All about the money: Cost modeling and optimization of cloud applications

Dr. Sebastian Baltes

empirical-software.engineering

THE UNIVERSITY of ADELAIDE

12th International Workshop on Genetic Improvement @ICSE2023

A pot of gold at the end of the cloud rainbow?

Often, there's rather an unexpectedly high bill

aws

Azure

Google Cloud

Alibaba Cloud

Painting:
"The arrival of the AWS bill."
Oil on canvas.

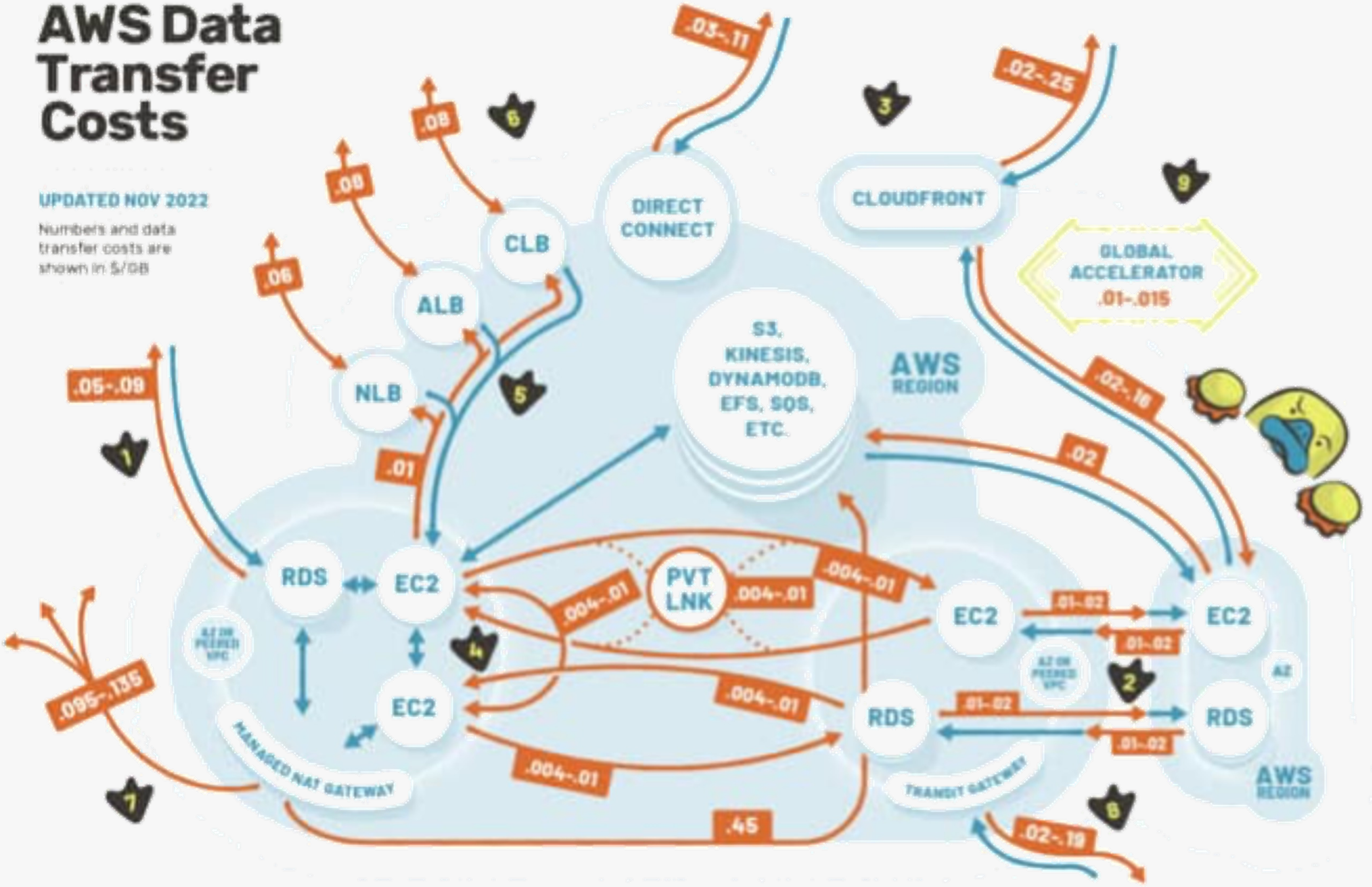
1:06 PM · Dec 23, 2022

https://twitter.com/tunguz/status/1606259838253641729

AWS Data Transfer Costs

UPDATED NOV 2022

Numbers and data transfer costs are shown in $/GB



the duckbill group

Still confused as hell? Get help at duckbillgroup.com

Inbound traffic is typically free – outbound is not. Some (but not all) internal traffic is **free.**

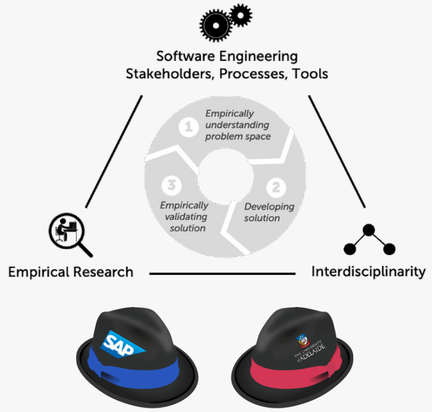
Outbound traffic costs are shown **per transmission.**

1. Direct outbound data starts at **$90/TB** for less than 10TB, and discounts with volume. **First 100GB is free.**
2. Region-to-region traffic is **$20/TB** when it exits a region for indicated services except between us-east-1 and us-east-2, where it's **$10/TB.** Even data wants to get out of Ohio.
3. Outbound CloudFront prices are variable by region and usage, but the free tier includes 1TB/month
4. Internal traffic via public or elastic IPs incurs **additional fees** in both directions.
5. Cross-AZ EC2 traffic within a region costs as much as region-to-region. ELB-EC2 traffic is **free** except outbound crossing AZs.
6. Elastic Load Balancing: Classic and Network LB is priced per GB. Application LB costs are in LCUs, not $/GB.
7. Traffic via Managed NAT Gateway – regardless of destination – costs an additional **$45/TB** on top of other transfer, including internal transfer (S3, Kinesis, etc.).
8. Variable by port speed and location. Data processing charges apply for each gigabyte sent to the AWS Transit Gateway – whether from a VPC, Direct Connect or VPN.
9. Global Accelerator charges a **$15-$105/TB** charge on top of existing data transfer rates, in whichever direction the data flow is more expensive.

Inspired by Open Guide to AWS's data transfer diagram
github.com/open-guides/og-aws

https://www.duckbillgroup.com/understanding-data-transfer-in-aws/

Good news for GI: A lot of (cost) optimization potential!



Personal background

My current role(s)

Principal Expert ESE

SAP SE

Walldorf, Germany

Adjunct Lecturer

University of Adelaide

Adelaide, Australia


Software Engineering Research Beyond
Disciplinary and Institutionalized Boundaries

Disciplinary Boundaries of Software Engineering

1968 NATO Software Engineering Conference, Garmisch, Germany

Disciplinary Boundaries of Software Engineering

- With a **traditional view** emphasizing software engineering's **roots in computer and systems engineering** many questions of modern software development **cannot be answered**.
- Examples:
    - *How can we develop visual programming environments without knowledge of cognition?*
    - *How can we fully grasp the implications of online code reuse without understanding copyright legislation and software licenses?*
    - *How can we systematically compare and optimize cloud application costs across vendors and abstractions without knowledge about workload and cost modeling?*

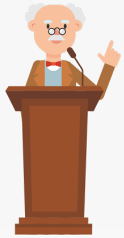
Personal Observation

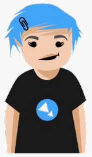
- Many of the problems **relevant** in the **software industry** are rooted in software engineering but often have an **interdisciplinary** angle.
- To be able to impact industry, academia needs to provide **actionable recommendations** addressing **problems rooted in practitioners' actual needs**.
- **Empirical research methods are essential** for identifying the above-mentioned problems (*problem space*) and corroborating recommendations/proposed solutions with empirical evidence (*solution space*).

Institutionalized Boundaries

*"If you were using **MDE** for building your mobile app, you'd see huge quality improvements, see this paper."*

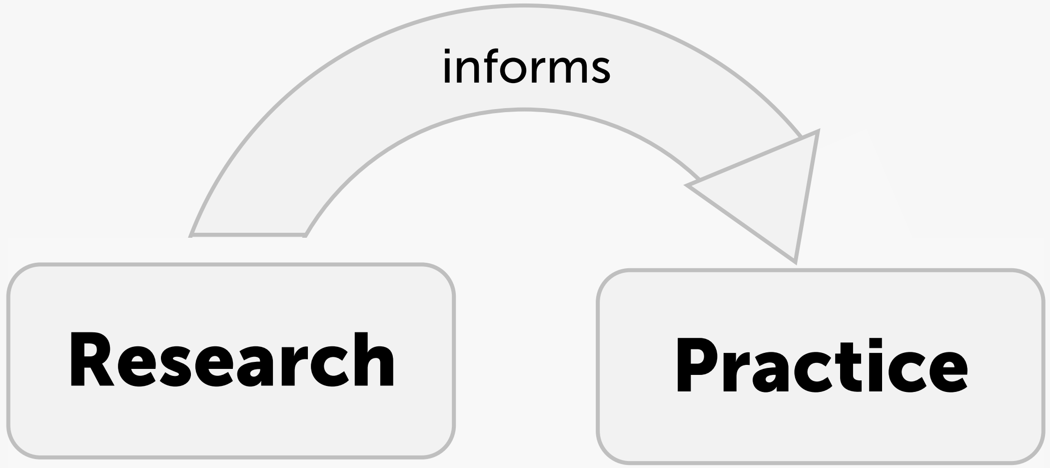
*"Have you heard about things like **time-to-market** and quickly responding to customer feedback? We're not building safety-critical software."*



