

# Big Data w chmurze

# Radostław Szmit

✉ radoslawszmit@gmail.com

🐦 @RadoslawSzmit

in rszmit



# Big Data Passion

🌐 <http://bigdatapassion.pl/>  
github.com/bigdatapassionpl

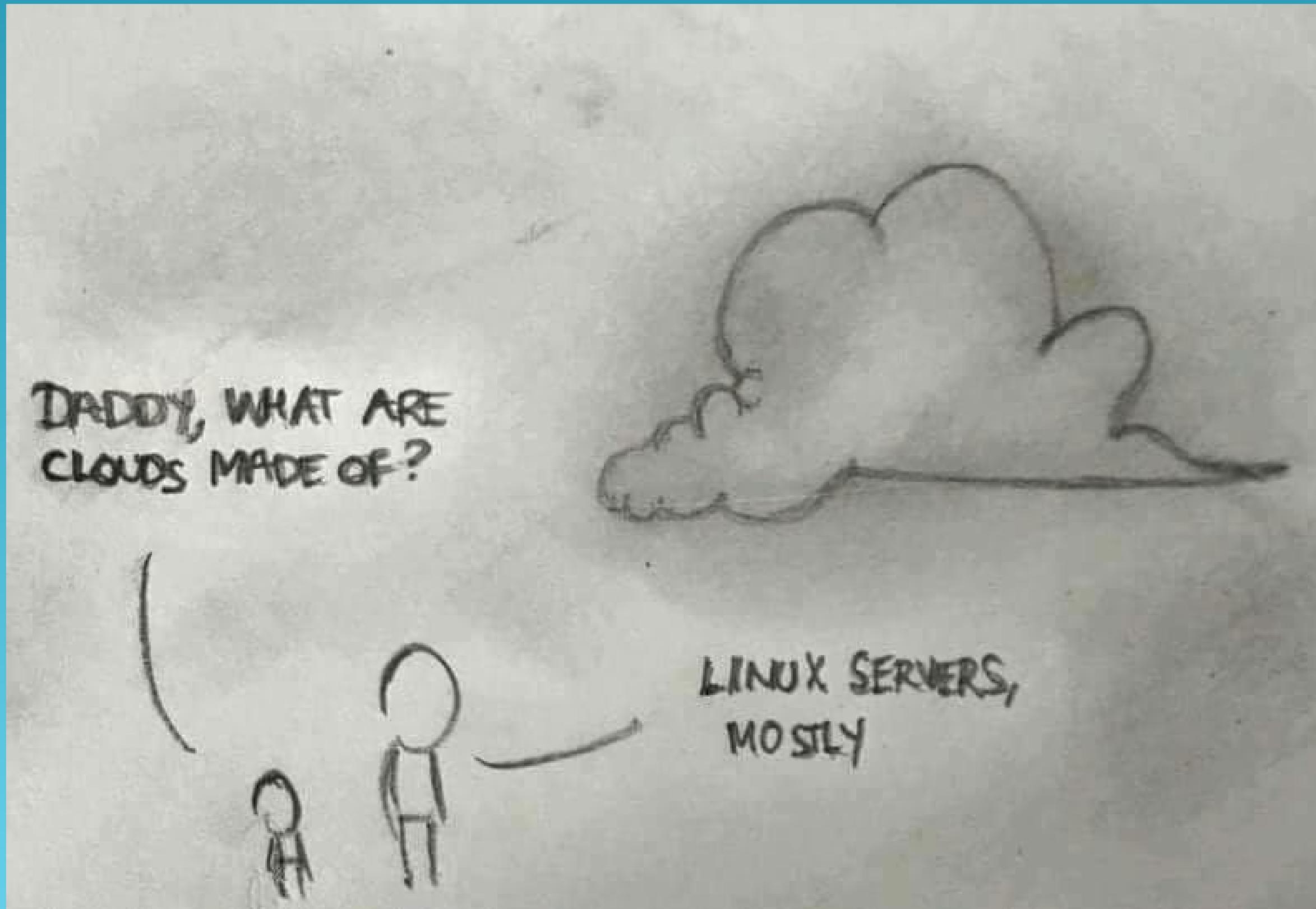


# Cel zajęć

- Fakty a nie marketing
- Technologie a nie sprzedaż
- Brak wojny DC vs Cloud :)
- Poznanie rozwiązań chmurowych na przykładzie AWS
- Porównanie z innymi rozwiązaniami chmury publicznej

# Chmury obliczeniowe

# Czym jest chmura obliczeniowa?



# Czym jest chmura obliczeniowa?

Chmura obliczeniowa to dostarczanie usług obliczeniowych — serwerów, magazynu, baz danych, sieci, oprogramowania, analiz itd. — za pośrednictwem Internetu („chmura”).

Jest ona oferowana przez zewnętrzne podmioty (dostawca chmury), **dostępna na żądanie** w dowolnej chwili oraz **skalująca** się w miarę zapotrzebowania.

Dostawcy zazwyczaj pobierają **opłaty** za usługi chmury obliczeniowej **w zależności od użycia** (moc obliczeniowa, przestrzeń dyskowa, transfer danych)

# Czym jest chmura obliczeniowa?



<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

# Czym jest chmura obliczeniowa?

- Cechy
  - Dostęp na żądanie
  - Dostęp z internetu
  - Wspólne zasoby z innymi klientami
  - Elastyczność
  - Płatność za użycie
- Model
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)

# Dlaczego interesować się chmurą?

- Szybkość wdrażania nowości (np. nie czekamy na rozstrzygnięcie przetargu)
- Dostosowujemy koszta do aktualnych potrzeb (pay as you use)
- Elastyczność
- Dynamiczna moc obliczeniowa
- Łatwość zmian naszych systemów

# Dlaczego interesować się chmurą?

- Skupiamy się na naszym biznesie a nie biurokracji
- Innowacyjność (dużo nowoczesnych usług np. Big Data lub AI)
- Większa wygoda
- Mniejsze koszta ludzkie
- Brak inwestycji, idealne środowisko na POC lub startup

# Dlaczego interesować się chmurą?

- Największy zysk z chmury mamy w projektach Big Data!
- 1oh działania 10 maszyn to koszt 1h działania 100 maszyn
- Ten sam wynik otrzymujemy znacznie szybciej za te same pieniądze!

# Przykład firmy Novartis

- Analiza danych medycznych (badania nad rakiem)
- Na jednym serwerze = 40 lat
- Potrzeba 50 tys rdzeni = koszt 40 milionów dolarów
- Przetworzenie w chmurze to gh, moc 87 tys rdzeni, koszt 4232 dolarów



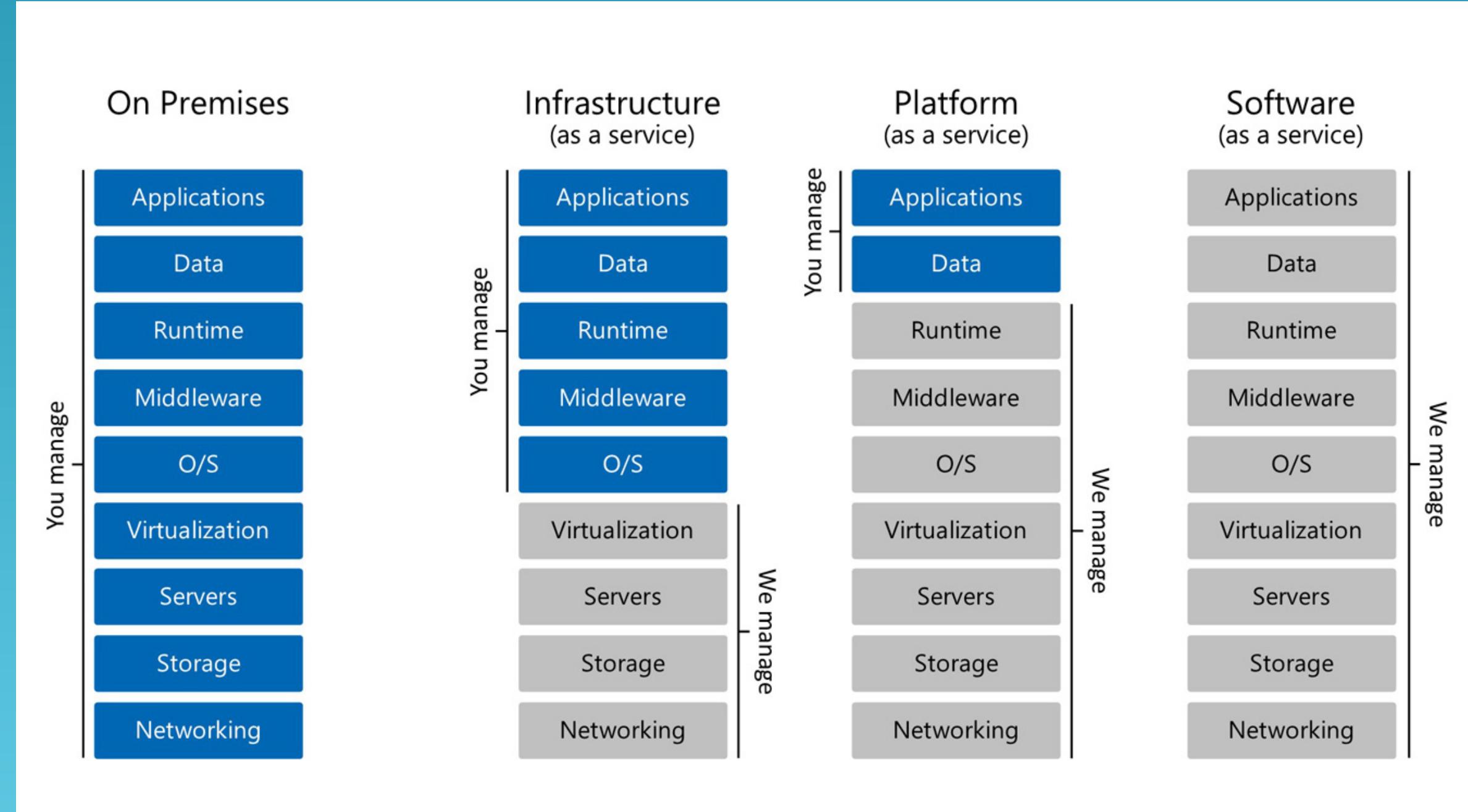
# Rodzaje chmur obliczeniowych

- Prywatne
  - Zbiór serwerów będących własnością naszej organizacji, choć niekoniecznie znajdujących się fizycznie w przestrzeni naszej organizacji
- Publiczne
  - Korzystamy z zasobów zewnętrznego dostawcy takiego jak Amazon, Google, Microsoft, Oktawave, etc.
- Hybrydowe
  - Łączą ze sobą obydwa modele gdzie np. dane “wrażliwe” mogą być trzymane w prywatnej chmurze zaś pozostałe w publicznej (istnieje wiele modeli chmur hybrydowych)

# Modele chmury obliczeniowej

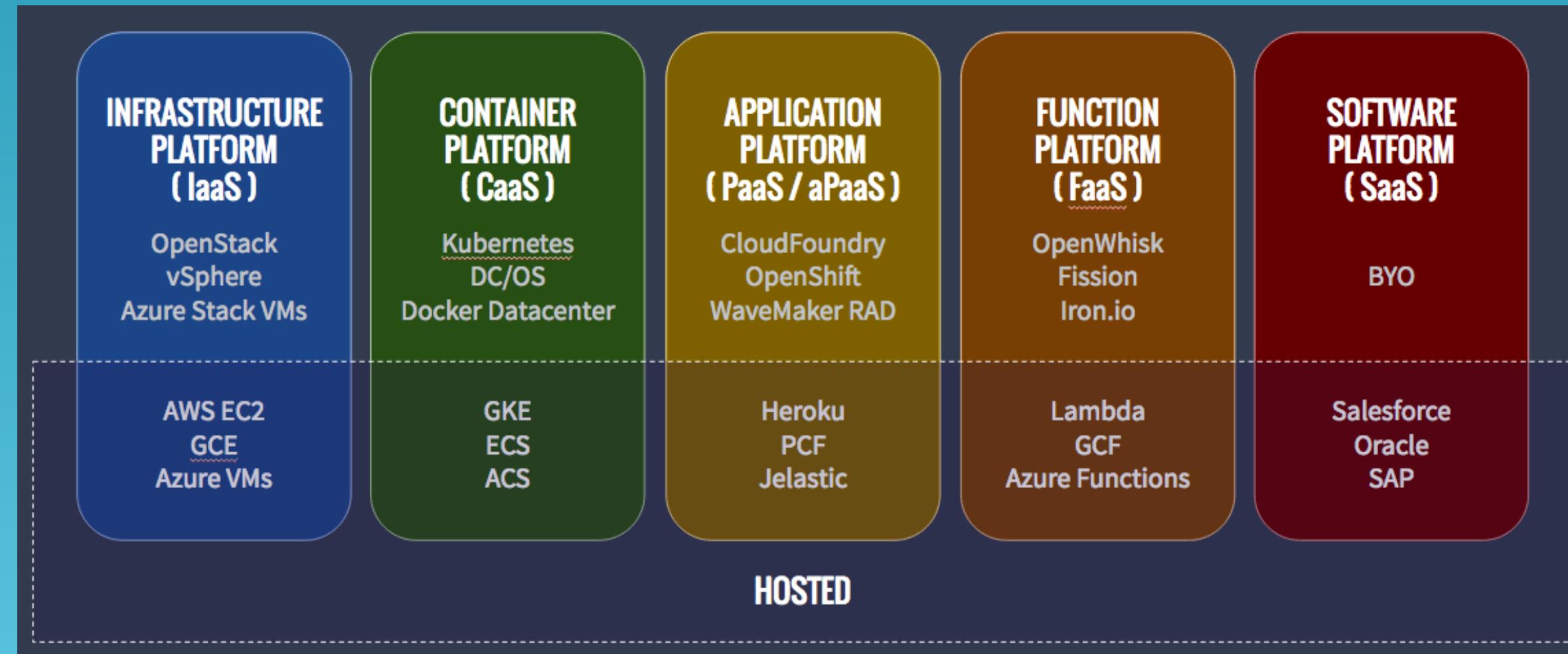
- IaaS (ang. Infrastructure as a Service)
  - Klient otrzymuje infrastrukturę informatyczną w oparciu o którą może instalować i rozwijać własne oprogramowanie
- PaaS (ang. Platform as a Service)
  - Klient otrzymuje platformę informatyczną w oparciu o którą może rozwijać własne systemy informatyczne
- FaaS (ang. Function as a Service)
  - Serverless computing, umożliwia klientowi uruchamianie swoich aplikacji, tak zwanych funkcji, bez troszczenia się o warstwę sprzętową odpowiedzialną za ich wykonanie
- SaaS (ang. Software as a Service)
  - Klient otrzymuje zestaw funkcjonalności bądź narzędzi w formie gotowego oprogramowania typu system CRM online

# Modele chmury obliczeniowej



<https://medium.com/@annilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-acf6022d36e>

# Modele chmury obliczeniowej



<https://medium.com/@annilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e>

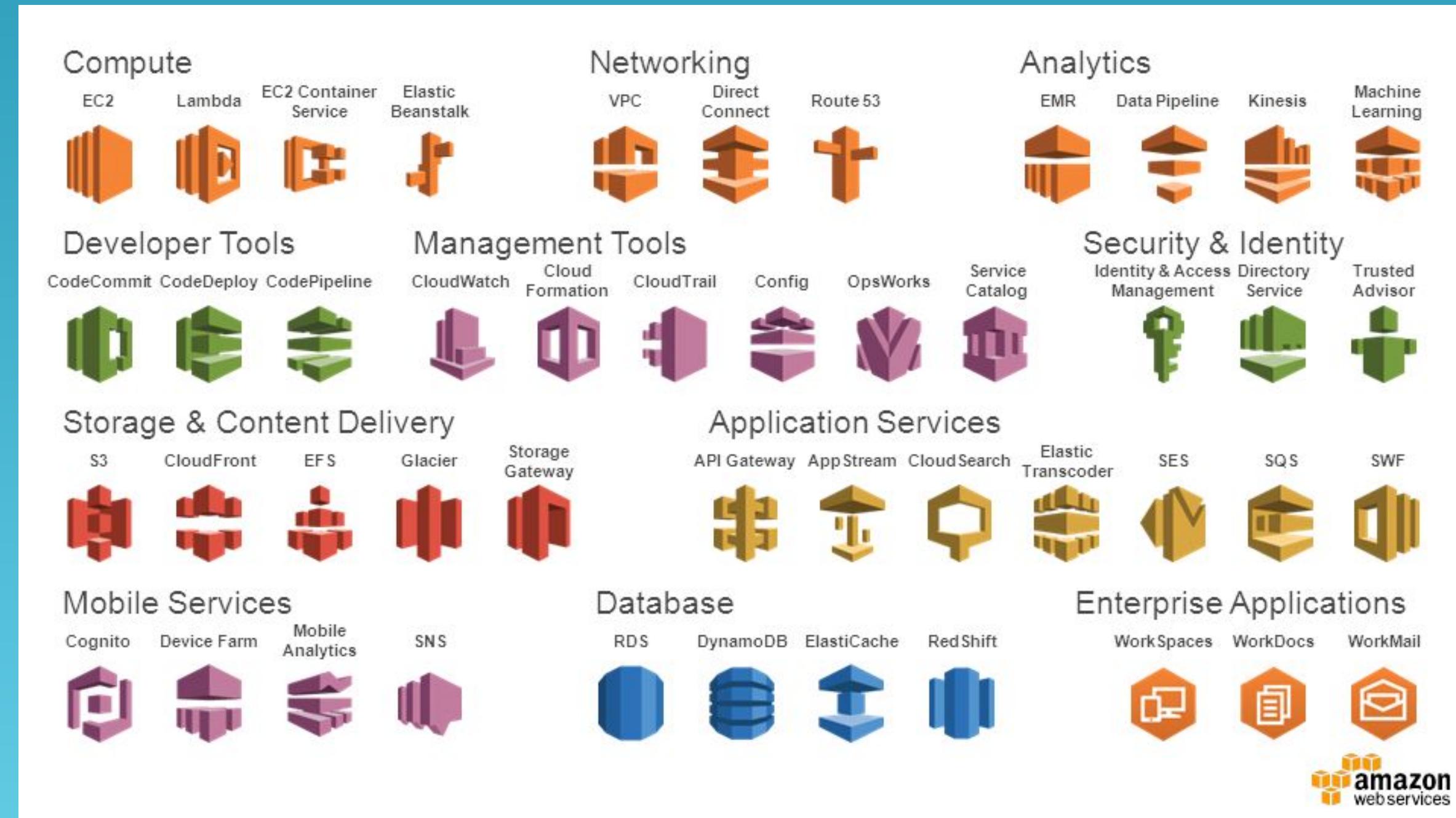
# Kilka możliwych dróg wdrożenia Big Data

- Korzystamy z chmury prywatnej i dostępnych narzędzi open source takich jak Hadoop, Spark, Hive i inne (samodzielne lub dystrybucja Big Data)
- Korzystamy z chmury publicznej, ale opartej o narzędzia open source dostarczane przez chmurę
- Korzystamy z chmury publicznej i dostępnych tam narzędzi jak BigQuery, BigTable, S3 czy DynamoDB
- Model hybrydowy łączący wszystkie powyższe w wybranym zakresie (wiele modeli / kombinacji chmur hybrydowych)

# Chmura to “building blocks”



# Wiele narzędzi



# Wiele narzędzi



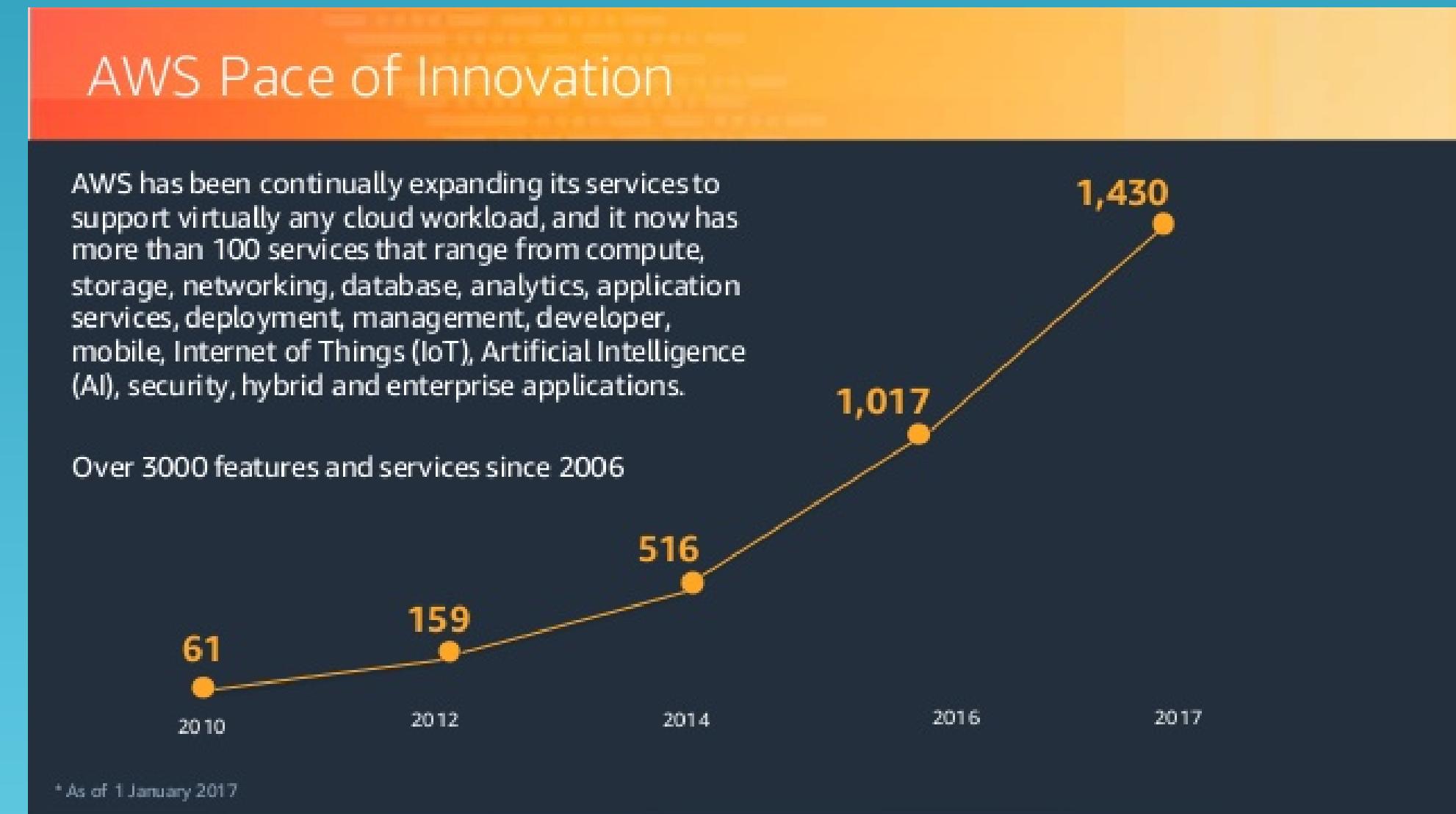
<https://www.youtube.com/watch?v=RNrsllweCno>

# Wiele narzędzi

Which Stream/Message Storage Should I Use?						
	Amazon DynamoDB Streams	Amazon Kinesis Streams	Amazon Kinesis Firehose	Apache Kafka	Amazon SQS (Standard)	Amazon SQS (FIFO)
AWS managed	Yes	Yes	Yes	No	Yes	Yes
Guaranteed ordering	Yes	Yes	No	Yes	No	Yes
Delivery (deduping)	Exactly-once	At-least-once	At-least-once	At-least-once	At-least-once	Exactly-once
Data retention period	24 hours	7 days	N/A	Configurable	14 days	14 days
Availability	3 AZ	3 AZ	3 AZ	Configurable	3 AZ	3 AZ
Scale / throughput	No limit / ~ table IOPS	No limit / ~ shards	No limit / automatic	No limit / ~ nodes	No limits / automatic	300 TPS / queue
Parallel consumption	Yes	Yes	No	Yes	No	No
Stream MapReduce	Yes	Yes	N/A	Yes	N/A	N/A
Row/object size	400 KB	1 MB	Destination row/object size	Configurable	256 KB	256 KB
Cost	Higher (table cost)	Low	Low	Low (+admin)	Low-medium	Low-medium

<https://www.youtube.com/watch?v=RNrsllweCno>

# Wiele narzędzi



# Modele chmury hybrydowej

- Trzymamy **część danych** w chmurze publicznej a część w prywatnej - względy bezpieczeństwa danych lub ograniczenia dyskowe chmury prywatnej
- Cloud **bursting** - korzystamy z chmury gdy potrzebujemy na jakiś czas większej mocy obliczeniowej niż może nam zapewnić nasza obecna infrastruktura chmury prywatnej
- **Multi region** - mamy chmurę prywatną w Polsce która nie zapewnia odpowiedniego czasu dostępu dla klientów z Azji lub USA, chmura staje się kopią naszych usług dla innych obszarów geograficznych
- Dynamiczna alokacja środowiska **testowego** - w chmurze testujemy i rozwijamy środowiska testowe korzystając z elastyczności i modelu płatności za użycie

# Modele chmury hybrydowej

- Przenosimy **część systemów / aplikacji** do chmury - korzystamy z automatycznej skalowalności czy narzędzi dostępnych tylko w chmurze
- **Nowe** systemy są w chmurze publicznej zaś stare zostają w środowisku w którym już są (integracja zamiast migracji, “data gravity”)
- Rozwiązanie **zapasowe**, HA, SLA, load balancing, disaster recovery - chmura publiczna zapewnia nam bezpieczeństwo danych lub usług

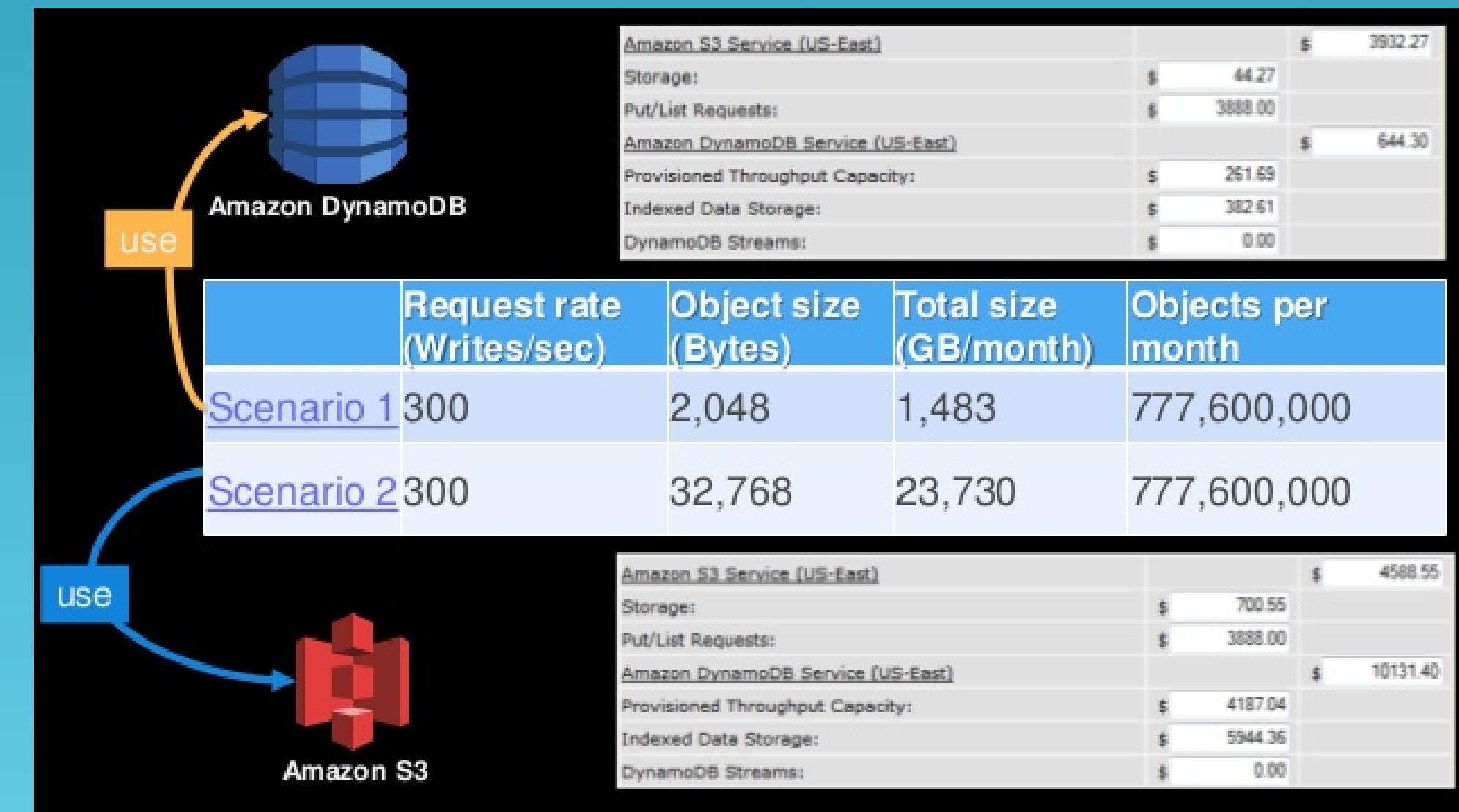
# Koszt chmury

- Ze względu na elastyczność względem użycia, niezwykle trudne bywa określenie kosztu korzystania z chmur obliczeniowych
- W zależności od zastosowań i użycia może różnić się także cena tych samych rozwiązań
- Trudno także porównać ceny pomiędzy dostawcami, w celu wybrania najkorzystniejszego cenowo rozwiązania
- Płacimy za
  - Przestrzeń dyskową
  - Dostęp do przestrzeni dyskowej
  - Moc obliczeniową
  - Transfer danych (sieć)
  - Inne

# Koszt chmury

- Koszt widoczny
  - Cena maszyny
  - Cena licencji
  - Użycie oprogramowania (saas)
- Koszt ukryty
  - Utrzymanie infrastruktury
  - Prąd
  - Pracownicy
  - Serwerownia
  - Zapasowe zasilanie i internet
  - Rozwój narzędzi
  - Bezpieczeństwo
  - Wycena ryzyka (odpowiedzialność)
  - inne...

