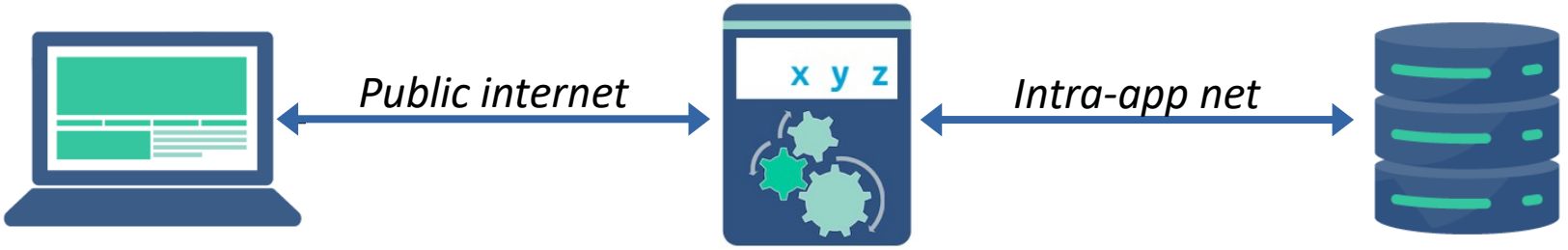
Cloud computing: concetti di base

- Siamo nell'ambito di distributed computing / distributed systems
- L'impiego tipico del cloud è supportare applicazioni con architettura multi-tier (tipicamente 3-tier):



	Tier 1	*Tier 2*	*Tier 3*
Funzione	Presentazione, interazione	Applicazione, Business logic	Gestione dati
Ruolo	Cliente	Server	DB server
Eseguito dal componente SW	Browser/ App mobile	Web server + app engine (PHP, Java)	DBMS (es. mysql, Oracle…)
Supportato dal componente infrastrutturale	PC / mobile	Host	Host

<u>Idea essenziale del cloud computing</u>

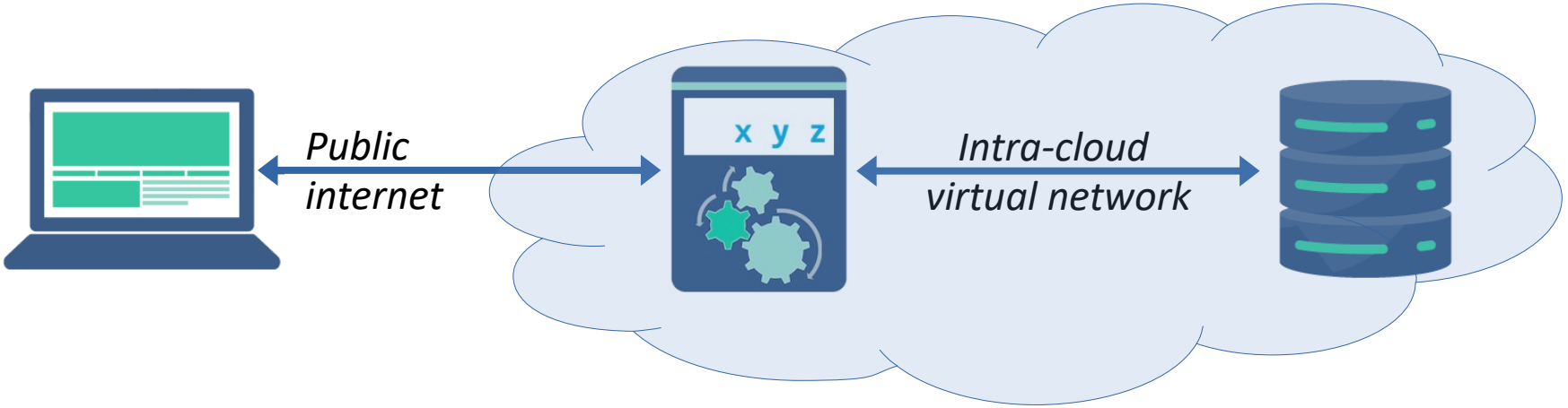
I componenti infrastrutturali di tier 2 (host del server), tier 3 (host del DB) e la rete che li collega sono *virtualizzati* sfruttando le risorse fisiche, sicché:

- ➔ l'applicazione gira su *host virtuali* collegati da una *rete virtuale*
- ➔ le effettive risorse fisiche sono condivise ("in *pooling*") tra quelle virtuali

Cloud pubblico

Possiamo ora localizzare l'architettura multi-tier rispetto al cloud:

- tier 2, 3 (virtuali) e la rete (virtuale) tra loro **risiedono nel cloud**



Poiché gli host virtuali su cui girano app e DB, e la rete virtuale che li collega, risiedono nel cloud: si dice anche che l'app è stata *cloudificata*

Di chi sono le risorse fisiche che supportano quelle virtuali del cloud? 3 casi:

1. appartengono a un *cloud provider* (Amazon AWS, Microsoft Azure…) che vende servizi cloud, si parla allora di **cloud pubblico**
2. *Colocation*: il **colocator** fornisce locali, cooling, power e link di rete, ma non l'hardware "informatico" (apparati di calcolo, storage e rete)
3. Soluzione **on premises**: tutte le risorse fisiche (locali e hardware) appartengono all'azienda per conto della quale girano le applicazioni

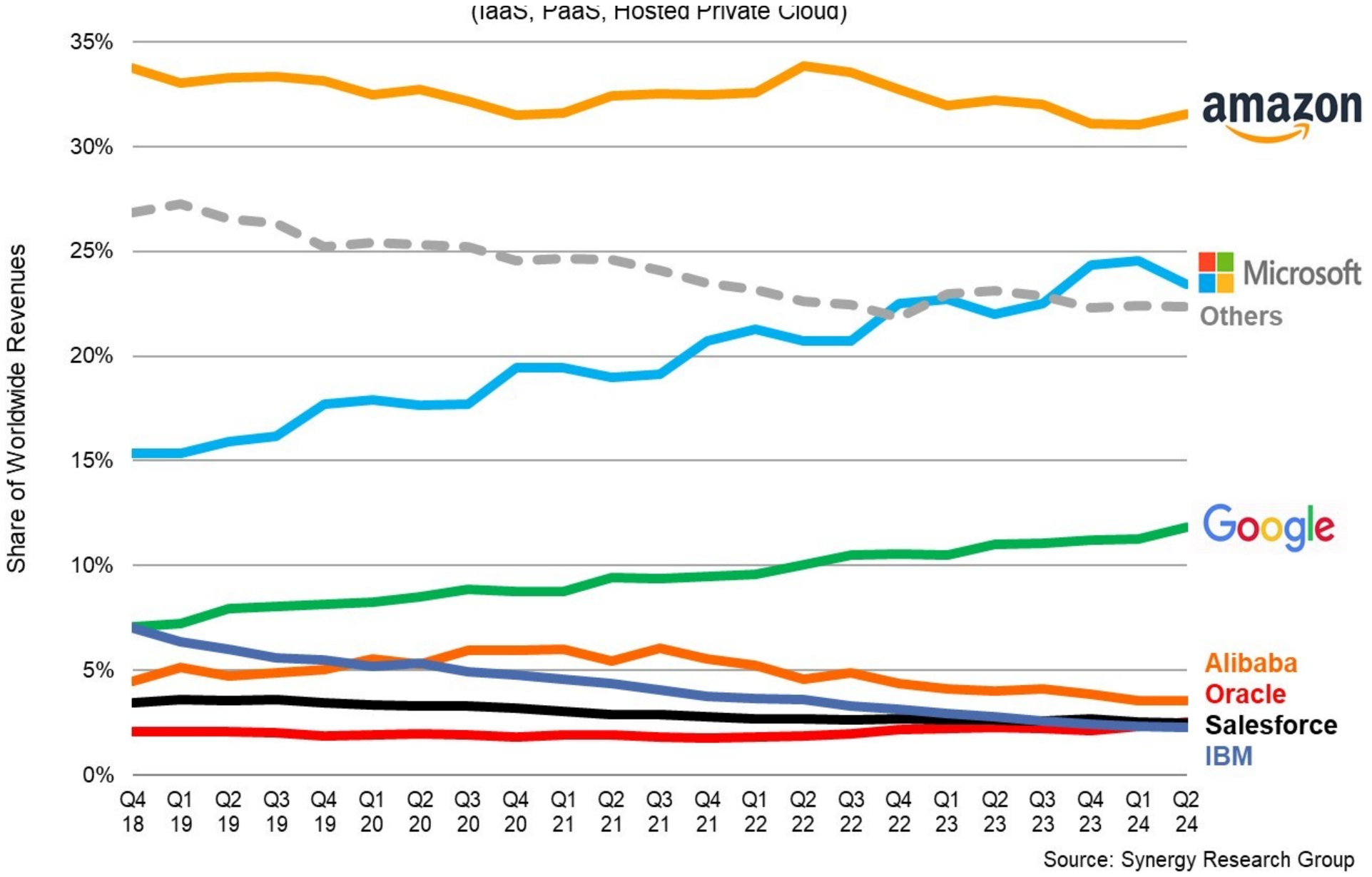
Cloud pubblico: vantaggi

- **Semplicità d'uso** (delle risorse virtuali rispetto a quelle fisiche)
- **Economicità** (*cost efficiency*):
  - no *upfront costs* (capitali iniziali per l'acquisto di infrastruttura fisica)
  - ➔ *OPEX vs. CAPEX*: OPerating vs. CAPital Expenditure (investimenti)
  - *risparmio* di costi di personale (per amministrare l'infrastruttura fisica)
  - *pay-as-you-go*: fatturazione a consumo, da cui *scalabilità economica*
- **Scalabilità** (tecnologica): l'ammontare di risorse virtuali impiegate è in grado di scalare al variare del workload (anche **dinamicamente**)
- **Qualità del servizio** o **QoS**
  - **SLA** (**S**ervice **L**evel **A**greement) contrattuale tra cloud provider e clienti
  - **H**igh **A**vailability (**HA**)
  - requisiti sulle risorse (garanzie del cloud provider al cliente riguardo a #core virtuali, RAM virtuale, disk size, bandwidth/throughput...)
- **Fault Tolerance** (attraverso *replicazione* e *distribuzione* geografica)
- **Sicurezza**: tra risorse virtuali è più facile assicurare la *segregazione*
- **Innovazione**: i mezzi dei grandi cloud provider garantiscono soluzioni tecnologiche sempre sul fronte avanzato dello stato dell'arte

Alcuni cloud provider

- AWS (Amazon Web Services): il più grande e antico (2006)
- Microsoft Azure
- Google Cloud Platform
- Alibaba (simile a AWS, per la Cina)
- Salesforce
- IBM
- Oracle Cloud
- Digital Ocean (vocazione Debian/Ubuntu, piccolo ma con comunità vivace e ottima documentazione)

Cloud providers: market share trend



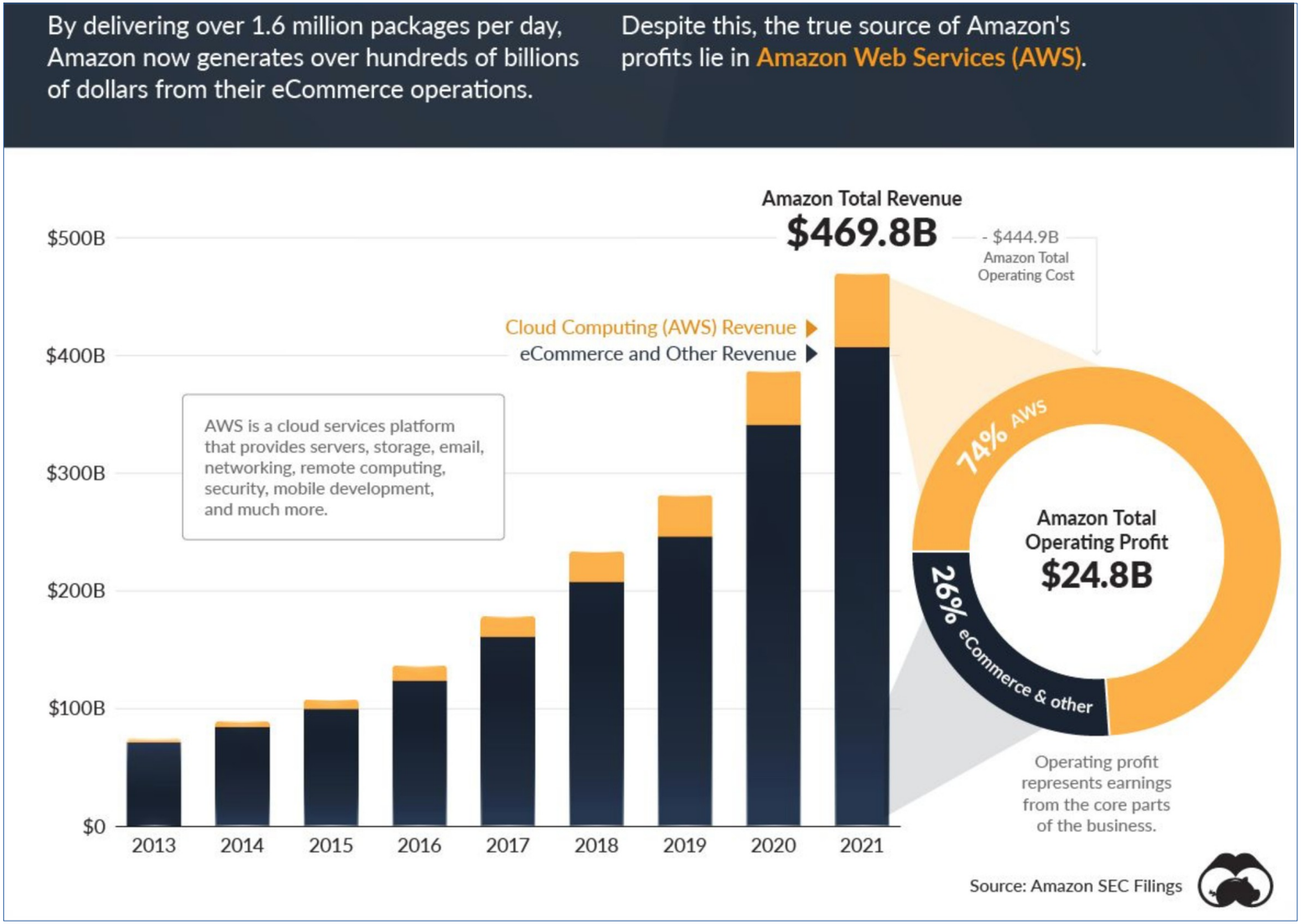
La figura è qui, vedi aggiornamenti su https://www.srgresearch.com/articles

AWS is large enough that… [from The Guardian, 2018]