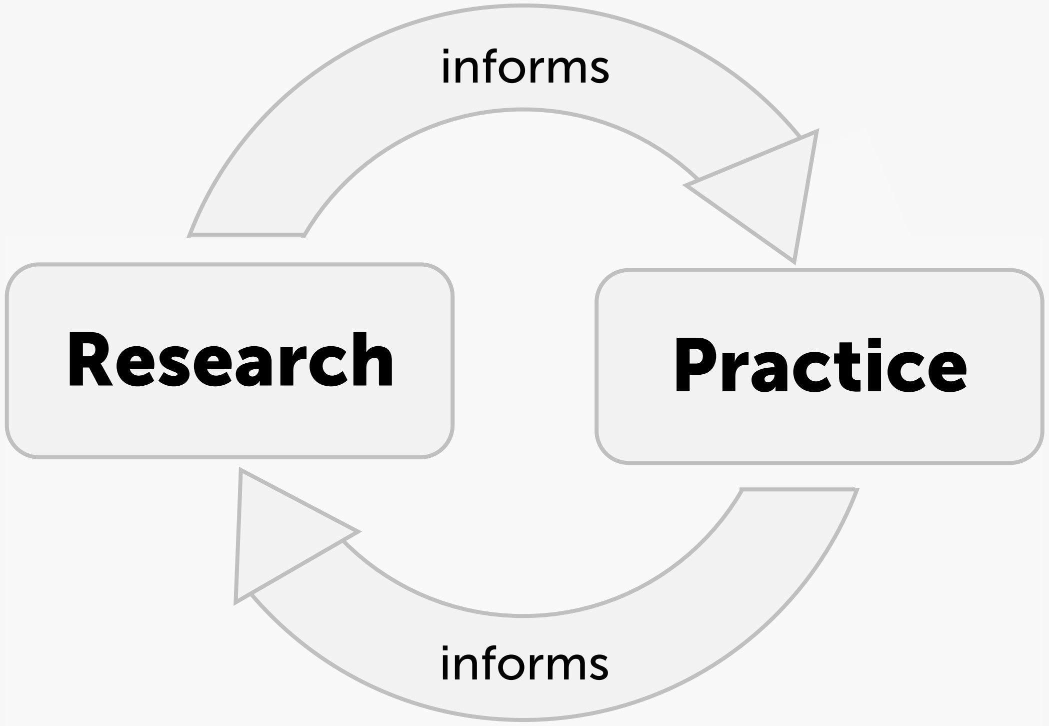


Research | **Practice**

Institutionalized Boundaries



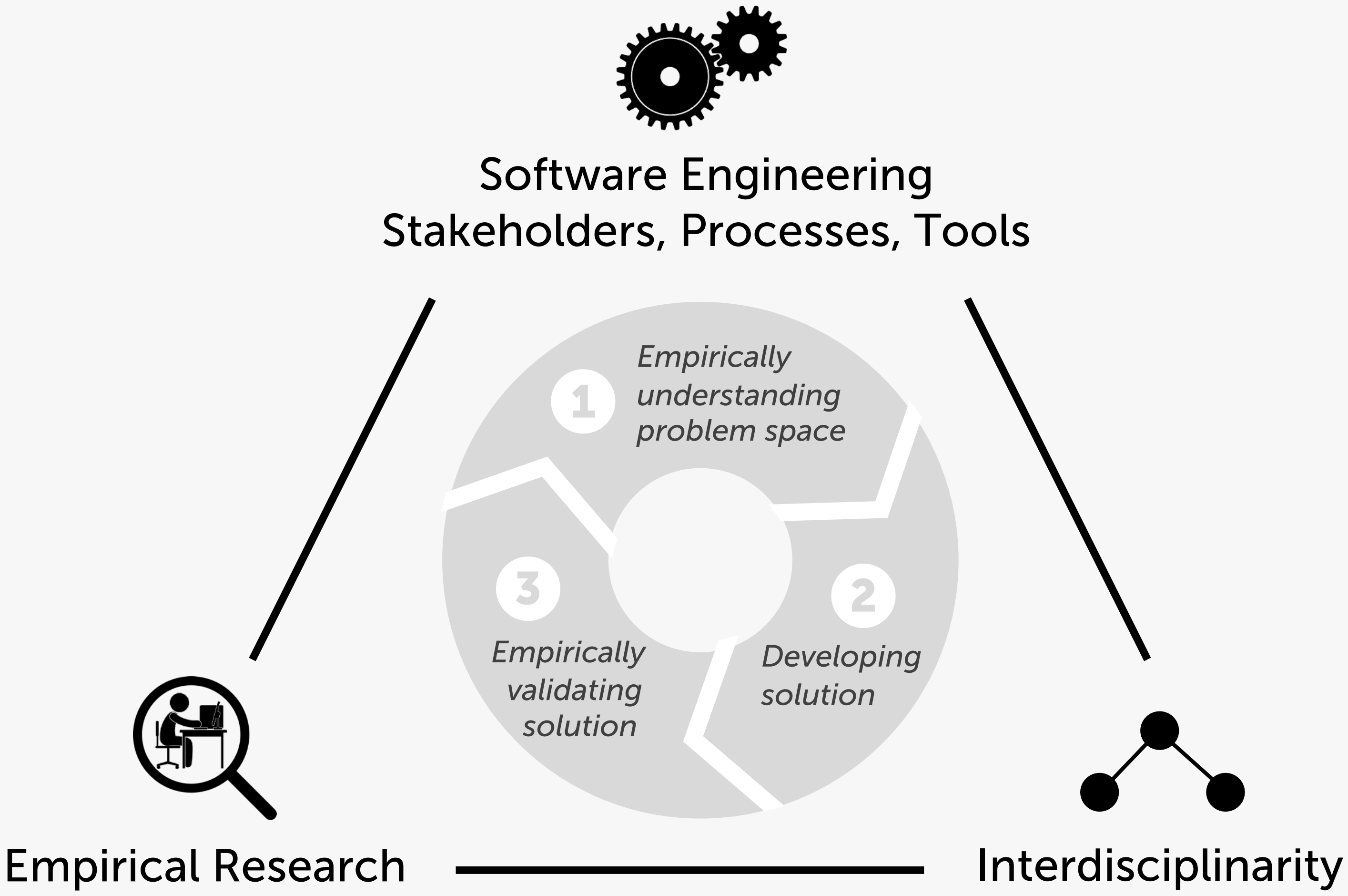
Institutionalized Boundaries



Implications for researchers:

1) Strong understanding of **state of practice** is essential.
2) To reach this understanding, we need to utilize **diverse empirical research methods** and **learn from other disciplines.**
3) To advance evidence-based practice, we need to **invest effort into communicating findings back** to practitioners.

Empirical Software Engineering



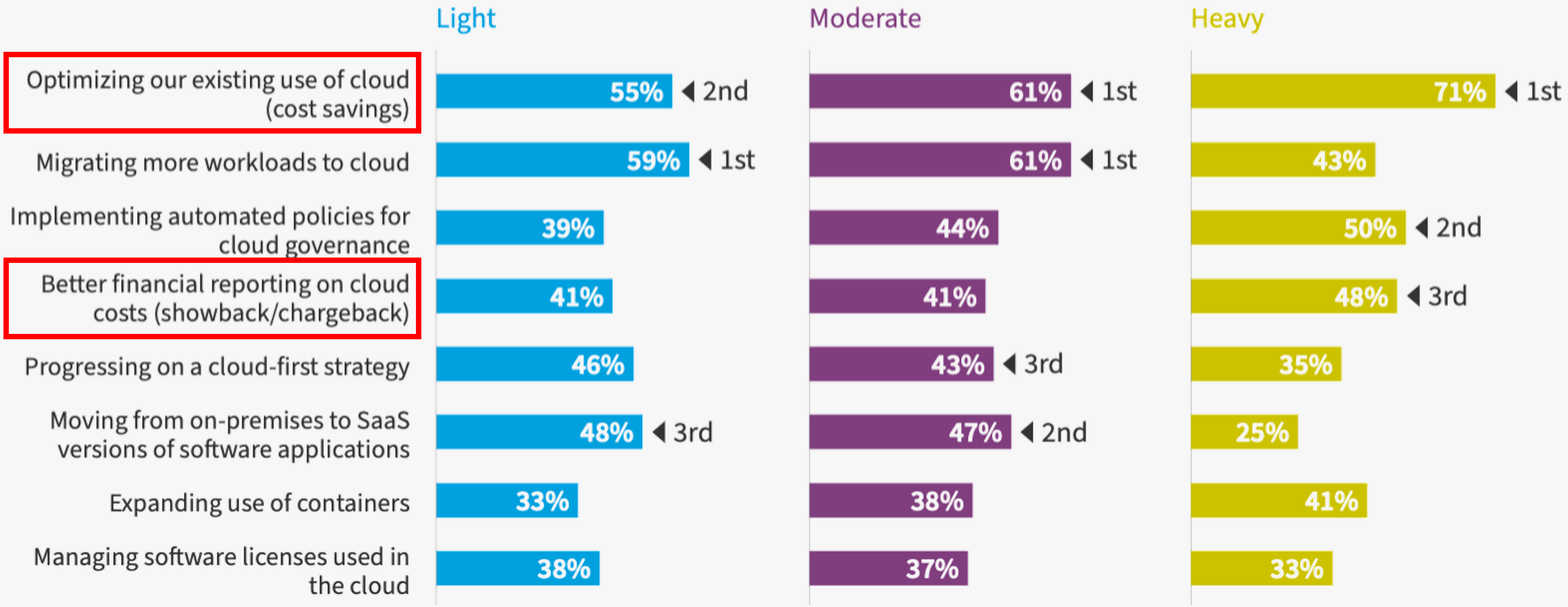
Why do companies move workloads to the cloud?

Why move to the cloud?

- **Cost transparency** and/or **cost reduction.**

- *Capacity* is the maximum *workload* a cloud layer can handle.
- **Scalability**: Ability of a cloud layer to *increase its capacity* by expanding its quantity of consumed lower-level services.
- **Elasticity**: Degree a cloud layer autonomously adapts capacity to workload over time.
  (definitions by Lehrig et al. 2015)

Top cloud initiatives by cloud usage for all organizations



N=750
Source: Flexera 2023 State of the Cloud Report

flexera

Cost transparency in the cloud is a problem

Hacker News new | past | comments | ask | show | jobs | submit

▲ Tell HN: I DDoSed myself using CloudFront and Lambda Edge and got a $4.5k bill
274 points by huksley 5 months ago | hide | past | favorite | 333 comments

https://news.ycombinator.com/item?id=31907374

@donkersgoed@hachyderm.io
@donkersgood

How a single-line bug cost us $2000 in AWS spend..

We recently refactored a Lambda Function. We extensively tested its functionality and released it into production. And everything still worked as expected. But then the billing alarm went off..

https://twitter.com/donkersgood/status/1635244161778737152

Reducing BigQuery Costs: How We Fixed A $1 Million Query

by Calvin Zhou • Data Science & Engineering
Nov 3, 2022 • 3 minute read

shopify

https://shopify.engineering/reducing-bigquery-costs

Introducing AWS Cost Anomaly Detection (Preview)

Posted On: Sep 25, 2020

https://aws.amazon.com/about-aws/whats-new/2020/09/introducing-aws-cost-anomaly-detection-prev

Companies moving away from the cloud...

Our cloud spend in 2022

Since we published why we're leaving the cloud, we've received a lot of questions about our actual spending. We're happy to share, both where we currently are and where we're going.

