

Cloud solutions

Marek Czajkowski
RTB House

Which topics will be covered today?

What is cloud computing and why it is used?

What are cloud delivery models?

Cloud security

Common cloud patterns

Cloud pricing models

Hybrid solutions and real-life scenarios



What is cloud computing?

“... a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies”

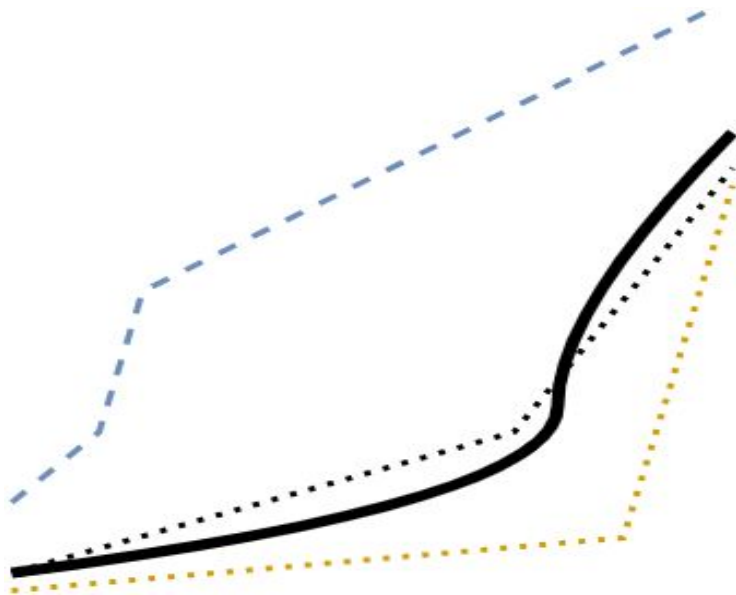
OR

“... a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources”

- *scalable*
- *delivered remotely*

Why cloud? (Business drivers)

- Capacity planning
 - Lead
 - Lag
 - Match
- Cost reduction
- Organizational agility



Why not earlier?

Technology innovations which enabled cloud computing:

- Clustering
- Grid computing
- Virtualization
- Broadband network and Internet architecture
- Data Center technology
- Web technology
- Multitenant technology
- Service technology

Scaling

Ability of the IT resource to handle increased or decreased usage demands

- **Horizontal** (out/in) - releasing of IT resources that are of the same type

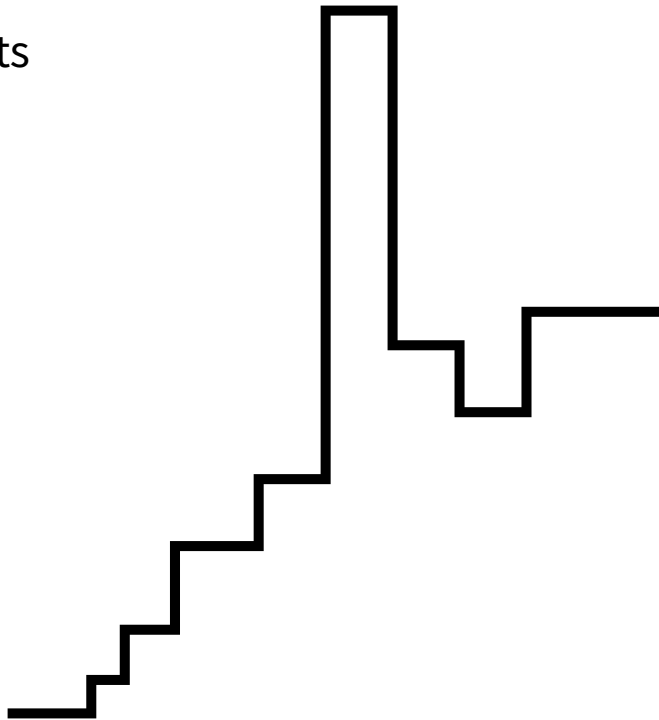
A common form of scaling within cloud environments

- **Vertical** (up/down) - when existing IT resource is replaced by another with higher/lower capacity

Less common in cloud environments due to the downtime required while the replacement is taking place (though more common in on-premise installations)

Benefits

- Reduced investments and proportional costs
 - On-demand access
 - Unlimited (?) computing resources
 - Fine-grained level
 - Abstraction
- Increased scalability
- Increased availability and reliability of
 - Watch SLA!



