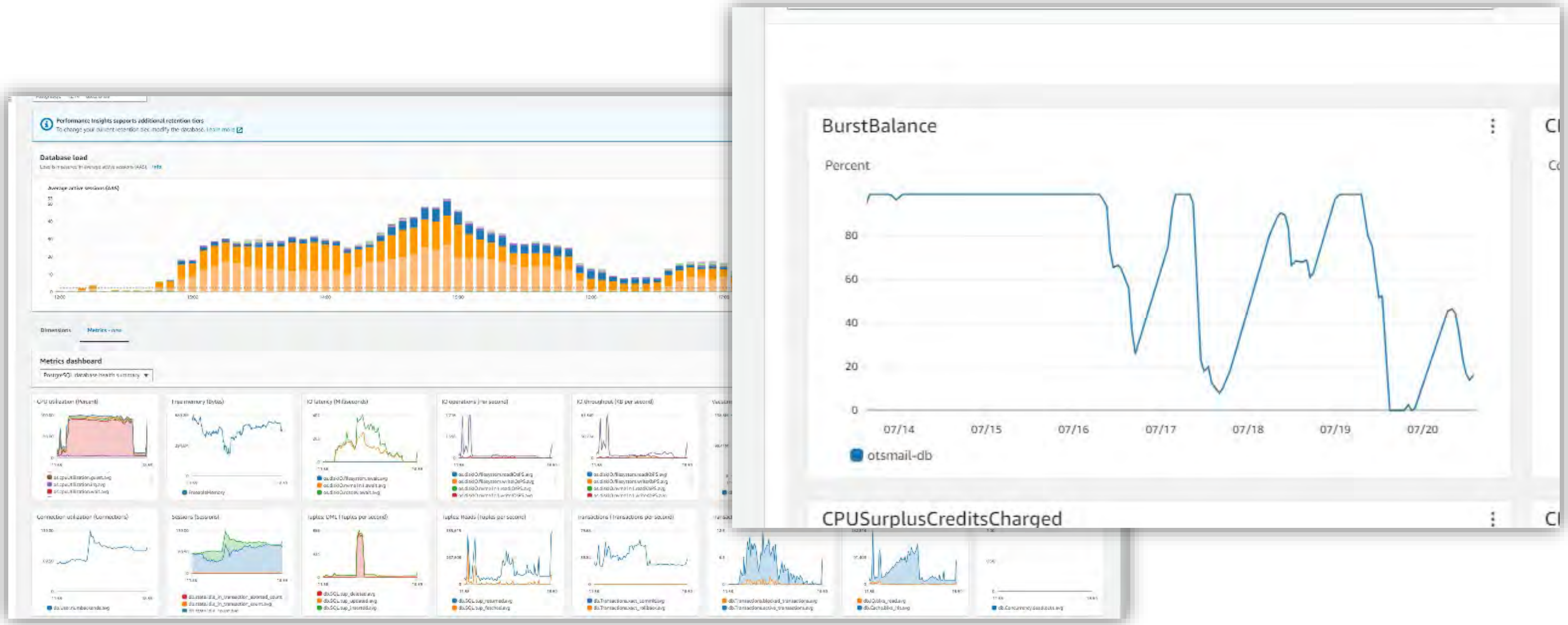


Databases, surprises and finding the approach.

Do you like surprises? Then keep running RDS with GP2 drive. Why?

Databases, surprises and finding the approach.

It’s all about the burst credits:



Performance Insights supports additional retention tiers
Database load
Average active sessions (AAS)
Dimensions
Metrics - new
Metrics dashboard
CPU utilization (Percent)
Free memory (Bytes)
IO latency (Milliseconds)
IO operations (Per second)
IO throughput (KB per second)
Connection utilization (Connections)
Sessions (Sessions)
Tuples: DML (Tuples per second)
Tuples: Reads (Tuples per second)
Transactions (Transactions per second)
BurstBalance
Percent
80
60
40
20
0
07/14
07/15
07/16
07/17
07/18
07/19
07/20
otsmail-db
CPUSurplusCreditsCharged

Databases, surprises and finding the approach.

- Existence of **GP3** makes GP2 pointless:
  - **No Burst Credits**
  - Provisioned 300 IOPS and 125 mb/s baseline; **Higher limits**!
  - You **don’t need to scale** the capacity to use drive stripping
  - **It’s cheaper**! 0,08$/GB/Month vs 0,10/GB/Month
  - Goes **over the limits** for $$$, has better performance for the price of gp2
- But **I/O waits** doesn’t mean you are having troubles with the storage!
- GP2 is bad for small drives, but looks great for **big volumes** with higher peaks

Databases, surprises and finding the approach.

You should not only monitor the storage and CPU usage.