Fernando Álvarez
SRE, Ops

37signals

HEY

Basecamp

https://dev.37signals.com/our-cloud-spend-in-2022/

...or moving their cloud applications to more traditional architectures

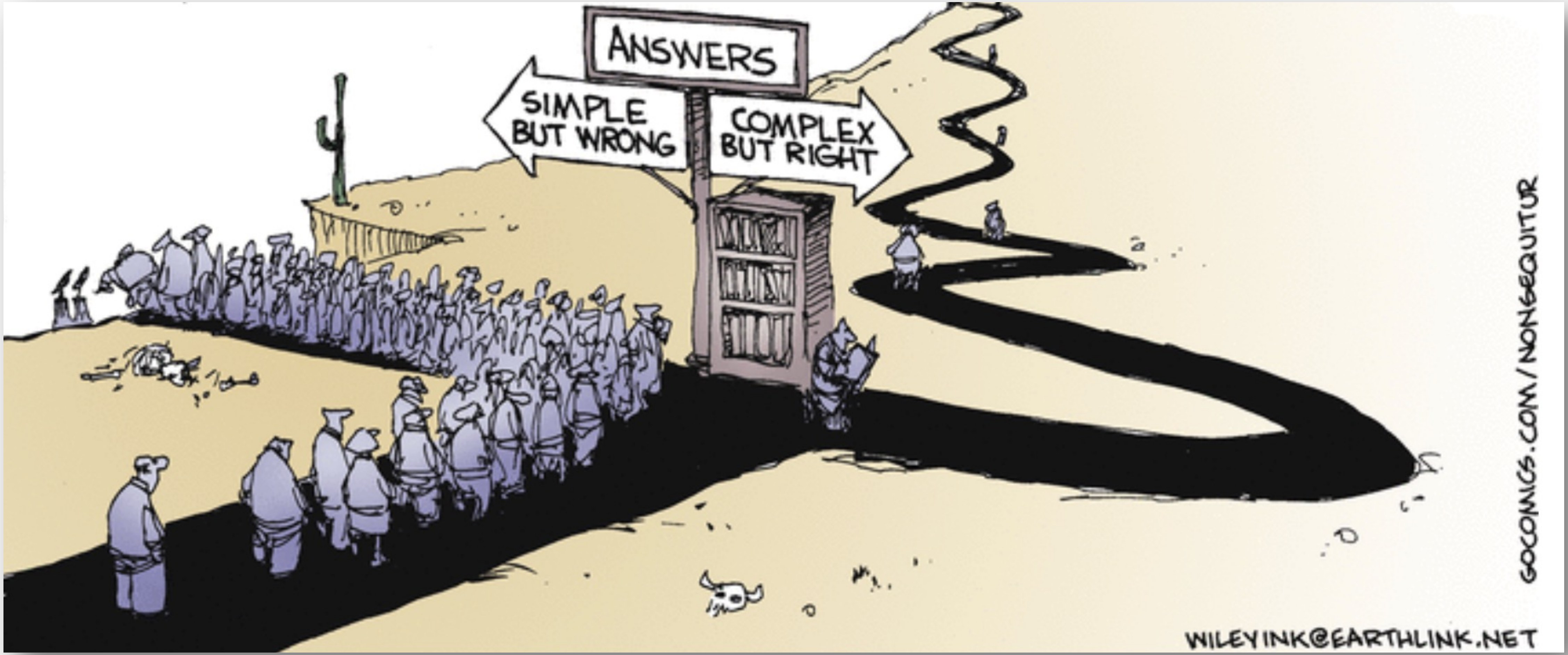
Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

prime video | TECH

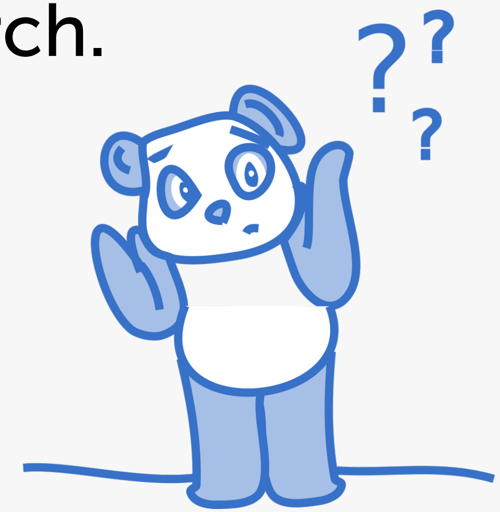
https://www.primevideotech.com/video-streaming/scaling-up-the-prime-video-audio-video-monitoring-service-and-reducing-costs-by-90

Hype-driven software engineering...



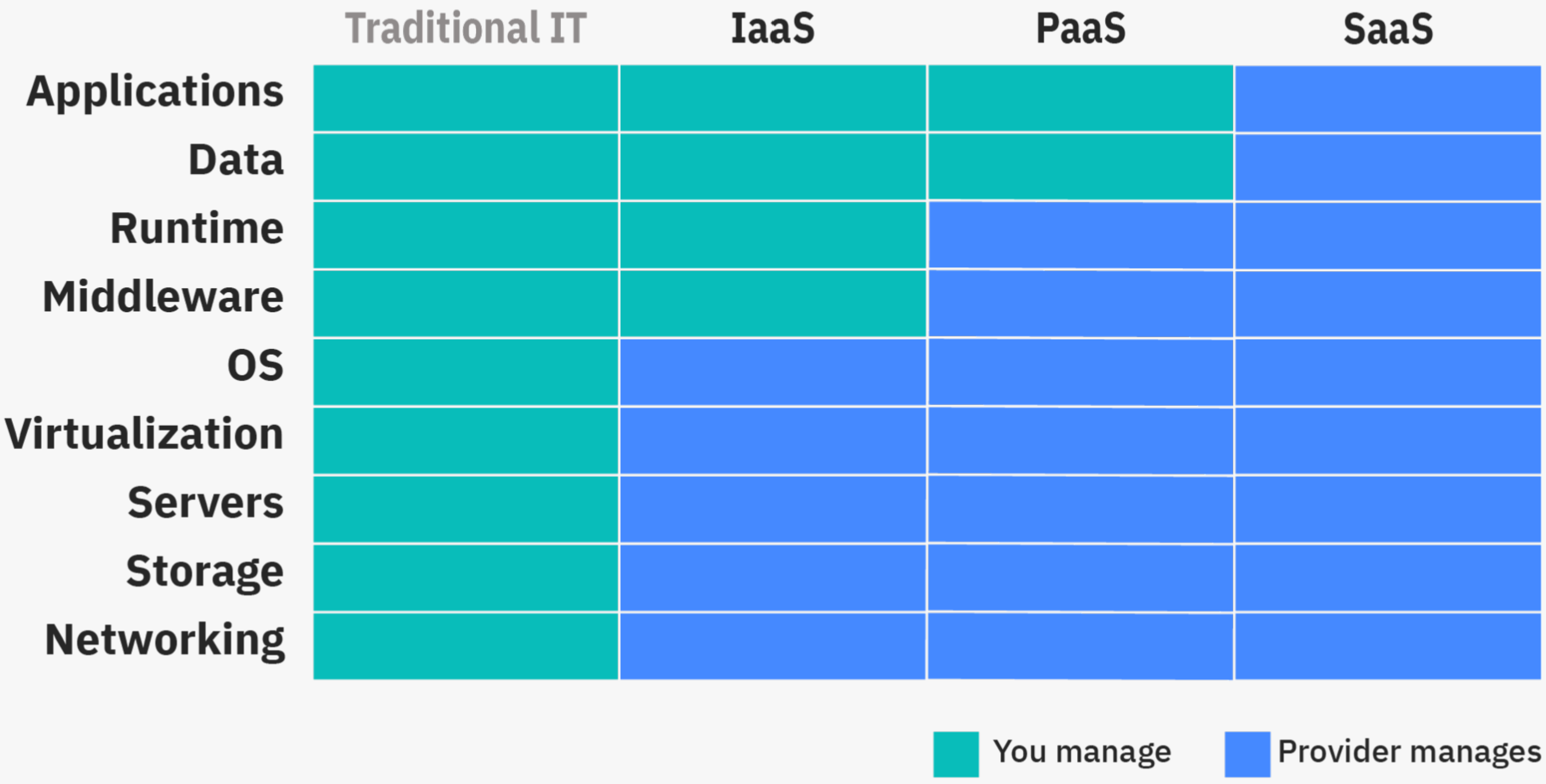
Observations

- A lot of **confusion** and **hype-driven discussions/decisions.**
- Great **opportunity for research** to step in and objectify the discussion.
- Spoiler: While **(operations) cost** is considered in related disciplines, it is an **essential but often overlooked non-functional property** in software engineering research.



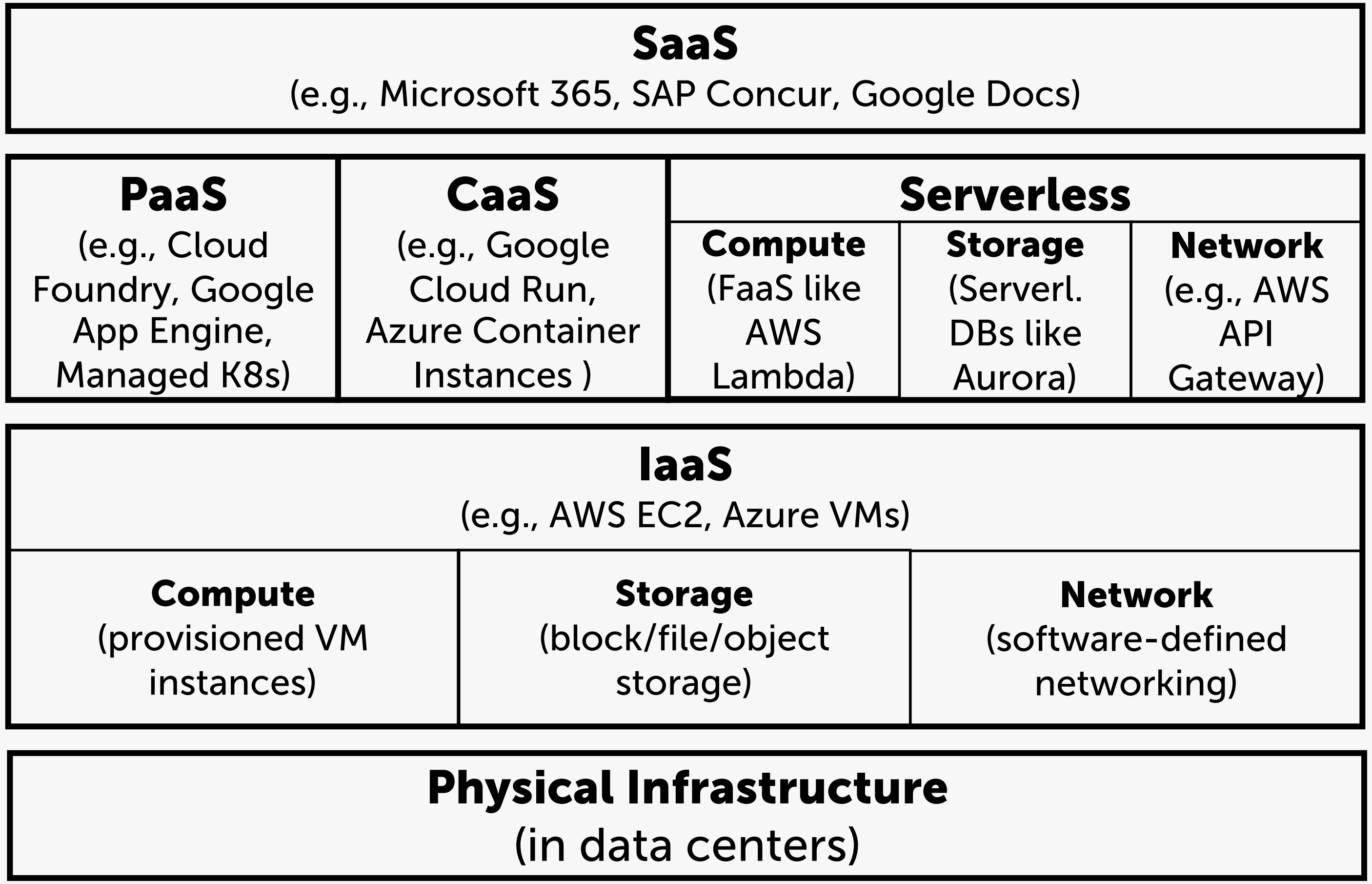
”The cloud” and its billing models

Cloud Services: Who manages what?



https://www.ibm.com/cloud/learn/iaas-paas-saas

The Cloud Stack



Can be provisioned **declaratively** via IaC files or **interactively** via web portals.

Pricing approaches in the cloud

- **Usage-based billing:**
  (aka consumption-based billing, pay-per-use, pay-as-you-go)
  Customers pay for what they use and/or how long they use a resource (by the hour/second). Billing usually monthly.