Risks & challenges

- Increased security vulnerabilities
 - overlapping trust boundaries can be challenging!
- Reduced operational governance control
- Limited portability between cloud providers
 - including internal egress!
- Multi-regional compliance and legal issues
 - migration between DCs

Cloud characteristics

- on-demand usage
- ubiquitous access
- multitenancy
- elasticity
- measured usage
- resiliency

Cloud delivery models

- Infrastructure-as-a-service (IaaS)
- Platform-as-a-service (PaaS)
- Software-as-a-service (SaaS)
- Other
- Combined

IaaS (Infrastructure as a Service)

Self-contained environment comprised of infrastructure-centric IT resources that can be accessed and managed via cloud service-based interfaces and tools

HW, network, connectivity, OS

Examples:

- Amazon S3 or Elastic Block Store
- Google Cloud Storage or Compute Engine

Types and brands of resources can vary (heterogenous)

PaaS (Platform as a Service)

Pre-defined “ready-to-use” environment typically comprised of already deployed and configured IT resources

Motivations:

- extend on-premise environment
- fully substitute on-premise environment
- become a cloud provider

Examples:

- Google App Engine or Cloud Functions
- (Amazon Elastic Beanstalk or Cloud9)

SaaS (Software as a Service)

Software program positioned as a shared cloud service and made available as a “product” or generic utility

Most commonly used model

Examples:

- Google Apps (eg Gmail)
- Dropbox
- Salesforce
- Cisco WebEx

Cloud delivery models - control levels

Model	Level of control granted to consumers	Functionality available to customer
SaaS	usage and usage-related config	access to front-end user-interface
PaaS	limited administrative	moderate level of administrative control over IT resources relevant to cloud consumer's usage of platform
IaaS	full administrative	full access to virtualized infrastructure-related IT resources and possibly to underlying physical IT resources

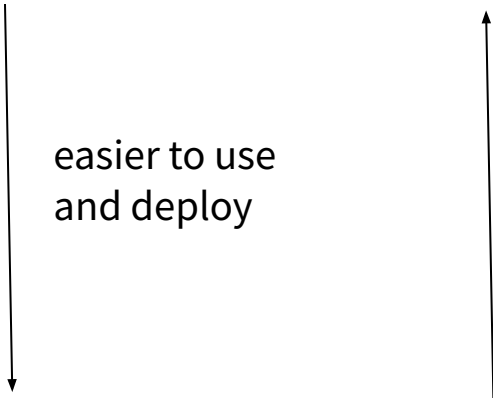
Cloud delivery models - summary

- IaaS
- PaaS
- SaaS

Additionally:

IaaS + PaaS

IaaS + PaaS + SaaS



easier to use
and deploy

more administrative rights

Cloud deployment models

- Public
- Community
- Private
- Hybrid

Cloud security



Cloud security - terms

- **Confidentiality** - being made accessible only to authorized parties
- **Integrity** - not having been altered by unauthorized party
- **Authenticity** - having been provided by an authorized source
- **Availability** - being accessible and usable during a specified time period
- **Threat** - potential security violation
- **Vulnerability** - weakness that can be exploited
- **Risk** - possibility of loss or harm arising from performing ac activity

