

# Cloud Computing

# Course plan

Learning session 2

Core Services

# Disclaimer

- Most Clouds completely abstract *real* infrastructure
- End-users deal with *services* on top of real infrastructure

# “Core Services” lecture plan

- Compute
- Storage
- Network
- Databases
- Events and Messaging

# Compute

Compute is ...

- infrastructure capable to run your workload/application
- CPU + memory

# Compute

Compute could be based on:

- *raw* Virtual Machines (VM)
- Functions (Function as a Service)
- Containers
- Mixed/Hybrid

# Compute

## Virtual Machine

- Configurable CPU, RAM, size/type of storage, network bandwidth
- Control OS components and configuration
- Create snapshots/images

Virtual Machines / EC2 / Compute Engine

# Compute

## Functions

- Run **code** in response to event (http or from other services)
- No VMs to manage
- Faster scale-out
- (often) Pay only for running time (ms)

Azure Function / AWS Lambda / Cloud Function



# Compute

## Containers

- Run single **container** without orchestrator
- No VMs to manage

Azure Container Instance / AWS Fargate / Cloud Run

# Compute

Kubernetes:

- Run **multiple containers on** (partially) cloud-managed **Kubernetes**
- Integrated with other cloud services
- (mostly) based on VMs

AKS / EKS / GKE

# Compute

Application-optimized:

- **VM** optimized for concrete app
- Pre-installed applications/services
- Often includes other cloud services (monitoring, dns, and others)
- Cloud-provider offers marketplace

App Service / Lightsail or Elastic Beanstalk / App Engine

# Compute

Batch:

- **VMs** optimized for high performance computing

Azure Batch / AWS Batch / Preemptible Virtual Machines

# Compute

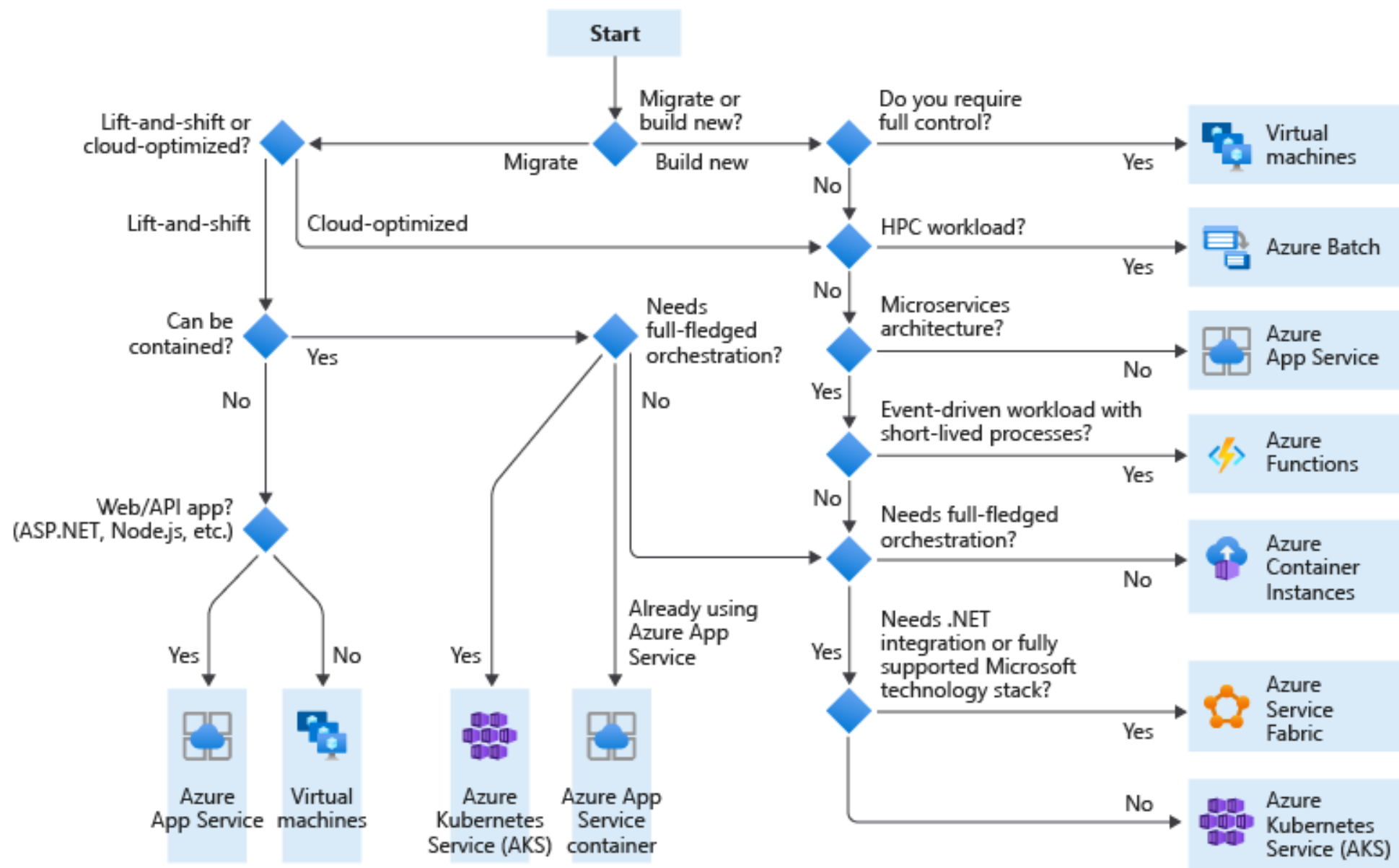
Other:

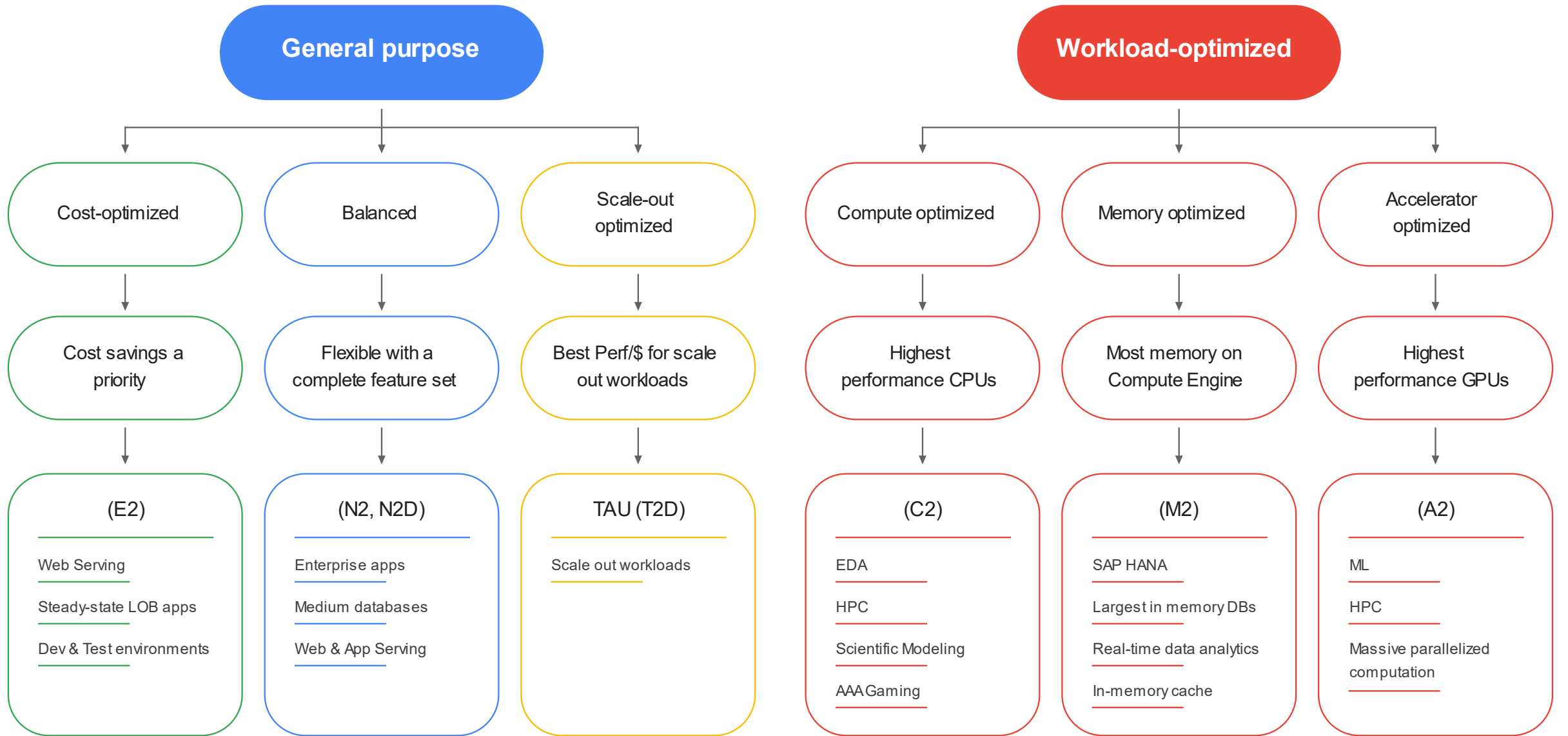
- [Service Fabric](#): distributed systems platform (azure's k8s)
- ???

# Compute

How to choose compute service?

- [Azure](#)
- [AWS](#)
- [GCP compute](#)







Compute

Demo

# Compute Summary

- **Compute** is an infrastructure capable to run workload/application
- Main types:
  - VM
  - Function
  - Container

# Storage

Storage is an infrastructure capable to store your data

# Storage

## Types of storage resources

- Block
- File system
- Object

# Storage

## Block Storage:

- Specific location on disks/memory
- Network access to raw block devices
- Each block could be referenced in different OS and partitioned independently

(Solid State Drive)



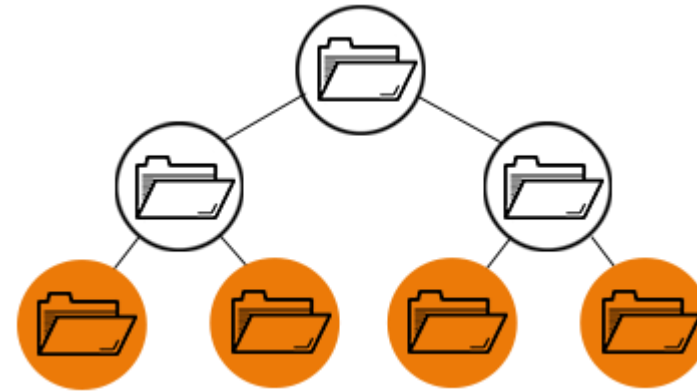
(Hard Disk Drive)



# Storage

## File Storage:

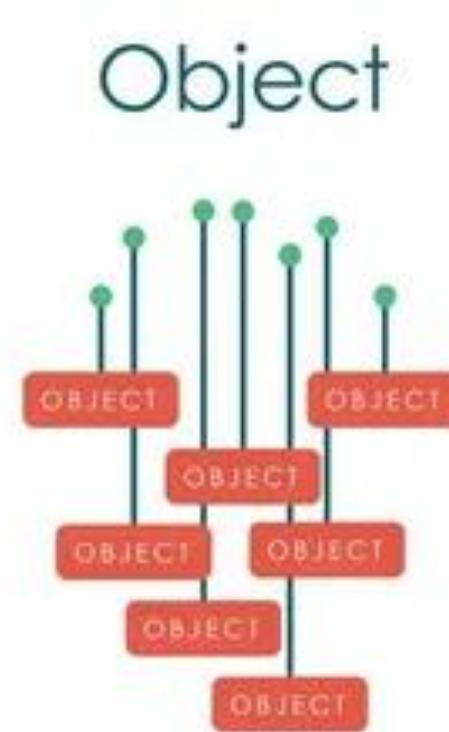
- Shared filesystem
- Folder structure
- File path and name