- **Subscription-based billing:**
  (aka reserved instances)
  Cusomters pay a recurring fee for a period of time, flat rate regardless of usage, for a specific configuration. Discounts often available for longer commitments, e.g., 1-3 years.

- **Hybrid approaches:**
  E.g., fixed monthly rate plus usage-based component.

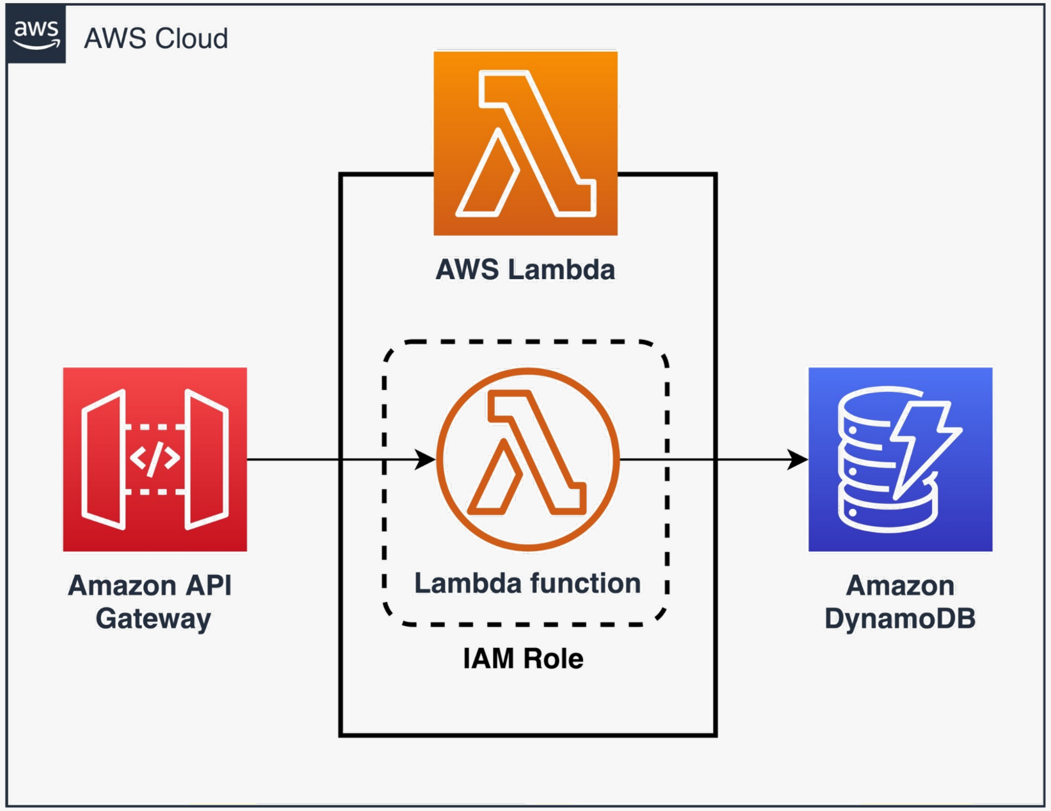
- **Special offers:**
  E.g., free tiers, transient/spot instances (unused capacity) offered at a discount (can be reclaimed if provider needs capacity)

Function-as-a-Service (FaaS)

- Cloud-computing service that allows to **execute code in response to events**, without managing complex infrastructure.
- “Serverless” offering

```java
public class LambdaRequestHandler
  implements RequestHandler<String, String> {
    public String handleRequest(String input, Context context) {
        context.getLogger().log("Input: " + input);
        return "Hello World - " + input;
    }
}
```

https://www.baeldung.com/java-aws-lambda



https://aws.amazon.com/blogs/architecture/field-notes-optimize-your-java-application-for-aws-lambda-with-quarkus/

Usage-based billing: AWS Lambda

- **Duration** a function was executed (rounded up to **ms**).
- **Price** depends on the **amount of memory** allocated to function.
- **CPU** power and other resources **proportionally allocated.**

AWS Lambda Pricing

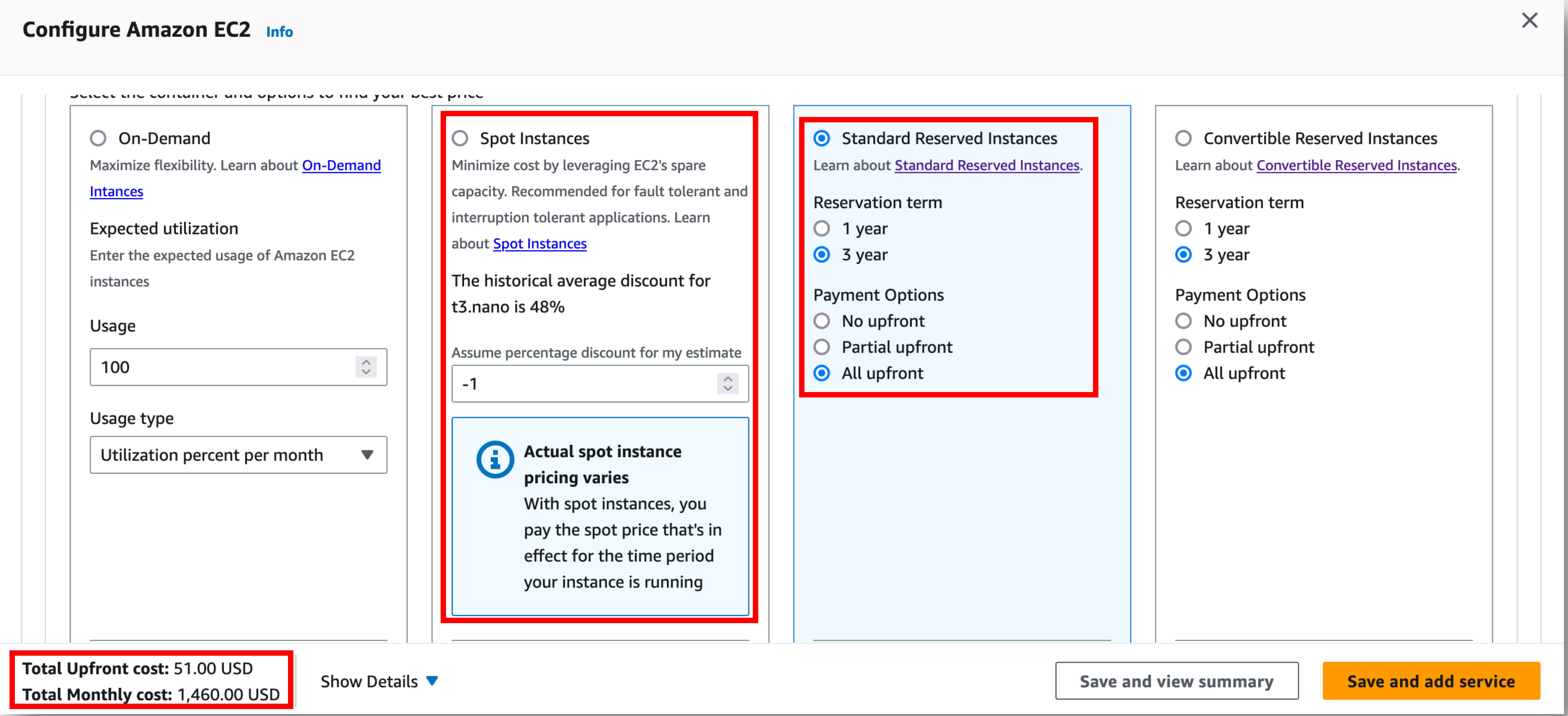
Region: US East (Ohio)

Architecture	Duration	Requests
x86 Price		
First 6 Billion GB-seconds / month	$0.0000166667 for every GB-second	$0.20 per 1M requests
Next 9 Billion GB-seconds / month	$0.000015 for every GB-second	$0.20 per 1M requests
Over 15 Billion GB-seconds / month	$0.0000133334 for every GB-second	$0.20 per 1M requests
Arm Price		
First 7.5 Billion GB-seconds / month	$0.0000133334 for every GB-second	$0.20 per 1M requests
Next 11.25 Billion GB-seconds / month	$0.0000120001 for every GB-second	$0.20 per 1M requests
Over 18.75 Billion GB-seconds / month	$0.0000106667 for every GB-second	$0.20 per 1M requests

Memory (MB)	Price per 1ms
128	$0.0000000021
512	$0.0000000083
1024	$0.0000000167
1536	$0.0000000250
2048	$0.0000000333
3072	$0.0000000500
4096	$0.0000000667
5120	$0.0000000833
6144	$0.0000001000
7168	$0.0000001167
8192	$0.0000001333
9216	$0.0000001500
10240	$0.0000001667

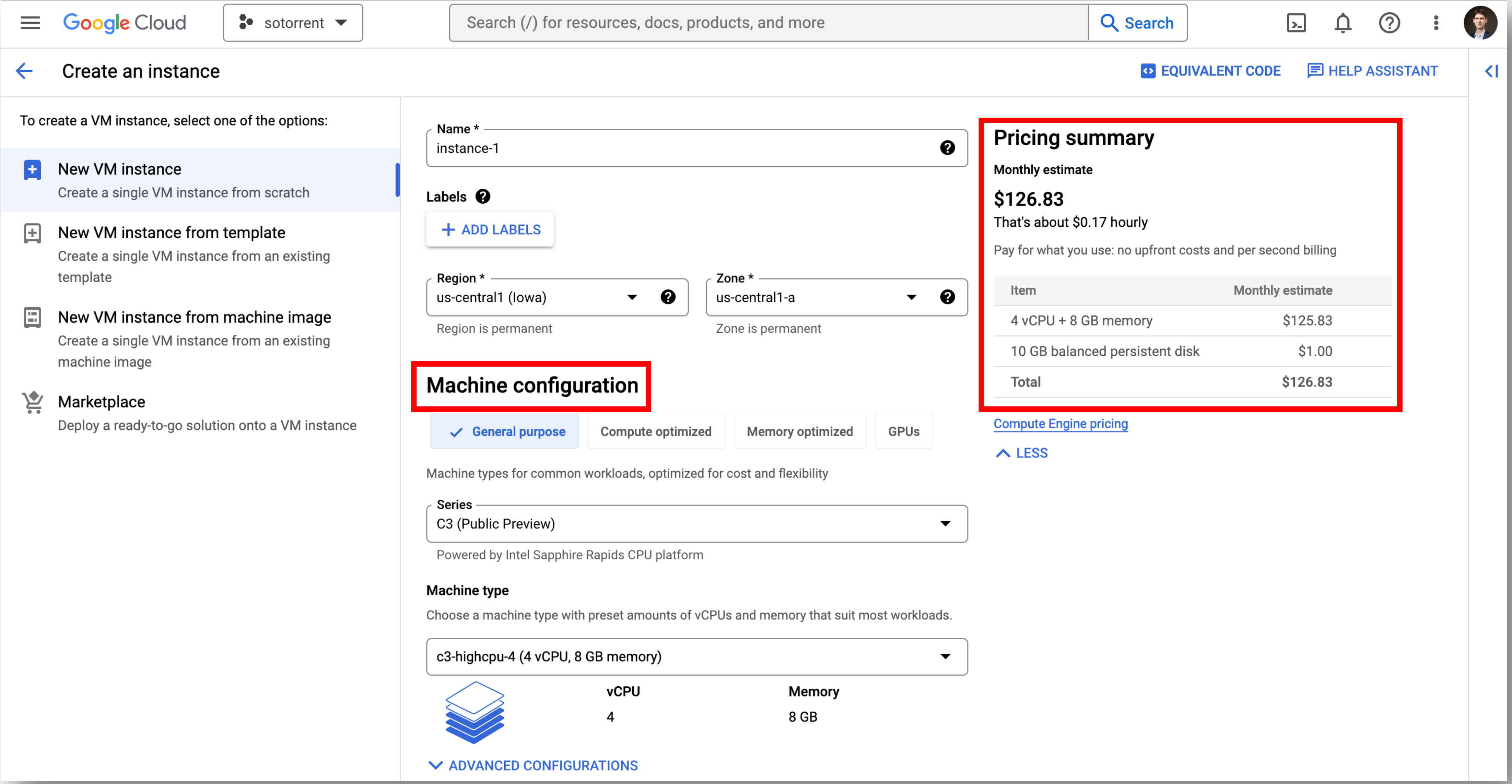
https://aws.amazon.com/lambda/pricing/

Subscription-based billing: Amazon EC2



https://calculator.aws/#/addService/ec2-enhancement

Provisioning via Web UI: Google Cloud



Infrastructure-as-Code (IaC): Terraform

HashiCorp Terraform

Example Usage

```
resource "google_service_account" "default" {
  account_id   = "service_account_id"
  display_name = "Service Account"
}

resource "google_compute_instance" "default" {
  name         = "test"
  machine_type = "e2-medium"
  zone         = "us-central1-a"

  tags = ["foo", "bar"]

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
      labels = {
        my_label = "value"
      }
    }
  }
```