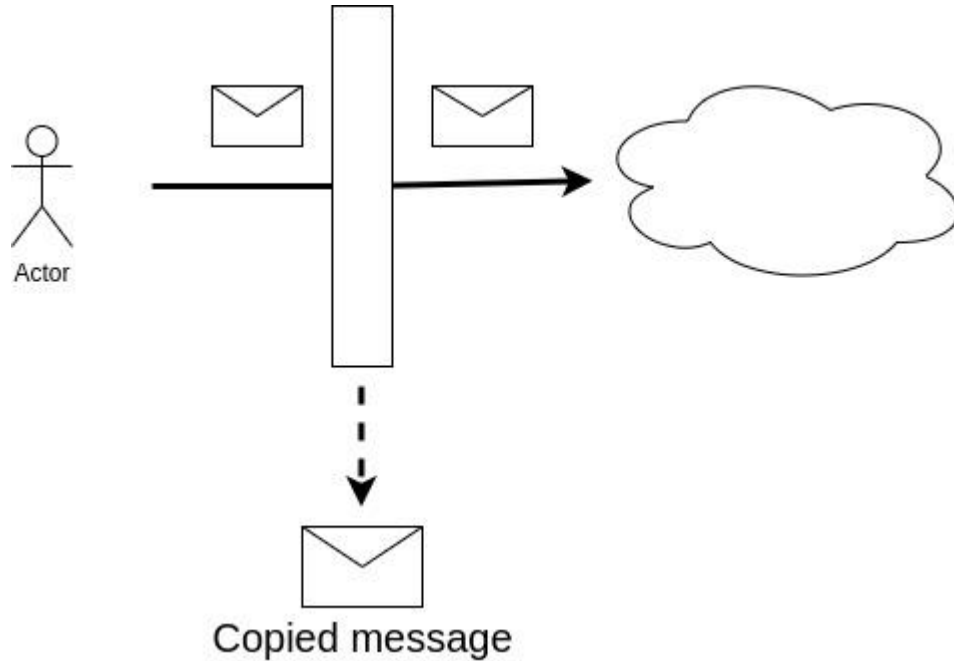
Cloud security - threat agents

- Anonymous attacker
- Malicious service agent
- Trusted attacker (malicious tenant)
- Malicious insider

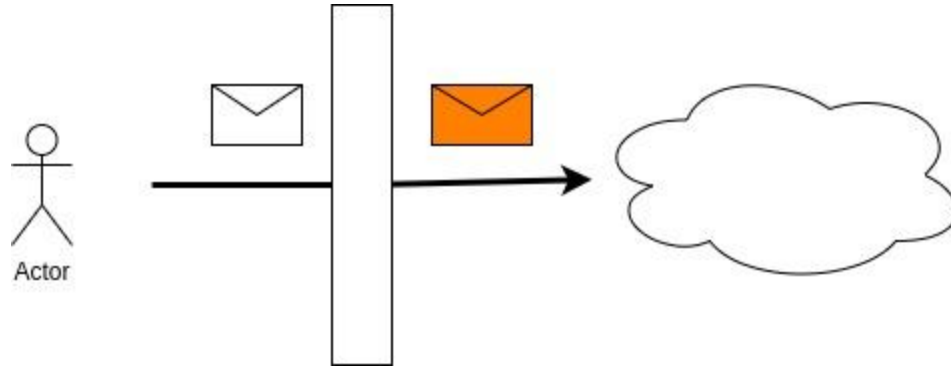
Cloud security - threats

- Traffic eavesdropping
- Malicious intermediary
- Denial of service
- Insufficient authorization
- Virtualization attack
- Overlapping trust boundaries