- **All kinds** of credits (CPU, Storage)
- DB **Load** and non-SQL DB Load
- **SWAP** usage
- Write & Read **Latency** and query depth
- Number of **connections**

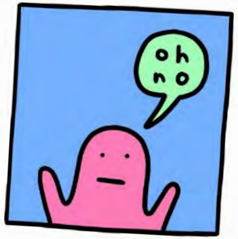
Databases, surprises and finding the approach.

But waits are still there!

How to make RDS work
A: Use read-replica
B: Upscale RDS
C: Use RDS Proxy
D: Call Kevin
imgflip.com

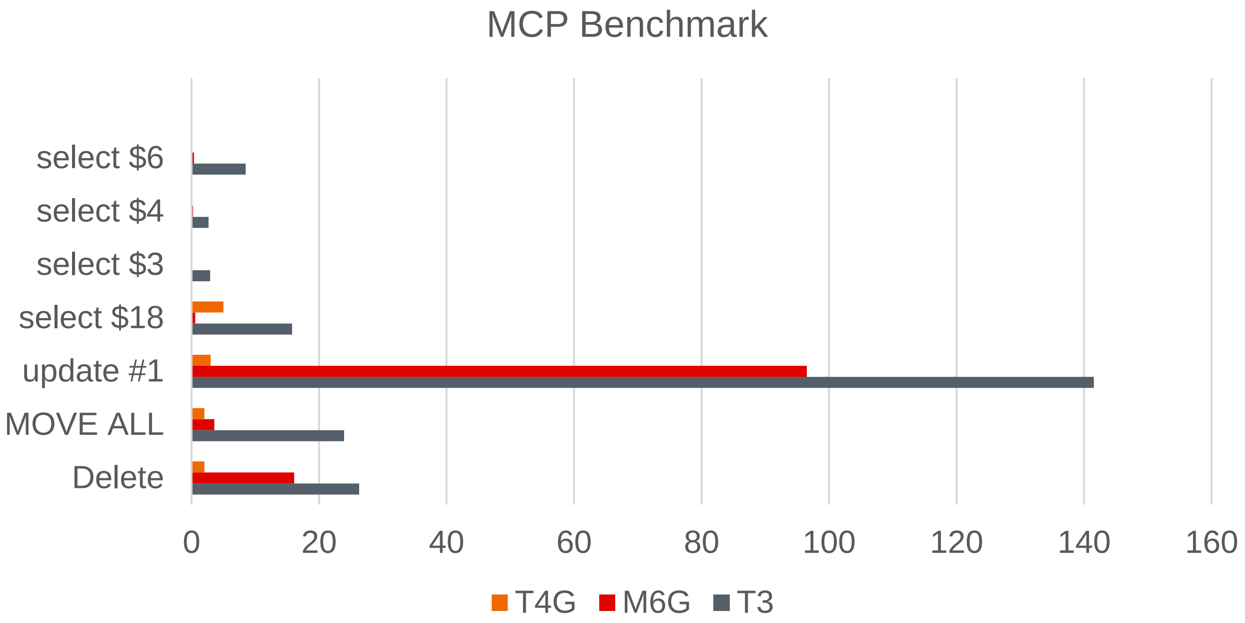
Databases, surprises and finding the approach.

- Use **Read-Replica**:
  - Leads to application **reengineering**…
  - **Springboot** can natively use read-replica endpoint if you mark the data
- **RDS Proxy**:
  - Makes sense for applications with a **high number** of connections (Lambda)
  - Doesn’t fix the issue with **read/writes**
  - But… It can reduce the **Multi-AZ** failover time by 79% (nice)
  - **Heimdall** Proxy can manage replicas instead of the app ($$$$)
- Upscale RDS? To which one? The manager says there is **no money**. ™
- **Call Kevin** – emergency case



Databases, surprises and finding the approach.

OK, we are scaling up... But to which one?



Databases, surprises and finding the approach.

OK, we are scaling up... But to which one?

	T3	M6G	T4G
Delete	26,294	16,1	17,255
MOVE ALL	23,921	3,58	9,36
update #1	141,533	96,513	100,917
select $18	15,77	0,57	1,18
select $3	3	0.094	0,119
select $4	3	0,23	0,278
select $6	8,5	0,38	0,439

Databases, surprises and finding the approach.