# Storage

## Object Storage:

- Flat-structure like key-value store
- Each object has associated metadata and Id
- Immutable objects: written or replaced
- Often accessed via (REST) API
- Often allows aggregation into buckets/containers



# Storage

AWS offerings:

- Block Storage: **EBS** (Elastic Block Storage)
- File System: **EFS** (Elastic File System)
- Object Storage: **S3** (Simple Storage Service) with multiple replication types and access tiers ([Storage Classes](#)) including **Glacier**

Bonus: **Snow Family**



# Storage

Azure offerings:

- Block Storage: [Azure Disk](#)
- File System: [Azure File](#) or [One Drive](#)
- Object Storage: [Storage Account](#), including [Blob](#), [Queue](#), [Table](#), [File Share](#) (File System)
  - Has multiple access tiers and replication modes

# Storage

GCP offerings:

- Block Storage: [Persistent Disk](#)
- File System: [Filestore](#) or [Google Drive](#)
- Object Storage: [Cloud Storage](#) with multiple access tiers ([Storage Classes](#)) and redundancy options

# Storage

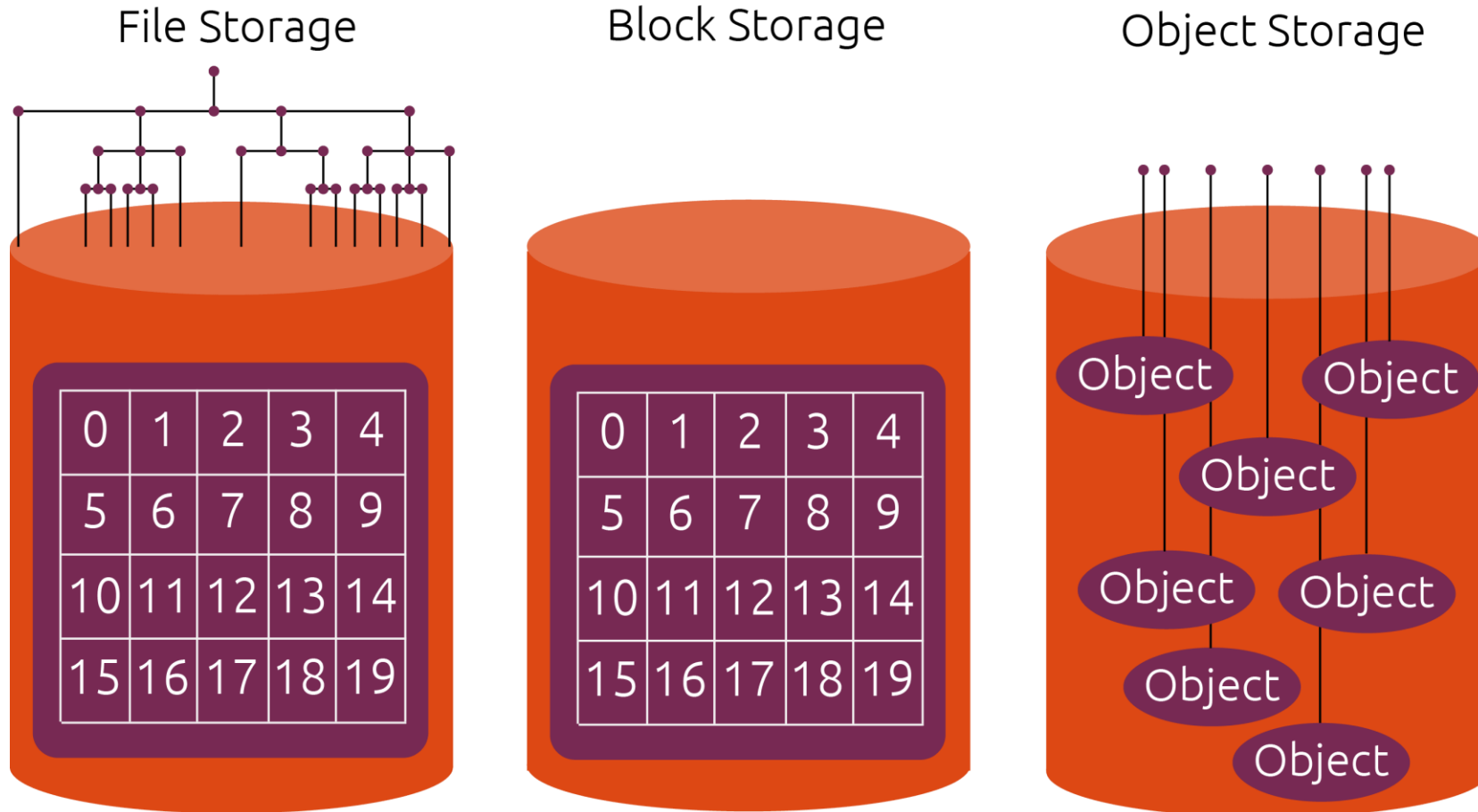
How to choose:

- Who uses the storage
- Usage patterns and performance: access frequency, size of data, size of files
- Pricing
- Access control

Storage

Demo

# Storage Summary



# Storage

## Examples

- VM disk
- DB Backup
- A bunch of json files
- Data Lake
- Share files within a team

## [Bonus] Storage



# Network

- Cloud is not really a network, but it provides networking services
- Networking services – are the ones that solve your **connectivity** and **isolation** problems
- [Software-defined network](#)



# Network

## Classless Inter-Domain Routing (**CIDR**)

- Used for IP-allocation and routing
- It divides the space using 1-bit steps
- Consists of two parts:
  - the most significant bit
  - network prefix

# Network

CIDR notation	Bits	Range
10.0.0.0/8	00001010.00000000.00000000.00000000	10.0.0.0 – 10.255.255.255
10.1.0.0/16	00001010.00000001.00000000.00000000	10.1.0.0 – 10.1.255.255
10.2.0.0/16	00001010.00000010.00000000.00000000	10.2.0.0 – 10.2.255.255
10.3.0.0/31	00001010.00000011.00000000.00000000	10.3.0.0 – 10.3.0.1
10.3.0.2/32	00001010.00000011.00000000.00000010	10.3.0.2 – 10.3.0.2