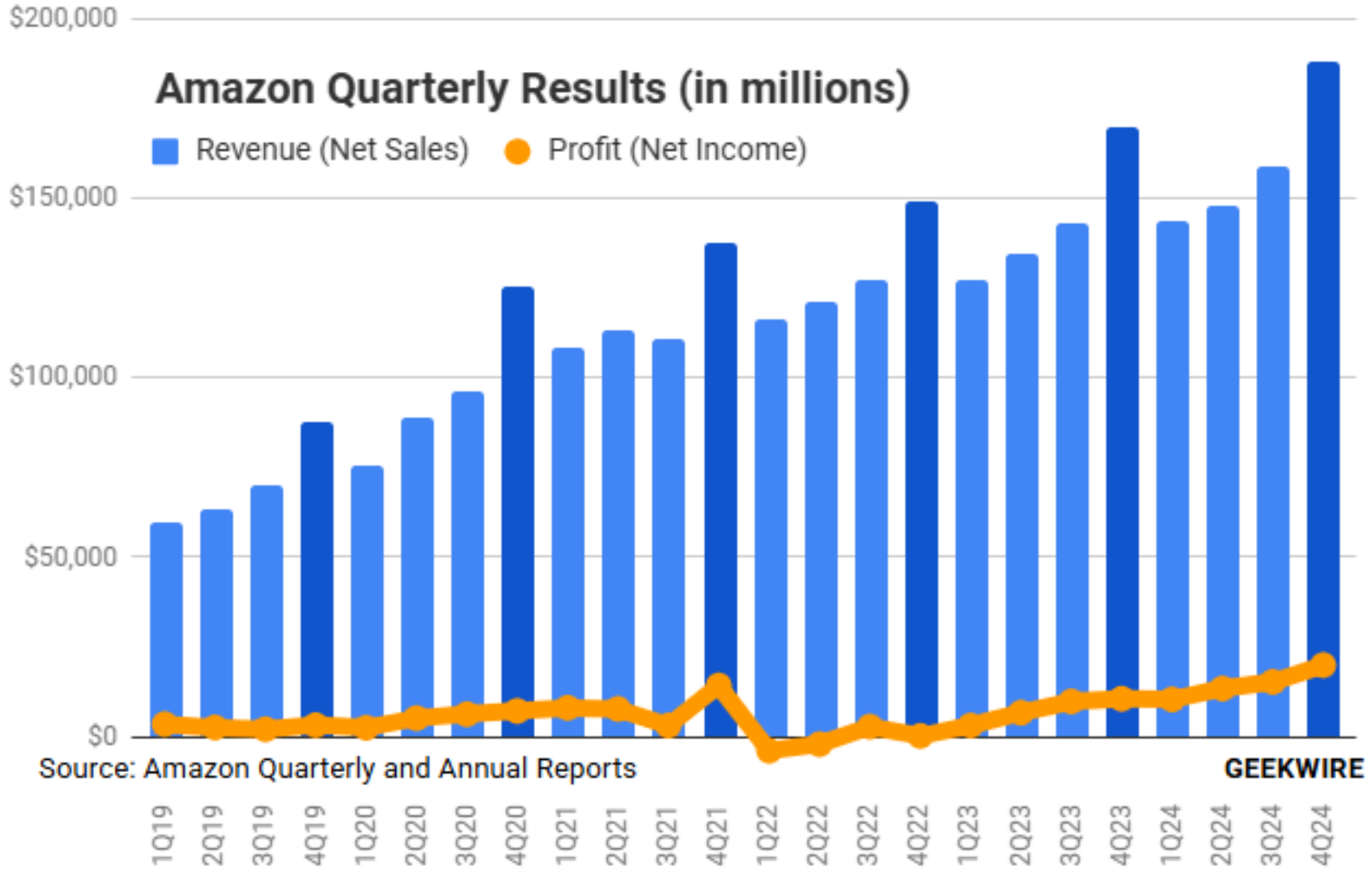
- [AWS] is now 10% of Amazon's overall revenue... Amazon divides its company into "US and Canada", "International", and "AWS"
  - AWS is large enough that it is dealt with on the same tier as the entire rest of the world [ecommerce]!
- AWS is large enough that Netflix, which accounts for around 1/3 of all internet traffic in North America, is just another customer.
- AWS is large enough that in 2016 they released the *Snowmobile*, a literal truck for moving data.
  - Now, if you want to upload a lot of data to Amazon's cloud, [they'll] drive a truck to your office, fill it with data, drive it back
    - at 1Gb/s, uploading 50TB will take 4 days; uploading 100 Petabytes would take a little over 25 years
    - to upload 100 PB – roughly 5M movies in 4k – there's no quicker way than driving it down the freeway

AWS: fatturato (*revenue*) e profitti

- AWS continua a pesare meno del 20% del fatturato totale di Amazon, ma è la linea di business a più alto valore aggiunto…
- dà ben il 74% dei profitti (2022)
- Fonte: https://www.visualcapitalist.com/aws-powering-the-internet-and-amazons-profits/



Amazon: fatturato e utili



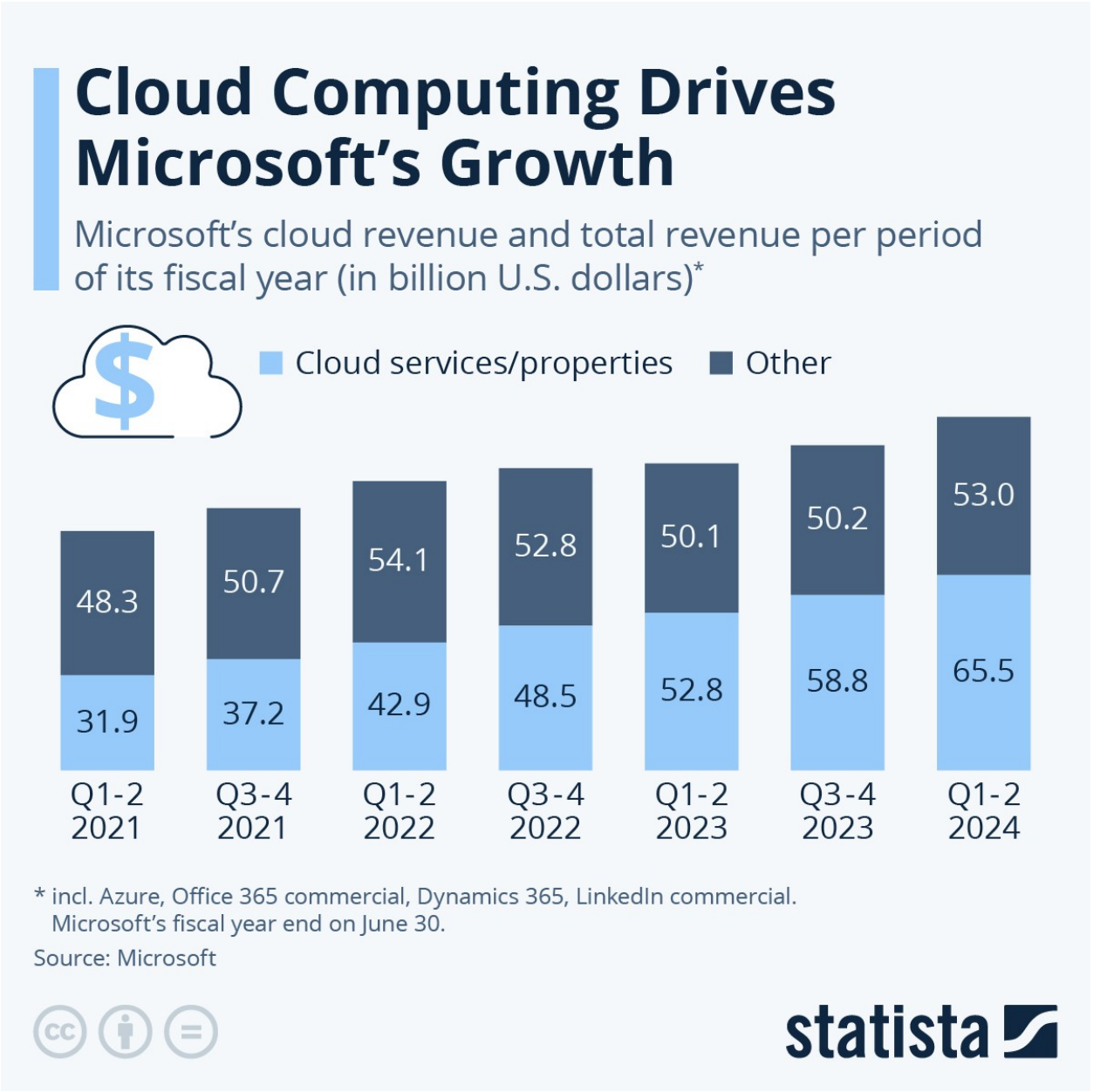
Il margine di Amazon (tutte le divisioni) è circa il 10%

AWS: fatturato e utili

Il margine di AWS è quasi il 40% !

Cloud e fatturato: Microsoft

- Anche per Microsoft, il cloud ha un peso crescente sul fatturato
- Dal 2023 ha superato ogni altra fonte di introiti



Hyperscalers: caratteristiche

Hyperscaler: operatore cloud globale con caratteristiche specifiche:

- *Scala*: immensa, con data center sparsi su molteplici regioni/continenti, ogni data center ospita da K a M di server (e relativa infrastruttura)
  - NB: a volte i data center sono fruiti dagli hyperscaler in *colocation*
- *Global Reach*: offre servizi da data center localizzati ovunque sul pianeta in modo da assicurare bassa latenza e alta disponibilità
- *Elasticity*: offre servizi altamente scalabili, permettendo agli utenti di scalare le risorse impiegate verso l'alto o il basso, adattandosi alle fluttuazioni della domanda dinamicamente e senza affrontare grossi investimenti iniziali
- *Cost Efficiency*: enormi economie di scala, grazie a *resource pooling* che ottimizza l'utilizzo delle risorse fisiche
  - si suppone che parte dei benefici connessi sia trasferita agli utenti, consentendo loro risparmi rispetto a un'infrastruttura IT *on premises* tradizionale
- *Innovazione*: capacità di mantenersi sul fronte avanzato dello stato dell'arte della tecnologia, trasferendone i benefici agli utenti

Chi sono gli *hyperscalers* e quanto pesano*?*

https://www.datacenterknowledge.com/manage/2023-these-are-world-s-12-largest-hyperscalers

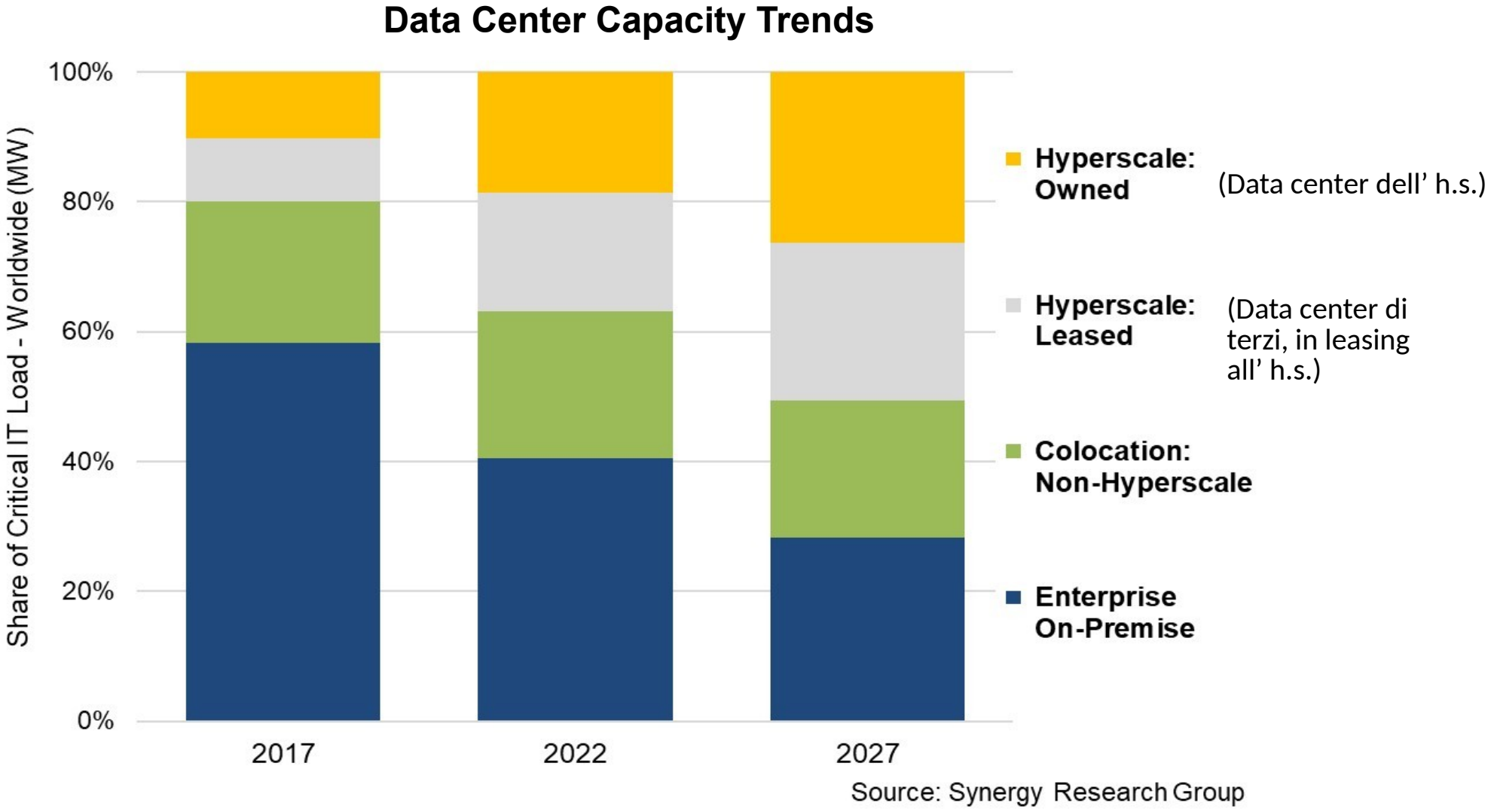
NB: non tutti gli hyperscaler sono provider di servizi cloud pubblici IaaS/PaaS

- p.es. Facebook/Meta fornisce solo SaaS, p.es. Facebook o WhatsApp

Hyperscaler	MW per data center installati al 2022	MW progettati al 2023
Google	3024	2905
Microsoft	2176	3344
Amazon	2480	2533
Meta	1790	2595
Apple	600	1403
Alibaba	1350	487
Huawei	494	192
Baidu	608	36
Tencent	487	152

Hyperscaling vs. colocation vs. on-premises

- https://www.srgresearch.com/articles/on-premise-data-center-capacity-being-increasingly-dwarfed-by-hyperscalers-and-colocation-companies



Cloud: privato

Nei casi 2 (colocation) e 3 (on premises) visti in precedenza, le risorse hardware sono di uso esclusivo di un'organizzazione.

Ciò non toglie che, all'interno dell'organizzazione, se ne possa comunque fruire in **modalità cloud**, cioè *virtualizzando* calcolo, storage e rete e interagendo con tali risorse virtuali attraverso Internet.