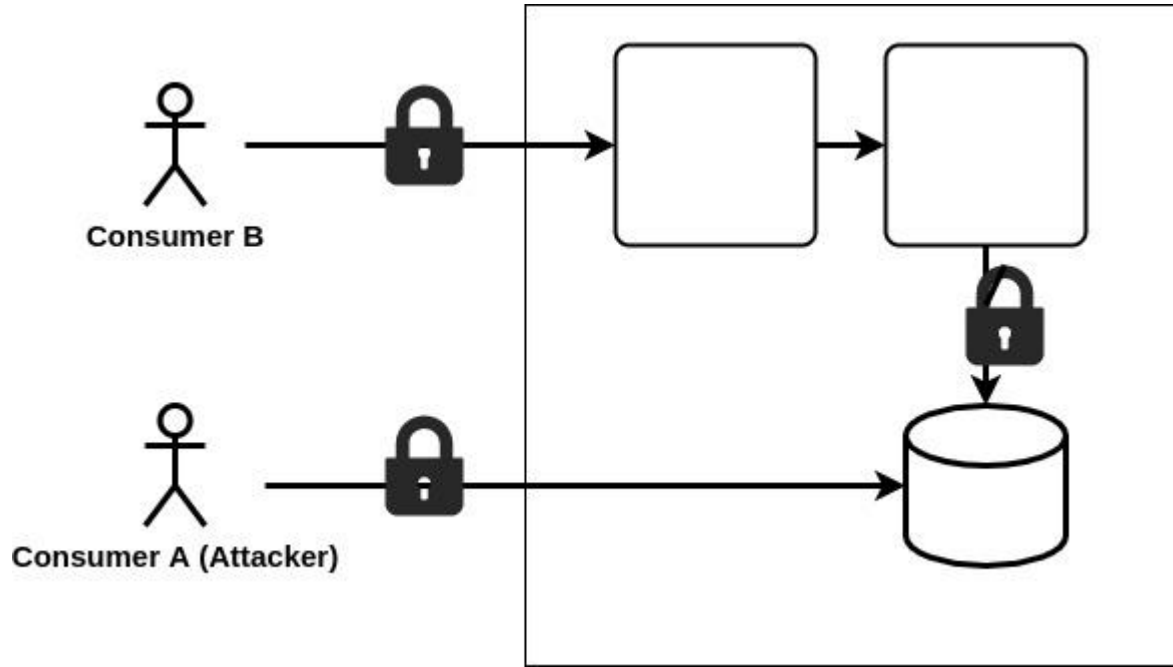
Traffic eavesdropper



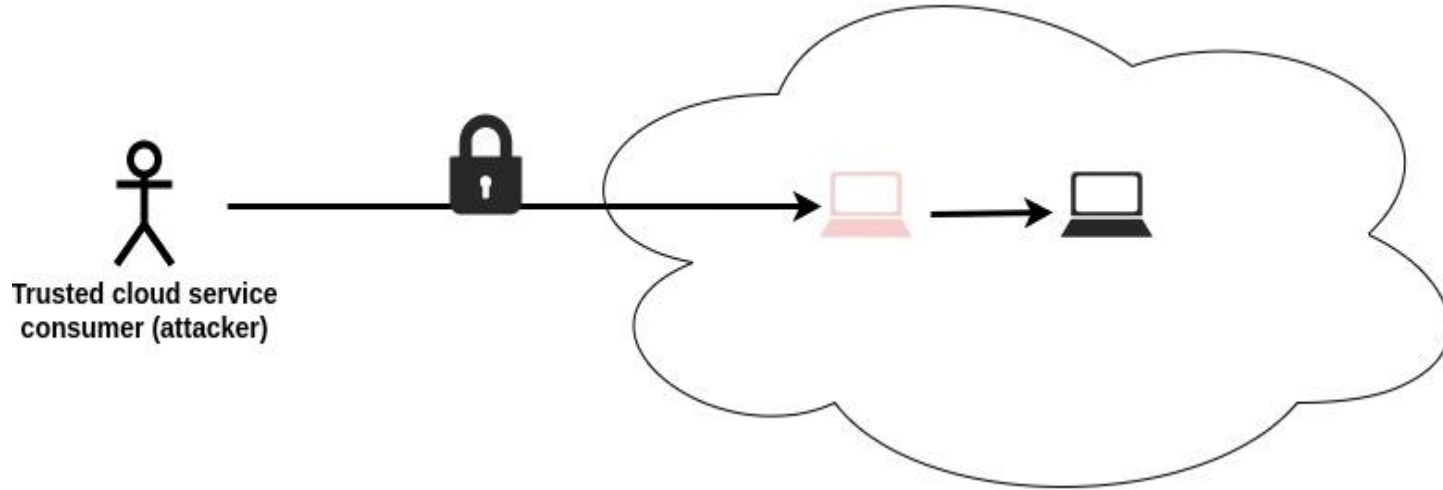
Malicious intermediary



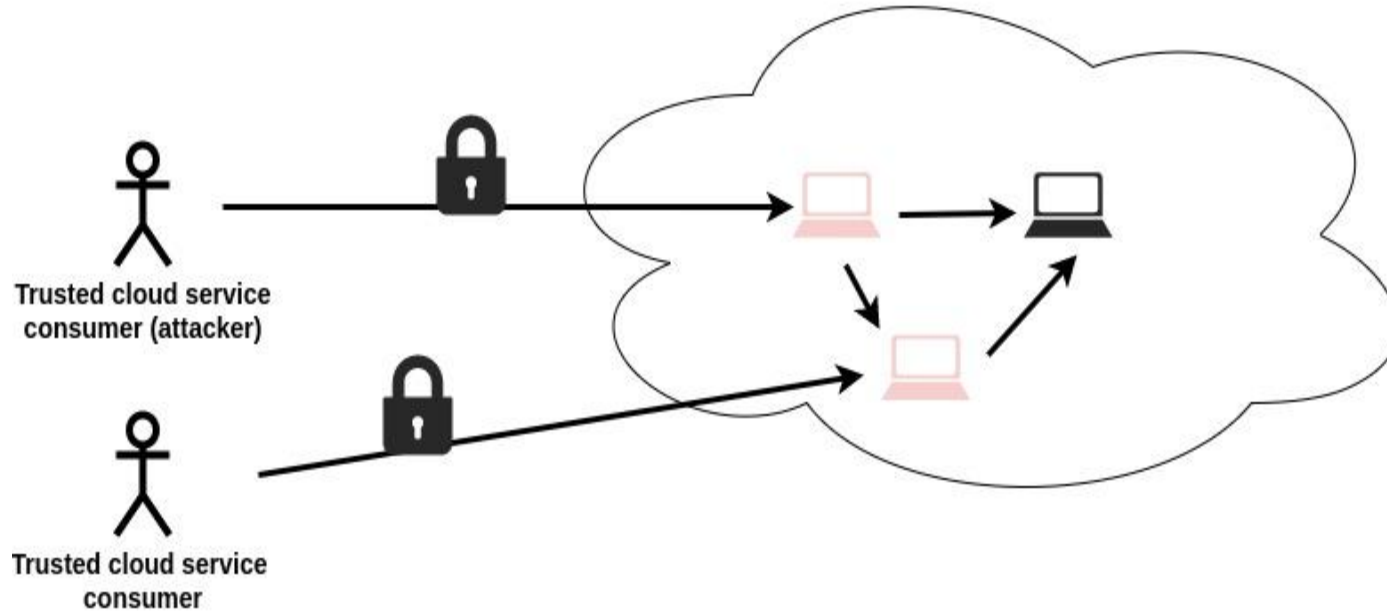
Insufficient authorization



Virtualization attack



Overlapping trust boundaries



Cloud security - additional considerations

- Flawed implementations
- Security policy disparity
- Contracts
- Risk management

Cloud architecture patterns



Horizontally scaling compute

- Fundamental pattern for cloud computing
- Scaling nodes for efficient utilization of cloud resources and operational efficiency
- Scaling is easily reversed
- Managing session state
- Autoscaling based on rules and signals (so that we minimize “HW” + “effort”)

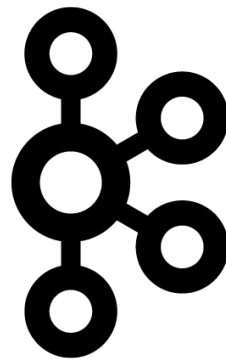


Queue centric Workflow

Pattern for loose coupling that focuses on asynchronous delivery of user requests.

Subset of CQRS

Easily to implement with Cloud (queues and storages)



Eventual consistency

Typical cloud approach is to provide **strong global consistency** for **read-after-modify**

When you upload a file to Google Cloud Storage, and you receive success response, the object is immediately available for download from any location in Google's global network.

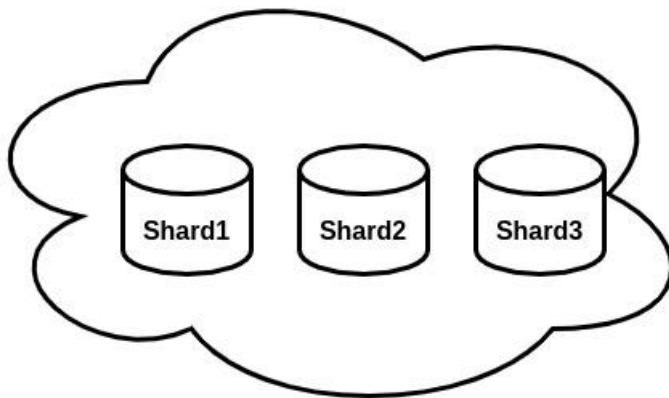
Database sharding

Database is divided into **several shards**, having the same schema

Each row appears **exactly in one shard**

Shards are **autonomous**

Sharding can be based on a **key-field, date**, etc



Busy signal

Cloud service responds with busy signal