https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_instance

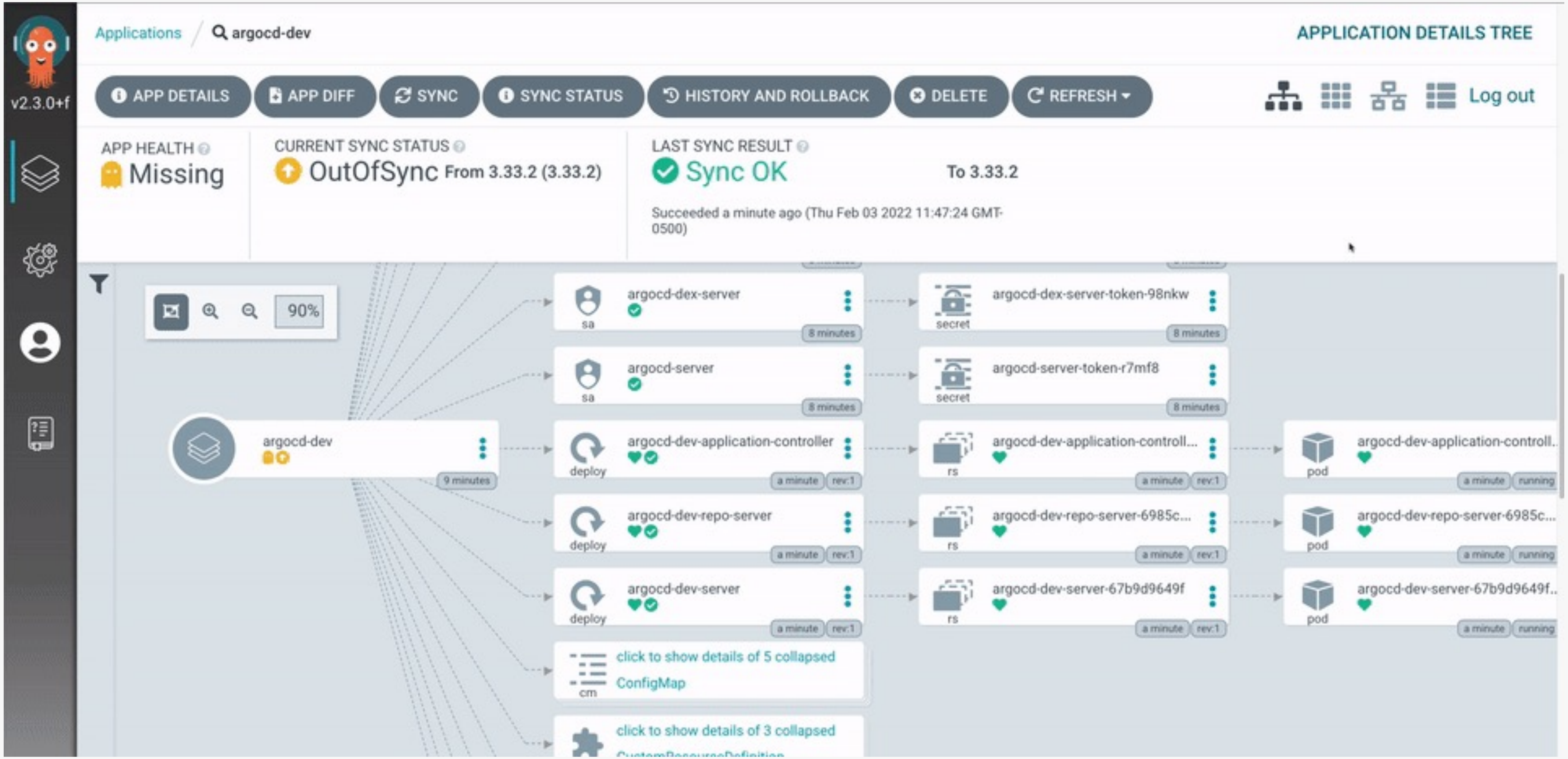
GitOps

Goal: Achieving the following properties for a (usually Kubernetes-based) GitOps-managed system:

1. **Declaratively** defined desired state.
2. **Versioned and immutable** desired state.
3. Software agents **automatically pull** desired state declarations from source.
4. Software agents **continuously observe** actual system state and **attempt to apply** desired state.

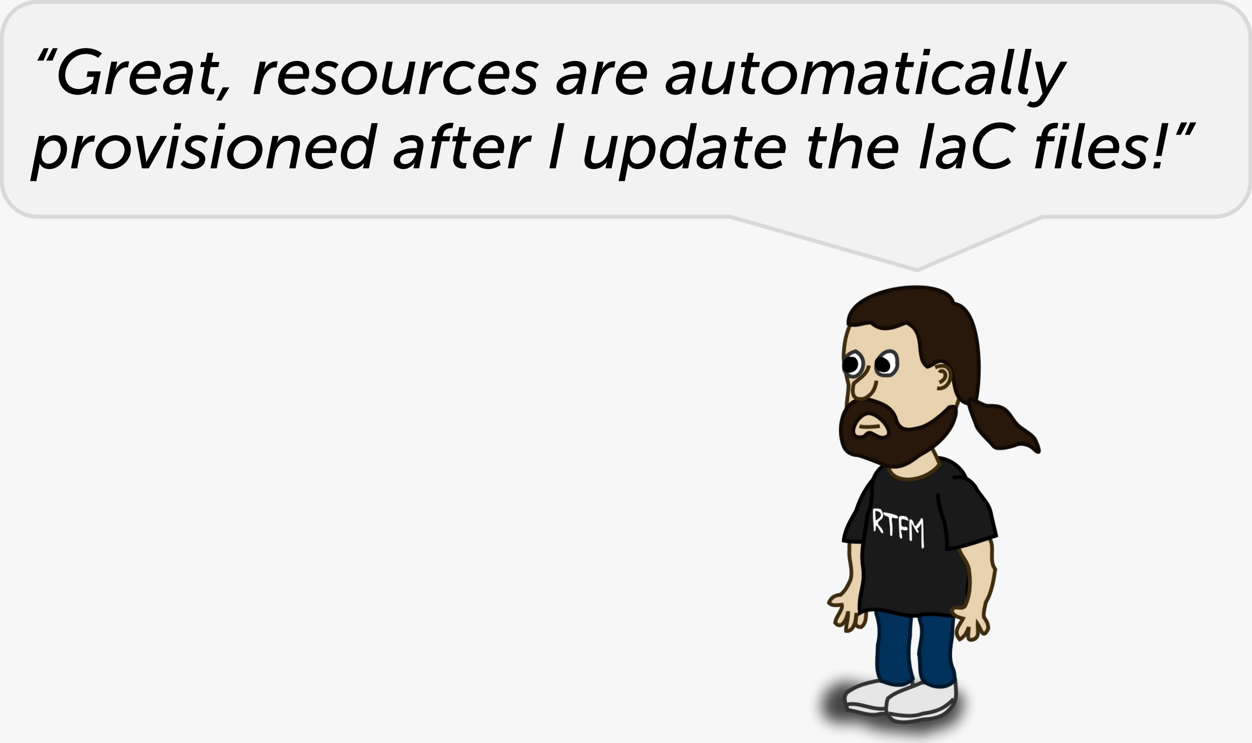
https://github.com/readme/featured/defining-gitops

GitOps: ArgoCD



https://argo-cd.readthedocs.io/en/stable/

GitOps



https://tinyurl.com/what-could-go-wrong-cartman

Cost transparency in the cloud is a problem

Hacker News new | past | comments | ask | show | jobs | submit

▲ Tell HN: I DDoSed myself using CloudFront and Lambda Edge and got a $4.5k bill
274 points by huksley 5 months ago | hide | past | favorite | 333 comments

https://news.ycombinator.com/item?id=31907374

@donkersgoed@hachyderm.io
@donkersgood

How a single-line bug cost us $2000 in AWS spend...

We recently refactored a Lambda Function. We extensively tested its functionality and released it into production. And everything still worked as expected. But then the billing alarm went off..