# Cena rozwiązań opartych o chmury



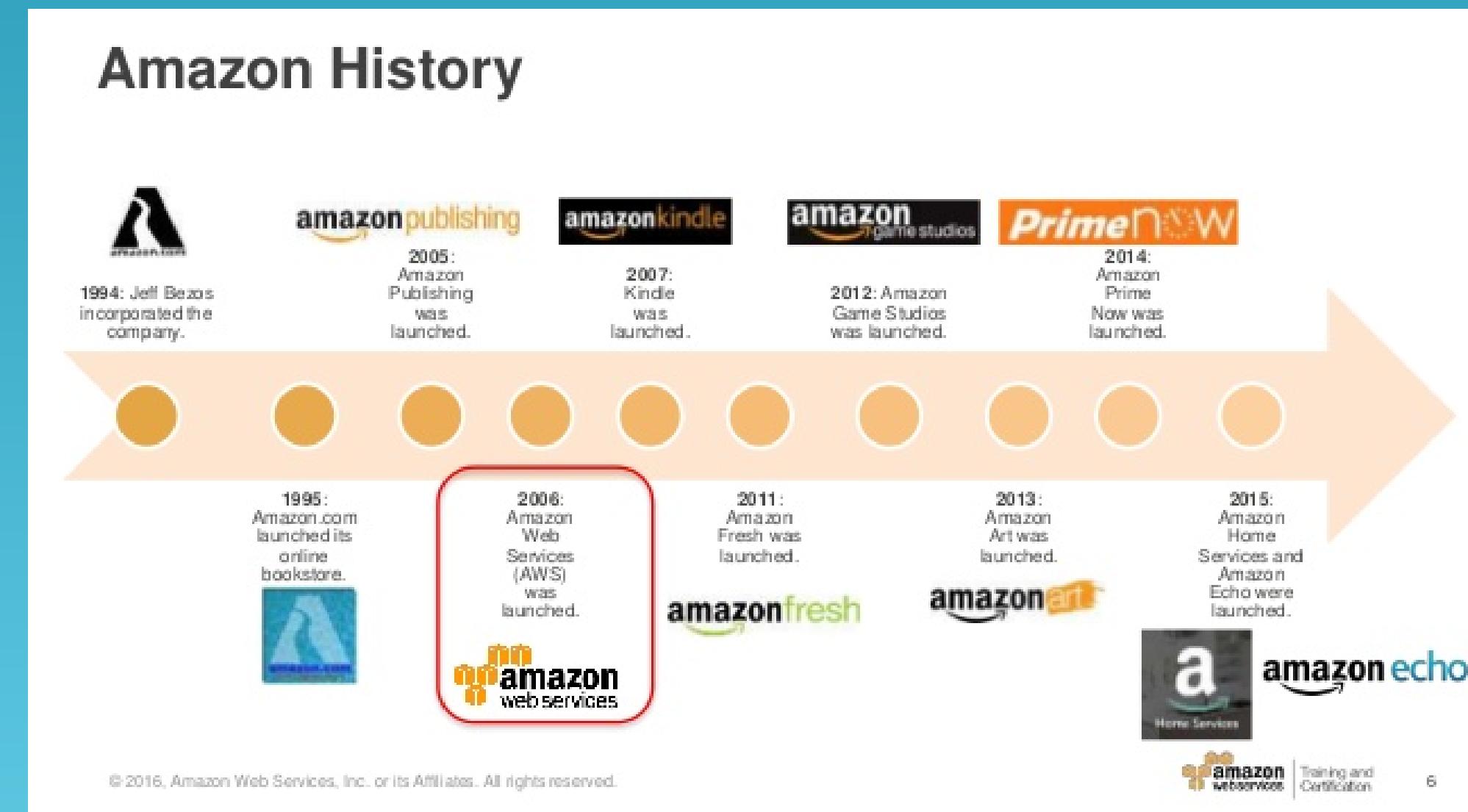
# Amazon Web Services



# Amazon Web Services

- Stworzony przez firmę Amazon (część firmy)
- Jedno z pierwszych rozwiązań tego typu na świecie (od marca 2006 roku)
- Wedle wielu statystyk AWS jest liderem w świadczeniu usługi chmur obliczeniowych (największa liczba klientów i wdrożeń)
- Dostarcza wiele własnych autorskich rozwiązań, także w świecie Big Data
- Uważana jest także za jedno z tańszych rozwiązań

# Historia AWS



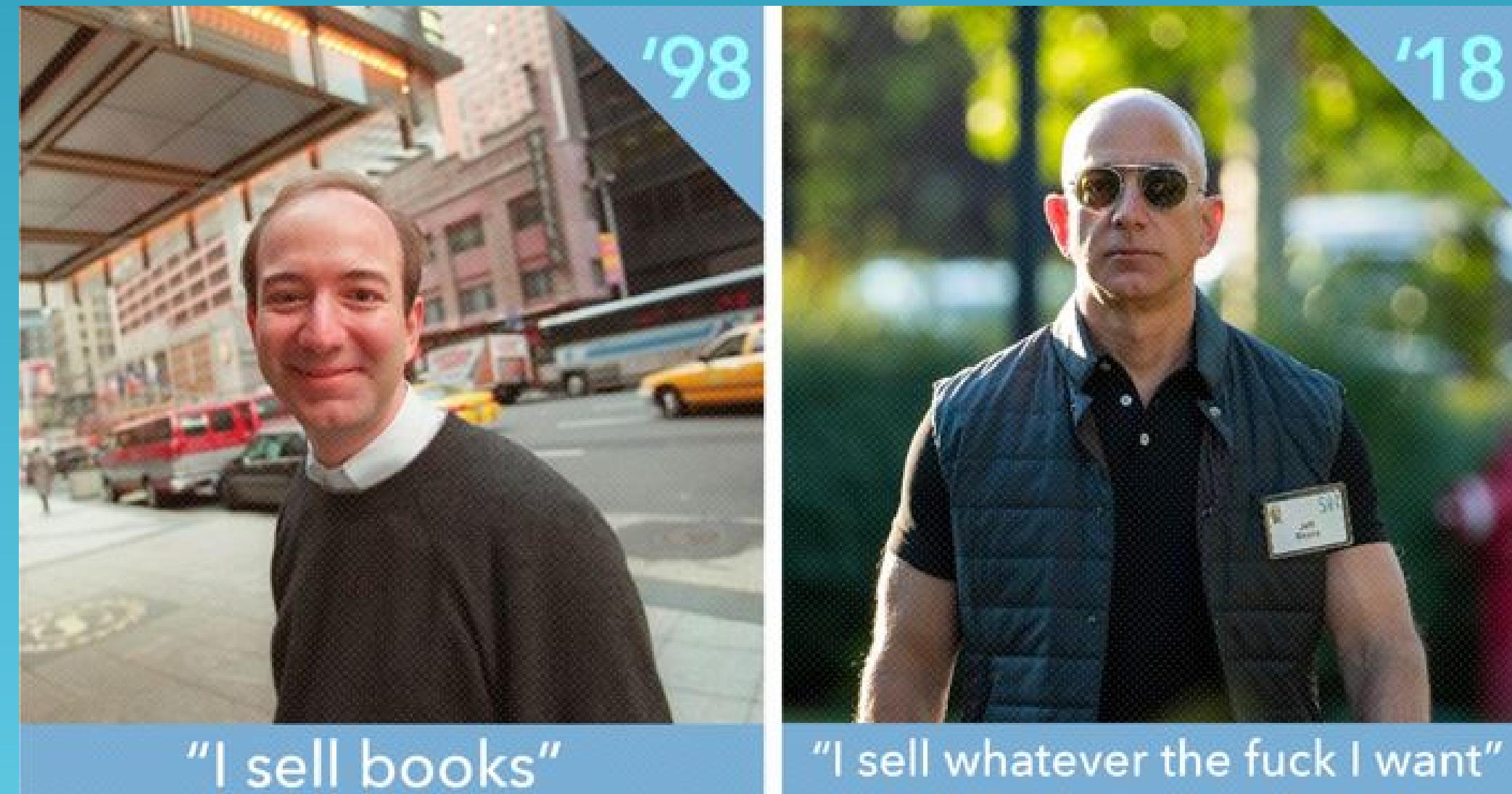
<http://jeff-barr.com/2014/08/19/my-first-12-years-at-amazon-dot-com/>

# The world's most valuable brands 2019

1. **Amazon** \$315.5 billion
2. Apple \$309.5 billion
3. **Google** \$309 billion
4. **Microsoft** \$251.2 billion
5. Visa \$177.9 billion
6. Facebook \$159 billion
7. **Alibaba** \$131.2 billion
8. **Tencent** \$130.9 billion
9. McDonald's \$130.4 billion
10. AT&T \$108.4 billion

<https://www.cnbc.com/2019/06/11/amazon-beats-apple-and-google-to-become-the-worlds-most-valuable-brand.html>

# Jeff Bezos



# Jeff Bezos

“

We've had three big ideas at Amazon that we've stuck with for 18 years, and they're the reason we're successful: Put the customer first. Invent. And be patient.

The most important thing: Focus obsessively on the customer.”

Jeff Bezos



# Klienci



# Klienci Enterprise



# Klienci Polska



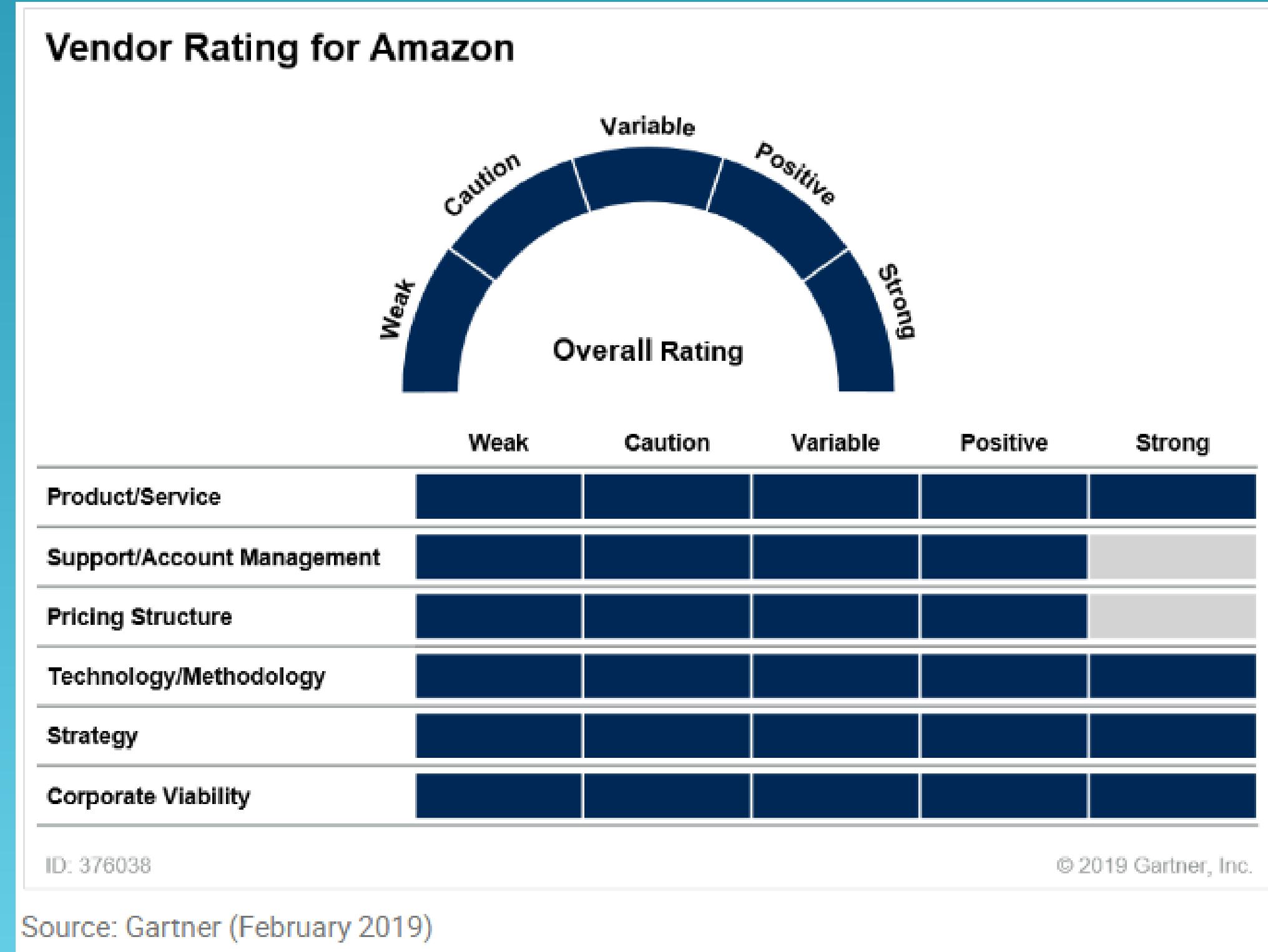
# Gartner Cloud 2018



# Gartner Cloud 2019



# Gartner Vendor Rating

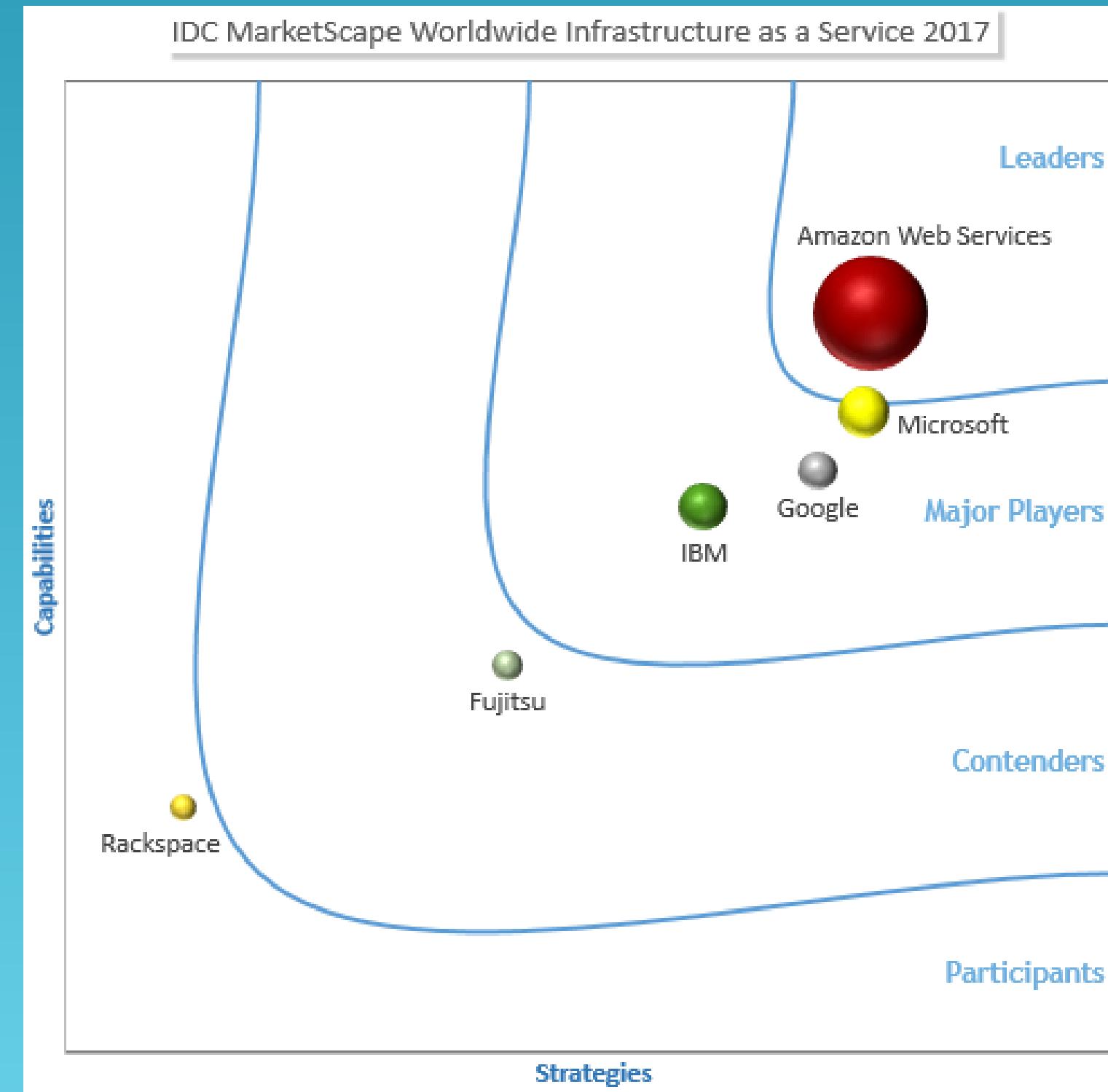


# The Forrester Wave™: Cloud Hadoop/Spark Platforms, Q1 2019

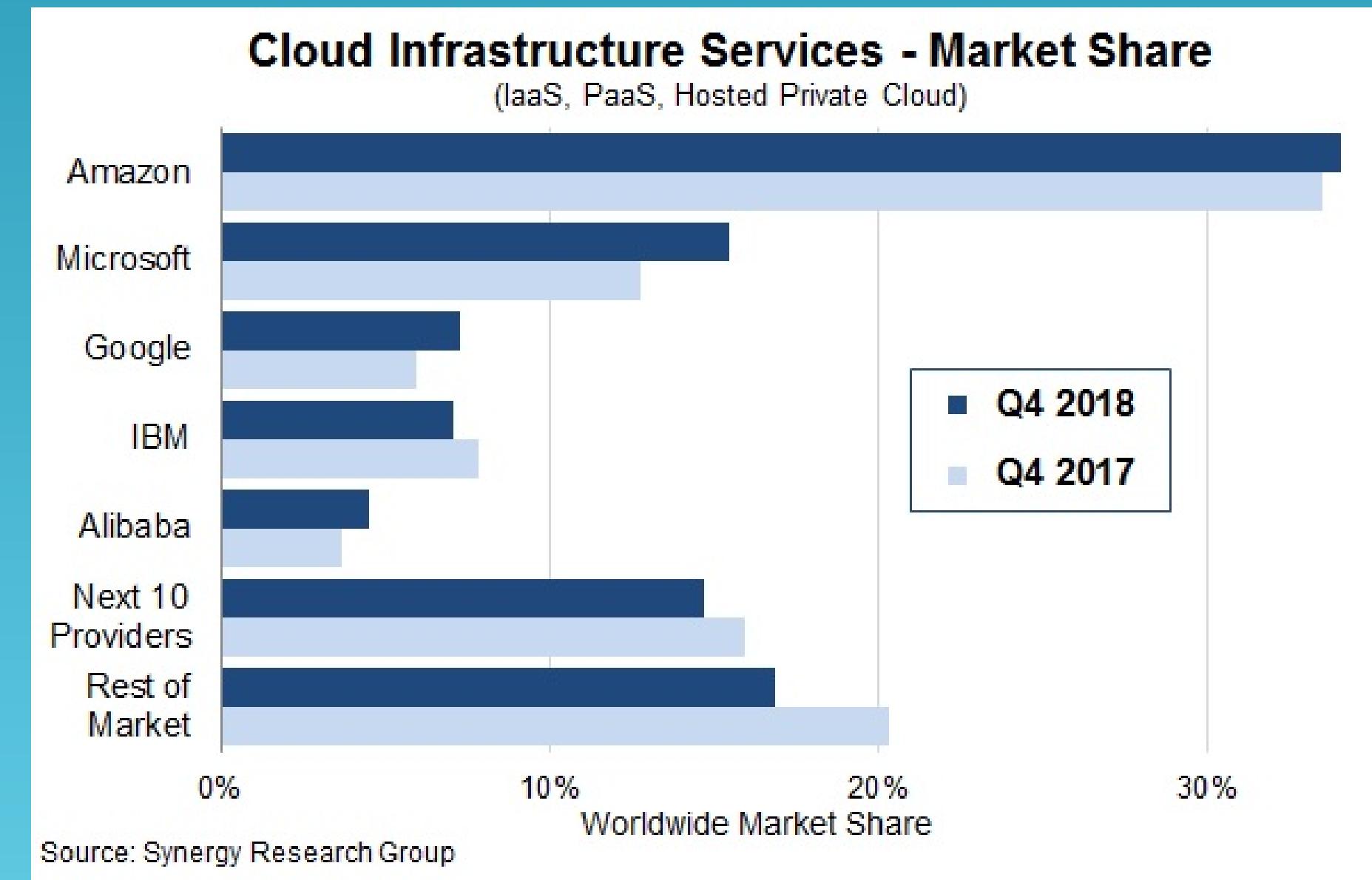


<https://reprints.forrester.com/#/assets/2/374/RES142663/reports>

# IDC MarketScape: Worldwide Infrastructure as a Service 2017 Vendor Assessment



# Fourth Quarter Growth in Cloud Services Tops off a Banner Year for Cloud Providers



<https://www.srgresearch.com/articles/fourth-quarter-growth-cloud-services-tops-banner-year-cloud-providers>

# Bardzo niebezpieczne dane!!!

## CLOUD WARS Q2 Revenue Projections

Top 10 Vendors	Q2 Rev Estimates	Growth Drivers
1. Microsoft	\$10.0B	Brilliant services + go-to-market
2. Amazon	\$8.0B	M&A to strengthen PaaS/SaaS?
3. Salesforce	\$3.74B (in Q ended 4/30)	Benioff bets all on 'Customer 360'
4. SAP	\$1.85B	Booming in 'Experience Economy'
5. IBM	\$5.0B	Must boost cloud growth rate
6. Oracle	\$1.8B	New growth via MSFT alliance?
7. Google Cloud	\$1.5B	Kurian revitalizes; huge potential
8. Workday	\$825M (in Q ended 4/30)	33% growth: 'Plan, execute, analyze'
9. Accenture	\$2.4B	Should top \$10B in calendar 2019
10. ServiceNow	\$825M	Superb position and execution

@bobevansIT

<https://cloudwars.co/top-vendors-35-billion-q2-cloud-revenue/>

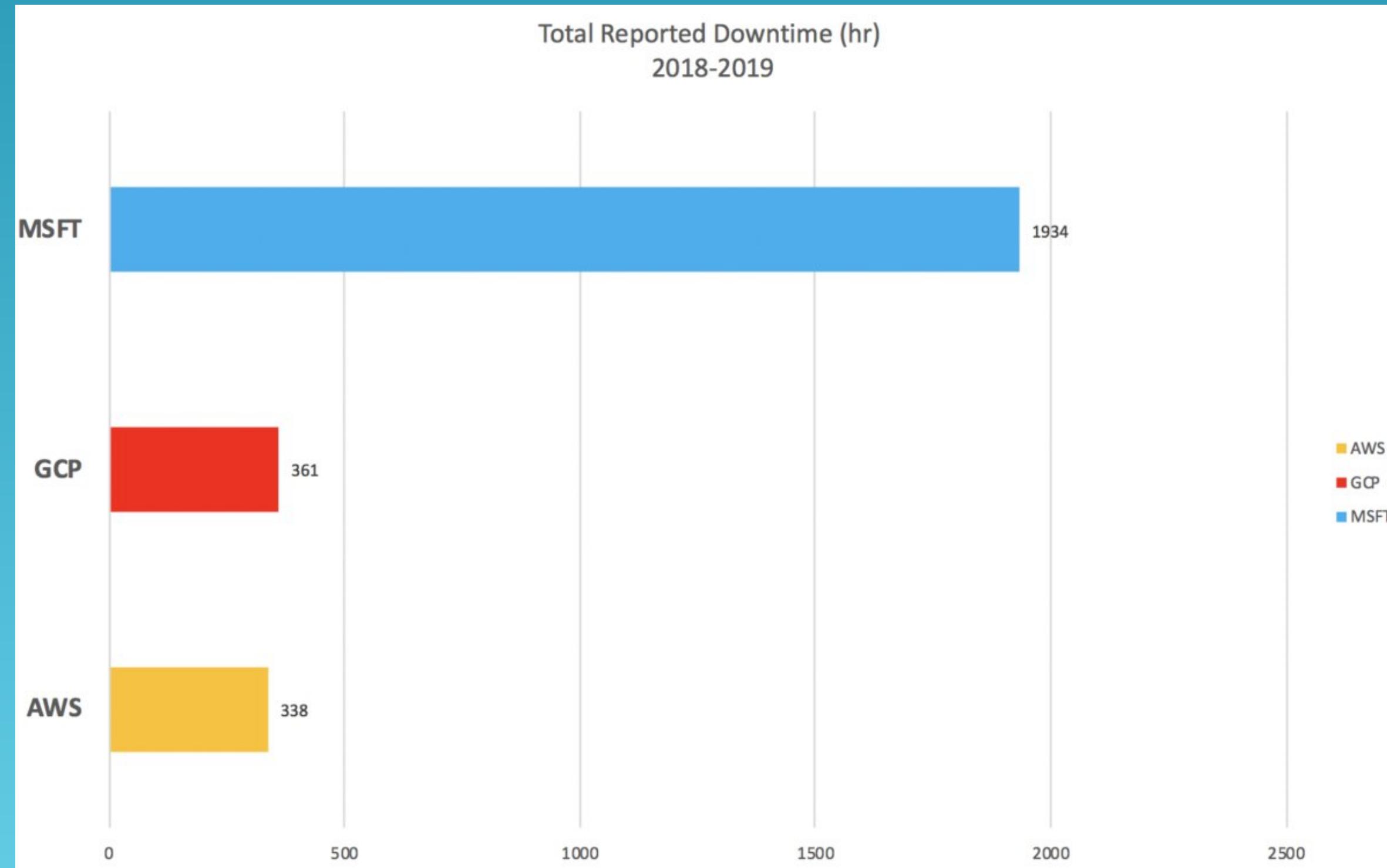
# Lepsze dane :)

"Intelligent Cloud Our Intelligent Cloud segment consists of our public, private, and hybrid server products and cloud services that can power modern business. This segment primarily comprises:

- Server products and cloud services, including SQL Server, Windows Server, Visual Studio, System Center, and related CALs, and Azure.
- Enterprise Services, including Premier Support Services and Microsoft Consulting Services."

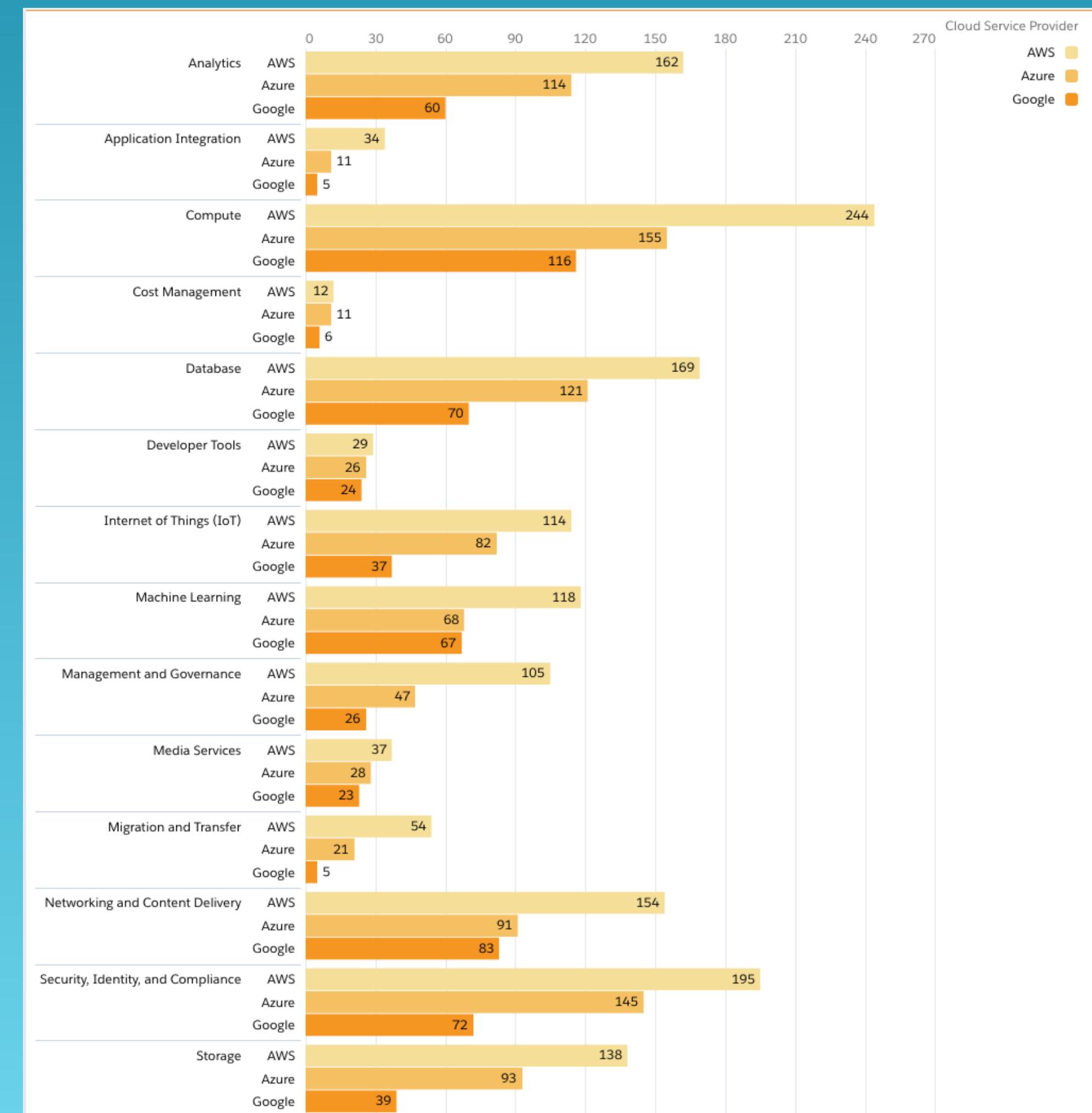
<https://www.microsoft.com/en-us/annualreports/ar2018/annualreport>

# Dość niebezpieczne dane...



<https://www.networkworld.com/article/3394341/when-it-comes-to-upptime-not-all-cloud-providers-are-created-equal.html>