How the application should react?

- Retry immediately
- Retry after delay (fixed or random)
- Retry with increasing delays
 - Linear or exponential backoff
 - Maximum delay set
- Throw an error

Colocate

Avoid unnecessary **network latency**.

Same **data center** or even rack

Other factors:

- Cost considerations
- Legal issues
 - GDPR
 - industries-specific regulations

... but potentially less robustness

Valet key

Efficiently using cloud storage services with untrusted clients

Use **keys or tokens** instead of direct access

As in real life: trust valet parking attendant to park your car but not to access the glove compartment

Number of keys with restricted access privileges and for a limited amount of time

Valet key vs Gatekeeper pattern

Important: **key rotation** mechanism

Multiside deployment

Install **one application in several DCs** (regions)

To some extent strategy opposed to “Collocate”

Allows for reducing **latency** when accessing resources and increase **high availability**

Geographic load balancing

Pricing motivations

Data sovereignty issues

Failover across data centers

Other cloud patterns

- Auto-scaling
- MapReduce
- Multitenancy and commodity hardware
- Network latency Primer

Hybrid approach for cloud/on-premise

Some data being kept in cloud and some on-premise

or

Same data but in different forms kept in cloud and on-premise

Motivations:

1. Gradual migration from on-premise to cloud
2. Complex processing of some data with huge volume/throughput
3. Legal issues