Alter-ego

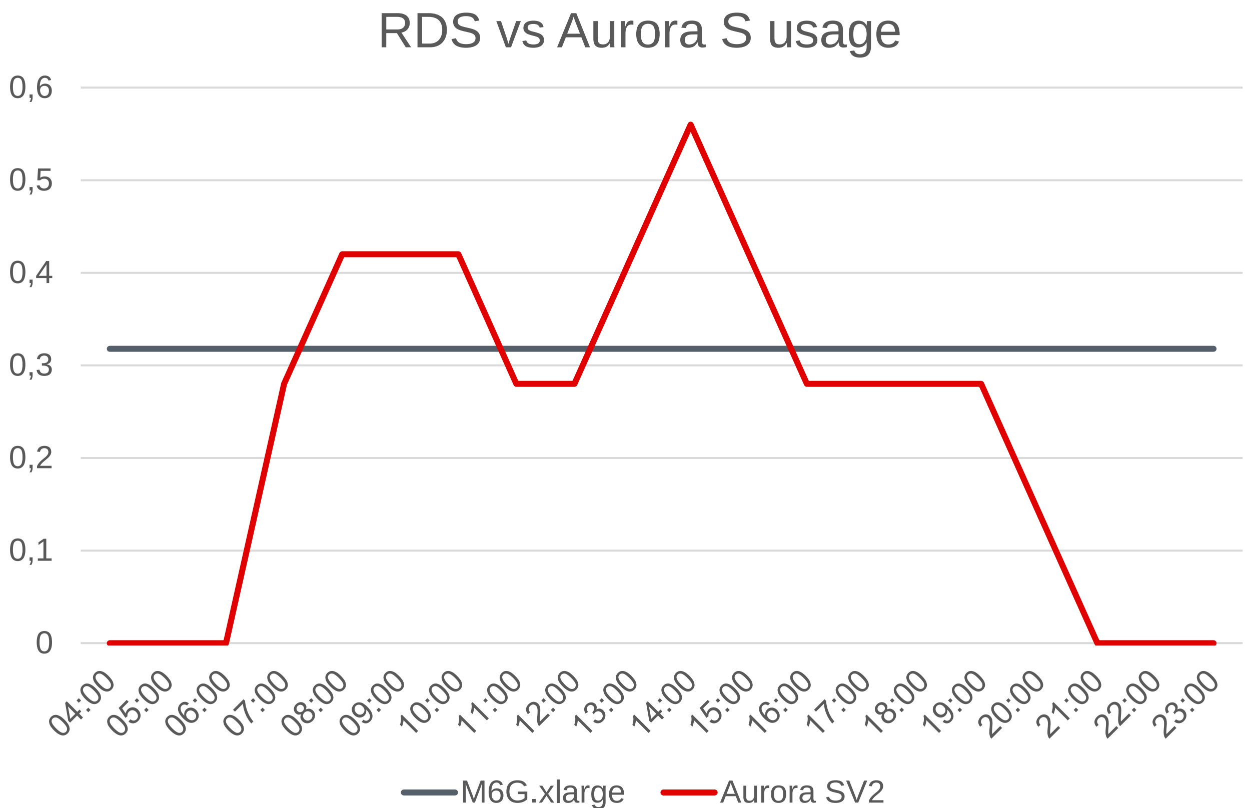
- If you are targeting a **steady production** starting point – M6G is a great choice
- T4G.Large gives comparable **performance** for less
  - 311$ vs 214$ in Multi-AZ Mode
  - Only if you have monitored and stable CPU credit consumption
- There is **no sense** to run T3 since we have T4G
- But… DB price is still **around the limit** and DB is idling outside the business hours.

Databases, surprises and finding the approach.