# AWS funkcjonalności



# Regiony

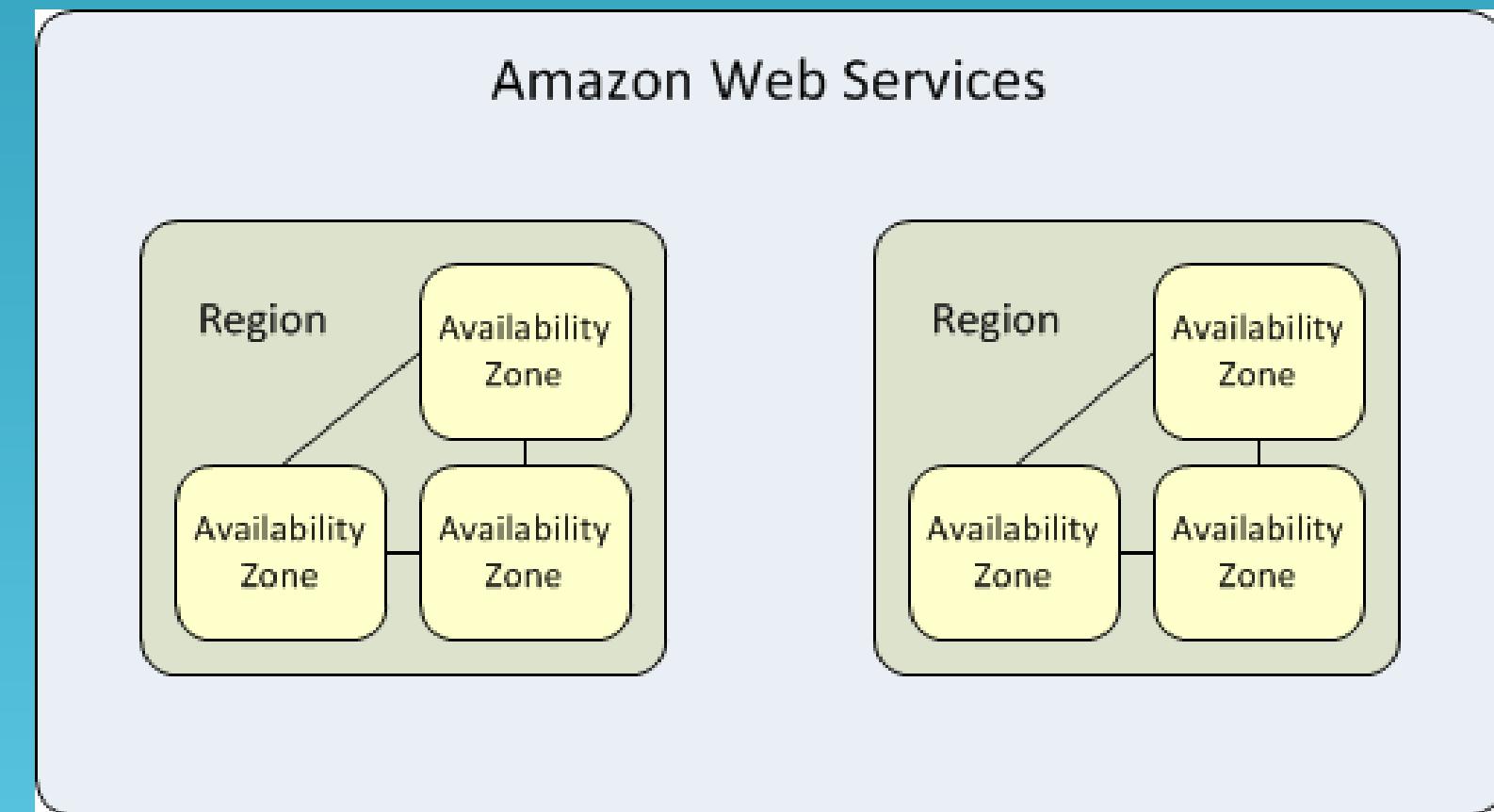


<https://aws.amazon.com/about-aws/global-infrastructure/>

# Regiony

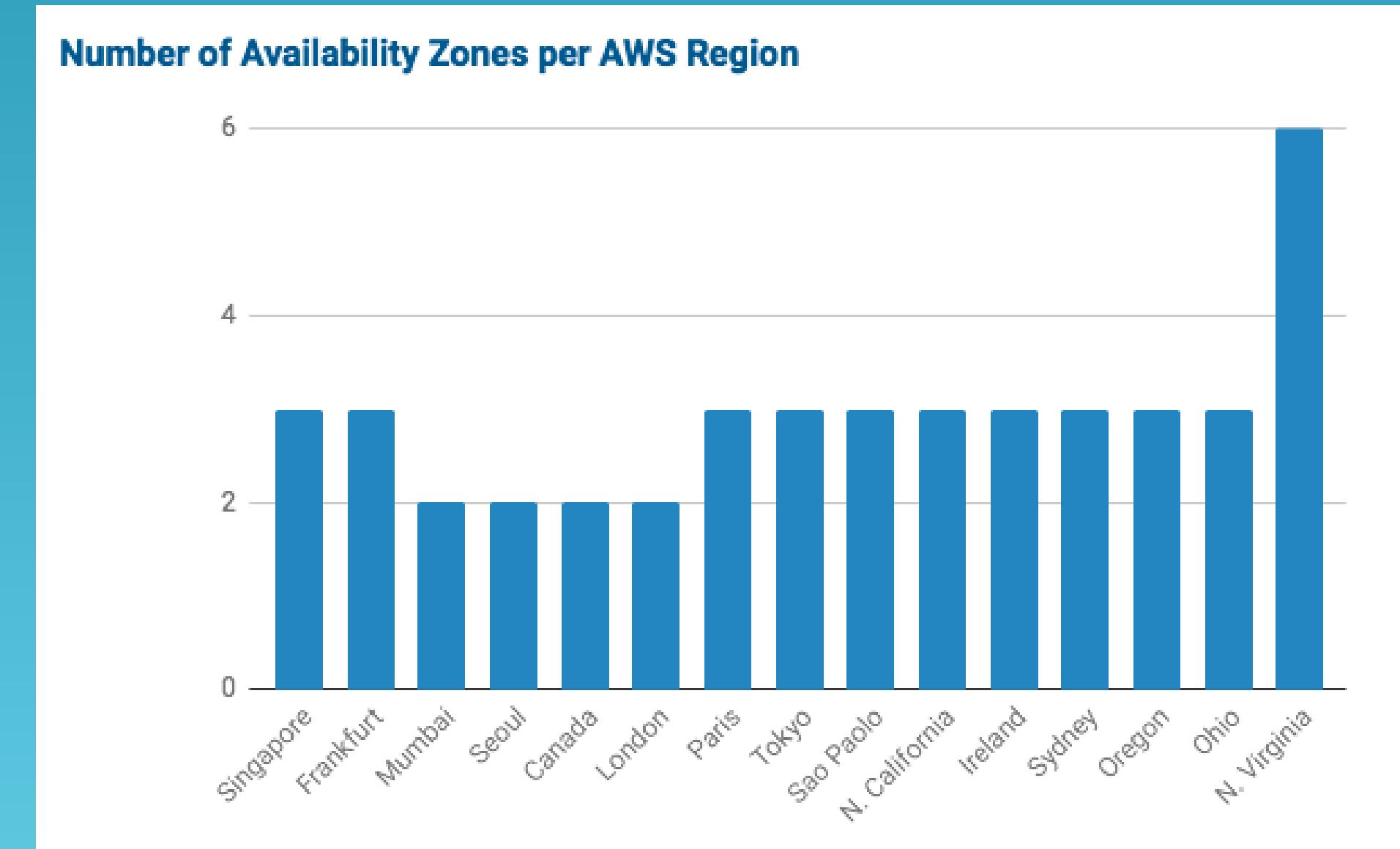
- 17 publicznych regionów: US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), EU (Frankfurt), EU (Ireland), EU (London), EU (Paris), EU (Stockholm), South America (São Paulo)
- Dwa niepubliczny region: GovCloud
- Dwa regiony specjalne w chinach wymagające specjalnego zgłoszenia: Beijing, Ningxia

# Regions & Availability Zones



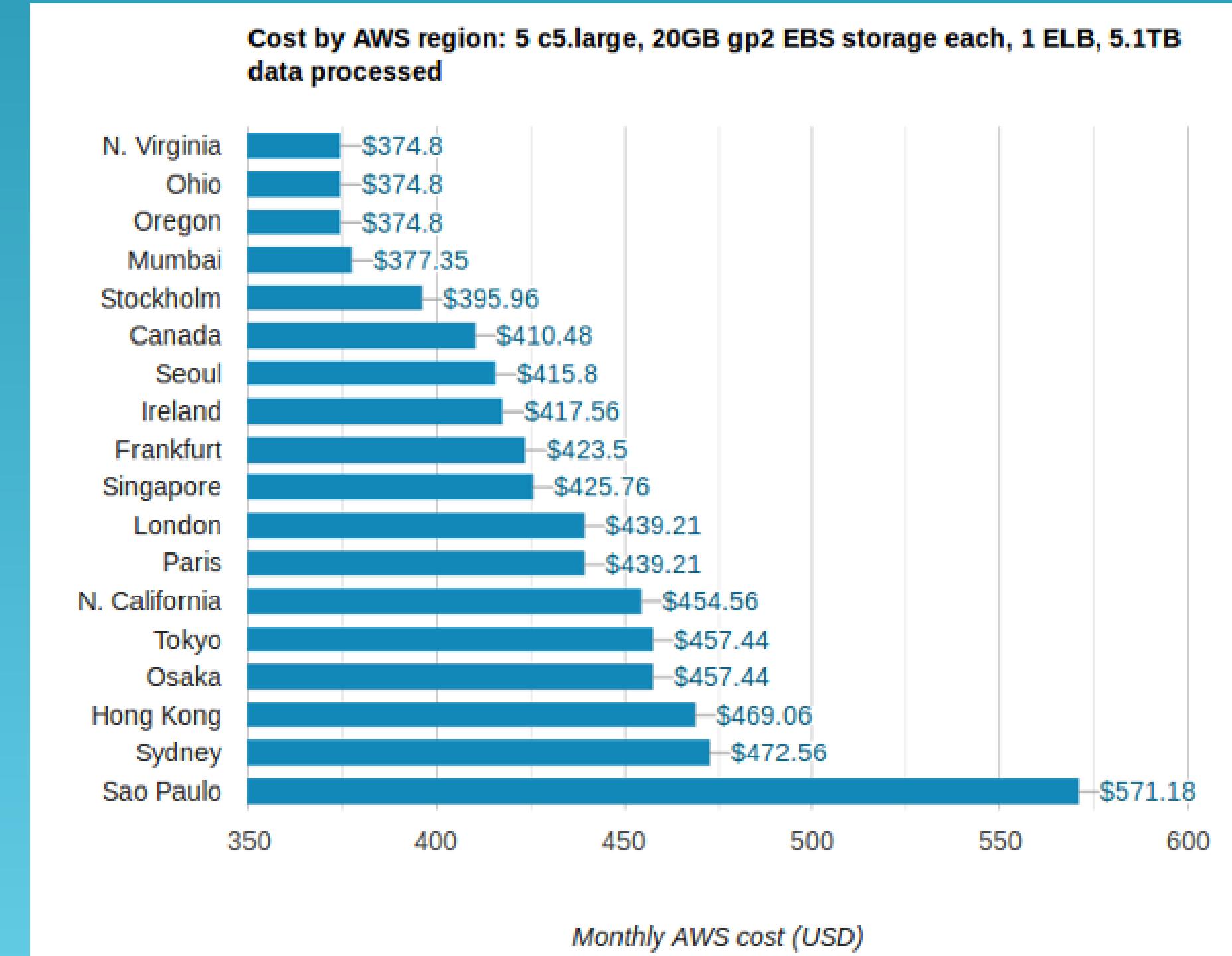
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

# Regions & Availability Zones



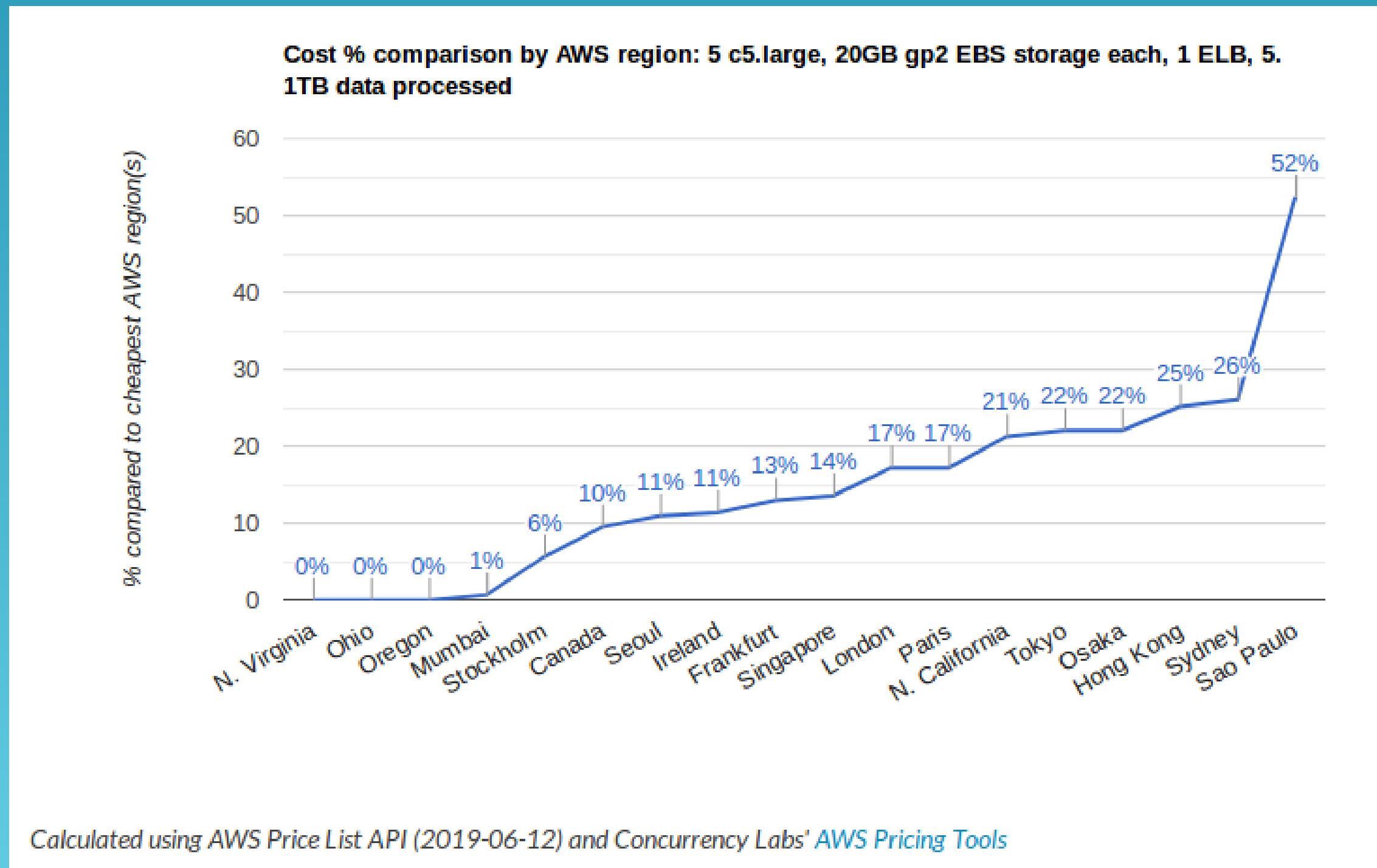
<https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/>

# Regions & Availability Zones



<https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/>

# Regions & Availability Zones



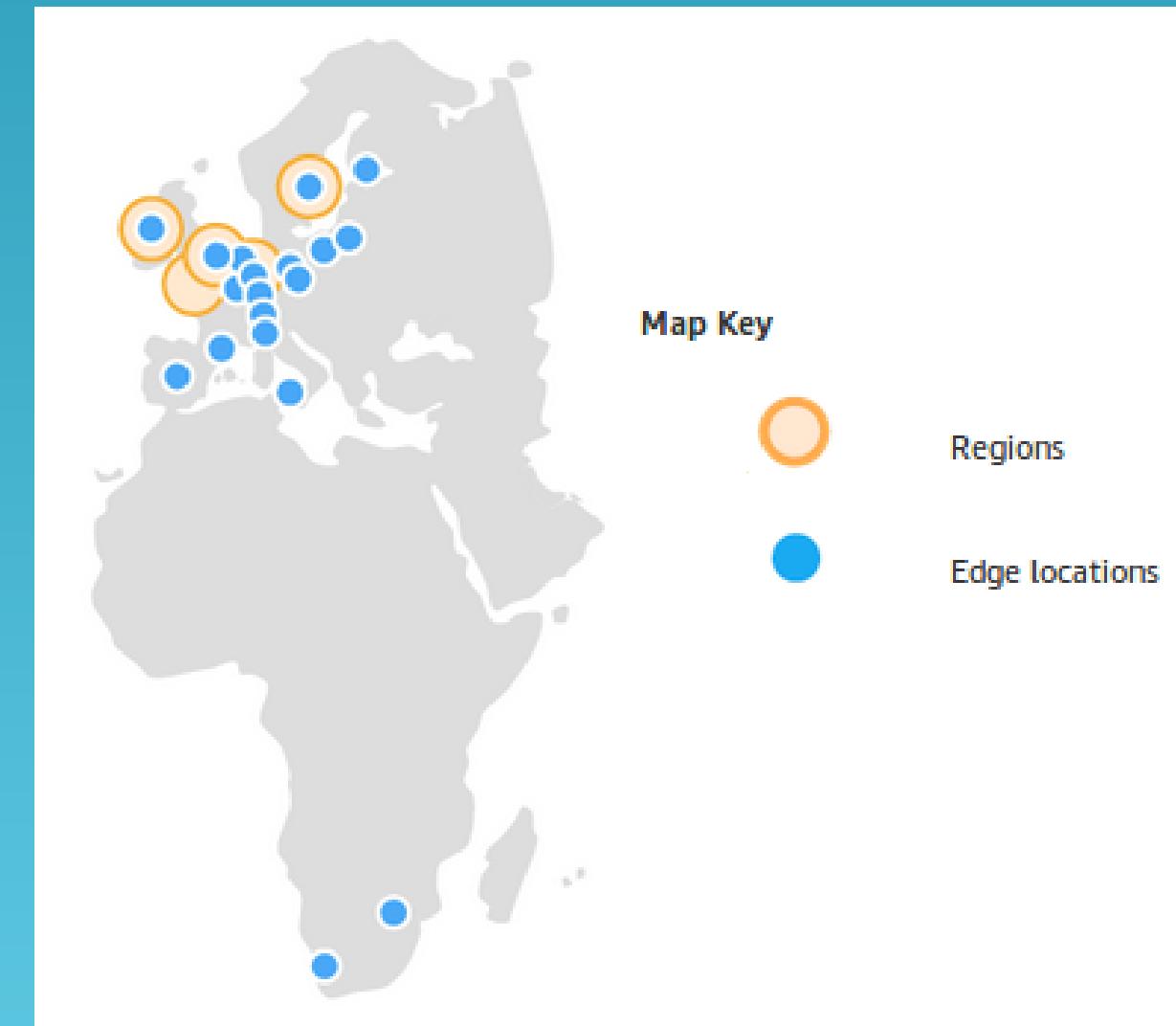
<https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/>

# Regions & Availability Zones

		Inter-region EC2 latency (ping)											
		TO											
FROM	N. Virginia	Ohio	Oregon	N. California	Sao Paolo	Ireland	Frankfurt	Mumbai	Tokyo	Seoul	Singapore	Sydney	
N. Virginia	2	26	323	149	257	140	169	423	291	340	522	449	
Ohio	23	4	487	101	299	159	189	592	267	316	481	408	
Oregon	159	133	211	41	349	247	280	515	178	228	409	307	
N. California	147	101	375	2	376	289	300	494	215	261	384	310	
Sao Paolo	261	299	427	376	2	390	383	603	508	567	756	677	
Ireland	139	161	449	287	381	1	44	266	415	464	535	590	
Frankfurt	170	187	470	301	377	43	5	233	434	493	675	600	
Mumbai	419	592	472	495	602	265	235	2	278	264	135	476	
Tokyo	296	270	313	219	508	417	438	278	2	66	138	205	
Seoul	338	316	392	262	566	463	494	263	66	1	144	262	
Singapore	525	481	479	385	753	528	695	137	139	143	4	333	
Sydney	449	408	456	311	683	589	606	476	205	265	334	1	
* all values are in milliseconds													

<https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/>

# Region Maps and Edge Networks



[https://aws.amazon.com/about-aws/global-infrastructure/regions\\_az/?p=ngi&loc=2](https://aws.amazon.com/about-aws/global-infrastructure/regions_az/?p=ngi&loc=2)

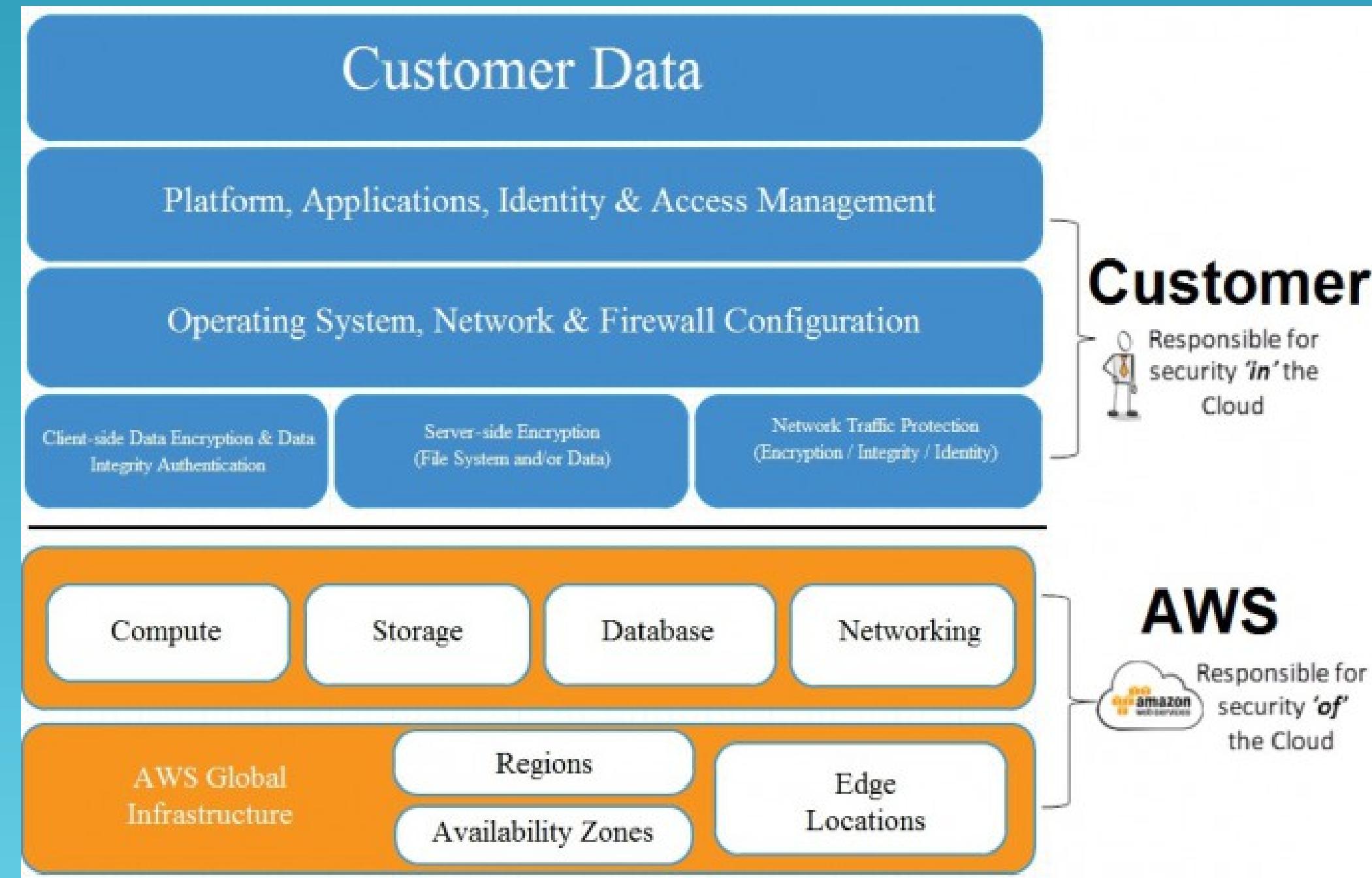
# Region Table

Americas	Europe / Middle East / Africa	Asia Pacific	Ireland	Frankfurt	London	Paris	Stockholm
Services Offered:							
Alexa for Business							
Amazon API Gateway			/	/	/	/	/
Amazon AppStream 2.0			/	/			
Amazon Athena			/	/	/		
Amazon Aurora - MySQL-compatible			/	/	/	/	/
Amazon Aurora - PostgreSQL-compatible			/	/	/	/	/
Amazon Chime							
Amazon Cloud Directory			/		/	/	
Amazon CloudSearch			/	/			
Amazon CloudWatch			/	/	/	/	/

<https://status.aws.amazon.com/>

<https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/?p=ngi&loc=4>

# Podział obowiązków w AWS



# AWS - najważniejsze adresy

- <https://aws.amazon.com/>
- <https://console.aws.amazon.com/>
- <https://docs.aws.amazon.com/>
- <https://aws.amazon.com/blogs/>
- <https://www.youtube.com/user/AmazonWebServices>

# Consola WWW

AWS Management Console

**AWS services**

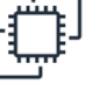
Find a service by name or feature (for example, EC2, S3 or VM, storage)

▶ Recently visited services

▶ All services

**Build a solution**

Get started with simple wizards and automated workflows.

**Launch a virtual machine**  
With EC2  
~2-3 minutes  


**Build a web app**  
With Elastic Beanstalk  
~6 minutes  


**Build using virtual servers**  
With Lightsail  
~1-2 minutes  


**Connect an IoT device**  
With AWS IoT  
~5 minutes  


**Start a development project**  
With CodeStar  
~5 minutes  


**Register a domain**  
With Route 53  
~3 minutes  


▶ See more

**Learn to build**

Learn to deploy your solutions through step-by-step guides, labs, and videos. [See all](#)

**Websites and Web Apps**  
3 videos, 3 tutorials, 3 labs  


**Storage**  
3 videos, 3 tutorials, 3 labs  


**Databases**  
3 videos, 3 tutorials, 3 labs  


**DevOps**  
3 videos, 3 tutorials, 3 labs

**Machine Learning**  
3 videos, 3 tutorials, 3 labs

**Big Data**  
3 videos, 1 lab

**Access resources on the go**

Access the Management Console using the AWS Console Mobile App. [Learn more](#)

**Explore AWS**

**Amazon Redshift**  
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. [Learn more](#)

**Run Serverless Containers with AWS Fargate**  
AWS Fargate runs and scales your containers without having to manage servers or clusters. [Learn more](#)

**Scalable, Durable, Secure Backup & Restore with Amazon S3**  
Discover how customers are building backup & restore solutions on AWS that save money. [Learn more](#)

**AWS Marketplace**  
Find, buy, and deploy popular software products that run on AWS. [Learn more](#)

**Have feedback?**

Submit feedback to tell us about your experience with the AWS Management Console.

# Big Data w chmurze

## AWS

### Warsztaty

# Dane do warsztatów

## <http://tiny.cc/wfbm3y>

[https://www.dropbox.com/sh;brjcqgxlms3xcsr/AAAqXkpOJBnzBPaqq\\_S3qklka?dl=0](https://www.dropbox.com/sh;brjcqgxlms3xcsr/AAAqXkpOJBnzBPaqq_S3qklka?dl=0)

# Podstawowe narzędzia w AWS

# Storage

Narzędzie	Opis
Simple Storage Service (S3)	Magazyn obiektowy (pliki + metadane)
Elastic Block Store (EBS)	Dyski dla maszyn EC2 (2 GB/s)
Elastic File System (EFS)	Współdzielony dysk sieciowy (NFS) dla systemów Linux (10+ GB/s)
FSx	Szybki system plików (100 GB/s) do uruchamiania bardzo ciężkich procesów obliczeniowych (Windows File Server lub Linux Lustre)
S3 Glacier	Tanie składowanie kopii bezpieczeństwa w S3
Storage Gateway	Komunikacja pomiędzy chmurą a zasobami lokalnymi (dane)
AWS Backup	Tworzenie kopii bezpieczeństwa tego co mamy w AWS jak EBS, EFS, bazy danych i więcej

# Compute

Narzędzie	Opis
Elastic Compute Cloud (EC2)	Maszyny wirtualne bądź fizyczne
Lightsail	Virtual Private Server (VPS)
Elastic Container Service (ECS)	Uruchamianie kontenerów Docker
Elastic Container Registry (ECR)	Repozytorium kontenerów Docker dla ECS
Elastic Container Service for Kubernetes (EKS)	Kubernetes zarządzany przez AWS, certyfikowana platforma
Lambda	Usługa typu Serverless, uruchamiamy kod "bez serwerów" w chmurze
Batch	Uruchamianie procesów w trybie wsadowym
Elastic Beanstalk	Hosting dla aplikacji Java, .NET, PHP, Node.js, Python, Ruby, Go, "Docker" za pomocą Apache HTTP, Apache Tomcat, Nginx, Passenger, and IIS
Serverless Application Repository	Repozytorium aplikacji Serverless
Elastic Load Balancing (ELB)	Load balancer

# Amazon Simple Storage Service (S3)

# S3

- Obiektowy magazyn danych
- Jedna z podstawowych usług w AWS, integruje się z wszystkimi innymi serwisami
- Dane organizowane są w “bucket” (kubel, wiadro)
- Brak limitu obiektów, brak ograniczenia wielkości “wiadra”, każdy obiekt do 5TB
- Wersjonowanie
- Uprawnienia
- Możliwość szyfrowania danych
- 99.999999999 % trwałości danych (durability) i 99.99% dostępności (availability)

# S3

- Dostęp HTTP, REST, SOAP
- Wysokie skalowalne, niezawodne i szybkie źródło danych
- Logi dostępu na potrzeby audytu (access logs)
- Obsługiwany przez wiele narzędzi Open Source i Big Data (także HDFS)
- Odpowiednikiem on-premise jest Apache Ozone dla HDFS

# S3

- Obiekt to plik powiązany z metadanymi
- Dane są trzymane w wiaderkach (“bucket”), jedno konto może mieć 100 wiaderek
- Bucket musi mieć unikalną nazwę w skali świata!
- Każdy bucket ma uprawnienia, może być publiczny
- <http://BucketName.s3.amazonaws.com/ObjectKey>
- Wspiera wersjonowanie obiektów

# S3 - cena

- Tanie przechowywanie wielu plików (także duże dane) - 23\$ za 1TB
- Płatność za użycie, cena zależy od lokalizacji
- Płacimy za składowanie, zapytania oraz transfer danych
- <https://aws.amazon.com/s3/pricing/>

# S3 - zastosowania

- Składowanie danych
- Data Lake
- Backup
- App File Hosting
- Media Hosting
- AMI i Snapshoty
- Software Delivery
- WWW Hosting

# S3 - przydatne adresy

- <https://aws.amazon.com/s3/>
- <https://aws.amazon.com/s3/features/>
- <https://aws.amazon.com/s3/storage-classes/>
- <https://aws.amazon.com/s3/getting-started/>
- <https://aws.amazon.com/s3/developer-resources/>
- <https://aws.amazon.com/s3/pricing/>
- <https://aws.amazon.com/s3/faqs/>
- <https://docs.aws.amazon.com/s3>

# S3 - Poziomy przechowywania danych

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive**
Designed for durability	99.999999999% (11.9%)	99.999999999% (11.9%)	99.999999999% (11.9%)	99.999999999% (11.9%)	99.999999999% (11.9%)	99.999999999% (11.9%)
Designed for availability	99.99%	99.9%	99.9%	99.5%	N/A	N/A
Availability SLA	99.9%	99%	99%	99%	N/A	N/A
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

<https://aws.amazon.com/s3/storage-classes/>

# S3 - CLI

```
aws s3 help
```

```
aws s3 ls s3://athena-examples/elb/plaintext/  
aws s3 ls s3://athena-examples/elb/plaintext/ --recursive
```

```
aws s3 ls s3://athena-examples/elb/ --recursive --human-readable --summarize
```

```
aws s3 ls s3://elasticmapreduce/samples/hive-ads/tables/impressions/
```

<https://docs.aws.amazon.com/cli/latest/reference/s3/index.html>

# S3 vs HDFS

AWS S3	HDFS
Transport po sieci	Data Locality
SaaS	IaaS lub PaaS
Serverless	Maszyny działające
Małe i duże zbiory danych	Duże zbiory danych
Małe i duże pliki	Bardzo duże pliki
Maksymalna wielkość pliku 5TB	Brak teoretycznego limitu wielkości pliku
Wydajność zależy od AWS	My sterujemy wydajnością (maszyny, dyski, sieć, etc.)
Klasy składowania danych	Dane Hot i Cold

<https://databricks.com/blog/2017/05/31/top-5-reasons-for-choosing-s3-over-hdfs.html>

# Amazon Elastic Block Store (EBS)

# EBS

- Przestrzeń dyskowa dla maszyn wirtualnych (block level store)
- Domyślna replikacja między “Availability Zone”
- Bardzo duża wydajność, wyższa cena niż S3 zależna od regionu
- Dostępne jest wiele typów: EBS General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), and Cold HDD (sc1) od 1GB do 16TB
- Można szyfrować
- Można tworzyć snapshoty składowane w S3

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>

# Import danych do AWS

# Snowball (100 TB)



# AWS Snowmobile



# AWS Snowmobile

## Video Playback Disabled

<https://www.youtube.com/watch?v=8vQmTZTq7nw&t=60>

# AWS Snowmobile

- 100 PB
- Ciężarówka AWS = 1000 SnowBalls
- Przy złączu 1Gbps transmisja do chmury trwała by 20 lat przez Internet

Ktoś tego używa :)

**Video Playback Disabled**

<https://www.youtube.com/watch?v=iB86NtOyw4E>

# Amazon Elastic Compute Cloud (EC2)

# EC2

- Usługa maszyn wirtualnych
- Dostarcza zasoby pod większość narzędzi w AWS
- Duży wybór, różne rodzaje maszyn  
<https://aws.amazon.com/ec2/instance-types/>
- AMI - szablony maszyn (system operacyjny, software i ustawienia)
- Instancja - uruchomiona lub zatrzymana maszyna
- Marketplace - sklep z gotowymi szablonami, także oprogramowanie firm trzecich, w cenie uwzględniona jest licencja, różne zasady supportu dla firm trzecich  
<https://aws.amazon.com/marketplace>
- Snapshoty - obraz instancji do odtworzenia (zapamiętany stan maszyny)

