⟨ PROFINIT ⟩

B0M33BDT - Cloud

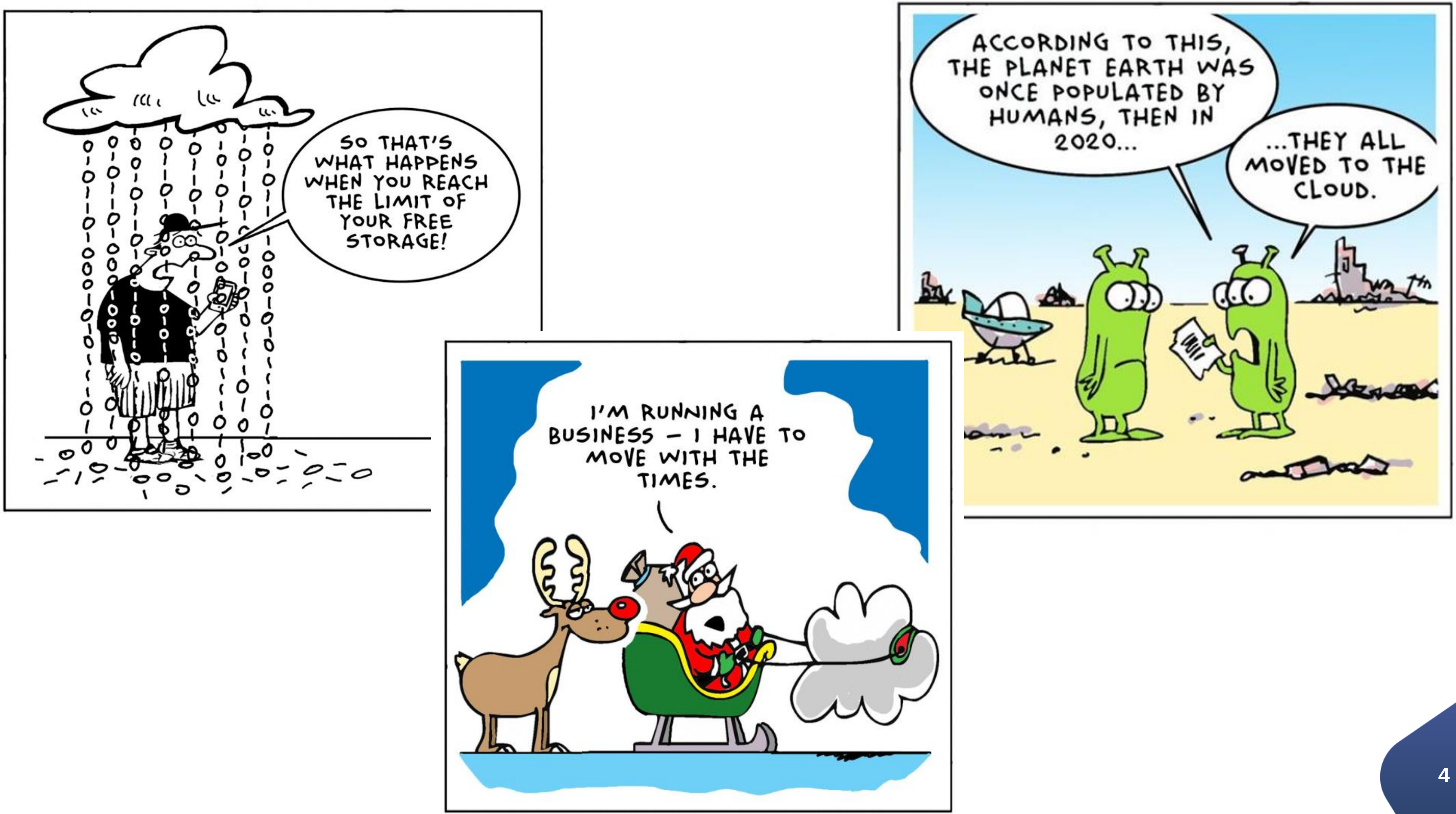
Petr Filas

22.11.2023

Agenda

- Cloud
- Cloud vendors
- Cloud use cases
- Cloud architectures
- Terraform
- Azure, AWS cost calculator