Real life cases

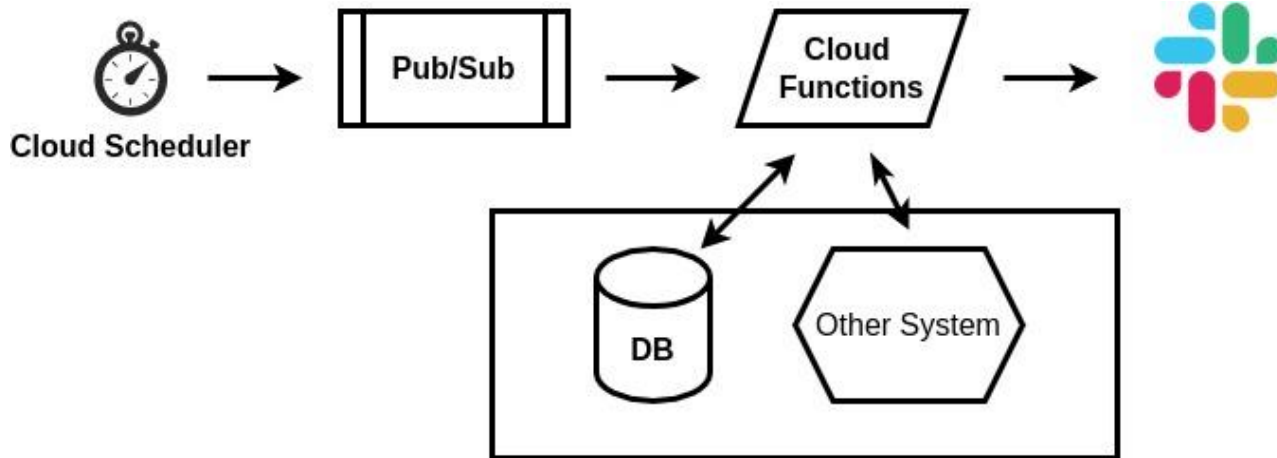


Real life case #1 - Independent alarm generator

Cloud services triggering alarm when **too few requests** processed by a platform in the recent 15 minutes

Check **scheduled** every 5 minutes

Alarm generated on a dedicated **slack** channel



Real life case #2 - Data checker and aligner (1)

Data stored both on-premise and in cloud in two or more different formats (eg BigQuery and HDFS)

- different access scenarios
 - BigQuery allows for complex queries in short time (ad-hoc analysis)
 - HDFS allows for low-cost repetitive jobs eg for machine learning
- BigQuery in Cloud, HDFS on-premise
 - costs, complexity, features
- Massive amount of data (order of 10 PBs and growing)

... but there can (and will be eventually!) discrepancy between the data sources

Real life case #2 - Data checker and aligner (2)

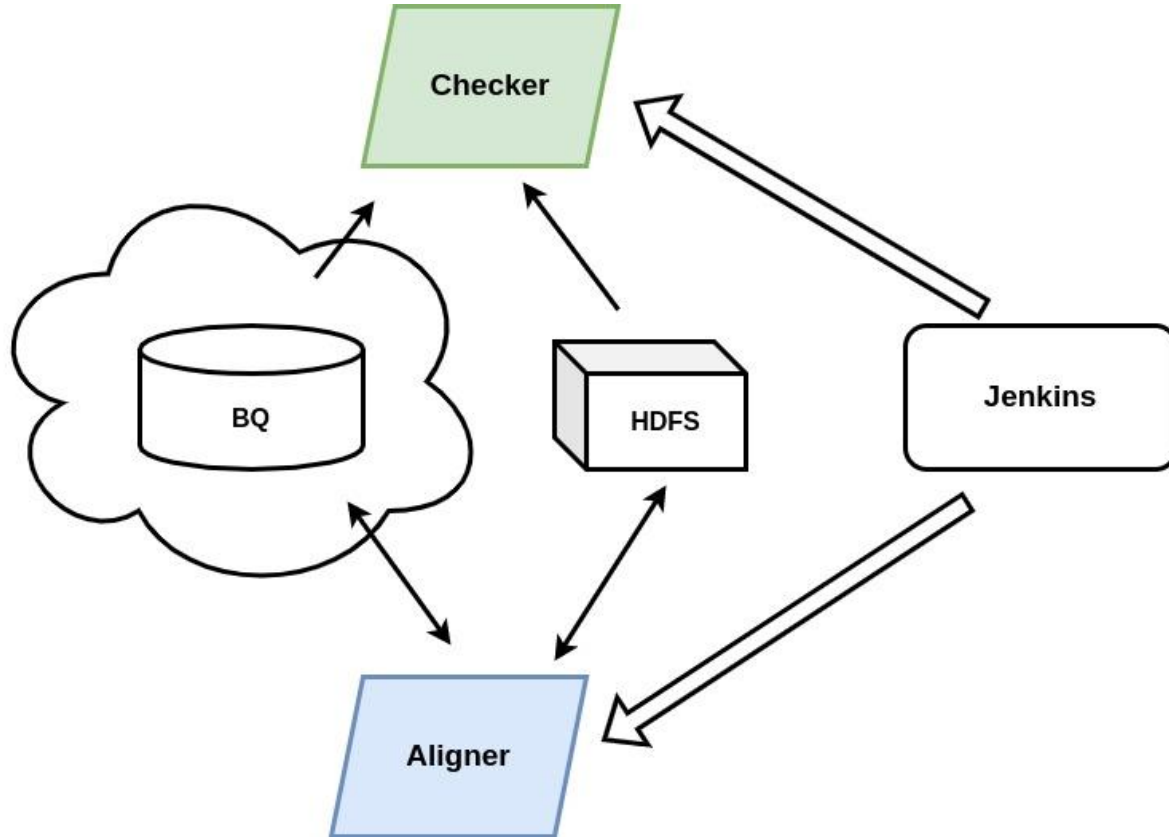
Discrepancy strategies:

- Master data source
- Minimum
- Maximum
- Majority

Need for the following applications:

- Checkers - discrepancy below 1%?
- Aligners

Real life case #2 - Data checker and aligner (3)



Real life case #3 - data kept in cloud, but processed on-premise

Various motivations:

- Special tools availability
- Complex processing, difficult to achieve (so far) on cloud
- Costs

Resource isolation & performance (1)

Important questions:

Do two virtual half computers run as fast as one physical whole computer?

Whether one person using a virtual half computer could run a CPU-intensive workload that spills over into the resources of another person using the second half computer and effectively steal some of the CPU cycles from the other person?

How about network bandwidth?

...and memory?

...or disk access?

Resource isolation & performance (2)

“Noisy neighbour” problem

Solution: bare-metal cloud

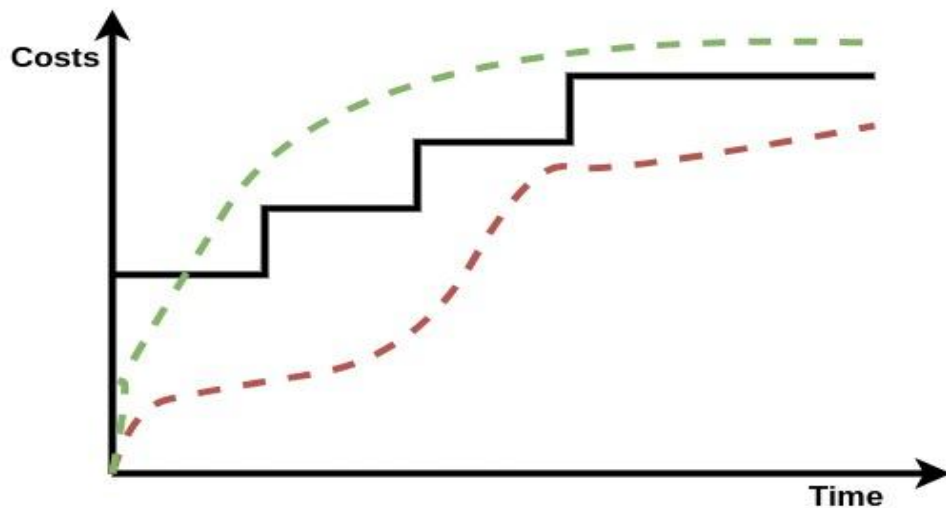
Runs one application at a time directly on the HW, which creates single-tenant environment and eliminates noisy neighbours

Container management system, eg Google Borg

the predecessor to Kubernetes

Cost metrics & pricing

There are two type of costs to take into account: **up-front** and **on-going**



Cloud usage cost metrics

- Network usage
- Server (CPU) usage
- Cloud storage device
- Cloud service
- Number of invocations
- Data ingestion/egress
- Other

Pricing model example - BigQuery (1)

- **Storage**
 - 0.02\$ per GB monthly for active storage (=20\$ per TB)
 - 0.01\$ per GB monthly for long-term storage (=10\$ per TB)
 - ... but first 10GB monthly free (start-ups!)
- **Query**
 - 5\$ per TB
 - we bill all “touched” records and fields, but no price for count(*)
- **Data ingestion**
 - Streaming 0.01\$ per 200MB (=50\$ per TB)
 - ... but batch insert free
- **Data extraction**
 - 1.1\$ per TB
 - ... but batch extraction to Google Cloud Storage free
- **Free operations**
 - Copy data, delete tables, etc