# EC2 - przydatne adresy

- <https://aws.amazon.com/ec2/>
- <https://aws.amazon.com/ec2/features/>
- <https://aws.amazon.com/ec2/getting-started/>
- <https://aws.amazon.com/ec2/developer-resources/>
- <https://aws.amazon.com/ec2/videos/>
- <https://aws.amazon.com/ec2/pricing/>
- <https://aws.amazon.com/ec2/faqs/>
- <https://docs.aws.amazon.com/ec2>

# EC2 - typyinstancji

- General Purpose – (A1, T3, T2, M5, M5a, M4, T3a)
- Computer Optimized – (C5, C5n, C4)
- Memory Optimized – (R5, R5a, R4, X1e, X1, z1d)
- Accelerated Computing - (P3, P2, G3, F1) - GPU i FPGA
- Storage optimized - (H1, I3, D2)

<https://aws.amazon.com/ec2/instance-types/>

<https://aws.amazon.com/ec2/physicalcores/>

# EC2 - zakup

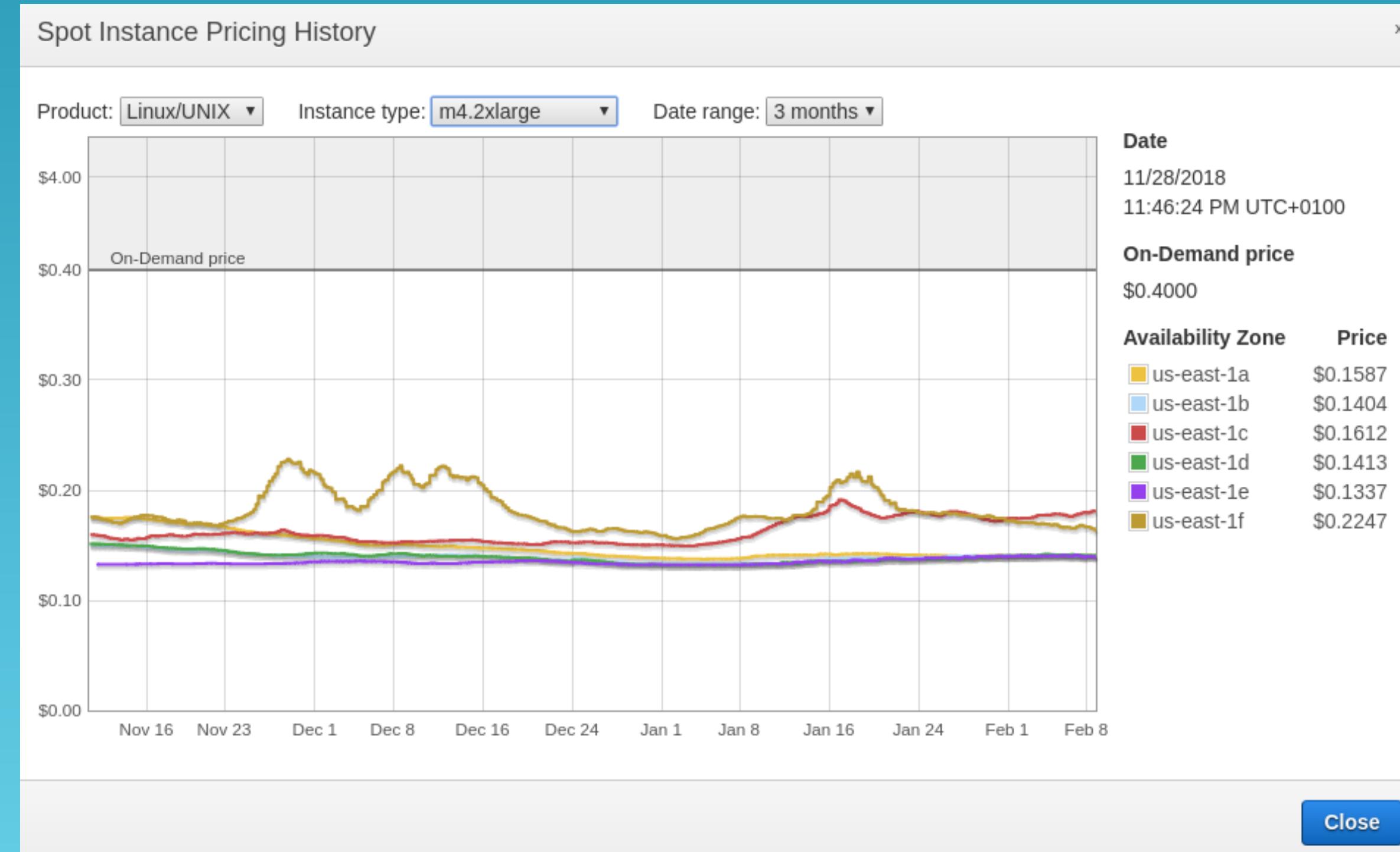
- On-Demand Instances - płatność za godzinę
- Reserved Instances - z góry rezerwujemy i płacimy za moc obliczeniową dzięki czemu mamy dużą zniżkę, rezerwacja na rok lub 3 lata
- Scheduled Instances - Instancje zarezerwowane w sposób powtarzalny (np. co miesiąc)
- Spot Instances - giełda wolnych zasobów, duże zniżki ale brak pewności że będziemy mieć żądaną moc
- Dedicated Hosts - Maszyny fizyczne na wyłączność

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html>

# EC2 - cena

- Cena zależy od szablonu maszyny i czasu uruchomienia
- Często naliczanie sekundowe (minimum 60 sekund), pozostałe godzinowo
  - <https://aws.amazon.com/blogs/aws/new-per-second-billing-for-ec2-instances-and-ebs-volumes/>
- Można optymalizować ceny za pomocą rezerwacji i giełdy (Spot):
  - <https://aws.amazon.com/ec2/pricing/on-demand/>
  - <https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/>
  - <https://aws.amazon.com/ec2/spot/>
- Można wykupić hosty dedykowane (fizyczne maszyny)
  - <https://aws.amazon.com/ec2/dedicated-hosts/pricing/>

# EC2 - instancje Spot



# EC2 - dyski

- Instance Store
  - Dane tracone wraz z maszyną (nawet tylko gdy ją zatrzymamy!)
  - Dane składowane lokalnie tam gdzie maszyna
- EBS
  - Dane niezależne od maszyny
  - Składowane w oddzielnych zasobach, komunikacja z maszyną

<https://aws.amazon.com/premiumsupport/knowledge-center/instance-store-vs-ebs/>

# EC2 - dyski

	<b>Restart</b>	<b>Stop / Start</b>	<b>Usunięcie</b>
Host	Ten sam	Nowy	-
IP	Zostaje	Publiczne IP nowe	-
Elastic IP	Zostaje	Zostaje	Odłączone
Instance Store	Zachowuje dane	Dane usunięte	Dane usunięte
EBS	Zachowuje dane	Zachowuje dane	Dane usunięte (domyślnie, można wybrać inaczej)
Naliczanie opłat	Kontynuowane	Stop i Start	Zatrzymany

# AWS Identity and Access Management (IAM)

# IAM

- Zapewnia bezpieczeństwo w AWS
- Uwierzytelnianie (authentication) - ustalenie tożsamości
  - coś co wiesz (something you know) – informacja znana tylko uprawnionym (hasło lub klucz prywatny)
  - coś co masz (something you have) – przedmiot będący w posiadaniu (klucz do zamka, token fizyczny)
  - coś czym jesteś (something you are) – metody biometryczne
- Autoryzacja (authorization) - dostęp do zasobów
- Użytkownicy i grupy użytkowników
- Multi-Factor Authentication (MFA)

# IAM

Narzędzie	Dostęp
Master account (root)	Email + Password
Management Console	Username + Password
API / CLI / SDK	Access ID + Secret Key

# IAM - dobre praktyki

Nie zaleca się korzystania z konta typu "root"!!!

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

# IAM - limity

[https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\\_iam-limits.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-limits.html)

# IAM

- Policies (polityki)
  - Mechanizm autoryzacji
  - JSON
  - Dwa typy
    - Managed - zarządzane przez AWS lub klienta
    - Inline - coś bardzo nietypowego używane w wąskim zakresie

# IAM

- Roles (role)
  - Można traktować jak coś w rodzaju użytkownika
  - Można przypisać do nich polityki
  - Można je przydzielić do usługi, użytkownika lub aplikacji

# IAM - przydatne adresy

- <https://aws.amazon.com/iam/>
- <https://aws.amazon.com/iam/faqs/>
- <https://docs.aws.amazon.com/iam/>

# AWS Command Line Interface (CLI)

# CLI

- Możliwość pracy z AWS w terminalu
- Dostęp do usług których nie ma w panelu www

# CLI - instalacja

```
pip3 install awscli --upgrade --user
```

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>

# CLI - konfiguracja

- Kilka metod konfiguracji

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>

# CLI - konfiguracja

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_DEFAULT_REGION=us-east-1
```

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-envvars.html>

# CLI - konfiguracja

~/.aws/credentials

[default]

aws\_access\_key\_id=AKIAIOSFODNN7EXAMPLE

aws\_secret\_access\_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

~/.aws/config

[default]

region=us-east-1

output=json

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html>

# CLI - przydatne adresy

- <https://aws.amazon.com/cli/>
- <https://docs.aws.amazon.com/iam/>
- <https://docs.aws.amazon.com/cli/latest/reference/index.html>

# Narzędzia Big Data w AWS

Narzędzie	Opis
Athena	Silnik SQL
EMR	Dystrybucja Big Data (Hadoop, Spark, Hive, etc.)
CloudSearch	Wyszukiwanie analogiczne do Elasticsearch
Elasticsearch Service	Elasticsearch zarządzany przez AWS
Kinesis (Kinesis Analytics / Kinesis Streams)	Silnik umożliwiający przetwarzanie strumienia danych
QuickSight	BI płatny za użytkownika (cena zależy od pamięci)
Data Pipeline	ETL + Workflow, można użyć narzędzi EMR, wiele źródeł i wyjść dla danych
AWS Glue	Silnik ETL oparty o Spark
Managed Streaming for Kafka (MSK)	Kafka zarządzana przez AWS

Use case	Narzędzia
Data Lake	S3, EMR (Hadoop)
ETL	Glue, EMR, Data Pipeline
Zapytania	Athena, EMR, Redshift
AI, ML	EMR, SageMaker
Data Catalog	Glue Data Catalog, EMR (Hive)

# AWS re:Invent 2018: Big Data Analytics Architectural Patterns & Best Practices

## Video Playback Disabled

<https://www.youtube.com/watch?v=ovPhelbY7U8>

# AWS Lake Formation

## Video Playback Disabled

<https://www.youtube.com/watch?v=uVF73MXYay8>

# Amazon Athena

# Athena

- Silnik SQL wykonujący zapytania do danych w S3
- Bazuje na silniku **Presto** (SQL Query)
- SQL DDL zgodny z **Hive DDL** (Data definition language)
- Usługa Serverless
- Automatyczne skalowanie mocy obliczeniowej
- Zintegrowana z AWS Glue Data Catalog i innymi usługami AWS (<https://docs.aws.amazon.com/athena/latest/ug/athena-aws-service-integrations.html>)
- Dostęp przez WWW, API oraz CLI
- Obsługa sterowników ODBC i JDBC
- Format: CSV, JSON, ORC, Avro, and Parquet

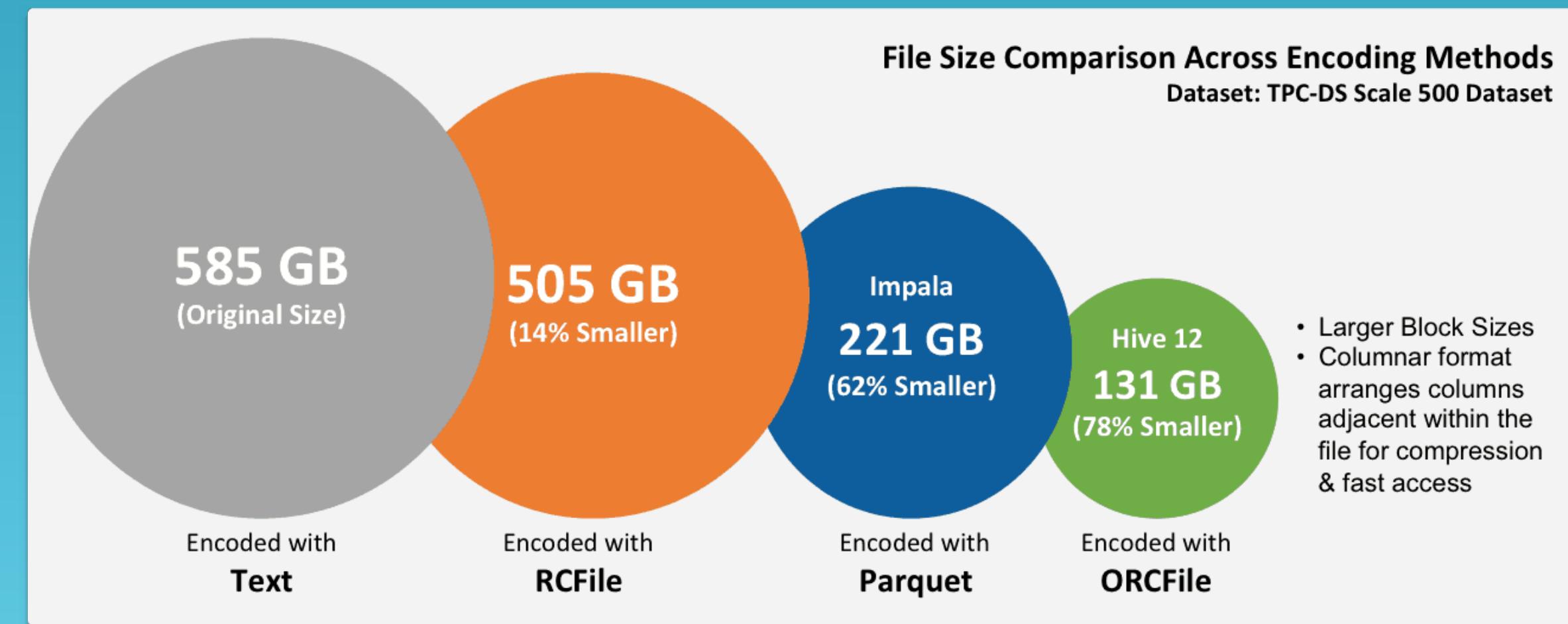
# Athena - przydatne adresy

- <https://aws.amazon.com/athena/>
- <https://aws.amazon.com/athena/features/>
- <https://aws.amazon.com/athena/getting-started/>
- <https://aws.amazon.com/athena/resources/>
- <https://aws.amazon.com/athena/pricing/>
- <https://aws.amazon.com/athena/faqs/>
- <https://docs.aws.amazon.com/athena>

# Athena - cena

- Płacimy za ilość "dotkniętych" danych przy wykonaniu zapytania
- \$5 za każdy zeskanowany 1 TB
- Można zwiększyć wydajność i zmniejszyć koszty za pomocą
  - partycjonowanie
  - kompresja (Snappy, Zlib, LZO, GZIP)
  - formaty kolumnowe (Apache ORC lub Apache Parquet)
- Zapytania anulowane też są płatne za to co zdążyła zeskanować
- Zapytania z błędem nie są płatne
- Płacimy z dokładnością do 1MB, minimum 10MB za zapytanie
- <https://aws.amazon.com/athena/pricing/>

# Athena - kompresja i format pliku



<https://hortonworks.com/blog/orcfile-in-hdp-2-better-compression-better-performance/>

# Athena - tabela filmów

```
DROP TABLE IF EXISTS movielens.movies;

CREATE EXTERNAL TABLE movielens.movies (
    `movieid` int,
    `title` string,
    `genres` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
    'serialization.format' = ',',
    'field.delim' = '@'
) LOCATION 's3://aws-educate-pw-radek/movielens/movies/'
TBLPROPERTIES (
    'has_encrypted_data'='false',
    'skip.header.line.count' = '1'
);

select * from movielens.movies limit 10;
```

# Athena - tabela ocen

```
DROP TABLE IF EXISTS movielens.ratings;

CREATE EXTERNAL TABLE IF NOT EXISTS movielens.ratings (
    `userid` int,
    `movieid` int,
    `rating` double,
    `time` int
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
    'serialization.format' = ',',
    'field.delim' = '@'
) LOCATION 's3://aws-educate-pw-radek/movielens/ratings/'
TBLPROPERTIES (
    'has_encrypted_data'='false',
    'skip.header.line.count' = '1'
);

select * from movielens.ratings limit 10;
```

# Athena - Zadanie 1

Na podstawie zbiorów movies.dat oraz ratings.dat wylicz średnią ocenę dla każdego filmu i zwróć TOP 10 filmów wszechczasów które mają minimum 100 ocen!

# Athena - Zadanie 1

```
select m.movieid, m.title, avg(r.rating) rate, count(1) votes
from movies m
join ratings r on m.movieid = r.movieid
group by m.movieid, m.title
having count(1) > 100
order by rate desc limit 10 ;
```

# Athena - Zadanie 2

Używając danych o filmach znaleźć gatunek, z którego pochodzi najwięcej filmów

# Athena - Zadanie 2

```
select genre
from movies
CROSS JOIN UNNEST(split(genres, '|')) AS t (genre);
```

# Athena - Zadanie 2

```
select sq.genre as genre, count(1) as counted
from (
    select genre
    from movies
    CROSS JOIN UNNEST(split(genres, '|')) AS t (genre)
) sq
group by genre
order by counted desc;
```

# Athena - wyniki

Wyniki są zapisywane w athena:

**aws-athena-query-results-<ACCOUNTID>-<REGION>**

w formacie:

**{QueryLocation}/{QueryName|Unsaved}/{yyyy}/{mm}/{dd}/{QueryID}.csv**

**{QueryLocation}/{QueryName|Unsaved}/{yyyy}/{mm}/{dd}/{QueryID}.csv.metadata**

# Athena - Web UI

- Zapytania
- Tworzenie/usuwanie tabel i baz danych
- Historia
- Zapisywanie zapytań
- Przegląd wyników (+ eksport)
- Szybkie polecenia na tabeli

# Athena - widoki

Możliwość tworzenia widoków (tylko logiczne)

```
CREATE OR REPLACE VIEW action_movies AS
SELECT *
FROM movies
WHERE genres like '%Action%';

select * from action_movies;
```

# Athena - partycje

```
aws s3 ls s3://elasticmapreduce/samples/hive-ads/tables/impressions/  
    PRE dt=2009-04-12-13-00/  
    PRE dt=2009-04-12-13-05/  
    PRE dt=2009-04-12-13-10/  
    PRE dt=2009-04-12-13-15/  
    PRE dt=2009-04-12-13-20/  
    PRE dt=2009-04-12-14-00/  
    PRE dt=2009-04-12-14-05/  
    PRE dt=2009-04-12-14-10/  
    PRE dt=2009-04-12-14-15/
```

# Athena - partycje

```
CREATE EXTERNAL TABLE impressions (
    requestBeginTime string,
    adId string,
    impressionId string,
    referrer string,
    userAgent string,
    userCookie string,
    ip string,
    number string,
    processId string,
    browserCookie string,
    requestEndTime string,
    timers struct<modelLookup:string, requestTime:string>,
    threadId string,
    hostname string,
    sessionId string)
```

# Athena - partycje

```
...  
PARTITIONED BY (dt string)  
ROW FORMAT serde 'org.apache.hive.hcatalog.data.JsonSerDe'  
with serdeproperties (  
'paths'='requestBeginTime, adId, impressionId,  
referrer, userAgent, userCookie, ip'  
)  
LOCATION 's3://elasticmapreduce/samples/hive-ads/tables/impressions/' ;
```

# Athena - partycje

MSCK REPAIR TABLE impressions

<https://docs.aws.amazon.com/athena/latest/ug/msck-repair-table.html>

# Athena - partycje

```
SELECT dt,impressionid
FROM impressions
WHERE dt<'2009-04-12-14-00' and dt>='2009-04-12-13-00'
ORDER BY dt DESC LIMIT 100
```

# Athena - PARQUET

```
CREATE TABLE movielens.movies_parquet
WITH (
    format='PARQUET',
    external_location='s3://aws-educate-pw-radek/movies_parquet'
) AS
select * from movies;
```

# Athena - ORC

```
CREATE TABLE movielens.movies_orc
WITH (
    format='ORC',
    external_location='s3://aws-educate-pw-radek/movies_orc'
) AS
select * from movielens.movies;
```

# Athena - PARQUET vs ORC

```
select count(*) from movielens.movies where genres like '%Action';
```

```
select count(*) from movielens.movies_parquet where genres like '%Action';
```

```
select count(*) from movielens.movies_orc where genres like '%Action';
```

# Amazon Elastic MapReduce (AWS EMR)

# EMR

- Dystrybucja Big Data tworzona przez Amazon
- Wbrew nazwie, jest to kompletna dystrybucja, nie tylko usługa MapReduce w chmurze
- Zarządzana przez AWS (Usługa managed, PAAS)
- Integracja z innymi narzędziami AWS
- Dane mogą być przechowywane i używane wprost z S3
- Dostępnych wiele maszyn EC2
- Płatność zgodnie z EC2

# EMR

- Dostępne większość znanych technologii open source dostępnych np. w HDP plus Presto, Flink czy MXNet
- Możliwość tworzenia klastra “na żądanie” i “niszczenie” go po zakończeniu prac (płatimy tylko za użycie)
- Tryb pracy jako “cluster”, działa dopóki go nie wyłączymy, oraz wersja “step” czyli zrób coś i zakończ działanie maszyn
- Dostęp web, skrypty CLI
- Wbudowane security (AWS based)
- Możliwość uruchomienia Notebooka w ramach klastra

# EMR - przydatne adresy

- <https://aws.amazon.com/emr>
- <https://aws.amazon.com/emr/getting-started/>
- <https://docs.aws.amazon.com/emr>
- <https://aws.amazon.com/emr/faqs/>

# EMR - tools

Software Configuration

Release emr-5.20.0

<input checked="" type="checkbox"/> Hadoop 2.8.5	<input type="checkbox"/> Zeppelin 0.8.0	<input type="checkbox"/> Livy 0.5.0
<input type="checkbox"/> JupyterHub 0.9.4	<input type="checkbox"/> Tez 0.9.1	<input type="checkbox"/> Flink 1.6.2
<input type="checkbox"/> Ganglia 3.7.2	<input type="checkbox"/> HBase 1.4.8	<input checked="" type="checkbox"/> Pig 0.17.0
<input checked="" type="checkbox"/> Hive 2.3.4	<input type="checkbox"/> Presto 0.214	<input type="checkbox"/> ZooKeeper 3.4.13
<input type="checkbox"/> MXNet 1.3.1	<input type="checkbox"/> Sqoop 1.4.7	<input type="checkbox"/> Mahout 0.13.0
<input checked="" type="checkbox"/> Hue 4.3.0	<input type="checkbox"/> Phoenix 4.14.0	<input type="checkbox"/> Oozie 5.0.0
<input type="checkbox"/> Spark 2.4.0	<input type="checkbox"/> HCatalog 2.3.4	<input type="checkbox"/> TensorFlow 1.12.0

# EMR - tools

- <https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-release-components.html>
- <https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-release-5x.html>

# EMR - porty

Usługa	Port
YARN ResourceManager	<a href="http://master-public-dns-name:8088/">http://master-public-dns-name:8088/</a>
YARN NodeManager	<a href="http://slave-public-dns-name:8042/">http://slave-public-dns-name:8042/</a>
Hadoop HDFS NameNode	<a href="http://master-public-dns-name:50070/">http://master-public-dns-name:50070/</a>
Hadoop HDFS DataNode	<a href="http://slave-public-dns-name:50075/">http://slave-public-dns-name:50075/</a>
Spark HistoryServer	<a href="http://master-public-dns-name:18080/">http://master-public-dns-name:18080/</a>
HBase UI	<a href="http://master-public-dns-name:16010/">http://master-public-dns-name:16010/</a>
Oozie	<a href="http://master-public-dns-name:11000/oozie/">http://master-public-dns-name:11000/oozie/</a>
Zeppelin	<a href="http://master-public-dns-name:8890/">http://master-public-dns-name:8890/</a>
JupyterHub	<a href="http://master-public-dns-name:9443/">http://master-public-dns-name:9443/</a>
Hue	<a href="http://master-public-dns-name:8888/">http://master-public-dns-name:8888/</a>
Ganglia	<a href="http://master-public-dns-name/ganglia/">http://master-public-dns-name/ganglia/</a>

<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-web-interfaces.html>

# EMR - Hive on HDFS

```
show databases;
```

```
create database movielens;
```

```
show tables;
```

# EMR - Hive on HDFS

```
DROP TABLE IF EXISTS movies;
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS movies (
  MovieID INT,
  Title STRING,
  Genres STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '@';
```

```
LOAD DATA INPATH '/user/hadoop/movielens/movies/movies.dat'
OVERWRITE INTO TABLE movies;
```

```
select * from movies limit 10;
```

# EMR - Hive on HDFS

```
drop table if exists ratings;
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ratings (
    UserID INT,
    MovieID INT,
    Rating DOUBLE,
    Ts INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '@';
```

```
LOAD DATA INPATH '/user/hadoop/movielens/ratings/ratings.dat'
OVERWRITE INTO TABLE ratings;
```

```
select * from ratings limit 10;
```

# EMR - Hive on HDFS - Zadanie 1

Na podstawie zbiorów movies.dat oraz ratings.dat wylicz średnią ocenę dla każdego filmu i zwróć TOP 10 filmów wszechczasów które mają minimum 100 ocen!

# EMR - Hive on HDFS - Zadanie 1

```
select m.title, avg(r.rating) rate, count(1) votes
from movies m
join ratings r on m.movieid = r.movieid
group by m.title
having votes > 100
order by rate desc limit 10 ;
```

# EMR - Hive on S3

```
drop table if exists moviess3;

CREATE external TABLE moviess3 (
    MovieID INT,
    Title STRING,
    Genres STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '@'
STORED AS
INPUTFORMAT
    'com.amazonaws.emr.s3select.hive.S3SelectableTextInputFormat'
OUTPUTFORMAT
    'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://aws-educate-pw-radek/movielens/movies/'
TBLPROPERTIES (
    "s3select.format" = "csv",
    "s3select.headerInfo" = "ignore"
);
```

# EMR - Hive on S3

```
drop table if exists ratingss3;

CREATE external TABLE ratingss3 (
    UserID INT,
    MovieID INT,
    Rating DOUBLE,
    Ts INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '@'
STORED AS
INPUTFORMAT
    'com.amazonaws.emr.s3select.hive.S3SelectableTextInputFormat'
OUTPUTFORMAT
    'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://aws-educate-pw-radek/movielens/ratings/'
TBLPROPERTIES (
    "s3select.format" = "csv",
    "s3select.headerInfo" = "ignore"
```

# EMR - Hive on S3 - Zadanie 1

Na podstawie zbiorów movies.dat oraz ratings.dat wylicz średnią ocenę dla każdego filmu i zwróć TOP 10 filmów wszechczasów które mają minimum 100 ocen!

# EMR - Hive on S3 - Zadanie 1

```
select m.title, avg(r.rating) rate, count(1) votes
from moviess3 m
join ratingss3 r on m.movieid = r.movieid
group by m.title
having votes > 100
order by rate desc limit 10 ;
```

# EMR - Hive - Zadanie 2

Używając danych o filmach znaleźć gatunek, z którego pochodzi najwięcej filmów

# EMR - Hive - Zadanie 2

```
SET s3select.filter=false;

select q.genre as genre, count(1) as counted
from (select explode(split(genres, "\\")) AS genre from movies) q
group by genre
order by counted desc;
```

# EMR - steps

Add step

X

Step type: Hive program

Name: Hive program

Script S3 location\*: s3://aws-educate-pw-radek/hive-s3-query.hql S3 location of your Hive script.  
s3://<bucket-name>/<path-to-file>

Input S3 location: s3:// S3 location of your Hive input files.  
s3://<bucket-name>/<folder>/

Output S3 location: s3:// S3 location of your Hive output files.  
s3://<bucket-name>/<folder>/

Arguments:  Specify optional arguments for your script.

Action on failure: Continue What to do if the step fails.

**Cancel** **Add**

---

# EMR - Pig

```
movies = LOAD '/user/hadoop/movielens/movies' USING PigStorage('@')  
AS (movieid:int, title:chararray, genres:chararray);
```

```
dump movies;
```

```
store movies into '/user/hadoop/pigresult' USING PigStorage('@');
```

# EMR - Spark

<https://github.com/sagespl/HADOOP>

```
git clone https://github.com/sagespl/HADOOP.git
cd HADOOP
mvn clean install -DskipTests=true
```

# EMR - Spark

```
val dataPath: String = "s3a://aws-educate-pw-radek"
```

```
cd spark/spark-core/  
mvn clean install -DskipTests=true -Puber
```

# EMR - Spark

```
scp -i ~/awseducate.pem \
target/spark-core-1.0-SNAPSHOT-jar-with-dependencies.jar \
hadoop@ec2-54-236-53-104.compute-1.amazonaws.com://home/hadoop
```

# EMR - Spark