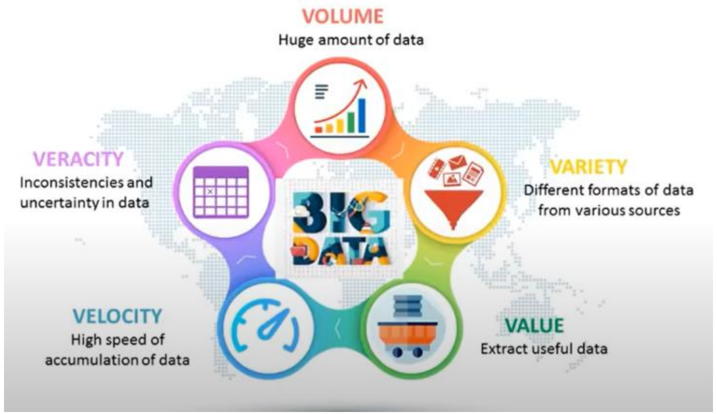
Cloud


SO THAT'S WHAT HAPPENS WHEN YOU REACH THE LIMIT OF YOUR FREE STORAGE!
I'M RUNNING A BUSINESS – I HAVE TO MOVE WITH THE TIMES.
ACCORDING TO THIS, THE PLANET EARTH WAS ONCE POPULATED BY HUMANS, THEN IN 2020...
...THEY ALL MOVED TO THE CLOUD.

Cloud

- What Cloud is?
  - Almost unlimited space in cloud
  - Collection of different servers, tools, services
  - Infrastructure orchestration etc.
  - “pay what you use” (GB, cores, security…)

- BigData and Cloud?
  - **Scalability of computing power and storage**
    - Scalability vs elasticity
  - Cost prediction
  - Click and Go solutions

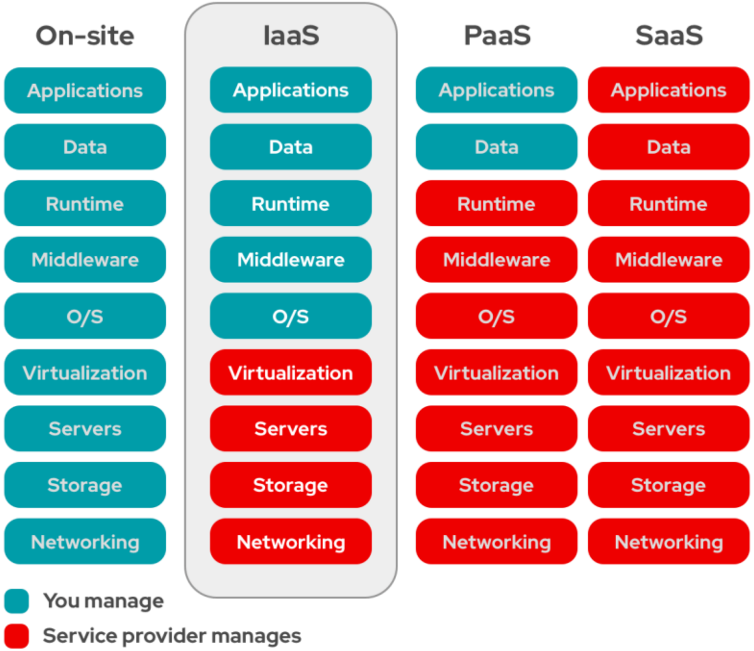

VOLUME
Huge amount of data
VERACITY
Inconsistencies and uncertainty in data
BIG DATA
VARIETY
Different formats of data from various sources
VELOCITY
High speed of accumulation of data
VALUE
Extract useful data

Cloud services

- Cloud service = service provided by a cloud provider via Internet.

- Cloud vendor manages services together with user
  - 3 different levels

    - **IaaS – Infrastructure as a Service**
    - **PaaS – Platform as a Service**
    - **SaaS – Software as a Service**