Si parla in tal caso di **cloud privato** (anziché pubblico)

Vantaggi comuni al cloud pubblico:

- pooling delle risorse fisiche, da cui…
- efficienza nella fruizione delle risorse fisiche
- *availability* (grazie a replicazione, elasticità, disaster recovery…)

Cloud privato vs. pubblico

Svantaggi del cloud privato = perdita di vantaggi propri del cloud pubblico:

- perdita di **scalabilità** (a fronte di picchi o crolli delle richieste)
- perdita di **flessibilità** (cioè no pay-as-you-go, servono upfront investments)
- perdita di **economie di scala** (accessibili ai grandi operatori)
- perdita della ricca **gamma** di servizi cloud pubblici, difficoltà rispetto a **innovazione**
- perdita di **presenza globale** (salvo che l'organizzazione ne affronti i costi)
- perdita di **focus** sul *core business*

Cloud privato vs. pubblico

Vantaggi propri del cloud privato (corrispondono a svantaggi del cloud pubblico)

- **Flessibilità**, rispetto a esigenze di integrazione con infrastruttura proprietaria (si parla di cloud ibrido)
- **Prestazioni** certe (cioè non esposte a fluttuazioni dovute alla domanda complessiva di innumerevoli clienti)
- **Customizzazione** possibile di servizi/funzionalità/etc. rispetto alle esigenze dell'organizzazione
- **Security e compliance**: non in senso assoluto, ma nel senso della possibilità di aderire a standard prescelti dall'organizzazione

Cloud: privato vs. pubblico

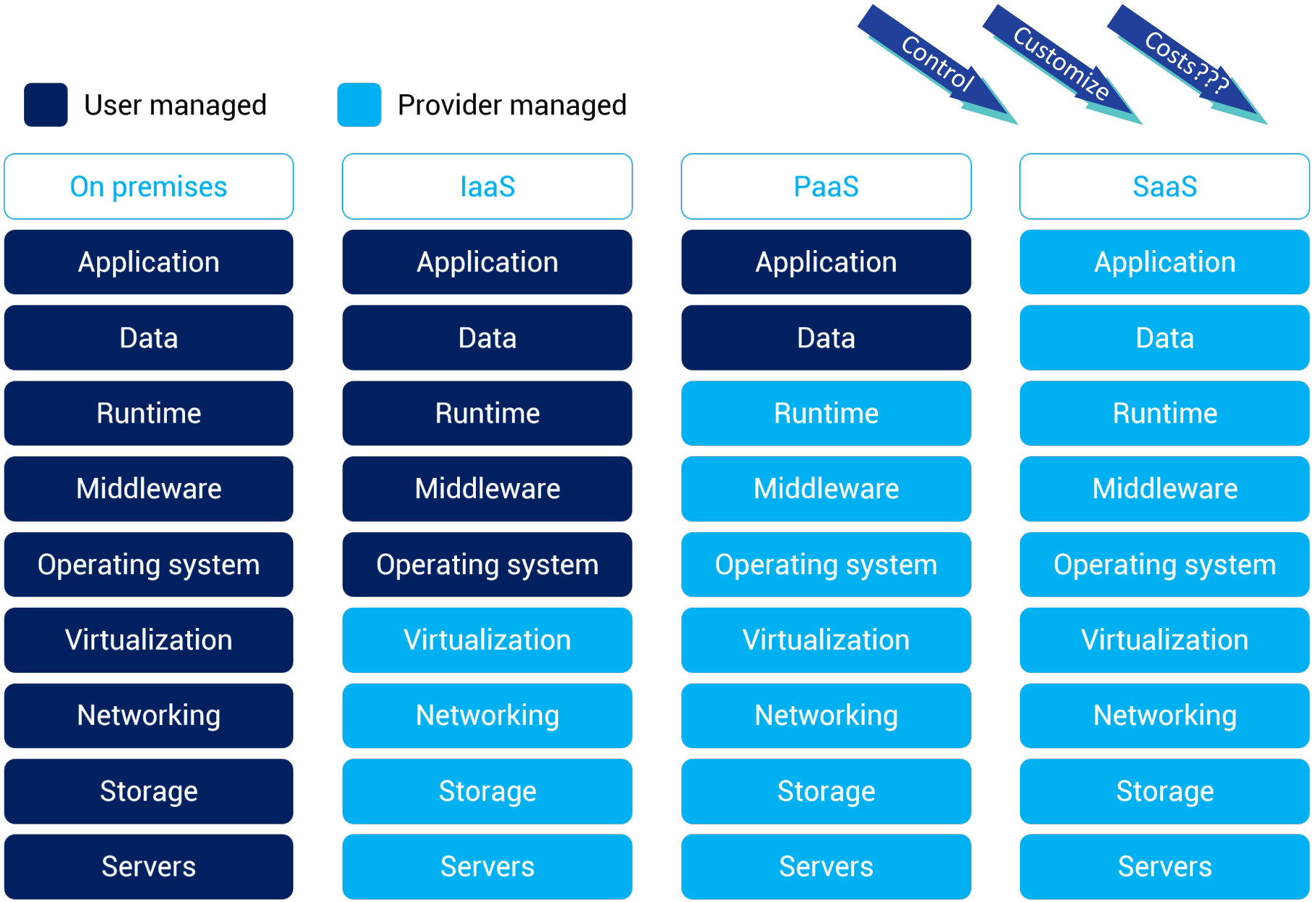
Alcuni software per cloud privati:

- VMware vSphere: focus su virtualizzazione dell'infrastruttura di calcolo, manca di soluzioni cloud "native", p.es. per storage, identity and access management, billing...
- Microsoft Azure Stack: Azure services on-premises
- AWS Outposts: AWS services on-premises
- Red Hat (IBM) OpenShift: a containerization platform, installabile on premises (ma anche fornita in cloud)
- Citrix CloudPlatform (ex Apache CloudStack): open-source
- OpenStack: il più completo tra gli open-source, con la più ampia *user base*, cerca di approssimare la ricchezza dei grandi cloud pubblici

Cloud Computing pubblico, secondo NIST: caratteristiche

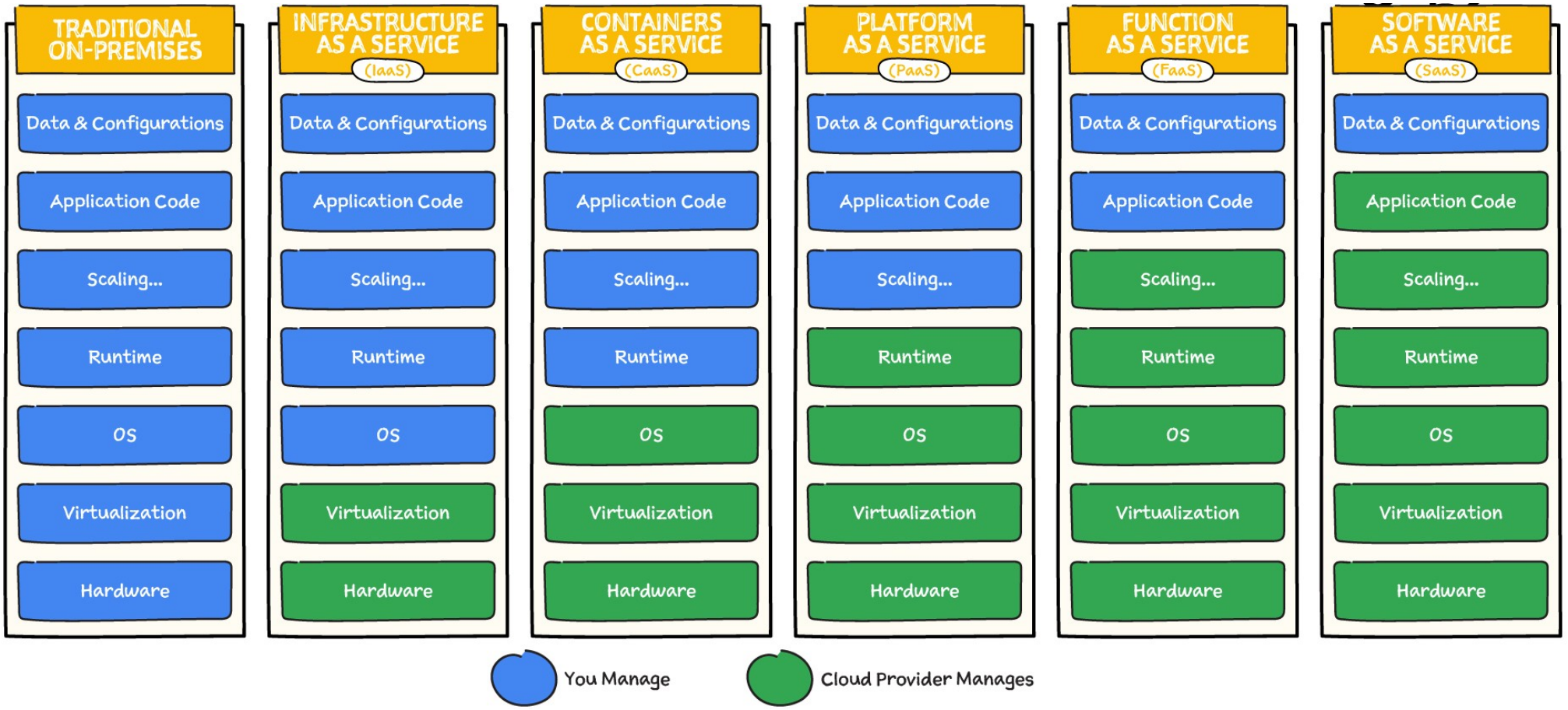
- *on-demand*: ciò che serve, quando serve e automatica-mente o comunque senza interazioni “fuori dal sistema”
- *accesso/utilizzo rete*: accesso via rete pubblica, disponibilità di rete privata per l’infrastruttura in uso
- *resource pooling*:
  - le risorse “concettuali” (CPU, storage) sono virtualizzate, ma...
  - altre risorse fisiche essenziali (locali del data center, power, raffreddamento…) sono condivise (in modo trasparente)
- *elasticità rapida*: scalabilità veloce ed *automatica* (on demand)
- (costo) servizi *misurabili* (pay-as-you-go, pay-for-use: contatori)

Cloud Computing: modelli di servizio XaaS (X as a Service): IaaS (Infrastructure), PaaS (Platform), SaaS (Software)



CaaS e FaaS

- CaaS: Containers as a Service, il S.O. è gestito dal provider
- FaaS: Function as a Service, l'utente definisce una funzione (p.es. Python, il cloud la esegue)



Cloud Computing secondo NIST: modelli di servizio - IaaS

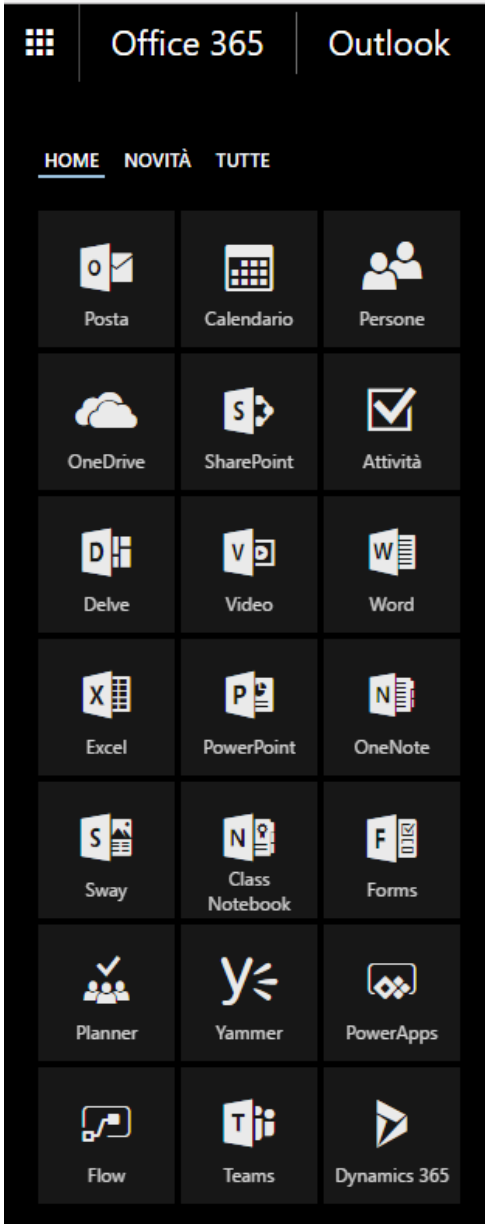
- **IaaS**, **I**nfrastructure **a**s **a** **S**ervice
- Esempi tipici: AWS, Openstack, MS Azure.
- *Consumer* has capability to provision: processing, storage, networks
- Physical resources are accessed as virtual entities:
  - VMs consisting of virtual CPUs, equipped with
  - virtual RAM,
  - virtual disks etc,
  - interconnected by virtual networks...
- On his virtual infrastructure, the customer can deploy and run arbitrary software, including OS, middleware and applications

Cloud C. secondo NIST: modelli di servizio - PaaS

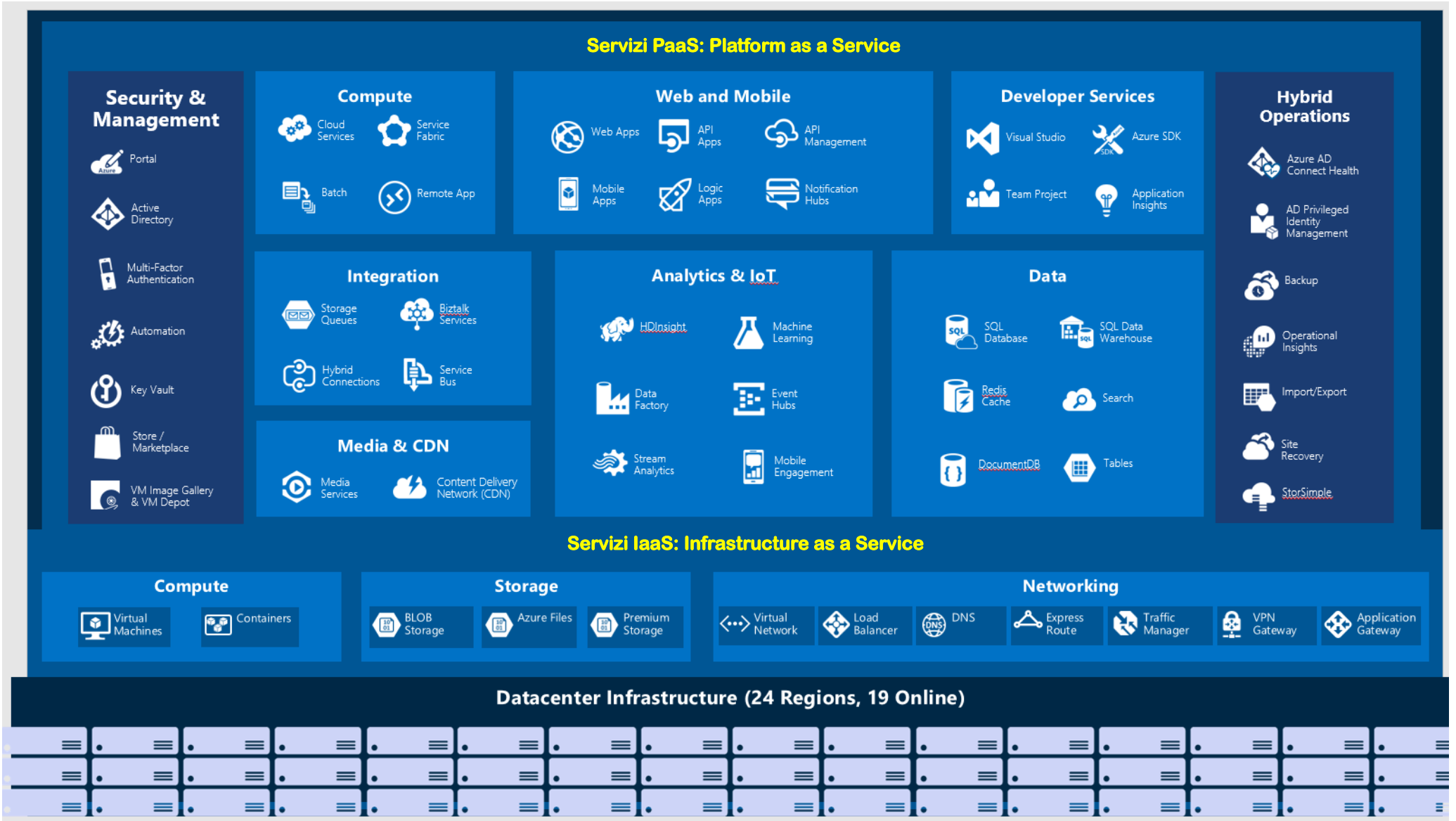
- **PaaS: P**latform **a**s **a S**ervice
- Esempi tipici: Google App Engine, OpenShift (Kubernetes on cloud, by RedHat), AWS RDS (DB relazionale) e DynamoDB (non relazionale)
- *Consumer* has capability to deploy applications onto the cloud infrastructure:
  - applications are created using programming languages, libraries, [OS], services, and tools supported by the provider
  - applications are hosted (run within) a (cloud) **platform**, i.e. a virtual environment made available *as a service* by the cloud provider
- *Consumer* does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, neither in a physical nor in a virtual sense.
- Consumer has control over the deployed applications and possibly some configuration settings for the application-hosting environment