```
ssh -i ~/awseducate.pem hadoop@ec2-54-236-53-104.compute-1.amazonaws.com
```

```
spark-submit --master yarn \  
--deploy-mode cluster \  
--num-executors 3 \  
--class pl.com.sages.spark.core.MovieGenres \  
spark-core-1.0-SNAPSHOT-jar-with-dependencies.jar
```

# EMR - steps

Add step X

Step type: Spark application

Name: Spark application

Deploy mode: Cluster Run your driver on a slave node (cluster mode) or on the master node as an external client (client mode).

Spark-submit options: `--class pl.com.sages.spark.core.MovieGenres` Specify other options for spark-submit.

Application location\*: `s3://aws-educate-pw-radek/spark-core-1.0-SNAPSHOT` Path to a JAR with your application and dependencies (client deploy mode only supports a local path).

Arguments:  Specify optional arguments for your application.

Action on failure: Continue What to do if the step fails.

Cancel Add

# EMR - notebooks

- Zeppelin
- Jupyter

# Sztuczna Inteligencja (AI) w AwS

# Serwisy AI

Usługa	Opis
SageMaker	Platforma Data Science i AI oparta o notebook Jupyter, umożliwia tworzenie serwisów
Comprehend	Natural Language Processing (NLP) and Text Analytics, analiza i rozpoznawanie tekstu
DeepLens	Sztuczna inteligencja zaszyta w kamerze korzystająca z modeli deep-learning, współpraca z innymi narzędziami AWS
Lex	Usługa chatbot, konwersacje głosowe lub tekstowe z użytkownikiem, Natural Language Understanding (NLU)
Machine Learning	Tworzenie modeli bez potrzeby kodowania
Polly	Zmiana tekstu na mowę (text to speech), 25 języków, także polski
Rekognition	Rozpoznawanie obiektów na obrazach i filmach
Transcribe	Automatic speech recognition (ASR), zmiana mowy na tekst
Translate	Tłumaczenie tekstów pomiędzy różnymi językami

# Serwisy AI

Usługa	Opis
Personalize (Preview)	Tworzenie modeli ludzkich na potrzeby rekommendacji
Forecast (Preview)	Tworzenie prognoz na podstawie naszych danych
Textract (Preview)	Optical Character Recognition (OCR), odczytywanie tekstu ze skanowanych dokumentów
AWS RoboMaker	Rozszerzenia dla Robot Operating System (ROS) w chmurze AWS
Deep Learning AMIs	Szablony maszyn EC2 z zainstalowanymi bibliotekami głębokiego uczenia maszynowego

# Serwisy AI

<https://aws.amazon.com/machine-learning/>

# Deep Learning AMIs

- <https://aws.amazon.com/machine-learning/amis/>
- Obrazy maszyn wirtualnych (AMI) przygotowane do pracy w Data Science lub AI
- Wspierane Amazon Linux, Ubuntu Linux, Windows 2016
- Wbudowane popularne biblioteki do Data Science i AI
- Apache MXNet i Gluon, TensorFlow, Microsoft Cognitive Toolkit, Caffe, Caffe2, Theano, Torch, PyTorch, Chainer, Keras

# Popularne narzędzie AI

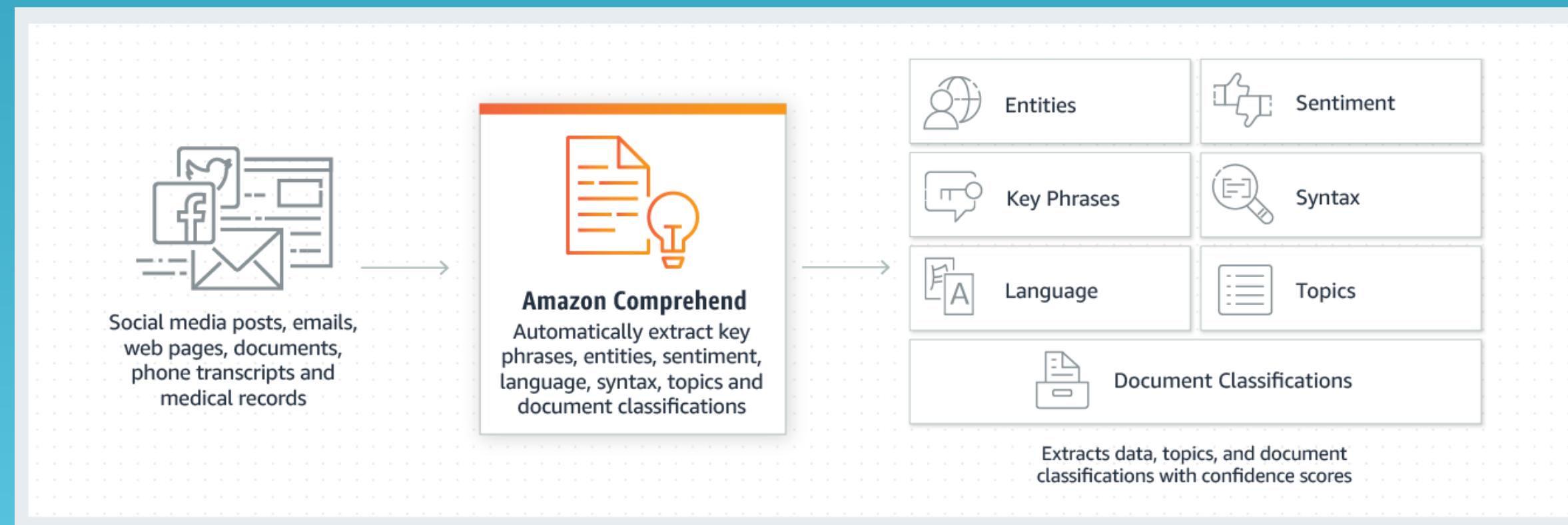
<http://tiny.cc/xacm3y>

# Amazon Comprehend

# Amazon Comprehend

- Natural Language Processing and Text Analytics
- Analiza języka naturalnego
- Dostępna dedykowana wersja dla medycyny
- Można trenować modele na podstawie swoich danych
- Klasteryzacja dokumentów (Topic Modeling)
- Detekcja sentymenu (nacechowanie emocjonalne tekstu)
- Wykrywanie obiektów, organizacji, lokalizacji, dat, osób, słów kluczowych
- Rozpoznawanie języka
- Rozpoznawanie części zdania

# Amazon Comprehend



# Amazon Comprehend

## Video Playback Disabled

<https://www.youtube.com/watch?v=hdXvVyVjPLg>

# Amazon Comprehend - przydatne adresy

- <https://aws.amazon.com/comprehend/>
- <https://aws.amazon.com/comprehend/pricing/>
- <https://aws.amazon.com/comprehend/faqs/>
- <https://docs.aws.amazon.com/comprehend>
- <https://docs.aws.amazon.com/comprehend/latest/dg/supported-languages.html>

# Amazon Comprehend

Amazon.com, Inc. is located in Seattle, WA and was founded July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable Seattle - based companies are Starbucks and Boeing.

# Amazon Comprehend

“I’m just saying ‘A Happy Birthday,’” said Owl carelessly.  
“It’s a nice long one,” said Pooh, very much impressed by it.  
“Well, actually, of course,  
I’m saying ‘A Very Happy Birthday with love from Pooh.’  
Naturally it takes a good deal of pencil to say a long thing like that.”

<https://orangemarmaladebooks.com/2010/04/22/fiction-favorites-winnie-the-pooh/>

# Amazon Comprehend

Tell me why?

I don't like Mondays.

Tell me why?

I don't like Mondays.

Tell me why?

I don't like Mondays.

I want to shoot

The whole day down.

<https://www.songtexte.com/songtext/the-boomtown-rats/i-dont-like-mondays-43c227a7.html>

# AWS DeepLens

# AWS DeepLens

## Video Playback Disabled

<https://www.youtube.com/watch?v=RhEVld4GwzU>

# AWS DeepLens - przydatne adresy

- <https://aws.amazon.com/deeplens/>
- <https://aws.amazon.com/deeplens/faqs/>
- <https://docs.aws.amazon.com/deeplens/>
- <https://www.amazon.com/AWS-DeepLens-learning-enabled-developers/dp/B075Y3CK37>

# Amazon Lex

# Amazon Lex

- Usługa umożliwiająca budowanie chatbotów
- Możemy to robić z poziomu GUI lub SDK
- Skaluje się automatycznie i jest zarządzana przez AWS
- Możemy rozmawiać z botem za pomocą poleceń głosowych lub pisanych
- Bazuje na usłudze Alexa (usługa w chmurze) z którą łączą się urządzenia Amazon Echo

# Amazon Lex - przydatne adresy

- <https://aws.amazon.com/lex/>
- <https://aws.amazon.com/lex/pricing/>
- <https://aws.amazon.com/lex/faqs/>
- <https://docs.aws.amazon.com/lex/>

# Amazon Echo Video Playback Disabled

<https://www.youtube.com/watch?v=FQn6aFQwBQU>

# Amazon Lex

- Intent -> zamiar konwersacji z użytkownikiem
- Utterances - wypowiedzi które sugerują powyższy cel
- Lex się domyślny podobnych zwrotów dzięki AI
- Slots - dane pobrane od użytkownika

# Amazon Lex

- Wypowiedzi
  - I would like to buy tickets
  - I would like to get tickets
  - I want to buy tickets
  - I want to get tickets
  - I am interested in concert tickets
- Zapytania
  - Data
  - Zespół
  - Liczba biletów
  - Lokalizacja

# Amazon Machine Learning

# Amazon Machine Learning

- Usługa uczenia maszynowego dla osób nie posiadających umiejętności technicznych
- Dla osób technicznych jest usługa SageMaker
- Umożliwia budowanie prostych modeli i predykcję za pomocą interfejsu użytkownika

# Amazon Machine Learning - przydatne adresy

- <https://aws.amazon.com/ml/>
- <https://aws.amazon.com/ml/pricing/>
- <https://aws.amazon.com/ml/faqs/>
- <https://docs.aws.amazon.com/ml/>

# Amazon Machine Learning

- <https://s3.amazonaws.com/aml-sample-data/banking.csv>
- <https://s3.amazonaws.com/aml-sample-data/banking-batch.csv>

# Amazon Polly

# Amazon Polly

- Usługa konwersji tekstu na mowę
- Obsługa wielu języków, także polski
- Wiele lektorów
- Możliwość definiowania dodatkowych efektów za pomocą SSML
- Korzystanie z www, CLI i API
- Eksport do S3

# Amazon Polly - przydatne adresy

- <https://aws.amazon.com/polly/>
- <https://aws.amazon.com/polly/pricing/>
- <https://aws.amazon.com/polly/faqs/>
- <https://docs.aws.amazon.com/polly/>
- <https://docs.aws.amazon.com/polly/latest/dg/supported-ssml.html>

# Amazon Polly

W Szczebrzeszynie chrząszcz brzmi w trzcinie  
I Szczebrzeszyn z tego słynie.  
Wół go pyta: "Panie chrząszczu,  
Po co pan tak brzęczy w gąszczu?"  
"Jak to - po co? To jest praca,  
Każda praca się opłaca."

"A cóż za to Pan dostaje?"  
"Też pytanie! Wszystkie gaje,  
Wszystkie trzcinę po wsze czasy,  
Łąki, pola oraz lasy,  
Nawet rzeczki, nawet zdroje,  
Wszystko to jest właśnie moje!"  
- Jan Brzechwa "CHRZĄSZCZ"

<https://www.tenpieknyswiat.pl/2007/05/09/w-szczebrzeszynie-chrzaszcz-brzmi-w-trzcinie>

# Amazon Polly

```
<speak>Hi! My name is <lang xml:lang="pl-PL">Radosław Szmit</lang>. I will read any text you type here.</speak>
```

# Amazon Polly

```
<speak>
He was caught up in the game.<break time="1s"/>
In the middle of the 10/3/2014
<sub alias="World Wide Web Consortium">W3C</sub>
meeting, he shouted, "Score!" quite loudly.
When his boss stared at him, he repeated
<amazon:effect name="whispered">"Score"</amazon:effect> in a whisper.
</speak>
```

# Amazon Polly

```
<speak>
    <amazon:breath duration="long" volume="x-loud"/>
    <prosody rate="120%"> <prosody volume="loud">
        Wow! <amazon:breath duration="long" volume="loud"/>
    </prosody> That was quite fast <amazon:breath
        duration="medium" volume="x-loud"/>.
    I almost beat my personal best time on this track. </prosody>
</speak>
```

# Amazon Polly

```
<speak>
    <say-as interpret-as="spell-out">Amazon Web Services</say-as>
</speak>
```

# Amazon Polly