CIDR [Calculator example](#)

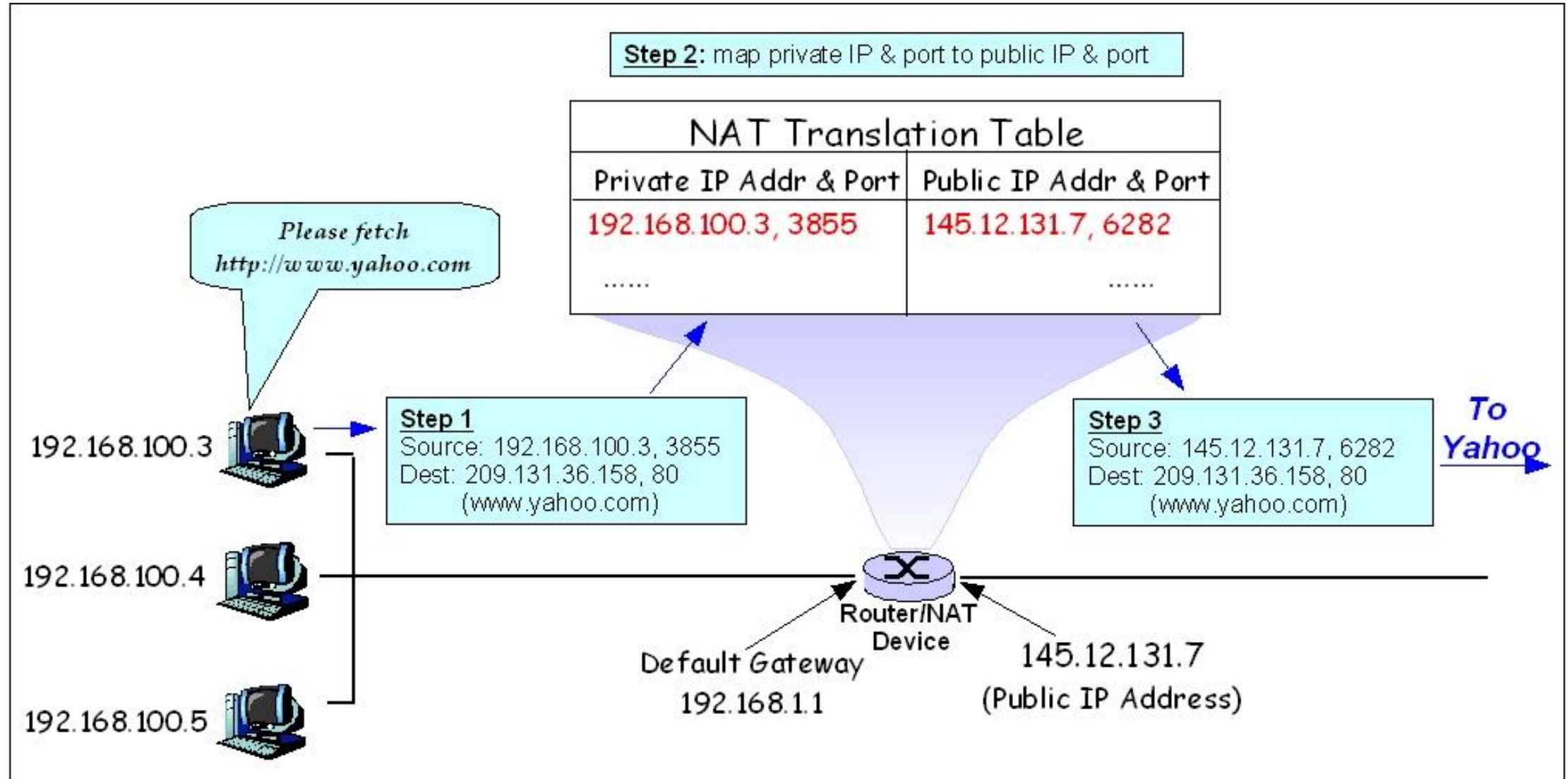
# Network

## Private Network

- a computer network that uses private IP address space
- anyone may use these addresses without approval from regional or local Internet registries
- private network devices communicate with public internet via [Network Address Translation](#) (NAT)

CIDR notation	IP range
10.0.0.0/8	10.0.0.0 – 10.255.255.255
172.16.0.0/12	172.16.0.0 – 172.31.255.255
192.168.0.0/16	192.168.0.0 – 192.168.255.255

# Network



# Network

## Virtual Network

- Like traditional network but in cloud and dedicated only for you
- Has **address space** specified as CIDR block
- Has 1..N **subnets** (CIDR)
- Can have Route table, security rules/groups, public/private endpoints
- Could be **peered** with other Virtual and on-premise networks

Virtual Network / Virtual Private Cloud (VPC) / Virtual Private Cloud

# Network

## Load Balancer (LB)

- distributes network traffic
- could be exposed publicly or private-only
- custom rules, auto-scaling, health-checks, and others
- integrates with other cloud-services