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage

🟥 Service provider manages

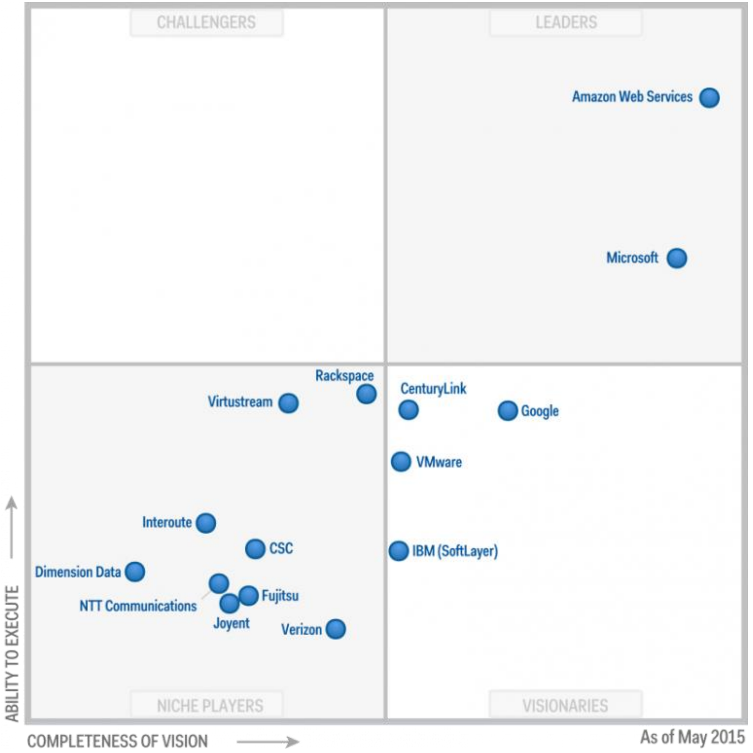
Cloud service vendors

- Amazon, Microsoft, Google, Alibaba = world-wide leaders
- **Amazon AWS**
  - Leader in cloud computing (first in 2008),
  - AI, Serverless deployments, IaaS
- **Microsoft Azure**
  - Leader in SaaS (MS platform)
  - Enterprise customers
- **Google GCP**
  - Google platform services
- **Alibaba Cloud**
  - Primary cloud in China

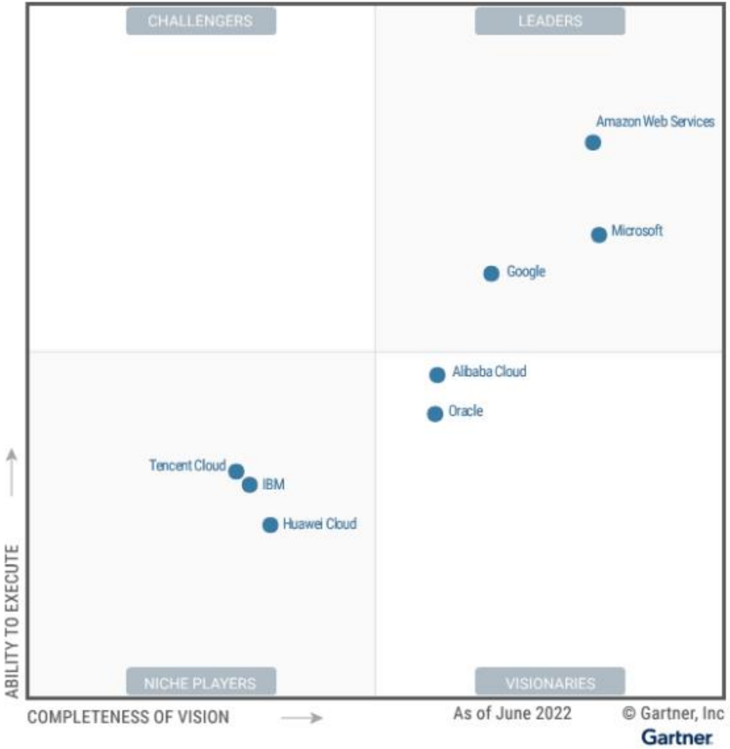
- Cloud has changed IT world (hardware, software, security, dataflow, infrastructure…)
- Almost every big player offers a cloud, cloud solution, cloud services to be hosted on cloud

Gartner – magic quadrant

❬ PROFINIT ❭



2015



2022

Public | Private | Hybrid Cloud

Cloud deployment options

PROFINIT

- A public cloud is where an independent third-party provider, owns and maintains compute resources that customers can access over the internet.

- A private cloud removes this sharing aspect of cloud computing, instead dedicating infrastructure and services to a single “user”.

- A hybrid cloud is a model in which a private cloud connects with public cloud infrastructure, enabling an organization to orchestrate workloads across the two environments.

Public clouds

- With a public cloud, all hardware, software, and other supporting infrastructure are owned and managed by the cloud provider
- In a public cloud, you share the same hardware, storage, and network devices with other organizations or cloud “tenants,” and you access services and manage your account using a web browser/API/CLI

Private clouds – how they are hosted and managed

- Virtual
  - A virtual private cloud is a private cloud that you can deploy within a public cloud infrastructure that enables an organization to run its workloads in logical isolation from every other user of the public cloud
  - Even though the server is shared by other organizations, the virtual logic ensures that a user's computing resources are private
- Hosted
  - The servers aren't shared with other organizations
  - The service provider configures the network, maintains the hardware and updates the software, but the server is occupied by a single organization
- Managed
  - Single-tenant environment fully managed by a third party. For example, the IT infrastructure for your organization could be purchased and maintained by a third-party organization in its data center. The third party provides maintenance, upgrades, support, and remote management of your private cloud resources.

Hybrid

- Software-only
  - Vendor provides the necessary software which runs on an organization's preexisting hardware
  - OpenStack

- Software and hardware
  - All-in-one bundle
  - It's a simple platform that exists on the user's premises and may or may not be provider-managed environments.
  - Azure stack

Cloud Use-cases

Use cases in real world

Prototyping (POC), Dev, Testing

- BD Architecture is defined when you need it and comply your project needs

- When you are not sure
  - What to use (sizing, platform)
  - Not ready to invest to hardware
  - If big data architecture is right for your project

Use cases in real world

Prototyping and Operating native services

- Usually cloud native services, such as Synapse, Redshift, Databricks, Snowflake etc.
  - Quick launch
  - You do not care about underlay infrastructure, licenses
  - Minimal administration
- Managed software (Kafka, Airflow …)
- This might not be most cost effective, if you’re provisioning resources too often for a long time.