https://twitter.com/donkersgood/status/1635244161778737152

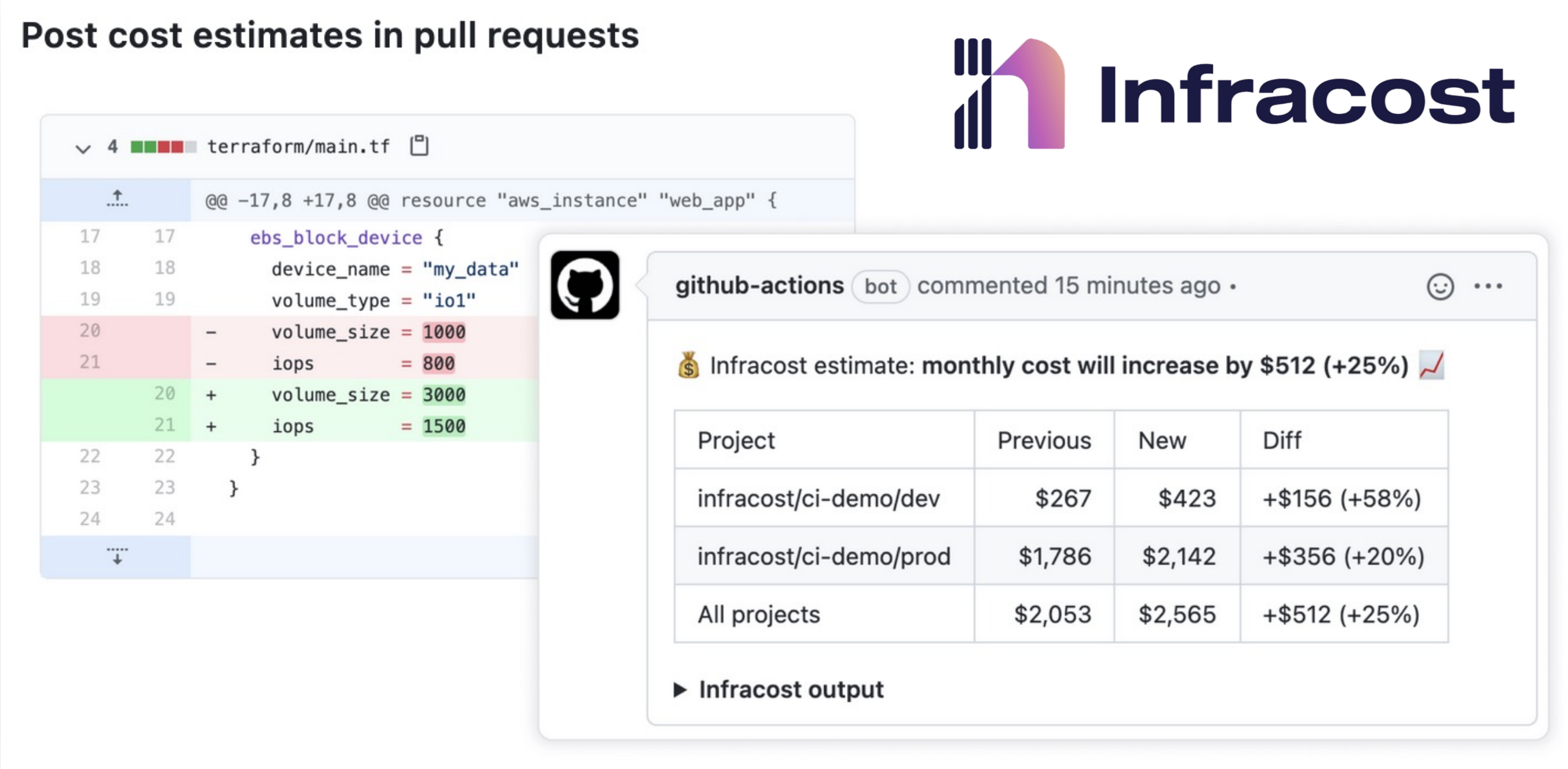
Reducing BigQuery Costs: How We Fixed A $1 Million Query

by Calvin Zhou • Data Science & Engineering
Nov 3, 2022 • 3 minute read

shopify

https://shopify.engineering/reducing-bigquery-costs

Mitigations: Infracost

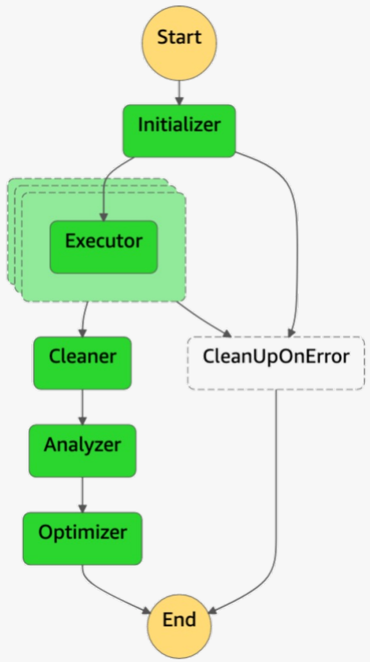


https://github.com/infracost/infracost

- Supports over 1,100 **Terraform** resources across AWS, Azure and Google (no other IaC formats)
- Focuses rather on **guardrails and policies** than on supporting **architecture decision making** (e.g., *"With certain workload assumptions, when will the decision to use serverless backfire?"*)

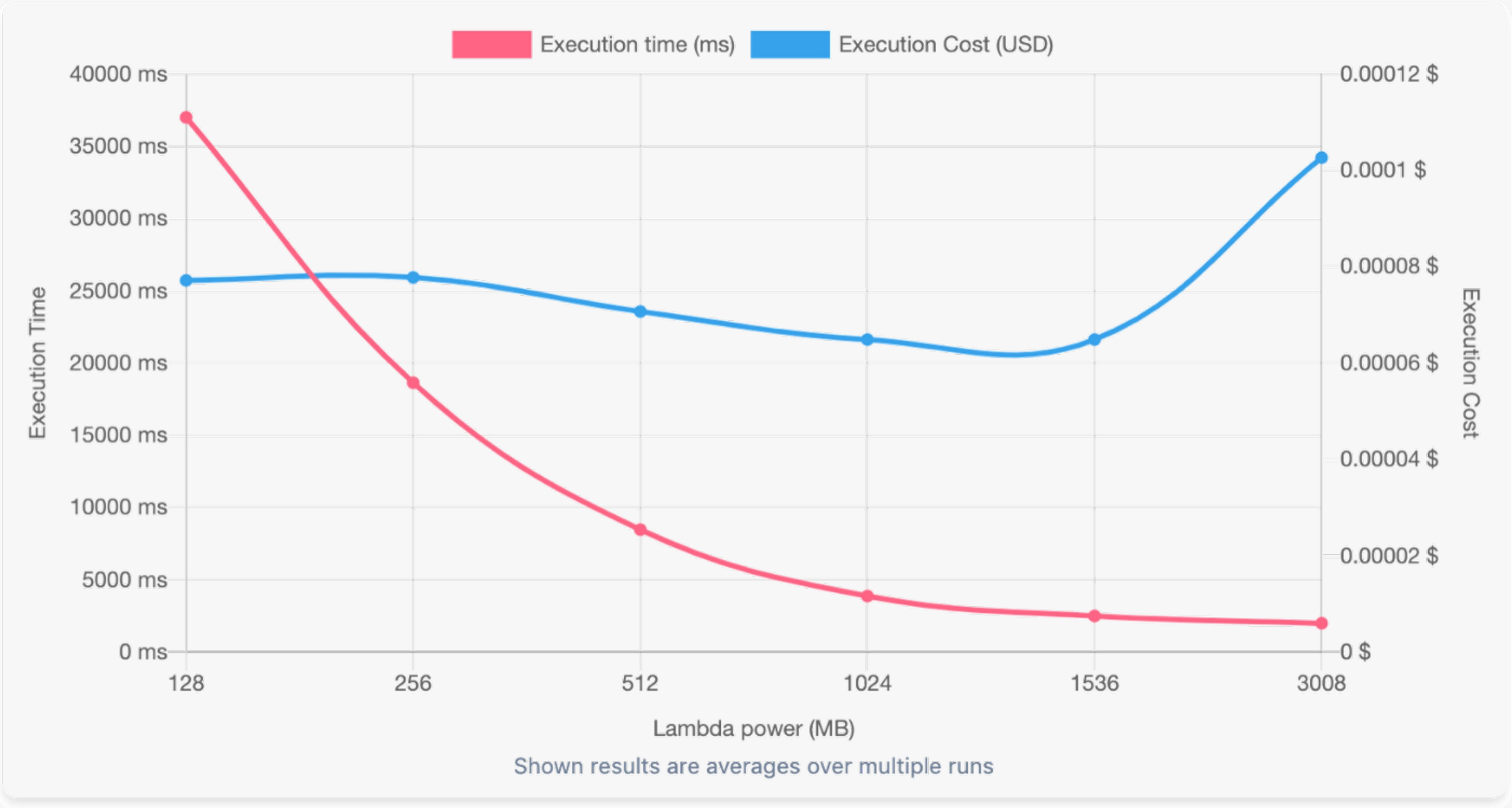
Mitigations: AWS Lambda Power Tuning

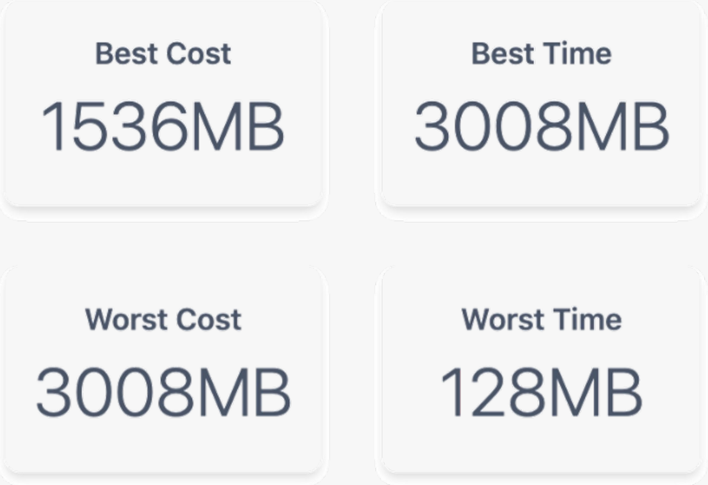
- AWS Lambda Power Tuning helps **optimize Lambda functions for cost and/or performance** in a data-driven way.
- **Invokes a given Lambda function with multiple configuration**, then **analyzes execution logs**, suggests best configuration minimizing cost and/or maximizing performance.
- Limitations:
  - *"Please note that the input function will be executed in your AWS account."*
  - Focus on on induvial functions (local vs. global optima)



https://github.com/alexcasalboni/aws-lambda-power-tuning

Mitigations: AWS Lambda Power Tuning





https://github.com/alexcasalboni/aws-lambda-power-tuning

Mitigation: OpenCost

- Vendor-neutral open source project for **measuring and allocating infrastructure and container costs in real time**.
- *"OpenCost shines a light into the black box of Kubernetes spend."*
- *"Real-time cost allocation, broken down by Kubernetes concepts down to the container level."*

→ More fine-grained reporting for K8s, reduce reporting delay.