Load Balancer / Elastic Load Balancer (ELB) / Cloud Load Balancer

# Network

## Domain Name System (DNS)

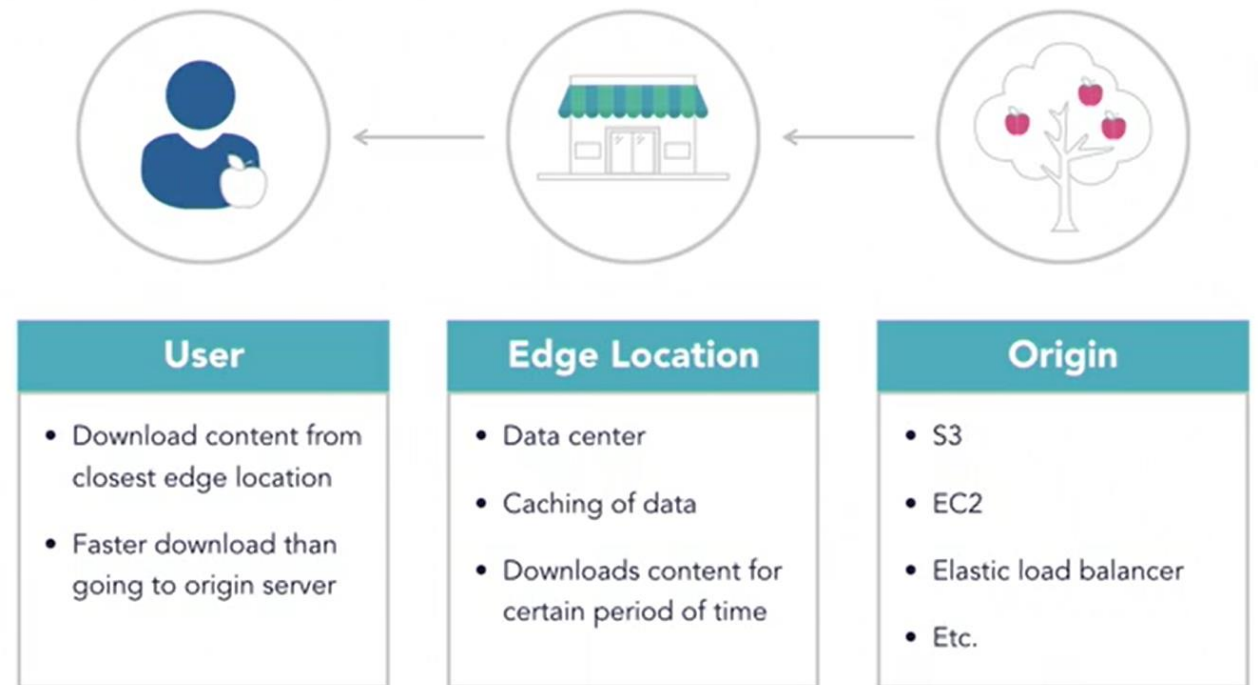
- *translates* names to IP addresses
- can work with cloud-internal resources ...
- ... or with public domain-registrars