Use cases in real world

- **Cloud Server instances with installed BD tools**
  - Long-term running, almost same as on-premise solution
  - Some ready-made images in cloud-shop
  - Development, Testing, Production
  - Easy to boost server instances, if you need
  - Long-term running might be very cost effective (long term plans)
  - You have to care about infrastructure, administration etc.

Use cases in real world

- **Serverless**
  - Serverless is a cloud computing application development and execution model that enables developers to build and run application code without provisioning or managing servers or backend infrastructure.
  - “Pure compute”
- **SaaS**
  - O365/Office.com, Gmail, DropBox
  - Netflix, Disney+, HBO GO
  - Is Databricks PaaS or Saas?

Cloud SLA (Service-level Agreement)

- **SLA** - an agreement between a cloud service provider and a customer that ensures a minimum level of service is maintained
  - **Availability** – usually at least 99,9% (but some services up to 99,99%)
  - **Manageability** – guarantee of managability and scalabilty
  - **Performance** – guarantee of consistent performance

How to save money in cloud?

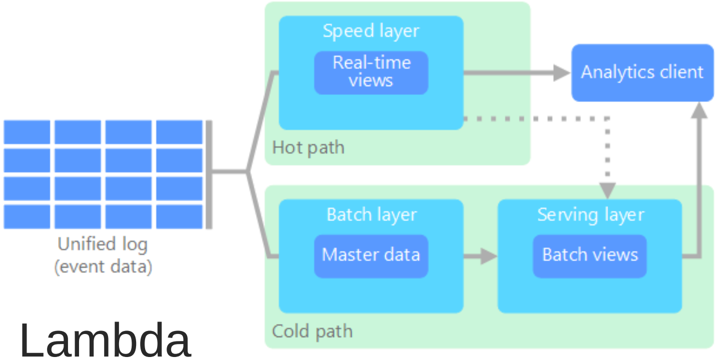
- **Question for million $$$**
- **The second biggest concern**
- **Turn off your instances, services**
  - Terraform is your friend
- **Size you solution properly**
  - Linear scaling?
- **Use reserved instances or spot instances**
- **Shared responsibility is a budget killer**

Cloud Architectures

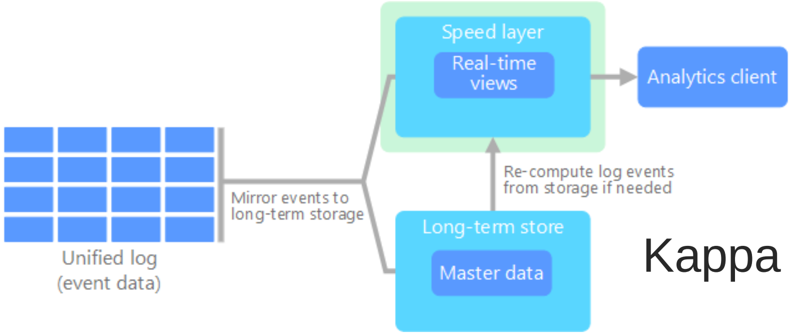
Big Data Architectures

- Cloud fits to BD architectures
  - Native components
  - Services
  - Hybrid solutions

- Lambda and Kappa (IoT)