```
aws polly start-speech-synthesis-task \
--region us-east-2 \
--endpoint-url "https://polly.us-east-2.amazonaws.com/" \
--output-format mp3 \
--output-s3-bucket-name your-bucket-name \
--output-s3-key-prefix optional/prefix/path/file \
--voice-id Joanna \
--text file://text\_file.txt
```

<https://docs.aws.amazon.com/polly/latest/dg/longer-cli.html>

# Amazon Rekognition

# Amazon Rekognition

- Usługa rozpoznawania obiektów na obrazach i filmach
- Moderowanie obrazów
- Wykrywanie twarzy i określanie cech
- Rozpoznawanie sławnych ludzi
- Porównywanie twarzy
- Odczytywanie tekstu (OCR)

# Amazon Rekognition - przydatne adresy

- <https://aws.amazon.com/rekognition/>
- <https://aws.amazon.com/rekognition/pricing/>
- <https://aws.amazon.com/rekognition/faqs/>
- <https://docs.aws.amazon.com/rekognition/>

# Amazon Transcribe

# Amazon Transcribe

- Automatic Speech Recognition
- Narzędzie do zmiany mowy (nagrania) na tekst
- Rozpoznawanie rozmówców (kto co powiedział gdy jest kilka osób)
- Import i eksport z S3
- Generowanie znaczników czasu
- Możliwość przekazywania własnych słów (np. dla nazw własnych)
- Możliwość przesyłania strumienia audio (np. z mikrofonu, kod Java)

# Amazon Transcribe - przydatne adresy

- <https://aws.amazon.com/transcribe/>
- <https://aws.amazon.com/transcribe/pricing/>
- <https://aws.amazon.com/transcribe/faqs/>
- <https://docs.aws.amazon.com/transcribe/>

# Amazon Translate

# Amazon Translate

- Narzędzie do automatycznego tłumaczenia tekstów pomiędzy różnymi językami
- Obsługa 21 języków
- Korzystanie z www, CLI i API

# Amazon Translate - przydatne adresy

- <https://aws.amazon.com/translate/>
- <https://aws.amazon.com/translate/features/>
- <https://aws.amazon.com/translate/resources/>
- <https://aws.amazon.com/translate/pricing/>
- <https://aws.amazon.com/translate/faqs/>
- <https://docs.aws.amazon.com/translate>
- [\*\*https://docs.aws.amazon.com/translate/latest/dg/what-is.html\*\*](https://docs.aws.amazon.com/translate/latest/dg/what-is.html)

# Amazon Translate

Była sobie raz mała słodka dziewczynka, którą każdy pokochał, kto ją tylko zobaczył, a najbardziej kochała ją babcia, która nie wiedziała wprost, co jeszcze jej dać. Pewnego razu podarowała jej kapturek z czerwonego aksamitu, a ponieważ bardzo ładnie jej leżał, nie chciała nosić niczego innego. Odtąd nazywano ją więc Czerwonym Kapturkiem. Pewnego dnia matka rzekła do dziewczynki. "Chodź, Czerwony Kapturku, masz tu kawałek placka i butelkę wina. Zanieś to babci, bo słaba jest i niedomaga. Babcia bardzo się ucieszy. Ruszaj w drogę nim nastanie upał, a idź ładnie i nie zbaczaj z drogi, bo inaczej sobie kark utracisz i babcia niczego nie dostanie. A gdy wejdziesz do izby, nie zapomnij powiedzieć Dzień Dobry i nie rozglądarka się po wszystkich kątach."

<https://www.grimmmstories.com/language.php?grimm=026&l=pl&r=en>

# Amazon Translate

Once upon a time there was a sweet little girl. Everyone who saw her liked her, but most of all her grandmother, who did not know what to give the child next. Once she gave her a little cap made of red velvet.

Because it suited her so well, and she wanted to wear it all the time, she came to be known as Little Red Riding Hood. One day her mother said to her: "Come Little Red Riding Hood. Here is a piece of cake and a bottle of wine. Take them to your grandmother. She is sick and weak, and they will do her well. Mind your manners and give her my greetings. Behave yourself on the way, and do not leave the path, or you might fall down and break the glass, and then there will be nothing for your sick grandmother."

<https://www.grimmsstories.com/language.php?grimm=026&l=pl&r=en>

# Amazon SageMaker

# Amazon SageMaker

- Usługa uczenia maszynowego wymagająca kompetencji od użytkownika
- Oparty o Jupyter Notebook
- Dużo usług dodatkowych, tworzenie "Endpointów"
- Wbudowane przykłady
- Obsługa wielu znanych bibliotek Open Source (TensorFlow, MXNet, PyTorch, Chainer, Scikit-learn, SparkML, Horovod, Keras, Gluon)
- Marketplace

# Amazon SageMaker - przydatne adresy

- <https://aws.amazon.com/sagemaker/>
- <https://aws.amazon.com/sagemaker/pricing/>
- <https://aws.amazon.com/sagemaker/faqs/>
- <https://docs.aws.amazon.com/sagemaker/>
- <https://github.com/aws/sagemaker-spark>
- <https://github.com/aws/sagemaker-python-sdk>
- <https://github.com/awslabs/amazon-sagemaker-examples>
- <https://github.com/awslabs/amazon-sagemaker-workshop>
- <https://github.com/awslabs/aws-sagemaker-emr-tutorial>

# Amazon SageMaker Neo

## Video Playback Disabled

<https://www.youtube.com/watch?v=RjhrugELYW8>

# Amazon SageMaker Marketplace

## Video Playback Disabled

<https://www.youtube.com/watch?v=rAMg6l5Hp4M>

# Amazon SageMaker Ground Truth

## Video Playback Disabled

<https://www.youtube.com/watch?v=gjiozYXHKc8>

# Amazon SageMaker RL

## Video Playback Disabled

<https://www.youtube.com/watch?v=6skqe2lul34>

# Nowe usługi AI w AWS

# Amazon Personalize

## Video Playback Disabled

<https://www.youtube.com/watch?v=gtArgQrJBzE>

# Amazon Forecast

## Video Playback Disabled

[https://www.youtube.com/watch?v=\\_-2E8iDEqFY](https://www.youtube.com/watch?v=_-2E8iDEqFY)

# Amazon Textract

## Video Playback Disabled

<https://www.youtube.com/watch?v=PHX7q4pMGbo>

# AWS RoboMaker

## Video Playback Disabled

<https://www.youtube.com/watch?v=sjxZAdm1utM>

# AWS DeepRacer

## Video Playback Disabled

<https://www.youtube.com/watch?v=C1iJYz7oijo>

# Amazon Elastic Inference

## Video Playback Disabled

<https://www.youtube.com/watch?v=dZ5FLzOIQFo>

# Serverless w AWS

# Serverless

- Uruchamianie kodu bez pojęcia "serwera", nie interesuje nas to
- podzbiorem Serverless jest Function as a Service (FaaS)
- Wiele usług w AWS jest typu Serverless jak S3, DynamoDB, Glue

# Najpopularniejsze usługi serverless

Narzędzie	Opis
Lambda	Uruchamianie naszego kodu (funkcji)
S3	Obiektowe składowanie danych
SQS	Rozproszona kolejka danych, jeden odbiorca wiadomości (queue)
SNS	Publish / Subscribe, wielu odbiorców (topic)
Step Functions	Serverless workflows
DynamoDB	NoSQL
Kinesis	Event Streaming
Athena	Silnik SQL do S3

# AWS Lambda

# AWS Lambda

- Najpopularniejsza usługa Serverless
- Usługa Stateless
- Funkcja Lambda, Function as a Service
- Jak mikrousługa, kawałek kodu, uruchomiony "bez serwera"
- Nie troszczymy się o zasoby sprzętowe
- AWS odpowiada za **automatyczne skalowanie i deploy** zgodnie z użyciem
- Niezawodność i wysoka dostępność (HA)
- Wbudowany monitoring i logowanie (CloudWatch)

# AWS Lambda - przydatne adresy

- <https://aws.amazon.com/lambda/>
- <https://aws.amazon.com/lambda/pricing/>
- <https://aws.amazon.com/lambda/faqs/>
- <https://docs.aws.amazon.com/lambda/>

# AWS Lambda - zastosowania

- Web
- Data processing
- IoT
- Mobile
- Aplikacje AI
- inne...

# AWS Lambda

- Node.js 8.10, Node.js 6.10
- Python 3.6, Python 3.7, Python 2.7
- Ruby 2.5
- Java 8
- Go 1.x
- .NET Core 2.1, .NET Core 2.0, .NET Core 1.0
- <https://docs.aws.amazon.com/lambda/latest/dg/lambdaruntimes.html>

# AWS Lambda - Jak uruchomić?

- SDK
- RESTful Web Service
- Inna Funkcja Lambda
- Alexa, Lex
- Schedule (cron)
- Inne usługi AWS (np monitoring)

<https://docs.aws.amazon.com/lambda/latest/dg/invoking-lambda-functions.html>

# AWS Lambda - ograniczenia

- 128 MB - 3008 MB pamięci RAM (co 64MB)
- Maksymalny czas wykonania to 15 minut (900 sekund)
- Maksymalnie 1000 równoległych uruchomień wszystkich funkcji w regionie
- **Limit równoległych uruchomień można zwiększyć kontaktując się z Support**
- <https://docs.aws.amazon.com/lambda/latest/dg/limits.html>

# AWS Lambda - płatność

- Płacimy tylko za faktyczne działanie (żądania i czas działania, dokładność 100ms)
- **Request** -> Uruchomienie funkcji Lambda
- 1 milion żądań jest za darmo w miesiącu
- 1 **GB-s** -> 1 GB pamięci używany przez sekundę
- 400,000 GB-s w miesiącu za darmo
- <https://aws.amazon.com/lambda/pricing/>

# Bazy danych w AWS

# Bazy danych w AWS

Narzędzie	Opis
Amazon Relational Database Service (Amazon RDS)	Usługa popularnych silników baz danych jak MySQL, PostgreSQL, MariaDB, MS SQL Server, Oracle Database,
RDS - Aurora	Baza relacyjna kompatybilna z MySQL i PostgreSQL dostosowana do pracy w chmurze, SaaS
Redshift	Rozproszona hurtownia danych, DW
DynamoDB	Rozproszona baza klucz wartość, na niej wzorowana jest Cassandra/Scylla
ElastiCache	Baza in-memory kompatybilna z Redis lub Memcached, SaaS
Neptune	Baza grafów
DocumentDB	Baza dokumentowa (jak MongoDB)
Timestream (preview)	Baza szeregów czasowych jako SaaS
Quantum Ledger Database (QLDB) (preview)	Baza kryptograficzna logów ("blockchain")
SimpleDB	Web serwis służący do tworzenia zapytań do danych strukturalnych w S3 (niedostępny w konsoli www)

# Wyszukiwanie danych

Narzędzie	Opis
CloudSearch	W pełni zarządzalna usługa wyszukiwania danych
Elasticsearch Service	Elastic Stack zarządzany przez AWS

# Odpowiedniki popularnych rozwiązań w AWS

On premise	AWS
HDFS (+ Ozone)	S3
SQL database *	RDS *
Warehouse database	Redshift
MongoDB	DocumentDB
Cassandra	DynamoDB
HBase	HBase (EMR)
Neo4J	Neptune
Redis / Memcached	ElastiCache
Druid	Timestream
Elasticsearch	CloudSearch / Elasticsearch

# Amazon Timestream

## Video Playback Disabled

<https://www.youtube.com/watch?v=oTPplyXoE3k>

# Amazon Quantum Ledger Database (QLDB)

## Video Playback Disabled

<https://www.youtube.com/watch?v=q002jbUuuSY>

# Data Transformation w AWS

# Data Transformation

- ETL (Extract, Transform, Load)
- ELT (Extract, Load, Transform)

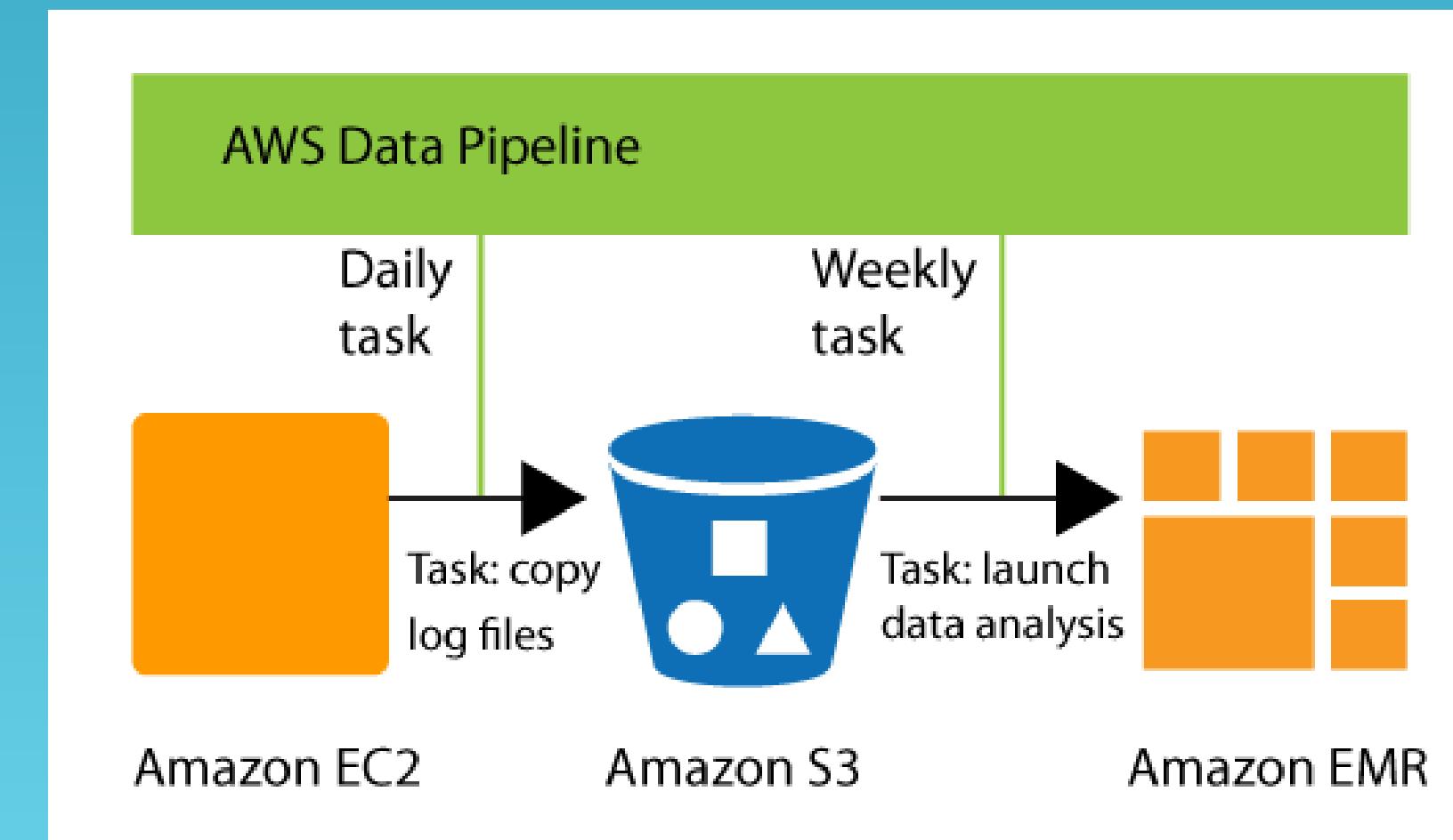
# Data Transformation

- AWS Data Pipeline
- AWS Glue

# AWS Data Pipeline

# AWS Data Pipeline

- Usługa pomagająca automatyzację procesu przesyłania i przetwarzania danych
- Można używać WWW, CLI, SDK, API
- Składowanie danych w: S3, RDS, Redshift, DynamoDB i inne
- Przetwarzanie: EC2, EMR



# AWS Data Pipeline - przydatne adresy

- <https://aws.amazon.com/datipeline/>
- <https://aws.amazon.com/datipeline/pricing/>
- <https://aws.amazon.com/datipeline/faqs/>
- <https://docs.aws.amazon.com/data-pipeline/>

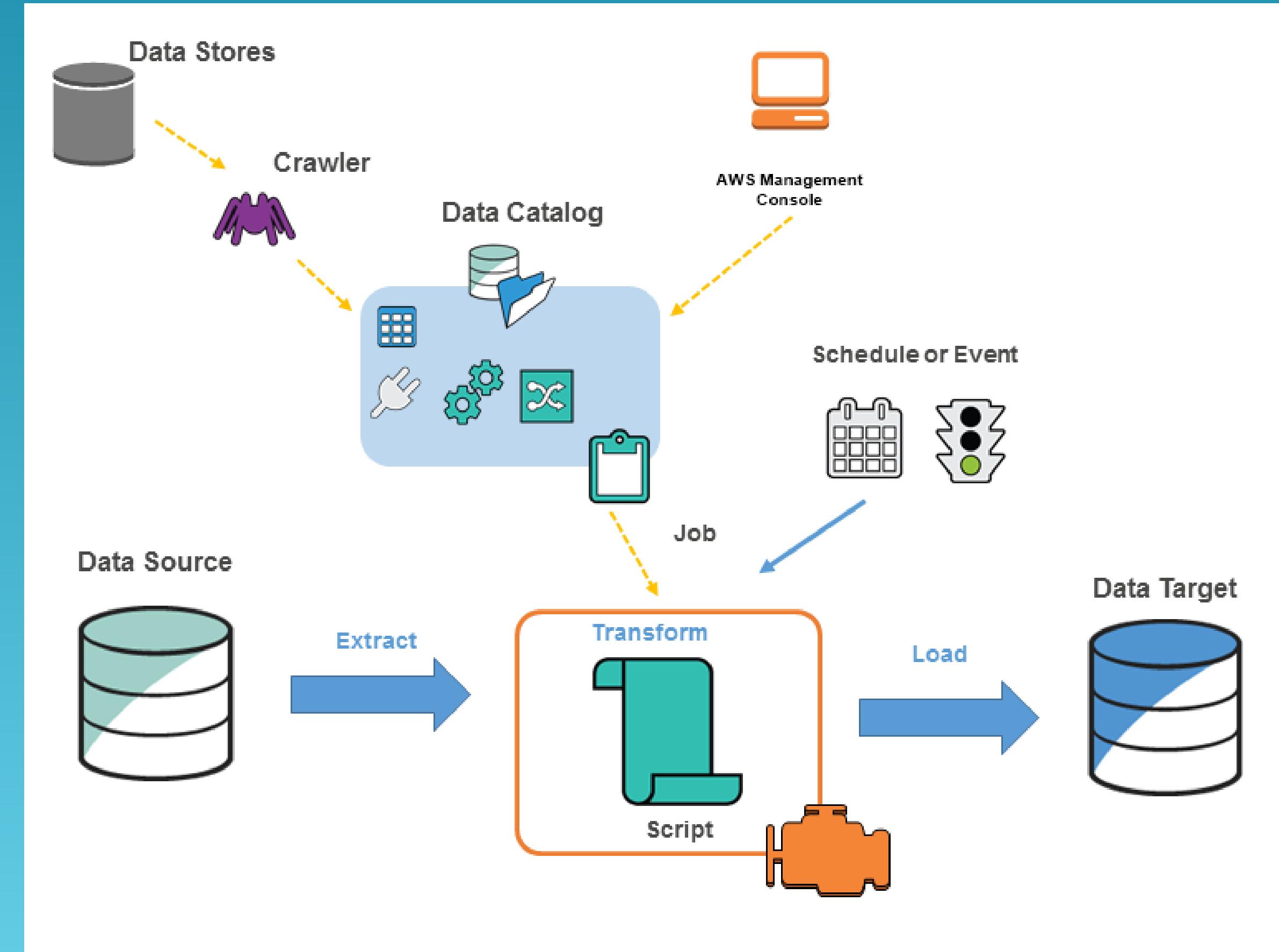
# AWS Glue

# AWS Glue

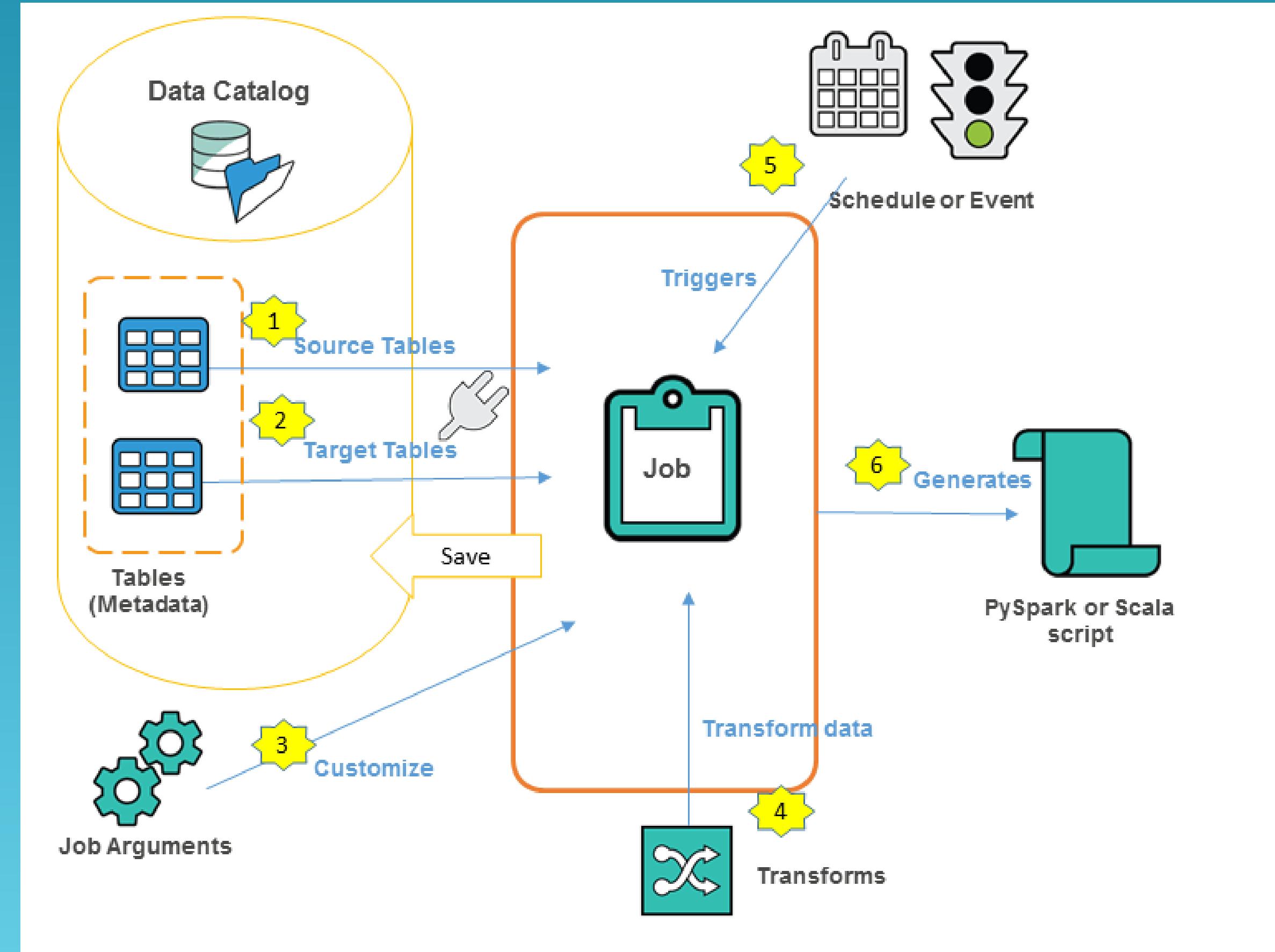
- Usługa **serverless** umożliwiająca realizację procesów ETL
- Bazuje na usłudze Apache Spark
- Wspiera język Python i Scala
- Automatycznie generuje kod (opcja, można modyfikować)
- Obsługuje wiele wbudowanych transformacji
- Integracja z innymi narzędziami AWS i źródłami danych

# AWS Glue - składowe

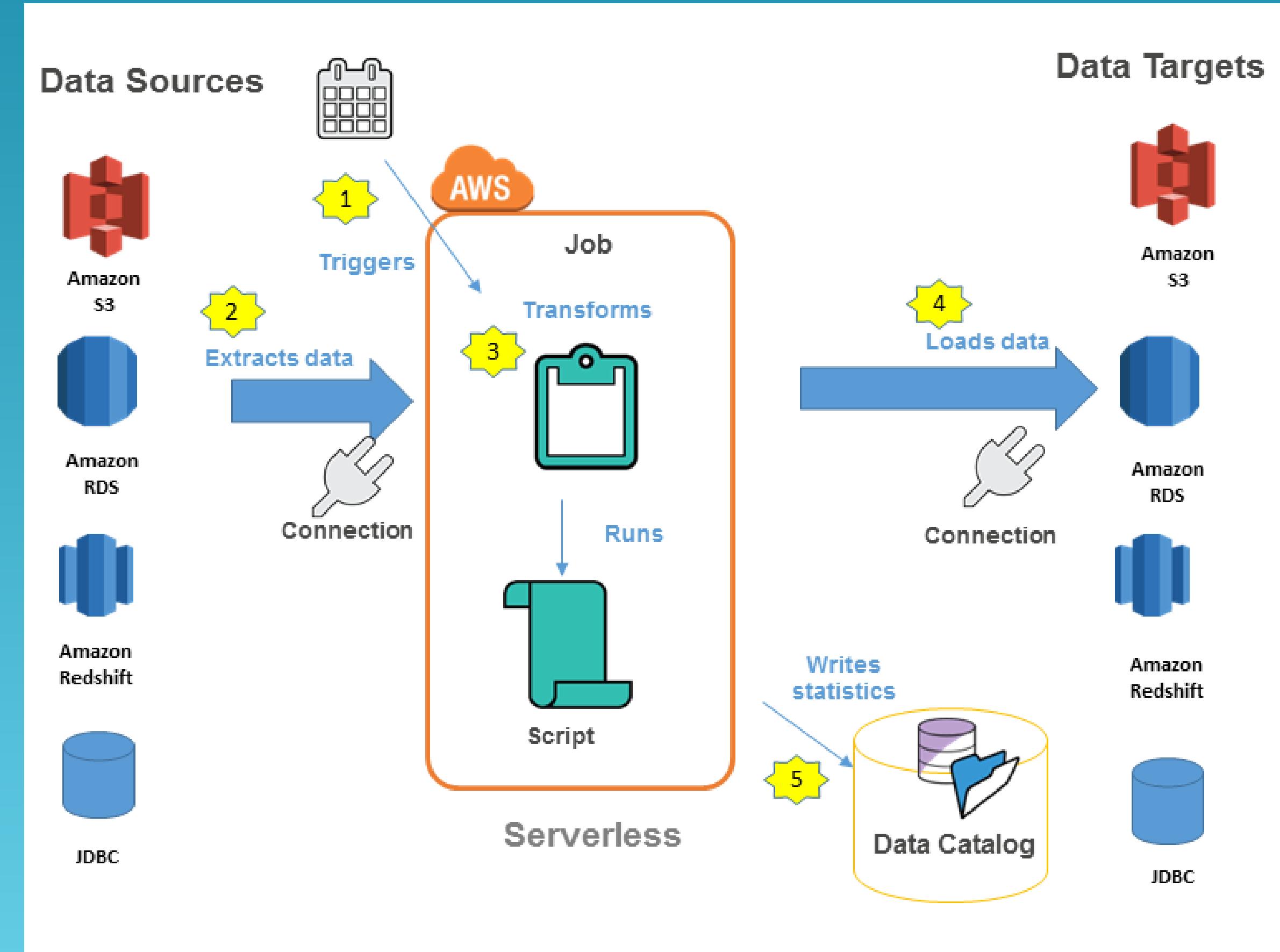
- Data Catalog - Składa metadane, odpowiednik Hive metastore
- Crawler - analizuje dane i generuje metadane
- Job - silnik ETL oparty o Spark



<https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html>



<https://docs.aws.amazon.com/glue/latest/dg/author-job.html>



<https://docs.aws.amazon.com/glue/latest/dg/monitor-glue.html>

# Glue - dane

- s3://elasticmapreduce/samples/hive-ads/tables/
- s3://athena-examples-us-east-1/flight/
- s3://gdelt-open-data/
- s3://awsglue-datasets/examples/

# AWS glue - przydatne adresy

- <https://aws.amazon.com/glue/>
- <https://aws.amazon.com/glue/pricing/>
- <https://aws.amazon.com/glue/faqs/>
- <https://docs.aws.amazon.com/glue/>
- <https://docs.aws.amazon.com/athena/latest/ug/glue-best-practices.html#schema-csv-quotes>
- [https://docs.aws.amazon.com/glue/latest/dg/create-an-iam-role-notebook.html?icmpid=docs\\_glue\\_console](https://docs.aws.amazon.com/glue/latest/dg/create-an-iam-role-notebook.html?icmpid=docs_glue_console)
- [https://docs.aws.amazon.com/glue/latest/dg/create-an-iam-role.html?icmpid=docs\\_glue\\_console](https://docs.aws.amazon.com/glue/latest/dg/create-an-iam-role.html?icmpid=docs_glue_console)

# Integracja w AWS

# Integracja w AWS

Narzędzie	Opis
Simple Queue Service (SQS)	Rozproszona kolejka danych, jeden odbiorca wiadomości (queue)
Simple Notification Service (SNS)	Publish / Subscribe, wielu odbiorców (topic)
Step Functions	Serverless workflows
Amazon MQ	Apache ActiveMQ
Simple Workflow Service (SWF)	Workflow

# Przetwarzanie strumieniowe

w AWS

# Managed Streaming for Kafka (MSK)

- W pełni zarządzalny przez AWS klaster Apache Kafka
- Niezwykle szybka w zapisie oraz odczycie
- Kafka Streams do przetwarzania strumieniowego
- Integracja z wieloma narzędziami (Kafka Connect)
- Używany przez wiele firm z całego świata
- Replikacja pomiędzy wiele DC
- Płatność za czas maszyn

# AWS Kinesis

- Usługa serverless analogiczna do Apache Kafka
- Dane strumieniowe zbierane są przez Kinesis Data Streams (KDS)
- Moduł analizy danych (Kinesis Data Analytics)
- Integracja z wieloma wyjściami na dane w AWS (Kinesis Data Firehose)
- Wsparcie dla transmisji video (Kinesis Video Streams)

# Przegląd innych narzędzi AWS

# AWS - narzędzia programisty

- <https://aws.amazon.com/developer/>
- <https://aws.amazon.com/getting-started/tools-sdks/>
- <https://aws.amazon.com/products/developer-tools/>

# AWS - dla programisty

- <https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/>
- <https://github.com/aws-samples/>
- <https://github.com/aws>
- <https://github.com/aws/aws-sdk-java>
- <https://github.com/boto/boto3>
- <https://github.com/awslabs>

# Microsoft Azure



# Azure

- Młoda chmura obliczeniowa (start 1 lutego 2010)
- Wsparcie dla klientów i technologii Microsoft
- Zalecana zwłaszcza dla firm będącymi partnerami firmy Microsoft (w przeciwieństwie do Google/AWS wspiera bardzo mocno systemy Windows oraz technologie Microsoft jak .Net/C#)
- Szybko zyskuje na popularności, także dzięki otwarciu mocnemu firmy Microsoft na technologie Open Source (Linux, Java, Python, Hadoop, Spark, etc.)

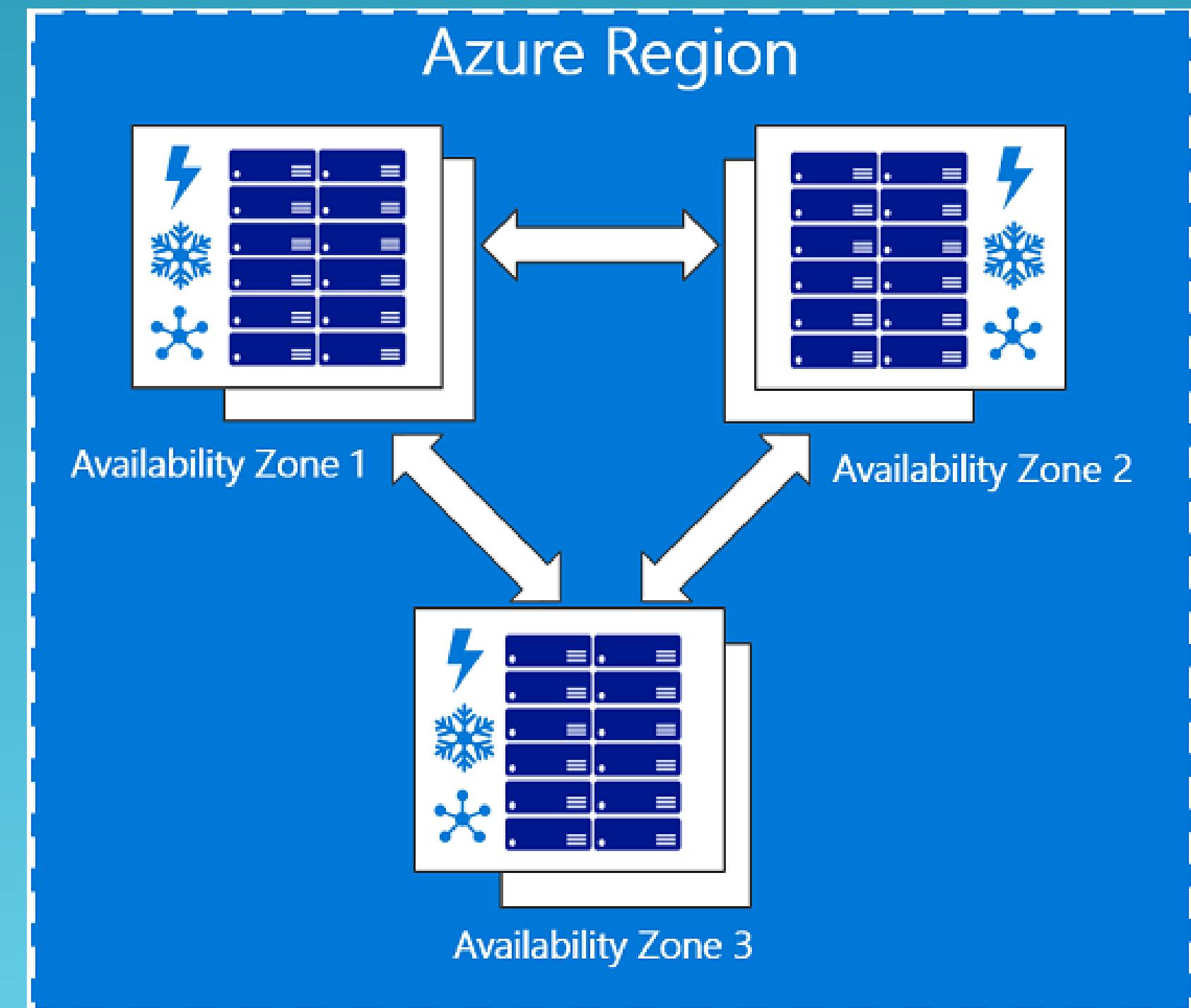
# Azure regiony

**54** — regiony  
— cały świat      **140** dostępne w  
140 krajach



<https://azure.microsoft.com/pl-pl/global-infrastructure/regions/>

# Azure regiony



# Azure regiony

- Chmura Microsoft Azure jest aktualnie dostępna w:
  - Europie Północnej (Irlandia)
  - Europie Zachodniej (Holandia)
  - Zachodnim Zjednoczonym Królestwie (Cardiff)
  - Południowym Zjednoczonym królestwie (Londyn)
  - Niemczech środkowych (Frankfurt)
  - Niemczech Północno-Wschodniej (Magdeburg)
- Planowane są lokalizacje w Europie:
  - Francja Środkowa (Paryż)
  - Francja Południowa (Marsylia)
- Dla usługi HDInsight dostępny jest region Europy Zachodniej oraz Europy Północnej oraz w mniejszym zakresie w Niemczech i Wielkiej Brytanii.

# Azure regiony

- Regiony obsługujące Availability Zones
  - East US 2
  - US Central
  - West Europe
  - France Central
- Serwisy obsługujące Availability Zones
  - Linux Virtual Machines
  - Windows Virtual Machines
  - Virtual Machine Scale Sets
  - Managed Disks
  - Load Balancer
  - Public IP address
  - Zone-Redundant Storage

# Azure regiony

Produkty	EUROPA PÓŁNOCNA	EUROPA ZACHODNIA	NIEMCY — INNE NIŽ REGIONALNE	NIEMCY ŚRODKOWE	NIEMCY PÓŁNOCNO- WSCHODNIE	ZACHODNIE ZJEDNOCZONE KRÓlestwo	POŁUDNIOWE ZJEDNOCZONE KRÓlestwo