Cloud C. secondo NIST: modelli di servizio - SaaS

- **SaaS**, **S**oftware **a**s **a** **S**ervice, eg. Gmail, Dropbox, Google Drive/Docs, MS 365…
- Consumer uses the provider's applications running on a cloud infrastructure:
  - **applications** are accessible from a *thin client interface*, eg. web browser (es. qui a destra), or a *program interface* (i.e., API)
  - consumer **does not manage or control**
    - either the underlying **cloud infrastructure** including: network, servers, OS, storage,
    - or detailed **application configuration** (over time, more and more configuration options are becoming available)
- In prospettiva: EaaS - Everything as a Service



Esempio - Azure: principali servizi IaaS/PaaS



https://www.slideshare.net/mgafar/extending-your-data-center-to-the-cloud-with-windows-azure

Servizi cloud managed

In un servizio **IaaS**, tipo gestione VM (istanze), l'utente **ottiene dal cloud** risorse (di calcolo, storage, rete) virtualizzate, ma deve provvedere:

- al ***provisioning***: richiesta e configurazione delle risorse (es. lancio istanze)
- al ***management***: gestione delle risorse ottenute (p. es. installazione/ configurazione di S.O., software di sistema e applicativo, security, maintenance, amministrazione di sistema …)

  (NB: il provisioning si può (o no) considerare (la prima) attività di management)

Un servizio PaaS **managed** sgrava l'utente (di parte) del carico di management

- p.es. AWS RDS (Relational Database Service) è un servizio managed:
  - l'utente sceglie il tipo (p.es. # core) dell'istanza/e su cui girerà il DBMS
  - ma non deve, né può, installarvi SW (da OS a DBMS)
  - il provisioning dell'istanza/e è esplicito, nel senso che l'utente sa che il cloud riserverà un'istanza/e per supportare il suo DB, e ne sceglie le caratteristiche
  - tuttavia, l'utente non deve gestire istanze (lanciarle/amministrarle/configurarle se non attraverso RDS (e per ciò che questo consente) o prefissarne il numero
- altro esempio: EKS (Elastic Kubernetes Service)

Esempio di cloud managed: AWS RDS

L'utente sceglie:

- il software di RDBMS (p.es. mysql o mariadb o postgres...)
- il tipo di istanza sottostante il DB (RAM, #vCPU, Mb/s)
- se il DB avrà un IP pubblico e … **poco altro**

RDS provvede, in modo trasparente per l'utente a:

- installare/configurare/aggiornate il SW necessario, dall'OS in su
- scalare (fino a centinaia di TB) lo storage per DB/tabelle
- scalare il numero di repliche (in lettura) del DB, in modo da garantire le prestazioni desiderate (high availability etc.)
- consentire manutenzione del DB senza interrompere il servizio (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments.html)
- backup e snapshot automatici dei dati e ...
- … **molto altro**

I tempi di creazione/attivazione di un DB RDS sono piuttosto alti, proprio per la complessità delle operazioni di management che si svolgono dietro le quinte

AWS RDS: costi

Il costo complessivo di un DB RDS è nell'ordine delle centinaia di $ al mese:

- un servizio managed come RDS costa più delle risorse che impiega
- il costo del management grava sull'utente, che però, in cambio, non deve provvedervi personalmente
- la convenienza effettiva di una scelta del genere dell'utente (delega/outsourcing di compiti/responsabilità) va sempre valutata in termini di **TCO**: Total Cost of Ownership (personale, know-how...)

Servizi cloud Serverless

Un servizio **serverless** astrae dal provisioning di risorse, che è del tutto affidato al cloud

- in particolare, l'utente non sa quali istanze svolgeranno i calcoli, né quante (spesso il servizio supporta l'*autoscaling* dinamico)
- p.es. AWS Lambda esegue funzioni (p.es. Python) definite dall'utente, che non deve preoccuparsi di nient'altro!

NB: poiché l'utente non ha cognizione delle risorse che supportano il servizio (in Lambda, p. es., possono variare da una richiesta all'altra), un servizio *serverless* deve provvedere interamente alla gestione delle risorse ed è quindi anche, necessariamente, (*fully*) *managed*

In servizi serverless, il costo per richiesta è assai più alto che creando delle istanze e installandovi/configurandovi il software necessario

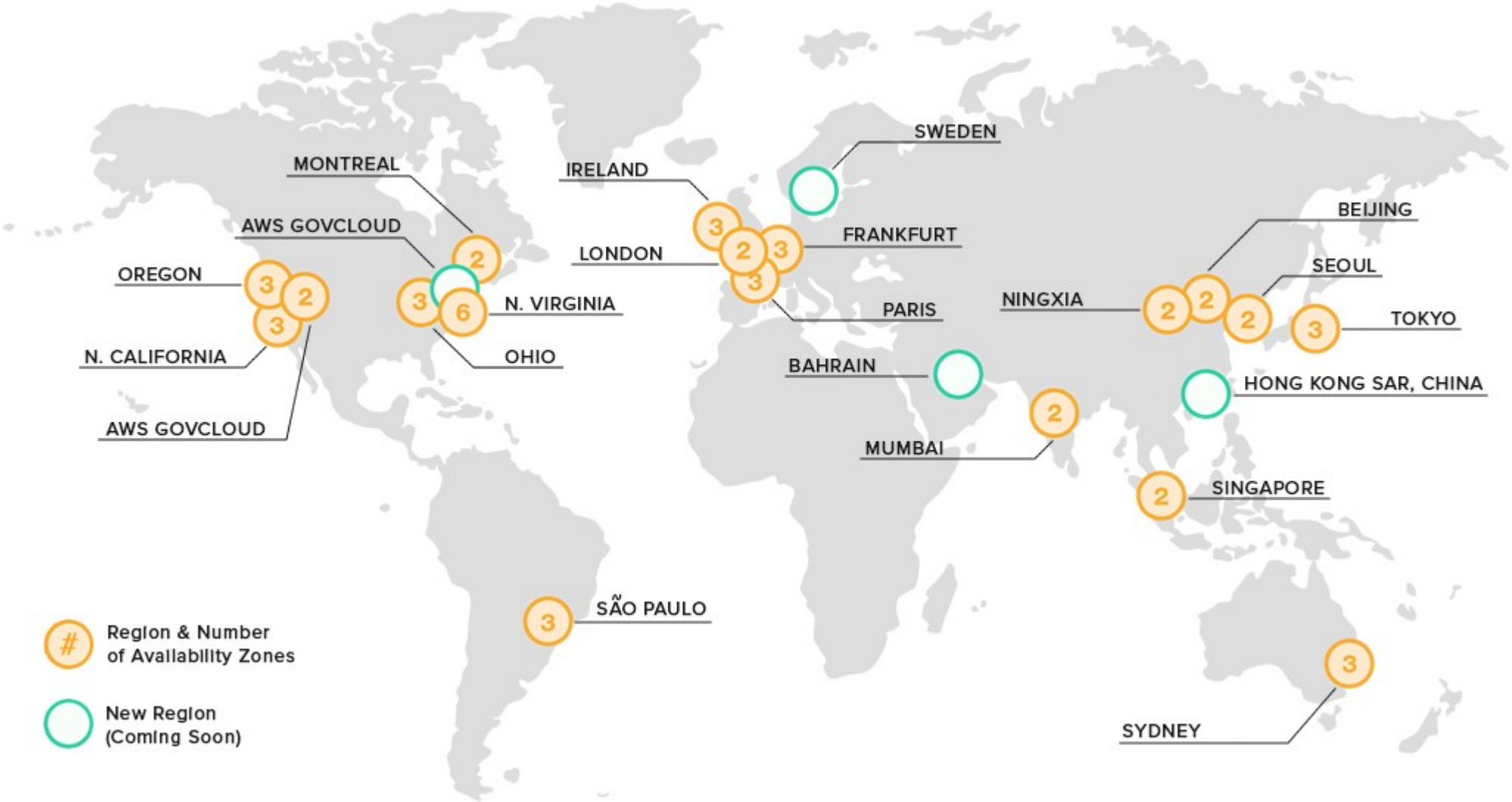
- … beninteso per un flusso di richieste (sempre) elevato
- per flussi limitati, il servizio serverless risulterà più conveniente

Modelli di public cloud secondo AWS

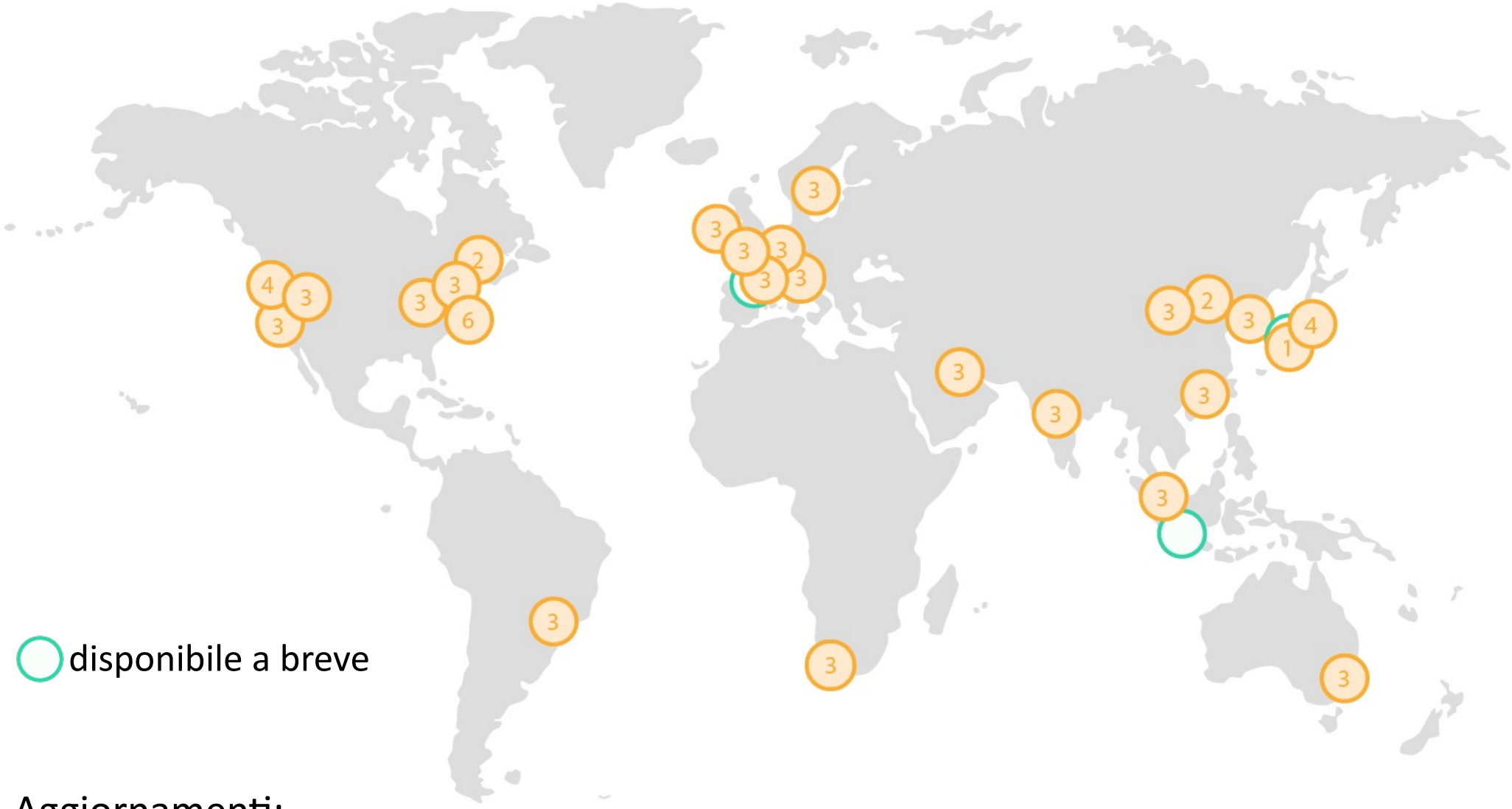
A conferma dei concetti illustrati finora, si veda anche l'inquadramento che ne fornisce la stessa AWS, alla URL:
https://aws.amazon.com/types-of-cloud-computing

E, per avere un'idea della figura professionale competente (assai richiesta), ecco il DevOps secondo Amazon AWS:
https://aws.amazon.com/devops/what-is-devops/?nc1=f_cc

Regioni AWS (2018, con nomi dei siti)