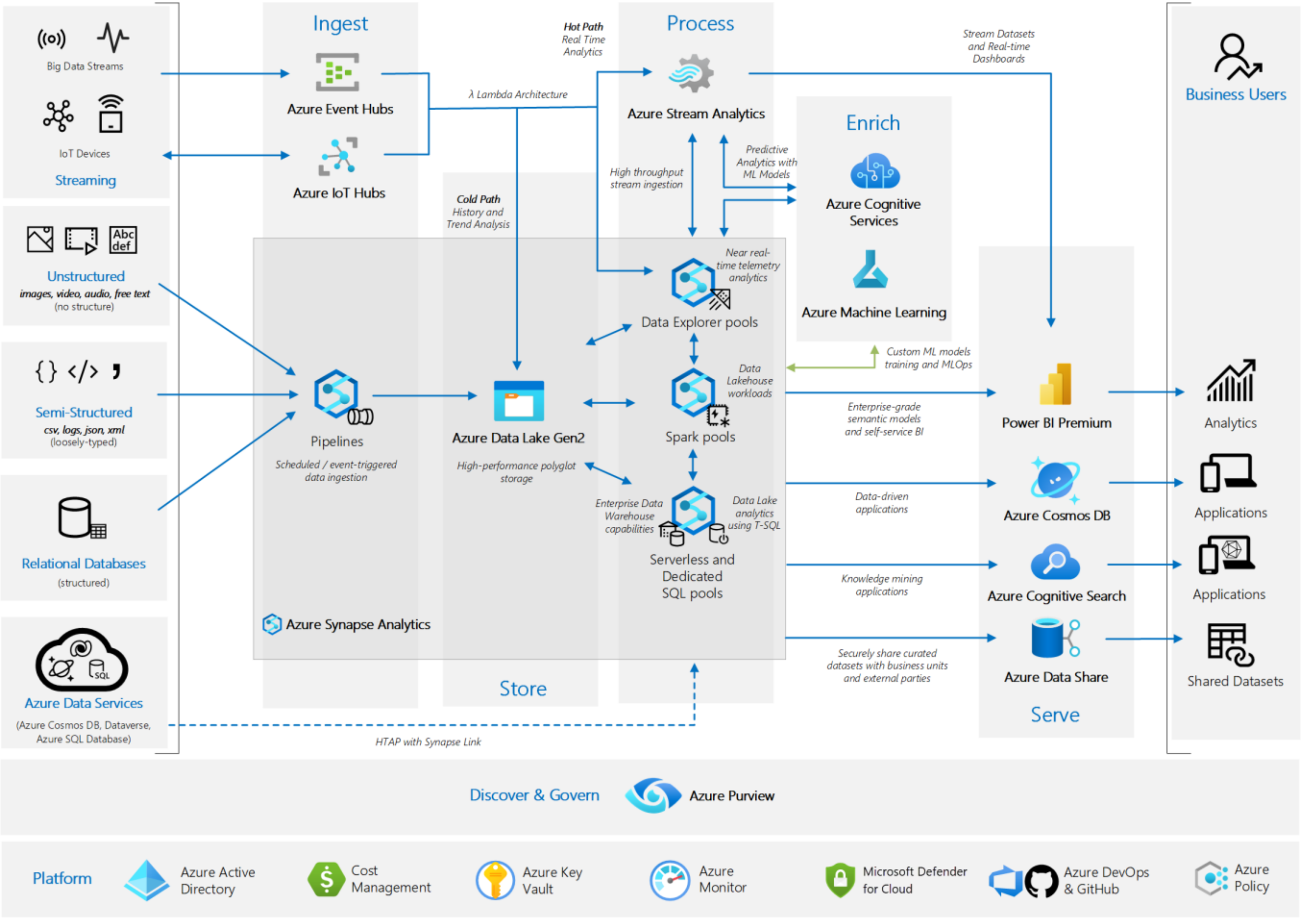


Lambda



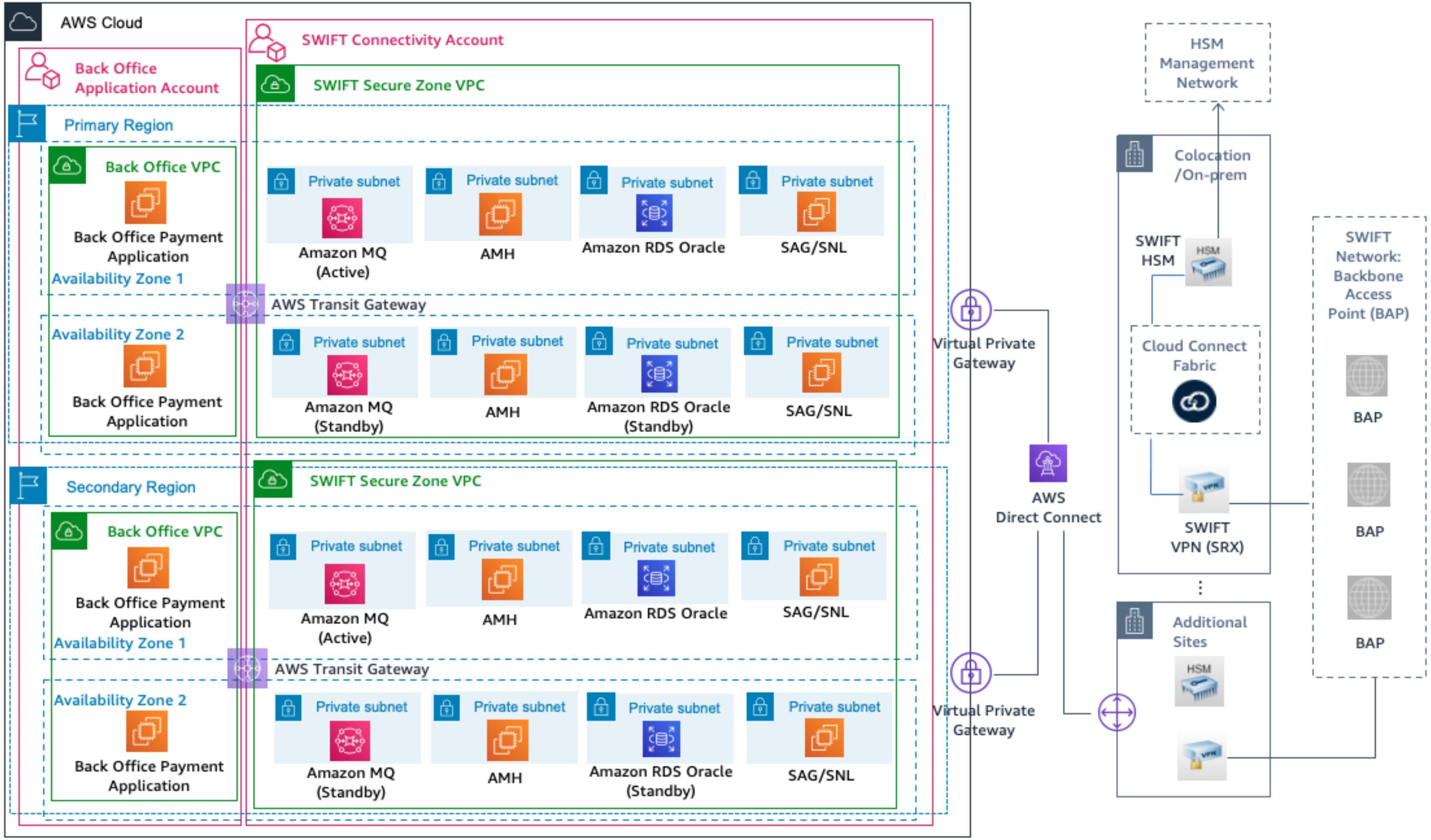
Kappa

Example of complex Azure BD analytic architecture



Examples of AWS architecture (reference)

PROFINIT



https://aws.amazon.com/blogs/architecture/architecting-swift-connectivity-on-amazon-web-services-aws/

Reference architectures

https://aws.amazon.com/blogs/architecture/
https://learn.microsoft.com/en-us/azure/architecture/browse/

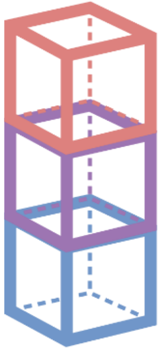
Terraform – infrastructure as a code (IaaC)

Terraform

- Open-Source automatization and management of your
  - (cloud) infrastructure
  - Your platform
  - Services
- Declarative language used – define WHAT result you want

Declarative

- Define what you have
  - Red, violet, blue cube
- Define what you want
  - A tower od red, violet, blue cubes