... but also available alternative pricing model (flat-rate)

Pricing model example - BigQuery (2)

Type of company	Trial mode	Start-up	Medium size	Large BigData company
Storage	20GB (0.2\$)	1TB (20\$)	100TB (3,000\$)	10PB (300,000\$)
Query	1000x100MB (0.5\$)	5,000x1GB (25\$)	20,000x10GB (1,000\$)	100,000x1TB (500,000\$)
Data ingestion	20GB streaming (1\$)	200GB streaming (10\$)	1TB streaming + batch (50\$)	100TB streaming + batch (5,000\$)
Total monthly cost	1.7\$	55\$	4,500\$	805,000\$

Pricing model example - Google Cloud Storage (1)

- Price slightly depends on location (region or multi-region)
- Data storage
 - A few classes (standard/nearline/coldline/archive) with different minimum storage duration (0/30/90/365 days) and slightly different availability (highest for standard)
 - Per TB storage per month from ca 1\$ (archive, cheapest regions) to ca \$25 (standard, most expensive regions)
- Data processing
 - Per 1M operations from \$4 (standard, class B operations) to \$500 (archive)
 - But some operations are free (mostly delete)
- Network usage
 - Egress ca 150\$ per TB
 - Ingress free
- Some other extra charges
 - Retrieval fees
 - Inter-region operations (only turbo)

Pricing model example - Google Cloud Storage (2)

Type of company	Trial mode	Start-up	Medium size	Large BigData company
Storage	20GB standard (0.5\$)	1TB standard (20\$)	30TB standard 30TB nearline 40TB archive (1,000\$)	1PB standard 2PB nearline 4PB coldline 6PB archive (90,000\$)
Processing	10k operations (0.2\$)	1M operations (10\$)	100M operations (5% on archive) (1,300\$)	10G operations (10% on archive) (150,000\$)
Network	20GB egress (3\$)	200GB egress (30\$)	10TB egress (1,500\$)	1PB egress (150,000\$)
Total monthly cost	3.7\$	60\$	4,500\$	390,000\$

Cloud pricing model - Google Cloud Functions (1)

- **Invocations**
 - 0.4\$ per 1M (first 2M free)
- **Compute Time**
 - Depends on amount of memory, #CPU and region
 - ca \$0.01 per GBxh
- **Networking**
 - Egress 120\$ per TB

Pricing model ex-le - Google Cloud Functions (2)

Type of function	Simple event-driven	High volume HTTP
Description	128MB memory 200MHz CPU 10M times per month runs 300ms each time	256MB memory 400MHz CPU 50M times per month runs 500ms, sends back 5kB data
Invocations	10M (3.2\$)	50MB (19.2\$)
Compute	(4\$)	(100\$)
Network	(0\$)	(30\$)
Total monthly cost	7.2\$	150\$

Cloud pricing - compare cloud and on-premise

- RAM, CPU or disk ca ~5x cheaper on-premise
- network (egress) ca ~1000x cheaper on-premise

... but:

- costs of personnel
- own HW must be used 24 month to get the above figures and in cloud billing can be eg for every 15 minutes, therefore server can be put down even for off-peak hours

Cloud pricing - practical tips&tricks

- Can be very huge!!!
- ... but some services are for free
 - Example: smart BigQuery data migration from US to EU
- Analyze in details what is charged
 - BigQuery - query price can significantly drop if only relevant fields are queried
 - BigQuery - “LIMIT” does not have any impact on pricing
- Concentrate on the most significant cost positions from your invoice

Why not cloud?

- Price
- Tools availability
- Level of control over infrastructure
- Infrastructure problems!

Summary

We have discussed:

- What is cloud computing
 - ...and in what models it can be delivered
 - ...and how about it's security
 - ...and common patterns used
 - ...and some real-life scenarios
 - ...and it's price models



Major points to remember

- Make argument-based decision about migrating to cloud
 - Difficult/expensive to rollback or migrate to other cloud
 - Think about scaling down
- Consider hybrid solutions
- More than one cloud is rather not recommended
 - Increased costs and difficulties
- Analyze price lists carefully
 - Sometimes there is indeed a free lunch



Literature

- [1] T. Erl, Z. Mahmood, R. Puttini, **Cloud Computing Concepts**, Technology & Architecture, 2013
- [2] **Google Cloud Platform in Action**, Manning, 2018
- [3] B. Wilder, **Cloud Architecture Patterns**, O'Reilly, 2012
- [4] S. R. Gonidawa, **Cloud Native Architecture and Design**, Apress, 2021

Questions