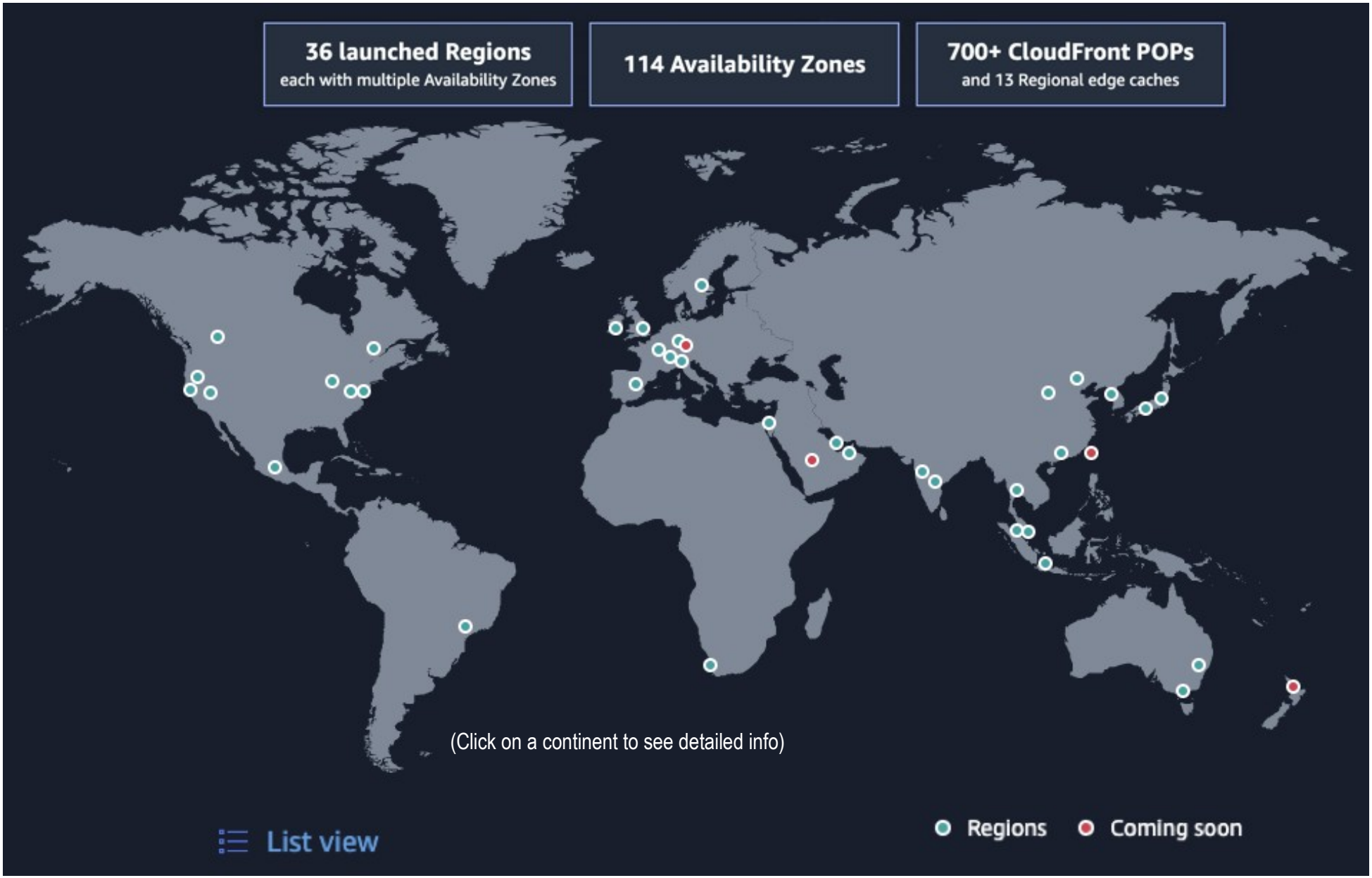
Regioni AWS (2020, senza nomi): in aumento!



Aggiornamenti:

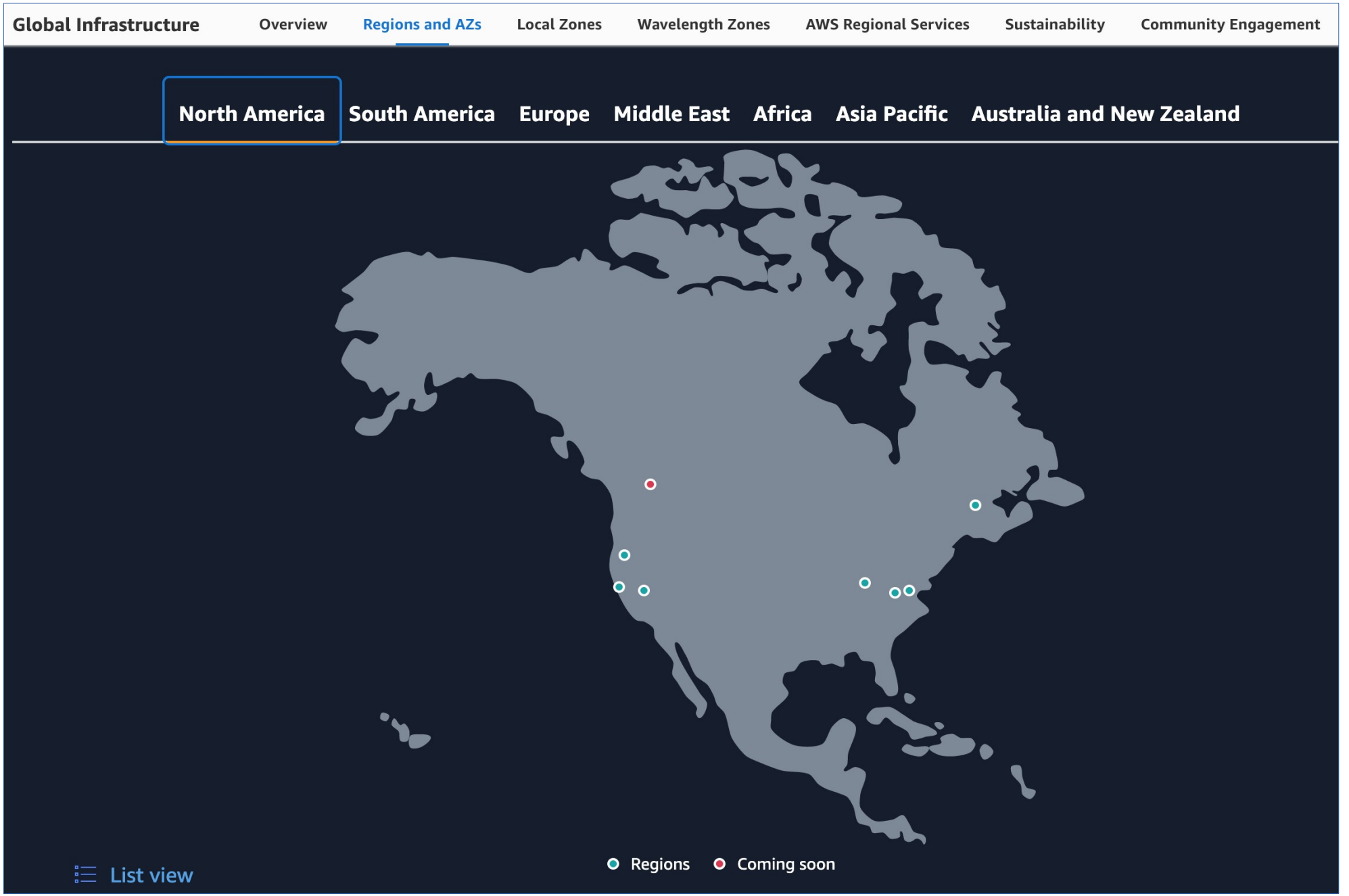
- https://aws.amazon.com/about-aws/global-infrastructure
- https://aws.amazon.com/about-aws/global-infrastructure/regions_az
- https://docs.aws.amazon.com/general/latest/gr/rande.html#regional-endpoints

aws.amazon.com/about-aws/global-infrastructure (2025)



Plans for 4 more Regions in New Zealand, Saudi Arabia, Taiwan, EU

AWS regions: https://aws.amazon.com/about-aws/global-infrastructure/regions_az



Regioni AWS

- I servizi AWS sono supportati da data center collocati in vari siti (AWS non pubblica le geolocazioni!)
- I siti AWS sono suddivisi, al livello più alto, in **regioni**
  - ogni sito, cioè, appartiene a una precisa regione
- Ogni regione corrisponde a una distinta **area geografica**
- I siti di una regione sono interconnessi dai **link ad alta velocità, privati** di AWS
- Grazie alle regioni, si possono posizionare le risorse (dati o calcolo) in modo ottimale rispetto agli utenti (latenza)
- L'elenco di regioni qui a destra (2022) è alla URL:

  docs.aws.amazon.com/general/latest/gr/rande.html#regional-endpoints

Region Name	Code
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

Regioni AWS: live

Qui a destra, l'elenco "live" delle regioni viene ottenuto da AWS stesso, via una shell Unix. **NB**:

- sono elencate solo le regioni disponibili per una data utenza
- occorre prima avere ottenuto l'accesso per la CLI (cliente testuale) aws (v. altra lezione)

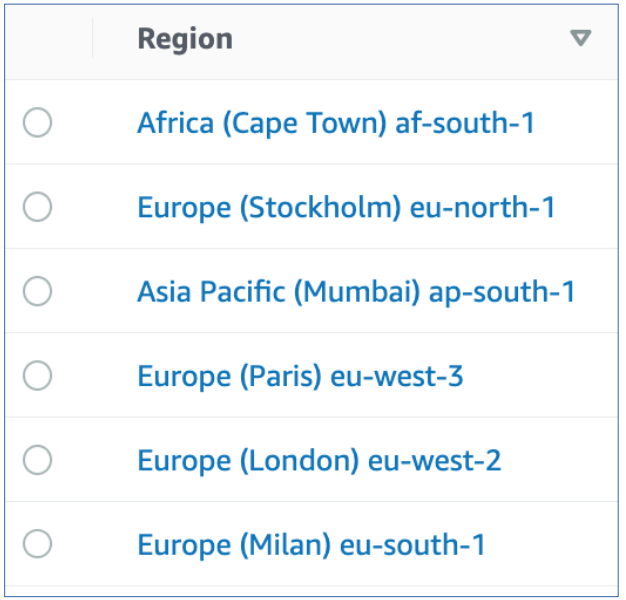
```
$  aws ec2 describe-regions --all-regions | \
 jq '.Regions.[].RegionName'
"ap-south-2"
"ap-south-1"
"eu-south-1"
"eu-south-2"
"eu-central-1"
...
“ap-southeast-4”
"us-east-1"
"ap-southeast-5"
"us-east-2"
"ap-southeast-7"
```

Oppure, vediamo qui a destra una (piccola) parte delle informazioni reperibili alla URL:

https://console.aws.amazon.com/ec2globalview

(in questo caso, serve l'accesso alla console Web di AWS)

Complessivamente, la pagina fornisce un utilissimo quadro delle risorse create dall'utente



Regioni e replicazione?

- **Replicare** (staticamente o dinamicamente) una risorsa, p.es. dei dati, è un accorgimento standard per garantire affidabilità/disponibilità:
  - ottimizzarne il **posizionamento** rispetto a un'utenza globale
  - supportare **disaster recovery**
- Potremmo quindi voler replicare, a questo scopo, le risorse AWS su più regioni
- In effetti, AWS fornisce supporto "nativo" alla replicazione delle risorse, ma solo **all'interno** di una regione, non **a cavallo** di più regioni
- Ogni regione AWS è isolata (non è cioè collegata alle altre via link privati o servizi AWS ad-hoc) *by design*:
  - ciò favorisce sicurezza (evita "sconfinamenti" interni ad AWS) e stabilità (evita possibili "effetti domino" da una regione all'altra)
- Un'organizzazione può però benissimo **replicare** autonomamente le proprie risorse su più regioni
  - In tal caso, però, il trasferimento/copia di una risorsa ad altra regione non è supportato da facility di AWS, bensì deve avvenire a cura del proprietario, e passare dalla Internet pubblica (link + lenti, - sicuri)

Criteri di scelta della regione

- **Minimizzare latenza per gli utenti finali**, prima di tutto
- **Costo** (per un dato servizio AWS varia con la regione, v. oltre)
- Per godere della **legislazione** desiderata, riguardo a protezione dei dati, consumer protection…

  Oppure perché si è obbligati a collocare i dati in una regione con una data **legislazione**
  - p.es., se si opera in Italia, i dati dovrebbero risiedere nella UE
- Se un'organizzazione supportata da AWS ha un'utenza **globale**, ma **diversificata** per area geografica, essa ricorrerà probabilmente a regioni AWS multiple, in modo da:
  - assicurare bassa latenza agli utenti delle varie regioni
  - consentire la specializzazione delle applicazioni per regione (p.es. in base a legislazione, cultura, etc.)

Criteri di scelta della regione / 2

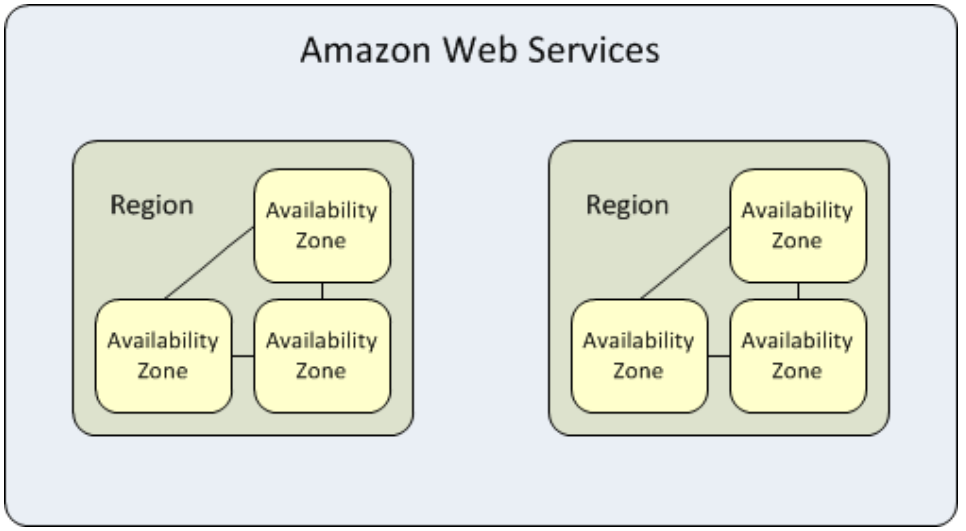
- Se l'utenza è **globale**, anche se non (troppo) diversificata, operare su regioni AWS multiple e replicare dati/elaborazione garantisce comunque migliori prestazioni e sicurezza, in termini cioè di:
  - bassa **latenza**
  - **high availability**
  - **disaster recovery**
- Ricordiamo di nuovo però che replicare dati e macchine *attraverso* la frontiera delle regioni non ha supporto "nativo", va fatto a mano, attraverso la Internet pubblica.