Procedural

- Define what you have
  - Red, violet, blue cube
- Define how to make what you want
  - Put blue cube
  - Put violet cube on blue cube
  - Put red cube on violet cube

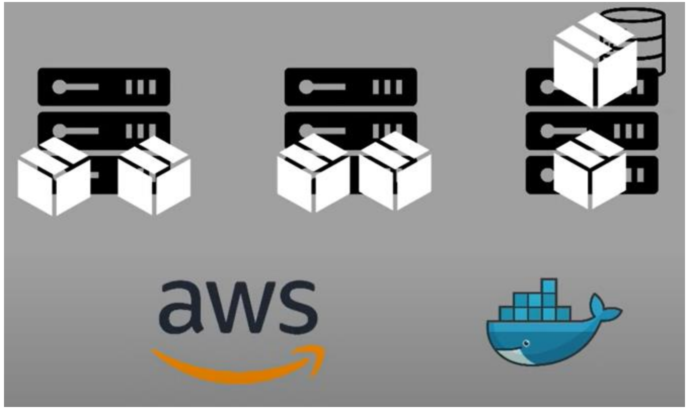
Ansible vs. Terraform

- Infrastructure as Code

Ansible	Terraform
Mainly a config tool (once infra is done)	Mainly infra provisioning
Deploy apps	Can deploy apps
Install/update software	
More mature	Relatively new tool
	Advanced in orchestration
Better for configuring infrastructure	**Better for provisioning infrastructure**

- DevOps usually use both tools

Terraform, case study



- 3 servers,
- Several microservices and database in Docker
- In usual, you must do steps:
  - Prepare private network
  - EC2 server instances
  - Install Docker, tools
  - Security, firewalls etc.
  - Deploy Docker containers.