OpenCost

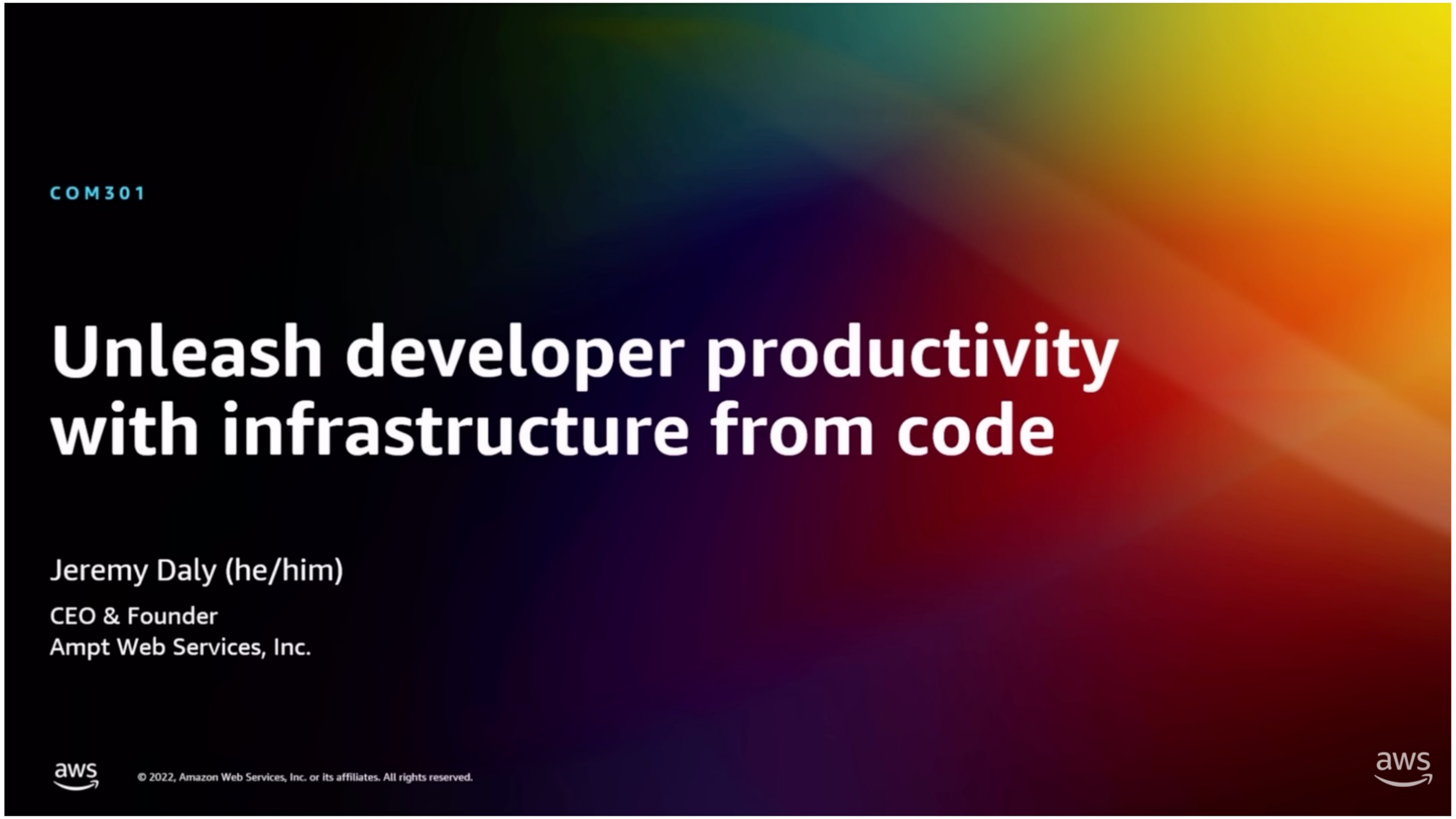
https://www.opencost.io/

Infrastructure-from-Code (IfC)

- *"[...] logical evolution of cloud. Instead of writing low-level, control-plane specific instructions, IfC **infers requirements from application logic** and **provisions the optimal cloud infrastructure**." - infrastructurefromcode.com*

- *"Programming languages and cloud infrastructure will **converge in a single paradigm**: where all resources required will be **automatically provisioned, and optimized** by the environment that runs it." - Shawn "swyx" Wang*

Infrastructure-from-Code (IfC)



AWS re:Invent 2022 - Unleash developer productivity with infrastructure from code (COM301)

https://www.youtube.com/watch?v=RmwKBPCo7o4

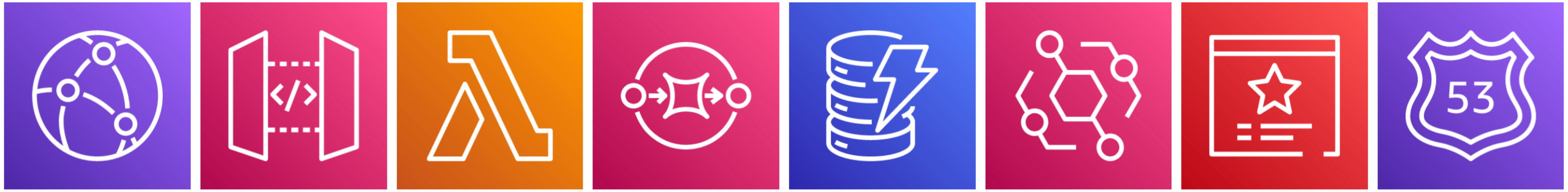
Infrastructure-from-Code (IfC)

For example, the following sample IfC implementation...

```
import { api, data, events } from '@some-ifc-sdk'

api.post("/users", async (req, res) => {
  const { email, name } = req.body;
  const newUser = await data.set(`user:${email}`, { email, name });
  res.send({ user: newUser });
});

data.on("created:user:*", ({ item }) => {
    console.log("New user created!");
    events.publish("user.created", { after: "1 day" } item)
});

events.on("user.created", (event) => {
    console.log('user.created event received!');
    // Send a follow up email, call an API, etc.
})
```

Automatically provisions and configures Amazon API Gateway

...when deployed to AWS, would automatically provision and configure the following resources...



...including mapping IAM permissions between services.

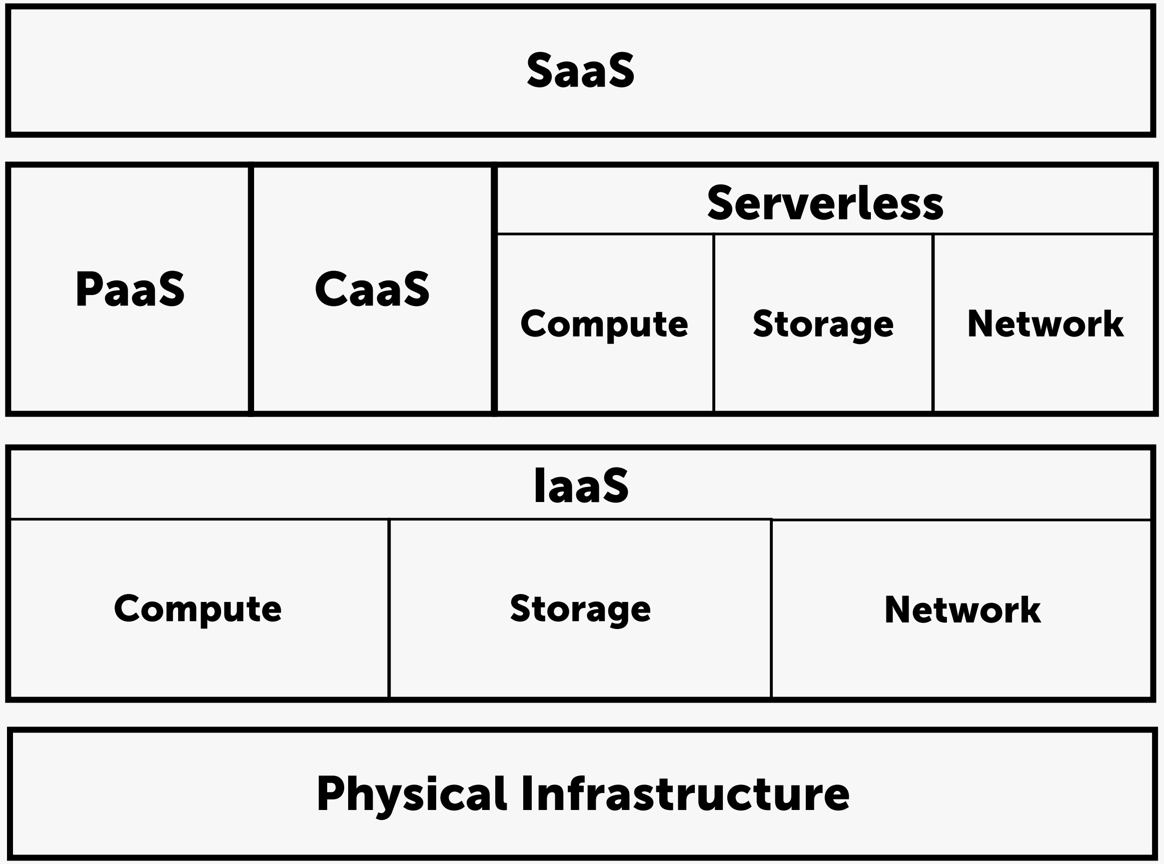
https://www.infrastructurefromcode.com/

Cost-aware architecture decision making for cloud applications

Cost-aware cloud architecture decisions

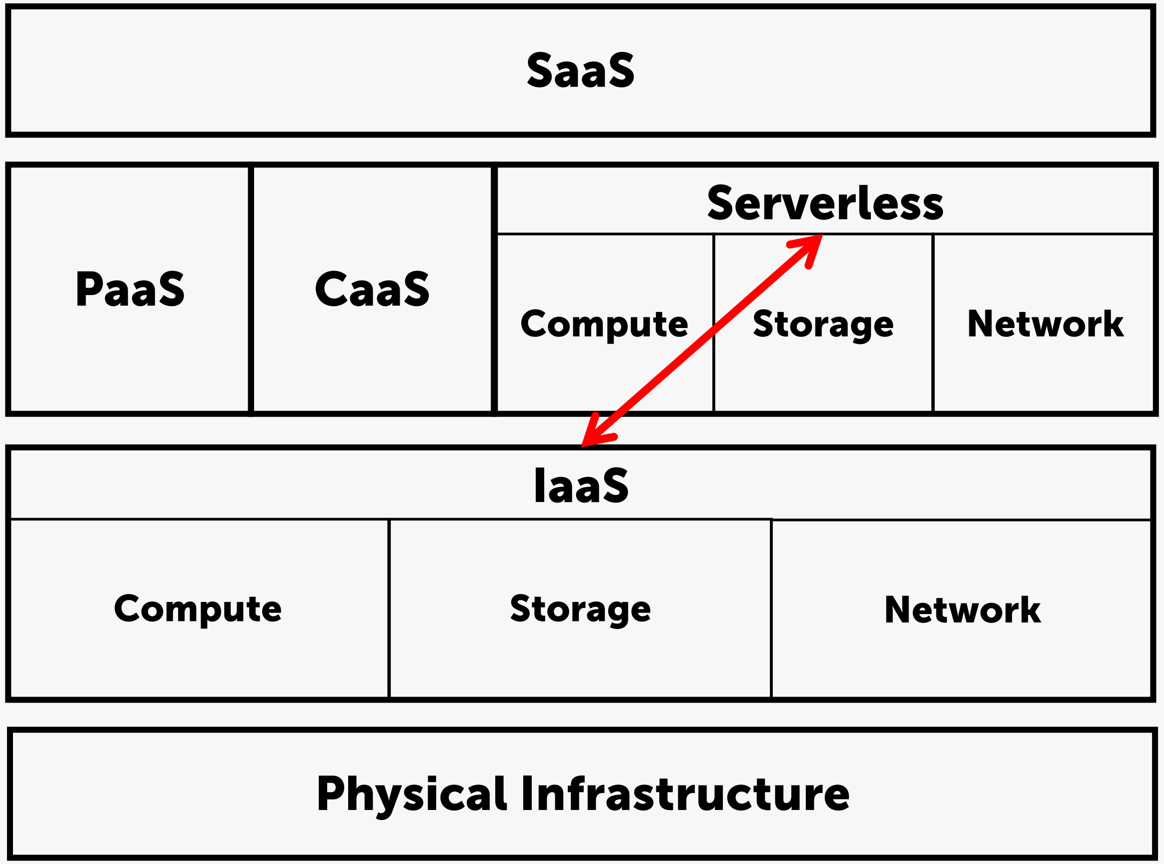
- Cloud-native developers frequently **modify IaC configs within editors/IDEs.**
- **Cost monitoring/estimation** tools available in web portals, mainly considered **downstream** task.
- Cost considerations need to be moved closer to **software architecture decision making.**
- Related topic: **Cloud resource demand management.**

Cost-aware cloud architecture decisions



A **vendor-agnostic cost model** for predicting compute and storage costs helping to reason about tradeoffs.