Azure DNS / Route 53 / Cloud DNS

# Network

## Content Delivery Network (CDN)

- Caches files on edge locations



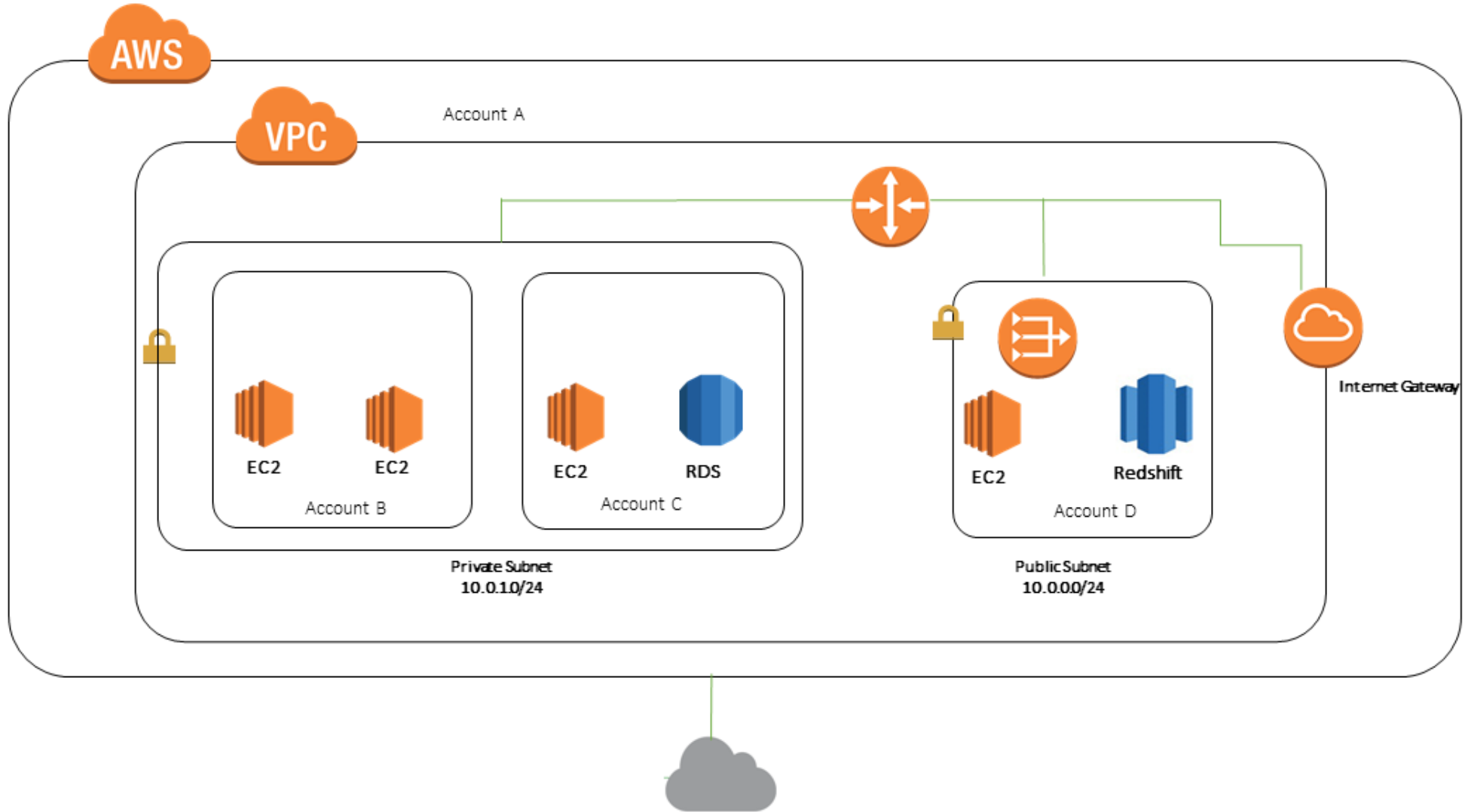
Azure CDN / CloudFront / Cloud CDN



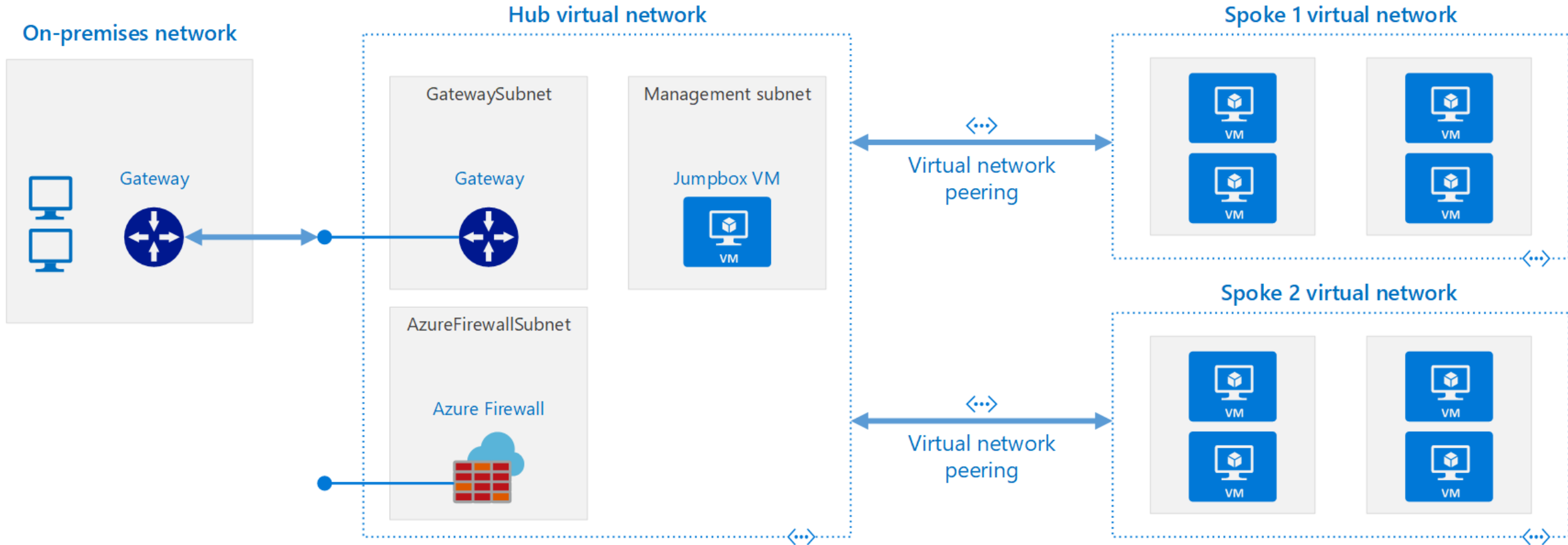
# Network

And a LOT of others, for example [AWS Ground Station](#) or [Azure Orbital](#)

# Network



# Network



# Network Summary

- Networking services solve your **connectivity** and **isolation** problems
- Virtual Networks
  - CIDR notation for address space
  - Split on subnets
  - NAT
  - security rules/groups
- Load Balancer
- DNS
- CDN

# Databases

Cloud Database is ...

- Place where you store and query data
- Managed by cloud provider

# Databases

## Cloud providers

- might propose a custom database ([CosmosDB](#), [Aurora](#), [DynamoDB](#), [BigTable](#), [Spanner](#))
- have product, which is compatible with popular solutions: postgresql, mysql, mongo, cassandra, redis, etc
  - **Compatible does not mean actual product behind.** Thus, read the docs

# Databases

## Cloud Database groups

- Relational
- Non-relational
- In-memory

# Databases

## Relational DBs in cloud

- Azure:
  - [Azure SQL](#) (MS SQL)
  - [Azure Database for MariaDB/MySQL/PostgreSQL](#)
- AWS:
  - [Aurora](#)
  - [Relational Database Service \[RDS\]](#) (Aurora, psql, mysql, mariadb, oracle, mssql)
- GCP:
  - [Cloud Spanner](#)
  - [Cloud SQL](#) (MySQL, PostgreSQL, and SQL Server)



# Databases

## Non-relational DBs in cloud

- Azure:
  - [CosmosDB](#) (with different APIs: mongo, cassandra, sql, gremlin)
  - [Storage Account Table](#)
- AWS:
  - [DynamoDB](#)
  - [DocumentDB](#) (mongo), [Keyspaces](#) (cassandra), [Neptune](#) (graph)
- GCP:
  - [Cloud BigTable](#) (more data)
  - [Firestore](#) (less data)