1. PROVISIONING INFRASTRUCTURE

2. DEPLOY APPS

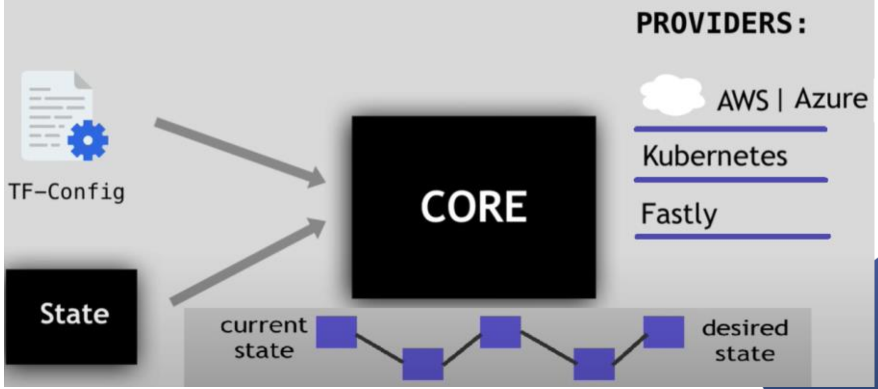
- 2 separate teams, usually
- Adding new servers, security setup, replicating from dev to prod…

⇒ AUTOMATION VIA TERRAFORM

Terraform, how it works

PROFINIT

- **Core** takes **input** and plans what needs to be created, updated, destroyed… from current **state** = execution plan
- Steps are executed with platform specific tools
- **Providers** (100 providers)
  - AWS, Azure (IaaS)
  - Kubernetes (Paas)
  - Fastly (SaaS)
- Each provider offers resources you can work with



Terraform - steps

PROFINIT

- **Get the latest version of terraform**
- **terraform init**
  - Initialize the environment
- **terraform refresh**
  - Refresh the state
- **terraform plan**
  - To see what will happen
  - Save the plan and apply it otherwise you are not sure what will be executed
- **terraform apply**
  - Do the job!

Terraform – steps - example

```
-/+ resource "aws_ecs_task_definition" "transformer" {

      ~ arn                           = "arn:aws:ecs:eu-west-1:70332xxxxxx2:task-
definition/test-transformer-uat:12" -> (known after apply)

      ~ container_definitions         = (sensitive) # forces replacement

      ~ id                            = “test-transformer-uat" -> (known after apply)

      - ipc_mode                      = "" -> null

      - pid_mode                      = "" -> null

      ~ revision                      = 12 -> (known after apply)

      - tags                          = {} -> null

        # (9 unchanged attributes hidden)

    }

Plan: 3 to add, 12 to change, 3 to destroy.

Saved the plan to: plan.tfplan

To perform exactly these actions, run the following command to apply:

    terraform apply "plan.tfplan"
```

Terraform – steps - example

```
Apply complete! Resources: 3 added, 12 changed, 3 destroyed.


Outputs:


common_container_sg = "sg-0ae9ceXXXXXXXXXda"

db_endpoint = “test-uat.ccxxxxxxwugh.eu-west-1.rds.amazonaws.com"
```

Terraform – state file

PROFINIT

- Terraform stores information about your infrastructure in a state file
- This state file keeps track of resources created by your configuration and maps them to real-world resources
- Extremely important!

- **Can be stored on**
  - Localhost
  - S3 / ADLS and another shared cloud storage
  - HashiCorp paid solution - consul

Terraform – sections

- Resource
  - Managed by terraform (has state)
- Module
  - Larger independent part
- Output
  - Output of the terraform script
- Provider
  - “Connector”
- Variable
  - Variable

Terraform – modules

- Terraform modules encapsulate groups of resources dedicated to one task, reducing the amount of code you have to develop for similar infrastructure components.
  - Structure your code
  - Create reusable parts
  - Work with variables
- Spin up compute cluster