  Citando la stessa documentazione AWS:

  > "*… after taking a snapshot of your existing volume, you can copy the volume to a separate region and attach it to a new instance, [thus]*
  >
  > - *increasing availability … and …*
  > - *achieving disaster recovery capability*"

  (v. anche https://docs.aws.amazon.com/en_us/AWSEC2/latest/UserGuide/ebs-copy-snapshot.html)
- How-to: https://n2ws.com/blog/how-to-guides/copying-snapshots-to-different-regions-to-achieve-ha

Availability zones

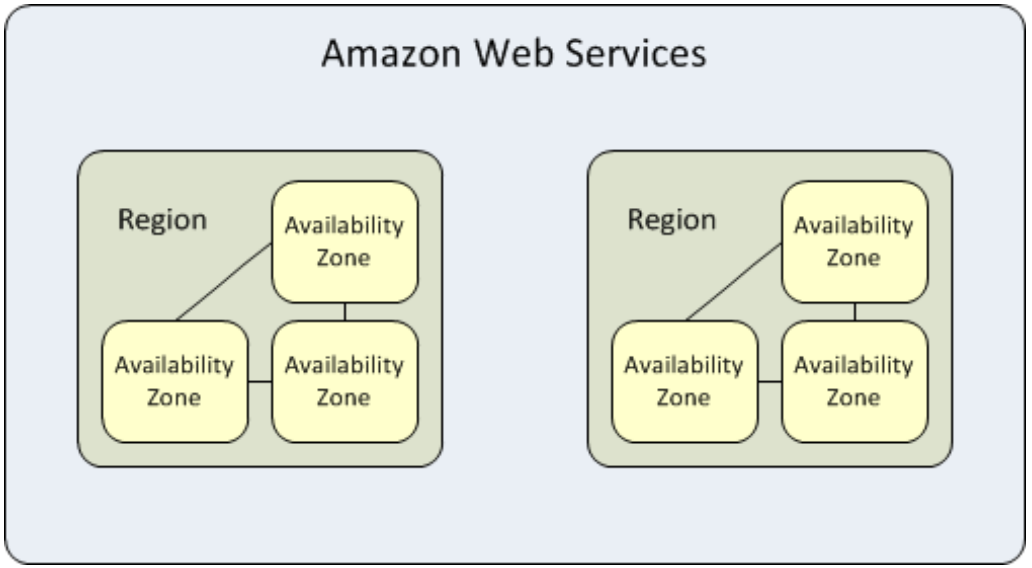


- Ogni regione è del tutto indipendente dalle altre
- Ogni regione è composta di multiple locazioni distinte, dette ***Availability Zones*** **(AZ)**
- Ogni Availability Zone è isolata rispetto alle altre regioni, ma le AZs in una regione sono connesse tra loro con link proprietari a bassa latenza
- Ogni istanza (macchina virtuale) è collocata in una AZ, scelta dall'utente (o in mancanza da AWS) quando l'istanza stessa viene lanciata
- Se delle istanze replicate vengono distribuite tra più AZ (in una regione), si può progettare un'applicazione in modo che, se l'instanza di una AZ fallisce, un'instanza in un'altra AZ possa intervenire a gestire le richieste degli utenti

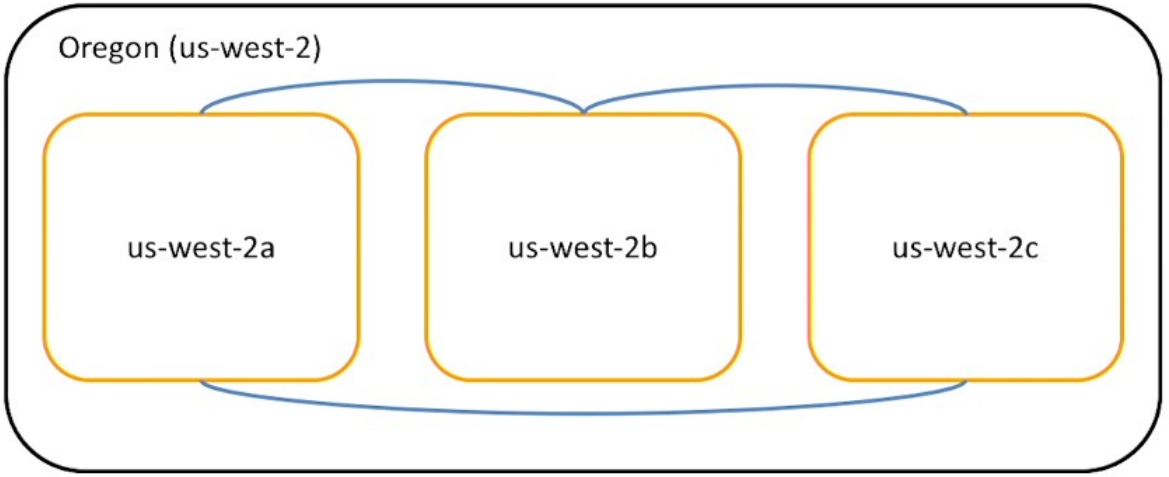
Availability zones / 2

- Ogni AZ ha alimentazione e connettività di rete indipendente
- Ciò rende improbabile che due AZ falliscano insieme
- Quanti Data Center supportano una AZ?
  - spesso si dice che ogni AZ ha dietro un singolo Data Centre
  - ma una AZ può, e dovrebbe, contenere più Data Centre collegati da linee private in fibra a bassa latenza e alto throughput

    (beninteso, mentre una AZ può contenere Data Centre multipli, nessun Data Centre è condiviso tra due AZ distinte!)
- I link veloci intra-AZ (e inter-AZ) hanno un ruolo chiave nell'assicurare ridondanza e alta disponibilità dei dati affidati ad AWS (S3, ma anche servizi database fully-managed e serverless)



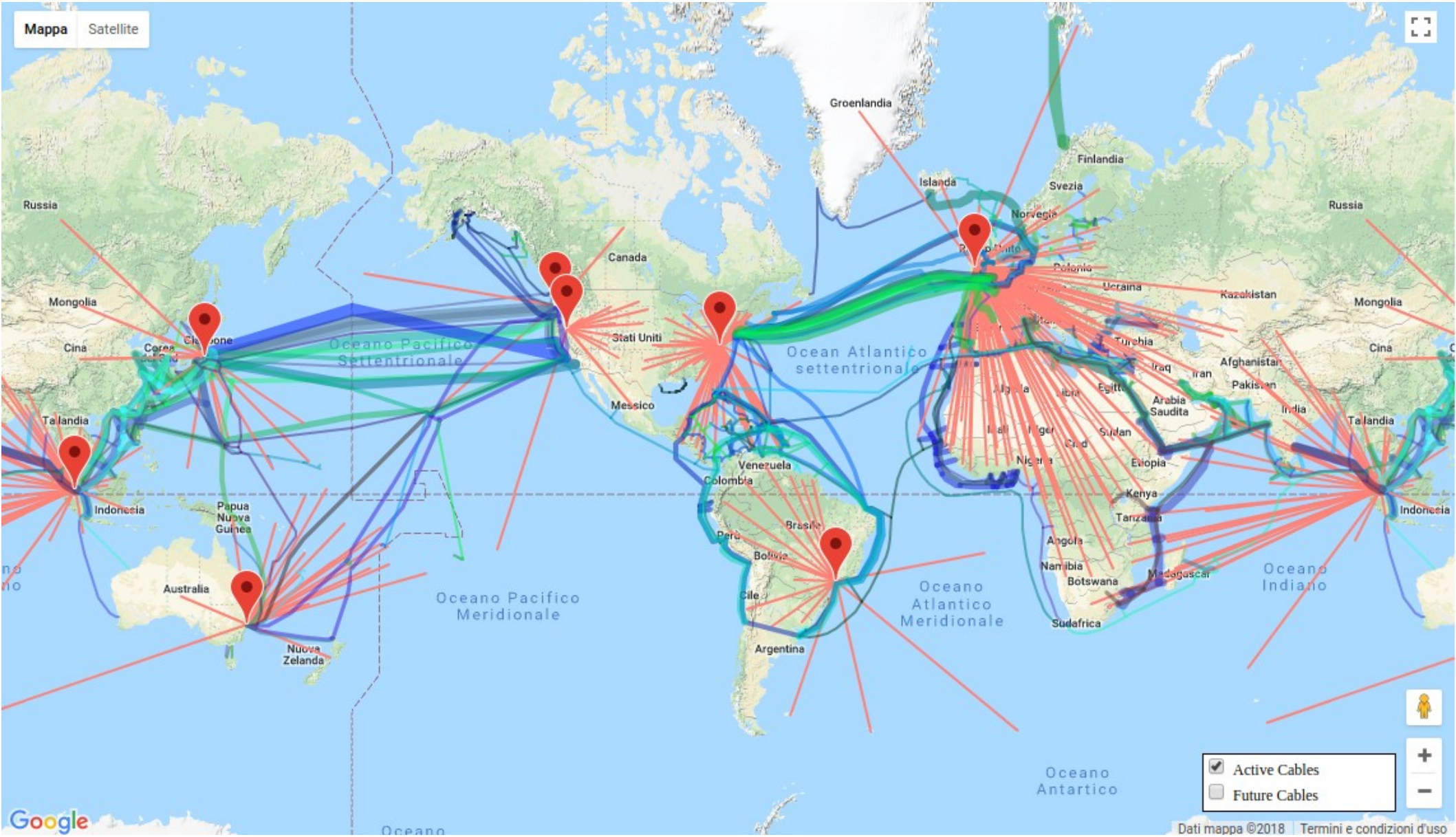
Availability **Z**ones: cosa sappiamo



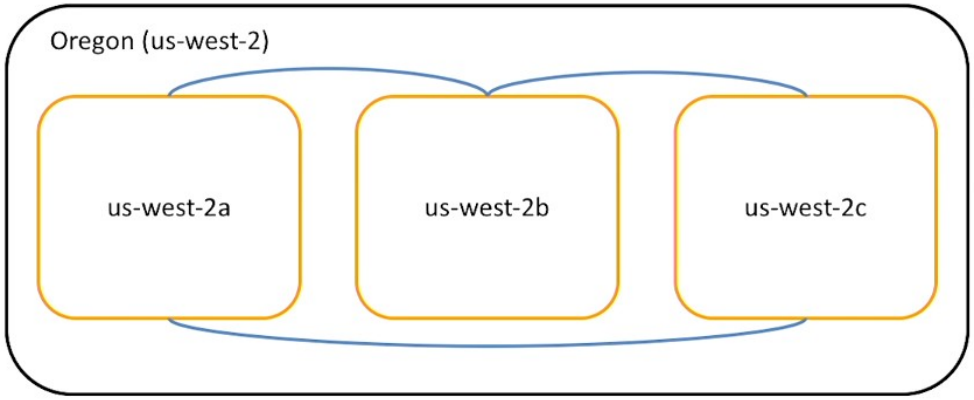
- Una AZ è denotata dal codice di regione + una lettera, p.es. **us-west-2a**
- Fisicamente, come detto, una AZ ha dietro un set di Data Centre
- **NB:** la AZ *us-west-2a* di un account può non essere "fisicamente" la stessa AZ *us-west-2a* di un altro account (cioè i rispettivi data centre sono diversi)
  - non c'è modo di "coordinare" le AZ tra account diversi: non ho modo di sapere se la mia *us-west-2a* è la tua *us-west-2a*, *us-west-2b* o *us-west-2c*
- La mappa delle AZ è "oscurata" per favorire il **load balancing**, ma anche per ragioni di **security**, infatti così...
  - si evita che, per abitudine o malizia, tutti attivino risorse in *us-west-2a*, sovraccaricandola!

Data Center AWS: cosa possiamo scoprire (con ping etc.)

http://turnkeylinux.github.io/aws-datacenters/



Availability **Z**one: HA e FT



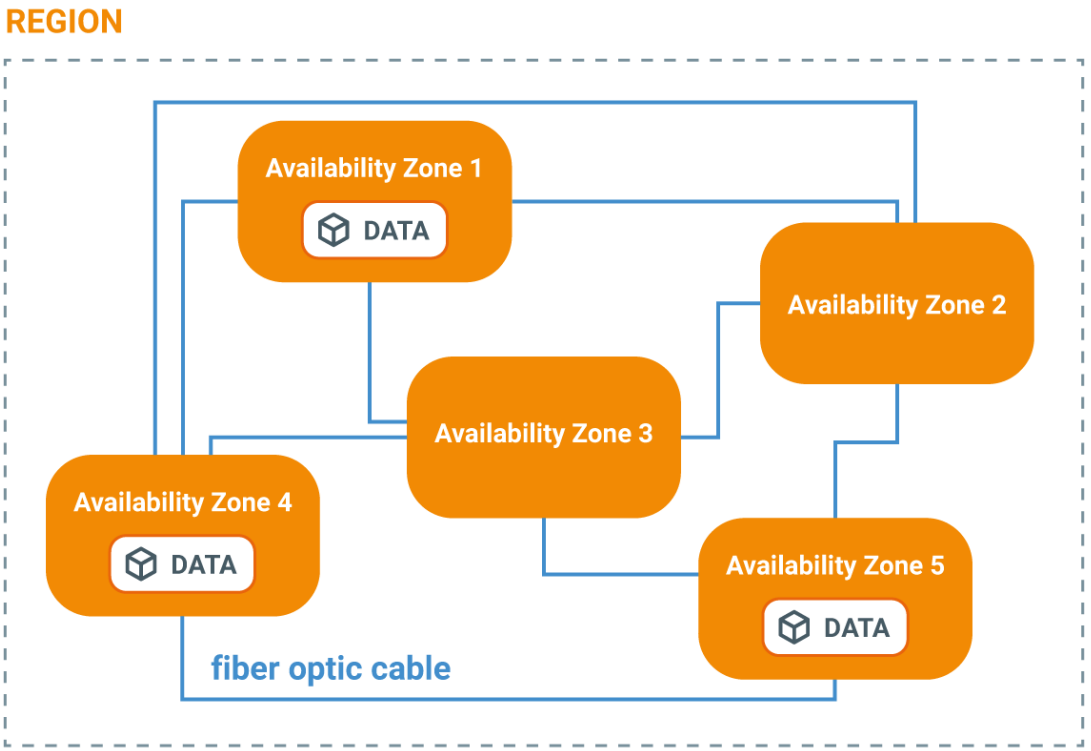
- Come detto, grosso modo:
  AZ = Uno o più Data Center
- I siti per le AZ sono scelti in base
  a considerazioni di fault tolerance:
  - bacini fluviali (inondazioni!) diversi
  - faglie tettoniche (terremoti!) diverse
  - reti elettriche (black out!) distinte
  - distanti (entro i 100 km)
- Le AZ sono collegate da fibra privata di Amazon
- Ogni regione ha ≥2 AZ (vedi mappa)

In sintesi, lasciamo la parola a AWS:

> Availability Zones are the key to *High Availability* (HA) and Fault Tolerance (FT), through resilience even to Data center loss