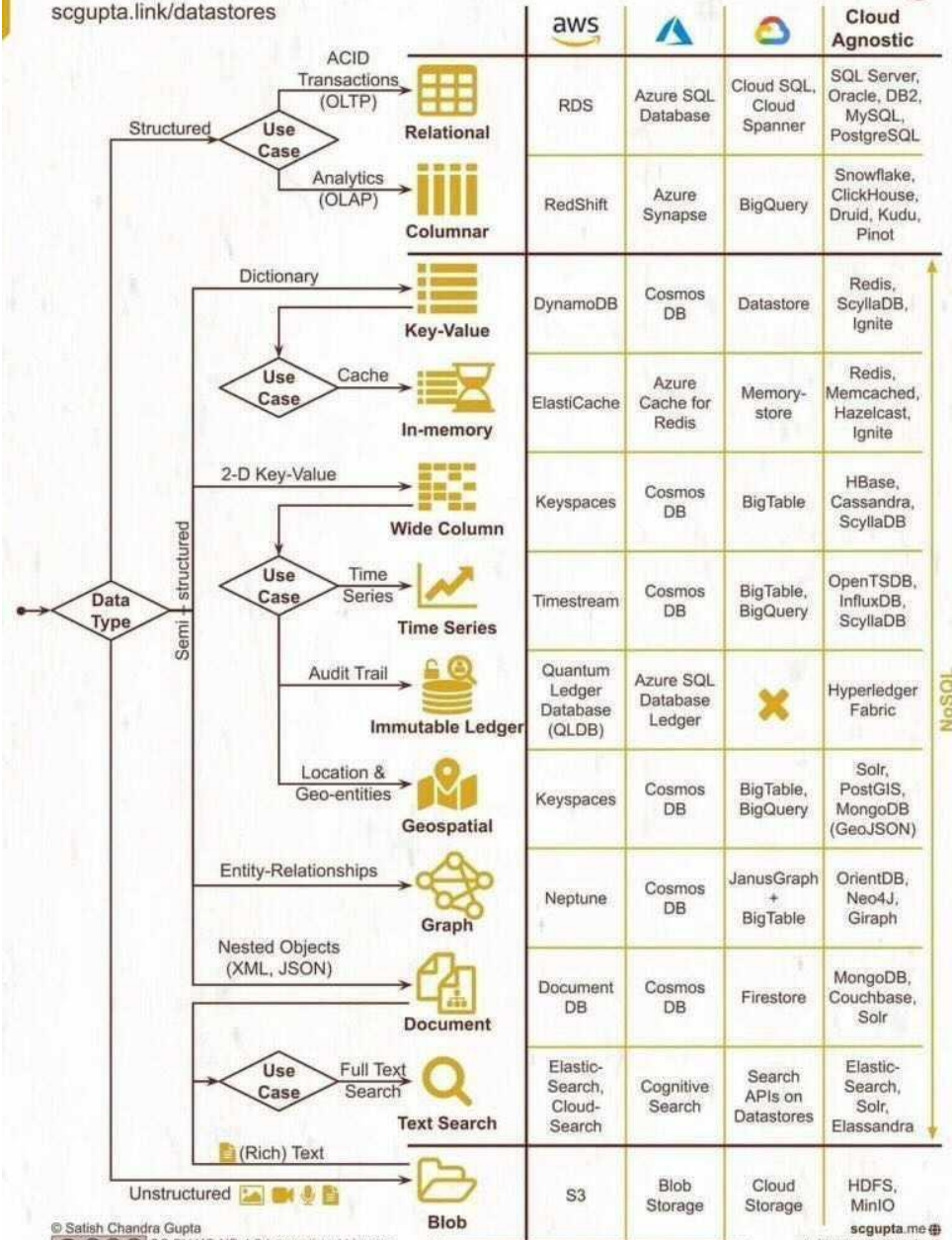
# Databases

## In-memory DBs in cloud

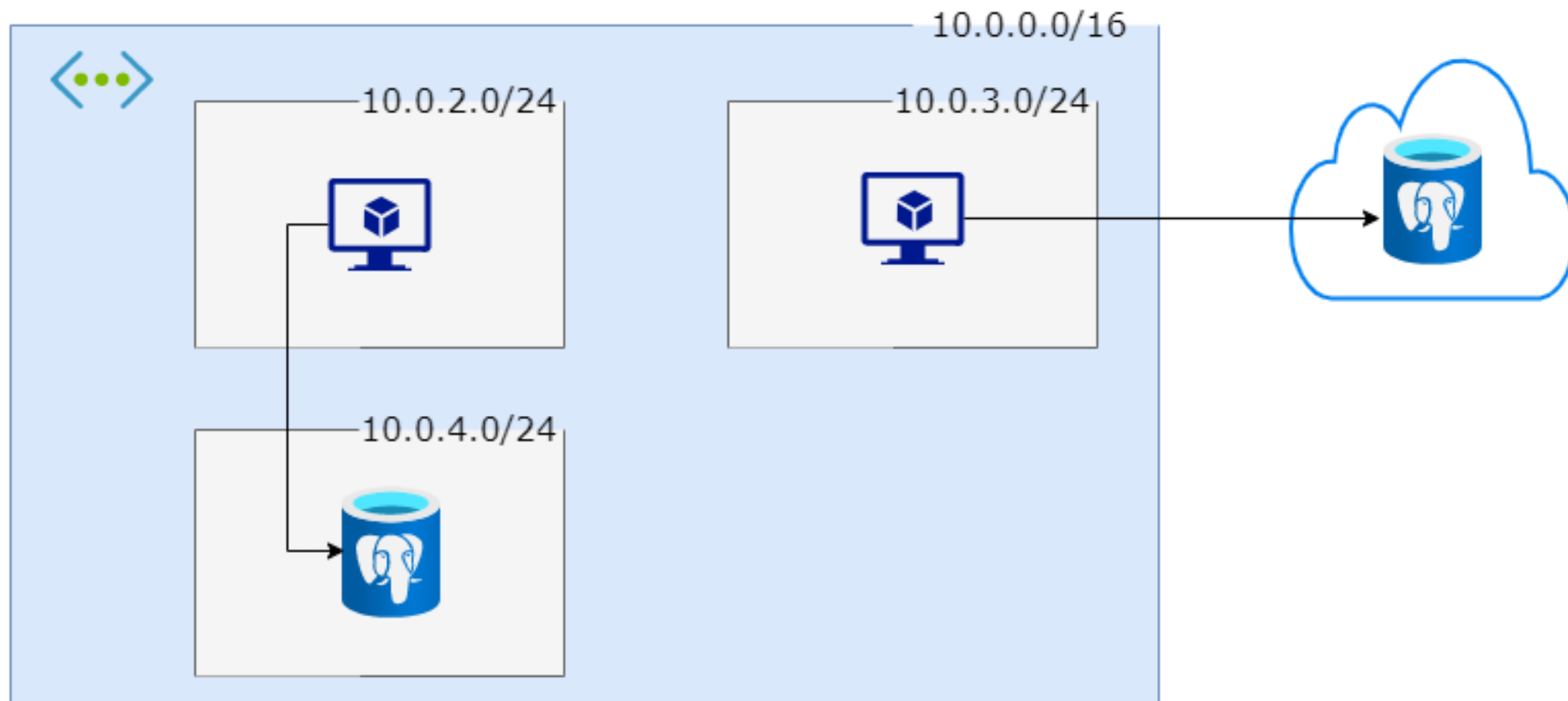
- Azure:
  - [Azure Cache for Redis](#)
- AWS:
  - [Amazon ElastiCache for Memcached/Redis](#)
- GCP:
  - [Memorystore](#) (Redis/Memcached)