HDInsight	●	●		●	●	●	●
HDInsight Linux	●	●		●	●	●	●
HDInsight Windows	●	●					
Stream Analytics	●	●		●	●	●	●
SQL Data Warehouse	●	●		●	●	●	●
Data Factory V2			●				
Przenoszenie i wysyłanie danych	●	●					●
SSIS Integration Runtime	●	●					
Data Factory	●	●					
Przenoszenie danych	●	●					●
Log Analytics			●				●
Data Catalog	●	●					
Data Lake Store	●						
Data Lake Analytics	●						

# Azure Data Lake

- Usługa przechowywania danych (bazuje na HDFS)
- Możliwość analityki danych (Data Lake Analytics)

# Azure HDInsight

- Usługa Big Data w chmurze Microsoft
- Oparta o dystrybucję Hortonworks
- HDInsight 3.6 = Hortonworks Data Platform 2.6
- Wcześniej tworzona własna dystrybucja oparta także o Windows, jednak twórcy wycofali się z tych prac, obecnie od wersji 3.4 tylko wspiera Linux'a
- Możliwość bezpośredniej pracy na Azure Data Lake

# Azure - serwisy

- Virtual Machines - maszyny wirtualne
- Functions - Serverless
- Container Service (AKS) - Kubernetes, CS/OS, Swarm
- Container Instances - kontenery bez orchestracji
- Batch - zadania wsadowe do przetworzenia

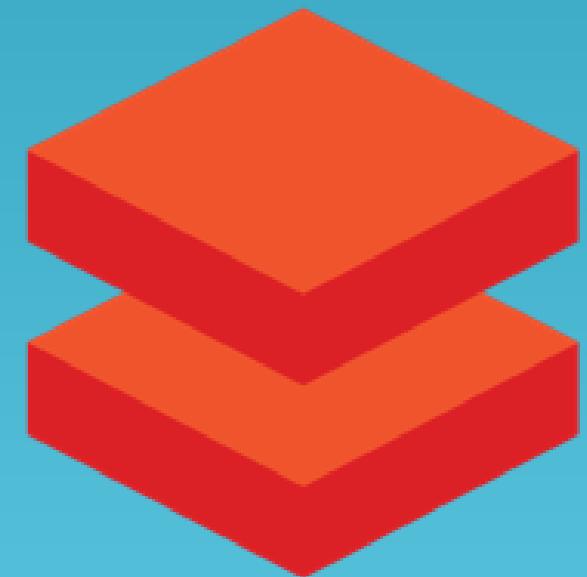
# Azure - serwisy

- Azure Databricks - Apache Spark w chmurze, dostarczany przez twórców narzędzia, firmę Databricks
- Machine Learning
- Data Factory - data flow, ETL
- SQL Data Warehouse - BI
- Event Hubs - strumieniowanie danych
- Stream Analytics - analizowanie strumieni danych
- Analysis Services - modelowanie danych w chmurze

# Azure - serwisy

- Cosmos DB - baza NoSQL
- SQL Database - baza relacyjna
- Database for MySQL / PostgreSQL
- Microsoft SQL Server + SQL Server Stretch Database
- Redis Cache
- Azure Search

# Databricks Unified Analytics Platform



databricks®

# Wprowadzenie do Databricks Platform

"I think that by 2020 most data will be in either public clouds or cloud-like private environments."  
-- Matei Zaharia (2015)

<https://www.kdnuggets.com/2015/05/interview-matei-zaharia-creator-apache-spark.html>

# Cloud Native

- AWS (2015)
- Community (2016)
- Azure (November 15, 2017)
- GCP (future plans)
- No on-premise plans

# Obecne trendy

- Data Lakes
- Data Warehouses
- Streaming
- AI (with ML and DL)

# Potrzeby biznesu

- Data cleansing (data cleaning)
- Fast data pipelines
- Scalable complex processing
- Machine learning

# Rodzaje płatności

- płatność za użycie (domyślnie)
- A Databricks unit ("DBU") is a unit of processing capability per hour, billed on per-second usage.
- płatność z góry na rok lub trzy lata (Databricks Unit pre-purchase plan)
- płatność na podstawie umowy - dodatkowe możliwości i support
- AWS Marketplace - płatność z góry dla dużych wdrożeń

# Architektura Databricks Platform

# Databricks Platform

- Bazuje na Apache Spark
- Dodatkowe biblioteki (np. Apache MXNet)
- Łatwa integracji z narzędziami vendora (AWS S3 / Azure Data Lake)
- Łatwa integracja z narzędziami zewnętrznymi (Snowflake, Tableau)
- Wiele języków
- Wbudowane mechanizmy bezpieczeństwa

# Inne narzędzia bazujące na Sparku

- IBM DB2 Warehouse (Blue engine)
- AWS Glue
- Oracle GoldenGate for Big Data
- Hitachi/Pentaho Data Integration (Kettle)

# Narzędzia

- Notebooks
- Apache Spark clusters
- DBFS

<https://docs.azure.databricks.net/user-guide/databricks-file-system.html>

<https://docs.microsoft.com/pl-pl/azure/data-lake-store/data-lake-store-comparison-with-blob-storage>

# Składowe

- Workspace -> Notebooks
- Databrick Runtime -> Spark engine
- Databricks File System (DBFS)
- Open APIs
- Databricks Enterprise Security

# Workspace type

- Standard & Premium
- Można zmienić później
- Służy do zarządzania Notebookami (Notatniki)

<https://docs.azuredatabricks.net/administration-guide/account-settings/upgrade-downgrade.html>

# Databricks Runtime

DATABRICKS RUNTIME



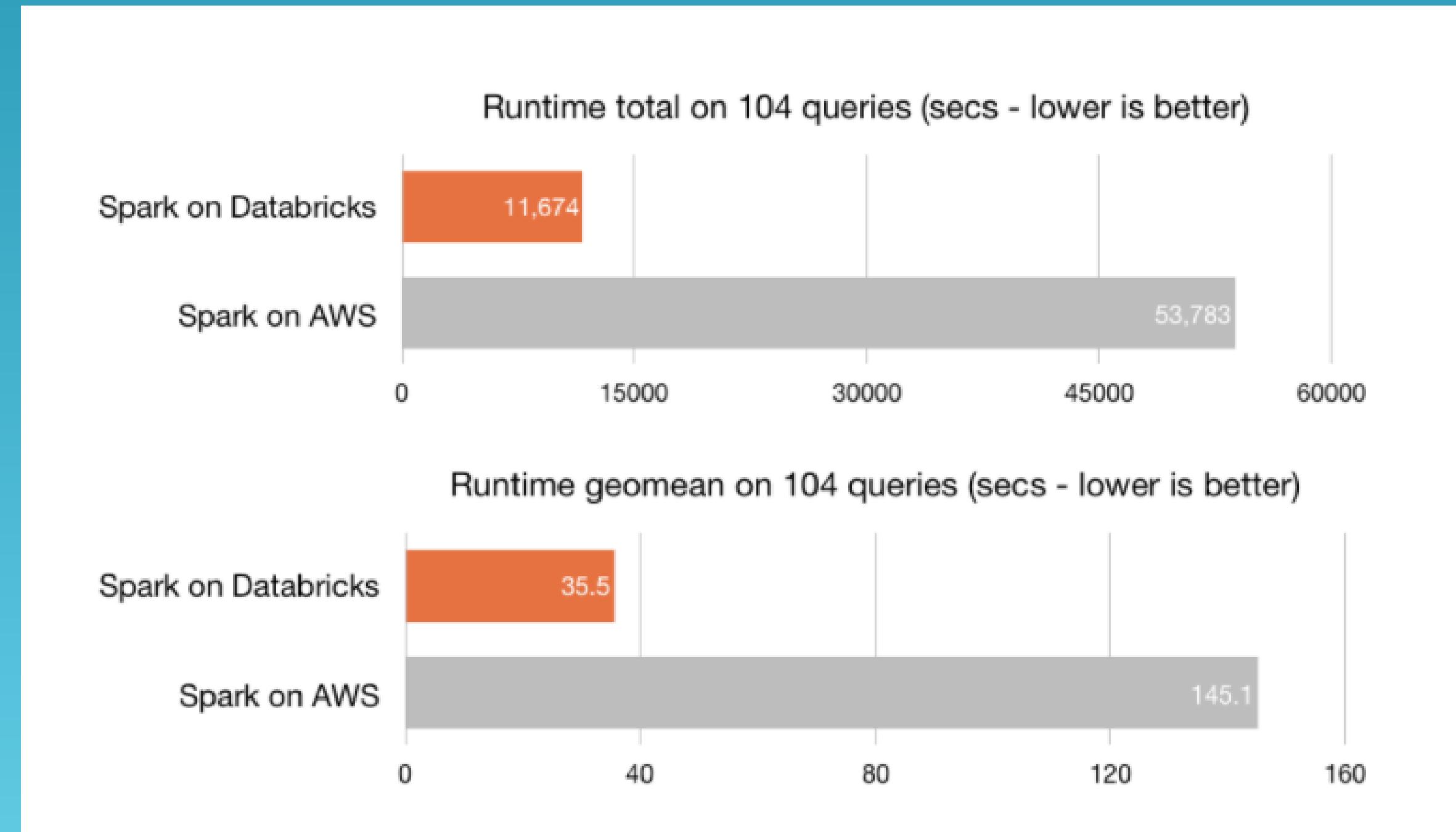
DATABRICKS  
I/O



DATABRICKS  
SERVERLESS

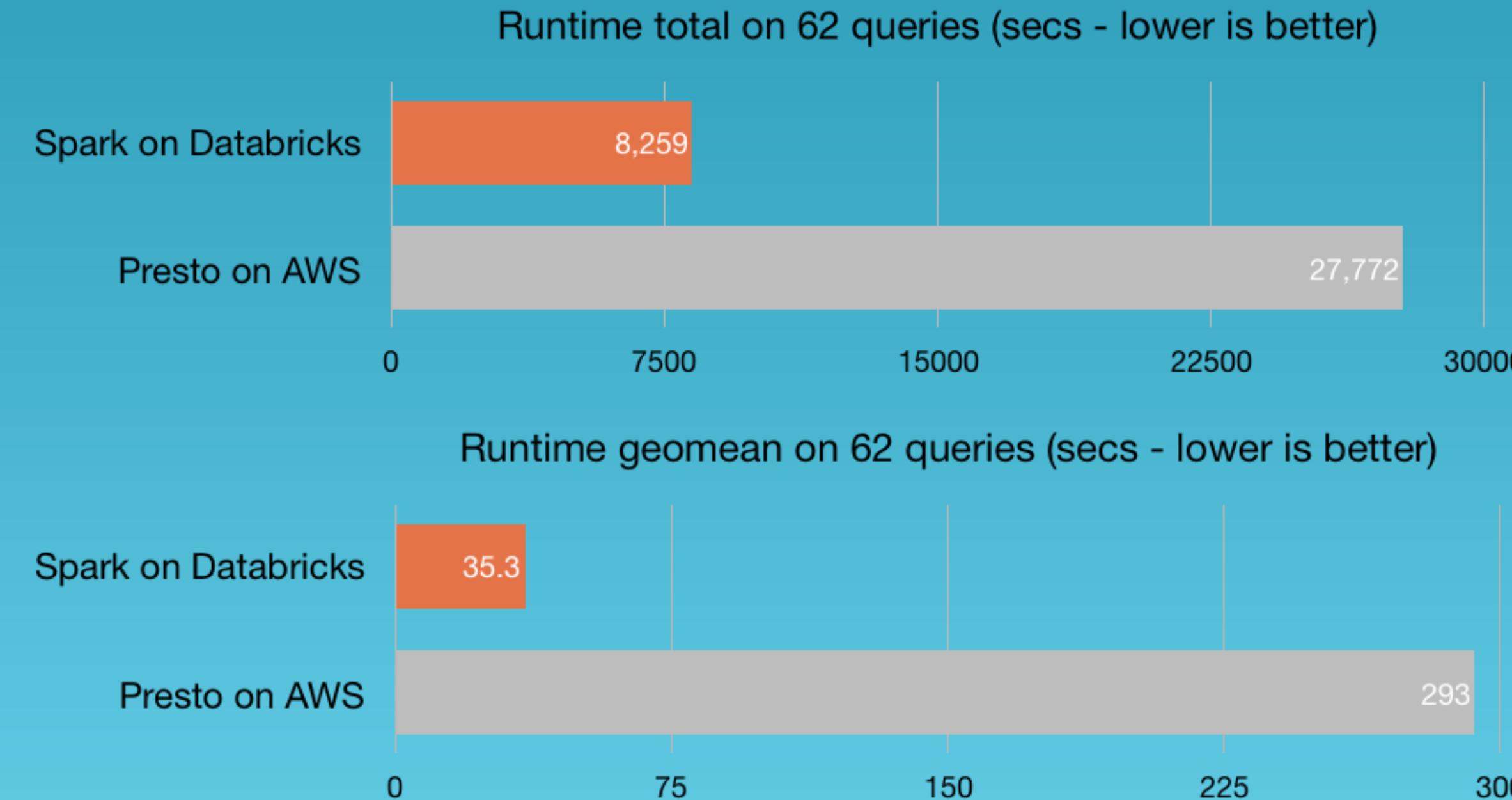
<https://databricks.com/product/databricks-runtime>

# Databricks Runtime



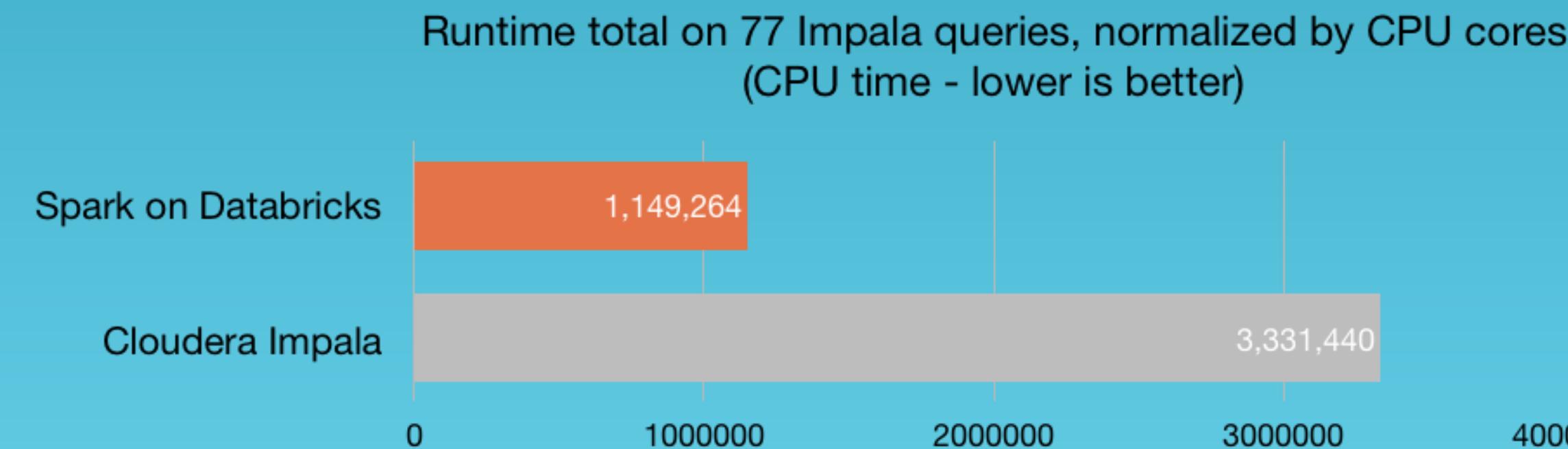
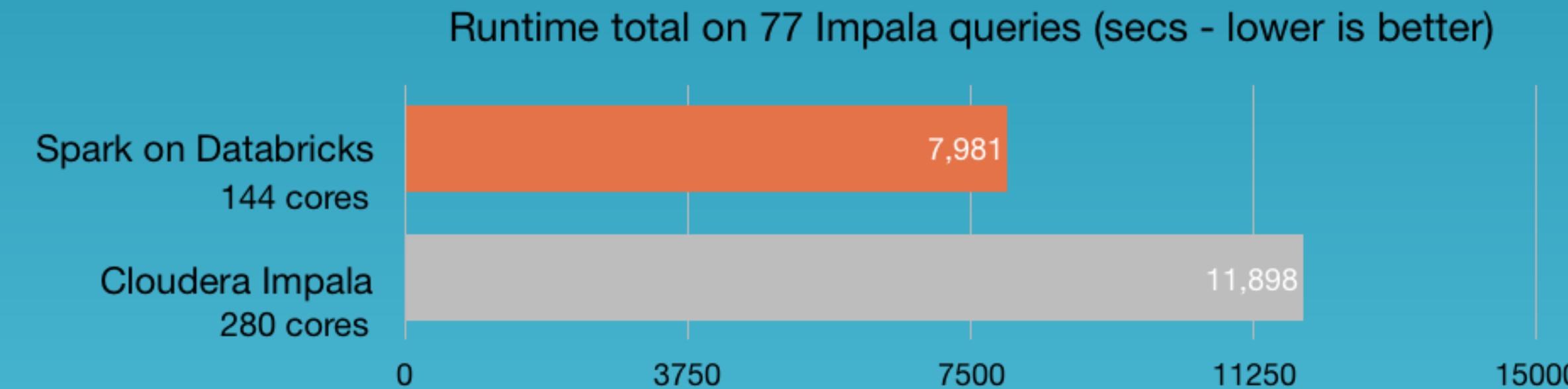
<https://databricks.com/blog/2017/07/12/benchmarking-big-data-sql-platforms-in-the-cloud.html>

# Databricks Runtime



<https://databricks.com/blog/2017/07/12/benchmarking-big-data-sql-platforms-in-the-cloud.html>

# Databricks Runtime



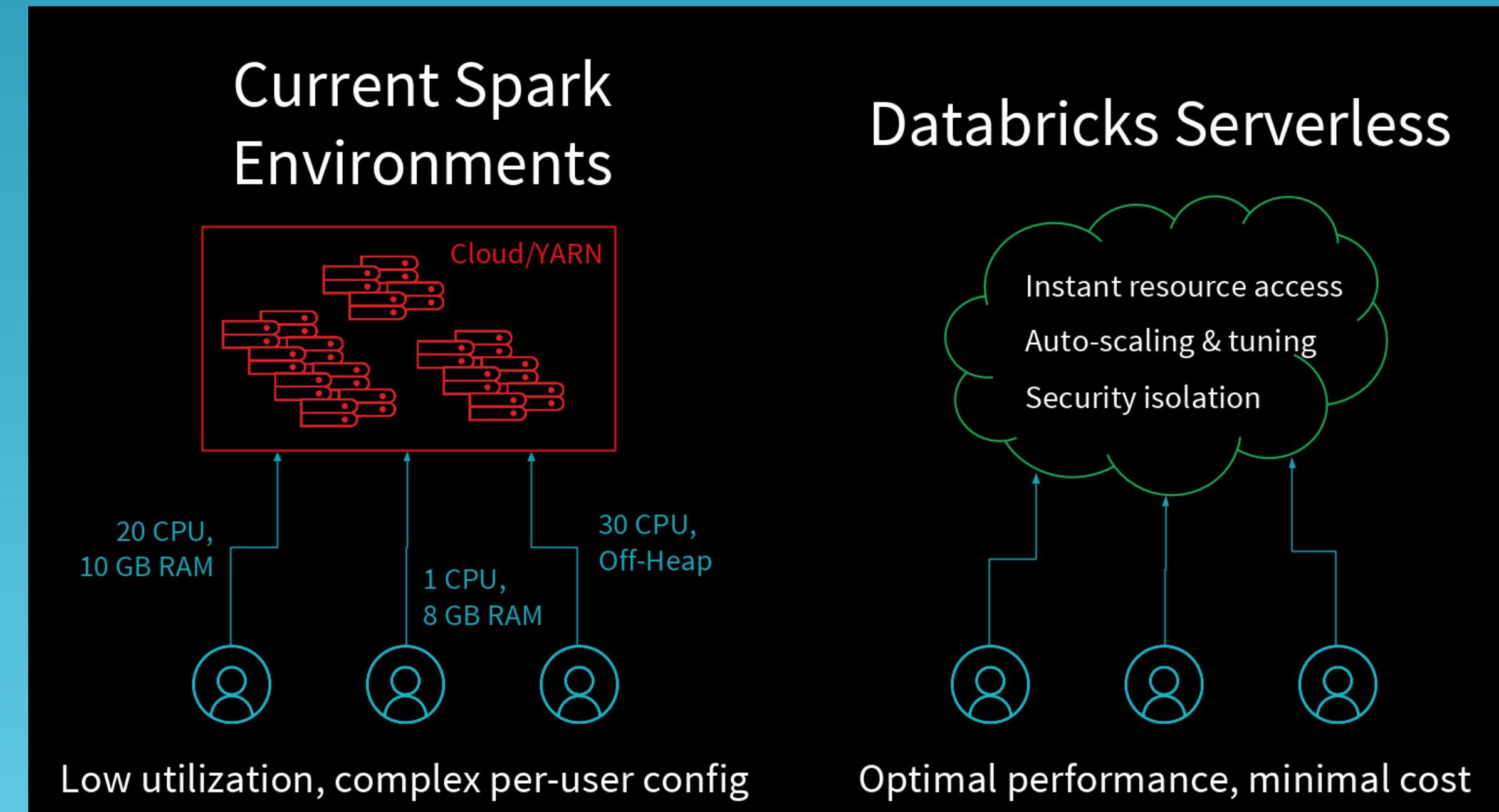
<https://databricks.com/blog/2017/07/12/benchmarking-big-data-sql-platforms-in-the-cloud.html>

# Cluster

- Interactive vs Job clusters
- Standard vs High Concurrency (Serverless)
- GPU klastry wymagają maszyn z kartami GPU (muszą być dostępne w naszym koncie)

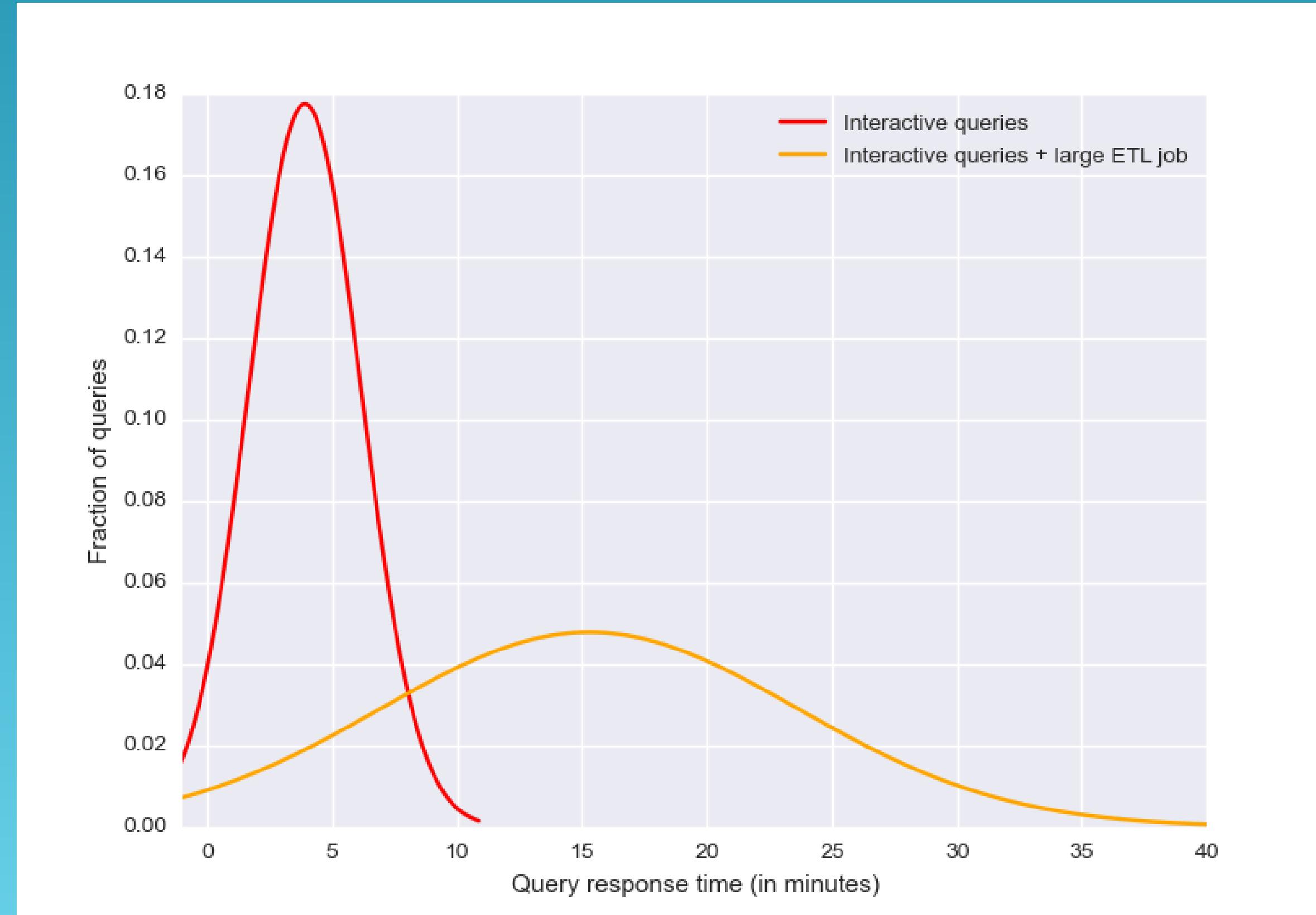
<https://databricks.com/blog/2017/06/07/databricks-serverless-next-generation-resource-management-for-apache-spark.html>

# Serverless Clusters

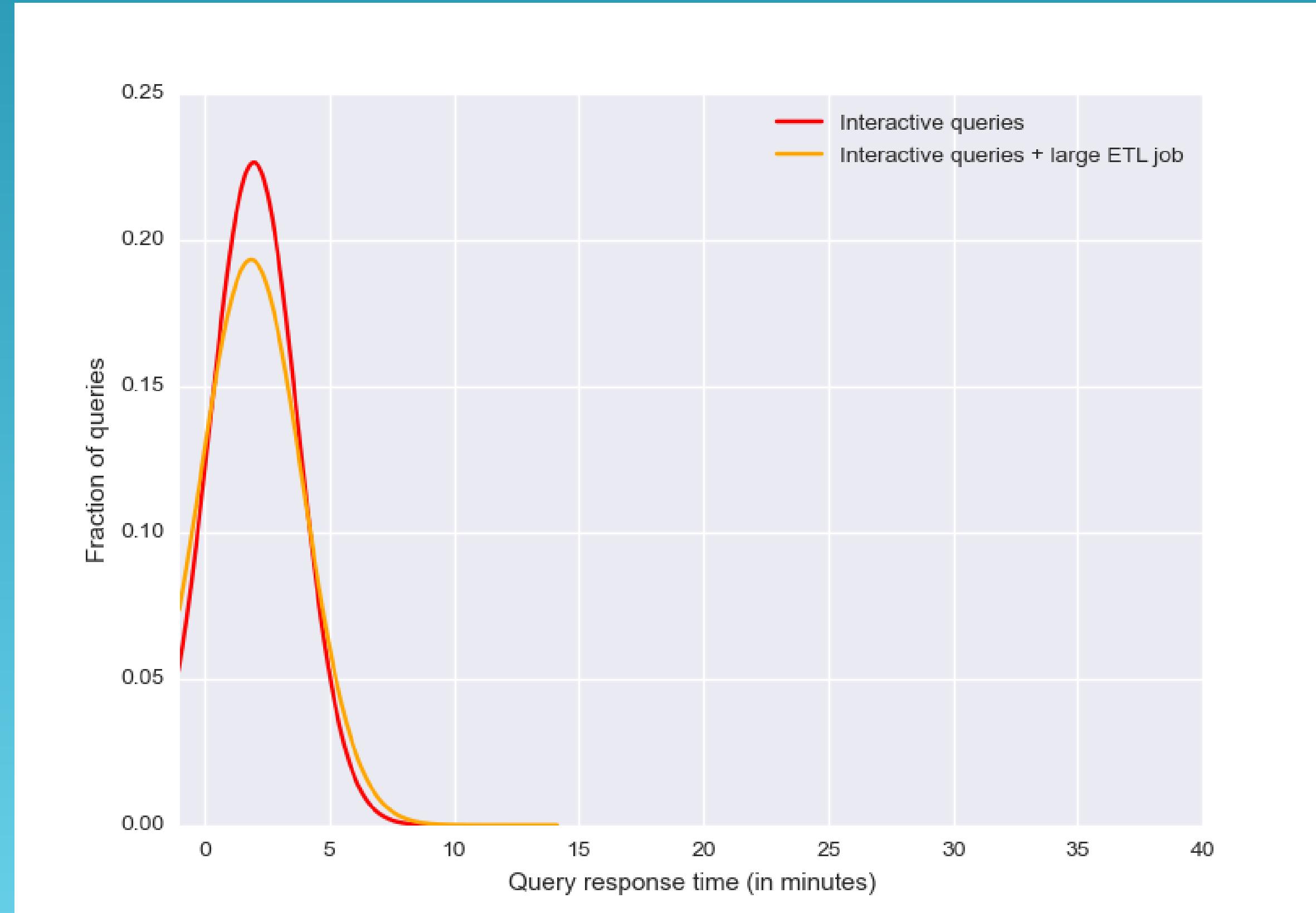


<https://databricks.com/blog/2017/06/07/databricks-serverless-next-generation-resource-management-for-apache-spark.html>

# Serverless Clusters



# Serverless Clusters



# Job

- Możliwość uruchomienia zadania Sparka zgodnie z harmonogramem
  - Odpowiednik spark-submit
  - Odpowiedni "job cluster type"
  - Dostępne API dla zadań

# Databricks File System

- API zgodne z HDFS
- Działa na Azure Blob Store lub AWS S3 (dzięki integracji z HDFS API)
- Wirtualna przestrzeń wspólna dla klastrów
- Możliwość montowania innych zasobów
- Zaleca się pracę na podmontowanym zasobie a nie na DBFS root
- Każdy użytkownik widzi dane z zamontowanego udziału

<https://docs.azuredatabricks.net/user-guide/databricks-file-system.html>

# DBFS root

- /FileStore: Imported data files, generated plots, and uploaded libraries. See The FileStore.
- /databricks-datasets: Sample public datasets.
- /databricks-results: Files generated by downloading the full results of a query.
- /databricks/init: Global and cluster-named init scripts (both deprecated).
- /user/hive/warehouse: Data and metadata for non-external Hive tables.

<https://docs.azuredatabricks.net/user-guide/databricks-file-system.html>

# Dostęp do DBFS

- Databricks CLI
- dbutils
- DBFS API (REST)
- Spark APIs
- Local file APIs (montowanie FUSE w /dbfs na lokalnych dyskach)
- High performance local APIs

<https://docs.azuredatabricks.net/user-guide/databricks-file-system.html>

# High performance local APIs

- Dla procesów ML które wymagają wydajnego dostępu do przestrzeni dyskowej zalecane jest korzystanie z Databricks Runtime 5.3 (lub wyżej) i zapis do folderu dbfs:/ml który jest zmapowany na sterowniku (driver) i węzłach roboczych (worker nodes)

<https://docs.azuredatabricks.net/user-guide/databricks-file-system.html>

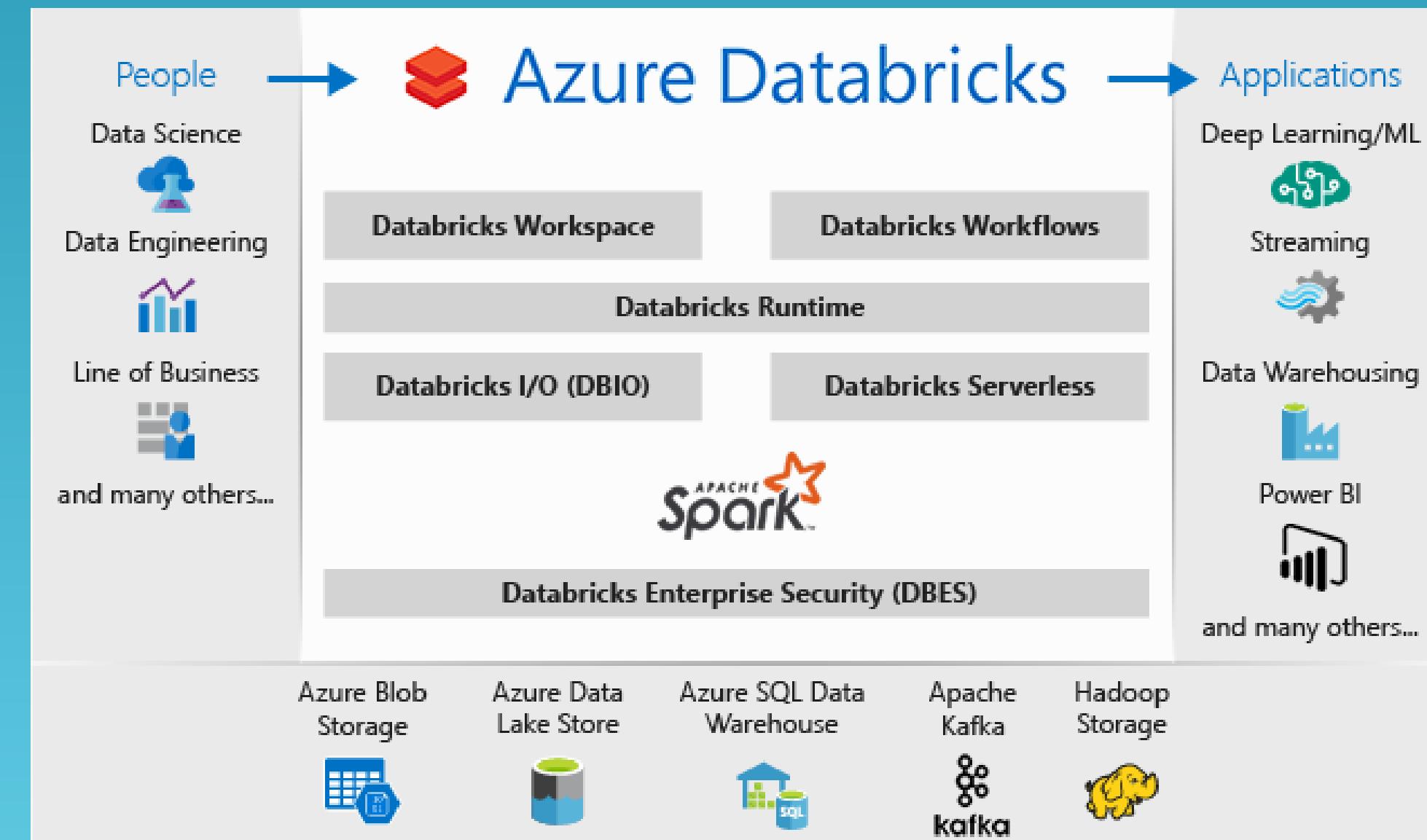
# FileStore

- Specjalny folder gdzie można wrzucać pliki
- Widoczny w WWW
- Widoczne przez funkcje displayHTML

<https://docs.azuredatabricks.net/user-guide/advanced/filestore.html>

# Azure Databricks

# Azure Databricks



<https://docs.microsoft.com/pl-pl/azure/azure-databricks/what-is-azure-databricks>

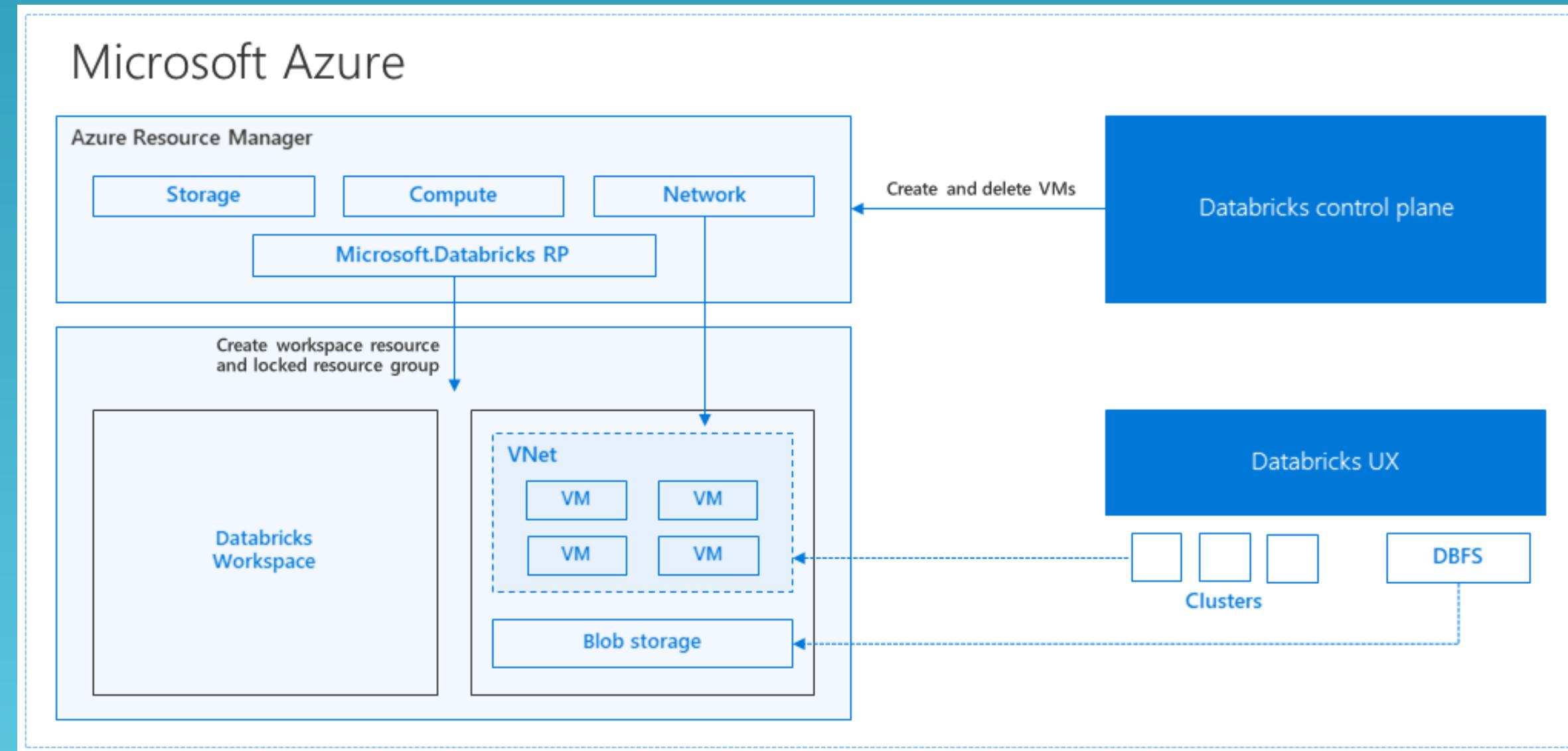
# Azure Databricks

- Efekt współpracy Databricks z Microsoft
- Usługa "pierwszej kategorii" w Azure
- Microsoft "mocno zainwestował" w firmę Databricks"
- Kontrolowane z Azure Portal
- Młodsza ale w 100% zgodna z wersją AWS (pod względem funkcjonalności)

# Integracja

- Active Directory
- SQL Data Warehouse, Cosmos DB, Blob Storage, Power BI, Azure Data Lake, Kafka, Hadoop, HDInsight, etc
- Integracja z innymi narzędziami Azure oraz narzędziami zewnętrznymi działającymi w Azure (np. Snowflake)

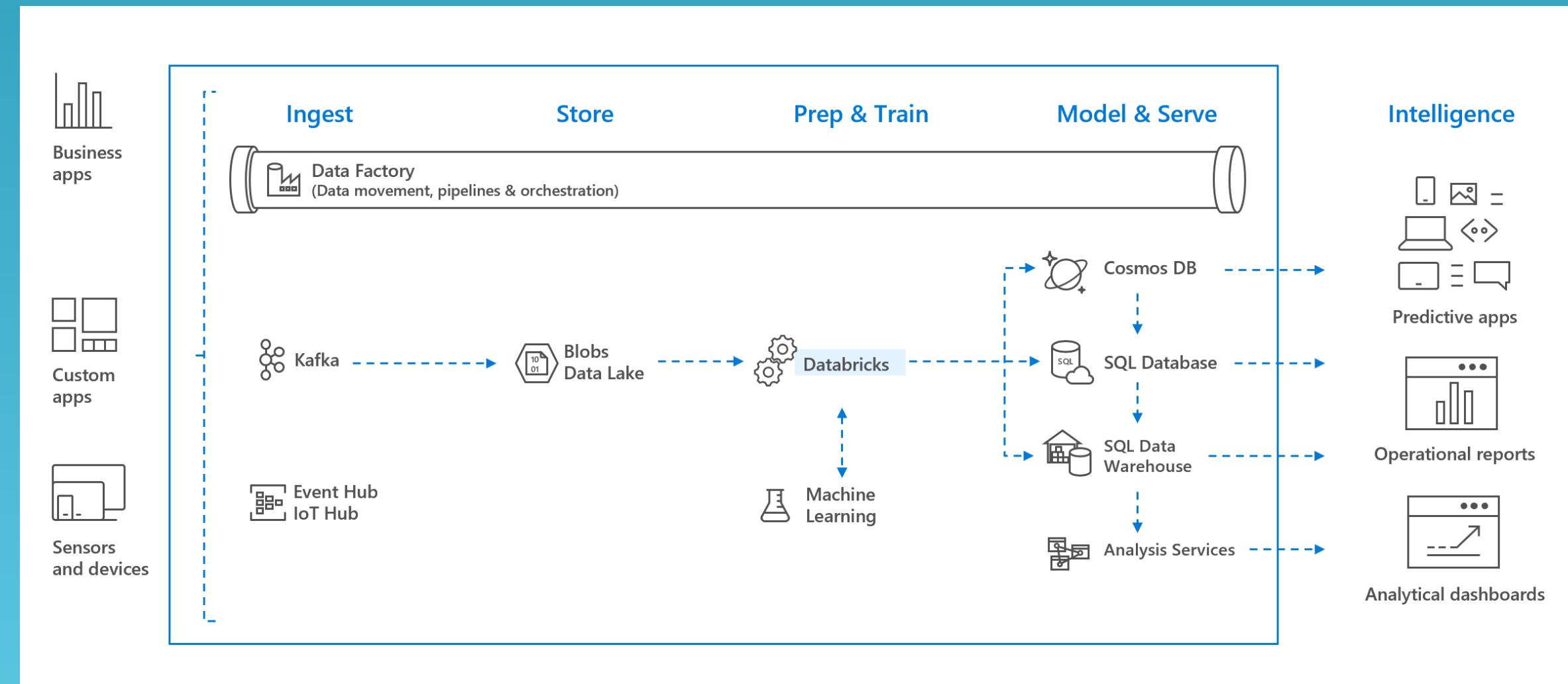
# Azure Databricks



<https://azure.microsoft.com/en-in/blog/a-technical-overview-of-azure-databricks/>

<https://databricks.com/blog/2017/11/15/introducing-azure-databricks.html>

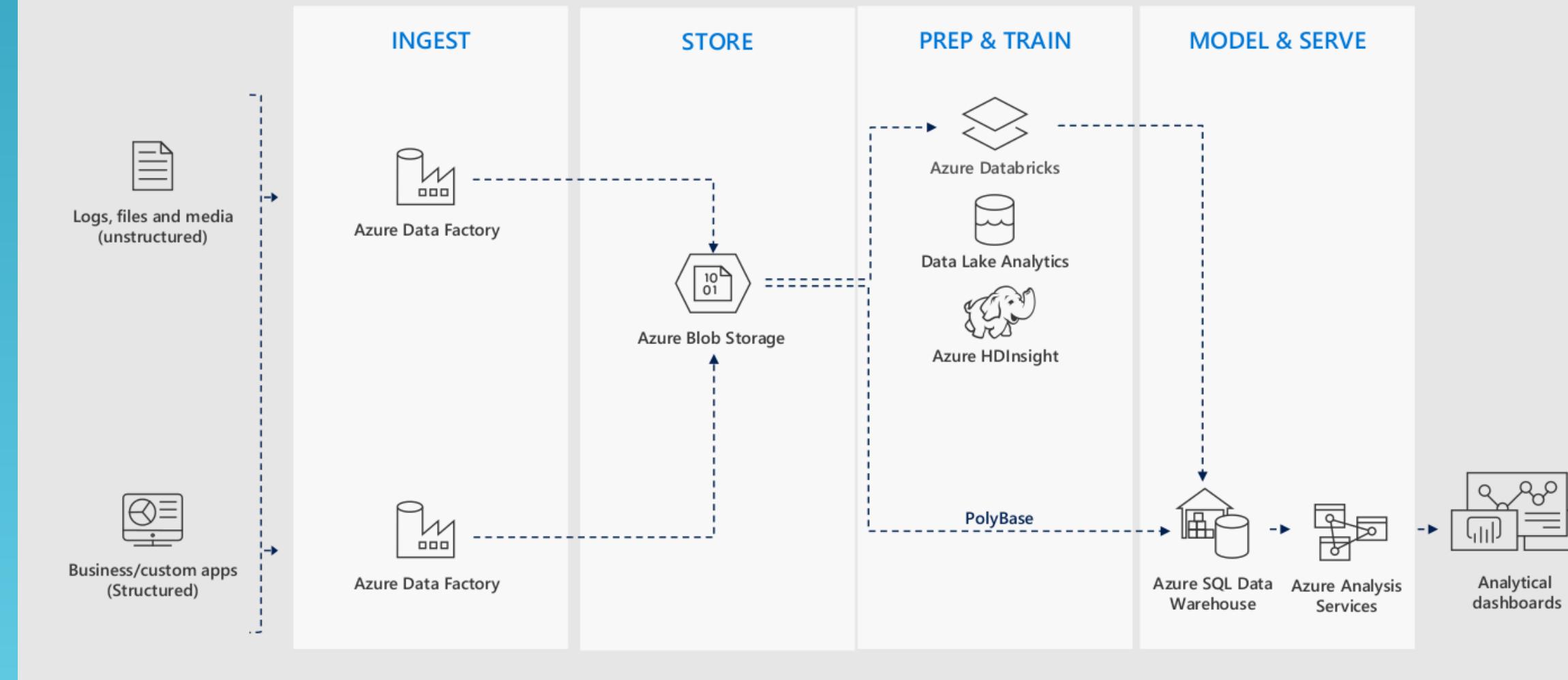
# Azure Databricks



<https://docs.microsoft.com/pl-pl/azure/azure-databricks/what-is-azure-databricks>

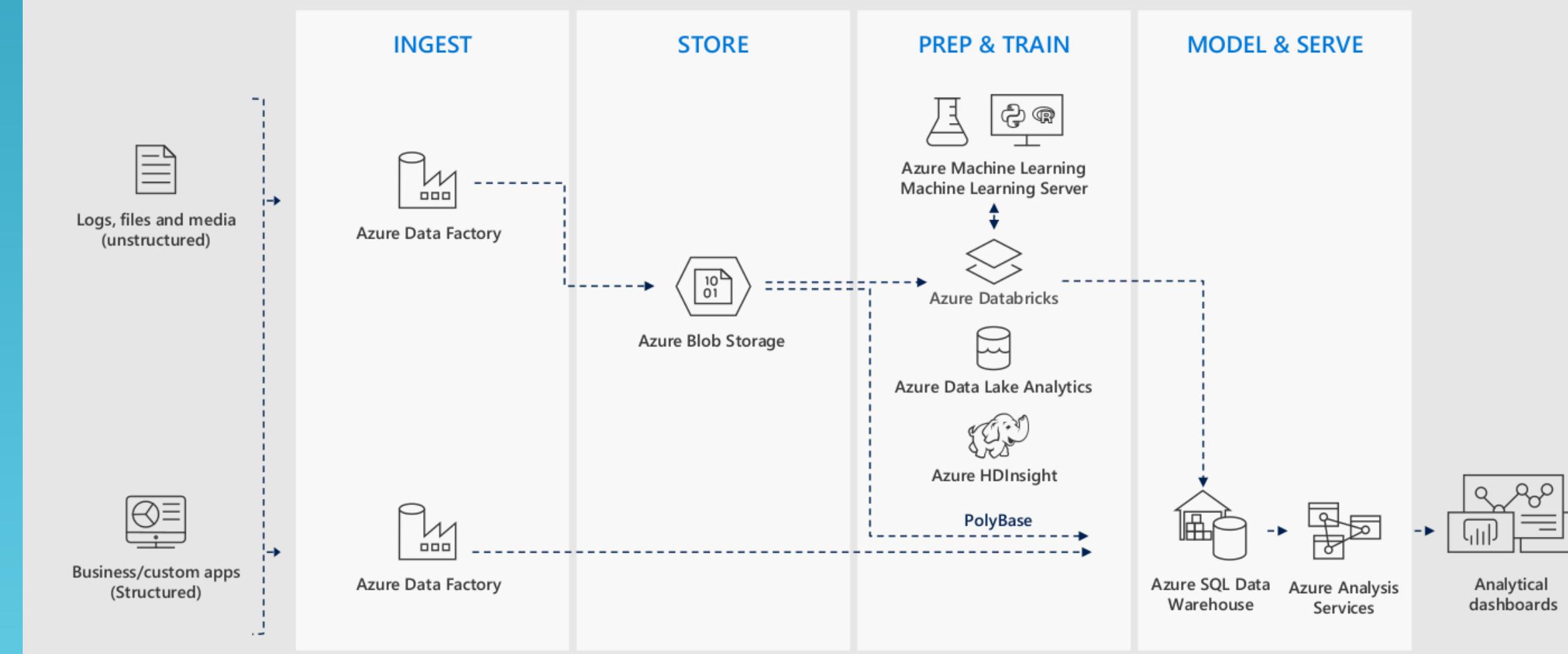
# Azure Databricks

## Modern Data Warehouse

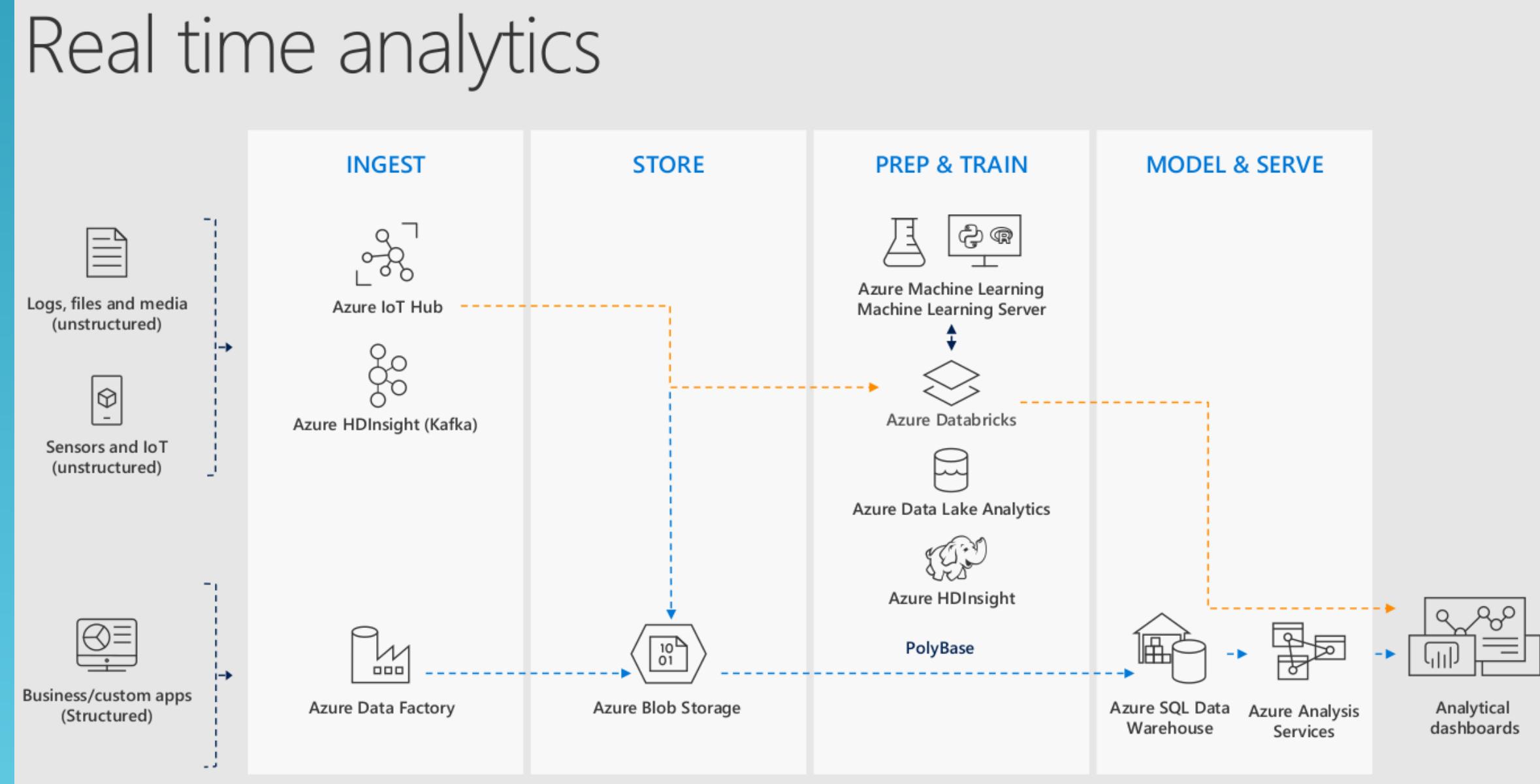


# Azure Databricks

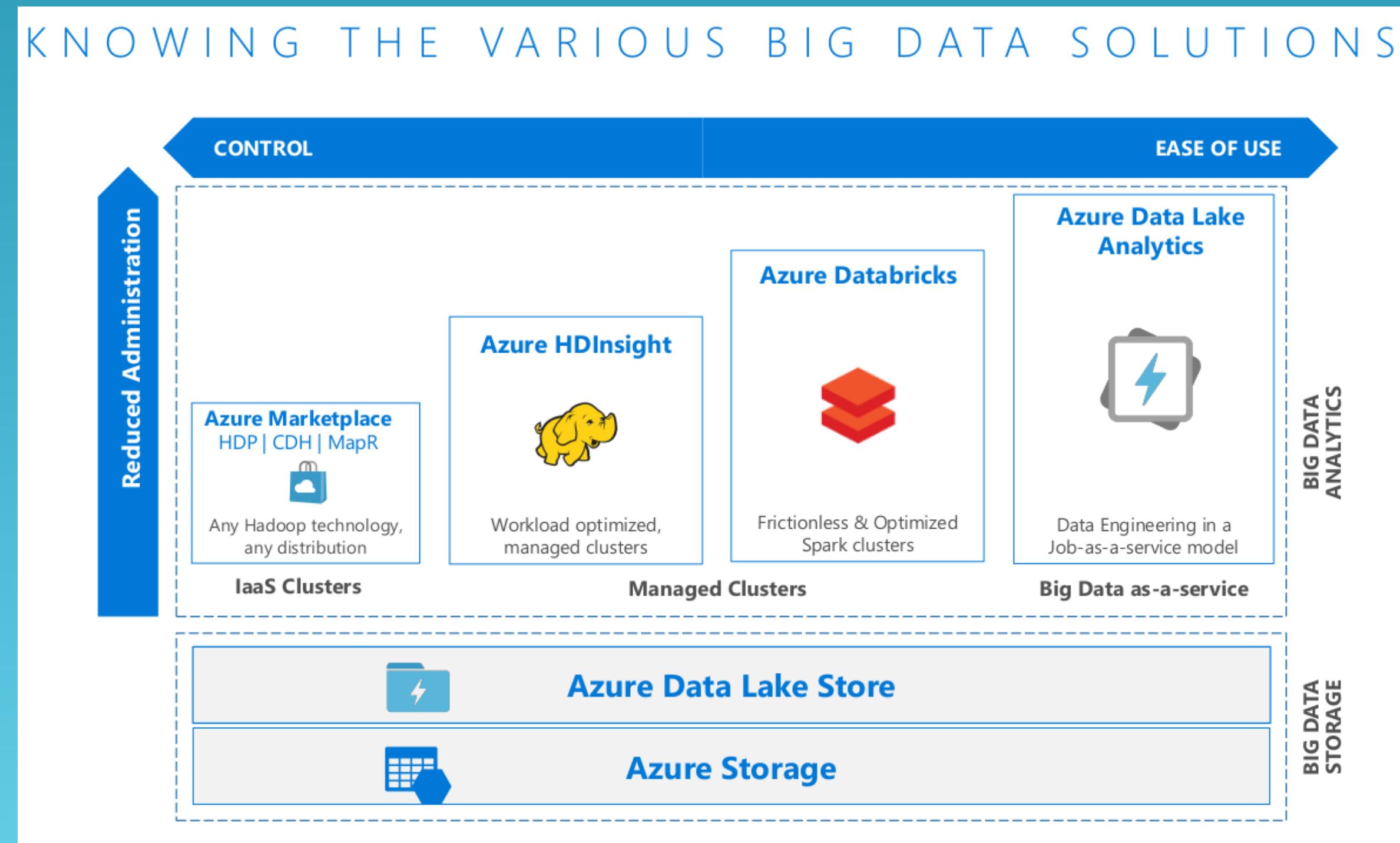
## Advanced Analytics on Big Data



# Azure Databricks



# Azure Databricks



# Azure Databricks pricing

- <https://databricks.com/product/azure-pricing>
- <https://azure.microsoft.com/en-gb/pricing/details/databricks/>

# Databricks Notebooks

# Notebooks

- Narzędzie do interaktywnej pracy, testów i eksperymentów
- Istnieje możliwość użycia także produkcyjne (nie zawsze prosto)
- Połączenie zalet Jupytera i Zeppelin (jest to nowy Notebook)
- Wiele języków w jednym Notebooku

# Przeglądarki

- Google Chrome (current version)
- Firefox (current version)
- Safari (current version)
- Microsoft Edge\* (current version)
- Internet Explorer 11\* on Windows 7, 8, or 10 (with latest Windows updates applied)

<https://docs.azuredatabricks.net/user-guide/supported-browsers.html>

# Kompatybilność

- Kod: Scala, Python, R, SQL
- Jupyter or RMarkDown
- DBC Archive (result included)
- HTML (result included)

# Funkcjonalność

- Tworzenie, usuwanie
- Import, Export
- Kontrola dostępu
- Uruchamianie na klastrze
- Harmonogram (scheduling) -> Jobs

# Magic commands

%md  
%sh  
%fs  
%sql  
%scala  
%python  
%r

<https://docs.azuredatabricks.net/user-guide/notebooks/notebook-use.html#mix-languages>

# Databricks Utilities

- Wbudowane funkcje pozwalające zarządzać
  - Systemem plików
  - Notebookami
- Polecenie %run pozwala uruchomić inny notebook z tego w którym jesteśmy (także równolegle!)
- Widgets pozwalają na parametryzacje Notebooków
- Secrets pozwalają na zapisywanie haseł do zewnętrznych baz bez ujawniania ich w notatniku
- Libraries pozwala na osadzanie bibliotek zewnętrznych (np Jar lub Python)

<https://docs.azuredatabricks.net/user-guide/dev-tools/dbutils.html>

# Wizualizacja

- Automatyczne generowanie wizualizacji
- Pokaż/schowaj kod
- Możliwość tworzenia dashboardów
- Od Databricks Runtime 5.1 lub wyżej, funkcja display wspiera pandas DataFrames
- <https://docs.databricks.com/user-guide/visualizations/index.html>

# Więcej

<https://docs.azuredatabricks.net/user-guide/notebooks/index.html>

# Zewnętrzne źródła danych

# Obsługiwane zewnętrzne źródła danych

- DBFS
- Hadoop Distributed File System (HDFS)
- Connecting to SQL Databases using JDBC

# Azure

- Connecting to Microsoft SQL Server and Azure SQL Database with the Spark Connector
- Azure Blob Storage
- Azure Data Lake Storage Gen1
- Azure Data Lake Storage Gen2
- Authenticate to Azure Data Lake Storage with your Azure Active Directory Credentials
- Azure Cosmos DB
- Azure SQL Data Warehouse

# AWS

- Amazon Redshift
- Amazon S3
- Amazon S3 Select

# Open Source

- Kafka (streaming)
- Cassandra
- Couchbase
- ElasticSearch
- Import Hive Tables
- MongoDB
- Redis

# Komercyjne rozwiązania

- Oracle
- Neo4j
- Riak Time Series
- Snowflake

# Pliki

- Images
- Binary Files
- Avro Files
- JSON Files
- LZO Compressed Files
- CSV Files
- Parquet Files
- Zip Files

# Azure Data Lake gen1

The screenshot shows the Azure portal interface for managing app registrations. The URL in the address bar is `Home > Katalog domyślny - App registrations > azure-data-lake`. The main title is **azure-data-lake**. On the left, there's a sidebar with links: **Overview** (selected), **Quickstart**, **Manage** (with **Branding** and **Authentication** sub-links), and a search bar labeled `Search (Ctrl+ /)`. The main content area has a header with **Delete** and **Endpoints** buttons, and a welcome message: **Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? →**. Below this, detailed information is listed:

Display name	: <code>azure-data-lake</code>
Application (client) ID	: <code>a97c947d-cfe9-4a39-8175-70a8ae62ee5b</code>
Directory (tenant) ID	: <code>5489fb45-ee1f-4751-b8d8-e5cb70c14e21</code>
Object ID	: <code>e689f48f-d7c0-4eef-a69d-31672b374a2c</code>

# Azure Data Lake gen1

The screenshot shows the Azure App Registrations portal. On the left, there is a sidebar with a 'Delete' button and an 'Endpoints' link. The main area has a blue header bar with an information icon and the text: 'Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations?'. Below this, there are four data rows:

Display name	: azure-data-lake
Application (client) ID	: a97c947d-cfe9-4a39-8175-70a8ae62ee5b
Directory (tenant) ID	: 5489fb45-ee1f-4751-b8d8-e5cb70c14e21
Object ID	: e689f48f-d7c0-4eef-a69d-31672b374a2c

On the right side, there are two columns of OAuth endpoints:

OAuth 2.0 authorization endpoint (v2)	<a href="https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/v2.0/authorize">https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/v2.0/authorize</a>
OAuth 2.0 token endpoint (v2)	<a href="https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/v2.0/token">https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/v2.0/token</a>
OAuth 2.0 authorization endpoint (v1)	<a href="https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/authorize">https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/authorize</a>
OAuth 2.0 token endpoint (v1)	<a href="https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/token">https://login.microsoftonline.com/5489fb45-ee1f-4751-b8d8-e5cb70c14e21/oauth2/token</a>

# Azure Data Lake gen1

The screenshot shows the 'Certificates & secrets' blade in the Azure portal. On the left, a sidebar lists several options: Certificates & secrets (selected), API permissions, Expose an API, Owners, Roles and administrators (Preview), Manifest, Support + Troubleshooting, and Troubleshooting.

The main area displays a table for client secrets. The table has columns: THUMBPRINT, START DATE, and EXPIRES. There is one entry in the table:

THUMBPRINT	START DATE	EXPIRES

**Client secrets**  
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[New client secret](#)

DESCRIPTION	EXPIRES	VALUE
azure data lake for databricks	12/31/2299	600*****

# Azure Data Lake gen1

The screenshot shows the Azure Data Lake Storage Gen1 Access blade. On the left, there's a file browser view of the 'movies' folder under 'bigdatapassion'. It lists a single file, 'movies.dat', with a size of 489 KB and last modified on 9/3/2019, 11:47:25 AM. On the right, the 'Access' blade is open, showing the results of a permission assignment:

**Access / (Folder)**

**Success message:** Successfully assigned permissions to azure-data-lake  
7 succeeded, 0 failed.

**Your permissions:**  
radoslawszmit@gmail.com's effective permissions on this folder are: Read, Write, Execute.

**Owners:**

User	Read	Write	Execute
89d0f5b8-dc45-4b02-bbf9-bfa6ce20d17a eac623... radoslawszmit_gmail.com#EXT#@radoslawszmi...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
00000000-0000-0000-0000-000000000000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Assigned permissions:**

User	Read	Write	Execute
azure-data-lake	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Everyone else:**

# Odczyt danych