Cost-aware cloud architecture decisions



Potential questions:

- *For a given expected workload, is it cheaper to utilize usage-based serverless offering or a subscription-based IaaS offering?*
- *Is a specific FaaS offering cheaper at AWS compared to Azure for a given workload?*

Minimal information required for a cost model

- Description/operationalization of **modeled resources**, e.g.,
  - Compute
  - Storage
  - Network
- Description of a **workload**
  - Database: Query, Dataset
  - Serverless: Function inputs (e.g., JSON), abstract description of runtime properties of function(s)
  - PaaS/CaaS/IaaS offerings: Much more complicated
- **Evolution of the workload** over time
  - Short-term peaks
  - Long-term development

The company perspective

- **Scenario:** A company wants to offer a novel database system aaS.
- Given a set of **benchmark workloads**, how to determine which cloud provider's IaaS setup is **cheaper in which scenarios without executing** (all of) the workloads?
- Once the system is live: When optimizing queries, there might be cases where a slight decrease in **performance** leads to significant **cost savings**.
- Input for cost model: query and dataset properties.

The research perspective

Software Engineering (SE)

- SE research focuses on **effort estimation** rather than monitoring/modeling/optimizing **operation cost**.
- However, since **DevOps** emerged, operations-related costs **moved closer to the daily work** of developers.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 1, JANUARY 1984

Software Engineering Economics

BARRY W. BOEHM

Services/Cloud Computing

2022 IEEE International Conference on Cloud Engineering (IC2E)

Streaming vs. Functions: A Cost Perspective on Cloud Event Processing

Tobias Pfandzelter[†*], Sören Henning[‡*], Trever Schirmer[†], Wilhelm Hasselbring[‡], David Bermbach[†]
[†]*TU Berlin & ECDF, Mobile Cloud Computing Research Group*
{tp,ts,db}@mcc.tu-berlin.de
[‡]*Kiel University, Software Engineering Group*
{soeren.henning,hasselbring}@email.uni-kiel.de

714 IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 7, NO. 4, OCTOBER-DECEMBER 2014

Using Parametric Models to Represent Private Cloud Workloads

Richard Wolski, *Member, IEEE*, and John Brevik

2009 IEEE International Conference on Cloud Computing

The Method and Tool of Cost Analysis for Cloud Computing

Xinhui Li, Ying Li, Tiancheng Liu, Jie Qiu, Fengchun Wang
IBM China Research Lab, BJ, 100193, China
{lixinhui, lying, liutc, qiujie, wangfc}@cn.ibm.com

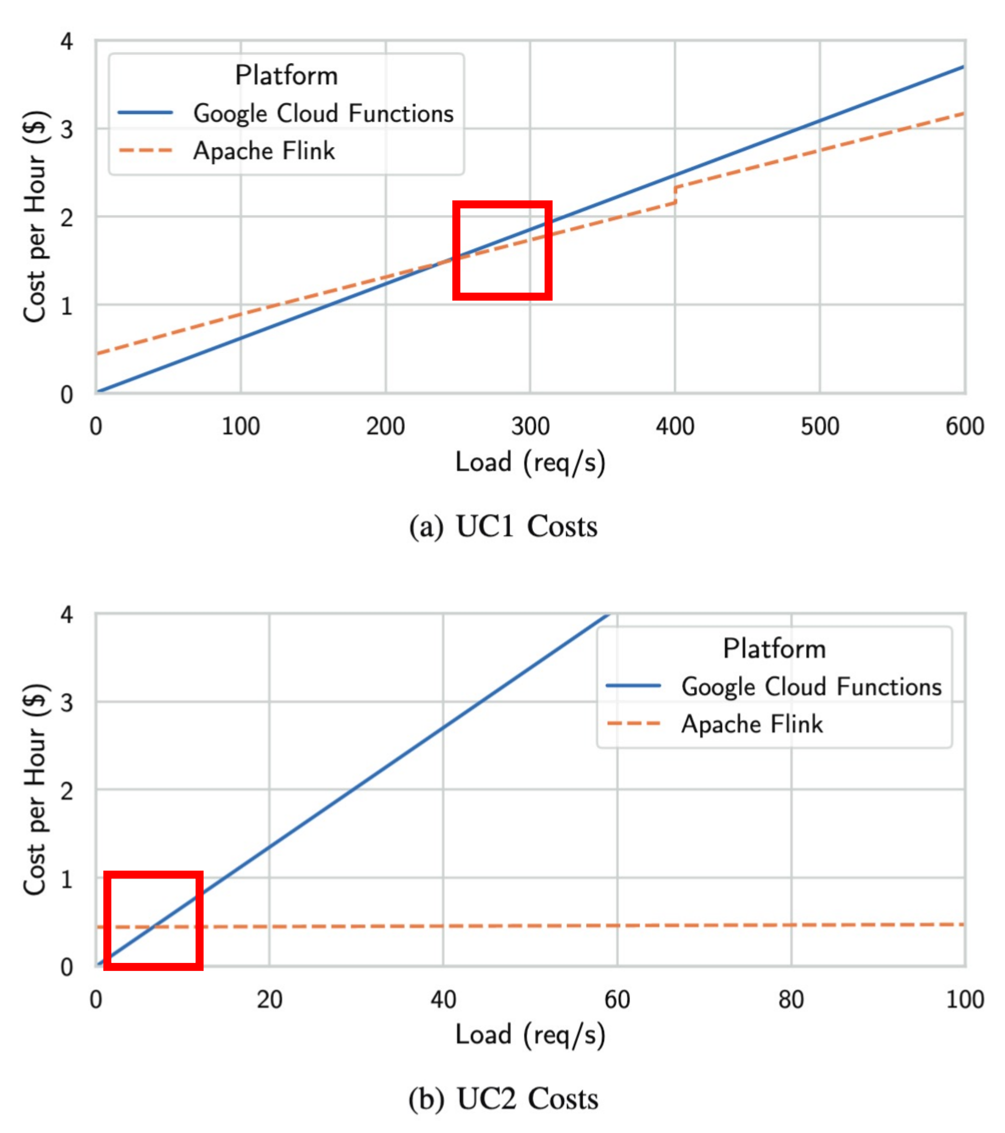
Example: Streaming vs. Functions

2022 IEEE International Conference on Cloud Engineering (IC2E)

Streaming vs. Functions: A Cost Perspective on Cloud Event Processing

Tobias Pfandzelter[†*], Sören Henning[‡*], Trever Schirmer[†], Wilhelm Hasselbring[‡], David Bermbach[†]
[†]*TU Berlin & ECDF, Mobile Cloud Computing Research Group*
{tp,ts,db}@mcc.tu-berlin.de
[‡]*Kiel University, Software Engineering Group*
{soeren.henning,hasselbring}@email.uni-kiel.de

- *UC1:* stateless storage use-case
- *UC1:* stateful sliding window aggregation use-case



(a) UC1 Costs

(b) UC2 Costs

Databases

Towards Cost-Optimal Query Processing in the Cloud

Viktor Leis
viktor.leis@fau.de
Friedrich-Alexander-Universität Erlangen-Nürnberg

Maximilian Kuschewski
maximilian.kuschewski@fau.de
Friedrich-Alexander-Universität Erlangen-Nürnberg

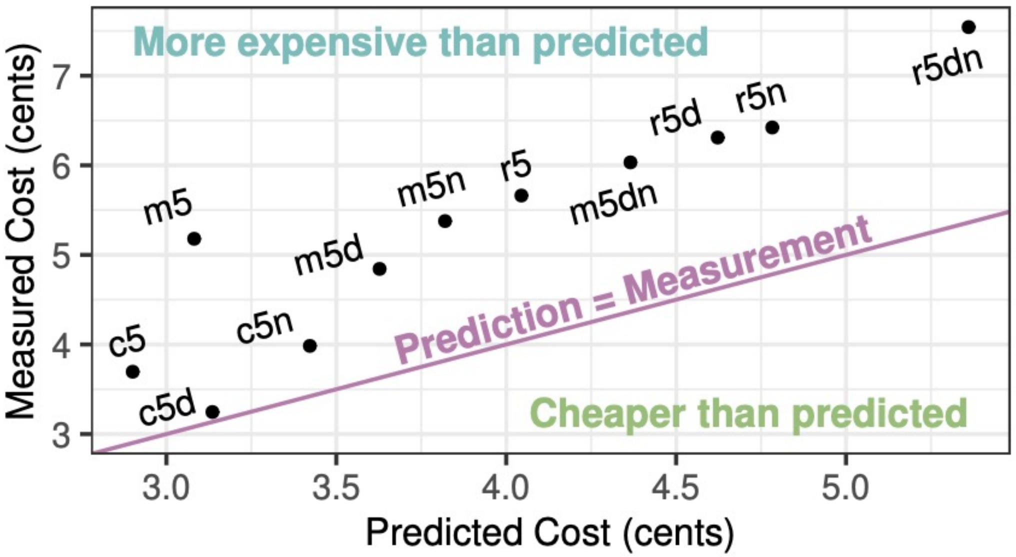


Figure 5: Prototype measurement vs. prediction on a 100 GB aggregation query

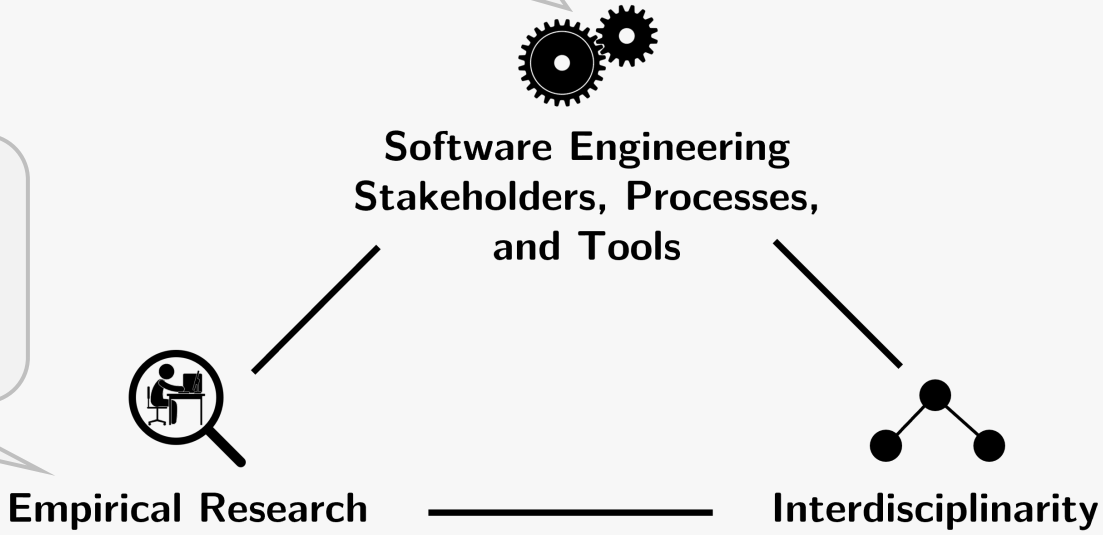
Cost-aware
Cloud Architecture

Potential next steps

Support engineers and organizations in choosing **suitable cloud architectures**, shifting **cost transparency** left using appropriate tooling.

Mining GitHub for **typical IaC/IfC setups**, retrieving **workloads characteristics** from observability data.

Existing work on **cost modeling** in other research communities.



Takeaways

- **Cost transparency is a problem** for cloud applications.
- Research mainly focused on cost-optimizing **database** or **serverless** workflows.
- More research needed on **cost models** allowing reasoning between cloud layers and vendors, particularly on the **long run** (”lock-in”).
- Cost transparency needs to **be integrated into tools** that modern software/platform engineers use (“shifting left”).
- Cost **optimization** needs to **consider other non-functional requirements** such as performance, scalability, elasticity.

Interested in collaborating?
Please reach out!

Cost-aware
Cloud Architecture

empirical-software.engineering

Dr. Sebastian Baltes