# SQL vs. NoSQL: Cheatsheet for AWS, Azure, and Google Cloud

scgupta.link/datastores



# Demo



# Events and Messaging

## Event streams in cloud

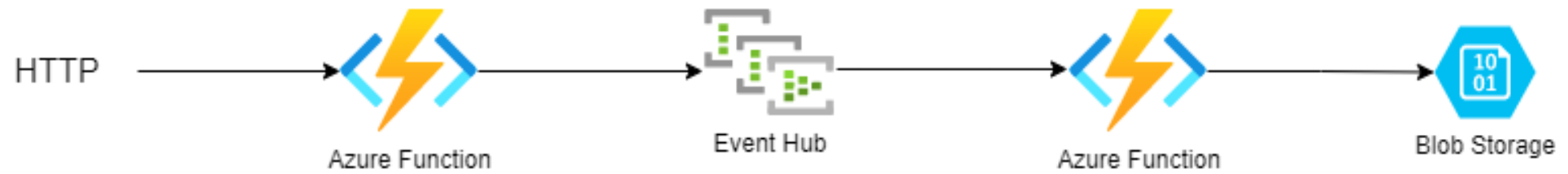
- Azure:
  - [Event Hub](#) (compatible with Kafka)
  - [Stream Analytics](#)
- AWS:
  - [Amazon Kinesis](#)
  - [Managed Streaming for Apache Kafka](#)
- GCP:
  - [PubSub](#)
  - [Dataflow](#)

# Events and Messaging

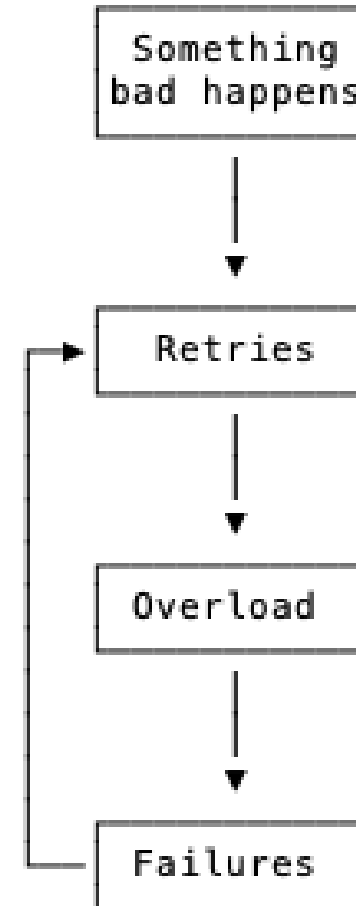
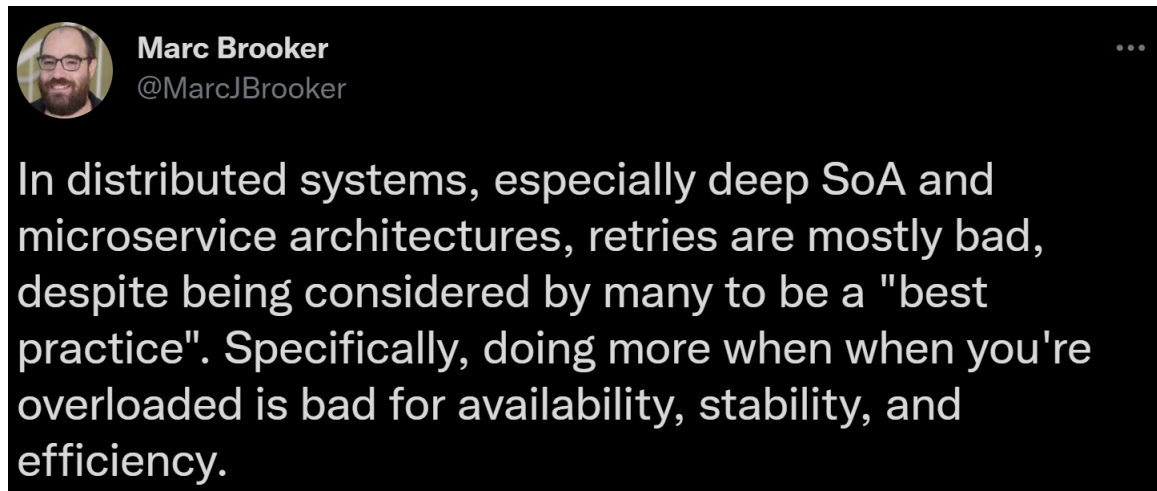
## Messaging in cloud

- Azure:
  - [Service Bus](#)
  - [Storage Account Queue](#)
- AWS:
  - [Simple Queue Service \(SQS\)](#)
  - [Amazon MQ](#) (ActiveMQ and Rabbit)
- GCP:
  - [PubSub](#)

# Demo



# Demo



<https://twitter.com/MarcJBrooker/status/1489651911640825858>



# Additional resources

- (video) [Safe Client Behavior](#)
- (comics) [Networking basics by Julia Evans](#)
- (video) [DynamoDB design-patterns](#) (applies to most “nosql”)

# Additional resources (optional)

- (video) [AWS Networking Fundamentals](#)
- (article) [How DNS works](#)
- (article) [What happens on DNS update](#)
- (article) [What is CDN](#)
- (course) Google's [Introduction to Computer Networking](#)

# Additional resources (optional)

- (article) [Online Event Processing](#)
- (safe-client articles): [bulkhead](#), [circuit-breaker](#), [poison-queue](#)
- (article) [AWS Aurora inside](#)
- (article) [DynamoDB internals](#) (medium paywall)
- (article) [AWS in plain English](#)