```
spark.read.format("csv")
    .option("mode", "FAILFAST")
    .option("inferSchema", "true")
    .option("path", "path/to/file(s)")
    .schema(someSchema)
    .load()
```

# Spark Read Mode

Read Mode	Opis
permissive	null jeśli uszkodzony rekord
dropMalformed	usuń uszkodzone rekordy
failFast	błąd jeśli wczytuje uszkodzone dane

# Zapis danych

```
dataframe.write.format("csv")
    .option("mode", "OVERWRITE")
    .option("dateFormat", "yyyy-MM-dd")
    .option("path", "path/to/file(s)")
    .save()
```

# Spark Save Mode

Save Mode	Opis
append	dopisuje do pliku/plików
overwrite	nadpisz wszystko
errorIfExists	błąd jeśli coś już istnieje
ignore	pomiń operację jeśli dane istnieją

# Spark JDBC

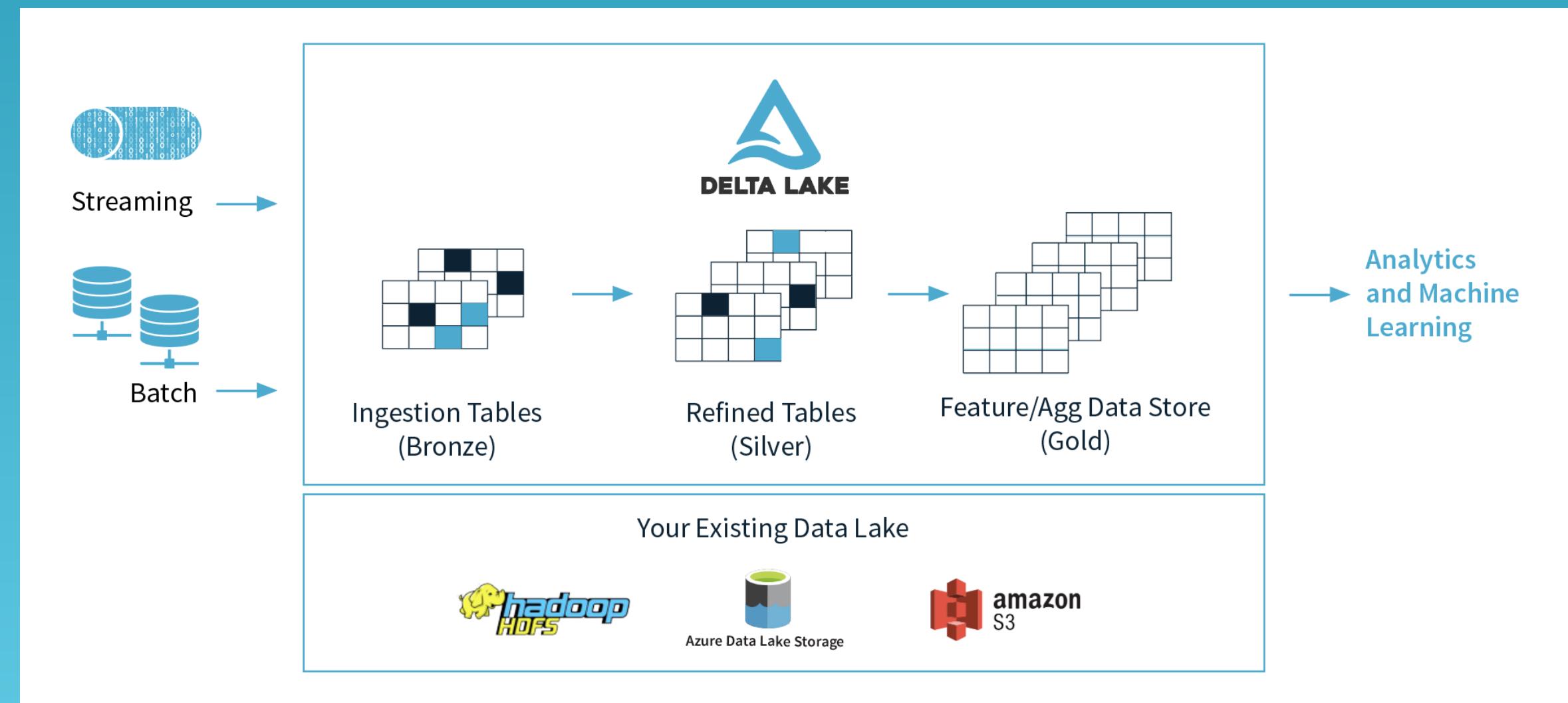
```
val spark = SparkSession.builder().appName("Spark reading jdbc").getOrCreate()

var df = spark.read.format("jdbc").
    option("url", s"${url}${config("schema")}"').
    option("driver", "com.mysql.jdbc.Driver").
    option("lowerBound", min).
    option("upperBound", max).
    option("numPartitions", numPartitions).
    option("partitionColumn", primaryKey).
    option("dbtable", config("table")).
    option("user", user).
    option("password", password)
    .load()

df.repartition(10).write.mode(SaveMode.Overwrite).parquet(outputPath)
```

# Databricks Delta

# Delta Lake (delta.io)



# Delta Lake ([delta.io](https://delta.io))

- Warstwa transakcyjna nad magazynem obiektowym (object store)
- ACID transactions
- Szybki dostęp do danych (zalety object store)
- Dane zapisywane są jako pliki Parquet w DBFS
- Zapis w formacie logu transakcyjnego (jak Kafka)
- Indeksy i partycje
- Wbudowana kompresja (np. Snappy)

# Delta Lake (delta.io)

- Wersjonowanie i pilnowanie Schematu
- Time Travel - snapshot danych by móc się cofnąć
- Obsługa batch i streaming jednocześnie
- Audyt zmian
- Update & Delete
- 100% kompatybilności z Apache Spark API

# DevOps w Databricks

# Integracja z Git

- Trzymanie notebooków w Git
- GitHub lub Bitbucket Cloud
- W wersji Azure także Azure DevOps Services

<https://docs.databricks.com/user-guide/notebooks/notebook-use.html#version-control>

<https://docs.azuredatabricks.net/user-guide/notebooks/github-version-control.html#github-version-control>

# REST API

- Możliwość wykonywania zaawansowanych automatyzacji za pomocą API (np. z programu lub skryptu)
- Zalecane użycie nowszego REST API 2.0 w zamian starszego REST API 1.2 (nie jest już wspierane)
- "The REST API supports a maximum of 30 requests/second per workspace. Requests that exceed the rate limit will receive a 429 response status code."
- Można korzystać łatwo przez Databricks CLI
- Zarządzanie zewnętrznymi narzędziami, np. Ansible

[https://galaxy.ansible.com/colemanjag1/ansible\\_databricks](https://galaxy.ansible.com/colemanjag1/ansible_databricks)

# Databricks CLI

- <https://docs.azuredatabricks.net/user-guide/dev-tools/databricks-cli.html>
- <https://docs.microsoft.com/pl-pl/azure/azure-databricks/databricks-cli-from-azure-cloud-shell>

# Databricks CLI - Instalacja

```
virtualenv --no-site-packages databricks-cli  
source ./databricks-cli/bin/activate  
pip install databricks-cli
```

```
databricks --version
```

# Python - co mamy?

```
pip list -o  
python -m pip list -o
```

# Databricks CLI - konfiguracja

```
$ databricks configure --token
Databricks Host (should begin with https://): https://westeurope.azuredatabricks.net
Token: *****
```

```
$ cat ~/.databrickscfg
[DEFAULT]
host = https://westeurope.azuredatabricks.net
token = *****
```

# Databricks CLI - polecenia

```
$ databricks workspace ls
Users
Shared
$ databricks fs ls
FileStore
databricks-results
delta
ml
tmp
user
vs-datasets
```

# Databricks CLI - polecenia

```
databricks fs ls dbfs:/  
databricks fs ls
```

```
databricks fs mkdirs dbfs:/bigdatapassion  
databricks fs cp -r /home/radek/Projects/BigDataPassion/training-dataset/ \  
dbfs:/bigdatapassion/  
databricks fs ls dbfs:/bigdatapassion
```

# Databricks Connect - Instalacja

```
virtualenv --no-site-packages databricks-connect
source ./databricks-connect/bin/activate
pip install -U databricks-connect==5.5.*
```

```
databricks-connect configure
```

```
databricks-connect test
```

# Databricks Connect - Użycie

```
$ python
>>>
>>> print(spark.range(100).count())
View job details at https://westeurope.azuredatabricks.net/?o=8869754808689755#/settings
100
>>>
```

# Administracja w Databricks

# Uwierzytelnianie (Authentication)

- <https://docs.azuredatabricks.net/api/latest/authentication.html>
- <https://docs.azuredatabricks.net/user-guide/secrets/index.html>

# Active Directory

- W przypadku Azure Databricks usługa jest domyślnie zintegrowana z Azure Active Directory
- Zaawansowane zarządzanie rolami wymaga jednak subskrypcji Premium

# Migracja do Databricks

- Migrating Single Node Workloads to Azure Databricks
- Migrating Production Workloads to Azure Databricks
- <https://docs.azuredatabricks.net/migration/index.html>

# Autoskalowanie klastrów

- Automatyczna funkcjonalność pozwalająca zwiększać lub zmniejszać moc naszego klastra
- Potrafi zaoszczędzić 30% kosztów użycia chmury
- Liczba wykonawców (executors) zmienia się w zależności od aktualnego stanu (stage) czy wielkości danych
- Decyzja odbywa się automatycznie na podstawie zbieranych statystyk

<https://databricks.com/blog/2018/05/02/introducing-databricks-optimized-auto-scaling.html>

# Uruchamianie zadań i praca produkcyjna

- UI - do testów
- API / CLI - użyteczne, ale wymaga dodatkowego nakładu pracy w stworzenie skryptu/programu
- Wbudowany mechanizm Jobów - prosty ale skuteczny
- Azure Data Factory - przydatne do prostych przepływów pracy z Azure
- Airflow - bardzo zaawansowane narzędzie stworzone przez Google (rozwijane także w Polsce)
  - <https://airflow.readthedocs.io/en/stable/integration.html?#databricks>
  - <https://docs.azuredatabricks.net/user-guide/dev-tools/data-pipelines.html#apache-airflow>

# Harmonogram Spark Submit

```
[  
  "--class",  
  "org.apache.spark.examples.SparkPi",  
  "dbfs:/bigdatapassion/spark-examples/spark-examples_2.11-2.4.4.jar",  
  "10"  
]
```

<https://spark.apache.org/docs/latest/submitting-applications.html>

<https://docs.azuredatabricks.net/api/latest/jobs.html#jobssparksubmittask>

# Troubleshoot jobs

- Job Details Pages / Spark UI
- Przekazanie logów do innego zasobu
  - parametr cluster\_log\_conf wskazujący na DBFS (nawet montowany punkt)
  - co 5 minut zrzut
  - <https://docs.azuredatabricks.net/api/latest/jobs.html#jobsclusterspecnew>

# Monitoring

- Zakładka "Metrics"
  - zapis metryk co 15 minut
  - dostępne w trakcie działania i po zakończeniu
  - oparte o Ganglia
  - można podłączyć swoje narzędzie (Azure Monitor, Datadog, )
- Powiadomienia email
  - <https://docs.azuredatabricks.net/api/latest/jobs.html#jobsjobsettingsjobemail>
  - można przekazać dalej (np. PagerDuty lub Slack)

# Metadane

- Spark korzysta ze swojego Hive Metastore
- Databricks prezentuje Hive Metastore w GUI (Tables View)
- Możliwość podłączenia do zewnętrznego Hive Metastore (np. HDInsight lub AWS Glue)
  - <https://docs.azuredatabricks.net/user-guide/advanced/external-hive-metastore.html>

# Artificial Intelligence in Databricks

# Sztuczna Inteligencja w Databricks

- Wsparcie dla Data Science i ML
- TensorFlow, Keras, XGBoost
- Horovod: Distributed training framework for TensorFlow, Keras, PyTorch, and Apache MXNet (Created by Uber)

<https://eng.uber.com/horovod/>

# Klastry ML

- Dostępne specjalne wersje klastrów "Databricks Runtime ML"
- Możliwość wybrania wersji GPU
- Jest także wersja z popularnym środowiskiem Anaconda

# MLflow

- open source framework do zarządzania cyklem uczenia ML
- optymalizacja przepływów pracy ML
- Komponenty: Tracking, Models, and Projects
- <https://databricks.com/mlflow>
- <https://mlflow.org/>

# Podsumowanie

# Więcej informacji można znaleźć:

- <https://docs.databricks.com/>
- <https://docs.azuredatabricks.net/>
- <https://databricks.com/resources>
- <https://kb.azuredatabricks.net/index.html>

# Google Cloud Platform



Google Cloud

# Google Cloud

- Stworzona przez Google Inc.
- Dość późno udostępniona do użytku innym firmom (6 października 2011)
- Bazuje na wewnętrznych rozwiązańach firmy niespotykanych w innych chmurach oraz bez możliwości ich samodzielnego zainstalowania
- Pracownicy firmy są pionierami Big Data na świecie, dzięki ich publikacji powstał Hadoop (Google File System, Google MapReduce, Google BigTable)
- Pod względem mocy obliczeniowej jest to najprawdopodobniej największa chmura obliczeniowa na świecie (wliczając usługi Google)

# Oracle Cloud

The Oracle logo, featuring the word "ORACLE" in a bold, white, sans-serif font. The letter "O" is unique, consisting of a thick vertical stroke on the left and a horizontal loop on the right. A registered trademark symbol (®) is positioned at the top right of the letter "E". The logo is set against a solid red rectangular background.

# Oracle Cloud

- Chmura obliczeniowa firmy Oracle
- Odpowiedź na chmurę firmy Microsoft i klientów migrujących się z rozwiązań opartych o własny sprzęt i licencji na bazy w chmurze
- Big Data Cloud oparty jest o dystrybucję Cloudera

# IBM Cloud



**IBM Cloud**

# IBM Cloud

- Chmura firmy IBM
- Podobnie jak Azure oparta o dystrybucję Hortonworks
  - Hortonworks Data Platform (HDP)
  - Hortonworks Data Flow (HDF)
- Integracja HDP/HDF z IBM Db2® Big SQL
- Dodatkowe narzędzia IBM
  - IBM Streams
  - IBM Big Replicate
  - IBM BigIntegrate
  - IBM BigQuality

# Alibaba Cloud

# Alibaba Cloud

- Stworzona przez firmę Alibaba, chiński gigant e-commerce
- Wbudowane rozwiązania firmy jak:
  - Image Search - wyszukiwarka obrazów
  - Intelligent Service Robot - chatbot dla biznesu
  - Dataphin - inteligentny silnik danych
  - DataV - narzędzie do wizualizacji danych
  - Quick BI - analityka biznesowa
  - Alibaba Cloud Elastic MapReduce (E-MapReduce) - dystrybucja Big Data (Hadoop, Spark, HBase, Pig, Hive)
  - MaxCompute - platforma do przetwarzania danych (SQL, MapReduce, Graph, and MPI)
  - DataWorks - narzędzie Big Data do przetwarzania danych, oparta o MaxCompute

# Oktawave

# Oktawave

- Polska chmura obliczeniowa
- Założona jako startup sfinansowany przez grupę K2
- Autorskie rozwiązania dyskowe (Tier 5 do 200 000 IOPS)
- Lokalizacja w Warszawie (Polska)

# Oktawave

- Wszystkie regiony w PL (Domaniewska, Warszawa)
  - PL-001 (procesory Intel)
  - PL-002 (procesory AMD)
  - PL-003 (procesory AMD)
  - PL-004 (procesory Intel)
  - PL-005 (procesory Intel)
- Dwa centra danych
- CPU jest taki sam dla wszystkich regionów równy 2,5 Ghz

# Pytania



Dziękuję