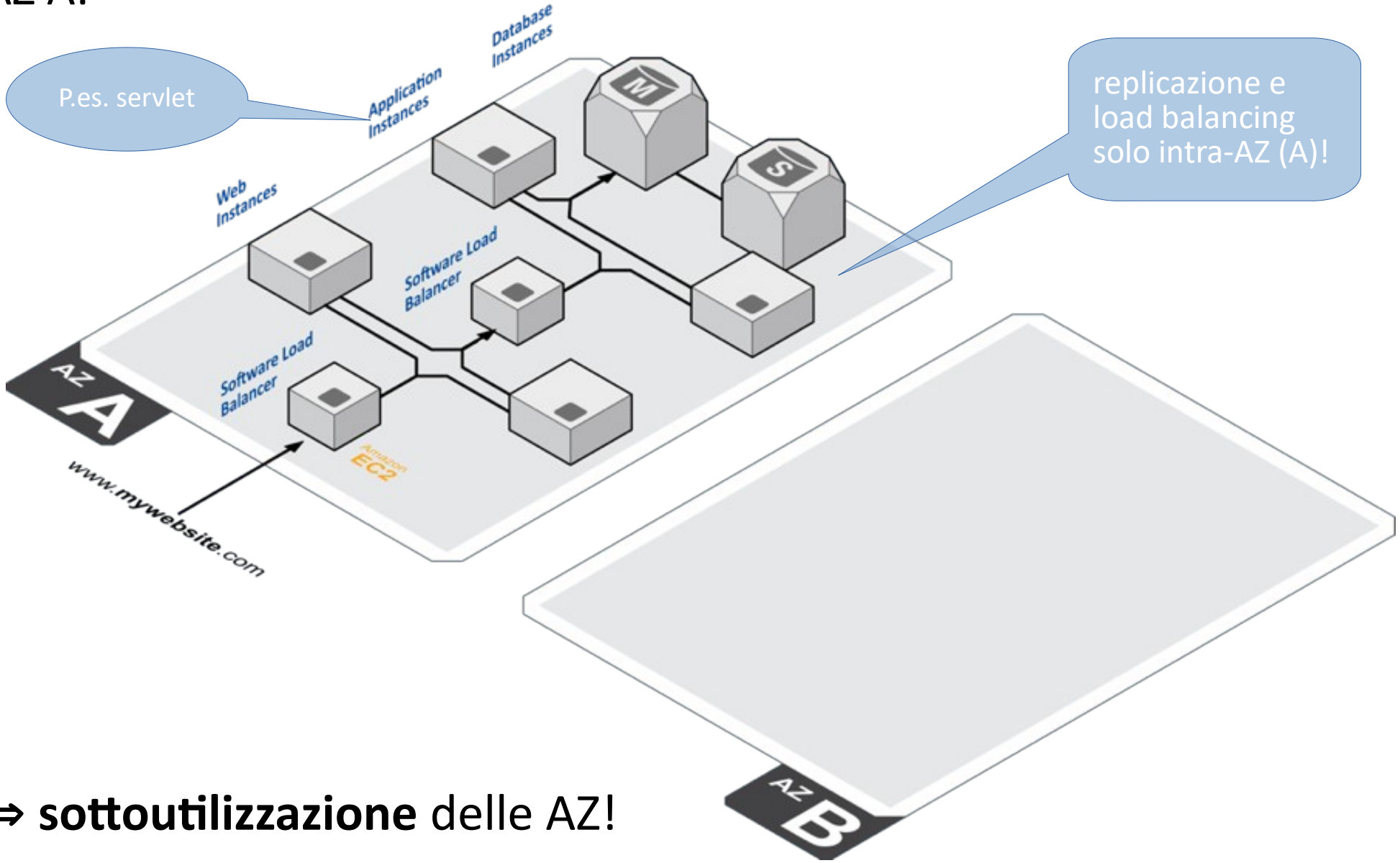
Availability Zones e dati

- Le Availability Zone (AZ) sono quindi (gruppi di) data center interconnessi via link ottici a bassa latenza
- Il servizio AWS di storage S3, per default, replica i dati all'interno di almeno 3 AZ di una stessa Region:
  - anche se un'intera AZ crollasse, comunque i dati resterebbero disponibili



Architettura cloud: 3-tier classico

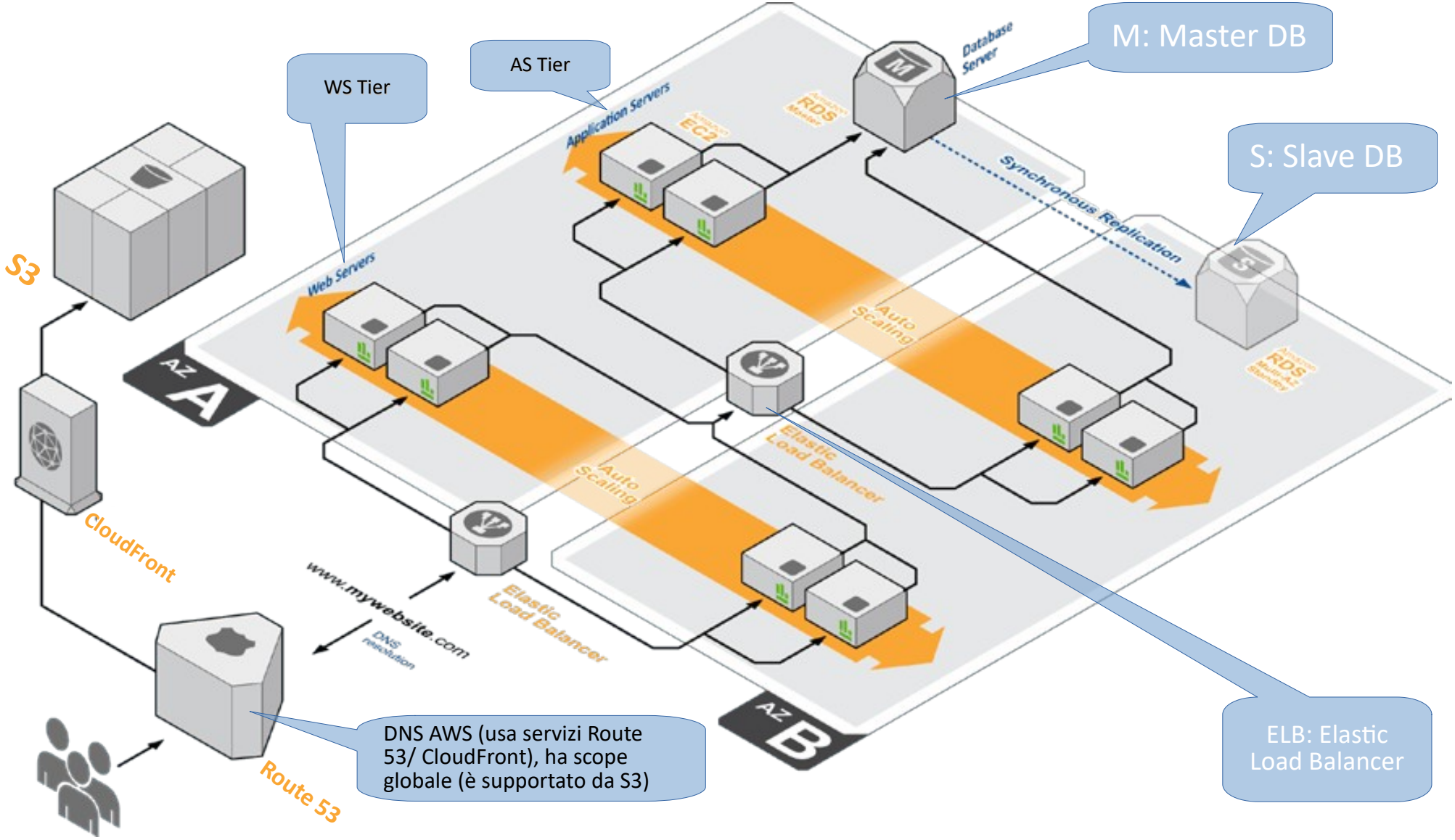
Il tier 2 (Web server + Engine) e il tier 3 (DB) sono concentrati nella AZ A:



⇒ **sottoutilizzazione** delle AZ!

Architettura cloud che sfrutta le AZ

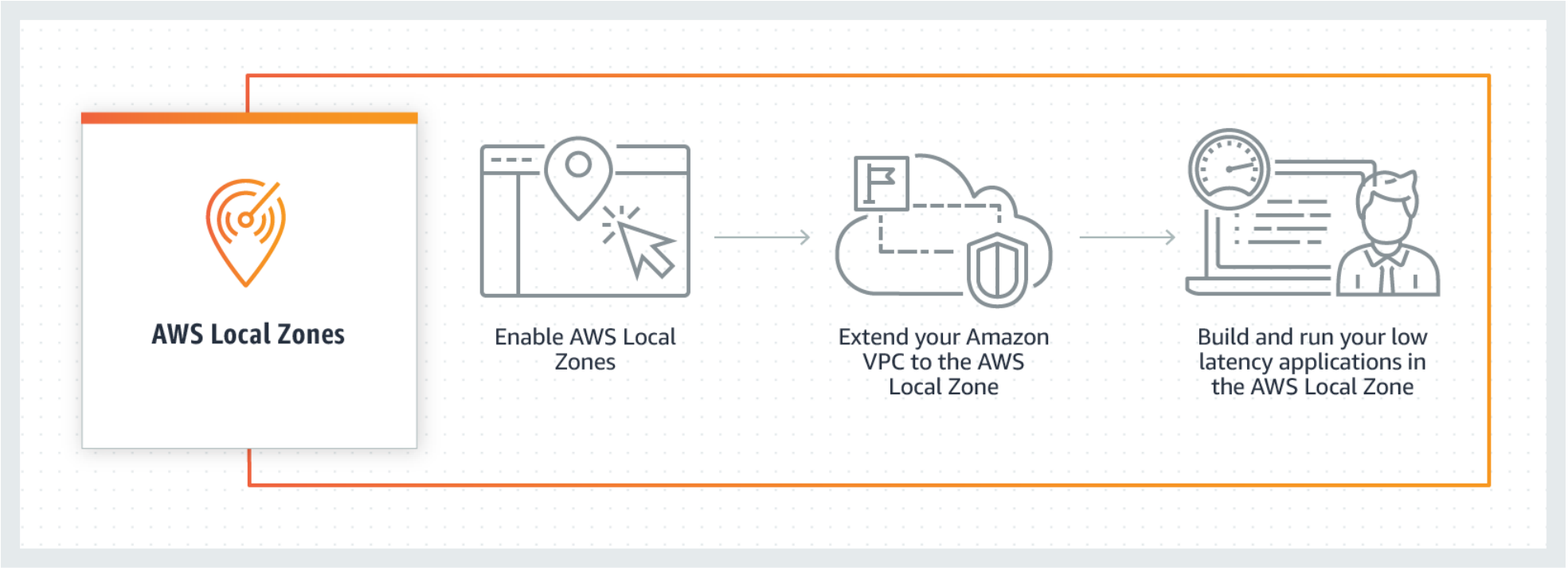
- Un ELB *cross-zone* ha un sub-ELB in A e uno in B, quindi è immune al fallimento di, p.es., A
- L'ELB di ciascun tier (WS/AS), in caso di fallimenti di istanze o intera AZ, instraderà le richieste verso istanze del tier ancora vive



New in AWS: Local Zones

Da https://aws.amazon.com/about-aws/global-infrastructure/localzones/

AWS Local Zones are a type of infrastructure deployment that places compute, storage, database, and other select AWS services close to large population and industry centers… for ultralow-latency applications



NB:

- *Local Zones*: collocazione nota
- *Availability Zones*: collocazione (precisa) ignota

New: AWS Local Zones

https://aws.amazon.com/about-aws/global-infrastructure/localzones/locations/

> *… We now have a total of 33 Local Zones; 16 outside of the US… We will continue to expand Local Zones to 20 metro areas in 17 countries, including Australia, Austria, Belgium, Brazil, Canada, Colombia, Czech Republic, Germany, Greece, India, Kenya, Netherlands, Norway, Philippines, Portugal, South Africa, Vietnam*

Alcuni esempi (notare gli *Zone Name*):

Querétaro, Mexico
Zone Name: us-east-1-qro-1a
Parent Region: US East (N. Virginia)

Santiago, Chile
Zone Name: us-east-1-scl-1a
Parent Region: US East (N. Virginia)

Seattle, US
Zone Name: us-west-2-sea-1a
Parent Region: US West (Oregon)

Amsterdam, Netherlands
Zone Name: eu-central-1-ams-1a
Parent Region: Europe (Frankfurt)

Athens, Greece
Zone Name: eu-south-1-ath-1a
Parent Region: Europe (Milan)

Bengaluru, India
Zone Name: ap-south-2-blr-1a
Parent Region: Asia Pacific (Hyderabad)

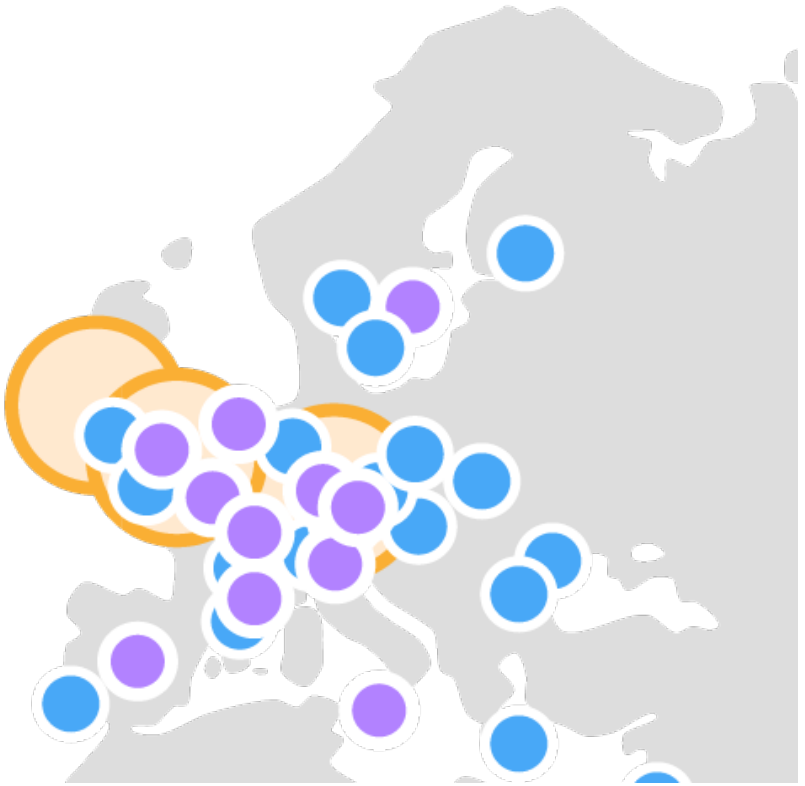
Se è sufficiente mantenere soltanto i dati vicino agli utenti (piuttosto che i servizi nel loro complesso), si può ricorrere a un approccio più tradizionale: una CDN, cioè AWS *CloudFront*

AWS: dati e CloudFront

- Un'organizzazione può riporre i propri dati nel cloud AWS, anziché in S3, presso **CloudFront**, il servizio CDN Content Delivery Network di AWS
- I dati di CloudFront sono nelle *Edge Location*, data center più piccoli di quelli che supportano le AZ, mirati alla distribuzione di contenuti a bassa latenza
- Edge Location sono più numerose e sparse di Local Z./AZ/regioni, p.es. in EU le LZ sono 1+7 (annunciate), le regioni sono 8, con tre AZ ciascuna, mentre:

Edge / Multiple Edge locations: Frankfurt am Main (17); Düsseldorf (3); Hamburg (6); Munich (4); Berlin (5); Barcelona (2); Madrid (10) Paris (11); Marseille (6); Milan (9); Rome (6); Palermo (1); Amsterdam (5); Manchester (5); London (25); Dublin (2); Vienna (3); Stockholm (4); Copenhagen (3); Helsinki (4); Athens (1); Brussels (1); Budapest (1); Lisbon (1); Oslo (2); Bucharest (1); Prague (1); Sofia (1); Warsaw (3); Zagreb (1); Zurich (2)

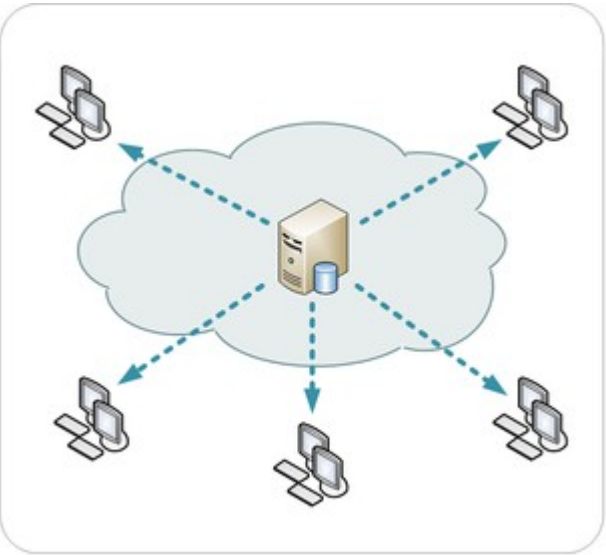
Regional Edge caches (v. anche oltre): Dublin, Ireland; Frankfurt, Germany; London, England



AWS **E**dge **L**ocations (1): CDN PoPs

Le Edge Locations di AWS hanno in effetti due scopi: il primo, come già detto:

1. come punti di accesso (POP, Point Of Presence) per l'utenza più vicina, **supportare *CloudFront***, la CDN (Content Delivery Network) di Amazon
   - una CDN migliora *latenze* e *throughput*, attraverso *caching* e *rilocazione* dei dati, collegando gli utenti all'Edge location ottimale, tra quelli disponibili
   - cf., nelle figure qui a destra, Edge locations mutiple vs. unico repository
   - applicazioni: si pensi a servizi come Netflix, Hulu o Amazon Prime Video



2. supportare Amazon Route 53...