Terraform – architecture

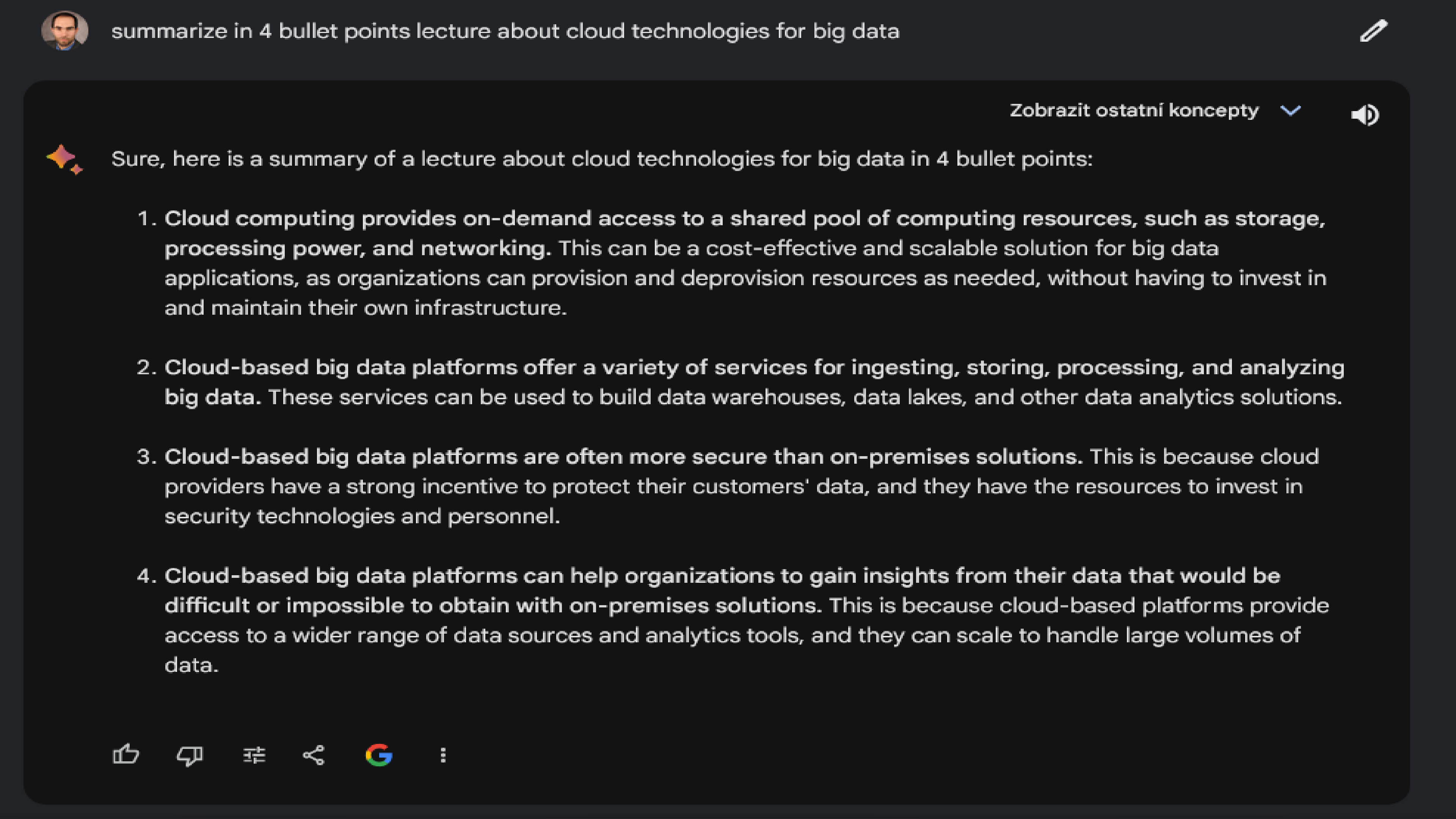
- Typically executed from GitHub actions or Jenkins
- First of all you compile your artifacts and store them somewhere
- Then run terraform to create infrastructure and use artifacts from the storage
- You can destroy your dev environment every day and have it off during the weekends ☺ (to be sure that you can build it from the scratch)

Cloud Costs aka FinOps

Cost Calculator

- Check how much you will pay what for
  - Azure Pricing Calculator
  - AWS Pricing Calculator
- Usually multi-criteria optimization task (time vs money)

Summary



summarize in 4 bullet points lecture about cloud technologies for big data

Sure, here is a summary of a lecture about cloud technologies for big data in 4 bullet points:

1. **Cloud computing provides on-demand access to a shared pool of computing resources, such as storage, processing power, and networking.** This can be a cost-effective and scalable solution for big data applications, as organizations can provision and deprovision resources as needed, without having to invest in and maintain their own infrastructure.

2. **Cloud-based big data platforms offer a variety of services for ingesting, storing, processing, and analyzing big data.** These services can be used to build data warehouses, data lakes, and other data analytics solutions.

3. **Cloud-based big data platforms are often more secure than on-premises solutions.** This is because cloud providers have a strong incentive to protect their customers' data, and they have the resources to invest in security technologies and personnel.

4. **Cloud-based big data platforms can help organizations to gain insights from their data that would be difficult or impossible to obtain with on-premises solutions.** This is because cloud-based platforms provide access to a wider range of data sources and analytics tools, and they can scale to handle large volumes of data.

〈 PROFINIT 〉

Questions?

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6

Tel.: + 420 224 316 016, web: www.profinit.eu

LINKEDIN
linkedin.com/company/profinit

TWITTER
@profinit_EU

FACEBOOK
facebook.com/Profinit.EU

YOUTUBE
Profinit EU, s.r.o.