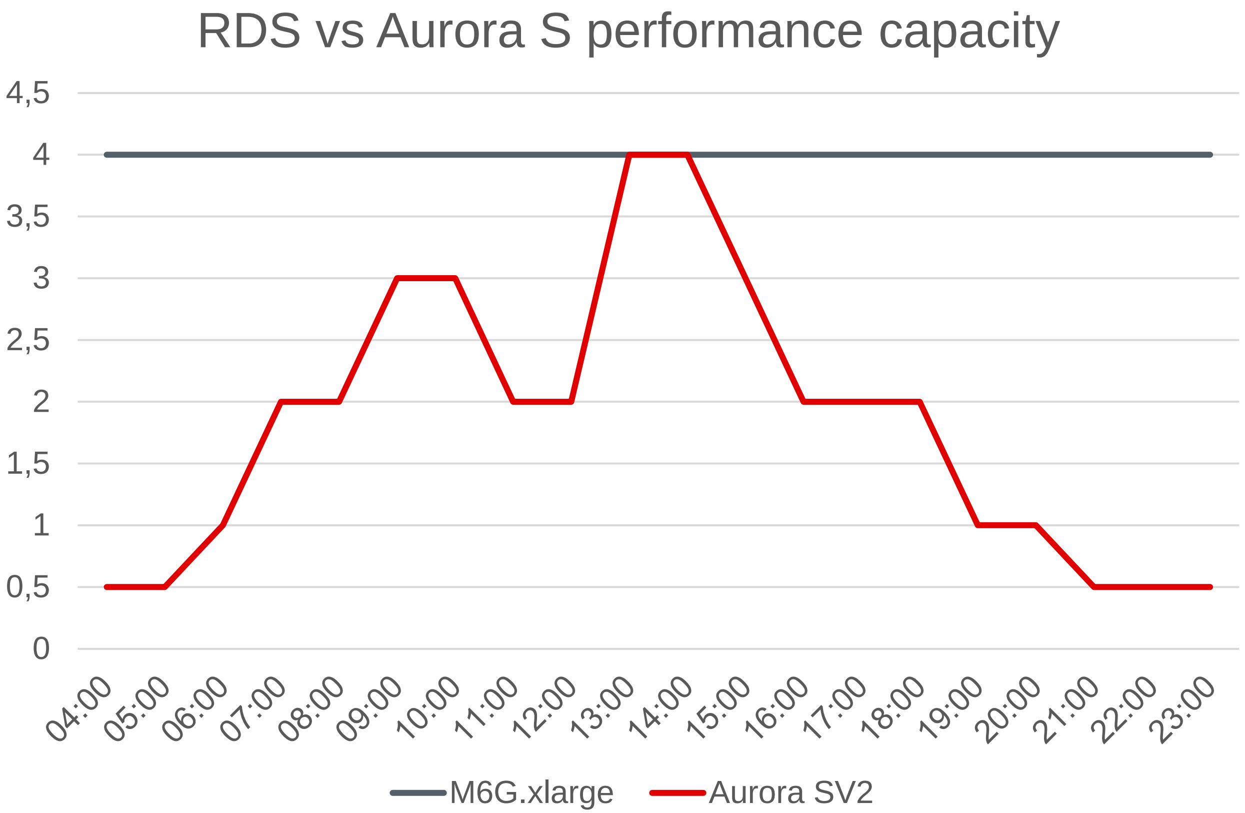
Solution? Aurora Serverless v2!

- Supports **autoscaling**! Independently!
- **Single endpoint**! Reader-only auto management!
- About **35% cheaper** in our configuration over RDS
  - RDS can still be cheaper on a big scale
  - Peak capacity price is scary, but we don’t use it!

Databases, surprises and finding the approach.



Databases, surprises and finding the approach.



Databases, surprises and finding the approach.

Aurora Serverless v2 price calculation

By Joe Howe

Conclusion:

- Communicate with your team
- Know the details of the services you use
- Same things can be different
- There are different ways to achieve the same target
- Details matter
- Be Creative!
- AWS gives endless opportunities

Thank you for your attention!

Add me on LinkedIn!

+43 664 88643880
dmytro.hlotenko@apa.at
www.darkjoney.at
www.apa.at

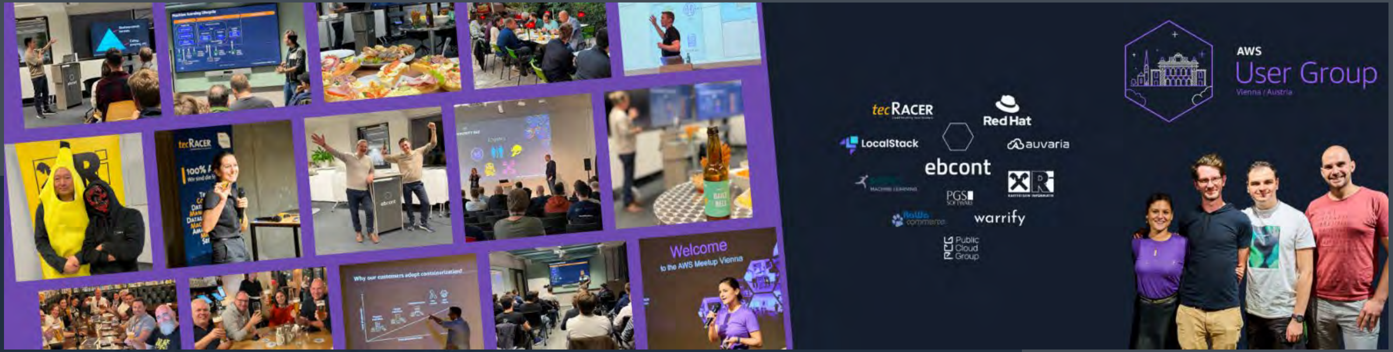
Thanks to Conf42 and Mark Pawlikowski for the invitation!

+43 664 88643880

dmytro.hlotenko@apa.at

www.darkjoney.at

www.apa.at



FÖRDERVEREIN AWS COMMUNITY

🇦🇹 🇩🇪 🇨🇭

aws-community.de

@AWSCommunityDE

We support and connect User Groups, Heroes & Community Builders of the DACH region

aws HEROES

COMMUNITY DAY DACH

AWS community builders

AWS User Groups