AWS **E**dge **L**ocations (2): Route 53

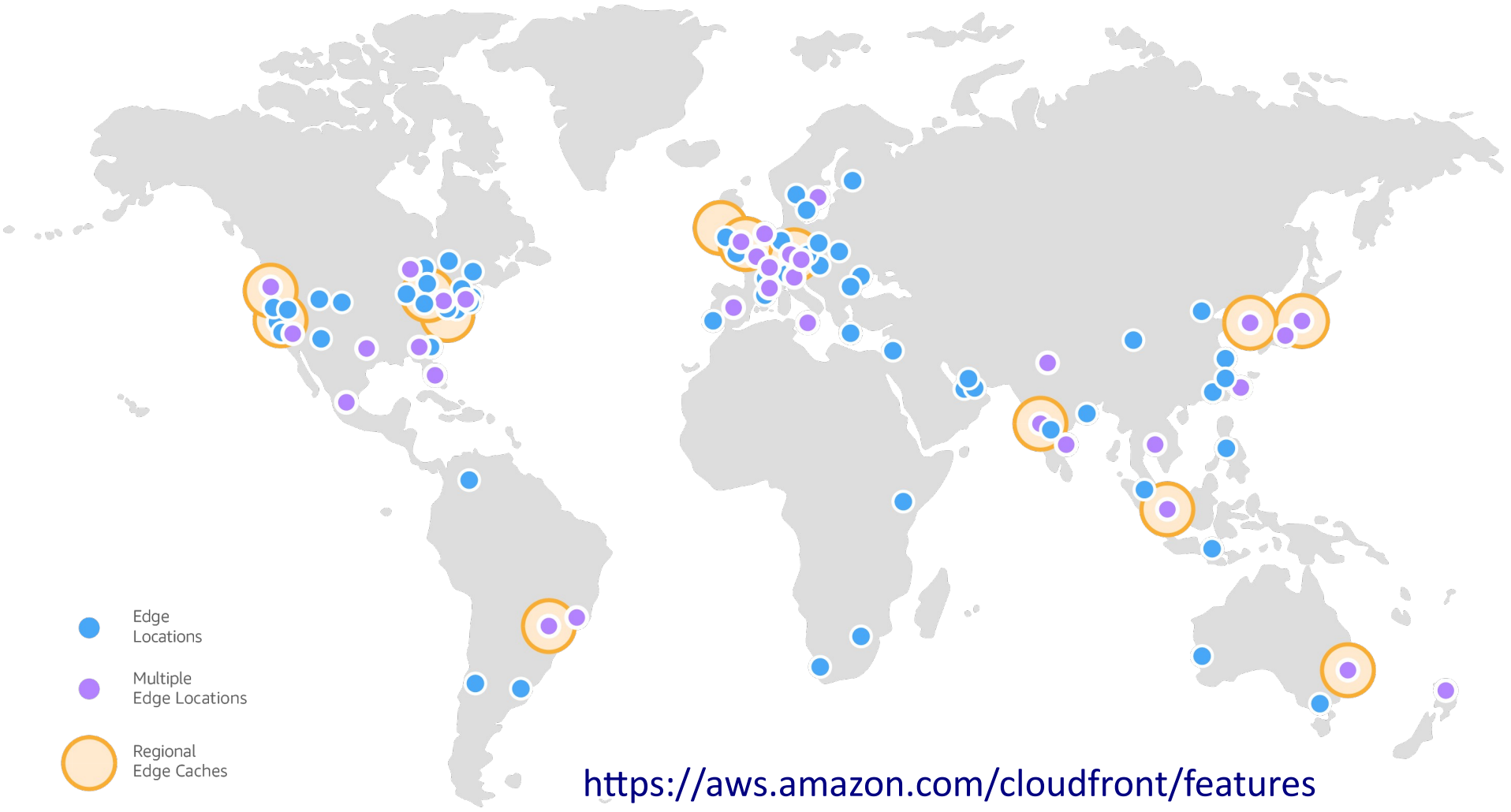
Le Edge Locations hanno due scopi:

1. supportare *CloudFront*...

2. **supportare** Amazon ***Route 53***, il DNS globale di AWS, per ottimizzare le query DNS riguardanti AWS
   - Con Route 53 di AWS, un'organizzazione/utente può definire i propri record DNS

sull'etimologia di *Route 53* v. https://en.wikipedia.org/wiki/Amazon_Route_53:
"The name is a possible reference to U.S. Routes, and "53" is a reference to the [DNS] TCP/UDP port 53"

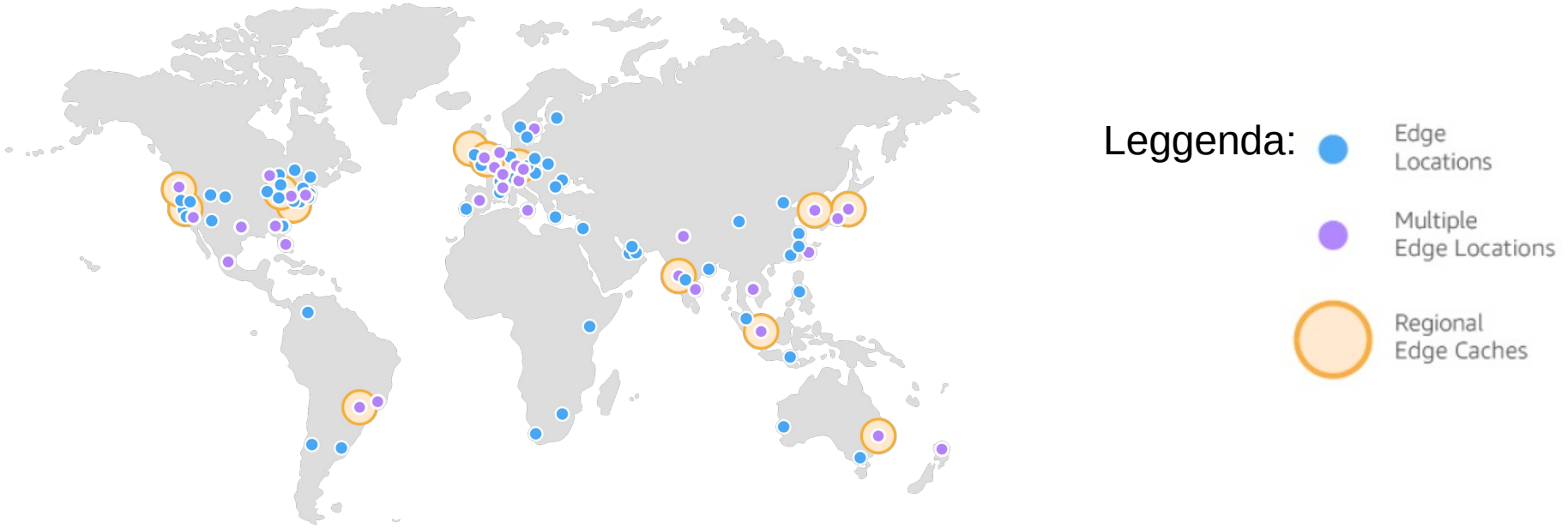
AWS *CloudFront*: Edge Locations (2022)



https://aws.amazon.com/cloudfront/features

- Vi è in effetti una gerarchia di Edge location CloudFront:
- Non ogni Region dispone di tutte le forme di Edge Location (es. Africa)
- L'utente non può scegliere (la politica di allocazione dei contenuti su) le Edge Location, ma, al più, l'area geografica (p.es. Asia/Europa/America)

CloudFront 2022: organizzazione



Le Edge Location in aree con utenza più numerosa fanno capo a una Regional Edge Cache

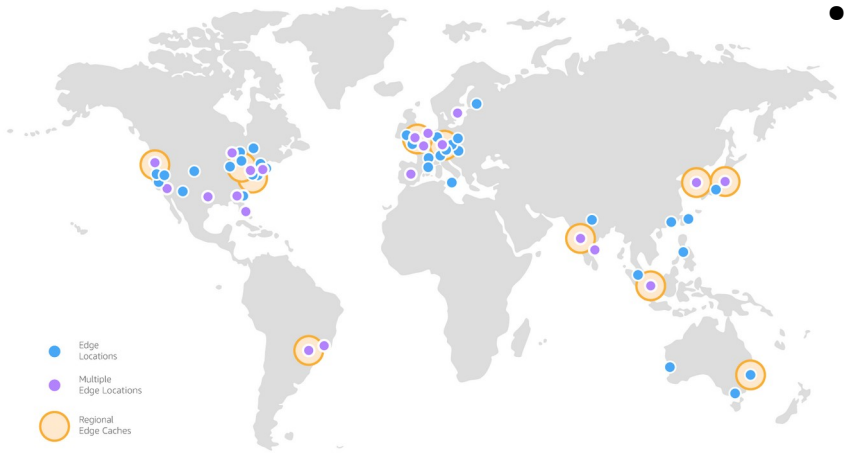
Le Regional Edge Caches memorizzano i dati più a lungo e fanno da cache “mid-tier” per le (più piccole) edge location

Insieme, edge location (300+) ed edge cache (13) sono detti PoP (Points of Presence):

- si trovano in oltre 90 città in 47 paesi
- sono connessi con le regioni AWS attraverso la rete backbone di AWS in fibra ridondata a 100Gb/s
- sono collegati alle reti dei principali operatori di rete (p.es. Tim, Wind, Fastweb...)

Ciò consente di distribuire con bassa latenza contenuti agli utenti finali

Amazon *CloudFront* sites (2018)



- **North America Edge Locations**: Ashburn, VA (3); Atlanta, GA (3); Boston, MA; Chicago, IL (2); Dallas/Fort Worth, TX (4); Denver, CO; Hayward, CA; Jacksonville, FL; Los Angeles, CA (3); Miami, FL (2); Minneapolis, MN; Montreal, QC; New York, NY (3); Newark, NJ (2); Palo Alto, CA; Philadelphia, PA; Phoenix, AZ; San Jose, CA; Seattle, WA (3); South Bend, IN; St. Louis, MO; Toronto, ON; **Regional Edge Caches**: Virginia; Ohio; Oregon
- **South America Edge Locations:** São Paulo, Brazil (2); Rio de Janeiro, Brazil (2); **Regional Edge Caches:** São Paulo, Brazil
- **Europe Edge Locations**: Amsterdam, The Netherlands (2); Berlin, Germany; Dublin, Ireland; Frankfurt, Germany (6); Helsinki, Finland; London, England (5); Madrid, Spain (2); Manchester, England; Marseille, France; **Milan**, Italy; Munich, Germany; **Palermo**, Italy; Paris, France (3); Prague, Czech Republic; Stockholm, Sweden (3); Vienna, Austria; Warsaw, Poland; Zurich, Switzerland; **Regional Edge Caches**: Frankfurt, Germany; London, England
- **Asia Edge Locations**: Chennai, India (2); Hong Kong, China (3); Kuala Lumpur, Malaysia; Mumbai, India (2); Manila, Philippines; New Delhi, India; Osaka, Japan; Seoul, South Korea (4); Singapore (2); Taipei, Taiwan; Tokyo, Japan (7); **Regional Edge Caches**: Mumbai, India; Singapore; Seoul, South Korea; Tokyo, Japan
- **Australia Edge Locations**: Melbourne; Perth; Sydney; **Regional Edge Caches**: Sydney

AWS: *scope* dei servizi

I servizi AWS hanno tre possibili (livelli di) *scope* (alternativi):

- **globale**, l'utente non sceglie l'ambito
  - IAM (Identity and Access Management (utenti/permessi)), CloudFront, Route 53, ...
- **regionale**, l'utente sceglie la regione per il servizio cui accede
  - DynamoDB, SimpleStorage (bucket), Elastic Load Balancing, Virtual Private Cloud...
  - servizi intrinsecamente High Availability, Fault Tolerance, grazie alle AZ multiple disponibili (automaticamente) nella regione
- **Availability Zone**, l'utente sceglie la *availability zone* (AZ)
  - Elastic Compute Cloud (EC2), Elastic Block Store (EBS), subnetting...
  - N.B.: *scope consistency* a livello di AZ
    - se si attiva un'istanza EC2 (una VM) in una AZ, e le si attacca un volume EBS (HD virtuale), questo deve vivere nella stessa AZ

AWS: outage (28/2/2017)

- Il cloud **non** è invulnerabile, si veda:
  https://www.impresacity.it/news/8329/un-errore-umano-la-causa-del-blackout-di-aws.html
- Il 28/2/17 la regione Us-East-1 ha perso quasi ogni operatività per un "banale" errore umano: lo “spegnimento” temporaneo di una serie sbagliata di server
  - Il team di S3, a scopo di debugging, ha lanciato un comando per scollegare pochi server da uno dei sottosistemi S3 utilizzati per la fatturazione
  - uno dei comandi fu inserito in modo scorretto, causando la rimozione di ben più server, compresi alcuni che supportavano altri due sottosistemi S3
  - uno di questi, incaricato di gestire l’indice, processava i metadati e le informazioni di localizzazione di tutti gli oggetti S3 presenti nella regione Us-East-1, elaborando tutte le richieste Get, List, Put e Delete.
- Risolvere la perdita di “una porzione significativa della capacità ha richiesto un riavvio completo dei sistemi”, ha scritto Amazon Web Services
- Ovviamente, durante l’inattività dei server, S3 non è riuscito a servire le richieste, bloccando di fatto anche altri servizi dipendenti dallo storage cloud di AWS, tra cui EC2, EBS, Lambda e la stessa dashboard web di AWS
- Una vicenda che sottolinea l’importanza di adottare strategie multicloud, che permettono, in casi come questo, di limitare fortemente i danni.

Per essere agnostici…: GCP

https://gizmodo.com/google-cloud-pension-fund-unisuper-1851476649

Google Accidentally Deleted $125 Billion Pension Fund's Account

About half a million customers of the Australian fund UniSuper were locked out of their accounts for a week

By Laura Bratton, Quartz Published Tuesday 2:35PM | Comments (5)

The Google Cloud logo at their booth at the Hannover Messe 2024 trade fair in Hannover, Germany. Photo: Krisztian Bocsi/Bloomberg (Getty Images)